Application of Model-based Control in Autonomous Vehicle Chassis Control and Motion Planning

by

Yijun Li

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Mechanical Sciences and Engineering)
in the University of Michigan-Dearborn
2021

Doctoral Committee:

Professor Taehyun Shim, Chair
Assistant Professor Youngki Kim
Dexin Wang, Ford Motor Company
Professor Yi Zhang

Yijun Li

yijunl@umich.edu

ORCID iD: 0000-0002-8559-4690

Dedication

To my family

Acknowledgements

First and foremost, I would want to express my gratitude to Professor Taehyun Shim, who served as my advisor. Working with Professor Shim over the last few years has been a pleasure. His understanding and experience in the controls and automotive dynamics have been tremendously beneficial and informative for my study. I'm grateful for the knowledge he has shared with me, as well as the fruitful and gratifying collaborative relationship we have had over the years. We had numerous intriguing talks throughout this research that generated and shaped many of the concepts offered in this dissertation. Getting to know him and learning from him has been a true pleasure.

I would like to thank Professor Yi Zhang, Assistant Professor Youngki Kim, and Dr. Dexin Wang for serving on the dissertation committee. Professor Zhang provided several helpful comments on the system modeling. Professor Kim's Predictive Control of Dynamic Systems course introduced me to the field of model predictive control. Dr. Wang guided me through my first Ford-UM alliance research project, and provided the test bench for validating my control algorithm. My gratitude also extends to those who worked with me in various research projects. Specifically, Timothy Offerle offered many helpful suggestions on the first alliance project. Lodewijk Wijffels and Joshua Guerra helped guide the second alliance project. I also appreciate Dong-Hwan Shin, Seonghun Lee, and Sungho Jin from DIGST for conducting experimental tests of the electromechanical braking system and providing helpful suggestions on the control system development.

I would like to thank my colleagues, Chinmay Patil, Bhushan Naik, Aditya Sanjay Kothari, Arun Joshi, Heeseong Kim, Syed Adil Ahmed, and Pavan Anandareddy in our research group. I also wish to thank my friends, Wei Guo, Xiaopeng Guo, and Yiteng Tu at the University of Michigan-Dearborn. I cannot imagine completing this long journey without their continuous encouragement and support.

Table of Contents

List of Figures

List of Tables

List of Appendices

Abstract

Automatic control systems are found ubiquitously in today's automobiles. This dissertation focuses on the development of control systems for autonomous vehicle applications using model-based control design approaches. We start with the control system development for chassis actuators. Firstly, we develop a closed loop torque overlay control system to improve the steering feel of an electric power-assisted steering (EPAS) system. This system has a reference model, a rack force estimator, and a tracking controller. A target steering feel is generated from a reference model with an rack force estimate to reflect the actual vehicle operating conditions. The performance of the proposed control system is evaluated through simulation and a hardware-in-the-loop test. Secondly, we present an EMB clamping force control system is developed that addresses several major challenges in practical implementation. A nonlinear EMB model including a novel clamping force model is introduced. A clamping force estimation and contact detection algorithm is proposed that requires only the existing measurements. Furthermore, a unified architecture is proposed to realize a smooth transition between gap closing and clamping force tracking. The performance of the control system is evaluated based on simulation.

In the second part, we developed a hierarchical nonlinear model predictive control (NMPC) framework for autonomous vehicle motion planning and control. At low level, a trajectory tracking NMPC is used to track the reference trajectory given by the high-level motion planner. A frequency shaped objective function is used to incorporate lower level actuator dynamics. Real-time implementation is realized by a fast NMPC algorithm based on RTI with condensing and control parameterization. Simulation results show that the control system yields good performance under various road conditions when the vehicle is operating at its handling limit. A high-level motion planner is formulated in the NMPC framework incorporating high order control barrier function (HOCBF) based collision avoidance constraint. The motion planner is able to generate dynamically feasible trajectories with respect to various constraints. Simulations of the motion planner and trajectory tracking controller demonstrate the effectiveness of the proposed control framework.

Chapter 1

Introduction

1.1 Motivation and Objectives

Automatic control has been playing an important role in automotive innovation since the 1990s. Tremendous efforts have been put into control systems development in the fields of powertrain, driveline, chassis, and recently advanced driver assistance systems, in order to meet the enhanced requirements for lower fuel consumption, lower emissions, and improved safety and comfort demands.

In the area of vehicle chassis control, attention is mainly focused on the development of active safety systems. Various active safety control systems, such as anti-lock braking system (ABS) [1], electronic stability control (ESC) [2], and active front steering (AFS) [3] have been commercially deployed on production vehicles. These standalone control algorithms aim to improve the vehicle's safety and stability by using a particular type of actuator, e.g. brake or steering system. As more and more actuators are available to control and the requirements become increasingly demanding, the amount of work on control systems development and calibration quickly becomes overwhelming. This motivated the development of the integrated vehicle dynamics control (IVDC) architecture in the 2000s [4]. In the IVDC framework, all the vehicle subsystems are coordinated coherently by control algorithms to achieve multiple control functions, such as yaw motion control, lateral motion control, and longitudinal motion control. Control allocation is often used to distribute required chassis force and moment over different actuators.

In recent years, conventional active safety control has evolved into a new generation of control systems, namely the advanced driver assistance (ADAS) systems [5]. The introduction of advanced sensing and computation systems enables new active safety functionalities such as adaptive cruise control (ACC), automatic emergency braking (AEB), and collision imminent steering (CIS).

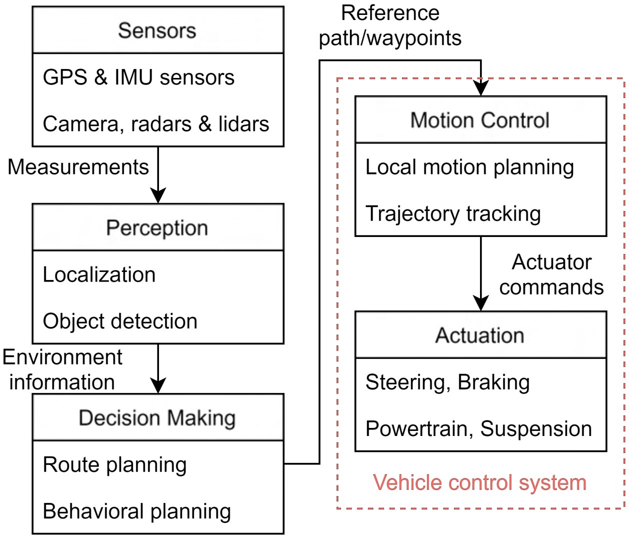


Figure 1.1: Schematic of autonomous vehicle system

The ultimate goal of the active safety system is fully autonomous driving. Autonomous driving not only improves driving safety, but also traffic density and fuel efficiency. Various research institutes and companies have invested a large amount of effort in the research and development of autonomous vehicles. Fig. 1.1 shows a typical schematic of an autonomous vehicle system. The whole system consists of sensors, perception systems, decision making systems, and vehicle control systems [6–8]:

- The system has various types of sensors for measuring the states of ego vehicle and surrounding objects. Some of the most commonly used sensors for autonomous vehicle include GPS, IMU sensors, cameras, radars, and lidars.
- The perception system processes the measurements from various sensors to construct the position of ego vehicle as well as environmental information around the vehicle, such as lane markers, traffic signs and signals, obstacles, etc.
- The decision making system is responsible for planning a route between the vehicle's starting location and destination. The route can be a geometrical path or a set of way points. Based on the planned route and environmental information, the behavioral planning algorithm determines which action the vehicle should take, such as lane keeping, lane changing, overtaking, etc. Finally, reference paths or waypoints are generated for the lower level control system.
- The motion control system is responsible for guiding the vehicle to follow the high level reference path while ensuring driving safety. A hierarchical architecture is usually

adopted to separate the tasks into local motion planning and trajectory tracking. The local motion planning generates a vehicle's motion trajectory over a horizon based on the reference path, vehicle states, and surrounding objects. Then the trajectory tracking controller computes the required actuation like steering angle, braking/driving force, etc, such that the vehicle follows closely to the reference trajectory.

- The actuation system consists of different low-level actuators such as steering, braking, powertrain, and suspension systems. It takes the commands from the motion control system and applies appropriate control input to each individual system to track the reference command.

In this dissertation, we aim to develop control systems for the motion control layer and the actuation layer. In particular, model-based control design methodology is used for control algorithm development. Traditionally, in the automotive industry, model-free control techniques are often applied. These methods are based on the time-consuming iterative calibrations of lookup tables, PID controller gains, and so on. This tuning methods might work well for standalone control algorithms like ABS or ESC, but quickly become intractable when dealing with complex chassis control such as the motion control system in autonomous driving applications.

On the contrary, model-based control design is a more systematic approach that relays on knowledge of system dynamics for controller development. Controllers are synthesized based on the control-oriented model and several weightings and tuning parameters. Simulations are used to evaluate the control performance, and the controller can be retuned conveniently by changing the corresponding weights and parameters based on the simulation results.

This dissertation presents our work on how model-based design approaches can be used for control system development of autonomous vehicle. To be specific, we try to stress how several important requirements can be addressed when designing model-based control systems for autonomous vehicle:

- Control structure that is easy to tune: An increasing number of algorithms are added to the existing base vehicle control systems to realize additional functionalities. The designed control algorithm should allow fast and convenient tuning to achieve target performance. For example, the control algorithm for the electric power-assisted steering (EPAS) system should enable flexible adjustment for steering feel such that the car maker can distinguish its products of different classes, e.g. family, luxury, sport, etc. The motion planning algorithm should be able to generate vehicle motions according to personal driving style, e.g. comfort, aggressive, etc. In this dissertation, we will see how reference model based torque overlay control is used for convenient adjustment of target steering feel on the basis of existing power assist functions. In electromechanical brake (EMB) control, we show how clamping force tracking bandwidth and gap closing speed can be adjusted by a couple of intuitive parameters. And for motion planning and control for autonomous driving, control performance can be tuned by relative weights of control objectives versus control inputs.

- Control robustness with respect to disturbances and model uncertainties: Vehicle control systems frequently experience various external disturbances and uncertainties, e.g. mechanical friction, measurement noise and delay, during operations. The controller's performance should be robust against these factors. In the development of EPAS steering feel control, we use sliding mode controller to realize robust reference tracking against external disturbance. In the development of EMB clamping force control, we will see a disturbance-observer based controller is used to find the balance between fast tracking with disturbance rejection and robust stability against model mismatch.

- Estimation of system parameters and states using limit sensor measurements: Due to hardware and cost limitations, some model parameters and states necessary for control system are not available by sensor measurement. Estimation techniques must be used to reconstruct these information. In the first part of the dissertation, we will see how pragmatic estimation algorithms are developed to effectively estimate some essential model parameters and states such as EPAS rack force, EMB clamping force and gap distance, for the control systems.

- Interfacing between high-level and low-level control systems: Controllers in different layers of the vehicle control system are often designed separately with different specifications and assumptions. For example, the actuator controllers in the actuation layer usually operate at a much faster rate than those in the motion control layer. And the motion control system might be developed based on simplified actuation models with some assumptions. Therefore, controllers in different layers should work cohesively in order to achieve good overall control performance. In this work, we show how the trajectory tracking controller takes into account of actuator dynamics using frequency shaping technique, how temporal based trajectory tracking nonlinear model predictive control (NMPC) and spatial based motion planning NMPC work together, and how the motion planner takes care of commands from high-layer decision making systems by formulating them into various path constraints.

- Real-time feasible controller design: Optimization-based controllers like the one proposed in the second part of this dissertation are usually computationally intensive, since optimization must be solved online in real-time. We show how specific numerical methods can be applied with control objectives and system characteristics in mind, such that real-time feasible control is realized.

1.2 Contributions

The major contributions of this dissertation are summarized as follows:

EPAS Steering Control

We propose a torque overlay control structure that achieves a flexible and intuitive steering feel adjustment without tedious calibrations on feed-forward gains and lookup tables. Specifically, a high-fidelity EPAS model reflecting realistic dynamics of the system is developed. In particular, load-dependent fiction is included in the model and its effect on steering feel is investigated. We propose a novel algorithm for load-dependent fiction estimation. By incorporating it with a Kalman filter, accurate rack force estimation is achieved. The rack force estimate is then used for the reference model to generate a realistic target steering feel. Hardware-in-the-loop simulations are conducted which demonstrate the effectiveness of the proposed estimation and control algorithm.

EMB Clamping Force Control

A novel clamping force model validated by testing results, which is able to accurately capture the nonlinearity and hysteresis effect. A straightforward process is also presented for estimating the clamping force. We develop a simple and effective gap distance estimator based on the Kalman smoother and a robust clamping force tracking controller based on the disturbance observer. In order to seamlessly integrate the gap closing and force tracking without a controller switching logic, a unified clamping force control scheme is designed. Extensive simulations are conducted to verify the effectiveness and robustness of the proposed control system.

NMPC based Trajectory Tracking Controller

A NMPC based trajectory tracking controller for vehicles equipped with four-wheel-steering (4WS) and four-wheel-independent-driving (4WID) are developed. In particular, a frequency shaped objective function is used for NMPC formulation to accommodate the dynamics of low-level actuators. A fast NMPC solving algorithm based on real-time-iteration (RTI) scheme with control parameterization is proposed to achieve a fast sample time with real-time feasibility. The performance of the proposed controller on vehicle stabilization and trajectory tracking is demonstrated by various simulations.

NMPC based Motion Planner for Autonomous Vehicle

A NMPC based motion planner for autonomous vehicle is developed. The motion planning problem is formulated in the spatial domain to enable simple constraint formulations. The NMPC simultaneously plans lateral and longitudinal trajectories of the vehicle that satisfy various constraints on control inputs, vehicle states, waypoints, and collision avoidance. We proposed a novel collision avoidance formulation in the NMPC framework using high-order control barrier functions (HOCBF). The HOCBF based collision avoidance constraint provides formal certification of safety in the sense of set invariance, which is able to ensure safety even with short prediction horizon. In order to allow flexible constraints on waypoints with fast computation, NMPC problem is solved based on the direct collocation method.

1.3 Dissertation Outline

Fig. 1.2 shows the layout of this dissertation. The dissertation consists of 6 chapters, which are organized as follows. Referring to Fig. 1.2, the main 4 chapters (Chapter 2 to Chapter 5) are divided into 2 parts. Part I includes Chapters 2 and 3, focusing on the control and estimation development for low-level vehicle chassis system. In Chapter 2, the development of EPAS steering feel control system is discussed. Chapter 3 discusses the clamping force control system for EMB. Part II consists of Chapters 4 and 5, containing the development of motion control system for autonomous vehicles. Chapter 4 describes the NMPC trajectory tracking control design. And Chapter 5 discusses the details of motion planning NMPC development. Finally, Chapter 6 summarizes the outcomes of the preceding chapters and concludes the study. Possible future work is discussed as well.

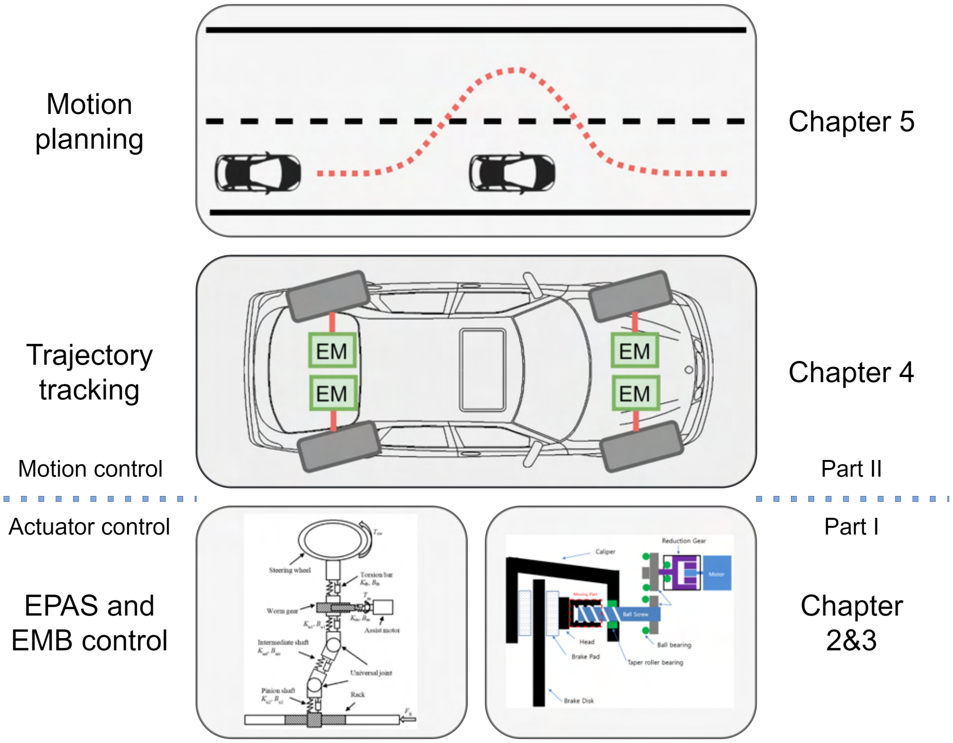


Figure 1.2: Dissertation layout

Part I

Control and Estimation for Vehicle Chassis System

Chapter 2

Steering Feel Control for EPAS System

The steering system is a key component of an automobile, affecting the vehicle handling and stability and the driver's comfort. It is also a key actuator for delivering the advanced driver assistance or autonomous driving functions. However, the conventional power assist control involves extensive tuning effort to meet the desired steering feel requirements. This chapter aims to develop a steering feel control system for EPAS based on torque overlay. The proposed control algorithm is able to realize the target steering feel given by a reference model. Different steering feels can be achieved by simple tuning of the reference model parameters without modifying the power assist functions.

EPAS system is also an essential part for autonomous vehicle. When the driver's input is absent as in autonomous vehicle, the assist motor of EPAS system is responsible for generating required steering angle to control the vehicle's motion. We will see from the development process that the EPAS steering feel control shares many common challenges with autonomous vehicle position control in terms of system modeling, disturbance estimation, and control. And the outcomes of this chapter can be straightforwardly applied to autonomous vehicle application.

2.1 Introduction

Electric power-assisted steering (EPAS) has become widely used in the automotive industry to replace the conventional hydraulic power assist steering (HPAS) systems because of its efficiency, modularity, and flexibility [9]. The EPAS system uses an electric motor rather than a hydraulic pump to provide assist torque such that the driver's steering effort is reduced. Moreover, EPAS provides more freedom than the HPAS system to achieve advanced driver assist functions, such as lane-keeping assist, lane-change assist, automatic parking, etc.

Despite the rapid development of advanced steering control systems, a fundamental

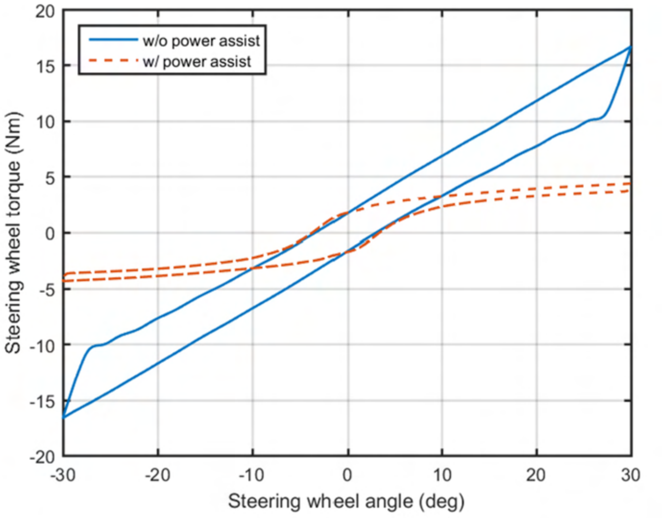


Figure 2.1: Steering feel curve with and without power assistance

issue has continued to trouble developers and engineers since the appearance of the EPAS system—the steering feel. Steering feel is defined as the haptic feedback from the steering wheel to the driver's hand. It originates from the tires' self-aligning moment while steering and conveys real-time information on the vehicle's dynamic status to the human driver. Thus, a good steering feel delivered by the EPAS system is critical for the driver's maneuvering of the vehicle [10]. However, good steering feel is not easy to achieve.

Steering feel is commonly evaluated by on-center handling test, where the steering angle is small and at low frequency and the vehicle is at high speed [11]. This characterizes typical highway situation, where lane-keeping and lane-changing are the primary maneuvers. As a important index of the on-center handling quality, steering feel represents the torque feedback to the driver with regard to the steering angle. The steering feel of Column-EPAS (C-EPAS) can be depicted by hysteresis curves of steering torque versus steering wheel angle. Figure 2.1 shows a example of steering feel curve of C-EPAS. Roughly speaking, the objective of steering feel control algorithm is to effectively shape this curve according to vehicle's driving conditions such that target steering feel is achieved.

The conventional methods for steering feel control are based on feed-forward approaches.The feed-forward approaches usually adopt different look-up tables and filters to calculate the assist torque. For example, Ziman He [9] developed three steering control modes as well as various compensation controls based on different boost curves. In [12], the assist torque is calculated by nonlinear functions so that the yaw rate oscillation is well damped and steering wheel returnability is improved. Yamazaki et al. [13] designed a steering torque compensation strategy to cancel the influence of vehicle dynamics on steering torque. One advantage of feed-forward control is that it is an analogue of the conventional HPAS control method, and thus its implementation is straightforward. Two main issues, however, limit the application of the feed-forward control methods. On the one hand, however, the development of feed-forward control typically involves the tuning of a blend of look-up tables and compensators in a range of vehicle speeds in order to achieve the target steering feel. Therefore, the development procedure is extremely time-consuming, often requiring the iterative tuning of the controller and target steering feel. On the other hand, feed-forward control lacks robustness against model uncertainties and disturbances. Since the assist torque is determined in feed-forward fashion, the resulting steering feel is subject to change in the presence of parameter variations, e.g., inertia and road friction, and external disturbance, e.g., road disturbance force and friction.

Distinct from the feed-forward methods, the feedback approaches control the steering feel in a closed-loop scheme such that the response of the EPAS system tracks the target steering feel. For instance, Tao [14] applied admittance control to an EPS system such that the steering system has the desired admittance. Alaa Marouf et al. [15] designed a controller for an EPS system based on the reference model to make the motor angle track the ideal one. Carlos et al. [16] presented a position-model reference control strategy for an EPS system. M. Moradkhani [17] et al. employed an H_∞ loop-shaping control method in the EPS system steering torque control. The feedback methods have two main advantages. Firstly, the control system is more robust in the presence of model uncertainties and disturbances. Therefore, the steering feel is kept the same regardless of the vehicle parameters and operation conditions. Another advantage is that the design of the feedback controller is independent from the target steering feel. Thus, tremendous tuning effort is saved.

Based on the preceding discussion, the feedback method is clearly more promising for EPAS steering feel control. However, there are still open questions regarding the design of feedback steering feel control. The steering feel of EPAS with feedback control is realized by tracking a reference signal. Therefore, the choice of reference signal directly affects the steering feel. The reference signal can be generated statically or dynamically. The static reference signal is usually calculated from boost maps calibrated based on experiments [18], while the dynamic reference is derived from some reference model that depicts the target steering feel characteristics [14]. Nevertheless, neither method is able to produce ideal steering feel under all operation conditions. On the one hand, the boost map represents only a steering feel that is predetermined and cannot adapt to different operation conditions. On the other hand, the artificial steering feel from a dynamic model is unrealistic to the driver due to the discrepancy between the model and the real plant.

To address the aforementioned problem of feedback steering feel control, the authors propose a torque overlay steering feel control algorithm based on a feedback control method. In the proposed system, a reference model is used to generate a reference steering feel where it uses a real-time rack force estimated from a rack force estimator and a tracking controller is employed to drive the real system variable to the reference one by overlay torque.

The remainder of this chapter is organized as follows: firstly, the Column-Electric Power Assist System (C-EPAS) plant model is presented in Section 2. Section 3 discusses the torque overlay control system. In Section 4, the rack force estimation algorithm is elaborated, which is needed in both reference model and tracking controller. The performance evaluation of the control system is presented with simulation and hardware-in-the-loop testing in Section 5. Finally, the conclusion is given in Section 6.

2.2 EPAS System Modeling

In this section, a detailed mechanical model of a C-EPAS system is built that includes important dynamics affecting the steering feel, such as the worm gear, intermediate shaft, pinion gear, and rack housing. This model is used as a plant model for validation of the estimation and control algorithm in the later sections. The C-EPAS system detects driver input torque from the torsion bar, and based on that the ECU sends a torque command to a DC motor to generate assistance toque. The assistance torque along with the driver's torque is converted to a rack force via the intermediate shaft and the rack and pinion gear to overcome the reactive force from the tie rods.

2.2.1 Mechanical Model

For modeling the C-EPAS, the steering wheel, motor armature, worm gear, intermediate shaft, pinion shaft, and rack are considered. The schematic of the C-EPAS model is shown in Fig. 2.2.

Equations (1) and (2) describe the dynamics of the steering wheel and the assist motor, respectively.

$$T_{sw} - B_{tb}\left(\dot{\theta}_{sw} - \dot{\theta}_g\right) - K_{tb}\left(\theta_{sw} - \theta_g\right) = J_{sw}\ddot{\theta}_{sw}, \tag{2.1}$$

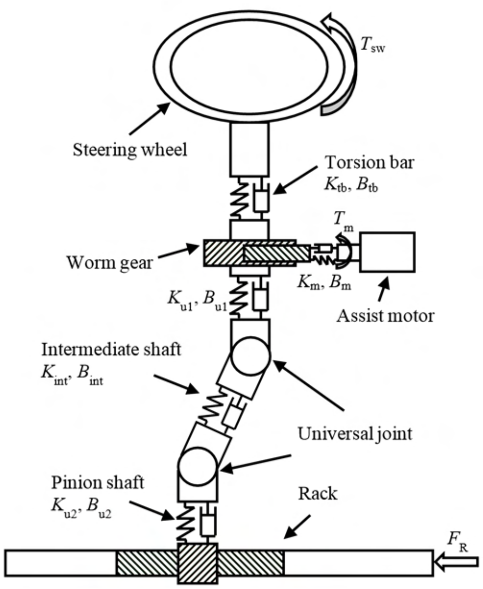


Figure 2.2: Schematic of C-EPAS model

$$T_m - B_{tb}\left(\dot{\theta}_m - \dot{\theta}_w\right) - K_m\left(\theta_m - \theta_w\right) = J_m\ddot{\theta}_m, \tag{2.2}$$

where T_{sw}, θ_{sw}, and J_{sw} are steering wheel torque, angle, and inertia, respectively. T_m, θ_m, and J_m are motor rotor torque, angle, and inertia, respectively. K_{tb} and B_{tb} are torsion bar stiffness and damping, respectively. K_m and B_m are motor rotor stiffness and damping, respectively. θ_g and θ_w are gear and worm angle, respectively.

The dynamics of worm and worm gear are depicted by

$$B_m\left(\dot{\theta}_m - \dot{\theta}_w\right) + K_m\left(\theta_m - \theta_w\right) - T_{bf,w} + F_{wx}r_w = J_w\ddot{\theta}_w, \tag{2.3}$$

$$B_{tb}\left(\dot{\theta}_{sw} - \dot{\theta}_g\right) + K_{tb}\left(\theta_{sw} - \theta_g\right) - B_{u1}\left(\dot{\theta}_g - \dot{\theta}_{u1,i}\right) + K_{u1}\left(\theta_g - \theta_{u1,i}\right) - T_{bf,g} + F_{gy}r_g = J_g\ddot{\theta}_g, \tag{2.4}$$

where $T_{bf,w}$ and $T_{bf,g}$ are motor shaft and gear shaft bearing friction, respectively. F_{wx} and F_{gy} are worm and gear force, respectively. r_w and r_g are worm and gear radius, respectively. J_w and J_g are worm and gear inertia, respectively.

Figure 2.3 illustrates the schematic of the intermediate shaft and u-joint. The equations for the intermediate shaft are

$$T_{u1,o} - T_{int} = J_{u1}\ddot{\theta}_{u1,o}, \tag{2.5}$$

$$T_{int} - T_{u2,i} = J_{u2}\ddot{\theta}_{u2,i}, \tag{2.6}$$

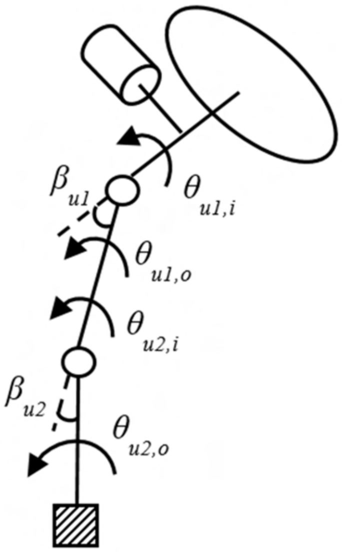


Figure 2.3: Schematic of intermediate shaft

$$T_{u1,o} = R^{-1}\left(\theta_{u1,i}\right)\left[B_{u1}\left(\dot{\theta}_g - \dot{\theta}_{u1,i}\right) + K_{u1}\left(\theta_g - \theta_{u1,i}\right)\right], \tag{2.7}$$

$$T_{u1,i} = R\left(\theta_{u2,i}\right)\left[B_{u2}\left(\dot{\theta}_{u2,o} - \dot{\theta}_p\right) + K_{u1}\left(\theta_{u2,o} - \theta_p\right)\right], \tag{2.8}$$

$$T_{int} = B_{int}\left(\dot{\theta}_{u1,o} - \dot{\theta}_{u2,i}\right) + K_{int}\left(\theta_{u1,o} - \theta_{u2,i}\right), \tag{2.9}$$

where $T_{u1/2,i/o}$ and $\theta_{u1/2,i/o}$ are upper/lower universal joint input/output torque and angle, respectively. T_{int} is intermediate shaft torque. J_{u1} and J_{u2} are upper and lower intermediate shaft inertia. K_{u1} and B_{u1} are stiffness and damping of worm gear shaft, respectively. K_{int} and B_{int} are stiffness and damping of intermediate shaft, respectively.

The transmission ratio of the universal joint R is calculated byThe transmission ratio of the universal joint R is calculated by

$$\frac{\dot{\theta}_o}{\dot{\theta}_i} = R\left(\theta_i\right) = \frac{\cos\beta}{1 - \sin^2\beta\cos^2\theta_i}, \tag{2.10}$$

where β is universal joint bending angle.

The following two equations represent the dynamics of pinion and rack, respectively.

$$B_{u2}\left(\dot{\theta}_{u2,o} - \dot{\theta}_p\right) + K_{u1}\left(\theta_{u2,o} - \theta_p\right) - T_{bf,p} + F_{py}r_p = J_p\ddot{\theta}_p, \tag{2.11}$$

$$F_{py} - F_{tl} - F_{f,r} = m_r\ddot{y}_r, \tag{2.12}$$

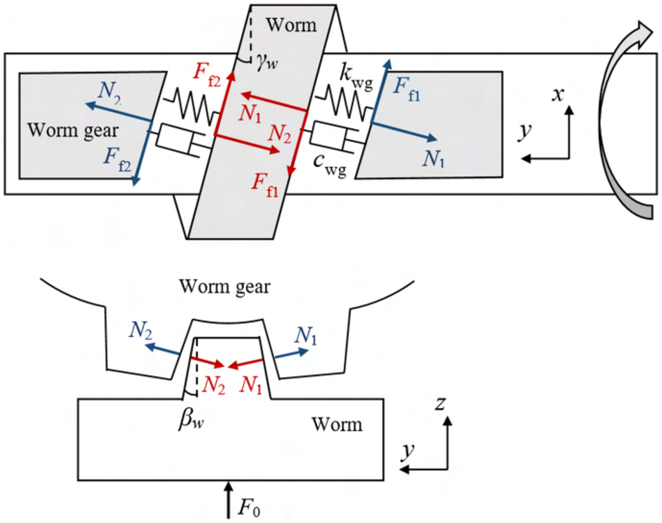


Figure 2.4: Schematic of worm gear

where θ_p is pinion shaft angle. K_{u2} and B_{u2} are stiffness and damping of pinion shaft, respectively. $T_{bf,p}$ is pinion shaft bearing friction. F_{py} is pinion gear force. r_p is pinion gear radius. J_p is pinion shaft inertia. F_{ry} is rack gear force. F_R is rack force from tie rod. $F_{f,r}$ is rack friction. m_r is rack mass. y_r is rack displacement.

As the key component of a C-EPAS system, modeling of the worm gear is of great importance to obtaining a good C-EPAS model. In general applications, an intentional backlash is usually provided in the worm gear mechanism to avoid teeth jamming and reduce unnecessary frictional losses. However, considering the frequent change of transmission direction of the worm gear in C-EPAS, a preload mechanism is adopted such that the rattling noise caused by backlash can be reduced during gear inverse rotation. Such a structure introduces additional mesh friction into the system whose value increases with increasing load. In order to capture such a feature in a C-EPAS system, the worm gear contact force is represented by a wedge-like model including worm gear compliance [19, 20]. The schematic of the worm gear model is shown in Fig. 2.4.

The normal contact force between worm and worm gear in Eqns. (2.3) and (2.4) can be calculated as a linear spring and dampers

$$N = k_{wg}\delta + c_{wg}\dot{\delta}, \tag{2.13}$$

where k_{wg} and c_{wg} are worm gear contact stiffness and damping, respectively. The deflection

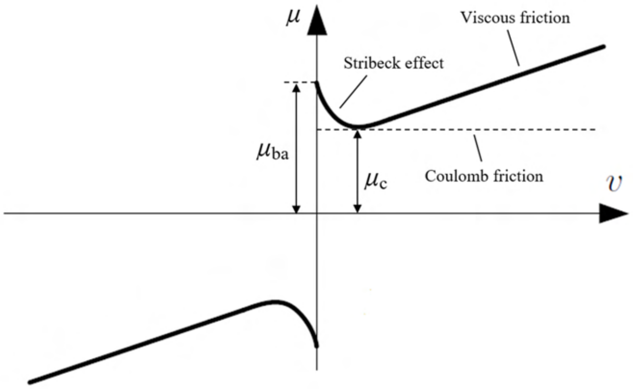


Figure 2.5: LuGre fiction characteristics

of gears can be calculated as

$$\delta = \theta_w r_w \sin\gamma_w - \theta_g r_g \cos\gamma_w + \delta_0, \tag{2.14}$$

where δ_w is worm gear lead angle. δ_0 is the initial deflection caused by worm gear preload F_0,

$$\delta_0 = \frac{F_0}{2k_{wg}\sin\beta_w}. \tag{2.15}$$

The mesh friction of the worm gear is calculated as

$$F_f = \mu N, \tag{2.16}$$

where the friction coefficient μ is modeled as a LuGre model [21]

$$\mu = \sigma_0 z + \sigma_1 \dot{z} + \sigma_2 v, \tag{2.17}$$

$$\dot{z} = v - \sigma_0 \frac{|v|}{g\left(v\right)} z, \tag{2.18}$$

$$g\left(v\right) = \mu_c + \left(\mu_{ba} - \mu_c\right) e^{-\left(v/v_s\right)^2}, \tag{2.19}$$

where σ_0, σ_1 are bristle stiffness and damping, respectively. σ_2 is viscous damping coefficient. μ_c and μ_{ba} are Coulomb and static friction, respectively. v_s is Stribeck velocity. Fig. 2.5 shows the static characteristics of the LuGre friction model.

Based on Eqns. (2.13)–(2.19), the contact force of the worm gear can be calculated as

$$F_{wx} = -N \cos \beta_w \sin \gamma_w - F_f \cos \gamma_w, \tag{2.20}$$

$$F_{gy} = N \cos \beta_w \cos \gamma_w - F_f \sin \gamma_w, \tag{2.21}$$

where β_w is the pressure angle of the worm gear.

The contact force of the rack and pinion gear is also modeled in the same fashion. It should be noted that the worm gear and the rack and pinion gear are modeled under the assumption of single-tooth contact, while the worm gear normally has higher contact, meaning that there are times when more than one tooth is in contact. The variation in the number of teeth in contact might introduce position-dependent friction in the gears. It has been observed that in transmission systems there exists position-dependent friction, which has oscillation characteristics [22] For a C-EPAS system, the position-dependent friction originates from the gear stiffness variation of the worm gear and the rack and pinion gear transmission.

The position-dependent friction of the C-EPAS system is modeled as a sine wave, shown in below, whose amplitude and frequency is dependent on the external load and the gear ratio.

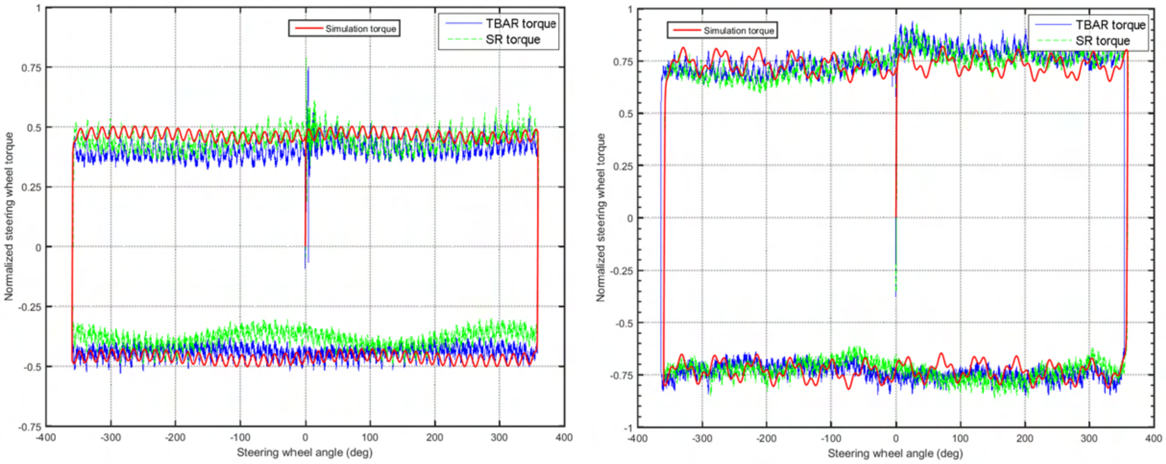
$$\tau_f(\theta) = A \sin(\omega\theta + \varphi). \tag{2.22}$$

In addition to the mesh friction of the worm gear and the rack and pinion, the friction of a C-EPAS system also originates from the bearings of the worm gear, pinion shaft, and rack housing. These frictions are also modeled by a LuGre model assumed to be independent of load and position.

The overall friction level is examined by simulation test. Figure 2.6 shows the comparison between simulation and test results. Figure 2.6(a) is the comparison result when the pinion and rack are detached from the steering column, and Fig. 2.6(b) is the result from the complete C-EPAS system. The simulation is carried out under constant steering wheel angle input at different rates without load. It can be seen that the friction level of our model matches pretty well to the test results.

2.2.2 Basic Control Strategy

The conventional power assistance functions have been embedded in the C-EPAS by the supplier to realize basic steering feel control. Figure 2.7 shows a block diagram of the basic steering control strategy used in this study. There are three basic control systems: power



(a) Steering column only

(b) Whole steering system

Figure 2.6: Friction level validation

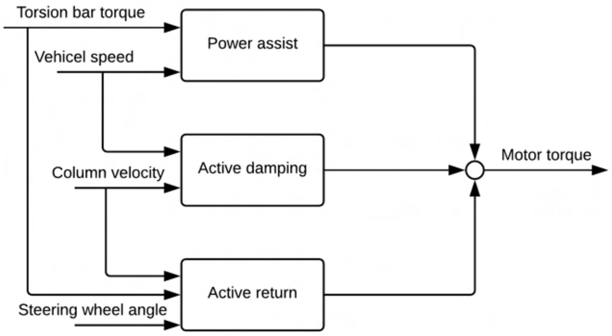


Figure 2.7: Block diagram of C-EPAS assist control strategy

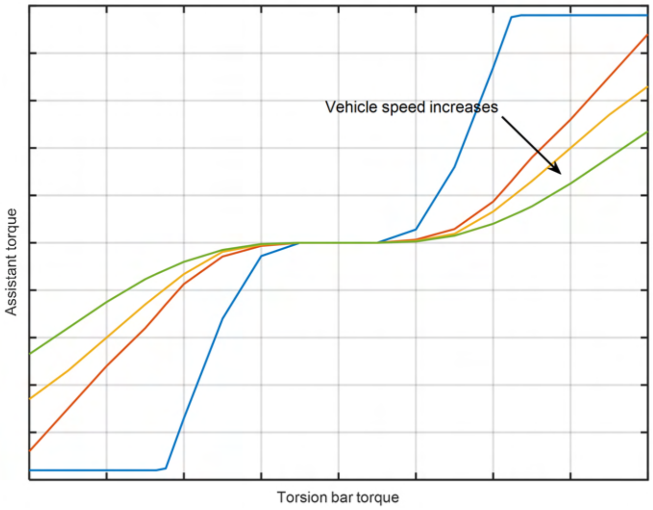


Figure 2.8: Boost curve for power assistance

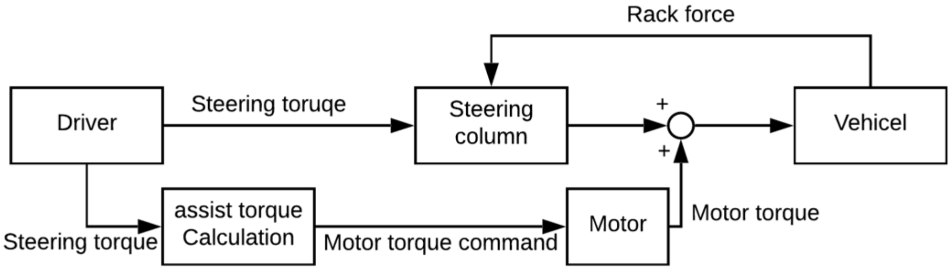


Figure 2.9: Schematic of Feed-forward steering feel control strategy

assistance, active damping, and active return control [12]. The role of power assistance is to reduce the steering effort, the active damping is to improve the steering stability, and the active return function helps the steering wheel return to center when a driver releases the steering wheel. The power assistance torque is calculated according to the boost curve shown in [23], and active return control is released by a PI controller. Figure 2.8 shows a example of boost curve for power assistance.

The basic power assistance is implemented as feed-forward torque injected to the C-EPAS system, as shown in Fig. 2.9. In Fig. 2.9, the motor torque is added to the EPAS system in feed-forward fashion calculated based on lookup tables. Although this method is easy to implement, it requires tedious parameters tuning on the parameters of feed-forward functions and lookup table. Another drawback of such a method is the lack of robustness against parameter variation, disturbance, etc.

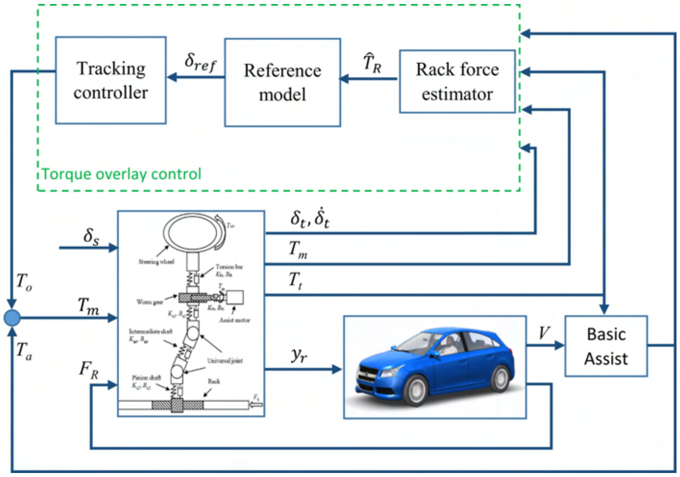


Figure 2.10: Overall schematic of C-EPAS steering feel control

2.3 Torque Overlay Control System

In this section, we introduce the EPAS steering feel control algorithm based on torque overlay. The torque overlay control is built upon the existing conventional power assistance algorithms. Figure 2.10 shows a schematic of the C-EPAS control system. This system consists of a reference model, tracking controller, and rack force estimator. The estimated real-time rack force from the rack force estimator is used to the reference model to set a reference pinion angle and angle rate. Then, the actual pinion angle is regulated by the tracking controller to follow the reference value. The following sections discuss in detail for individual component.

Although steering feel control is our focus in this chapter, we want to stress that the control structure in Fig. 2.10 can be applied to steering position control for autonomous vehicle application by simply disable the reference model, and replace the source of reference signal δ_{ref} to the steering angle command from high-level motion control system, e.g. the NMPC controller in Chapter 4. Moreover, the rack force estimation and friction estimation developed later in Section 3 can be used as feed-forward compensations to improve the control performance.

2.3.1 Reference Model

The reference model provides a target steering feel that the tracking controller must follow with overlay torque. The target steering feel can be generated by various approaches such as lookup table [18], spring and damper [24], and dynamic model [14].

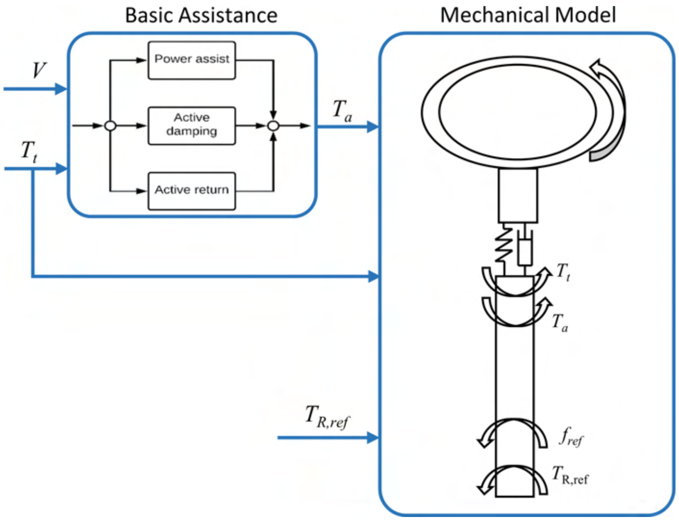


Figure 2.11: Schematic of reference model with C-EPAS and basic control systems

In our study, a joint reference model is proposed. This joint model combines basic assistance functions with a reduced-order EPAS mechanical model. The mechanical model of EPAS provides an intuitive basis for tuning the steering feel with physically-related parameters such as friction and damping coefficients, and the basic assistance module takes advantage of the conventional feed-forward steering feel control method so that the resulting reference steering feel is familiar to the driver. Figure 2.11 shows the schematic of the reference model.

The dynamics of the reference model is given as

$$T_t - T_{f,ref} - T_{R,ref} + T_a = I_{ref}\ddot{\delta}_{ref}. \tag{2.23}$$

In the above reference model, the steering system is considered as one stiff bar with assuming that the inertias and frictions at different location are lumped together on the column compared to the high order EPAS model. The inputs of the reference model are torsion bar torque T_t, friction torque$T_{f,ref}$, rack load torque $T_{R,ref}$, and assist torque T_a. Using these inputs, the reference steering column angle δ_{ref} is computed from Eq. (2.23). With torsion bar sensor, the torsion bar torque can be measured and the assist torque is also accessible from the basic assist control system. Since there is no sensor to measure directly the rack reaction torque in production vehicles, a rack force emulation based on a rack force model and vehicle states has been widely used [25]. Although this approach is straightforward, the major downside is that realistic rack force is difficult to obtain due to modeling error. In addition, the emulated rack force is unable to capture the change of rack force caused by

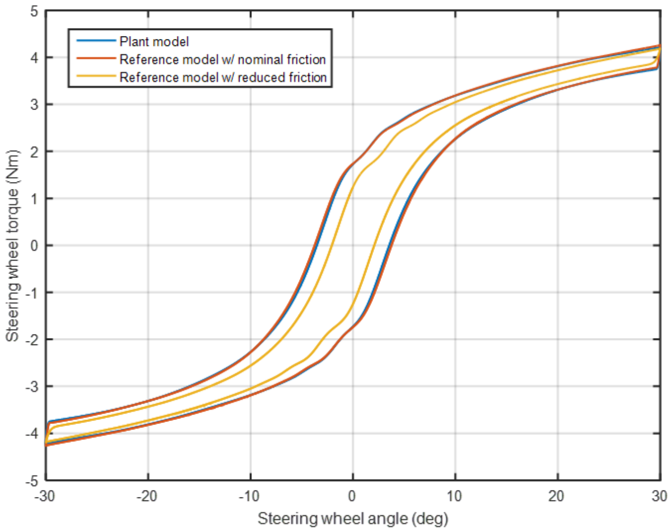


Figure 2.12: Comparison of steering feel hysteresis with reference and plant EPAS model

real-time driving condition changes, e.g., road friction variation. Therefore, the driver will perceive a misleading steering feel since the actual rack force is masked by the rack force model.

In the proposed control system, the actual rack force is estimated from the rack force estimator unlike other emulation methods. In order to provide a realistic steering feel to drivers, it is necessary that the estimated rack force should be closely matched with the actual rack force. The rack force estimator will be discussed in Section 3.

As discussed earlier, the friction level in the steering system significantly affects the steering feel. Since the target steering feel is based on the reference model, a reduced friction level is used to the reference model compared to the plant model, i.e., $T_{f,ref}$ is chosen to be smaller than the overall friction of the plant, such that the target steering feel will have a small hysteresis feel. Figure 2.12 compares the steering feel hysteresis between the reference model and the plant model. As seen clearly from Fig. 2.12, it is possible to set different target steering feels with different friction levels in the steering system. With nominal friction in the reference model, the target steering feel is similar to that of the plant model. The steering feel of the reference model with reduced friction has a narrower hysteresis compared to the plant model, producing a lighter steering feel to the driver [23].

2.3.2 Tracking Controller

The role of the tracking controller is to regulate the C-EPAS to track the reference by applying overlay torque. It is important for a tracking controller to have good robust performance under disturbance, road friction variation, steering system friction variation, etc.

The reduced-order C-EPAS model for controller design is written as

$$T_t + T_m i_{wg} + F_R/i_p - f_e = I_e \ddot{\delta}_c. \tag{2.24}$$

The reduced-order model lumps the inertia of the worm gear, worm and motor shaft, intermediate shaft, pinion shaft, and rack into a single equivalent one, and the frictions in several locations in the high-order model are also represented by an equivalent frictional torque.

The equivalent inertia and friction in Eq. (2.24) are given as

$$I_e = I_s + (I_w + I_m)i_{wg}^2 + I_{uj} + I_{lj} + I_p + m_r/i_p^2, \tag{2.25}$$

$$f_e = f_{gb} + (f_{wb} + \mu_w N_{wp})i_{wg} + \mu_g N_{wp} + f_{pb} + \mu_r N_{pp}, \tag{2.26}$$

where $\mu_w = \frac{\mu_{wg} r_w}{\sin\beta_w}$, $\mu_g = \frac{\mu_{wg} r_w \tan\gamma_w}{\sin\beta_w}$, $\mu_p = \frac{\mu_{rp} \sin\gamma_p r_p}{\sin\beta_{rp}\cos\gamma_r}$, $\mu_r = \mu_R - \frac{\mu_{rp}\tan\gamma_r}{\sin\beta_{rp}}$ are, respectively, the equivalent friction coefficients of worm, worm gear, pinion and rack on the column, due to the preload. μ_{wg} and μ_{rp} are the mesh friction coefficient of worm gear and pinion gear, respectively. N_{wp} and N_{pp} are the preload of worm gear and pinion gear, respectively.

The sliding mode control (SMC) technique is widely used to control systems with large uncertainty. For the tracking controller, a sliding mode controller is developed for this study. The sliding variables used for our controller are given as

$$s = \lambda_1 e + \lambda_2 \dot{e}, \tag{2.27}$$

$$e = \delta_c - \delta_{ref}, \tag{2.28}$$

where λ_1 and λ_2 are positive gains of tracking error and error rate, respectively, and δ_{ref} is reference pinion angle.

The equivalent control is obtained by letting $\dot{s}(t) = 0$, which is the necessary condition for the tracking error to remain on the sliding surface. Subtracting (2.23) from (2.24) gives

$$I_{ref}(\ddot{\delta}_c - \ddot{\delta}_{ref}) + I_{\Delta}\ddot{\delta}_c = T_o - (T_R - \widehat{T}_R) - (f_e - f_{ref}), \tag{2.29}$$

where I_{Δ} is the inertia difference between reference model and plant, T_a is the overlay torque, and reference rack force is the estimated rack force $\widehat{T}_R$ from the Kalman filter.

Taking the derivative of (2.27) and substituting (2.29) becomes

$$\dot{s} = \lambda_1 \dot{e} + \lambda_2 \left[\frac{1}{I_{ref}} (T_o - \Delta T_R - \Delta f - I_{\Delta}\ddot{\delta}_c) \right], \tag{2.30}$$

where $\Delta T_R = T_R - \widehat{T}_R$, and $\Delta f = f_e - f_{ref}$.

The equivalent control is derived by equaling (2.30) to 0,

$$T_{o,eq} = \Delta T_R + \Delta f + I_{\Delta}\ddot{\delta}_c - \frac{\lambda_1 I_{ref}}{\lambda_2}\dot{e}. \tag{2.31}$$

In Eq. (2.31), ΔT_R and $I_{\Delta}\ddot{\delta}_c$ are unknown, and Δf can only be approximated based on the reference and the nominal plant model, while its actual value will vary over time due to wear. Therefore, the sliding mode cannot be obtained by equivalent control alone. The applicable equivalent control is

$$T_{o,eq} = \Delta\widehat{f} - \frac{\lambda_1 I_{ref}}{\lambda_2}\dot{e}, \tag{2.32}$$

where $\Delta\widehat{f}$ is the nominal friction difference.

In order to compensate for the deviation of the representative point due to parameter variations and disturbances, a switching controller must be designed to drive the output trajectory to move toward and reach the sliding surface. For this purpose, the Lyapunov function can be chosen as

$$V(t) = \frac{1}{2}s^2(t), \tag{2.33}$$

with $V(0) = 0$ and$V(t) > 0$ for $s(t) \neq 0$. A sufficient condition to guarantee that the trajectory of the error will approach the sliding phase is to select the control strategy, also known as the reaching condition:

$$\dot{V}(t) = s\dot{s} < -\eta|s|. \tag{2.34}$$

To satisfy the reaching condition, the equivalent control $T_{o,eq}$ given in (2.32) is augmented by a switching control term, $T_{o,N}$. The sliding mode controller is designed as

$$T_o = T_{o,eq} + T_{o,N}. \tag{2.35}$$

If Eqns. (2.30), (2.32), and (2.35) are inserted into the reaching condition (2.34), it is obtained that

$$\frac{\lambda_2}{I_{ref}} s \left[T_{o,N} - \Delta T_R - (\Delta f - \Delta\widehat{f}) - I_{\Delta}\ddot{\delta}_c\right] < -\eta|s|. \tag{2.36}$$

The switching control $T_{o,N}$ in (2.36) is chosen as

$$T_{o,N} = -Ksgn(s), \tag{2.37}$$

where K is the a positive constant and means the upper bound of uncertainty, and $sgn(\cdot)$ denotes signum function, defined as

$$sgn(s) = \begin{cases} 1 & if\ s > 0 \\ 0 & if\ s = 0 \\ -1 & if\ s < 0 \end{cases}. \tag{2.38}$$

Substituting (2.37) back to (2.36), one has

$$K > |\Delta T_R| + |\Delta f - \Delta\widehat{f}| + |I_{\Delta}\ddot{\delta}_c| + \frac{I_{ref}}{\lambda_2}\eta. \tag{2.39}$$

This implies that $\dot{V}(t)$ is a negative definite function as long as K is greater than the sum of the rack force estimation error, the friction torque variation, and the inertia force difference. Therefore, the sliding surface s is attractive for this system. The convergence rate can be adjusted by η.

To alleviate the chattering issue of the sliding mode control, a saturation function has been used via introduction of a thin boundary layer around the sliding surface. The switching control $T_{o,N}$ is modified as

$$\widehat{K}_{o,N} = -Ksat(\frac{s}{\Phi}) = \begin{cases} -Ksgn(s) & if\ |s| \geq \Phi \\ -K\frac{s}{\Phi} & otherwise \end{cases}, \tag{2.40}$$

where Φ is a positive constant that defines the thickness of the boundary layer affecting the steady-state accuracy and robustness.

2.4 Rack Force and Load-Dependent Friction Estimation

The rack force estimator is needed to provide real-time rack force such that the target steering feel reflects the real driving conditions. It is shown in [26] that the friction of the steering system affects the performance of the rack force estimate significantly. A friction estimation algorithm was proposed based on the relationship between steering torque and angle [27]. The main concept of the friction estimation algorithm is summarized here to make the paper self-contained.

2.4.1 Disturbance Observer based Rack Force Estimator

Based on the low-order model, the rack force can be estimated by a disturbance observer. The state space presentation of the C-EPAS system is given in the following:

$$\begin{bmatrix} \dot{\delta}_c \\ \ddot{\delta}_c \end{bmatrix} = A \begin{bmatrix} \delta_c \\ \dot{\delta}_c \end{bmatrix} + B_u \begin{bmatrix} T_t \\ T_m \end{bmatrix} + B_w \begin{bmatrix} f_e \\ F_R \end{bmatrix}, \tag{2.41}$$

$$\begin{bmatrix} \delta_c \\ \dot{\delta}_c \end{bmatrix} = C \begin{bmatrix} \delta_c \\ \dot{\delta}_c \end{bmatrix}, \tag{2.42}$$

where $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, $B_u = \begin{bmatrix} 0 & 0 \\ 1/I_e & i_{wg}/I_e \end{bmatrix}$, $B_w = \begin{bmatrix} 0 & 0 \\ -1/I_e & 1/i_p I_e \end{bmatrix}$, and $C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.

There are two unknown inputs in Eq. (2.41), f_e and F_R. Augmenting the state vector with both f_e and F_R by assuming that they are slowly changing, i.e., $\dot{f}_e = \dot{F}_R = 0$, we have the extended state space equations

$$\begin{bmatrix} \dot{\delta}_c \\ \ddot{\delta}_c \\ \dot{f}_e \\ \dot{F}_R \end{bmatrix} = A_E \begin{bmatrix} \delta_c \\ \dot{\delta}_c \\ f_e \\ F_R \end{bmatrix} + B_R \begin{bmatrix} T_t \\ T_m \end{bmatrix}, \tag{2.43}$$

$$\begin{bmatrix} \delta_c \\ \dot{\delta}_c \end{bmatrix} = C_E \begin{bmatrix} \delta_c \\ \dot{\delta}_c \\ f_e \\ F_R \end{bmatrix}, \tag{2.44}$$

where $A_E = \begin{bmatrix} A & B_w \\ 0 & 0 \end{bmatrix}$, $B_E = \begin{bmatrix} B_u \\ 0 \end{bmatrix}$, and $C_E = \begin{bmatrix} C & 0 \end{bmatrix}$.

The observability of system (2.43) and (2.44) is checked by the rank of matrix $Ob = \begin{bmatrix} C_E & C_E A_E & C_E A_E^2 & C_E A_E^3 \end{bmatrix}^T$. It can be shown that the rank of $Ob = 3$, which is smaller than the system order, indicating that the extended system is unobservable. It is thus impossible to estimate all the system states no matter what observer is used. The unobservability of the system can also be interpreted in the physical aspect: since the steering column is modeled as a single rigid shaft, the rack force F_R and steering friction f_e have the same effect on the dynamics of the column. Without including additional sensors or dynamics, they are indistinguishable from each other.

In order to estimate F_R based on steering system dynamics, f_e must be known a priori or determined by another mechanism. For the moment, we just assume that f_e is a known input to the system. The state space presentation of the C-EPAS system is discretized as

$$x_{k+1} = F x_k + G u_k, \tag{2.45}$$

$$y_k = H x_k, \tag{2.46}$$

where $x = \begin{bmatrix} \delta_c & \dot{\delta}_c & F_R \end{bmatrix}^T$, $u = \begin{bmatrix} T_t & T_m & f_e \end{bmatrix}^T$, and $y = \begin{bmatrix} \delta_c & \dot{\delta}_c \end{bmatrix}^T$. The matrices F, G, and H are derived using the zero-order-hold method.

The system represented by Eqns. (2.45) and (2.46) is fully observable so that the rack force F_R can be estimated using any linear observer [28], e.g., a Luenberger observer. In this work, we adopt a Kalman filter to optimally estimate the rack force while attenuating the noise effect. The Kalman filter is given by

$$P_k^- = F P_{k-1}^+ F^T + Q, \tag{2.47}$$

$$K_k = P_k^- H^T (H P_k^- H^T + R)^T, \tag{2.48}$$

$$\boldsymbol{x}_k^- = F \boldsymbol{x}_{k-1}^+ + G u_{k-1}, \tag{2.49}$$

$$\boldsymbol{x}_k^+ = \boldsymbol{x}_k^- + K_k (y_k - H \boldsymbol{x}_k^-), \tag{2.50}$$

$$P_k^+ = (I - K_k H) P_k^- (I - K_k H)^T + K_k R K_k^T, \tag{2.51}$$

where Q and R are the covariance matrices of system noise and measurement noise, respectively.

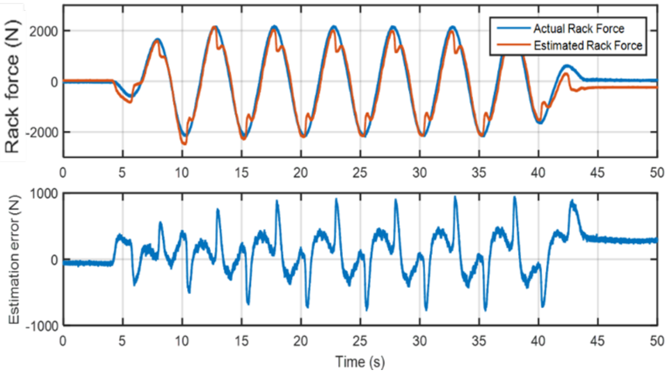


Figure 2.13: Rack force estimation without friction estimation

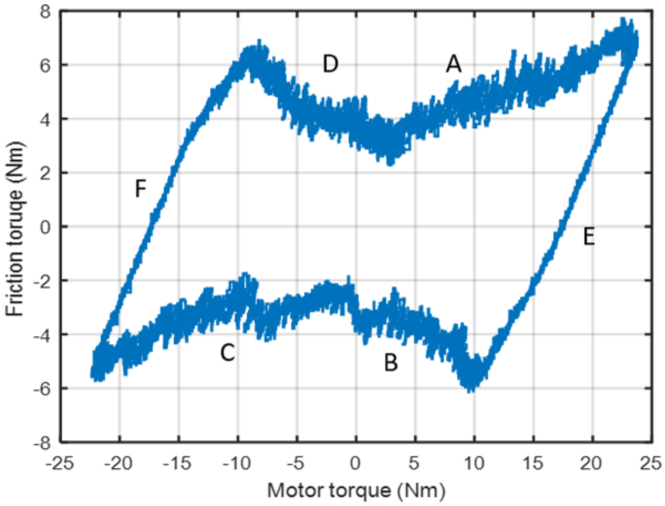


Figure 2.14: C-EPAS system friction vs. assist motor torque

The effect of steering system friction on steering feel can be seen from Fig. 2.13. It shows the rack force estimation result when the steering system friction in the estimator is approximated by Coulomb friction while the real friction in the plant is not. The estimation shows great error near the peak of the rack force due to the discrepancy in the frictions.

2.4.2 Load-Dependent Friction

Figure 2.14 shows the overall friction of an C-EPAS system with respect to assist-motor torque based on low-frequency sinusoidal steering test data from an C-EPAS test bench [29] As shown in Fig. 2.14, the friction torque is quite different from the commonly used Coulomb friction, as it varies almost linearly with motor torque during steering (A and C) and return maneuvers (B and D).

The model of load-dependent friction f_e^i is expressed as

$$f_e^i = \begin{cases} f_0 & |T_m| < T_{0,i} \\ f_0 + \lambda_{f,i}(|T_m| - T_{0,i}) & otherwise \end{cases}. \tag{2.52}$$

where $i = s, r$ represents the steering and return maneuvers, respectively. f_0 is the load-independent friction resulting from preload, $\lambda_{f,i}$ is the load dependency coefficient, and $T_{0,i}$ is the motor torque where load dependency starts to appear.

The algorithm is based on the relationship between the steering angle and torque that are measured by torsion bar sensors. In the case of low-frequency steering input, e.g. 0.2Hz sinusoidal input, the inertia force in Eq. (2.24) is negligible. Thus, the torque balance equation of EPAS can be expressed as

$$T_t + T_m i_{wg} = T_{app} = -F_R/i_p + f_e, \tag{2.53}$$

where T_{app} denotes the total applied torque from the measured driver and the assist motor.

When the vehicle is operated in the linear range at a constant forward velocity with low-frequency input, the rack forces of steering and return at the same steering angle are similar, i.e., $F_R^s(\delta_c^r) \sim F_R^r(\delta_c^r)$ when $\delta_c^s = \delta_c^r$. Therefore, by subtracting the steering and return states, we have

$$|T_{app}^s(\delta_c^s) - T_{app}^r(\delta_c^r)| = |f_e^s| + |f_e^r|. \tag{2.54}$$

Equation (2.54) indicates that the difference between the total applied torque of the steering and return states at the same steering angle is equal to the sum of frictions of both states at that angle. Figure 2.15 depicts this relationship, where the input is the low-frequency sinusoidal steering angle. Figure 2.15 shows the hysteresis loop of the total applied torque versus the steering angle from simulation. According to the relationship in Eq. (2.54), the width of hysteresis indicates the sum of friction of the steering and return states. The torque-versus-angle relationship in Fig. 2.15 can be divided into two sections in each steering and return state. The hysteresis in the small steering angle remains constant, which corresponds to the load-independent portion of the friction. As the steering angle keeps increasing, the load-dependent effect appears after the change point δ_b. As a result, the hysteresis becomes greater. In addition, the relationship between T_{app} and δ_c can be approximated as the two affine functions A_1 and A_2 in each section, respectively.

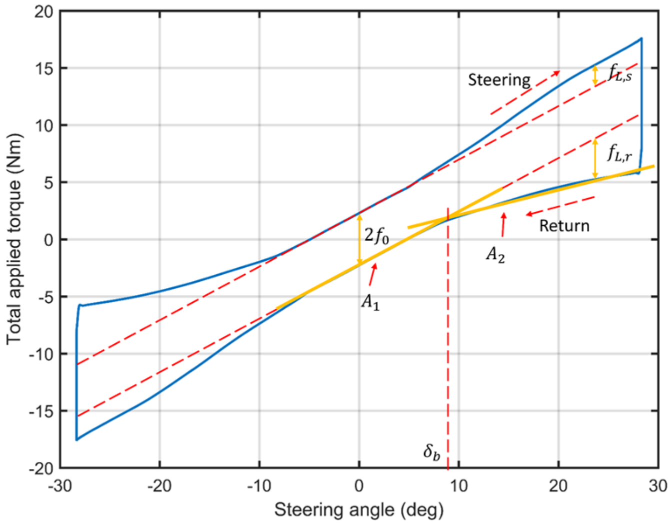


Figure 2.15: Relationship between applied torque and angle

The relationship between the hysteresis width and friction torque is given as

$$T^s_{app}(\delta^s_c) - T^r_{app}(\delta^r_c) = 2f_0 + f_{L,s} + f_{L,r}, \tag{2.55}$$

where $f_{L,s}$ and $f_{L,r}$ are the load-dependent portion of the friction in the steering and return states, respectively. To estimate f_0, $f_{L,s}$, and $f_{L,r}$, the measurements of T_t, T_m, δ_c, and T_{app} are filtered and stored (denoted by superscript *). Then, the affine functions A1 and A2 are identified using least square regressions by changing the break point δ^*_b from $\delta^*_{c,min}$ to $\delta^*_{c,max}$ iteratively. After all iterations are completed, the optimal parameters (λ_f, f_0, and δ^*_b) with the smallest regression error are chosen.

Next, the load-dependent portion of friction $f_{L,s}$ and $f_{L,r}$ can be determined by the relationship between T^*_{app} and δ^*_c using optimal parameters identified in the previous step:

$$f^*_{L,s} = |T^*_{app} - (\rho_f \delta^*_c + f_0 sign(\dot{\delta}^*_c))| \text{ if } \delta^*_c \dot{\delta}^*_c > 0, \tag{2.56}$$

$$f^*_{L,r} = |T^*_{app} - (\rho_f \delta^*_c + f_0 sign(\dot{\delta}^*_c))| \text{ if } \delta^*_c \dot{\delta}^*_c < 0, \tag{2.57}$$

where ρ_f and f_0 are the identified slope and offset of A_1.

Finally, the coefficients $\lambda_{f,i}$ and $T_{0,i}$ can be obtained by the least squares method, using the equation

$$f^*_{L,i} = \lambda_{f,i}(|T^*_m| - T_{0,i}), \ i = s, r. \tag{2.58}$$

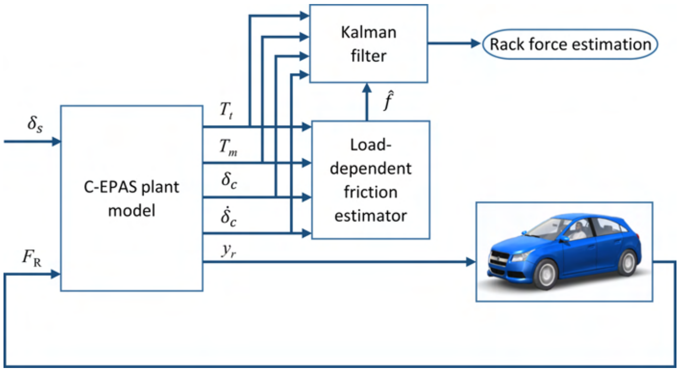


Figure 2.16: Schematic of rack force estimation with friction estimation

For the proposed estimation algorithm to have good performance, the vehicle should be running in a low-frequency on-center steering maneuver at constant speed, such that the rack forces in the steering and return states can be canceled. Figure 2.16 shows the schematic of rack force estimation with friction estimation.

2.5 Simulation and Experimental Results

The proposed rack force estimation and steering feel control algorithm are evaluated using simulations and experimental testings.

2.5.1 Rack Force Estimation Simulation

In this case, the simulated vehicle is running on a high μ ($\mu = 1$) road surface at 75km/h while the steering excitation of a 0.2Hz sinusoid with amplitude of 30° is applied. The response of total applied torque vs. steering angle during 10s simulations is plotted in Fig. 2.17. As discussed in the previous section, the load-dependency characteristic of friction is reflected in the hysteresis width of the curve. Based on the raw measurements, the data points indicated by circular marks are obtained, which will be used for friction estimation in the following.

Figure 2.17 also shows the lines of two linear regressions, which represent the load-independent and load-dependent portions, respectively. The slopes change of the two portions is clearly seen. The change point of the load-dependent and independent parts is found by the iteration regression algorithm. Figure 2.18 shows the regression errors of the iteration procedure, where the lowest point corresponds to the change point.

The load-dependent friction can then be estimated by the difference between the regression line and the measurement data in Fig. 2.17 . The load-dependent friction is plotted with

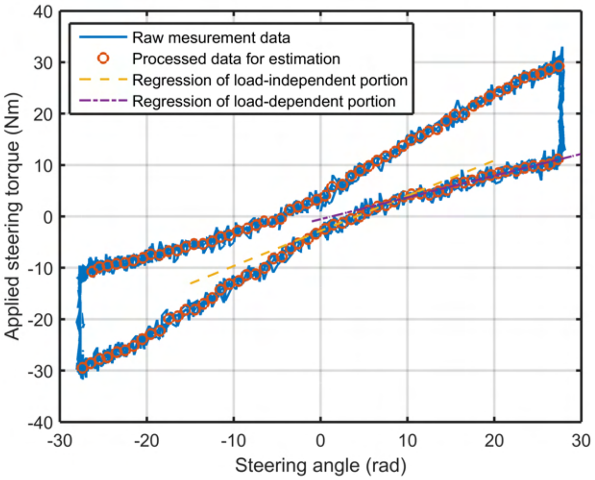


Figure 2.17: Total applied torque vs. steering angle

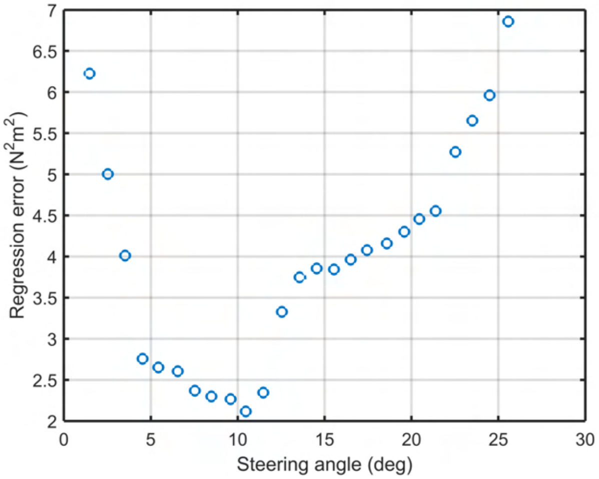


Figure 2.18: Iterative regression results

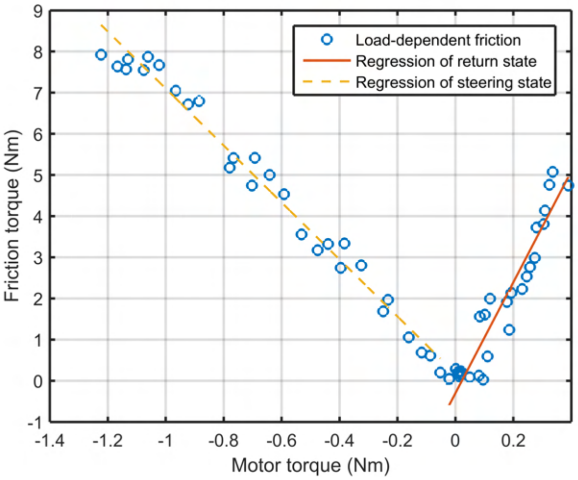


Figure 2.19: Load-dependent friction vs. motor torque

respect to the motor torque in Fig. 2.19. It can be seen clearly in Fig. 2.19 that the relationship between friction and motor torque is almost linear. The different dependency on motor load in the return and steering states is also captured. By linear regression, the load-dependent friction model, which is later used for the rack force estimation, is identified by two straight lines.

After the load-dependent friction is identified, the estimated friction parameters are applied to the friction model used for rack force estimation. Root mean square error (RMSE):

$$RMSE = \sqrt{\frac{\sum_1^n \left(F_R - \widehat{F}_R\right)^2}{n}}, \tag{2.59}$$

and normalized root mean square error (NRMSE) over the range of the measurement data:

$$NRMSE = \frac{RMSE}{F_{R,max} - F_{R,min}} \tag{2.60}$$

are introduced as quantitative error analysis.

Figure 2.20 and 2.21 show the friction and rack force estimation results, respectively. As can be seen in Fig. 2.20, the estimated friction successfully captures the load-dependency characteristic existing in the plant model. As a result, the rack force estimate with load-dependent friction (RMSE=84N and NRMSE=2.0%) shown in Fig. 2.21 is much improved compared to that with load-independent friction only (RMSE=511N and NRMSE=11.9%).

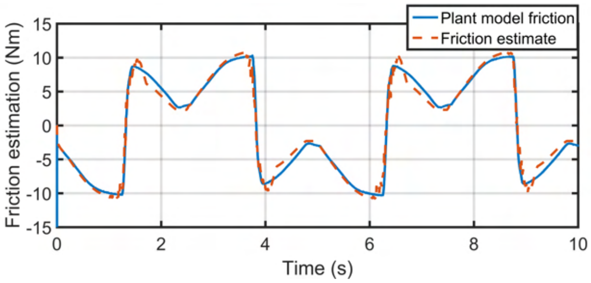


Figure 2.20: Friction estimation

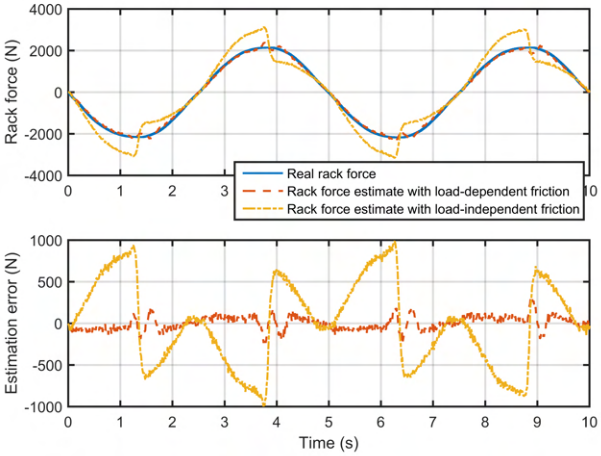
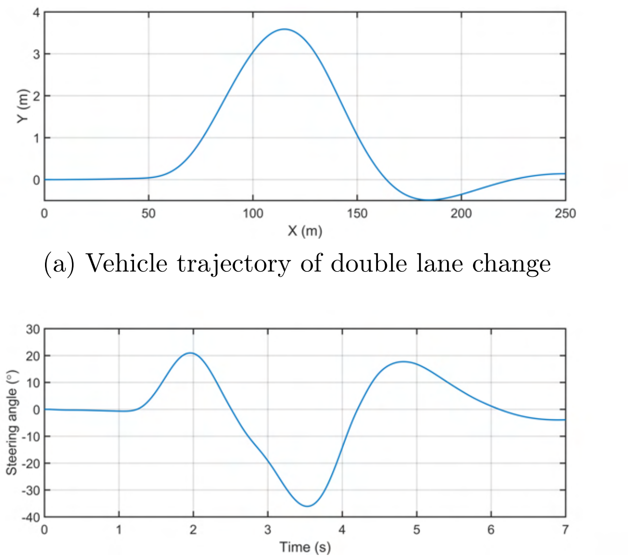


Figure 2.21: Rack force estimation



(a) Vehicle trajectory of double lane change

(b) Steering angle input of double lane change

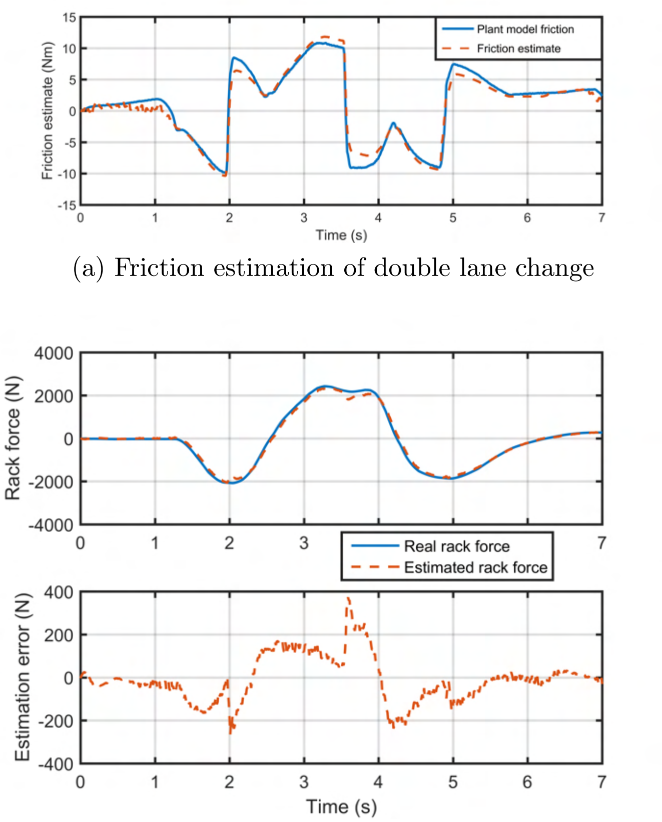
Figure 2.22: Vehicle trajectory and steering angle of DLC maneuver

The steering inputs are standard sinusoidal waves in the previous cases. In this section, the vehicle is controlled by the Carsim driver model, such that the rack force estimation is evaluated in more realistic scenarios. The vehicle performs a double lane change (DLC) maneuver at 120km/h with μ=0.85. The load-dependent friction is assumed to be identified using sinusoidal input. Figure 2.22 present the vehicle trajectory and steering wheel angle during the simulation.

The friction and rack force estimation results are shown in Fig. 2.23, respectively. As is shown, a good friction estimate is achieved, although estimation is slightly worse in the return state compared to the steering state. A good rack force estimation is also obtained with small RMSE=104N and NRMSE=2.6% in the presence of the saturation of rack force.

2.5.2 Steering Feel Control Simulation

The proposed steering feel control algorithm is evaluated through simulation in Carsim and Matlab/Simulink environments. At first, the steering feel under low-frequency input is simulated. For low-frequency steering feel simulation, a 0.2Hz, 30° sinusoidal steering wheel angle input is applied while vehicle runs at 75km/h. The steering feel control results are shown in Fig. 2.24. We can see that the steering feel without torque overlay control has a larger hysteresis compared to the reference. On the other hand, with activation of torque overlay control, the steering feel hysteresis matches the reference one.



(a) Friction estimation of double lane change

(b) Rack force estimation of double lane change

Figure 2.23: Friction and rack force estimation of DLC maneuver

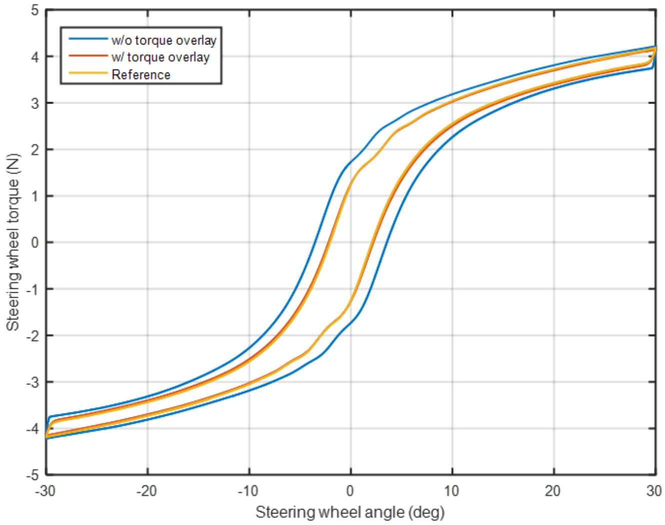


Figure 2.24: Comparison of steering feel hysteresis with and without torque overlay control

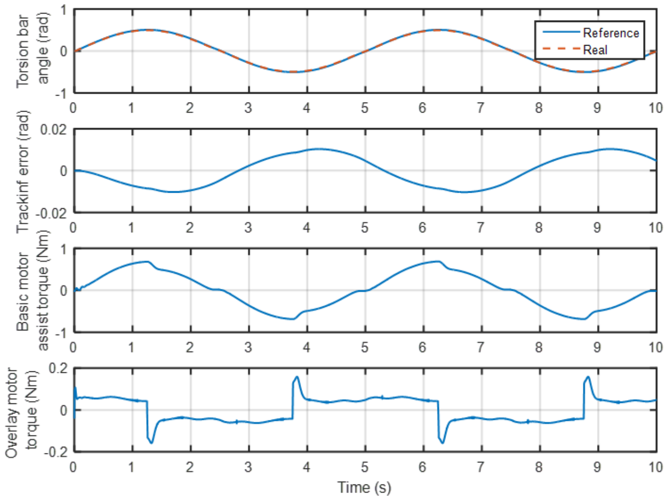


Figure 2.25: Comparison of tracking controller performance

Figure 2.25 compares tracking performance of the tracking controller. It can be observed that it tracks smoothly with the reference signal without any chattering and its tracking error is less than 2%. The overlay torque used is also smooth, with the maximum value below 0.2Nm. These simulation results show the effectiveness of the torque overlay control algorithm on steering feel control.

The sine-sweep steering feel frequency response is shown in Fig. 2.26. Sine-sweep steering wheel angle inputs (0.2 to 3Hz) with 15° amplitude are applied at a 120km/h vehicle speed. As shown in Fig. 2.26, the steering feel with torque overlay control is well matched to the reference one.

2.5.3 Hardware-in-the-Loop Test

In order to validate the proposed algorithm, Hardware-in-the-loop (HiL) simulations were performed. Real measurement data from the torsion bar and motor sensors have been gathered on an EPAS HiL simulator. Moreover, a rack force sensor is installed on the HiL system in order to compare the estimated value and the real torque applied on the rack. The hardware configuration of the HiL system is shown in Fig. 2.27. The data are recorded under a 0.2Hz sinusoidal steering angle input with 30° amplitude. The vehicle speed is 75km/h.

Figure 2.28 shows the rack force estimation results. It is shown in Fig. 2.28 that the rack force estimate is good. The amplitude of the estimate error is less than 200N. Figure 2.29 shows the responses of torque overlay control system from the HIL test with a 0.2Hz sinusoidal steering input. The HIL system consists of C-EPAS hardware, dSpace simulator, rack load

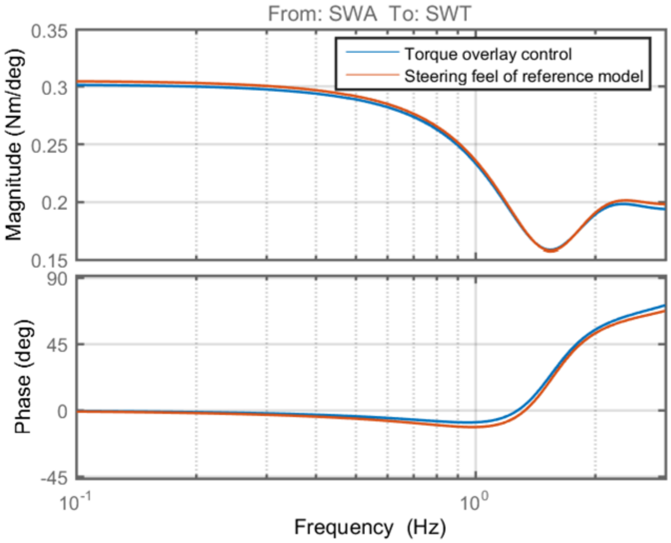


Figure 2.26: Sine-Sweep steering feel frequency response

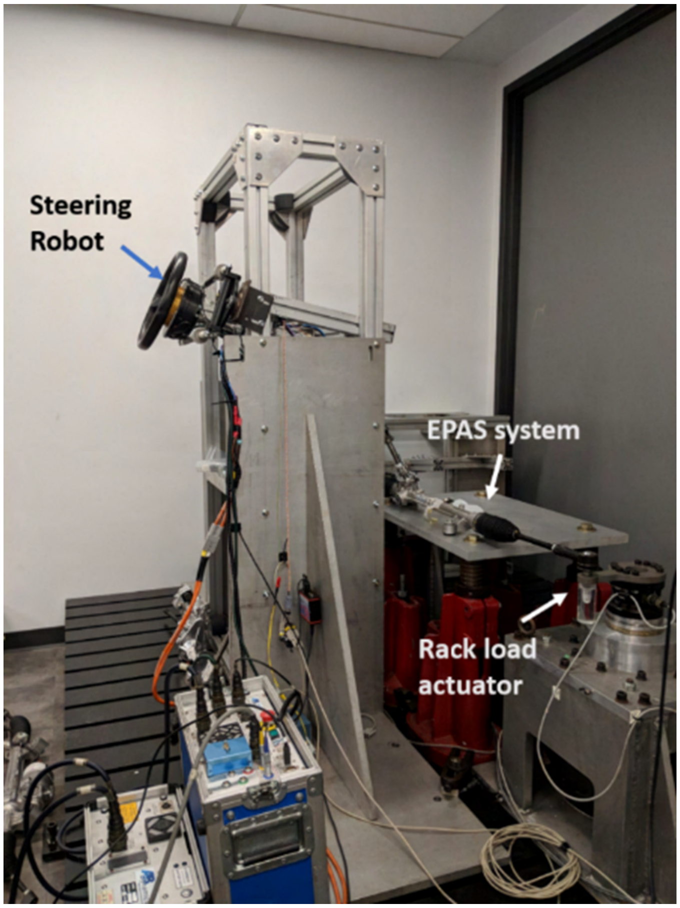


Figure 2.27: EPAS hardware-in-the-loop simulator

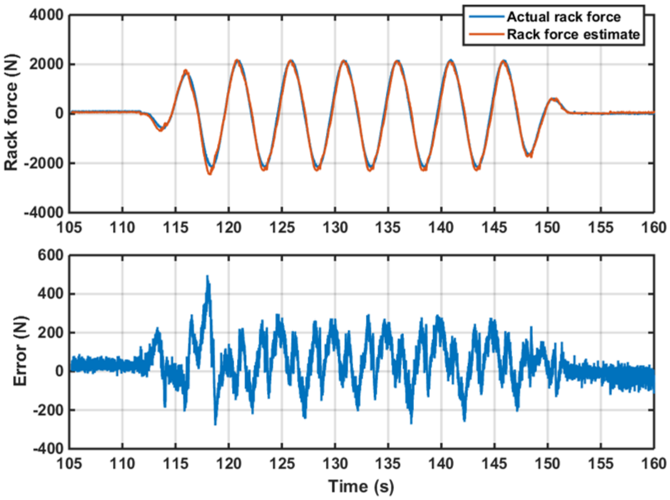


Figure 2.28: Rack force estimation of HIL test

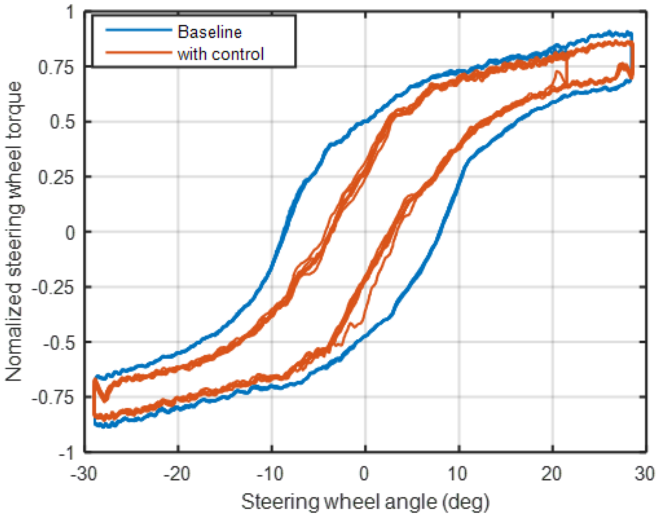


Figure 2.29: HIL test results of torque overlay control

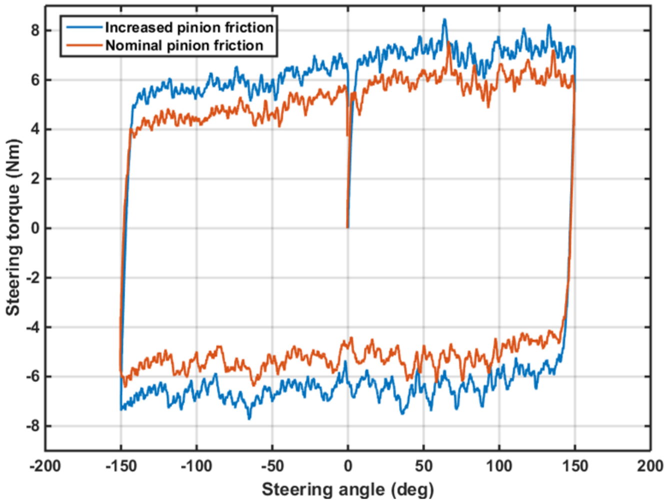


Figure 2.30: Torque-to-rotate test results

actuators, and steering robot. As can be seen in Fig. 2.29, the steering feel hysteresis response is improved (a smaller steering feel hysteresis) with the proposed torque overlay control.

To evaluate the robustness of the control system's performance, the actual friction on the pinion shaft is adjusted to change the steering system friction level. Figure 2.30 shows the steering torque vs. steering angle from the torque-to-rotate test. In Fig. 2.30, the one with the higher torque-to-rotate corresponds to the system with increased pinion friction. It can be seen that the friction of the steering system is increased by about 1Nm compared to the nominal system.

Figure 2.31 shows the rack force estimation results in the case of increased pinion friction under low-frequency sinusoidal input. We can see from Fig. 2.31 that the estimation of rack force maintains a satisfactory performance. The error of the estimation is generally below 300N. The steering feel of the system with increased friction is compared with the nominal system in Fig. 2.32. We first compare the steering feels of increased and nominal friction without torque overlay control. The steering feel of the system with increased friction has a bigger hysteresis than that of the nominal system. However, the steering feel with torque overlay control is almost the same in both cases, which means that the torque overlay control system performs consistently well in the presence of friction variation.

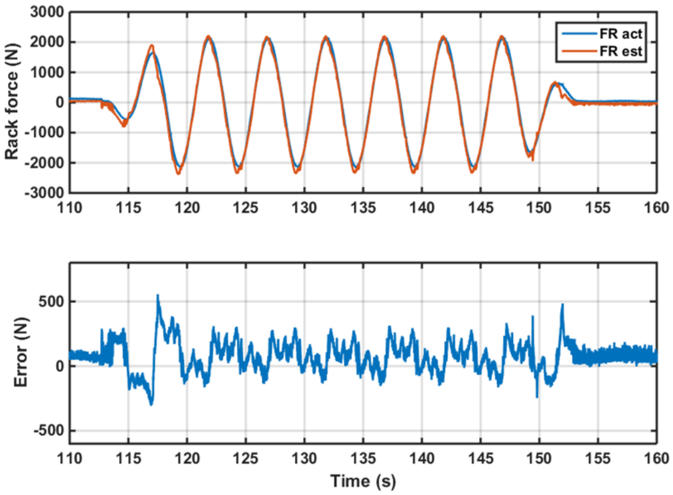


Figure 2.31: Rack force estimation with increased pinion friction

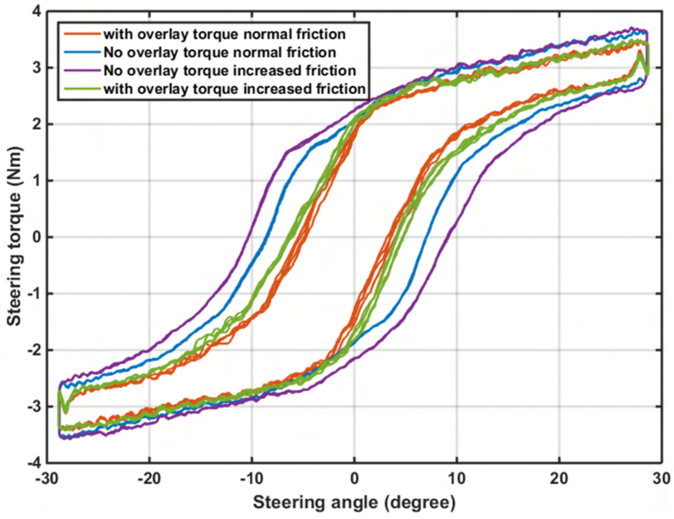


Figure 2.32: Steering feel of steering system with increased and nominal friction

2.6 Conclusions

In this chapter, we proposes a steering feel improvement methodology utilizing overlay torque and feedback control. The simulation and test results demonstrate that (1) the reference model can provide a more realistic target steering feel due to the inclusion of real-time rack force, (2) rack force can be successfully estimated by the Kalman filter integrated with friction estimation, and (3) steering feel can be improved by tracking the target steering feel despite variations in vehicle parameters and operation conditions.

Chapter 3

Clamping Force Control for Electromechanical Brake System

Along with the steering system, the brake system is another important chassis component that significantly affects vehicle's performance and safety. In fact, many of the widely implemented active safety systems, for example ABS and ESC, are built on the basis of brake system. In particular, with growing interest of vehicle electrification and autonomous vehicle systems, electromechanical brake (EMB) system becomes important and has widely gained attention.

In this chapter, we design a clamping force control system for EMB. EMB generates the clamping force between the brake pad and disk through an electric motor and mechanical transmission. Control of the clamping force has significant impact on the brake performance. The EMB control shares many similarities to the EPAS control in previous chapter, for example, mechanical friction being a major disturbance and estimation needed as feedback to the controller. But it also has its uniqueness in terms of system modeling, control algorithm and architecture. We propose a unified clamping force control framework that incooperates estimation, contact detection, gap closing and tracking control.

3.1 Introduction

Inspired by the fly-by-wire system in the aerospace industry, drive-by-wire technology has been getting increasing attention from the automotive industry in recent years. Regarding the brake-by-wire (BBW) technology, the first implementation is the electro-hydraulic brake (EHB) system. It retains the hydraulic linkages of the conventional brake system while it realizes by-wire functionality. The electro-mechanical brake (EMB) system is introduced as the second generation of BBW, which further improves performance. Compared to the EHB, it replaces the hydraulic components with an electric motor and mechanical transmissions that connect the actuator and brake pad.

The EMB system offers various advantages, such as the reduction of volume and weight and ease of maintenance. More importantly, the fast response of the electric motor and the computational power of the electronic control unit (ECU) facilitate the implementation of advanced driver assistance systems (ADAS) such as automated emergency braking (AEB) and electronic stability control (ESC) systems.

One major research topic of the EMB system is the clamping force control. The brake torque is related to the clamping force through the friction between the brake pad and disk. Thus, the quality of the clamping force control directly affects the brake performance. The challenges of EMB clamping force control include the friction of the bearing and gear transmission, as well as the nonlinear stiffness and hysteresis of the caliper and brake pad. Furthermore, the difficulty of measuring real-time clamping force as feedback also must be taken into consideration during controller design and implementation.

Many control architectures have been proposed for clamping force control. A popular control method is the cascade PID control [30–32], which consists of three feedback controllers positioned sequentially from the inner to the outer loop for regulating the current, speed, and force, respectively. This control structure is conceptually simple and intuitive. However, the effort for tuning the controllers is extensive, since three loops are coupled together and must be tuned iteratively to achieve the best overall performance. To cope with the nonlinear friction in the EMB system, a feedforward compensation of Coulomb friction is added to the cascade control system [33]. It is shown that the tracking error is reduced, and a finer force modulation is achieved. Similarly, an inverse model controller is developed in [34], where various types of friction and nonlinearities of the individual components are modeled and compensated for using an inverse model. Although the feedforward control can linearize the EMB plant to improve the effectiveness of the feedback controller, its performance is very sensitive to modeling error. Even slight model uncertainties will yield a significant performance degradation or even instability.

To improve the robustness of the control system, more advanced control techniques have been introduced. An H_∞ controller for clamping force tracking is designed in [35], where the parametric uncertainty and unmodeled dynamics of EMB are considered as structured uncertainty. The simulation shows that the H_∞ controller is less sensitive to the actuator perturbations yet too conservative compared to the conventional PI controller, resulting in a sluggish response. In order to obtain the ideal tracking performance in the presence of uncertain parameters, an adaptive sliding mode controller is developed in [36], in which the parameter of a lumped friction model is identified online and incorporated into a sliding model controller such that the tracking performance is robust against varying friction. While it presents a promising result, only friction variation is considered, and the adaptation mechanism might fail when there are multiple fast-varying parameters in the plant. Other control algorithms such as model predictive control [37, 38] and time-optimal control [39] are also studied. Although these methods give optimal controllers in theory, they rely heavily on accurate modeling of the plant and are not quite suitable for fast low-level control applications due to the more demanding computational effort.

Aside from the tracking control, there are also practical problems to be considered. First of all, the clamping force is usually not measured due to the difficulty of sensor installation and cost. In [32], a stiffness model that describes the relationship between motor angle and clamping force is proposed for clamping force estimation. A similar approach has also been adopted in [36] because of its simplicity. However, the hysteresis of the clamping force is not considered here, which could result in a large error if the hysteresis is significant in the real system. In [40], the clamping force is divided into linear and nonlinear part, respectively. Then a Kalman filter is designed to estimate the clamping force as well as other states. A data-driven approach for clamping force estimation is proposed in [41] that relies only on the motor current and voltage. However, this method requires a large amount of data from various experiments. It does not consider clamping force hysteresis, either. The clamping force models in [42] and [43] consider both the nonlinearity and the hysteresis, which requires identification of extra parameters using experimental data. Another practical problem is contact detection and gap management. An air gap must be maintained between the brake pad and disk to avoid damage caused by unwanted contact. Furthermore, the air gap distance must be kept constant for the best control performance despite the wear of the brake pad. Several algorithms have been developed for this problem. For example, the gradient of motor torque is used in [43] to detect the contact point. However, the accuracy of contact detection will be affected by measurement noise.

In this chapter, a clamping force control system is developed that consists of the clamping force estimator, gap distance estimator, and tracking controller. The rest of the chapter is organized as follows. The mechanical and clamping force models of the EMB are introduced in Section 3.2. Clamping force estimation, gap distance estimation, and a tracking controller are developed respectively in Section 3.3. In Section 3.4, closed-loop simulations of the proposed clamping force control system are performed, and the control performance is discussed. Finally, conclusions are drawn in Section 3.5.

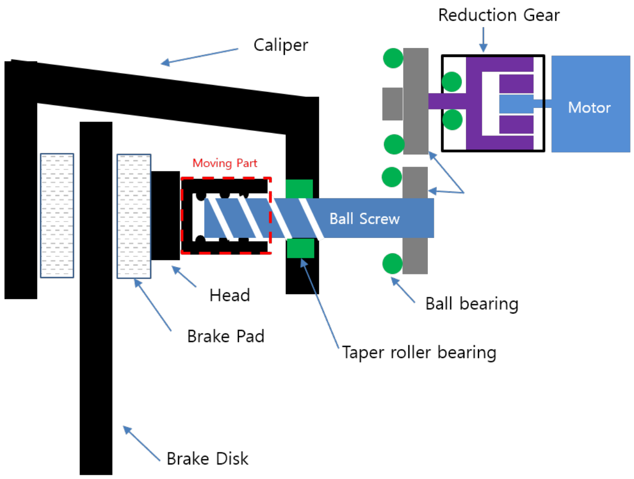


Figure 3.1: EMB schematic

3.2 EMB Modeling

EMB modeling is introduced in this section. Figure 3.1 shows the schematic of the considered EMB system. The rotational motion of the motor is converted into a linear motion of the head through the reduction gear and the screw thread gear. The linear motion of the head generates a clamping force between the pad and the disk, which provides the braking torque.

3.2.1 Mechanical Model

The permanent-magnet DC motor is modeled as an ideal DC model using the following two equations:

$$V_a = R_a I_a + K_{emf}\dot{\theta}_m + L_a \dot{I}_a, \tag{3.1}$$

$$J_m \ddot{\theta}_m = K_{motor} I_a - B_m \dot{\theta}_m - \tau_f - T_L, \tag{3.2}$$

where V_a is the armature voltage, R_a is the armature resistance, L_a is the armature inductance, K_{emf} is the back-emf coefficient, J_m is the motor rotor inertia, K_{motor} is the torque constant, B_m is the rotor damping coefficient, $\dot{\theta}_m$ is the rotor angular speed, I_a is the armature current, τ_f is the friction torque, and T_L is the load torque.

In Eq. (3.2), the load torque T_L is expressed as the torsional torque resulting from the compliance and damping of the motor shaft and gear train

$$T_L = k_m \left(\theta_m - i_g \theta_s\right) + c_m \left(\dot{\theta}_m - i_g \dot{\theta}_s\right), \tag{3.3}$$

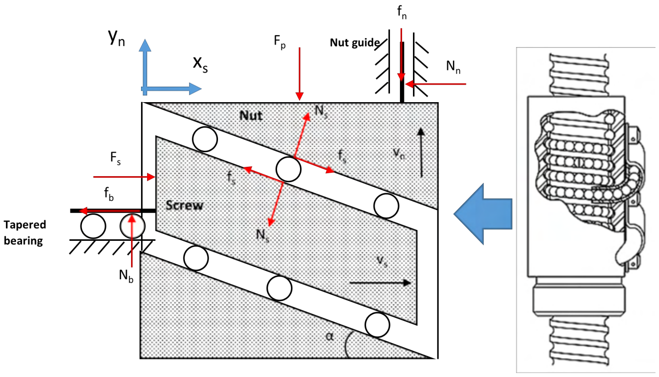


Figure 3.2: Schematic of ball screw

where k_m, and c_m are respectively the stiffness and damping of the motor shaft and gear train, i_g is the reduction gear ratio, and θ_s is the rotational angle of the ball screw.

The reduction gear is modeled by the ideal reduction ratio, including transmission efficiency, which is written as

$$T_s = i_g \eta_g T_L, \tag{3.4}$$

where η_g is the gear efficiency and T_s is the drive torque on the ball screw.

The rotary motion of the ball screw is converted to the linear translation of the ball screw nut. The unwrapped planar model is used for modeling the ball screw, with the assumptions that the screw ball is rigid and the motion of screw and nut are constrained along the x_s and y_n axes, respectively. Figure 3.2 illustrates the schematic of the ball screw model.

The dynamics of the ball screw and nut are expressed as following two equations, respectively

$$m_{eq}\ddot{x}_s = F_s - f_b - N_s \sin\alpha - f_s \cos\alpha, \tag{3.5}$$

$$(m_n + m_p)\,\ddot{y}_n = N_s \cos\alpha - f_s \sin\alpha - F_p - f_n, \tag{3.6}$$

where m_{eq} is the equivalent translational mass of the screw, α is the screw lead angle, m_n is the nut mass, m_p is the lumped mass of the brake pad and head, and x_s and y_n are the translation displacement of the ball screw and the nut along their axis, respectively. The external forces of the ball screw are the driving force from reduction gear F_s, tapered bearing friction f_b, screw and nut contact force N_s, and screw ball rolling friction f_s. The external forces applied on the screw nut are the clamping force F_p and the housing friction f_n.

The torque and force of the ball screw can be converted by the following equations

$$F_s = \frac{T_s}{r_s}, \tag{3.7}$$

$$f_b = \frac{T_{fb}}{r_s}, \tag{3.8}$$

where r_s is the pitch radius of the ball screw and T_{fb} is the bearing friction torque calculated by the LuGre friction model.

The screw ball rolling friction f_s can be calculated as

$$f_s = \mu_s N_s, \tag{3.9}$$

where μ_s is the friction coefficient modeled by the LuGre friction model.

In Eq. (3.5), the equivalent translational mass m_{eq} is the lumped mass, including the planetary gear, the spur gear, and the screw shaft, which are expressed by the following equations

$$J_{eq} = J_{pg} i_g^2 + J_{sg} + J_s, \tag{3.10}$$

$$m_{eq} = \frac{J_{eq}}{r_s^2}, \tag{3.11}$$

where J_{pg} is the planetary gear inertia seen from the motor side, J_{sg} is the spur gear inertia, and J_s is the ball screw inertia.

In the unwrapped model, the rotational angle θ_s of the screw shaft is expressed by the equivalent translational displacement of x_s using the equation below

$$x_s = \theta_s r_s. \tag{3.12}$$

The relationship of the screw shaft and nut displacement is expressed as

$$x_s \tan\alpha = y_n. \tag{3.13}$$

Using Eqns. (3.6)–(3.13), we can combine (3.6) and (3.13) by eliminating the contact force N_s, which gives

$$\ddot{\theta}_s = \frac{T_L - T_{fb} - r_s G F_p}{J_{eq} + r_s^2 G \left(m_n + m_p\right) \tan\alpha}, \tag{3.14}$$

$$\ddot{y}_n = \theta_s r_s \tan\alpha, \tag{3.15}$$

where $G = \frac{\tan\alpha + \mu_s}{1 - \mu_s \tan\alpha}$.

The dynamic LuGre friction model is adopted in our model, which is written as

$$T_f = \sigma_0 z + \sigma_1 \dot{z} + \sigma_2 \dot{\theta}, \tag{3.16}$$

$$\dot{z} = \dot{\theta} - \sigma_0 \frac{\left|\dot{\theta}\right|}{g\left(\dot{\theta}\right)} z, \tag{3.17}$$

$$g\left(\dot{\theta}\right) = T_c + \left(T_{ba} - T_c\right) e^{-\left(\dot{\theta}/\dot{\theta}_s\right)^2}, \tag{3.18}$$

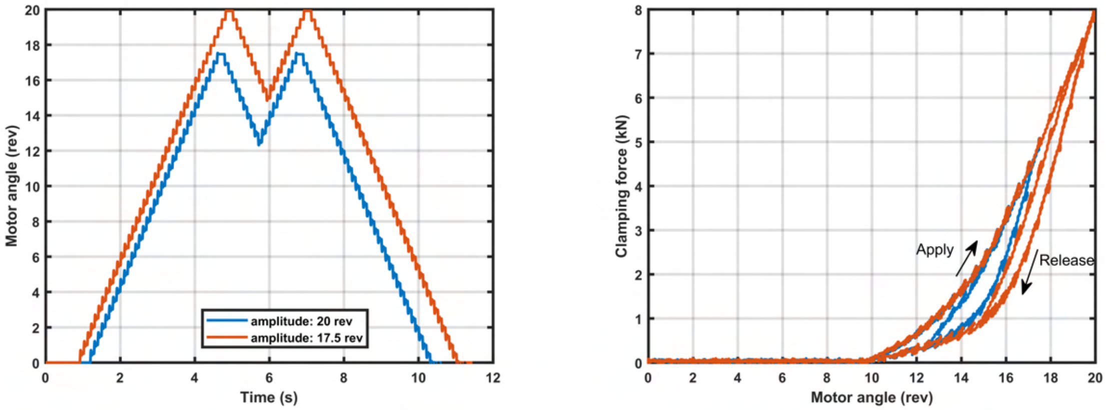
where σ_0 and σ_1 are the bristle stiffness and damping, respectively, σ_2 is the viscous friction coefficient, T_c and T_{ba} are the Coulomb and static friction torque, respectively, $\dot{\theta}_s$ is the Stribeck velocity; and $\dot{\theta}$ is the angular speed.

3.2.2 Clamping Force Model

A correct understanding of the clamping force characteristics is necessary before undertaking the modeling. Experimental testing of the EMB hardware was conducted by our research colleagues at DGIST to obtain the real clamping force. For the detailed experimental setup, one can refer to [44].

Figure 3.3 shows the clamping force experimental test results. We can see that there is a dead zone where clamping force remains zero before it is generated. This dead zone corresponds to the initial air gap between the brake pad and disk, which is handled by the gap management system. The clamping force is generated as soon as the brake pad contacts the disk. Two main aspects of the clamping force are clearly shown, the nonlinearity and the hysteresis, which are determined by the properties of the brake caliper and brake pad [45].

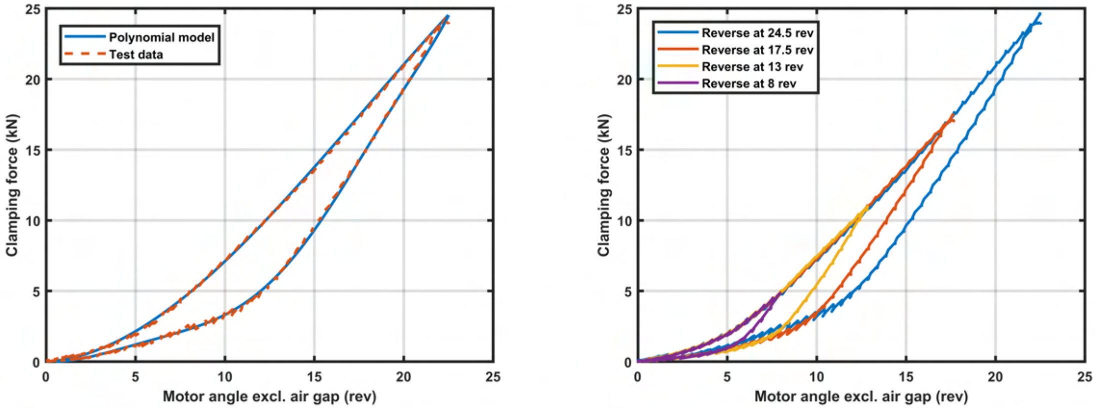
It has been shown that the nonlinearity can be approximated nicely by polynomials as functions of motor angle [31, 46]. Figure 3.4 shows various clamping force from measurements. From Fig. 3.4(a), we can see the polynomial model can be used to fit the test data pretty well. However, the hysteresis cannot be presented by the memoryless polynomial model. There are a few algorithms that try to incorporate the hysteresis into the clamping force model



(a) Motor input profile

(b) Clamping force

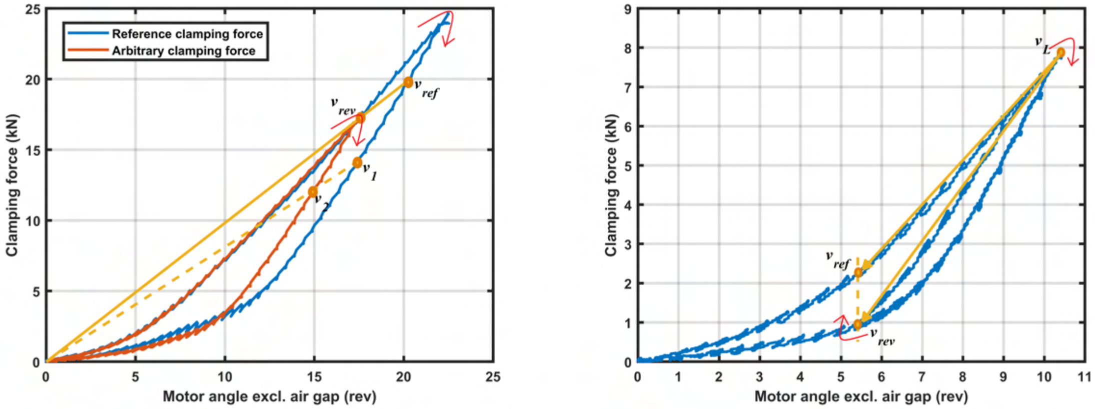
Figure 3.3: Clamping force of EMB



(a) Test data compared with polynomial stiffness model

(b) Clamping force of various reverse points

Figure 3.4: Clamping force measurement



(a) Releasing clamping force

(b) Applying clamping force

Figure 3.5: Clamping force modeling

[47], yet they usually involve complicated models and extensive experiments, which are not applicable for real-time estimation.

In order to capture the hysteresis while preserving the simplicity of the polynomial model, an alternative solution is proposed here. We first study the transition from the applying to the releasing operation. Figure 3.4(b) shows the test results of the clamping force starting from the same position and then reversing at various points. From Fig. 3.4(b), we see that all the curves initially follow the same trajectory during the applying. From the overlapped applying curves, the clamping force trajectories begin to differentiate during the releasing operation according to their reverse points. One observation is that the releasing clamping forces are almost parallel to each other despite differences in the reverse points. This indicates that the clamping force with different reverse points might be derived from one reference curve.

Based on the above observation, the approach to obtaining the releasing clamping force operation is illustrated by Fig. 3.5(a). The blue curve is called the reference clamping force, which represents the envelope of the EMB operation region. It is modeled by the polynomial functions. The red curve is an arbitrary trajectory of the clamping force. Taking the contact point as the origin, two vectors are constructed, $v_{rev} = \begin{bmatrix} F_{rev} & \theta_{rev} \end{bmatrix}^T$ and $v_{ref} = \begin{bmatrix} F_{ref} & \theta_{ref} \end{bmatrix}^T$, which are the vector from the origin to the reverse point and its extension on the reference curve, respectively. The ratio between v_{rev} and v_{ref} is

$$r_r = \frac{\|v_{rev}\|}{\|v_{ref}\|}, \tag{3.19}$$

where $\|\cdot\|$ denotes the 2-norm.

Using r_r, the unknown clamping force at point v_2 can be calculated by scaling the vector v_1 on the reference curve

$$\begin{bmatrix} \widehat{F}_r(\theta_2) \\ \theta_2 \end{bmatrix} = \begin{bmatrix} P_{ref}(\theta_1) \\ \theta_1 \end{bmatrix} r_r, \tag{3.20}$$

where θ_1 and θ_2 are respectively the motor angle corresponding to v_1 and v_2, $\widehat{F}_r$ is the releasing clamping force of interest, and P_r is the polynomial model of the reference releasing clamping force.

The applying clamping force is modeled similarly, as shown in Fig. 3.5(b). The idea is to model the unknown clamping force by scaling and rotating the reference curve. Equations (21) and (22) give the scaling factor and rotation angle

$$r_a = \frac{\|v_{rev} - v_L\|}{\|v_{ref} - v_L\|}, \tag{3.21}$$

$$\cos\gamma = \frac{\langle v_{rev} - v_L, v_{ref} - v_L \rangle}{\|v_{rev} - v_L\| \, \|v_{ref} - v_L\|}, \tag{3.22}$$

where v_{rev} is the current reverse point, v_L is the last reverse point, v_{ref} is the applying clamping force on the reference curve at the same motor angle as that of v_{rev}, and $\langle\cdot\rangle$ denotes the inner product.

The unknown applying clamping force is mapped by the reference clamping force between v_{ref} and v_L using follwing equation

$$\begin{bmatrix} \widehat{F}_a(\theta_a^*) - F_L \\ \theta_a^* - \theta_L \end{bmatrix} = \begin{bmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{bmatrix} \begin{bmatrix} P_a(\theta_a) - F_L \\ \theta_a - \theta_L \end{bmatrix} r_a, \tag{3.23}$$

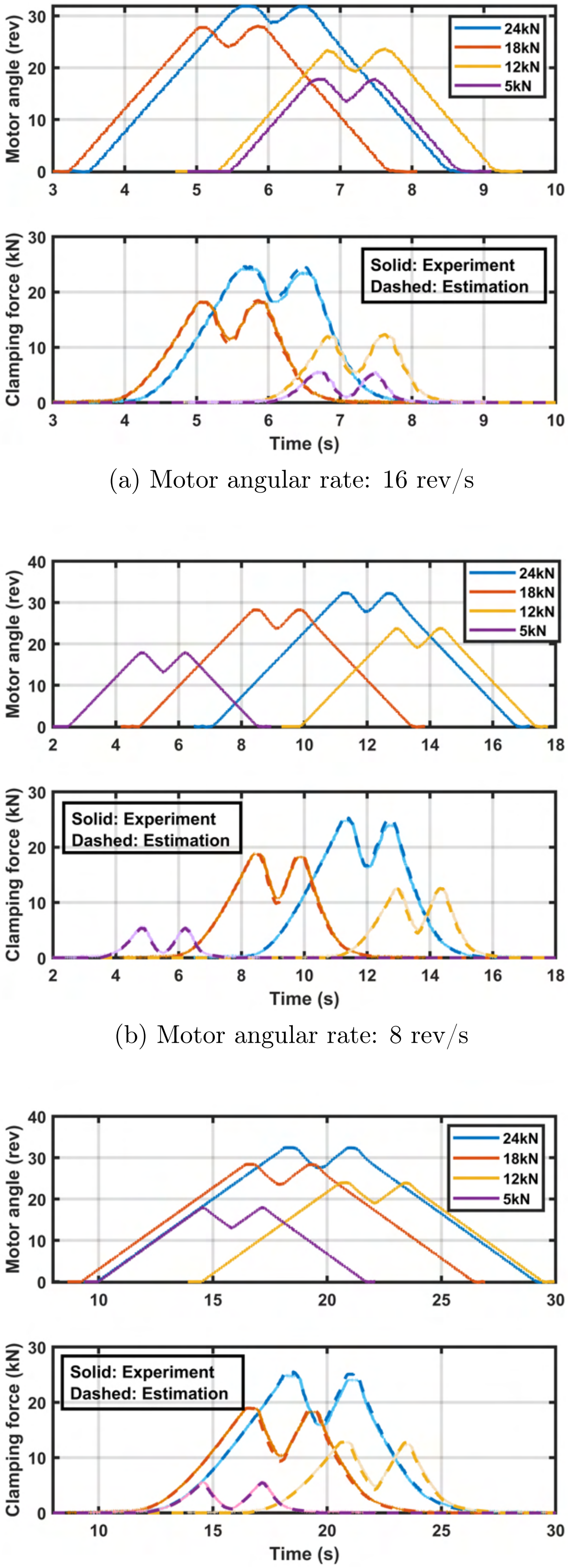
where θ_a and θ_a^* are respectively the motor angle of the reference and the modeled applying clamping force, $\widehat{F}_a$ is the applying clamping force of interest, and P_a is the polynomial model of the reference applying clamping force.

The proposed clamping force model produces the hysteresis by simple linear transformation without introducing additional dynamics and parameters, which is suitable for real-time implementation for clamping force control.

To validate the proposed clamping force model, experimental tests have been conducted.

Motor angle ramp inputs with different rate are performed to achieve different clamping force profiles. Then the clamping force measurements are compared with the output of the estimation to validate its performance. The results are shown in Fig. 3.6, where the input rates are 16, 8, and 4 rev/s and the maximum clamping forces are 24, 18, 12, and 5kN.

We can see that the clamping force of the simulation agrees closely with the experimental data under various input conditions. This demonstrates the proposed clamping force model can be used to produce accurate clamping force.



(a) Motor angular rate: 16 rev/s

(b) Motor angular rate: 8 rev/s

(c) Motor angular rate: 4 rev/s

Figure 3.6: Comparison between clamping force model and experimental results

3.3 Clamping Force Control System Development

The objective of the clamping force control system is to realize the clamping force requests that might be from either the driver or the high-level control systems. Despite its simple and generic SISO control structure, the clamping force control is subject to several practical challenges. First, the clamping force is not measured, due to both the cost and the technical constraints of installing a load cell on the brake pad. Second, the gap management algorithm should be properly integrated to achieve a seamless transition between gap closing and force tracking control. Last but not least, the controller needs to be robust against model nonlinearities and disturbances such as varying pad stiffness and friction.

A high-level schematic of the proposed clamping force control system in Fig. 3.7. As shown in Fig. 3.7, clamping force F_{cl} is estimated online using the available motor angular position and speed measurements and a novel clamping force model. Motor torque command T_m is generated from a disturbance-observer-based tracking controller with robustness to model uncertainty and disturbance. The reference clamping force F_{cl}^* is given by the gap management block which unifies the gap closing and force tracking without controller switching.

3.3.1 Clamping Force Estimator

There are two unknown inputs in the system, i.e., the clamping force F_{cl} and the lumped friction torque on the motor shaft T_f. One may attempt to use the common unknown input disturbance observer (UIDO) to estimate these unknown disturbances. However, the real challenge here is the lack of observability of this system. The idea of UIDO is to augment the unknown disturbance into the system states by constructing an auxiliary dynamic system based on which the disturbances are generated. The augmented plant is described as

$$\begin{bmatrix} \dot{\theta}_m \\ \ddot{\theta}_m \\ \dot{z} \end{bmatrix} = \begin{bmatrix} A & B_d C_d \\ 0 & A_d \end{bmatrix} \begin{bmatrix} \theta_m \\ \dot{\theta}_m \\ z \end{bmatrix} + \begin{bmatrix} B_u \\ 0 \end{bmatrix} T_m, \tag{3.24}$$

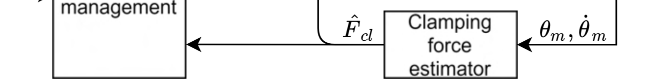


Figure 3.7: High-level schematic of clamping force control system

$$y = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} \theta_m \\ \dot{\theta}_m \\ z \end{bmatrix}, \tag{3.25}$$

where $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, $B_u = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $A = \begin{bmatrix} 0 & 0 \\ -i_{tot}/J & -1/J \end{bmatrix}$, and $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. J is the lumped inertia on the motor shaft, i_{tot} is the total transmission ratio of reduction gear and ball screw. T_m is the motor torque, and z is the states of disturbances.

Then an observer can be designed to estimate the augmented states if the pair

$$\left\{ \begin{bmatrix} A & B_d C_d \\ 0 & A_d \end{bmatrix}, \begin{bmatrix} C & 0 \end{bmatrix} \right\} \tag{3.26}$$

is observable.

A frequent choice is to model the disturbances as piecewise constant inputs, which gives $A_d = 0$ and $C_d = I$. However, it turns out that the pair in (3.26) is not observable in this way. An intuitive explanation is that F_{cl} and T_f perturb the state equation in the same way through the same channel. Therefore, it is impossible to differentiate them by the observer. A solution is to replace the fictitious dynamics of the step-wise friction model with a physical-based model, e.g., the LuGre model as in [48]. However, such a method will require the friction parameters to be precisely known. This is a rather strong and impractical assumption, considering that the friction is supposed to be an “unknown disturbance” in the first place.

In our study, an alternative approach for clamping force estimation is used. The algorithm is based on the clamping force model introduced in Section 3.2.2, where the clamping force is derived from two reference curves, respectively for applying and releasing. The clamping force estimation is then equivalent to finding these two reference curves.

To obtain the coefficients of the polynomial model of the reference curves, we use an idea similar to that first introduced in [49]. The torque balance of the EMB system is given by

$$T_{m,A} = F_{cl,A} i_{tot} + J\ddot{\theta}_{m,A} + T_{f,A}, \tag{3.27}$$

$$T_{m,R} = F_{cl,R} i_{tot} + J\ddot{\theta}_{m,R} + T_{f,R}, \tag{3.28}$$

where the subscripts A and R denote the applying and releasing operation, respectively.

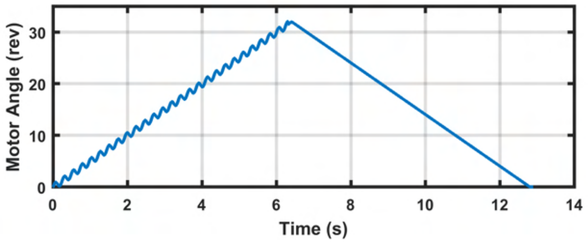


Figure 3.8: Motor angle input for clamping force estimation in applying

Adding Eqns. (3.27) and (3.28) together and rearranging the equation, we get

$$F_{cl,A} + F_{cl,R} = \frac{1}{i_{tot}} \left[T_{m,A} + T_{m,R} - J\ddot{\theta}_{m,A} - J\ddot{\theta}_{m,R} - T_{f,A} - T_{f,R} \right]. \tag{3.29}$$

Equation (3.29) can be used for clamping force estimation given several assumptions. Firstly, it is assumed that the absolute value of T_f is determined by θ_m only, that is, $|T_f| = f(\theta_m)$. Secondly, we assume that the hysteresis is small if the motor angle amplitude is small.

Given the preceding assumptions, when $\theta_{m,A} = \theta_{m,R}$ and $\dot{\theta}_{m,A}\dot{\theta}_{m,R} < 0$, the following relationships hold

$$T_{f,A} = -T_{f,R}, \tag{3.30}$$

$$F_{cl,A} = F_{cl,,R}. \tag{3.31}$$

Thus Eq. (3.29) can be simplified to

$$F_{cl} = \frac{1}{2i_{tot}} \left[T_{m,A} + T_{m,R} - J\ddot{\theta}_{m,A} - J\ddot{\theta}_{m,R} \right], \tag{3.32}$$

where T_m is directly measured and $\ddot{\theta}_m$ can be obtained by the numerical derivative of measurement $\dot{\theta}_m$.

In Eq. (3.32), clamping force is obtained by the readily available motor torque and angular speed. Then this estimation is used for polynomial reference curves identification for the clamping force model. For implementation, the algorithm requires the motor to pass the same angular position in opposite directions within a small amplitude. A specific input profile is designed to facilitate the procedure, as shown in Fig. 3.8. It is shown in Fig. 3.8 that the motor angle consists of two components, a slow ramp input with large amplitude superimposed by a high frequency sinusoid with small amplitude. The amplitude of the ramp input decides the range of motor angles that the stiffness curves cover and produces a large hysteresis. The sine wave enforces the motor passing the same angle in opposite directions

Table 3.1: The sign of measurements and their derivatives before/after the contact

Mode	$\operatorname{sgn}(T_m)$	$\operatorname{sgn}\left(\dot{T}_m\right)$	$\operatorname{sgn}\left(\dot{\theta}_m\right)$	$\operatorname{sgn}\left(\ddot{\theta}_m\right)$
Speed control	1/1	0/1	1/1	0/-1

in each period, and the small amplitude ensures that the local hysteresis is not excited too much. For the releasing curve, the sinusoid will be added to the descending branch instead.

The shape of the reference curve will change due to pad wear and temperature [49]. Heuristic approaches can be used to compensate for their effect. Because the wear condition changes slowly during the lifespan of the brake system, the characteristic curve can be reidentified regularly in the maintenance process. It is shown in [49] that the reference curves at different temperatures can be differentiated by a scaling factor. Therefore, reference curves at different temperature can be adapted from a basic curve using a scaling factor as a function of temperature [50].

3.3.2 Gap Distance Estimation

When the EMB is at rest, a certain air gap between the brake pad and disk must be maintained to avoid unintended braking and overheating. However, the air gap distance will gradually vary from its initial setup due to mechanical wear, degrading the performance of clamping force estimation and control. Therefore, contact detection and air gap distance estimation is indispensable for EMB clamping force control.

Due to the lack of an automatic wear adjustment mechanism [51] and contact pin [52] in our EMB system, the contact point must be estimated through the existing measurements. Depending on how the motor is controlled, measurements will change differently upon contact. In previous studies, various signals such as I_a [47] and $\frac{dT_m}{d\theta_m}$ [49] have been used as indicators for contact detection. Table 3.1 gives the sign of the measurements and their derivatives before and after the instant of contact under motor speed control.

Under motor speed control, $\operatorname{sgn}\left(\dot{T}_m\right) = 0$ before contact, as only constant T_m is needed to compensate for the bearing friction. Also $\operatorname{sgn}\left(\ddot{\theta}_m\right) = 0$ because the motor speed is regulated to the constant set point. As soon as contact occurs, motor speed will immediately decrease due to the generated clamping force. Furthermore, $\operatorname{sgn}\left(\dot{T}_m\right)$ will also change to positive, because the controller will request more control effort trying to keep the speed to the set point. Therefore, among the available measurements, both $\dot{T}_m$ and $\ddot{\theta}_m$ are suitable for

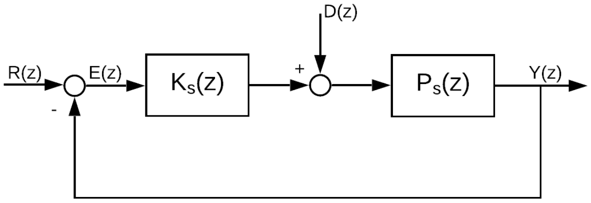


Figure 3.9: Closed loop speed control system

contact detection, as their directions change immediately upon contact.

In this study, $\ddot{\theta}_m$ is chosen for contact detection. The condition is expressed as

$$\left|\ddot{\theta}_m\right| > S, \tag{3.33}$$

where S is a small positive number to avoid false detection due to noise.

Choosing $\ddot{\theta}_m$ as the indicator of contact, two tasks are left. First is to design a speed controller such that the motor speed reaches the set point before the contact. The other is to derive $\ddot{\theta}_m$ from the noisy measurement of $\dot{\theta}_m$.

The motor speed control model is written as

$$J\dot{\omega}_m = T_m - B\omega_m + d, \tag{3.34}$$

where ω_m is the motor angular speed, B is the damping coefficient approximating the viscous friction, and d is the external disturbance, including friction and clamping force.

The closed loop system is shown in Fig. 3.9. In Fig. 3.9, $P_s\left(z\right)$ is the discretized plant by the zero-order-hold method, and $K_s\left(z\right)$ is the controller to be designed. This is a classic motor speed control problem. However, for our application, it is not a good idea to make the controller yield the best tracking and disturbance rejection, because it will mask the change of $\ddot{\theta}_m$ and raise the difficulty in detecting the contact point. Rather, we want the controller to work just fine before the contact and then degrade after contact such that $\ddot{\theta}_m$ changes as much as possible upon the contact.

Based on the preceding reasoning, the well-known internal model principle (IMP) [53] is used to design the controller. The key difference in control before and after contact is the

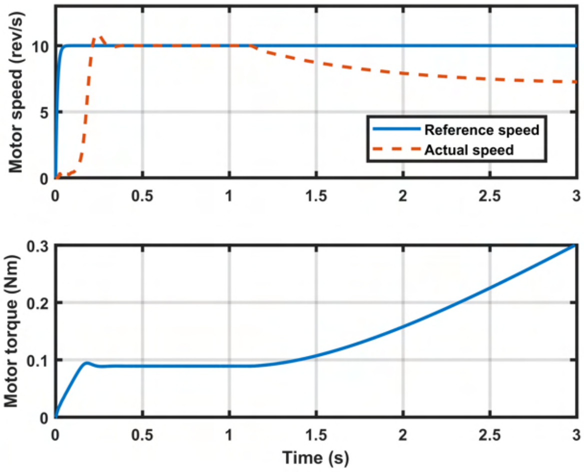


Figure 3.10: Simulation results of IMP based speed control

type of disturbance. The disturbance pre-contact is mainly bearing friction, which resembles a step disturbance, while the post-contact disturbance largely comes from the clamping force, which appears like a nonlinear spring. The IMP tells us that in order to reject the input disturbance asymptotically, the feedback controller must incorporate the “structure” of the disturbance into it. For a step disturbance, the transfer function of controller thus must have the term $1 - z^{-1}$ in this denominator.

The controller and plant are written as

$$K_s\left(z^{-1}\right) = \frac{N_k\left(z^{-1}\right)}{D_k\left(z^{-1}\right) D_d\left(z^{-1}\right)}, \tag{3.35}$$

$$P_s\left(z^{-1}\right) = \frac{z^{-1} N_P\left(z^{-1}\right)}{D_P\left(z^{-1}\right)}. \tag{3.36}$$

According to the IMP and the control objective, we choose $D_d\left(z^{-1}\right) = 1 - z^{-1}$ to reject the constant friction disturbance only. Then $N_k\left(z^{-1}\right)$ and $D_k\left(z^{-1}\right)$ are determined by solving the Diophantine equation

$$z^{-1} N_P\left(z^{-1}\right) N_k\left(z^{-1}\right) + D_P\left(z^{-1}\right)\left(1 - z^{-1}\right) D_k\left(z^{-1}\right) = \eta\left(z^{-1}\right), \tag{3.37}$$

where $\eta\left(z^{-1}\right) = 1 + \eta_1 z^{-1} + \eta_1 z^{-1} + \ldots + \eta_q z^{-q}$ is the desired closed-loop characteristic polynomial.

Figure 3.10 shows the simulation of the IMP controller. The reference speed input is a step with amplitude of 10 rev/s passed through a 1st order low-pass filter. We can see from

Fig. 3.10 that the settling time of motor speed is about 0.3s. There is no steady-state error after the transient. At 1.1s, when the brake pad touches the disk, the motor speed diverges from the set point immediately and the motor torque also starts to increase. This is expected because the IMP-based controller is able to handle only a constant disturbance.

Now the task is to calculate the motor angular acceleration from the speed measurement. A Rauch–Tung–Striebel (RTS) smoother is used here to obtain the acceleration from a noisy speed measurement. The RTS smoother is a two-way Kalman filter that finds the optimal estimate of the state at each time step k while using the measurements up to and including the final time N [54].

The discrete-time estimation model for the motor speed and acceleration is constructed as

$$\begin{bmatrix} \dot{\theta}_{m,k+1} \\ \ddot{\theta}_{m,k+1} \end{bmatrix} = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_{m,k} \\ \ddot{\theta}_{m,k} \end{bmatrix} + \begin{bmatrix} \frac{T_s^2}{2} \\ T_s \end{bmatrix} \dddot{\theta}_{m,k}, \tag{3.38}$$

$$\dot{\theta}_{m,k} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_{m,k} \\ \ddot{\theta}_{m,k} \end{bmatrix} + v_k, \tag{3.39}$$

where T_s is the sample time and $\dddot{\theta}_{m,k} \sim (0, Q)$ and $\dddot{v}_k \sim (0, R)$ are respectively the process and measurement noise, assuming both Gaussian, zero-mean, uncorrelated, and white.

The state of the system is first estimated by a standard forward Kalman filter

$$P_{f,k}^- = F P_{f,k-1}^+ F^T + G Q G^T, \tag{3.40}$$

$$K_{f,k} = P_{f,k}^- H^T \left(H P_{f,k}^{-1} H^T + R \right)^{-1}, \tag{3.41}$$

$$\widehat{x}_{f,k}^- = F \widehat{x}_{f,k-1}^+, \tag{3.42}$$

$$\widehat{x}_{f,k}^+ = \widehat{x}_{f,k}^- + K_{f,k} \left(y_k - H \widehat{x}_{f,k}^- \right), \tag{3.43}$$

$$P_{f,k}^+ = \left(I - K_{f,k} H \right) P_{f,k}^- \left(I - K_{f,k} H \right)^T + K_{f,k} R K_{f,k}^T. \tag{3.44}$$

Then the following RTS smoother equations are initialized and executed backwards for $k = N-1, \ldots, 1, 0$

$$\widehat{x}_N = \widehat{x}_{f,N}^+, \tag{3.45}$$

$$P_N = P_{f,N}^+, \tag{3.46}$$

$$I_{f,k+1}^- = \left(P_{f,k-1}^- \right)^{-1}, \tag{3.47}$$

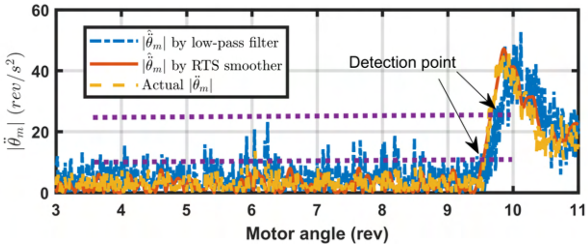


Figure 3.11: Simulation results of contact detection

$$K_k = P_{f,k}^{+} F^T I_{f,k+1}^{-}, \tag{3.48}$$

$$P_k = P_{f,k}^{+} - K_k \left(P_{f,k+1}^{-} - P_{k+1} \right) K_k^T, \tag{3.49}$$

$$\widehat{x}_k = \widehat{x}_{f,k}^{+} + K_k \left(\widehat{x}_{k+1} - \widehat{x}_{f,k+1}^{-} \right), \tag{3.50}$$

where matrices F, G, and H are the state, input, and output matrix in (3.38) and (3.39), respectively. Q and R are the covariance of process and measurement noise, respectively. P_f is the state covariance matrix. K_f is the Kalman filter gain. $\widehat{x}_f$ is the state estimate. y is the measurement. The superscript $+$ and $-$ represent a posteriori and a priori estimate, respectively. Details on the RTS smoother can be found in [54].

Figure 3.11 shows the simulation results of contact detection. The real air gap between the brake pad and disk is 9.5 motor revolutions. As shown in Fig. 3.11, the motor acceleration estimation from the RTS smoother is very close to the actual acceleration. The threshold S is set as small as 11rev/s^2 and the resulting gap distance estimation is 9.51rev. Figure 3.11 also shows the acceleration estimation using finite difference passed by a 1st order low-pass filter. The obtained signal not only suffers from significant noise, but also shows noticeable phase lag. In order to rule out the fault detection, the threshold must be raised to 25rev/s^2, which gives an overestimated air gap of 9.72rev.

In summary, the proposed distance detection and distance estimation algorithm is fast and accurate. In practice, the whole procedure should be performed on a regular basis when the vehicle starts up. The estimated gap distance can be used for adjusting the initial motor position to maintain the preset value.

3.3.3 Clamping Force Tracking

The clamping force is controlled to track the reference by the motor. Generally, the EMB system should have a fast and robust response with respect to disturbances and model uncertainties. The mathematical model for the EMB clamping force control can formulated

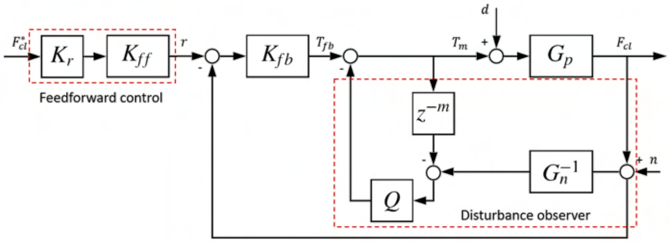


Figure 3.12: Block diagram of the DOB control of clamping force

as

$$J\ddot{\theta}_m = T_m - i_{tot}K_{cl}\theta_m - T_f, \tag{3.51}$$

where K_{cl} is a time-varying nonlinear spring element to capture the nonlinearity and hysteresis of the actual clamping force.

A disturbance-observer-based (DOB) control is used in our study. This method has been proven to be very effective in high-precision position control such as hard disk drives [55] and positioning tables [56]. The control schematic is shown in Fig.3.12. Each block in Fig. 3.12 represents a discrete-time transfer function. On the prototype EMB system, the CAN sample time is 2ms, which is used for clamping force regulation, while the DC motor current controller is operated at a much faster 200μs sample time. Therefore, the motor controller is neglected in the following study.

In Fig. 3.12, $G_p\left(z^{-1}\right)$ is the plant model with unknown parameters and dynamics. $G_n^{-1}\left(z^{-1}\right)$ is the inverse of the nominal model factorized out of the delay term z^{-m}. $Q\left(z^{-1}\right)$ is a low-pass filter, whose role will be discussed shortly. $r\left(k\right)$, $T_{fb}\left(k\right)$, $d\left(k\right)$, and $n\left(k\right)$ are the reference, feedback control input, disturbance torque, and measurement noise, respectively. By a straightforward algebra, the transfer function from $T_{fb}\left(k\right)$, $d\left(k\right)$, and $n\left(k\right)$ to the output $F_{cl}\left(k\right)$ are respectively expressed as

$$G_{F_{cl}T_{fb}} = \frac{G_p\left(z^{-1}\right)G_n\left(z^{-1}\right)}{G_n\left(z^{-1}\right) + \left(G_p\left(z^{-1}\right) - G_n\left(z^{-1}\right)z^{-m}\right)Q\left(z^{-1}\right)}, \tag{3.52}$$

$$G_{F_{cl}d} = \frac{G_p\left(z^{-1}\right)G_n\left(z^{-1}\right)\left(1 - Q\left(z^{-1}\right)z^{-m}\right)}{G_n\left(z^{-1}\right) + \left(G_p\left(z^{-1}\right) - G_n\left(z^{-1}\right)z^{-m}\right)Q\left(z^{-1}\right)}, \tag{3.53}$$

$$G_{F_{cl}n} = \frac{-G_p\left(z^{-1}\right)Q\left(z^{-1}\right)}{G_n\left(z^{-1}\right) + \left(G_p\left(z^{-1}\right) - G_n\left(z^{-1}\right)z^{-m}\right)Q\left(z^{-1}\right)}. \tag{3.54}$$

Chosen as a low-pass filter, $Q\left(z^{-1}\right) \cong 1$ in the low-frequency range and $Q\left(z^{-1}\right) \cong 0$

Table 3.2: Asymptotic properties of transfer functions with DOB control

	Low frequency where $Q\left(z^{-1}\right) \cong 1$	High frequency where $Q\left(z^{-1}\right) \cong 0$
$G_{F_{cl}T_{fb}}$	$G_n\left(z^{-1}\right)$	$G_p\left(z^{-1}\right)$
$G_{F_{cl}d}$	0	$G_p\left(z^{-1}\right)$
$G_{F_{cl}n}$	-1	0

in the high-frequency range. If the delay is small, such that $G_n\left(z^{-1}\right) \cong G_n\left(z^{-1}\right) z^{-m}$, then Equations (3.52)–(3.54) can be approximated to Table 3.2.

From Table 3.2, the DOB control has the following key features:

(1) In low frequency, the control system resembles the nominal plant $G_n\left(z^{-1}\right)$, and the disturbance is rejected.

(2) In high frequency, the control system degrades to the original plant $G_p\left(z^{-1}\right)$, and the effect of measurement noise is attenuated.

The first feature is particularly useful, because it allows feedback and feedforward controllers to be designed on the nominal system, regardless of the model uncertainties and disturbances. For our application, it means that the nonlinearity and hysteresis of the clamping force can be regulated to a constant linear spring element. Then simple linear controllers can be used in the outer loop for reference tracking without introducing any complex nonlinear or adaptation techniques.

The second feature tells us that the bandwidth of $Q\left(z^{-1}\right)$ must be limited according to the measurement noise, a typical trade-off in feedback controller design. Moreover, the design of the Q-filter also closely relates to the robust stability of the uncertain system by representing the uncertainty as a multiplicative perturbation

$$G_p\left(z^{-1}\right) = G_n^*\left(z^{-1}\right)\left(1 + \Delta\left(z^{-1}\right)\right), \tag{3.55}$$

where $G_n^*\left(z^{-1}\right) = G_n\left(z^{-1}\right) z^{-m}$.

A sufficient condition of robust stability is given by

$$\left|\Delta\left(e^{-j\omega T}\right) Q\left(e^{-j\omega T}\right)\right| < 1, \forall\omega. \tag{3.56}$$

Table 3.3: Uncertain parameters of plant model

Parameters	Nominal value	Variation (%)
J	$1.1 \times 10^{-5}(\mathrm{Kgm}^2)$	$[-10, 10]$
K_{cl}	$0.22(\mathrm{kN}/\mathrm{rev})$	$[-15, 70]$
B	$2 \times 10^{-4}(\mathrm{Nms}/\mathrm{rad})$	$[-10, 70]$
τ_m	$1(\mathrm{ms})$	$[-10, 10]$

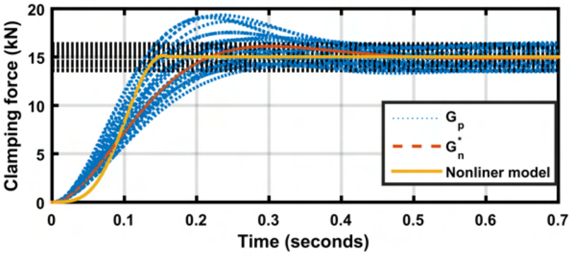


Figure 3.13: Step response of linear and nonlinear EMB models

A candidate Q-filter suggested in [57] is used in this paper

$$Q(s) = \frac{3\tau s + 1}{(\tau s + 1)^3}, \tag{3.57}$$

where τ is the time constant of the Q-filter. The Q-filter is discretized using bilinear transformation for implementation.

In order to determine the time constant of the Q-filter, $\Delta(z^{-1})$ is calculated using the following equation

$$\Delta\left(z^{-1}\right) = \frac{G_p(z^{-1}) - G_n^*(z^{-1})}{G_n^*(z^{-1})}. \tag{3.58}$$

In Eq. (3.58), $G_p(z^{-1})$ is a zero-order-hold discretization of the continuous plant with uncertain parameters

$$G_p(s) = \frac{K_{cl}}{Js^2 + Bs + i_{tot}K_{cl}} \frac{1}{\tau_m s + 1}. \tag{3.59}$$

The nominal value and variation of uncertain parameters in $G_p(s)$ are summarized in Table 3.3.

$G_n^*(z^{-1})$ is defined by taking the nominal values in Table 3.3 and letting $\tau_m = 0$. Figure 3.13 shows the step response of uncertain model $G_p(s)$, nominal model $G_n^*(z^{-1})$, and nonlinear plant model with an amplitude of 15kN. We can see that the nominal model and the nonlinear

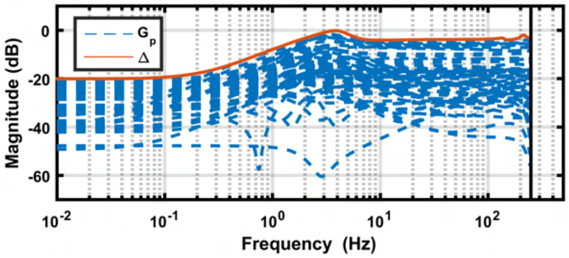


Figure 3.14: Magnitude response of $G_p\left(z^{-1}\right)$ and $\Delta\left(z^{-1}\right)$

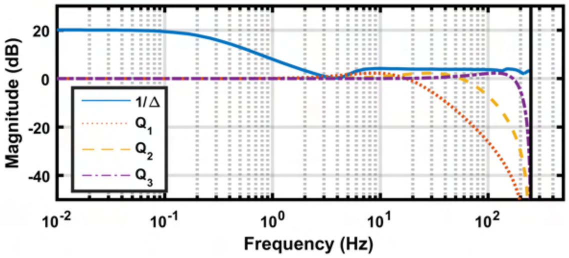


Figure 3.15: Q-filter bandwidth selections

model have the same steady-state value. However, there is discrepancy in the transient response. The low stiffness in the small clamping force makes the nonlinear model response slower than that of the linear model, while high stiffness in the large clamping force makes its response faster. Although the nonlinear model cannot be represented by a single nominal model, its response is mostly covered by the set of the uncertain model. Therefore, if the proposed controller can provide robust performance against the uncertain model, it will also give similar performance to the nonlinear model. Figure 3.14 shows the magnitude response of 60 random samples of $G_p\left(z^{-1}\right)$ and $\Delta\left(z^{-1}\right)$ as their upper boundary.

According to the stability condition, τ should be selected such that (3.56) is satisfied. Figure 3.15 shows 3 candidates of Q-filter with comparison to the $^1/_\Delta$. It is seen that among the three of them, $Q_1\left(z^{-1}\right)$ does not meet the condition, while the other two do. The bandwidth of the Q-filter also should not be too high, due to the noise and possible unmodeled dynamics in the high frequency. Therefore, $Q_2\left(z^{-1}\right)$ is applied.

After determining the disturbance observer, the feedback and feedforward controller design is straightforward. A PI controller is used for feedback control. The loop shaping design method is applied such that the gain crossover frequency and phase margin meet the specifications, which are written as the following equations

$$\left|K_{fb}\left(e^{j\omega_c T}\right) G_n\left(e^{-j\omega_c T}\right)\right| = 0, \tag{3.60}$$

$$\arg\left[K_{fb}\left(e^{j\omega_c T}\right) G_n\left(e^{-j\omega_c T}\right)\right] = PM - \pi, \tag{3.61}$$

where is $K_{fb}\left(e^{j\omega_c T}\right)$ the freqeuncy response of the discrete-time PI controller, ω_c is the gain crossover frequency of the loop transfer function, and PM is the phase margin.

Originally introduced by Tomizuka [58], a zero-phase error tracking controller (ZPETC) is designed as feedforward controller based on the approximated inversion of the closed-loop system. Denote the closed-loop system with the nominal model and PI controller as

$$G_{cl}\left(z^{-1}\right) = \frac{z^{-d} B_c^s\left(z^{-1}\right) B_c^u\left(z^{-1}\right)}{A_c\left(z^{-1}\right)}, \tag{3.62}$$

where $d = 1$ is the relative degree of the transfer function, $B_c^s\left(z^{-1}\right)$ is the part of the numerator that is suitable to invert, and $B_c^u\left(z^{-1}\right) = b_{c0}^u + b_{c1}^u z^{-1} + \ldots + b_{cs}^u z^{-s}$ is the one that is not suitable to invert, which must include unstable zeros outside the unit circle. All three zeros of $G_{cl}\left(z^{-1}\right)$ are inside the unit circle in our case, but the zero that locates near $(-1, 0)$ is not inverted in order to avoid high frequency oscillation.

The ZPETC is given by

$$K_{ff}\left(z^{-1}\right) = \frac{z^d A_c\left(z^{-1}\right) B_c^u(z)}{B_c^s\left(z^{-1}\right)\left[B_c^u(1)\right]^2}. \tag{3.63}$$

Notice that z^d and $B_c^u(z) = b_{c0}^u + b_{c1}^u z + \ldots + b_{cs}^u z^s$ are not causal. Therefore, a total of $d + s$ steps preview of the reference input is needed.

An extra reference model $K_r\left(z^{-1}\right)$ is added in addition to $K_{ff}\left(z^{-1}\right)$ to further shape the required input-output response of the overall system. Figure 3.16 shows the Bode plot of $K_r\left(z^{-1}\right)$ and $K_r\left(z^{-1}\right) K_{ff}\left(z^{-1}\right) G_{cl}\left(z^{-1}\right)$. It can be seen that the frequency response of the overall system is very similar to that of the reference model. Note that their phase responses are essentially identical thanks to the ZPETC.

3.3.4 Unified Control Structure

An underlying assumption that has been made throughout the development of the tracking controller is that the brake pad already touches the disk at the operation of the controller.

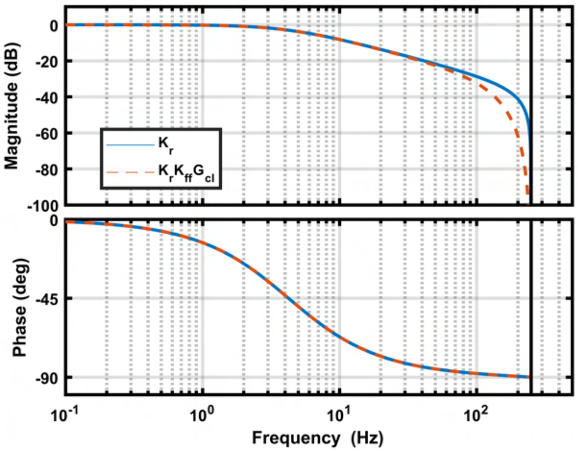


Figure 3.16: Frequency response of $K_r\left(z^{-1}\right)$ and $K_r\left(z^{-1}\right) K_{ff}\left(z^{-1}\right) G_{cl}\left(z^{-1}\right)$

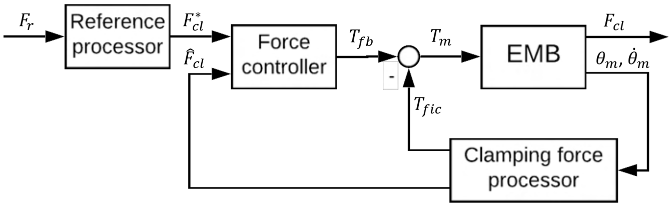


Figure 3.17: High-level schematic of unified control system

However, as we know, there is an initial air gap between the brake pad and disk when the EMB starts from rest. This gap must be handled carefully; otherwise, a large impact will occur upon contact and damage the system. One approach that is often used is to apply an additional position controller to eliminate the initial gap and then switch back to the force controller [36, 43] at contact. However, the whole control system becomes complicated due to the extra position controller. Furthermore, smooth transition generally cannot be automatically achieved, even if each controller is well designed separately.

Here, a unified framework is proposed so that the same DOB-based tracking controller can be used for gap closing, resulting in a seamless transition. Since the DOB-based controller takes the force as reference, the motor position must be converted to force in order to utilize the same controller. Figure 3.17 depicts the high-level schematic of the unified control system.

Two additional blocks are used for the unified system. In the clamping force processor, the estimated clamping force is augmented by a fictitious force, as shown in Fig. 3.18. As shown in Fig. 3.18, the actual clamping force is elevated by F_G, and a segment of linear

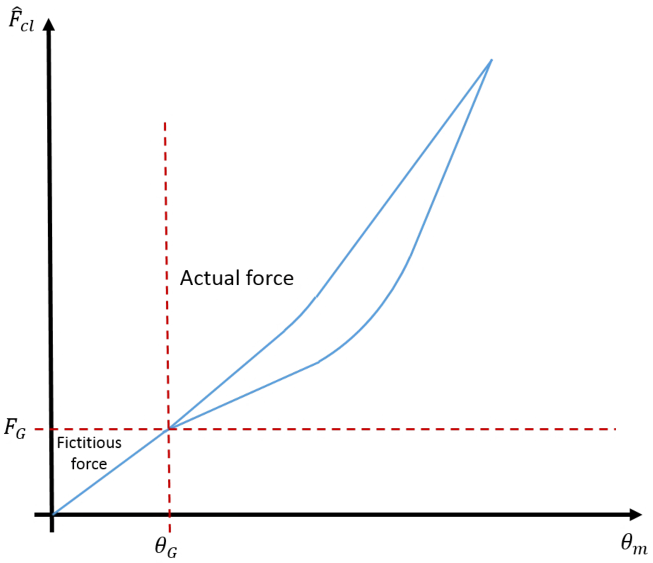


Figure 3.18: Augmented clamping force

fictitious force is inserted by connecting the origin and (θ_G, F_G), where θ_G is the initial gap distance. This fictitious force is responsible for converting the motor position to a corresponding clamping force. F_G is set such that the slop of the fictitious force is equal to the nominal stiffness to retain the best performance of the force control. Because the fictitious force F_{fic} cannot be actually generated by the plant, it will be added to the motor torque command to mimic its effect.

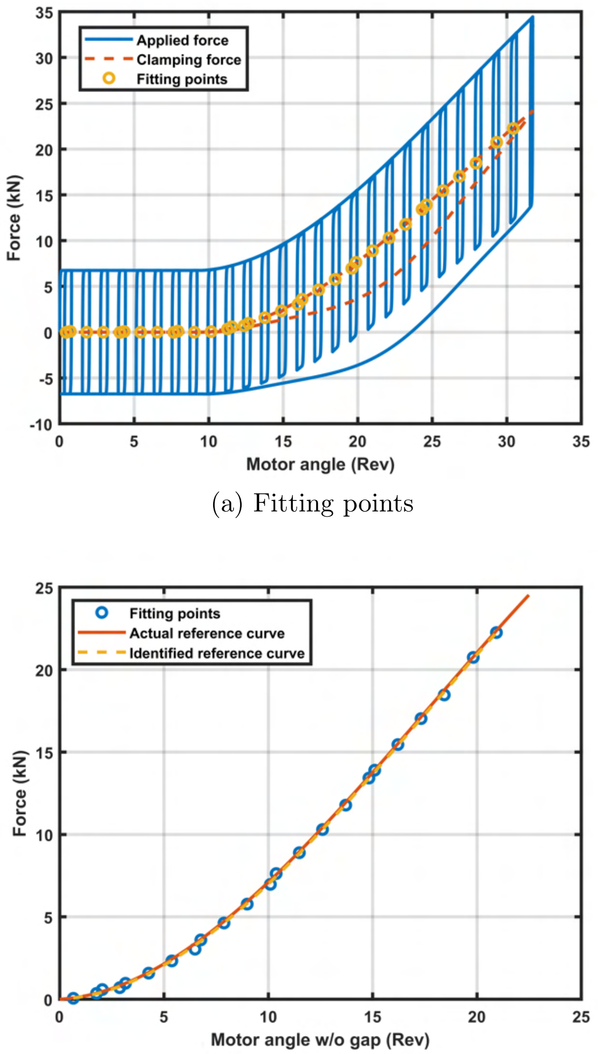
The reference processor in Fig. 3.17 modifies the reference force to accommodate the fictitious force. Equation below gives the modified reference

$$F_{cl}^{*} = \begin{cases} \omega_G \frac{F_G}{\theta_G} t & \text{if } \theta_m < \theta_G \\ F_r + F_G & \text{else} \end{cases}, \tag{3.64}$$

where F_r is the effective reference force and ω_G is the designed travel speed. The larger ω_G is, the faster the gap is eliminated. We see that the proposed method not only unifies the force and the position control but also provides a simple tuning knob for adjusting the speed of gap closing.

3.4 Simulation Results

The controller is implemented in Matlab/Simulink to verify its performance. The nonlinear EMB model in Section 3.2 is used as the plant. Model parameters used in the simulations are listed in *Appendix.*



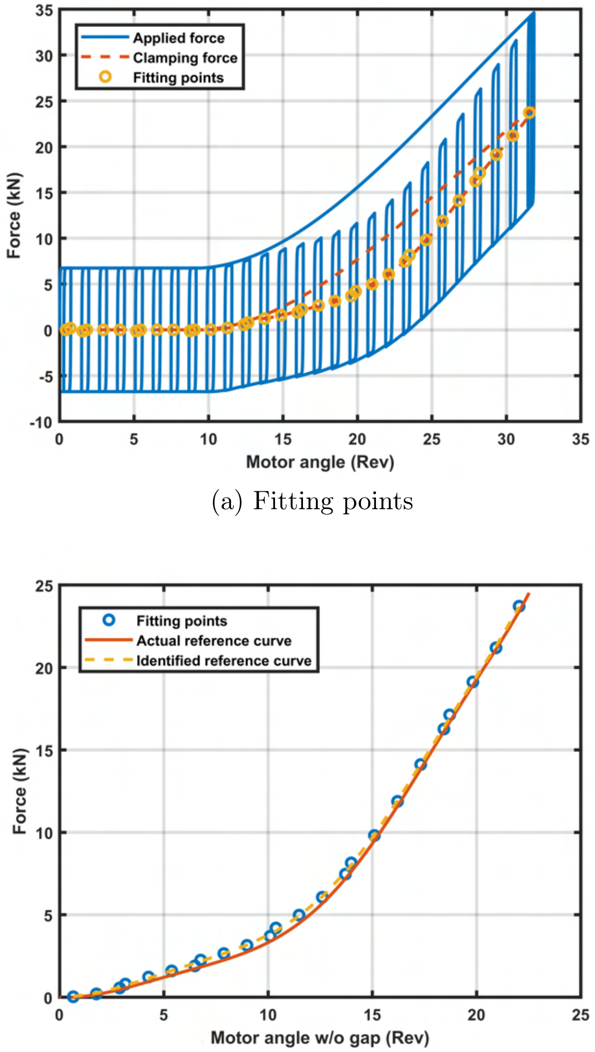
(a) Fitting points

(b) Polynomial fitting for reference curve

Figure 3.19: Identification of applying clamping force

3.4.1 Clamping Force Estimation

The polynomial functions of the reference clamping force are first identified. Figure 3.19 shows the simulation results under the input profile that is given in Fig. 3.8. In Fig. 3.19, we can see that the applied force from the motor consists of multiple small circles corresponding to the high-frequency superimposed sinusoids. The data points used for reference curve identification then can be derived from those small circles, which are indicated as circular marks. The polynomial function is identified using the least square method based on the fitting points obtained in Fig. 3.19(a). Figure 3.19(b) shows the identified reference clamping force compared to the actual one using a 3rd order polynomial function. We can see that the



(a) Fitting points

(b) Polynomial fitting for reference curve

Figure 3.20: Identification of releasing clamping force

two curves are virtually identical.

Similarly, the reference releasing clamping force is identified. Figure 3.20 shows the derivation of the fitting points, where the high-frequency sinusoid is added to the releasing operation. Figure 3.20(b) is the result of reference curve fitting using a 7th-order polynomial function. We can see that the identified reference curve is also very close to the actual one.

The two identified reference clamping forces are used in the clamping force model for esimation. The performance of this estimation will be evaluated based on the tracking control simulation.

3.4.2 Clamping Force Tracking Control

To validate the tracking performance, simulations are performed with various clamping force commands. Firstly, the stair-step reference clamping force with large and small amplitudes is used.

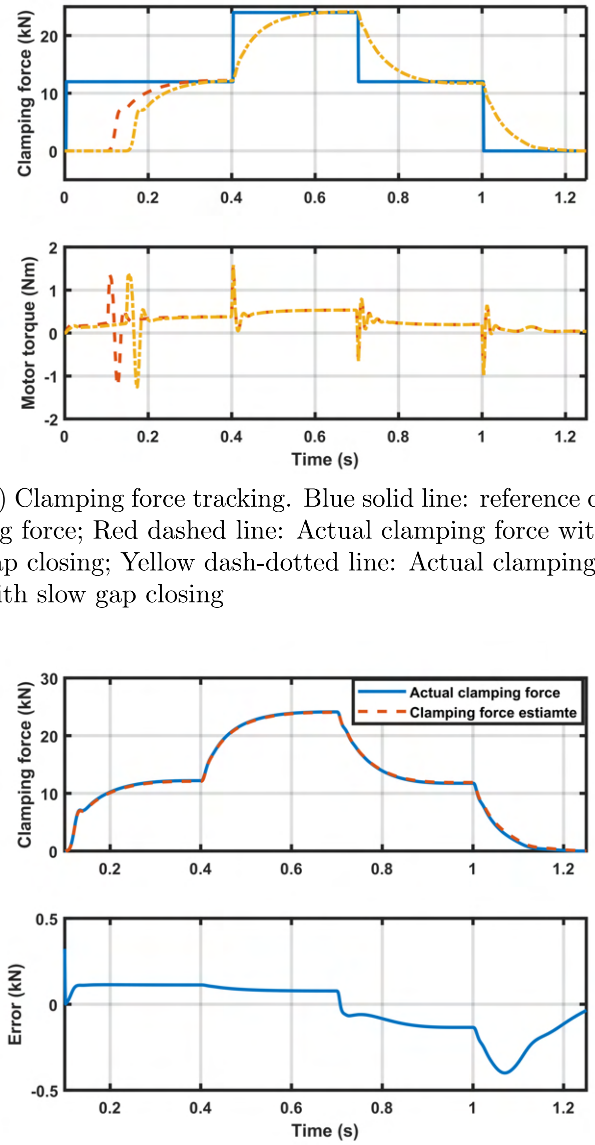
Figure 3.21 shows the simulation results where the step amplitude is 12kN, which is about half of the maximum clamping force that can be achieved by our prototype. This condition is considered an extreme situation, which could happen in reality, for example, when the driver or automatic emergency brake (AEB) system suddenly demands a large amount of brake to avoid a collision.

We can see in Fig. 3.21(a) that the step response of the clamping force has a settling time of less than 0.2s with no overshoot and very small steady-state error. The motor torque input is also below the 2Nm constraint. Notice that the motor torque is not symmetric in the applying and releasing operations due to the nonlinearity and hysteresis, while the output clamping force shows very uniform responses whether applying or releasing, which clearly indicates the nominal model regulation ability of the DOB-based controller. Figure 3.21(a) also gives the response of fast and slow gap closing, respectively, by simply adjusting the ω_G in Eq. (3.64). No matter what ω_G is used, the transition to the force control upon contact is smooth and the tracking performance is identical.

The clamping force estimation is shown in Fig. 3.21(b). The overall performance of the estimation is good, with a maximum error of less than 500N. The largest error happens during the release operation at small clamping force.

The control performance under stair-step reference with small amplitude of 2kN is also evaluated. This simulation is used to evaluate the control ability when the driver or high-level control requires a more refined brake control. As shown in Fig. 3.22(a), the clamping force has a generally consistent response, where a similar settling time and steady-state error is achieved in each step. Slightly larger steady-state errors can be found in the release operation at small clamping force. This is due to the less accurate clamping force estimation in this region, which is shown in Fig. 3.22(b).

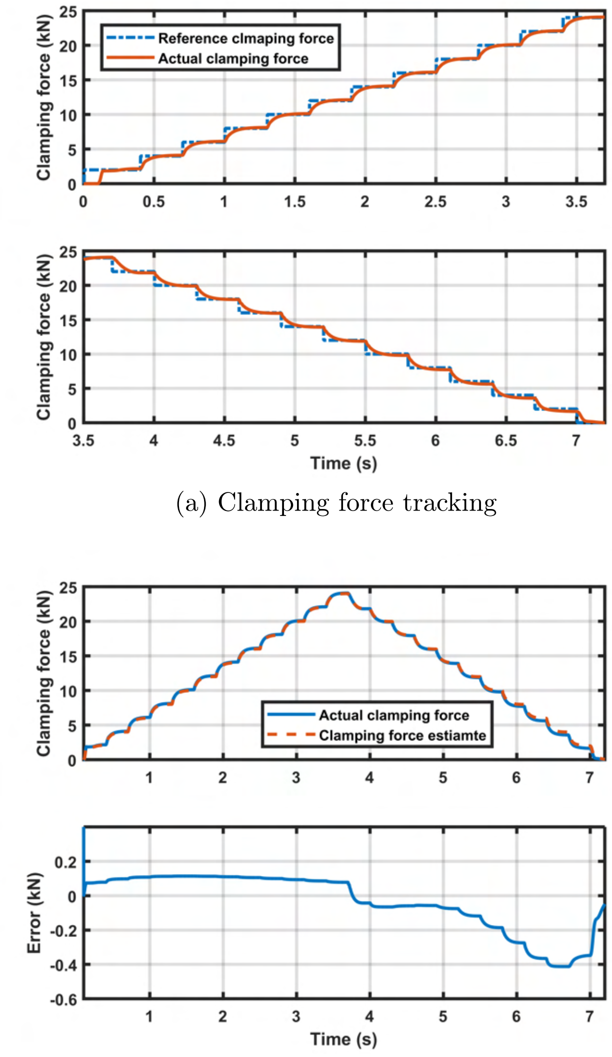
Next, simulations on sinusoidal references tracking are performed. This kind of simulation mimics the modulation of the brake force during operation of the anti-lock brake system



(a) Clamping force tracking. Blue solid line: reference clamping force; Red dashed line: Actual clamping force with fast gap closing; Yellow dash-dotted line: Actual clamping force with slow gap closing

(b) Clamping force estimation

Figure 3.21: Simulation of stair-step reference with large amplitude



(a) Clamping force tracking

(b) Clamping force estimation

Figure 3.22: Simulation of stair-step reference with small amplitude

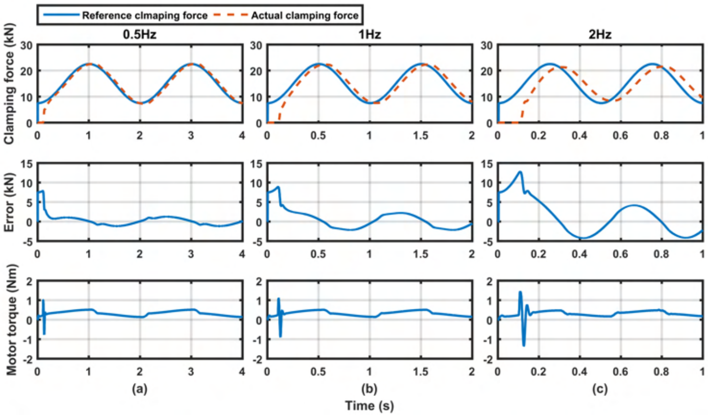


Figure 3.23: Clamping force tracking of sinusoidal reference with 50% modulation

Table 3.4: Control performance of sinusoidal reference with 50% modulation

50% modulation about 15kN	Amplitude (%)	Phase lag (°)
0.5Hz	100	9.2
1Hz	99.5	16.2
2Hz	85.5	33.1

(ABS) or electronic stability control (ESC).

Figure 3.23 shows the simulation results of the sinusoidal reference with 50% amplitude modulation about 15kN, whose frequency is respectively 0.5Hz, 1Hz, and 2Hz. We can see that the response of the clamping force is almost linear, resembling the dynamics of a low-pass filter. This is again an outcome of the DOB-based controller. Table 3.4 lists the amplitude and phase lag of the clamping force to the reference.

Figure 3.24 shows the simulation results of the sinusoidal reference with 10% amplitude modulation about 15kN. Compared to the preceding results, the response with small modulation appears less linear. This is due to the stick-slip friction, which is usually more pronounced in slow motion. Nevertheless, all the performance are reasonably good, as summarized in Table 3.5.

Table 3.5: Control performance of sinusoidal reference with 10% modulation

50% modulation about 15kN	Amplitude (%)	Phase lag (°)
0.5Hz	98.3	12.4
1Hz	96	15.5
2Hz	93	34.6

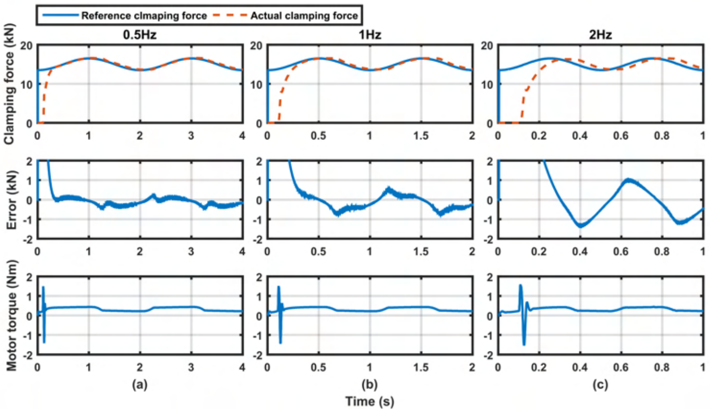


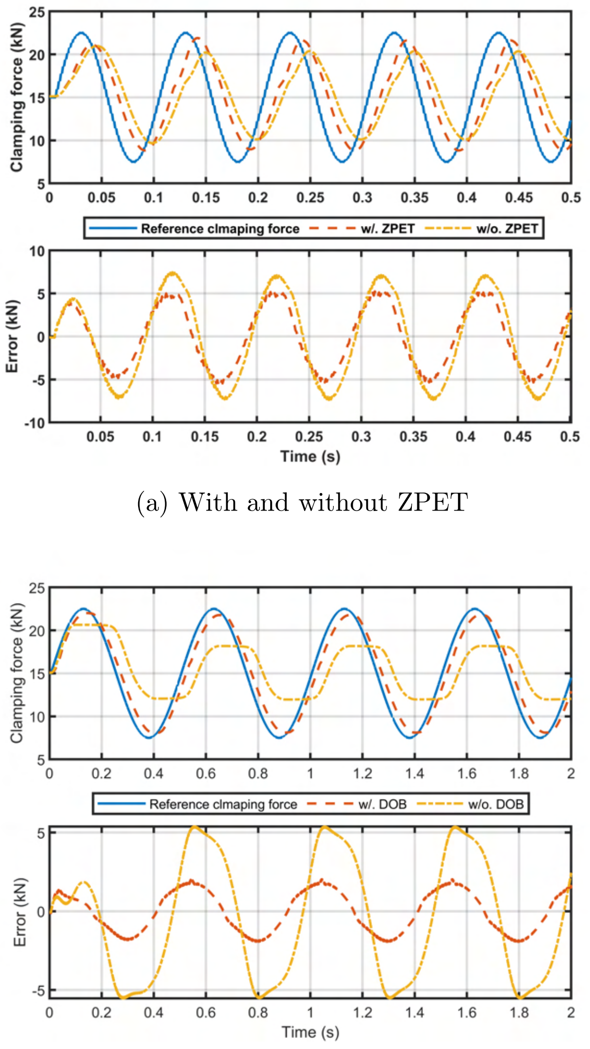
Figure 3.24: Clamping force tracking of sinusoidal reference with 10% modulation

Simulation with more demanding reference input is presented in Fig. 3.25(a), where the reference is a 10Hz sinusoidal signal. It is shown that clamping force can be tracked with reasonable amplitude decay and phase lag. Figure 3.25 also shows a comparison with cases in which either the ZPET or the disturbance observer is absent to further demonstrate the benefits of the proposed control strategy. In Fig. 3.25(a), we can see that the amplitude and phase error of tracking with ZPET is much smaller than without it. The benefit of ZPET in reducing the phase delay is clearly shown. Figure 3.25(b) shows the benefit of using the disturbance observer. Without the help of the disturbance observer, the PID controller alone cannot track the reference signal at all, since it must also take care of the disturbance by itself.

3.4.3 Robust Performance Evaluation

In this section, we are going to evaluate the robustness of the control system performance under different conditions. The reference input is 2Hz sinusoid in all cases. Firstly, the effect of time delay is shown in Fig. 3.26. In Fig. 3.26, 2ms and 4ms time delays are respectively added to the measurement feedback during simulation. The discrete-time DOB controller can accommodate time delay easily by adjusting the delay steps m of z^{-m} in Fig. 3.12[59]. By respectively setting $m = 1$ and 2, we can see that the time delay shows only minor impact on control performance. With the longer time delay, tracking error becomes slightly larger and more oscillatory at reverse points.

Secondly, the plant model parameters are varied to investigate their effect on control performance. Figure 3.27 shows the simulation results with the variations of +100% bearing friction, +50% motor time constant, and -50% pad mass, respectively. We can see from Fig.



(a) With and without ZPET

(b) With and without disturbance observer

Figure 3.25: Comparison with different controller

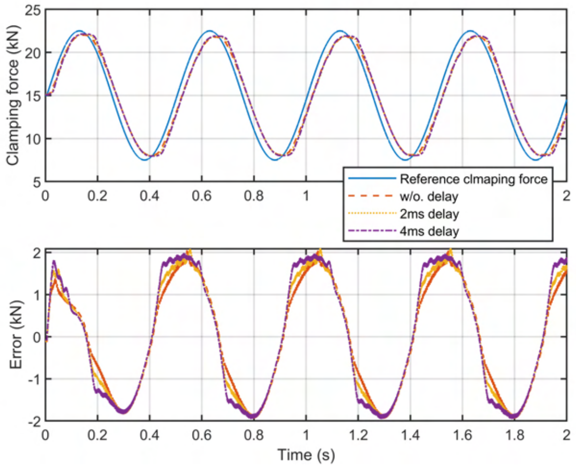
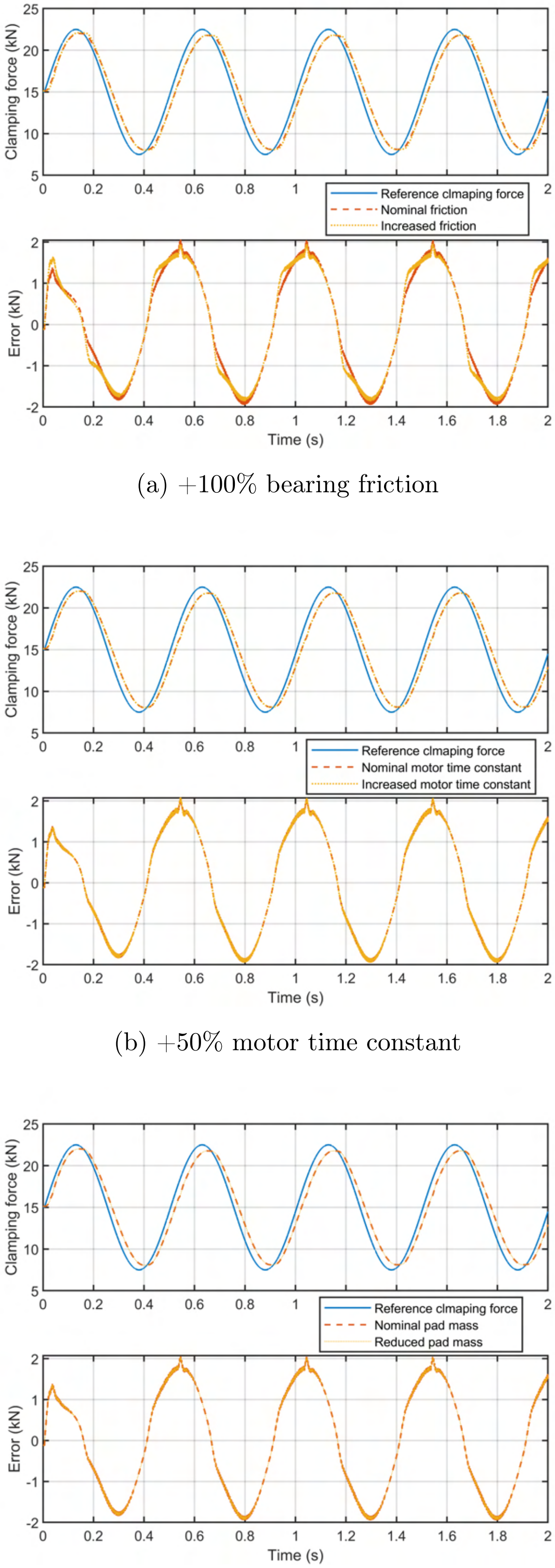


Figure 3.26: Effect of time delay

3.27 that the control performance is almost unchanged under variations in model parameters. It shows that the proposed control system provides robust performance.

Finally, noise is injected into the measurements in the simulation to investigate how the control performance is affected. Figure 3.28 shows the simulation results when motor speed measurement is corrupted by white noise where measurement delay is also included. The standard deviation of the noise is chosen as 10% of the amplitude of the clean signal. The noisy speed measurement is then integrated to be used as a position measurement. Although the tracking error shows some small oscillation and offset due to noise and delay, it shows from Fig. 3.28 that the error magnitude is maintained similarly to that of nominal case.

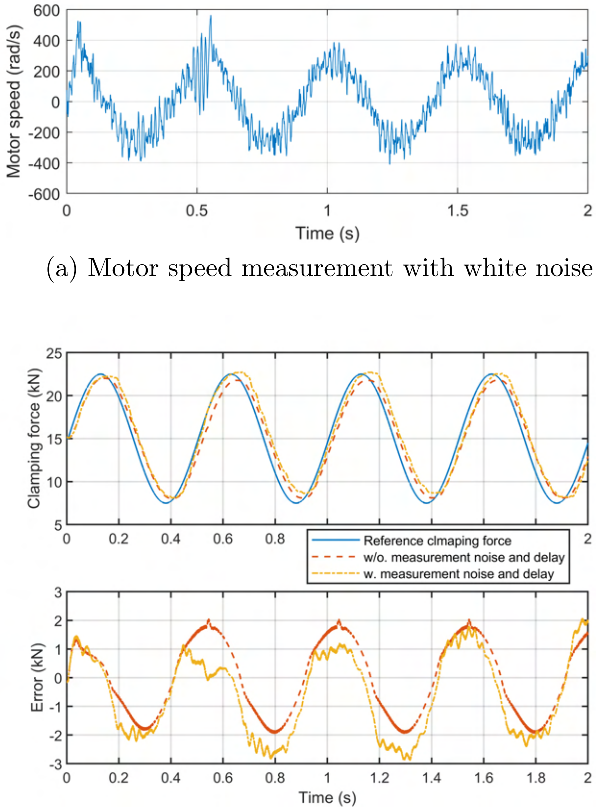


(a) +100% bearing friction

(b) +50% motor time constant

(c) -50% pad mass

Figure 3.27: Simulation with model parameters variation



(a) Motor speed measurement with white noise

(b) Control performance

Figure 3.28: Simulation with noisy measurement

3.5 Conculsions

In this chapter, the clamping force control system for EMB is developed. Firstly, a novel clamping force model is proposed that is able to capture both the nonlinearity and hysteresis of the clamping force using only two reference polynomial functions. An estimation algorithm that requires only available measurements is given to identify these reference polynomial functions. Secondly, a contact detection and gap distance estimation algorithm is developed based on the IMC-based controller and RTS smoother. Simulation results show that the proposed algorithm gives an accurate air gap distance. Thirdly, a clamping force tracking controller is designed based on a disturbance observer with PI and ZPET controllers. In addition, a unified position/force control system is proposed to eliminate the initial gap before the force control. Simulation results show that the clamping force is estimated accurately. The tracking controller also performs well in all types of references.

Part II

Optimization-based Control for High-Level Motion Planning and Trajectory Tracking

Chapter 4

NMPC based Trajectory Tracking Control for Vehicle with 4WS and 4WID

Being a key component in the autonomous vehicle, the vehicle motion control system is responsible for guiding the vehicle along the high level reference path while maintaining driving safety. We will focus on the control algorithm development for vehicle motion control system in this chapter and the following one. The motion control systems are of the higher level than the actuator control system discussed in the previous chapters. Compared to the low-level actuator controller, the design of vehicle motion control faces some additional challenges as follows:

(1) The vehicle motion control system is a nonlinear multi-input-multi-output (MIMO) system. Therefore, the control algorithm must be able to coordinate different actuators to achieve the desired outputs of the system.

(2) The control system is subject to various constraints in terms of the physical limits of actuators, safety and performance constraints on system states. So, the control algorithm should also be capable of handling the constraints in a systematic way.

When considering all these requirements, it becomes clear that model predictive control (MPC) is probably one of the most suitable control methodology that checks all the boxes. It is well known that MPC provides a very flexible framework which enables the handling of multi-variable nonlinear models with constraints. However, MPC also suffers from serious computation demand when applied to fast and nonlinear systems. Therefore, the MPC formulation and its numerical implementation must be chosen carefully with system characteristics and control goals in mind. In the following two chapters, we will present in detail how the Nonlinear-MPC (NMPC) technique can be applied to automotive motion control systems practically with good performance.

In this chapter, we will tackle the trajectory tracking problem for a vehicle equipped with 4-wheel-steering (4WS) and 4-wheel-independent-driving (4WID) system using the NMPC method.

4.1 Introduction

The application of MPC to (semi-) autonomous vehicle trajectory tracking can be found in many literature. Raffo et al. [60] presented a model predictive controller structure for solving the path-tracking problem. Two linear MPC algorithms are compared to determine the most appropriate implementation strategy. A cascade control structure is used which considers both kinematic and dynamic control. In [61], a model predictive control based path tracking algorithm was proposed to achieve accurate and smooth tracking for an autonomous vehicle. An optimal steering command was calculated by using a QP optimization method based on the integrated model. A model predictive control (MPC) approach for controlling an active front steering system in an autonomous vehicle is presented in [62]. NMPC and linear-time-varying (LTV) MPC are formulated, and their tracking performance and computational complexity are compared. The effectiveness of the proposed MPC formulation is demonstrated by simulation and experimental tests up to 21 m/s on icy roads. Similar LTV MPC frameworks are also used in [63] and [64]. Gao et al. [65] proposed a computationally efficient path-following control strategy of autonomous electric vehicles (AEVs) with yaw motion stabilization. A path-following model, a single-track vehicle model, and a magic formula tire model, is constructed. The continuation/generalized minimal residual (C/GMRES) algorithm is adopted to reduce the computational burden. The simulation results show that the proposed strategy can achieve desirable path following and vehicle stability efficacy, while greatly reducing the computational burden. In [66], an MPC based path-following controller with steering angle envelopes is proposed. Constraints regarding road sides and lateral stabilization are formulated as the steering angle envelopes to achieve better computational efficiency. Xiang et al. [67] proposed a path tracking coordinated control strategy for a Four-in-Wheel-Motor independent-drive electric vehicle. The strategy consists of upper level model predictive controller and a low-level force distributor. The NMPC is based on a 7-degree-of-freedom vehicle model with dynamic adjustment of control target weight coefficients.

The controller developed in this chapter is used by an over-actuated vehicle with 4WS and 4WID. In addition, the vehicle will likely operate near the handling limits where tire force saturates. These facts raise two key questions that have to be addressed during controller development: first, how to apply the control efforts effectively to the actuators of different dynamic characteristics, and second, how to solve the nonlinear optimal control problem

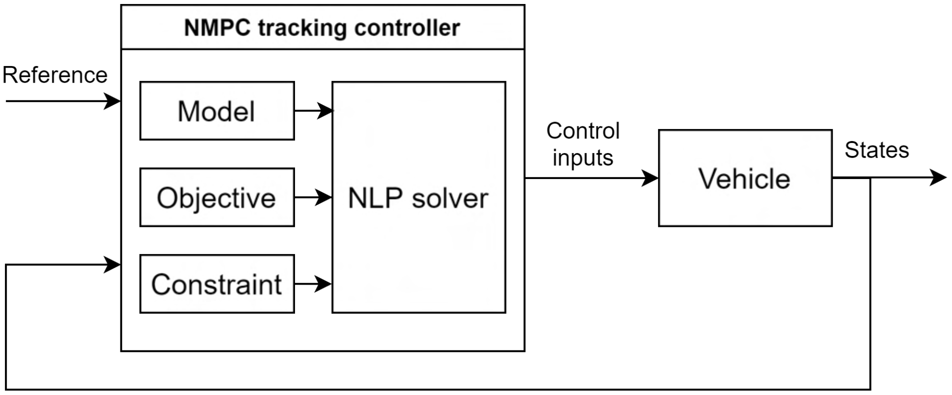


Figure 4.1: Schematic of NMPC trajectory tracking control system

efficiently in a real-time feasible manner. For the first question, the frequency shaping technique will be incorporated into the NMPC framework, which allows us to compute control inputs with consideration of actuators' dynamical characteristics. While for the latter one, we propose a fast NMPC algorithm based on real-time iteration (RTI) scheme with control parameterization. Figure 4.1 depicts the schematic of the trajectory tracking NMPC control system. The NMPC controller computes the control inputs to the vehicle based on the reference signal from the upper-level system and feedback from the vehicle, such that the vehicle's actual states follow the reference in an optimal way.

The rest of this chapter is structured as follows. In Section 4.2, the vehicle model used in the NMPC formulation is described. Section 4.3 presents the detailed formulation of the NMPC control problem, which includes the frequency shaping method and the fast NMPC algorithm. In Section 4.4, simulation results of the proposed control algorithm are discussed. Finally, Section 4.5 summarizes the proposed controller design.

4.2 Vehicle Modeling

This section describes the vehicle model that will be used for the NMPC formulation in the later sections. Vehicle models with various levels of complexity have been introduced in the literature for different purposes. Regarding the NMPC based trajectory tracking application, the model should capture the significant dynamics of the vehicle but also be as simple as possible to enable real-time feasibility. The following assumptions are made when deriving the vehicle model:

(1) The vehicle's lateral acceleration may approach the limit during emergency maneuver, e.g. obstacle avoidance.

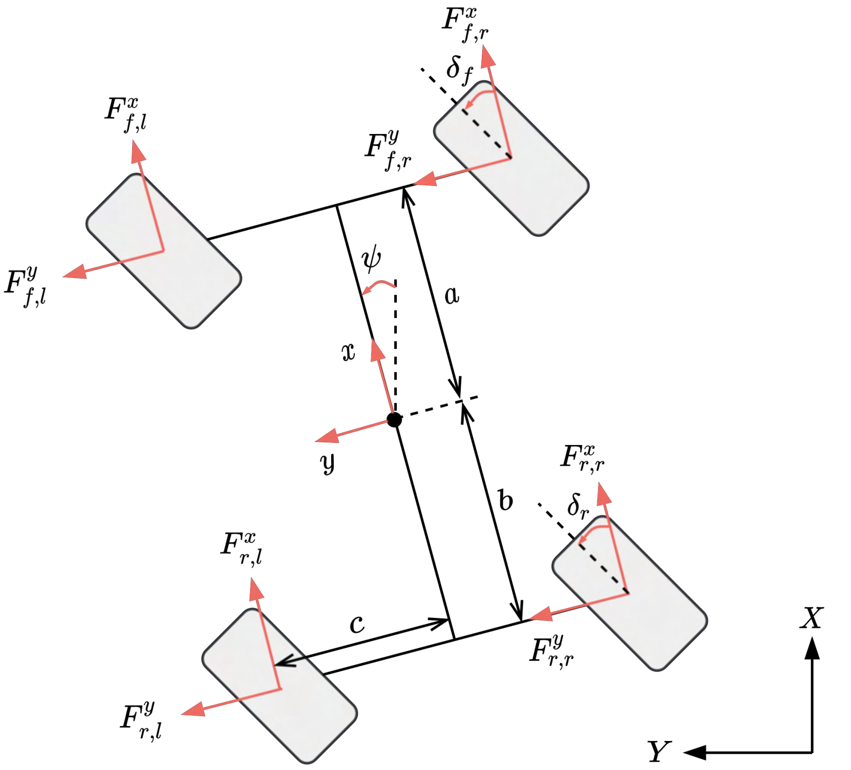


Figure 4.2: Schematic of chassis model

(2) The vehicle longitudinal speed may vary in a large range due to the reference command from a higher level motion planner.

(3) The roll, pitch, and heave motion of the vehicle are not included. But the effect of load-transfer is considered.

Based on the above assumptions, a double-track chassis model with longitudinal, lateral, and yaw degrees of freedom (DOF), incorporating load transfer and a Magic Formula tire model, is adopted in this study, which strikes a nice balance between fidelity and simplicity. The following sections describe the derivation of the vehicle model.

4.2.1 Vehicle Model Development

The vehicle dynamics are modeled as a rigid body sitting in a two-dimensional global X-Y plane, and its corresponding velocities are described in a body-fixed x-y coordinate. Figure 4.2 shows the schematic of the vehicle model. The model dynamics are given in Eqns. (4.1) - (4.3) [63].

$$m\ddot{x} = m\dot{y}\dot{\psi} + F^x_{f,l} + F^x_{f,r} + F^x_{r,l} + F^x_{r,r}, \tag{4.1}$$

$$m\ddot{y} = -m\dot{x}\dot{\psi} + F^y_{f,l} + F^y_{f,r} + F^y_{r,l} + F^y_{r,r}, \tag{4.2}$$

$$I\ddot{\psi} = a\left(F^y_{f,l} + F^y_{f,r}\right) - b\left(F^y_{r,l} + F^y_{r,r}\right) + c\left(-F^x_{f,l} + F^x_{f,r} - F^x_{r,l} + F^x_{r,r}\right), \tag{4.3}$$

where m is the vehicle mass, I is the vehicle yaw moment of inertia, a and b are respectively the distance of front and rear axles from the center of gravity, c is the half track, $\dot{x}$ and $\dot{y}$ are respectively the vehicle longitudinal and lateral speed in body-fixed coordinate, ψ is the vehicle yaw angle, $\dot{\psi}$ and $\ddot{\psi}$ are respectively the yaw rate and acceleration of the vehicle, F^x and F^y are respectively the tire forces along the longitudinal and lateral vehicle axles. We use two subscript symbols to denote four wheels, where the first one representing front/rear (f/r) axle, and the second one left/right (l/r) wheel.

The vehicle's position in global frame is expressed as

$$\dot{X} = \dot{x}\cos\psi - \dot{y}\sin\psi, \tag{4.4}$$

$$\dot{Y} = \dot{x}\sin\psi + \dot{y}\cos\psi. \tag{4.5}$$

The relationship between tire forces along the vehicle axes and tire forces along the wheel axes are

$$F^x = F_l\cos\delta - F_c\sin\delta, \tag{4.6}$$

$$F^y = F_l\sin\delta + F_c\cos\delta, \tag{4.7}$$

where F_l and F_c are the tire longitudinal and cornering forces along wheel axes, respectively, δ is the steering angle.

4.2.2 Tire Slip and Force

When steering the wheel, lateral slip angle α develops. It is the ratio of the wheel's velocities along the lateral and longitudinal axes, resolved in the wheel's local coordinates, which is illustrated in Fig. 4.3 and can be written as

$$\tan\alpha = -\frac{v^c}{v^l}, \tag{4.8}$$

where v^c and v^l are the cornering and longitudinal velocities, respectively, which are expressed as

$$v^l = v^y\sin\delta + v^x\cos\delta, \tag{4.9}$$

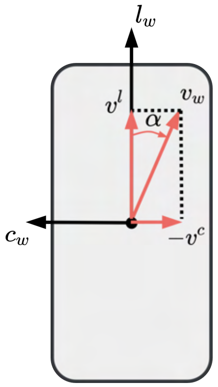


Figure 4.3: Wheel local coordinate system and slip angle

$$v^c = v^y \cos\delta - v^x \sin\delta, \tag{4.10}$$

where v^x and v^y are the wheel velocities along vehicle body-fixed axes, which are given as

$$v^x_{f/r,l} = \dot{x} - c\dot{\psi}, \tag{4.11}$$

$$v^x_{f/r,r} = \dot{x} + c\dot{\psi}, \tag{4.12}$$

and

$$v^y_{f,l/r} = \dot{y} + a\dot{\psi}, \tag{4.13}$$

$$v^y_{r,l/r} = \dot{y} - b\dot{\psi}. \tag{4.14}$$

When torque is applied to the wheel, longitudinal slip ratio λ is generated, which is defined as

$$\lambda := \frac{R_w\omega - v^l}{v^l} \in [-1, \infty) \tag{4.15}$$

where R_w is the wheel effective radius, and ω is the wheel rotational velocity. Fig 4.4 shows the schematic of wheel longitudinal slip.

While it is possible to treat λ as a system state by including wheel rotational dynamics, we will directly regard λ as the control input to the vehicle model that will be determined by the NMPC controller. This setup is valid if we assume there is a low-level controller responsible for tracking the slip ratio reference passed by the NMPC controller. Such wheel slip controllers have been studied for decades by many researchers and can be found more and more common in the modern vehicles [68, 69]. The motivation for doing so is threefold:

(1) The wheel dynamics are usually orders of magnitude faster than the chassis dynamics.

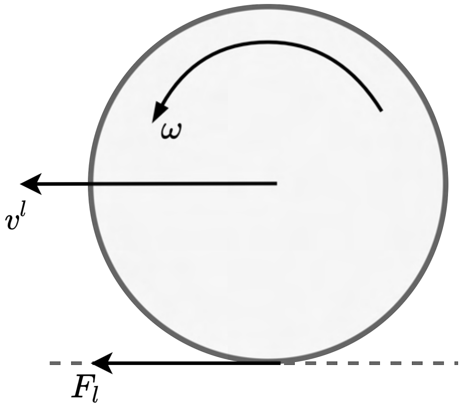


Figure 4.4: Wheel longitudinal slip schematic

If included in the vehicle model, it will render the underlying dynamical system stiff, which hinders the numerical solving of NMPC.

(2) The wheel dynamics increases the number of states by four, which again makes numerical computation more difficult.

(3) The wheel dynamics are often subject to many uncertainties, such as road friction variation, mechanical friction, and external disturbance. Dedicated control algorithms have been proposed in many literature to address these challenges and achieve good performance. Therefore, it is wise to let those controllers do the job rather than leave all the burden on the NMPC controller.

However, caution must be taken when we formulate the NMPC using this configuration. Since the slip ratio becomes the control input now, we must carefully choose the weighting parameters and constraints such that its profile will not exceed the capability of the physical actuator.

The vehicle's motion is almost exclusively determined by the tire forces. Therefore, it is very important to have a tire model that realistically reflects the tire force. In this study, we choose the Pacejka's Magic Formula model, given by [70]

$$F_0\left(m\right) = \mu F^z \sin\left(C \arctan\left(Bm - E\left(Bm - \arctan\left(Bm\right)\right)\right)\right), \tag{4.16}$$

where m is either λ or α depending on the querying tire force, B, C, E are constant coefficients obtained from experimental data, μ is the road friction coefficient, and F^z is tire load. Figure 4.5 shows the tire force form Eq. (4.16) for different road surface conditions under the nominal

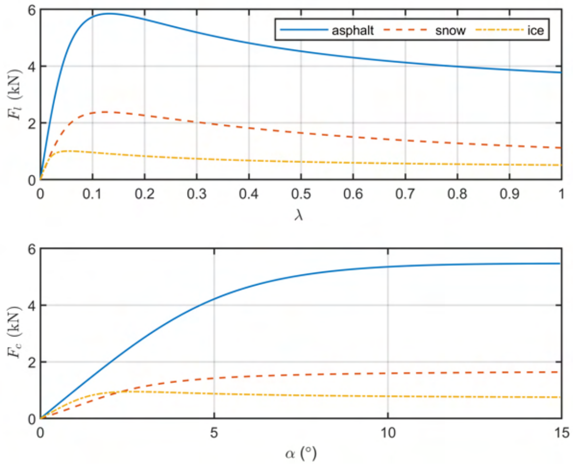


Figure 4.5: Tire force form Magic Formula Model with $F^z = 5718N$

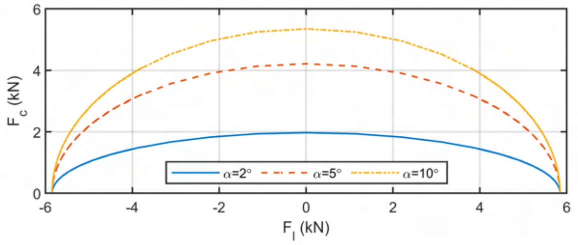


Figure 4.6: Tire force form under combined slip condition with $F^z = 5718N$

tire load when the vehicle is at rest. In Fig. 4.5, the $\mu = 1$, 0.4, and 0.17 receptively for the asphalt, snowy and icy surface, which are typical values based on the experimental results [71].

Equation (4.16) only describes the tire force under pure slip condition, i.e. either λ or α is zero. Since the NMPC controller is supposed to simultaneously perform steering and braking/driving, it is necessary to model the tire force in a combined slip condition. In this study, the friction-ellipse method [72] is used. This method computes the cornering force F_c given the longitudinal force F_l as

$$F_c = F_{0,c}\left(\alpha\right)\sqrt{1-\left(\frac{F_l}{\mu_l F^z}\right)^2}, \tag{4.17}$$

where F_l is computed by Eq. (4.16). Figure 4.6 shows the tire forces under combined slip conditions using Eq. (4.17).

4.2.3 Load Transfer

We can see from Eq. (4.16) that the tire force is affected by tire normal load F^z. During at-the-limit maneuvers such as obstacle avoidance, the vehicle will experience very high lateral acceleration, thus inducing significant load transfer. In order to include this effect while avoiding modeling complex suspension dynamics, we will make the following assumption:

The transient load shift effect due to suspension dynamics is not modeled. The load transfer is calculated algebraically by assuming that the normal forces produce moments about the C.G. that are in static equilibrium with the C.G. moments due to the forces at the tire contact patches.

With the above assumption, the steady-state load transfer model is expressed straightforwardly as

$$2\Delta\bar{F}^z_x = \left(F^x_{f,l} + F^x_{f,r} + F^x_{r,l} + F^x_{r,r}\right)\frac{h}{a+b}, \tag{4.18}$$

$$2\Delta\bar{F}^z_y = \left(F^y_{f,l} + F^y_{f,r} + F^y_{r,l} + F^y_{r,r}\right)\frac{h}{2c}, \tag{4.19}$$

where h is the height between the vehicle's C.G. and ground. Note Eqns. (4.18) and (4.19) create algebraic loops between load transfer $\Delta\bar{F}^z_{x/y}$ and tire force $F_{l/c}$ as they depend on each other mutually. In principle, this is not a issue to NMPC formulation if we consider the system as differential-algebraic equations (DAEs) [73]. However, doing so will generally make the optimization more complicated to solve. Therefore, we will relax the algebraic loops by introducing first-order lags for load transfer:

$$\tau_L \Delta\dot{F}^z_{x/y} + \Delta F^z_{x/y} = \Delta\bar{F}^z_{x/y}, \tag{4.20}$$

where τ_L is the time constant.

Finally, the tire normal load is given by

$$F^z_{f,l} = \bar{F}^z_{f,l} - \Delta F^z_x - \Delta F^z_y, \tag{4.21}$$

$$F^z_{f,r} = \bar{F}^z_{f,r} - \Delta F^z_x + \Delta F^z_y, \tag{4.22}$$

$$F^z_{r,l} = \bar{F}^z_{r,l} + \Delta F^z_x - \Delta F^z_y, \tag{4.23}$$

$$F^z_{r,r} = \bar{F}^z_{r,r} + \Delta F^z_x + \Delta F^z_y, \tag{4.24}$$

where $\bar{F}_*^z$ is the nominal load when vehicle is at rest.

4.3 Nonlinear Model Predictive Control of Trajectory Tracking

The objective of NMPC is to compute in real-time the optimal control inputs, minimizing the discrepancy between the actual vehicle trajectory and the reference. While doing this, the controller must also take into account the physical limitations of actuators, such as amplitude, rate, and bandwidth. In this section, the objective function is formulated with system dynamics and control inputs as equality and inequality constraints. We will also discuss how to solve the formulated optimal control problem (OCP) using real-time iteration scheme (RTI) and control parameterization, and how to incorporate actuator bandwidth limitation using frequency shaping.

4.3.1 Formulation of the Optimal Control Problem

The vehicle model presented in the previous section is written as

$$\dot{\xi}\left(t\right) = f\left(\xi\left(t\right), u\left(t\right)\right), \tag{4.25}$$

$$\eta\left(t\right) = C\xi\left(t\right), \tag{4.26}$$

where $f\left(\xi\left(t\right), u\left(t\right)\right)$ are given by Eqns. (4.1)-(4.3), (4.4)-(4.5), and (4.20), $\eta = \begin{bmatrix} \dot{x} & \psi & Y & X \end{bmatrix}^T$ is the control output, $\xi = \begin{bmatrix} \dot{y} & \dot{x} & \psi & \dot{\psi} & Y & X & \Delta F_x^z & \Delta F_y^z \end{bmatrix}^T$ is the state vector, $C = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$ is the output matrix, and $u = \begin{bmatrix} \delta_f & \delta_r & s_{ff} & s_{fl} & s_{rl} & s_{rr} \end{bmatrix}^T$ is the control input.

The OCP is then formulated over the time horizon $t \in [t_0, t_f]$ as

$$\begin{aligned}
\underset{\xi(t),u(t)}{\text{minimize}} \quad & \frac{1}{2}\int_{t_0}^{t_f} \|\eta(t) - \eta_r(t)\|_Q^2 + \|u(t)\|_R^2 \, dt + \frac{1}{2}\|\eta(t_f) - \eta_r(t_f)\|_P^2 && (4.27) \\
\text{subject to} \quad & \xi(t_0) = \xi_0 && (4.28) \\
& \dot{\xi}(t) = f(\xi(t), u(t)) && (4.29) \\
& \eta(t) = C\xi(t) && (4.30) \\
& u_L \le u(t) \le u_U && (4.31) \\
& \dot{u}_L \le \dot{u}(t) \le \dot{u}_U. && (4.32)
\end{aligned}$$

Equation (4.28) denotes the initial condition constraint, where $\xi_0 \in \mathbb{R}^8$ is the initial state vector. We assume ξ_0 is available either through measurement or estimation. Equation (4.27) is the objective function that penalizes the reference tracking deviation as well as the control effort, where $\|\cdot\|$ denotes the Euclidean norm and its subscript represents the weighting matrix. $\eta_r \in \mathbb{R}^4$ is the reference control output that is given by the upper-level controller. Equations (4.31) and (4.32) are the control amplitude and rate constraints, respectively. The subscriptions L and U denote the lower and upper bounds, respectively. In our control architecture, it is the upper-level controller's responsibility to generate a dynamically feasible trajectory while avoiding obstacles. Therefore, the sole goal of the tracking controller is to follow that reference state trajectory, and only simple box constraints on the control inputs are considered here. Though the algorithms in this chapter can be easily extended to cases where complex path constraints are presented.

The continuous-time OCP defined in Eqns. (4.28)-(4.32) is solved repeatedly online at each time step k. The optimal control trajectory during $t \in [t_0, t_0 + T_s]$ is then applied to the system, where T_s is the sampling period of the NMPC. Due to the nonlinearity and nonconvexity of the OCP, its solution can not be found trivially. We introduce numerical methods based on RTI and control parameterization in the following to solve this problem.

4.3.2 Objective Function with Frequency Shaping

In the OCP (4.28)-(4.32), the control input is constrained by its amplitude (4.31) and rate (4.32). This is the common way to incorporate the physical limits of the actuators in the MPC framework. However, the time-domain description might not be the best choice in some cases. In the case of vehicle chassis control, actuators' characteristics are better described in the frequency-domain since many of them, such as steering and braking, are electromechanical systems which are often designed to work at certain frequency ranges. Furthermore, the steering or slip control themselves might also possess different actuator dynamics for the front and rear axles. To effectively incorporating different frequency characteristics of the actuators into NMPC framework, a frequency shaping mechanism is adopted in the NMPC formulation.

The frequency shaping technique was first introduced in the context of LQG control [74] and later applied to automotive control in [75]. Recently, it has been used in the MPC scheme in the field of robotic control [76]. The basic idea of frequency shaping is based on Parseval's theorem, which states the following relationship between the time and frequency domain descriptions of a signal

$$\int_0^\infty x^T(t)\,x(t)\;dt = \frac{1}{2\pi}\int_0^\infty X^T(-j\omega)\,X(j\omega)\;d\omega, \tag{4.33}$$

where $x(t)$ is a square integrable signal defined on $t \in [0,\infty)$, and $X(j\omega) = \mathcal{F}\{x(t)\}$ represents its continuous Fourier transform.

Now we consider an infinite horizon objective function

$$J = \frac{1}{2}\int_0^\infty \|\eta(t) - \eta_r(t)\|_Q^2 + \|u(t)\|_R^2\,dt. \tag{4.34}$$

Equation (4.34) can be convert into frequency domain using Eq. (4.33)

$$J = \frac{1}{4\pi}\int_0^\infty H^T(-j\omega)\,QH(j\omega) + U^T(-j\omega)\,RU(j\omega)\;d\omega, \tag{4.35}$$

where $H(j\omega) = \mathcal{F}\{\eta(t) - \eta_r(t)\}$, and $U(j\omega) = \mathcal{F}\{u(t)\}$. We can see that the objective function of conventional MPC formulation corresponds to constant weighting over the whole frequency range. In order to enable frequency shaping on the control effort, the constant weighting matrix R is replaced by the frequency-dependent matrix $\widetilde{R}(j\omega)$. The modified objective function now becomes

$$J_f = \frac{1}{4\pi}\int_0^\infty H^T(-j\omega)\,QH(j\omega) + U^T(-j\omega)\,R_f(j\omega)\,U(j\omega)\;d\omega, \tag{4.36}$$

with $\widetilde{R}(\omega)$ defined as

$$R_f(j\omega) := \begin{bmatrix} R_{f,1}^T(-j\omega) & & \\ & \ddots & \\ & & R_{f,n_u}^T(-j\omega) \end{bmatrix} R \begin{bmatrix} R_{f,1}(j\omega) & & \\ & \ddots & \\ & & R_{f,n_u}(j\omega) \end{bmatrix},$$

where $R_{f,1}(j\omega), \ldots, R_{f,n_u}(j\omega)$ are the weightings that shape the frequency response of each control input separately. Apply Parseval's theorem again to Eq. (4.34), we can transform the frequency domain objective function back into time domain

$$J_f = \frac{1}{2} \int_0^{\infty} \|\eta(t) - \eta_r(t)\|_Q^2 + \|u_f(t)\|_R^2 \, dt, \tag{4.37}$$

where $u_f(t)$ is the filtered control input given as

$$u_f(t) = \mathcal{F}^{-1} \left\{ \begin{bmatrix} R_{f,1}(j\omega) & & \\ & \ddots & \\ & & R_{f,n_u}(j\omega) \end{bmatrix} U(j\omega) \right\}. \tag{4.38}$$

We can see that instead of penalizing the control inputs directly, the frequency-shaped objective function takes into account the filtered ones. Thus, by adjusting the diagonal elements of the weighting function R_f, we are able to shape the frequency response of each control input independently.

To accommodate the new objective function into the NMPC formulation, we first approximate the infinite horizon function to a finite one

$$J_f \cong \frac{1}{2} \int_{t_0}^{t_f} \|\eta(t) - \eta_r(t)\|_Q^2 + \|u_f(t)\|_R^2 \, dt + \frac{1}{2} \|\eta(t_f) - \eta_r(t_f)\|_P^2. \tag{4.39}$$

Here the cost during $t \in [t_f, \infty)$ is lumped into a terminal term $\frac{1}{2}\|\eta(t_f) - \eta_r(t_f)\|_P^2$. In practice, this approximation will have little impact on control performance if the weighting P is chosen properly. It can even be neglected if the prediction horizon is long enough. In this dissertation, we use $P = kQ$, where k is a scaling factor to ensure the stability and tuned by trial and error based on the simulation results. We then augment the prediction model with the following dynamics

$$\dot{z}_R = A_R z_R + B_R u, \tag{4.40}$$

$$u_f = C_R z_R + D_R u, \tag{4.41}$$

which is the state-space representation of the frequency-dependent weighting R_f. The dimension of the state z_R is the sum of the orders of $R_{f,1,\ldots n_u}(j\omega)$. The matrices A_R, B_R, C_R, and D_R are obtained by the Matlab function tf2ss.

In summary, the frequency-shaped NMPC is formulated as

$$\underset{\xi(t),u(t)}{\text{minimize}} \quad \frac{1}{2}\int_{t_0}^{t_f} \|\eta_a(t)-\eta_{a,r}(t)\|_{Q_f}^2 + \|u(t)\|_{R_f}^2 + 2\xi_a^T(t)\,S u(t)\ dt + \frac{1}{2}\|\eta(t_f)-\eta_r(t_f)\|_P^2 \tag{4.42}$$

$$\text{subject to} \quad \xi_a(t_0) = \xi_{a,0} \tag{4.43}$$

$$\dot{\xi}_a(t) = f_a(\xi_a(t), u(t)) \tag{4.44}$$

$$\eta_a(t) = C_a \xi_a(t) \tag{4.45}$$

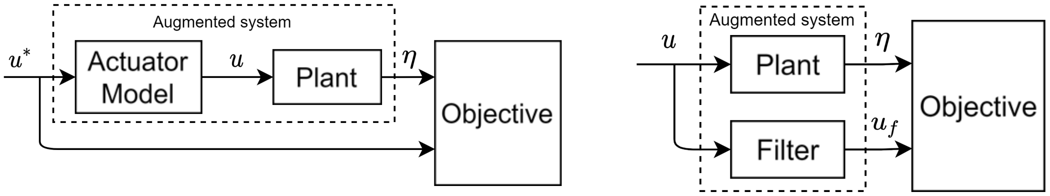
$$u_L \le u(t) \le u_U \tag{4.46}$$

$$\dot{u}_L \le \dot{u}(t) \le \dot{u}_U, \tag{4.47}$$

where $Q_f := \begin{bmatrix} Q & \\ & C_R^T R C_R \end{bmatrix}$, $R_f := D_R^T R D_R$ and $S := \begin{bmatrix} 0 \\ C_R^T R D_R \end{bmatrix}$ are the augmented weighting matrices, $\xi_a := \begin{bmatrix} \xi \\ z_R \end{bmatrix}$ is the augmented states, $\eta_a := \begin{bmatrix} \eta \\ z_R \end{bmatrix}$ is the augmented outputs, $f_a(\xi_a(t), u(t)) := \begin{bmatrix} f(\xi(t), u(t)) \\ A_R z_R + B_R u \end{bmatrix}$ is the augmented dynamics, and $C_a := \begin{bmatrix} C & I \end{bmatrix}$ is the augmented output matrix.

We can see that the frequency-shaped NMPC includes an additional cross term in the objective function as well as augmented filter states and dynamics in the constraints. These impose no fundamental difficulty in solving the NMPC. They do introduce small computation overhead because of the increase of problem dimension. However, since the frequency shaping filter is linear time-invariant and decoupled from the nonlinear plant, the additional constraint relating to it can be fulfilled easily in the optimization routine with little effort.

Finally, we want to stress some differences between the frequency shaping technique and actuator model embedding method. The actuator model embedding is a very common approach to incorporate the actuator dynamics into MPC formulation. The basic idea is to augment the original plant with actuator models. In this way, the input to the plant is the output of the actuator model and the input command to the actuator model is treated as the decision variable of OCP. The difference between these two schemes is depicted in Fig. 4.7. In actuator embedding method, the actuator model is connected to the plant in series. While for frequency shaping, the filter and plant are positioned in parallel. This decoupled feature not only benefits numerical computation, but also provides more freedom in controller tuning. Unlike the actuator model, the shaping filter can be designed freely according to the control



(a) Actuator model embedding

(b) Frequency shaping

Figure 4.7: Comparison between actuator embedding and frequency shaping methods

objective, regardless of the actual physical characteristics of the actuators. Moreover, the frequency shaping method can be seamlessly applied to state weighting, while actuator model embedding only works on control input.

Together with the conventional time domain constraint such as amplitude and rate, the frequency shaped objective function allows us to more comprehensively represent the dynamics of low-level actuators.

4.3.3 Fast NMPC based on RTI and Control Parameterization

Numerical methods for solving the continuous-time OCP can be generally classified into three categories: (1) dynamic programming; (2) indirect approaches; and (3) direct approaches [77]. Although being able to solve the OCP up to global optimality, dynamic programming suffers seriously from the so-called "curse of dimensionality", meaning the computational complexity increases exponentially in the dimension of the system states [78]. Therefore, it is rarely used in practice for real-time nonlinear systems. The more commonly used methods for solving the continuous-time OCP are the indirect and direct approaches. The former one involves solving a boundary value problem that stems from the first-order necessary conditions of optimality. Modern BVP solvers have been proposed to treat this problem effectively [79]. However, it is still generally difficult to handle inequality constraints and singular arcs when using this method. On the other hand, thanks to the great advances in nonlinear programming (NLP) algorithms in recent years, direct approaches have become more and more popular. In the direct methods, the continuous-time system is transcribed into discrete-time system using some numerical method, which entails a finite-dimensional NLP. Then, the resulted NLP is solved by a numerical solver, which is able to handle inequality path constraints nicely.

There exist two distinct types of direct approaches, namely the sequential method and the simultaneous method. The sequential method, also known as single shooting, uses control inputs as decision variables together with a numerical integrator to simulate the whole state trajectory. NLP solver is then employed to determine the optimal control input sequence. Since the control inputs are the only decision variables, the NLP form sequential method is usually small scale. However, the drawback of the sequential method is that the nonlinearity of the dynamics also propagates via the simulation process, which makes the solution very sensitive to the initial guess. The simultaneous method provides a remedy to this issue. Simultaneous method itself consists of two approaches, i.e. direct multiple shooting and direct collocation [77]. Direct multiple shooting divides the whole time horizon into multiple short segments, and simulates the system separately inside each segment using an embedded integrator. The continuity of dynamics between adjacent segments is enforced by equality constraints on the system states. In this way, the nonlinearity is confined to each segment, rendering the NLP easy to treat. Direct collocation takes a step further, discretizing the control and state variables completely. The state equations are often discretized based on an orthogonal collocation method using polynomial approximations. Direct collocation is fully simultaneous and requires no nested integrator. Both direct methods entail large-scale NLPs but have the advantages of handling unstable systems. The choice between these two methods should be made by considering the problem's nature as well as the solver available. In this chapter, direct multiple shooting is adopted for solving the tracking NMPC. In the next chapter, we will discuss the application of direct collocation to the motion planing NMPC.

Using direct multiple shooting, the OCP of Eqns. (4.42)-(4.47) is transcribed as

$$\underset{\substack{s_0, s_1, \ldots, s_N, \\ u_0, u_1, \ldots, u_{N-1}}}{\text{minimize}} \quad \frac{T_s}{2} \sum_{i=0}^{N-1} \left(\|\eta_i - \eta_{r,i}\|_Q^2 + \|u_i\|_R^2 + 2 s_i^T S u_i \right) + \frac{1}{2} \|\eta_N - \eta_{r,N}\|_P^2 \tag{4.48}$$

$$\text{subject to} \quad \xi_0 - s_0 = 0 \tag{4.49}$$

$$s_{i+1} - F_i \left(s_i, u_i\right) = 0, \; \forall i \in \{0, \ldots, N-1\} \tag{4.50}$$

$$\eta_i = C s_i \tag{4.51}$$

$$u_L \leq u_i \leq u_U, \; \forall i \in \{0, \ldots, N-1\} \tag{4.52}$$

$$\Delta u_L \leq u_i - u_{i-1} \leq \Delta u_U, \; \forall i \in \{1, \ldots, N-1\} \, . \tag{4.53}$$

For the clarity of presentation, we have omitted the subscription a and f in the states and weighting matrix. Equations (4.48)-(4.53) is a finite-dimensional NLP whose decision variables are system states s_i as well as control inputs u_i. The time horizon $[t_0, t_f]$ is divided into $N = \frac{t_f}{T_s}$ segments of equal length defined as $[t_i, t_{i+1}]$ for $i = 0, \ldots, N-1$. The control inputs u_i is piecewise constant discretization of its continuous counterpart over these segments, i.e. $u\left(t\right) = u_i$ for $t \in [t_i, t_{i+1}]$. Equation (4.48) is discretized objective function, which is simply the Riemann sum of the original integration. Equation (4.49) is the initial value constraint. In principle, this constraint can be easily eliminated from the NLP and results in one less decision variable. However, it is kept here explicitly to achieve the feature of the tangential predictor, also called in the literature, the initial value embedding [80]. Equation (4.53) is the constraint on the control input change. Equation (4.50) is the continuity constraint of system dynamics, where $F_i\left(s_i, u_i\right)$ is a simulator based on some numerical integration method which outputs the states at the next time step. Depending on the system considered, many numerical methods can be used, such as matrix exponential, Runge-Kutta (RK) methods, and multistep methods. Moreover, multiple integration steps can be implemented in one discrete interval to achieve higher accuracy if necessary. In our case, since the system dynamics consists of both the nonlinear plant and linear filters, we use two different integration schemes. For the nonlinear plant model, we use the popular 4th-order RK scheme to construct a simulator, which is described by Algorithm 4.1. For the linear filter, the discrete-time dynamics is obtained via matrix exponential.

Algorithm 4.1 Computing $F_i\left(s_i, u_i\right)$ using 4th-order RK scheme

```
Input: $s_i$, $u_i$, $N_i$, $T_s$
System dynamics: $f$
Output: $s^+ := F_i(s_i, u_i)$
$s^+ \leftarrow s_i$
$h \leftarrow \frac{T_s}{N_i}$
for $j = 1, \dots, N_i$ do
    $k_1 \leftarrow f(s^+, u_i)$
    $k_2 \leftarrow f(s^+ + \frac{h}{2}k_1, u_i)$
    $k_3 \leftarrow f(s^+ + \frac{h}{2}k_2, u_i)$
    $k_4 \leftarrow f(s^+ + hk_3, u_i)$
    $s^+ \leftarrow s^+ + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$
end for
```

$$z_{R,k+1} = A_d z_{R,k} + B_d u_k, \tag{4.54}$$

where

$$A_d = e^{A_R T_s}, \tag{4.55}$$

$$B_d = \int_0^{T_s} e^{A_R (T_s - \tau)} B_R \, d\tau. \tag{4.56}$$

In theory, the NLP of (4.48)-(4.53) can be solved by any general-purpose NLP slover, e,g. Matlab's fmincon. From a practical point of view, however, the numerical algorithm for solving the NLPs of NMPC must be tailored for real-time applications. The development of fast NMPC has been a hot topic in the research community, and different methods have been proposed. There are several possible ways to classify these methods. Here, we divide them into three categories: the first one relays on specialized optimization solvers exploiting the problem structure directly, the second one reformulates the problem's structure, and the third one solves only sub-optimal solutions.

The examples that belong to the first category include HPIPM, OOQP, FORCES PRO, and so on [81–83]. These tailored solvers exploit the particular structure of the MPC problems, usually by the interior point method, to achieve excellent performance. The development of numerical solvers is beyond the scope of this study, and we will not discuss further on this topic.

The second category of methods manipulates the formulations of NLP to reduce the computational complexity. They are useful when sophisticated solvers are not available due to hardware limitations or cost. Some common approaches belonging to the second category can be found in the context of linear MPC control. For example, the control horizon can be chosen shorter than the prediction horizon by fixing the last control input until the end of the prediction horizon [84]. In this way, the number of decision variables is reduced. In the same spirit, the move blocking technique is also used to reduce the problem dimension by constraining the control input to be constant over several discretization steps [85]. Another similar method is so-called nonuniform grid scheme, where the system is discretized with varying step sizes [86]. Typically, the discretization grid is dense at the beginning and becomes more sparse towards the end of the prediction horizon. By this way, fewer decision variables is required while remaining a relatively long prediction horizon.

While the above methods are based on the idea of reducing the optimization scale and hoping it can be solved in one sample period, the third kind of method takes a completely different route. These methods are called time-distributed optimization (TDO) for NMPC [87]. Instead of seeking a fully converged solution to the NLP in each time step, an approximated solution is obtained by using only limited iterations of the optimization routine, and this solution is consistently improved over the whole operation duration of the system. In such a way, a much faster sampling time can be used since the controller doesn't need to wait for the NLP to converge, and the feedback can be updated more frequently to cope with model mismatch or external disturbance. Several methods have been proposed based on TDO [88–90]. The most notable one among them is the real-time iteration scheme [91].

In this study, a fast NMPC algorithm is developed incorporating the distinctive features of the second and third types of approaches. In the proposed algorithm, the RTI with

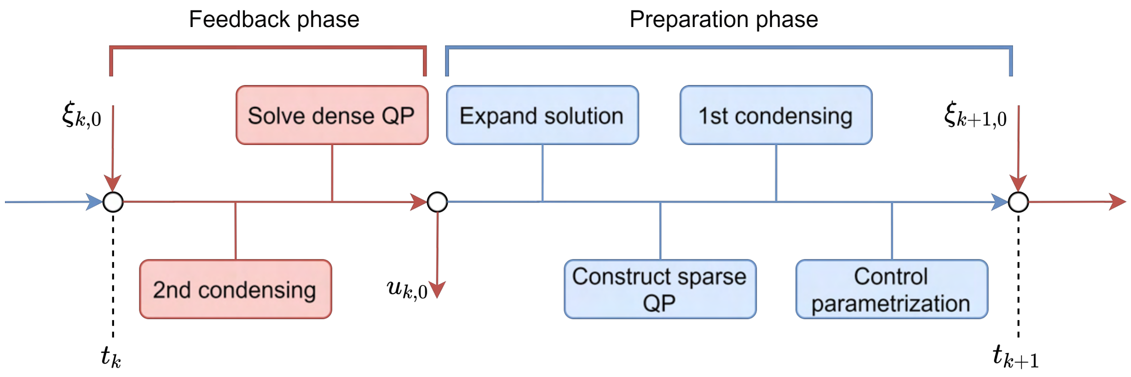


Figure 4.8: Fast NMPC algorithm timeline

condensing is used to enable fast sampling rate, then control parameterization technique is applied on top of the condensed QP to further reduce the optimization complexity. Figure 4.8 shows the timeline of the algorithm in one sample period. The algorithm is divided into a feedback phase and a preparation phase. The feedback phase starts at t_k when the measurement/estimation of current states $\xi_{k,0}$ is available. In the feedback phase, a dense but small-scale QP is formulated by a second condensing, which is then solved quickly by a QP solver to obtain the control input $u_{k,0}$. This control input is sent to the plant immediately to achieve a short feedback delay. For the rest of the sample period, a preparation phase is conducted to prepare the data needed for the next sample. Firstly, the solution of dense QP is expanded to full scale to generate the initial guess for the QP to be solved in the next sample period. Most of the data needed to construct this QP is already available even before the new state feedback comes in. The scale of this sparse QP is then partially reduced by an efficient condensing procedure, and further reduced by control parameterization. At t_{k+1}, feedback phase is repeated again to formulate a fully reduced QP when the new state $\xi_{k+1,0}$ arrives. The major steps of the algorithm are explained in detail in the following.

4.3.3.1 Sparse QP Construction

At each sample step, a sparse QP is obtained from the original NLP (4.48)-(4.53) based on the sequential quadratic programming (SQP) method. Due to the partial separability of the

Lagrangian function, the QP problem can be written as

$$\underset{\Delta s_i,\Delta u_i}{\text{minimize}} \quad \sum_{i=0}^{N-1}\left(\frac{1}{2}\begin{bmatrix}\Delta s_i\\ \Delta u_i\end{bmatrix}^T H_i \begin{bmatrix}\Delta s_i\\ \Delta u_i\end{bmatrix} + \begin{bmatrix}\Delta s_i\\ \Delta u_i\end{bmatrix}^T g_i\right) + \frac{1}{2}\Delta s_N^T P \Delta s_N + \Delta s_N^T g_N \tag{4.57}$$

$$\text{subject to} \quad \xi_0 - s_0^g - \Delta s_0 = 0 \tag{4.58}$$

$$r_i + \Delta s_{i+1} - A_i \Delta s_i - B_i \Delta u_i = 0,\ \forall i \in \{0,\ldots,N-1\} \tag{4.59}$$

$$u_L \leq u_i^g + \Delta u_i \leq u_U,\ \forall i \in \{0,\ldots,N-1\} \tag{4.60}$$

$$\Delta u_L \leq u_i^g - u_{i-1}^g + \Delta u_i - \Delta u_{i-1} \leq \Delta u_U,\ \forall i \in \{1,\ldots,N-1\} \tag{4.61}$$

where $\Delta s_i = s_i - s_i^g$, $i = 0,\ldots,N$ and $\Delta u_i = u_i - u_i^g$, $i = 0,\ldots,N-1$ are the full Newton steps to be computed by the QP. s_i^g and u_i^g are the initial guesses of the QP. H_i is the ith diagonal block of the Hessian of the original NLP's Lagrangian. g_i is the gradient of the original objective function with respect to $\begin{bmatrix}\Delta s_i^T & \Delta u_i^T\end{bmatrix}^T$. Equations (4.58)-(4.61) are the original constraints linearized at the initial guess. $r_i = s_{i+1}^g - F_i\left(s_i^g, u_i^g\right)$ is the residue of the continuity constraint. A_i and B_i are the Jacobian matrices of the dynamics to Δs_i and Δu_i, respectively.

The initial guess of QP takes the values of the shifted solution at the previous time step, also known as warm-start:

$$s_i^g = \tilde{s}_{i+1},\ i = 0,\ldots N-1, \tag{4.62}$$

$$u_i^g = \tilde{u}_{i+1},\ i = 0,\ldots N-2, \tag{4.63}$$

$$s_N^g = F_i\left(s_{n-1}^g, u_{i-1}^g\right), \tag{4.64}$$

where the $\sim$ denotes the variables at previous sample time. $\tilde{s}_i = \tilde{s}_i^g + \Delta\tilde{s}_i$ and $\tilde{u}_i = \tilde{u}_i^g + \Delta\tilde{u}_i$ are the previous initial guess updated by the full Newton step. To avoid the expensive calculation of exact Hessian of Lagrangian, we use Gauss-Newton Hessian approximation [92]. In particular, since the original objective function is already in the linear quadratic form, the diagonal block of Hessian can be simply written as

$$H_i = \begin{bmatrix} C^TQC & S \\ S & R \end{bmatrix},\ i = 0,\ldots N-1, \tag{4.65}$$

which is a constant block diagonal matrix.

The gradient g_i is given as

$$g_i = \begin{bmatrix} C^T Q & S \\ S & R \end{bmatrix} \begin{bmatrix} C s_i^g - \eta_{r,i}\left(\xi_0\right) \\ u_i^g \end{bmatrix}, \; i = 0, \ldots N-1, \tag{4.66}$$

$$g_N = C^T P \left(C s_N^g - \eta_{r,N}\left(\xi_0\right)\right). \tag{4.67}$$

When writing Eqns. (4.66) and (4.67),we stress that the reference input $\eta_{r,i}$, $i = 0, \ldots, N$ are the functions of measured/estimated system states. To be more specific, the reference input $\eta_{r,i}$ is obtained by interpolation based on the trajectory given by the high-level planner and the latest system states. Therefore, the computation of g_i must wait until the new states arrive, causing an additional delay in the feedback phase. To release this overhead, we replace ξ_0 in Eqns. (4.66) and (4.67) with the previous QP solution at the first time step, i.e. $\widetilde{s}_1$, so that g_i, $i = 0, \ldots, N$ can be updated during the preparation phase without the states feedback. The effect of this approximation will be small if the actual plant tracks the reference trajectory closely.

4.3.3.2 The First Condensing

With Guass-Newton approximation, Equations (4.57)-(4.61) describe a convex QP with a special structure. To see it more clearly, the QP is rewritten as

$$\begin{aligned} \underset{\Delta w_s, \Delta w_u}{\text{minimize}} \quad & \frac{1}{2}\Delta w_s^T T^s \Delta w_s + \frac{1}{2}\Delta w_u^T T^u \Delta w_u + \Delta w_s^T T^c \Delta w_u + \Delta w_s^T \tau^s + \Delta w_s^T \tau^u && (4.68) \\ \text{subject to} \quad & A \Delta w_s = B \Delta w_u + r && (4.69) \\ & E \Delta w_u \leq e && (4.70) \end{aligned}$$

where we partition the decision variables into $\Delta w_s := \begin{bmatrix} \Delta s_1^T & \Delta s_2^T & \cdots & \Delta s_N^T \end{bmatrix}^T$ and $\Delta w_u := \begin{bmatrix} \Delta s_0^T & \Delta u_0^T & \cdots & \Delta u_{N-1}^T \end{bmatrix}^T$. The weighting matrices and gradients in the objective function are defined as

$$T^s := diag\left(T_1^s, T_2^s, \ldots, T_N^s\right) = \begin{bmatrix} C^T Q C & & & \\ & C^T Q C & & \\ & & \ddots & \\ & & & C^T P C \end{bmatrix},$$

$$T^u := \begin{bmatrix} T_0^u & T_{0,1}^u & & \\ T_{1,0}^u & T_1^u & & \\ & & \ddots & \\ & & & T_N^u \end{bmatrix} = \begin{bmatrix} C^T Q C & S & & \\ S^T & R & & \\ & & \ddots & \\ & & & R \end{bmatrix},$$

$$T^c := \begin{bmatrix} 0 & 0 & T_1^c & & \\ \vdots & & & \ddots & \\ & & & & T_{N-1}^c \\ 0 & \cdots & & & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & S & & \\ \vdots & & & \ddots & \\ & & & & S \\ 0 & \cdots & & & 0 \end{bmatrix},$$

$$\tau^s := \begin{bmatrix} \tau_1^s \\ \tau_2^s \\ \vdots \\ \tau_N^s \end{bmatrix} = \begin{bmatrix} C^T Q \left(C s_1^g - \eta_{r,1}\right) + S u_1^g \\ C^T Q \left(C s_2^g - \eta_{r,2}\right) + S u_2^g \\ \vdots \\ C^T P \left(C s_N^g - \eta_{r,N}\right) + S u_N^g \end{bmatrix},$$

and

$$\tau^u := \begin{bmatrix} \tau_0^u \\ \tau_1^u \\ \vdots \\ \tau_N^u \end{bmatrix} = \begin{bmatrix} C^T Q \left(C s_0^g - \eta_{r,0}\right) + S u_0^g \\ S \left(C s_0^g - \eta_{r,0}\right) + R u_0^g \\ \vdots \\ S \left(C s_{N-1}^g - \eta_{r,N-1}\right) + R u_{N-1}^g \end{bmatrix}.$$

Equation (4.69) describes the relationship between Δw_s and Δw_u, which is derived form the continuity constraint Eq. (4.59) with the definitions

$$A := \begin{bmatrix} I & & & \\ -A_1 & I & & \\ & \ddots & \ddots & \\ & & -A_{N-1} & I \end{bmatrix},$$

$$B := \begin{bmatrix} A_0 & B_0 & & & \\ & & B_1 & & \\ & & & \ddots & \\ & & & & B_{N-1} \end{bmatrix},$$

and

$$r := \begin{bmatrix} r_0 \\ r_1 \\ \vdots \\ r_{N-1} \end{bmatrix}.$$

Equation (4.70) assembles the remaining initial condition constraints and control effort constraints, where the matrices E and e can be trivially obtained from Eqns. (4.58), (4.60), and (4.61).

We can see that the scale of QP can be greatly reduced by eliminating Δw_s using the linear relationship of Eq. (4.69), due to the fact that A is always invertible. The so-called condensing step yields the following dense QP

$$\underset{\Delta w_2}{\text{minimize}} \quad \frac{1}{2}\Delta w_u^T T_c \Delta w_u + \Delta w_u^T \tau_c \tag{4.71}$$

$$E\Delta w_u \leq e \tag{4.72}$$

where

$$T_c := \begin{bmatrix} T_{0,0} & \cdots & T_{N,0}^T \\ \vdots & \ddots & \vdots \\ T_{N,0} & \cdots & T_{N,N} \end{bmatrix} = T^u + M^T T^s M, \tag{4.73}$$

$$\tau_c := \begin{bmatrix} \tau_0 \\ \vdots \\ \tau_N \end{bmatrix} = \tau^u + M^T \left(\tau^s + T^s m\right), \tag{4.74}$$

$$M := \begin{bmatrix} M_{1,1} & M_{1,2} & & & \\ M_{2,1} & M_{2,2} & M_{2,3} & & \\ \vdots & \vdots & \vdots & \ddots & \\ M_{N,1} & M_{N,2} & M_{N,3} & \cdots & M_{N,N+1} \end{bmatrix} = A^{-1}B, \tag{4.75}$$

$$m := \begin{bmatrix} m_1 \\ \vdots \\ m_N \end{bmatrix} = A^{-1}r. \tag{4.76}$$

Furthermore, thanks to the sparse structure of A and B, the condensing can be done efficiently with complexity of $\mathcal{O}\left(N^2\right)$ using block-wise forward/backward substitution. Following the similar idea in [93], the condensing procedure is given in Algorithms 4.2-4.5.

4.3.3.3 Control Parameterization

After the first condensing, the dimension of QP is reduced form $N\left(n_u + n_x\right) + n_x$ to $Nn_u + n_x$. The scale of QP grows linearly with respect to the prediction horizon, which means the QP can become quite large if the prediction horizon is long. However, a long prediction

Algorithm 4.2 Computing M

```
**Input:** $A$, $B$
**Output:** $M$
$M_{1,1} \leftarrow A_0$
**for** $k = 2, \ldots, N$ **do**
    $M_{k,1} \leftarrow A_{k-1} M_{k-1,1}$
**end for**
**for** $i = 2, \ldots, N+1$ **do**
    $M_{i-1,i} \leftarrow B_{i-2}$
    **for** $k = i-1, \ldots, N$ **do**
        $M_{k,i} \leftarrow A_{k-1} M_{k-1,i}$
    **end for**
**end for**
```

Algorithm 4.3 Computing m

```
**Input:** $A$, $r$
**Output:** $m$
$m_1 \leftarrow r_0$
**for** $i = 2, \ldots, N$ **do**
    $m_i \leftarrow A_{i-1} m_{i-1} + r_{i-1}$
**end for**
```

horizon is usually necessary in vehicle motion control. The long prediction horizon helps to stabilize the system. It is also crucial to ensure a feasible solution when the vehicle encounters obstacles or sharp turn. Therefore, in order to utilize a long prediction horizon without complicating the optimization, the control parameterization method is used in our study. Control parameterization for fast MPC control was first introduced in [94], and was adopted in several real-world applications [95–98]. Open source toolbox pNMPC is also developed by [99], which formulates the OPC in single shooting method and relies on derivative-free optimization. Here, we embed control parameterization into multiple shooting RTI framework.

Control parameterization can be seen as a more general form of move blocking. Unlike move blocking which assumes constant control input over each blocking segment, control parameterization utilizes customized functions to characterize the control input. In general, the parameterization of a piece-wise constant control profile can be defined as a map

$$u_k = \mathcal{U}_{pwc}\left(k, p\right), \tag{4.77}$$

where $p \in \mathbb{P} \subseteq \mathbb{R}^{n_p}$ is a set of parameters in some admissible space, which is usually low-dimensional compared to control input. This map is then substituted into the original

Algorithm 4.4 Computing T_c

Input: M, T^u, T^s, T^c

Output: T_c

```
$W_{N,1} \leftarrow T_N^s M_{N,1}$
for $k = N, \ldots, 2$ do
    $T_{k,0} \leftarrow B_{k-1}^T W_{k,1}$
    $W_{k-1,1} \leftarrow A_{k-1}^T W_{k,1} + T_{k-1}^s M_{k-1,1}$
end for
$T_{1,0} \leftarrow B_0^T W_{1,1}$
$T_{0,0} \leftarrow T_0^u + A_0^T W_{1,1}$
for $i = 1, \ldots, N$ do
    $W_{N,i+1} \leftarrow T_N^s M_{N,i+1}$
    for $k = N, \ldots, i+1$ do
        $T_{k,i} \leftarrow B_{k-1}^T W_{k,i+1}$
        $W_{k-1,i+1} \leftarrow A_{k-1}^T W_{k,i+1} + T_{k-1}^s M_{k-1,i+1}$
    end for
    $T_{i,i} \leftarrow B_{i-1}^T W_{i,i+1} + T_i^u$
end for
for $k = 1, \ldots, N-1$ do
    for $i = 1, \ldots, k+1$ do
        $T_{k+1,i} \leftarrow T_{k+1,i} + T_k^{cT} M_{k,i}$
    end for
end for
$T_{1,0} \leftarrow T_{1,0} + T_{1,0}^u$
```

Algorithm 4.5 Computing τ_c

Input: M, m, T^s, T^c, τ^u, τ^s

Output: τ_c

```
$W_N \leftarrow \tau_N^s + T_N^s m_N$
for $i = N, \ldots, 2$ do
    $\tau_i \leftarrow B_{i-1}^T W_i + \tau_i^u + T_i^{cT} m_i$
    $W_{i-1} \leftarrow A_{i-1}^T W_i + \tau_{i-1}^s + T_{i-1}^s m_{i-1}$
end for
$\tau_1 \leftarrow B_0^T W_1 + \tau_1^u + T_1^{cT} m_1$
$\tau_0 \leftarrow A_0^T W_1 + \tau_0^u$
```

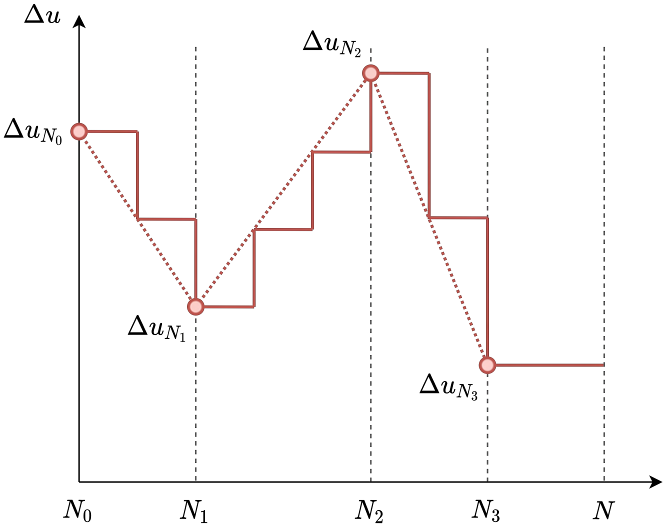


Figure 4.9: Linear interpolating parameterization with $n_p = 4$

formulation of OCP to yield a reduced NLP with decision variables of p. The parameterization schemes are very diverse depending on the problems. It is shown that the impact on control performance will be minor while faster computation can be achieved, if the parameterization scheme is chosen carefully [94].

To accommodate the control parameterization method for the RTI scheme, we choose to parameterize the Newton steps of condensed QP, i.e. Δu_k, instead of the direct control input u_k. A linear interpolating parameterization is used, as illustrated in Fig. 4.9. A number of control inputs at some intermediate sample instants within the prediction horizon are chosen as parameters

$$p := \begin{bmatrix} p_0^T & p_1^T & \cdots & p_{np-1}^T \end{bmatrix}^T = \begin{bmatrix} \Delta u_{N_0}^T & \Delta u_{N_1}^T & \cdots & \Delta u_{N_{n_p-1}}^T \end{bmatrix}^T,$$

where $N_0 := 0$ and $N_i \in \{1, 2, \ldots, N\}$ for $i = 1, 2, \ldots, n_p - 1$. We also define $N_{n_p} := N$.

The controls in between these instants are obtained by the affine combination of the predecessor and successor nodes as

$$\Delta u_i = \left(1 - \frac{i - N_j}{N_{j+1} - N_j}\right) \Delta u_{N_j} + \frac{i - N_j}{N_{j+1} - N_j} \Delta u_{N_{j+1}}, \tag{4.78}$$

where $j = \max\{j \in \{0, \ldots, n_p\} \,|N_j \leq i\}$. The advantage of using such parameterization is that the relationship between u and p can be conveniently expressed as a linear map

$$\begin{bmatrix} \Delta u_0 \\ \Delta u_1 \\ \vdots \\ \Delta u_{N-1} \end{bmatrix} = \Pi_u p, \tag{4.79}$$

where

$$\Pi_u := \begin{bmatrix} I & & & & & \\ \Pi_{1,0} & \Pi_{1,1} & & & & \\ \vdots & \vdots & & & & \\ \Pi_{N_0-1,0} & \Pi_{N_0-1,1} & & & & \\ & I & & & & \\ & \Pi_{N_1+1,1} & \Pi_{N_1+1,2} & & & \\ & \vdots & \vdots & & & \\ & \Pi_{N_2-1,1} & \Pi_{N_2-1,2} & & & \\ & & & \ddots & \ddots & \\ & & & & I & \\ & & & & \Pi_{N_{n_p-2}+1,n_p-2} & \Pi_{N_{n_p-2}+1,n_p-1} \\ & & & & \vdots & \vdots \\ & & & & \Pi_{N_{n_p-1}-1,n_p-2} & \Pi_{N_{n_p-1}-1,n_p-1} \\ & & & & & I \\ & & & & & \vdots \\ & & & & & I \end{bmatrix} \in \mathbb{R}^{Nn_u \times n_p n_u}$$

is a constant matrix and each block of it is a diagonal matrix, which renders the whole matrix very sparse. The matrix Π_u can be computed by Algorithm 4.6.

Substitute Eq. (4.79) into the condensed QP (4.71)-(4.72), we get a parameterized QP

$$\underset{\Delta w_p}{\text{minimize}} \quad \frac{1}{2}\Delta w_p^T T_\Pi \Delta w_p + \Delta w_p^T \tau_\Pi \tag{4.80}$$

$$E_\Pi \Delta w_p \leq e \tag{4.81}$$

where

$$\Delta w_p := \begin{bmatrix} \Delta s_0^T & p^T \end{bmatrix}^T,$$

Algorithm 4.6 Computing Π_Δ

```
Input: $n_u$, $n_p$, $N$, $N_0, N_1, \ldots, N_{n_p-1}$
Output: $\Pi_u$
for $k = 0, \ldots, n_p - 2$ do
    for $i = N_k + 1, \ldots, N_{k+1} - 1$ do
        $\Pi_{i,k+1} \leftarrow \frac{i-N_k}{N_{k+1}-N_k} I$
        $\Pi_{i,k} \leftarrow I - \Pi_{i,k+1}$
    end for
    $\Pi_{N_k,k} \leftarrow I$
end for
for $i = N_{n_p-1}, \ldots, N$ do
    $\Pi_{i,n_p-1} \leftarrow I$
end for
```

$$T_\Pi := \begin{bmatrix} I & \\ & \Pi_u \end{bmatrix}^T T_c \begin{bmatrix} I & \\ & \Pi_u \end{bmatrix},$$

$$\tau_\Pi := \begin{bmatrix} I & \\ & \Pi_u \end{bmatrix}^T T_c,$$

and

$$E_\Pi := E \begin{bmatrix} I & \\ & \Pi_u \end{bmatrix}.$$

The QP has degrees of freedom $n_p n_u + n_x$, which is completely independent of the length of the prediction horizon. It should be noted that the parameterization process creates additional computation overhead. But its effect should be small due to the sparsity of the matrix Π_u.

4.3.3.4 The Second Condensing

The computation process described until now can be performed before the arrival of state feedback. They account for the majority of the computation load of NMPC. When the state feedback ξ_0 is actually available, we can quickly conduct a second condensing to further reduce the QP size by eliminating Δs_0 from Δw_p. The resulted QP is given as

$$\underset{p}{\text{minimize}} \quad \frac{1}{2} p^T T_{pp} p + p^T \left(\tau_p + T_{ps} \left(\xi_0 - s_0^g \right) \right) \tag{4.82}$$

$$E_p p \leq e - E_s \left(\xi_0 - s_0^g \right) \tag{4.83}$$

where T_{pp} and T_{ps} are the partitions of $T_\Pi := \left[\begin{array}{c|c} T_{ss} & T_{sp} \\ \hline T_{ps} & T_{pp} \end{array}\right]$, τ_p is the partition of $\tau_\Pi := \left[\begin{array}{c} \tau_s \\ \hline \tau_p \end{array}\right]$, and E_p is the partition of $E_\Pi := \left[\begin{array}{c} E_s \\ \hline E_p \end{array}\right]$.

The QP (4.82)-(4.83) is now fully condensed without any sparsity. It can be efficiently solved by a generic QP solver. Once the solution p^* is obtained, the control input to the plant can be constructed immediately as $u_0^* = u_0^g + p_0^*$.

4.3.3.5 Solution Expansion

The final step is to expand the condensed QP solution so that it can be used as the initial guess for the QP of the next time step. The process is fairly straightforward. Since the Gauss-Newton Hessian approximation is used, the multipliers become irrelevant, and all we need are the primal variables. We first expand the p into Δw_p by

$$\Delta w_p = \begin{bmatrix} \xi_0 - s_0^g \\ p \end{bmatrix}, \tag{4.84}$$

then Δw_u is obtained as

$$\Delta w_u = \begin{bmatrix} I & \\ & \Pi_u \end{bmatrix} \Delta w_p, \tag{4.85}$$

finally, the full Newton step is given as

$$\begin{bmatrix} \Delta w_s \\ \Delta w_u \end{bmatrix} = \begin{bmatrix} M \Delta w_u + m \\ \Delta w_u \end{bmatrix}. \tag{4.86}$$

4.4 Simulation Results

We implement the vehicle model in Matlab. The plant model includes 1st order transfer functions to reflect the effect of low-level actuator control systems. The NMPC is formulated using the symbolic framework of the open-source tool CasADi. The model and controller parameters are listed in Appendix A. The condensing and control parameterization are implemented in Matlab with the required derivatives provided by algorithmic differentiation of CasADi [100]. The open-source qpOASES is used for solving the condensed QP [101]. All simulations were performed on a desktop computer with a 3.50GHz Intel Core i5-6600K CPU and 8GB RAM.

4.4.1 High μ Braking

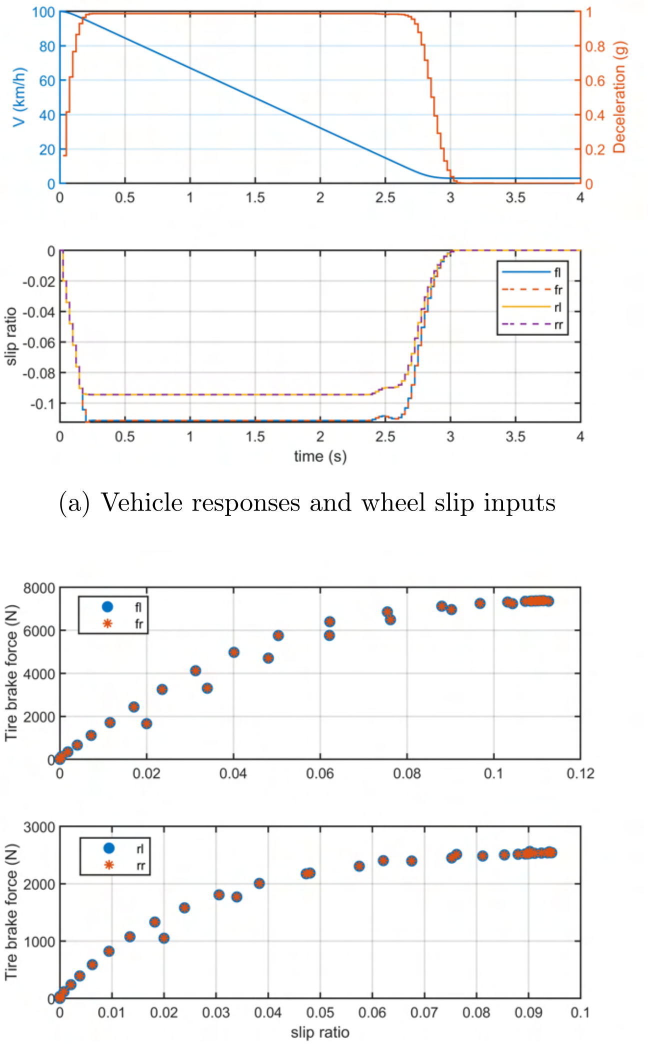
In the high μ braking simulation, the vehicle brakes at an initial speed of 100km/h on an asphalt road with $\mu = 1$ until it stops. This simulation is used to evaluate if the slip ratio profile from NMPC is realistic for level-level actuators. We use prediction step $N = 20$, and sample time $T_s = 25$ms. The control inputs are parameterized using linear interpolation. By some trial and error based on simulation results, we choose the parameterization points to be $N_i = [0, 1, 11, 15]$.

We first determine the amplitude and rate constraints on the slip ratio. Although some literature put hard constraints on slip ratio to be within the positive slope of tire force [102], we will relax the slip ratio constraint by allowing it to vary from 0 to -1 in order to avoid infeasibility due to tracking error of actuator control. Since the tire nonlinearity is considered explicitly in the NMPC, we expect the controller to be able to avoid excessive slip by itself without explicit constraints. For the rate constraint, we set the $-1 \leq \dot{\lambda} \leq 1$ to match the actuator limit. The derivation of rate constraint is included in Appendix B. A high-pass frequency shaping filter $R_f(s) = \frac{\alpha_1 s+1}{\alpha_2 s+1}$ is applied to wheel slip inputs to avoid high-frequency excitation, whose parameters are given in Appendix A.

In the simulations, we command the controller to track a constant target deceleration with entry speed of 100km/h and no steering action. Figure 4.10 shows the simulation results when the target deceleration equals 1g. Figure 4.10(a) shows the time history of vehicle speed, deceleration, as well as the slip ratio commands. We can see that the target deceleration is achieved, and the slip ratio command is also smooth. It is also interesting to see that the controller put more effort on the front axle than the rear one because of the fact that the front axle braking is more effective due to the load transfer. Figure 4.10(b) shows the tire brake force profiles. We can see that both front and rear tires reach their peak values but stay out of the negative slop region. It shows the controller is able to exploit the maximum tire force while keeping the vehicle operating in stable region.

4.4.2 Low μ Braking

Now we are going to investigate the braking control performance on a snowy surface, where $\mu = 0.4$. In particular, we use the prediction model in the NMPC controller that corresponds to $\mu = 1$. Furthermore, the target deceleration is remained to be 1g despite the maximum achievable deceleration is only 0.4g. The discrepancy between the prediction model, target



(a) Vehicle responses and wheel slip inputs

(b) Tire force profiles

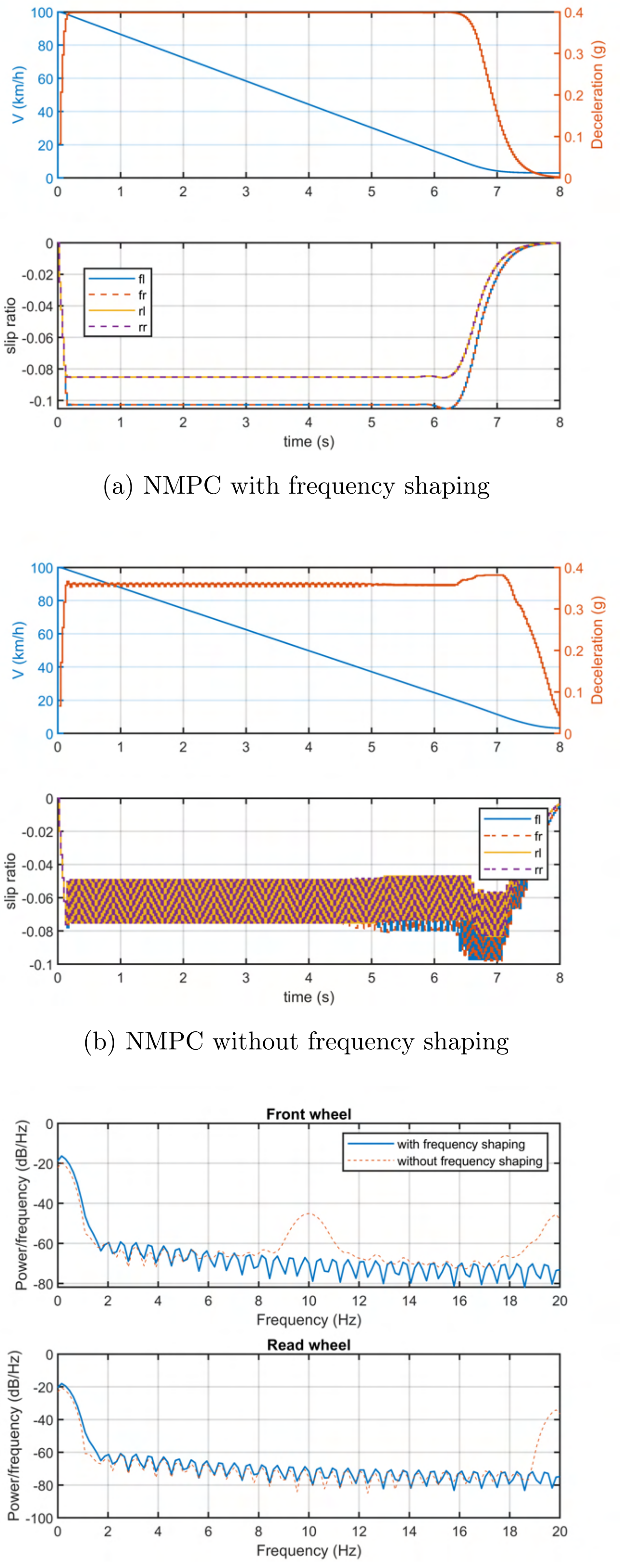
Figure 4.10: High μ ($\mu = 1$) braking with target deceleration of 1g

and the physical limit might occur due to the difficulty on real-time road friction estimation. We want to see how the controller performs under this discrepancy. Figure 4.11 shows the vehicle response and wheel slip inputs of the simulation. For comparison, we present two simulation results, where Fig. 4.11(a) corresponds to the response using NMPC with frequency shaping and Fig. 4.11(b) is the response using NMPC without frequency shaping. We can see from Fig. 4.11 that the vehicles achieve the similar amount of deceleration in both cases. Although the decelerations are similar, from Fig. 4.11(a), we see that the wheel slip command of NMPC with frequency shaping is smooth. However, as shown in Fig. 4.11(b), the control input of NMPC without frequency shaping suffers serious high frequency chattering during the steady state, which is very undesirable for low-level actuator. The control input frequency characteristics of the two cases can also be observed for the power spectral density plots of Fig. 4.11(c). We can see that the control inputs of no frequency shaping case show high frequency peak in the high frequency, while those of frequency shaping case show consistent low density in the high frequency range.

Without frequency shaping, the only way to remove the chattering is to increase the weighting on the control input. Figure 4.12 shows the simulation result where the control input weighting is increased. We can see that the chattering of control input is diminished with larger weighting. On the other hand, the deceleration also becomes smaller due to the increased weighting on control input. The tradeoff between control performance and input can be explained by Fig. 4.13. For the NMPC without frequency shaping, the weighting over the different frequency is adjusted as a whole. Therefore, the low frequency performance must be sacrificed when we increase the weighting to eliminate the high frequency chattering. Whereas for the NMPC with frequency shaping, we have two tuning knobs to independently adjust the weighting in the low frequency and high frequency. Therefore, the low frequency control performance will not be compromised by large weighting in the high frequency.

4.4.3 Split μ Braking

In the split μ braking, the tires on the left hand side operate on asphalt with $\mu = 1$, and the right hand side tires operate on a snowy surface with $\mu = 0.4$. It is a challenging task because the discrepancy in road friction between left and right might create a large yaw moment disturbance that destabilizes the vehicle. A good controller should achieve a short barking distance while keeping the vehicle stable. For the trajectory tracking purpose, it should also keep the lateral and yaw angle deviation as small as possible. To better examine the slip control performance, we will not use steering control in the simulation.



(a) NMPC with frequency shaping

(b) NMPC without frequency shaping

(c) Power spectral density

Figure 4.11: Low μ (μ=0.4) braking with target deceleration of 1g

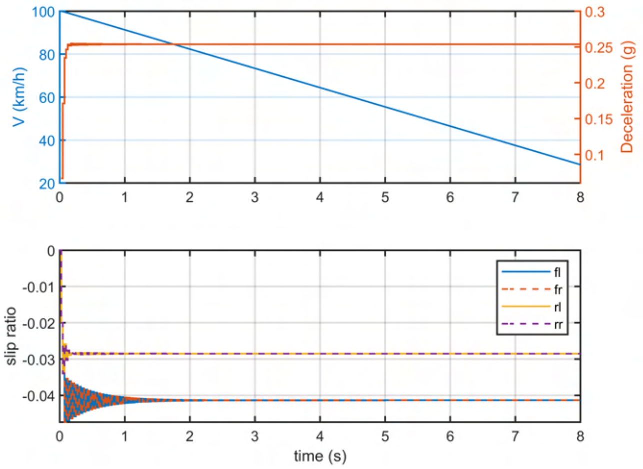


Figure 4.12: Low μ (μ=0.4) braking with no frequency shaping and increased control input weighting

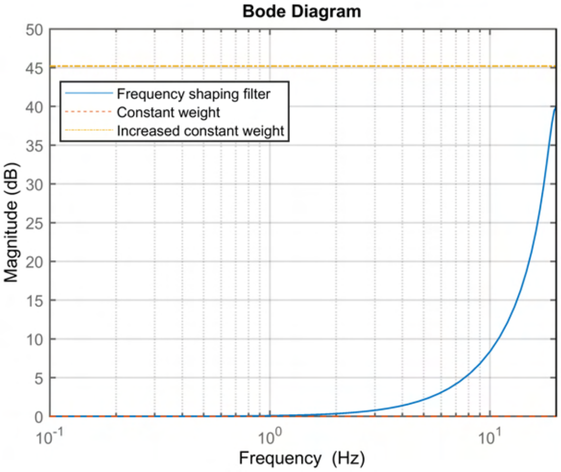


Figure 4.13: Control input weighting

Figure 4.14 shows the simulation results with target deceleration of 0.4g which is the maximum possible deceleration on the snowy surface. We can see form Fig. 4.14(a) that the vehicle tracks the reference path closely with very small lateral and yaw angle deviations. Figure 4.14(b) shows that the deceleration reaches the target value and the control inputs are smooth. Especially, the slip ratio of the high μ side is smaller than that of the low μ side such that the longitudinal forces of two sides are balanced. From Fig. 4.14(c), we can see that the tire forces of the right tires are saturated to achieve the maximum deceleration, and the forces on the left tires are kept similar to those of right tires to avoid generating large yaw moment disturbance.

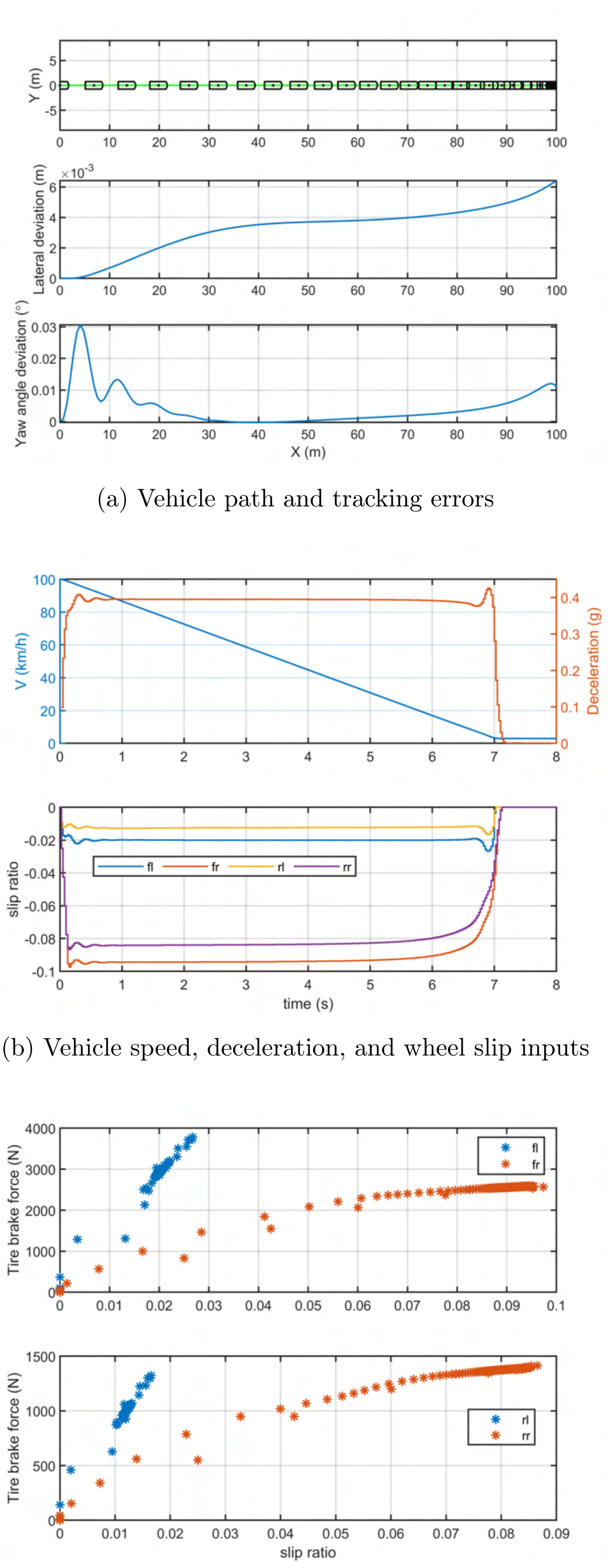
We also conduct a simulation with target deceleration of 0.3g. Figure 4.15 shows the results. We can see that the vehicle is able to achieve the desired deceleration using smooth control inputs. And the tracking deviations are even smaller than the previous case.

4.4.4 Sine with Dwell

We have evaluated the slip control performance of NMPC in the previous two simulations. Now we will assess the full control capability of NMPC both 4WS and 4WID. We first perform the sine with dwell (SD) test. The SD test defined by FMVSS 126 regulation [103] is a standard procedure used to evaluate the performance of the ESC system on vehicle lateral stabilization. Here we utilize a similar procedure with small modifications to evaluate the trajectory tracking control performance.

Firstly, we generate a SD trajectory using an uncontrolled vehicle. The open-loop front wheel steering profile is shown in Fig. 4.16, which is obtained according to the standard of FMVSS 126. Figure 4.17 shows the uncontrolled vehicle trajectory under this steering input with an entry speed of 80km/h. Due to the lack of stability control and large steering angle, we can clearly see the oversteer tendency towards the end of the simulation.

To evaluate the performance of our control algorithm, the open-loop trajectory is used as a reference to the NMPC control for tracking. In the closed-loop simulation, we evaluate two sets of configurations, one with 4WS and slip control and the other with only FWS and slip control. Figure 4.18(a) shows the trajectory tracking performance of the controllers. We can see that both control configurations are able to mitigate the oversteering of vehicle and improve the vehicle stability. The vehicle trajectory is also close to the reference one in both cases. The 4WS with wheel slip control shows superior tracking performance than the FWS. Figure 4.18(b) shows the vehicle speed, lateral acceleration, and side slip angle comparison.

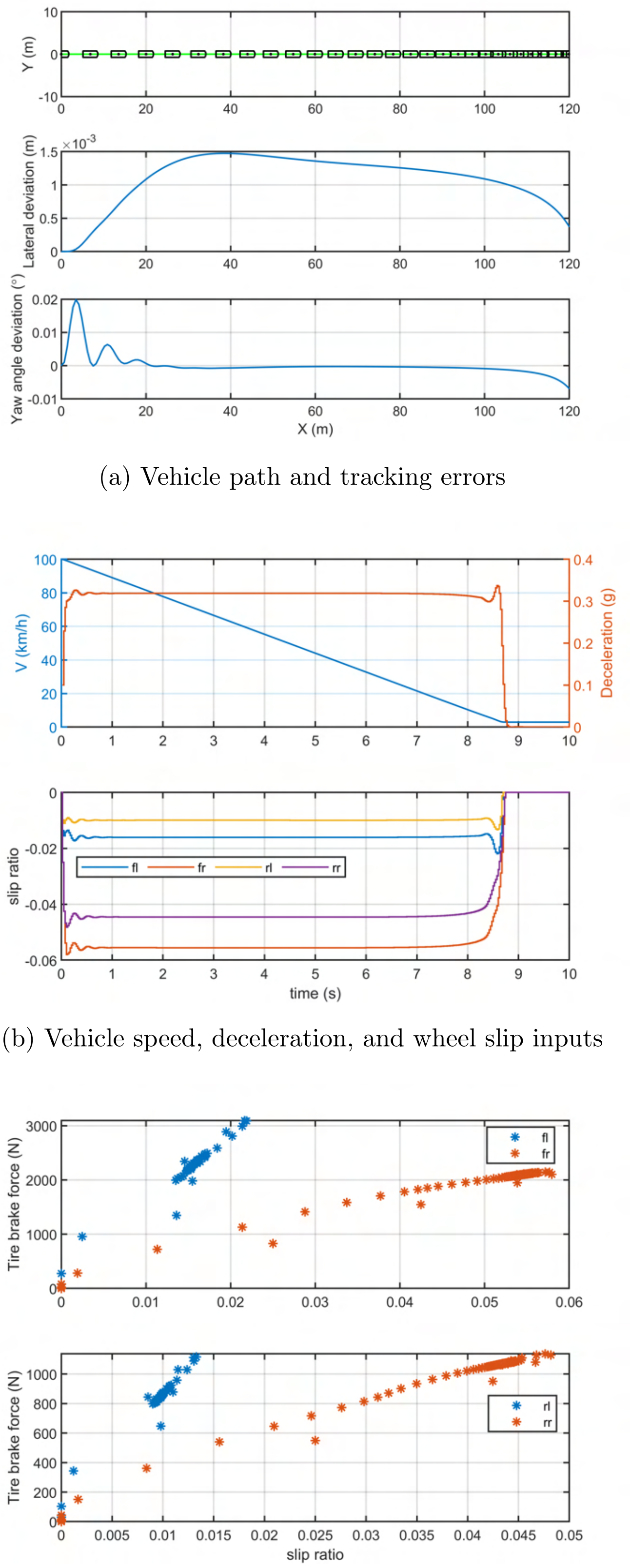


(a) Vehicle path and tracking errors

(b) Vehicle speed, deceleration, and wheel slip inputs

(c) Tire force profiles

Figure 4.14: Split μ (left: $\mu = 1$; right: $\mu = 0.4$) braking with target deceleration of 0.4g



(a) Vehicle path and tracking errors

(b) Vehicle speed, deceleration, and wheel slip inputs

(c) Tire force profiles

Figure 4.15: Split μ (left: $\mu = 1$; right: $\mu = 0.4$) braking with target deceleration of 0.3g

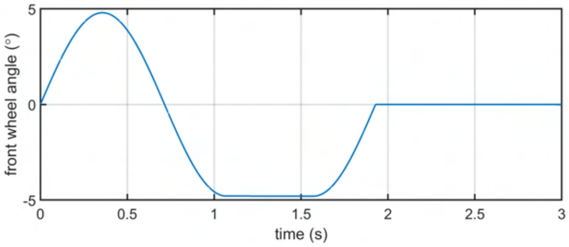


Figure 4.16: Front wheel steering angle of sine with dwell simulation

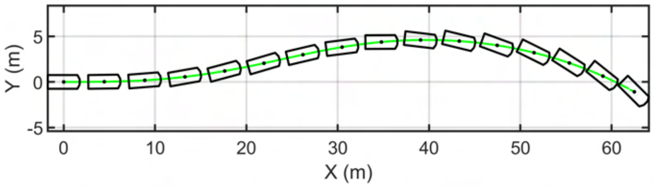


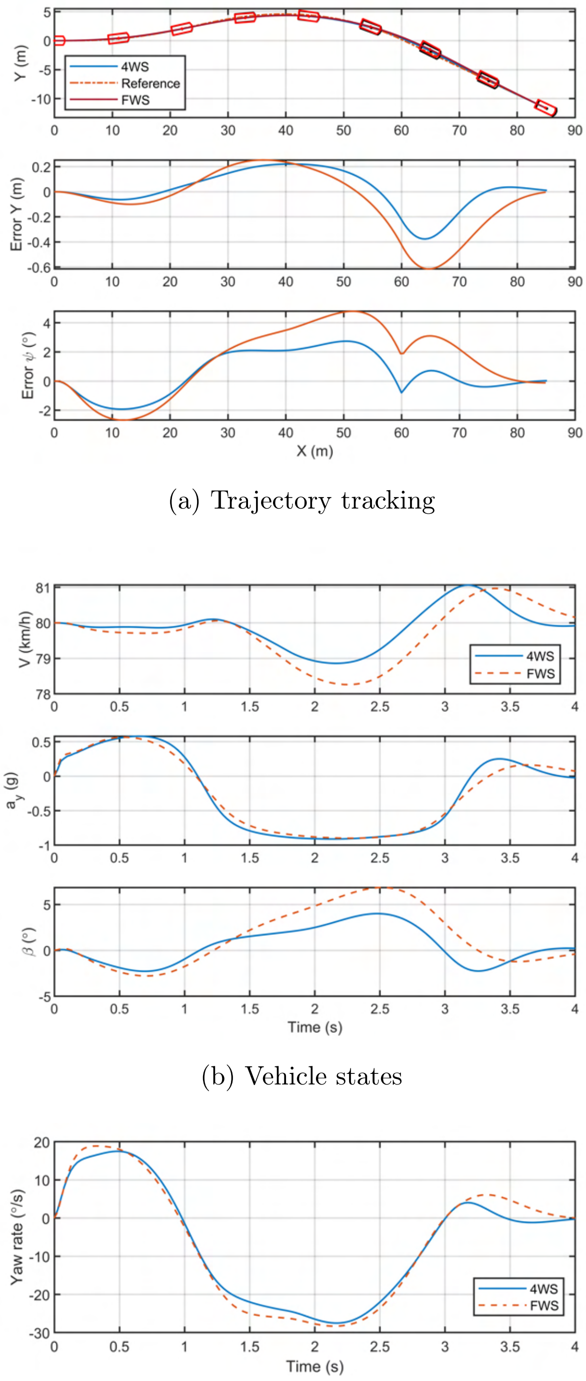
Figure 4.17: Open-loop trajectory of sine with dwell simulation. The green solid line denotes vehicle C.G. trajectory, and the black polygon denotes vehicle's shape.

We can see that the 4WS with slip control achieves smaller vehicle speed variation and side slip angle. The lateral accelerations in both cases are close to 0.9g which indicates the vehicle is operating at the handling limit. Figure 4.18(c) shows the vehicle yaw rate response. According to the standard test criteria, the yaw rate should decay to 35% and 20% of the peak value at 2.929s and 3.679s, respectively. The simulation result shows that the controlled vehicle satisfy these criteria in both cases.

Figure 4.19 shows the control input profiles of the two control setups. We can see that both control setups generate smooth input profiles. For the 4WS case, rear steering is controlled to be in-phase with the front steering to reduce the side-slip angle. In the FWS case, the front wheel steering shows a similar profile to that of 4WS. However, due to the lack of rear steering, the slip control becomes more active in order to generate additional yaw moments, which also results in large vehicle speed variation.

4.4.5 High Speed Double Lane Change

In this simulation, the vehicle is required to make a double lane change (DLC) maneuver at high speeds with different road friction conditions. For 4WS, rear-wheel is only used for micro steering by penalizing its high frequency maneuvers using frequency shaping. For comparison purposes, we also perform the same simulations using the vehicle with front wheel steering (FWS) and 4WID vehicle. The DLC track is set up according to the ISO 3888-1 standard [104]. The vehicle speed in each simulation is selected as the highest one at which



(a) Trajectory tracking

(b) Vehicle states

(c) Yaw rate response

Figure 4.18: Closed-loop control of sine with dwell ($\mu = 1$)

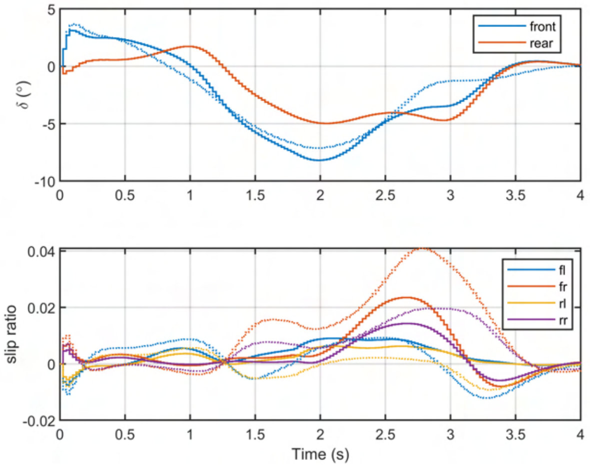


Figure 4.19: Control inputs profiles of sine with dwell. The solid line denotes the response of 4WS and the dashed line denotes the response of FWS.

the maneuver can be completed within the boundary by the vehicle equipped with 4WS and 4WID. The reference path is given by functions of displacement in X direction

$$Y_{ref} = \frac{d_{y1}}{2}\left(1 + \tanh z_1\right) - \frac{d_{y2}}{2}\left(1 + \tanh z_2\right),$$

$$\psi_{ref} = \arctan\left(d_{y1}\frac{1}{\cosh^2 z_1}\frac{1.2}{d_{x1}} - d_{y2}\frac{1}{\cosh^2 z_2}\frac{1.2}{d_{x2}}\right),$$

where $z_1 = a_1\left(X - b_1\right) - c_1$, $z_2 = a_2\left(X - b_2\right) - c_2$. $a_{1,2}$, $b_{1,2}$, $c_{1,2}$, $d_{y1,2}$, and $d_{x1,2}$ are constant parameters which are given in the Appendix A. The reference longitudinal velocity $\dot{x}$ is same as the vehicle's entry speed.

We first conducted the simulation on an asphalt road ($\mu = 1$) with a vehicle entry speed of 140km/h. The vehicle tracking performance is shown in Fig. 4.20. We can see that both vehicles are able to finished the maneuver while remaining stable and inside the boundary. The vehicle with 4WS shows better tracking performance with slightly smaller lateral and yaw angle deviations. Figure 4.21 shows the vehicle speeds, lateral accelerations, and side slip angles of the two vehicles. It can be seen that the vehicle with FWS has larger fluctuations in speed tracking. The lateral accelerations are very similar. We can see that the maximum lateral accelerations are approaching to 0.9g in both cases, which indicates that the vehicle really reaches its handling limit in simulation. We can also see the vehicle with 4WS has smaller side slip angle. The control inputs of two vehicles are shown in Fig. 4.22. We can see that all the control inputs are smooth. For the 4WS case, the rear steering is of similar

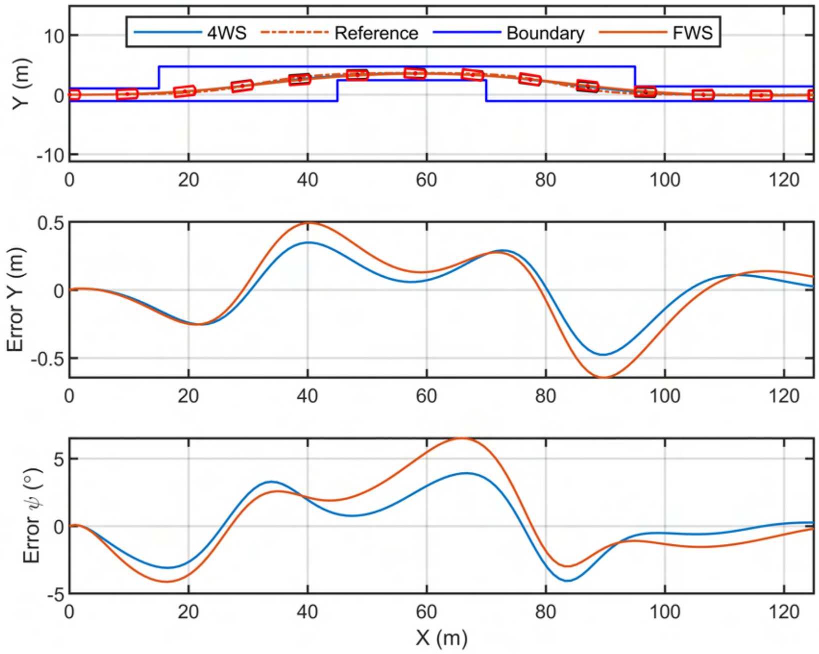


Figure 4.20: DLC on asphalt at 140kph (μ=1). Top plot: the vehicle with 4WS is outlined by black polygon, and the one with FWS is outlined by red polygon.

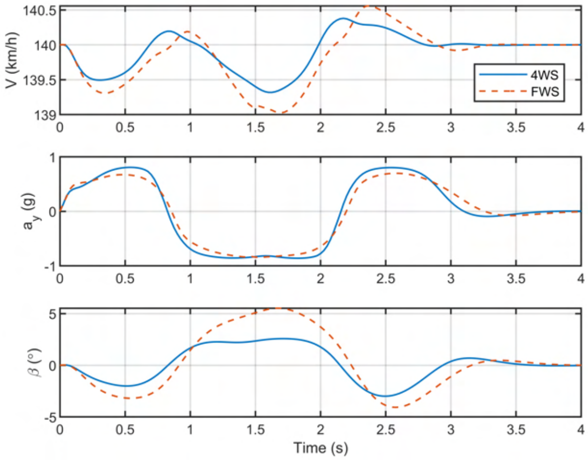


Figure 4.21: Vehicle state responses of DLC on asphalt (μ=1) at 140kph

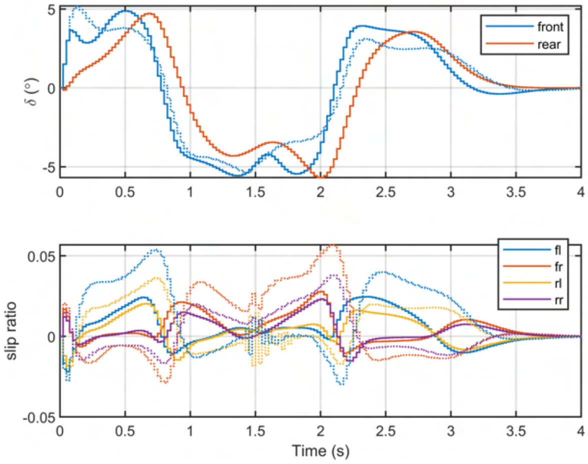


Figure 4.22: Control input demands of DLC on asphalt (μ=1) at 140kph. The solid line denotes 4WS, and the dashed line denotes FWS.

amplitude and phase as those of the front steering. We can also observe that the FWS vehicle has larger control actions in slip ratios due to the absence of rear steering control. These larger actions of slip ratio also cause more oscillations in vehicle speed.

Next, simulations are executed on a snowy surface ($\mu = 0.4$) at an entry speed of 70km/h. The vehicle tracking performance is shown in Fig. 4.23. Similarly to the previous simulation, the 4WS vehicle shows slightly better tracking performance. Figure 4.24 shows the vehicle speed, lateral accelerations, and side slip angles comparisons. Again, larger speed variations and side slip angle are shown in the FWS vehicle, and lateral accelerations are similar. Control inputs are smooth as shown in Fig. 4.25.

We then perform the simulation on an icy surface ($\mu = 0.17$) at a speed of 50km/h. This is the most challenging case, as the tire fore can become highly nonlinear and easily saturated. The results are presented in Fig. 4.26, 4.27, and 4.28. Both vehicles complete the maneuver successfully. We can see that there is a little chattering in front steering due to the fact that vehicle becomes considerably more difficult to stabilize on icy surface. From the lateral acceleration response, we can see that the vehicle has reach its physical limit with $a_y \cong 0.15g$. Nevertheless, the overall control performance is satisfactory in this extreme situation.

Finally, we check the real-time feasibility of the controller. Figure 4.29 shows computation time for the three DLC simulations presented previously. For comparison purposes, we also show the computation time of the NMPC without control parameterization. We can see that the computation time of the controller with control parameterization is less than those

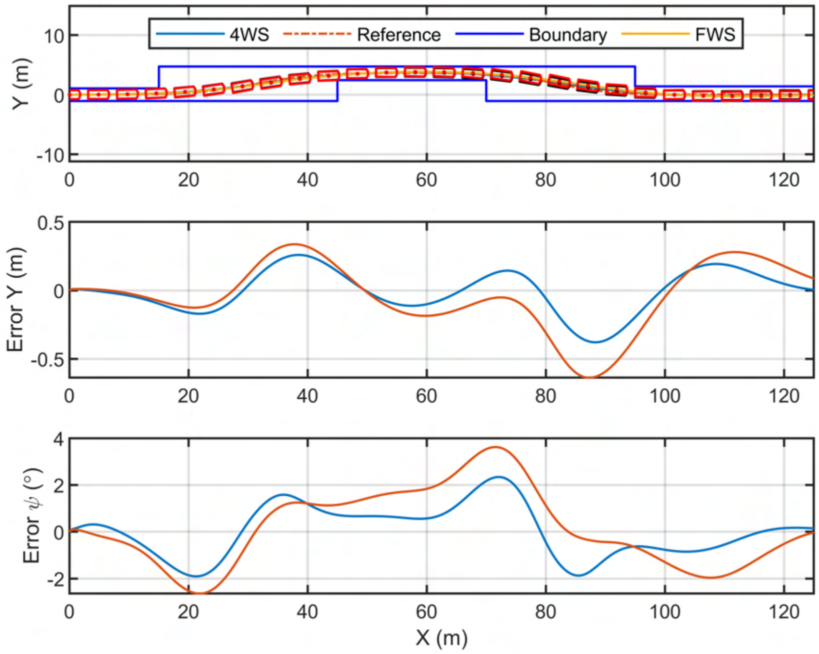


Figure 4.23: DLC on snowy surface (μ=0.4) at 70kph. Top plot: the vehicle with 4WS is outlined by black polygon, and the one with FWS is outlined by red polygon.

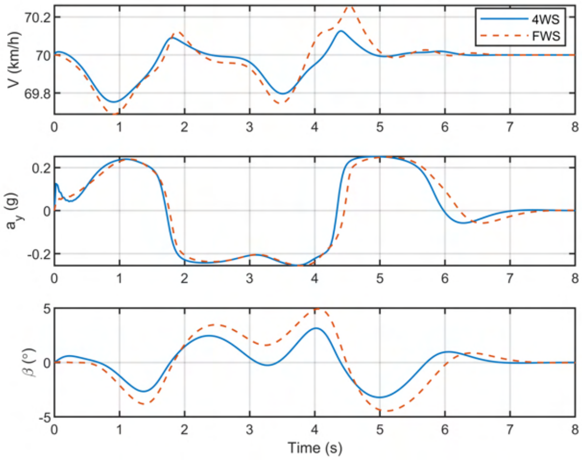


Figure 4.24: Vehicle state responses of DLC on snowy surface (μ=0.4) at 70kph

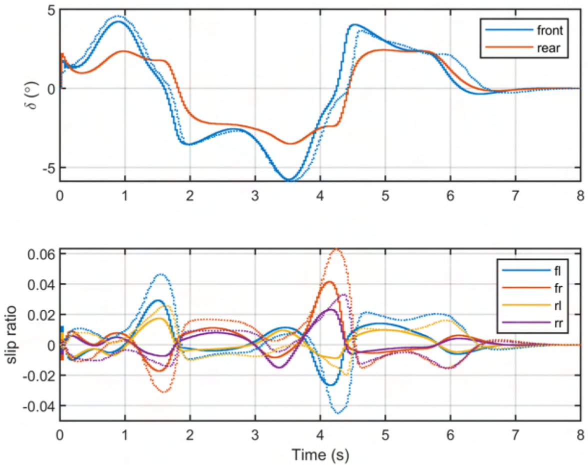


Figure 4.25: Control input demands of DLC on snowy surface (μ=0.4) at 70kph. The solid line denotes 4WS, and the dashed line denotes FWS.

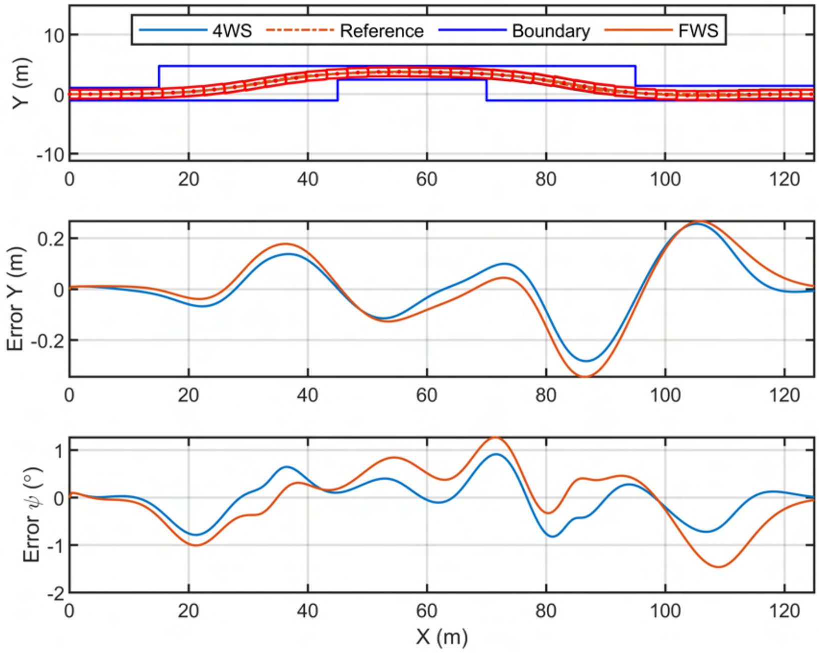


Figure 4.26: DLC on icy surface (μ=0.17) at 50kph. Top plot: the vehicle with 4WS is outlined by black polygon, and the one with FWS is outlined by red polygon.

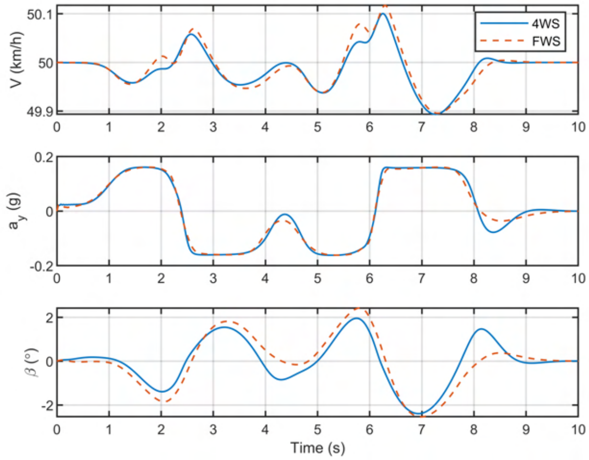


Figure 4.27: Vehicle state responses of DLC on icy surface (μ=0.17) at 50kph

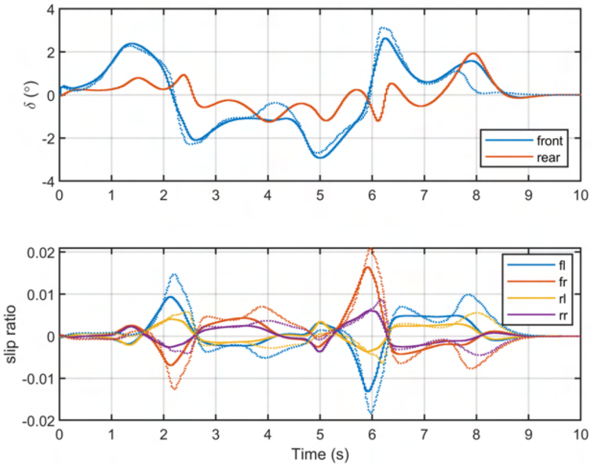
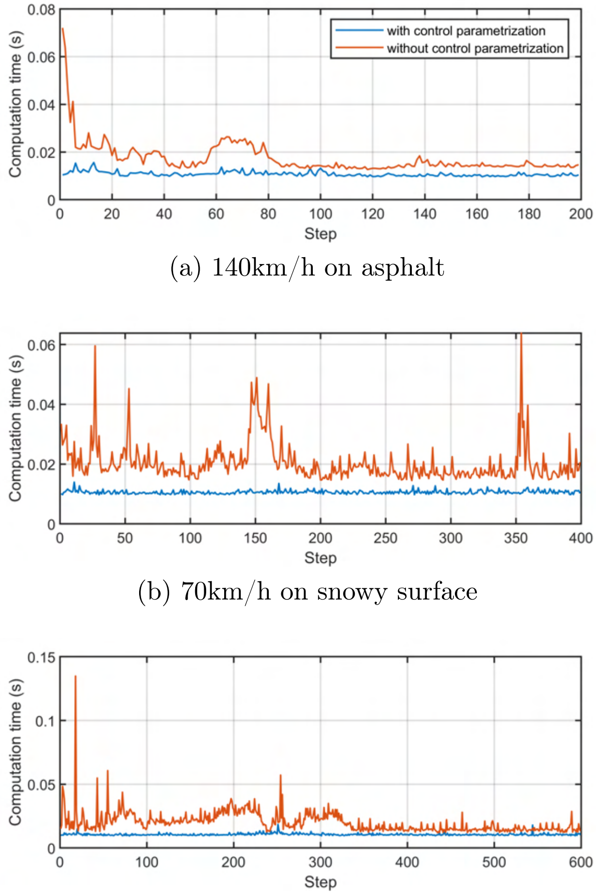


Figure 4.28: Control input demands of DLC on icy surface (μ=0.17) at 50kph. The solid line denotes 4WS, and the dashed line denotes FWS.



(a) 140km/h on asphalt

(b) 70km/h on snowy surface

(c) 50km/h on icy surface

Figure 4.29: Computation times of DLC simulations

Table 4.1: Tracking performance comparison

Road surface	Asphalt		Snowy		Icy	
Control Parameterization	with	without	with	without	with	without
Maximum lateral deviation (m)	0.76	0.79	0.39	0.37	0.39	0.34
Maximum angular deviation (°)	5.79	4.86	2.22	2.18	2.11	2.24

without and smaller than the sampling period (25ms). Table shows the tracking performance of the controller with and without control parameterization. We can see that there are only minor differences in performance between the two controllers.

4.5 Conclusions

A NMPC controller is proposed for autonomous vehicle trajectory tracking using 4WS and 4WID. The controller is based on a nonlinear double track model with a Magic Formula tire model. Frequency shaping is used to allocate control inputs for steering and slip control. A fast NMPC algorithm is proposed based on RTI with condensing and control parameterization. The proposed control system is tested through simulations. It is shown that the vehicle achieves good braking distance with smooth control inputs. Sine with dwell and double lane change simulations show that the controller is able to stabilize the vehicle at the handling limit and achieve good tracking performance under various road frictions.

Chapter 5

Motion Planning for Autonomous Vehicle based on NMPC

In this chapter, we develop a motion planning algorithm for autonomous vehicles. The motion planning algorithm aims to generate a reference trajectory for the lower-level tracking controller. The algorithm should generate maneuvers like braking, acceleration, lane keeping or lane change. The trajectory should be dynamically feasible so that it can be realized by the vehicle within physical limitations. In addition, other requirements regarding driving safety, comfort, and traffic rules might also be enforced to the trajectory planning.

5.1 Introduction

MPC has been widely used for vehicle motion planning mainly because of its ability to handle different constraints systematically and explicitly. In [105], a trajectory generation algorithm for unmanned ground vehicles is proposed based on MPC. Information on obstacles is incorporated online into the NMPC framework and then solved online with nonlinear programming. Lee et al. [106] presented a scenario-based MPC trajectory planning algorithm that consists of spatial planning with embedded temporal optimization that leverages waypoint information on the road. Trajectory candidates are generated by solving spatial and temporal optimization problems for each feasible scenario, and then integrated together. In [107], a collision imminent steering system is developed using nonlinear model predictive control to perform a lane change at high speed in a highway environment. Two formulations are presented based on minimum distance formulation and minimum slip formulation, respectively. Taherian et al. [108] proposed an integrated trajectory planning with collision avoidance based on MPC. A torque vectoring controller is also designed for lateral-yaw stabilization under low friction and crosswinds on highways.The steering input computed from the MPC is used to steer the vehicle along the reference trajectory while the torque vectoring controller provides additional lateral-yaw stability. In [109], authors presented a vehicle motion planning and control framework based on MPC. Collision avoidance is ensured by transforming the predicted motion and uncertainty of other road users into constraints within the MPC formulation. Hajiloo et al. [110] proposed a emergency collision avoidance system using steering and differential braking. MPC is used for developing a combined path planning and tracking controller with a hierarchical structure that considers collision avoidance, vehicle stability, and path tracking. Quirynen et al. [111] presents a hierarchical control framework for obstacle avoidance in autonomous driving systems. A particle-filtering is used for low-rate, long-term motion planning. A NMPC-based vehicle controller is used to track the reference trajectory and avoid obstacles. In [112], a two-stage nonlinear nonconvex control approach for autonomous vehicle driving is developed. An outer-loop nonlinear model predictive control is adopted for generating the collision-free trajectory with the resultant input based on a simplified vehicle model. The inner loop is a simple linear feedback controller based on an optimal preview distance.

In this chapter, we will present the application of NMPC to autonomous vehicle motion planning. Note that the underlying control framework is the same as that used in the previous chapter. Nevertheless, the algorithm in this chapter differs significantly from the previous one in terms of problem formulation as well as numerical method. Furthermore, we incorporate control barrier function into the NMPC scheme for safety guarantee.

The rest of this chapter is organized as follows. In Section 5.2, the vehicle model for motion planning is described. Section 5.3 gives the definitions of motion planning problem. The NMPC formulation and numerical method are discussed in Section 5.4. In Section 5.5, we present simulation results of proposed motion planning algorithm. Finally, Section 5.6 concludes the chapter.

5.2 Vehicle Modeling

In order for the NMPC to generate a feasible trajectory, the vehicle model must be able to reflect essential physical properties of the real system. On the other hand, the vehicle model should not be over-complicated, since the optimization of motion planning is often computationally expensive due to the various kinds of complex path constraints. In this work, a spatial domain nonlinear point-mass model is used.

5.2.1 Point-Mass Vehicle Model

Figure 5.1 shows the schematic of the model used in this work. The vehicle is represented by a particle, and its motion is described with respect to the road adherent Frenet frame. n

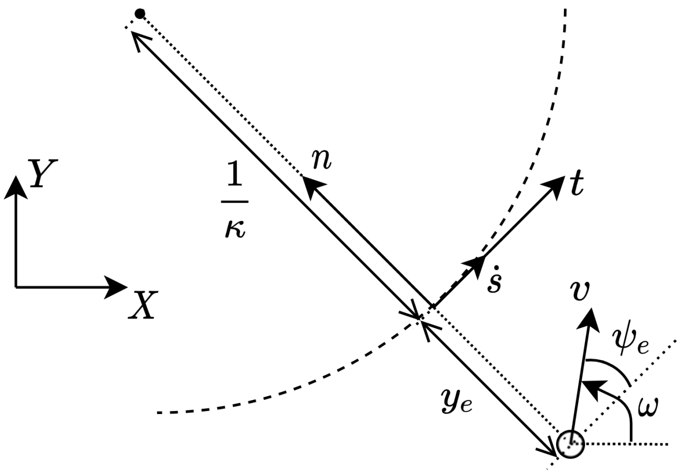


Figure 5.1: Schematic of point mass model

and t are the two axis of the Frenet coordinate which respectively point to the normal and tangential direction of the reference path. This reference path could represent the center line of the lane or any reference path from the decision making module at a high level. The origin of the Frenet coordinate coincides with the projection of the vehicle along the n direction to the reference path.

The equations of motion of the vehicle are given by

$$\dot{s} = \frac{1}{1 - y_e \kappa\left(s\right)} v \cos \psi_e, \tag{5.1}$$

$$\dot{y}_e = v \sin \psi_e, \tag{5.2}$$

$$\dot{\psi}_e = \omega - \frac{\kappa\left(s\right)}{1 - y_e \kappa\left(s\right)} v \cos \psi_e, \tag{5.3}$$

$$\dot{v} = a, \tag{5.4}$$

where v is the vehicle speed, a is the vehicle acceleration. ψ_e is the angular deviation between v and t axis, y_e is the lateral deviation of vehicle to the reference path along n axis, s is the arc length traveled by the vehicle's projection on the reference path. ω is the angular speed of vehicle's heading with respect to global axis, and κ is the curvature of the reference path, which is a function of s. Note that the model implies $1 - y_e \kappa\left(s\right) \neq 0$, which is valid under normal driving conditions where $y_e \ll {}^{1}\!/\!_{\kappa}$.

Equations (5.1)-(5.4) are compactly written as

$$\dot{\xi}\left(t\right) = f_{\kappa(s)}\left(\xi\left(t\right), u\left(t\right)\right), \tag{5.5}$$

where $\xi := \begin{bmatrix} y_e & \psi_e & v & s \end{bmatrix}^T$ and $u := \begin{bmatrix} \omega & a \end{bmatrix}^T$. The subscript κ indicates that the curvature enters the dynamics as a varying parameter. We assume the state of system and road curvature are known from measurement or estimation.

5.2.2 Spatial Transformation

In this section, we perform a spatial transformation to the time-dependent dynamics (5.5). The procedure is similar to what is given in [113]. This reformulation changes the independent variable of the model from time t to arc length s. Assuming $\dot{s} > 0$, i.e. vehicle is always driving forward, we have the following relationship

$$\xi' := \frac{d\xi}{ds} = \frac{d\xi}{dt}\frac{dt}{ds} = \frac{\dot{\xi}}{\dot{s}}. \tag{5.6}$$

We will use the superscript $'$ to denote the derivative of a variable with respect to s in the rest of the chapter. Apply Eq. (5.6) to (5.5), we obtain the vehicle model in the spatial domain as

$$\xi'\left(s\right) = \frac{f_{\kappa(s)}\left(\xi\left(t\right), u\left(t\right)\right)}{\dot{s}} := f^s_{\kappa(s)}\left(\xi\left(s\right), u\left(s\right)\right). \tag{5.7}$$

Now the dynamics of state variables are functions of s. Later, we will see that the spatial formulation renders road boundaries into simple box constraints on the state. It also makes the parameter κ an explicit function of the independent variable, thus simplifying the simulation of the system. Note that the independent spatial variable s now becomes a clock state with dynamics $s' = 1$ embedded in the system (5.7).

5.2.3 Convert Cartesian Coordinates to Frenet Coordinates

The sensor measurements of the vehicle are usually given by the global Cartesian coordinates. In addition, the trajectory tracking control system is also formulated in Cartesian coordinate frame. Therefore, the conversion between Cartesian coordinates and Frenet coordinates must be performed such that the tracking and planning control systems can work together. Assuming that the reference path is represented by a discrete set of coordinates, $\begin{bmatrix} X_{r,i} & Y_{r,i} & s_{r,i} & \kappa_{r,i} \end{bmatrix}^T$ for $i = 0, 1, \ldots, N_r$, and the vehicle's coordinates are $\begin{bmatrix} X & Y & \psi \end{bmatrix}^T$, we want to find out the index corresponding to the closest point to the vehicle,

$$p = \underset{i=0,1,\ldots,N_r}{\arg\min} \quad (X_{r,i} - X)^2 + (Y_{r,i} - Y)^2 . \tag{5.8}$$

Then we can derive the vehicle's coordinates in Frenet frame as

$$s = s_{r,p}, \tag{5.9}$$

and

$$y_e = \sqrt{(X_{r,p} - X)^2 + (Y_{r,p} - Y)^2}. \tag{5.10}$$

For the angle coordinate, we first calculate the reference path's heading angle by the integration

$$\psi_{r,i} = \int\limits_0^{s_{r,i}} \kappa_r ds + \psi_0, \tag{5.11}$$

where ψ_0 is the initial heading angle of the reference path. This integration can be computed numerically using the trapezoidal method. Then the angle coordinate of the vehicle can be expressed as

$$\psi_e = \arctan \frac{\dot{Y}}{\dot{X}} - \psi_{r,p} \cong \psi - \psi_{r,p}, \tag{5.12}$$

where the approximation is valid if the side slip angle $\beta \cong 0$.

Finally, the vehicle speed is given by

$$v = \sqrt{\dot{X}^2 + \dot{Y}^2}. \tag{5.13}$$

5.3 Motion Planning Problem Definition

In this section, we define the cost function and various constraints, except for collision avoidance, for the motion planning problem. They will be used later to formulate the NMPC. The safety-critical collision avoidance constraint will be discussed in detail in the next section.

5.3.1 Objective Function

A quadratic objective function is defined to minimize the deviation from the reference path and control input.

$$J = \frac{1}{2} \int_{s_0}^{s_f} q_1 y_e^2 (s) + q_2 \psi_e^2 (s) + q_3 (v (s) - v_r (s))^2 + \|u (s)\|_R^2 \, ds, \tag{5.14}$$

where q_1, q_2, and q_3 are the weighting factors for lateral deviation, angular deviation, and velocity tracking error, respectively. R is the weighting matrix for control inputs. Note that the integral is defined form s_0 to s_f in the spatial domain instead of time domain making the prediction horizon equals $s_f - s_0$ in further distance.

5.3.2 Control Input Constraints

The inputs to the vehicle model are yaw rate ω and acceleration a. These two inputs are physically bounded by the maximum yaw moment and driving/braking force available to the vehicle, which depends on both the actuator capability and road friction. The constraints on actuator capability are expressed by box constraints

$$\omega_L \leq \omega\left(s\right) \leq \omega_U, \tag{5.15}$$

$$a_L \leq a\left(s\right) \leq a_U. \tag{5.16}$$

And the constraints due to road friction are represented by the friction circle

$$\left(v\left(s\right)\omega\left(s\right)\right)^2 + a\left(s\right)^2 \leq \left(\mu g\right)^2, \tag{5.17}$$

where μ is the road friction coefficient. This is a nonlinear constraint with control inputs and state coupled.

5.3.3 State Constraints

The vehicle's lateral deviation should stay within the road boundary. Thanks to the spatial formation, this can be expressed as simple box constraints

$$y_{e,L}\left(s\right) \leq y_e\left(s\right) \leq y_{e,U}\left(s\right), \tag{5.18}$$

where $y_{e,L}\left(s\right)$ and $y_{e,U}\left(s\right)$ are road boundaries which are functions of the spatial variable s obtained by curve fitting based on sensor data. If temporal formation is used instead, inequality (5.18) will be a nonlinear constraint since the variable s itself is a nonlinear function of time. Similarly, the yaw angle error is constrained by

$$\psi_{e,L}\left(s\right) \leq \psi_e\left(s\right) \leq \psi_{e,U}\left(s\right). \tag{5.19}$$

The vehicle speed limit is enforced by the constraint

$$v_L(s) \leq v(s) \leq v_U(s). \tag{5.20}$$

where v_L and v_U may vary with different road segments. Sometimes it is also useful to constrain the maximum lateral acceleration of the vehicle for driving comfort considerations

$$a_{y,L}(s) \leq v(s)\,\omega(s) \leq a_{y,U}(s). \tag{5.21}$$

Note that it is also an input/state coupled constraint.

5.3.4 Waypoint Constraints

The waypoint constraint enforces the vehicle passing by an intermediate point on the route at a particular time. This type of constraint is useful when there exists higher-level algorithm, e.g. A* or RRT, providing waypoints for the vehicle to follow. It also allows us to handle tasks with time-scheduling requirements, for example, autonomous intersection management. The spatial form enables easy formulation of waypoint constraints as t is now a function of the space coordinates. Recall Eq. (5.1), we can write the dynamics of time in the spatial domain as

$$t' := \frac{dt}{ds} = \frac{1}{\dot{s}} = \frac{1 - y_e \kappa(s)}{v \cos\psi_e}. \tag{5.22}$$

The time dynamics (5.22) is augmented to the point-mass model (5.7), resulting in an additional state t. Then the waypoint constraint can be expressed as

$$t_{wp,L} \leq t(s_{wp}) \leq t_{wp,U}, \tag{5.23}$$

$$t_{wp,L} \leq y_e(s_{wp}) \leq y_{wp,U}, \tag{5.24}$$

$$\psi_{wp,L} \leq \psi_e(s_{wp}) \leq \psi_{wp,U}, \tag{5.25}$$

$$v_{wp,L} \leq v(s_{wp}) \leq v_{wp,U}, \tag{5.26}$$

where $s_{wp} \in \mathcal{S}_{wp}$ is the spatial coordinate from a set of waypoints. Equations (5.23) and (5.24) respectively defines the constraints of time and lateral displacement of the waypoint. Equations (5.25) and (5.26) can be used to constrain the heading angle and speed of the vehicle when traversing waypoints, if necessary.

So far, we have presented the major constraints that motion planning might encounter, except for collision avoidance. In the next section, we will discuss how to formulate this constraint using control barrier function [114].

5.4 Collision Avoidance Constraints based on Control Barrier Function

We present a new formulation for collision avoidance in motion planning NMPC in this section. The constraint is based on the recently developed theory of control barrier function.

5.4.1 Conventional Formulation of Collision Avoidance Constraint

Collision avoidance constraints are used to prevent vehicles from crashing into surrounding obstacles. Under spatial formulation, the easiest way is to treat the obstacle blocking the lane by tightening the road boundary on one side, as shown in Fig. 5.2. In this way, collision avoidance becomes a very simple box constraint that can be absorbed into the road boundary constraint (5.18). However, it also raises the necessity of upper-level logic responsible for optimally assigning obstacles to either side of the road, which is far from a trivial task [115].

Without the need for additional supervisory logic, another common approach to modeling the collision avoidance constraint is to use distance constraints based on Euclidean norm

$$h\left(\xi\right) := \frac{\left(s - s_{obs}\right)^2}{\left(2l_1\right)^2} + \frac{\left(y_e - y_{e,obs}\right)^2}{\left(2l_2\right)^2} - 1 \geq 0, \tag{5.27}$$

where s_{obs} and $y_{e,obs}$ are the spatial coordinates of the obstacle, l_1 and l_2 are the geometric parameters of the constraint ellipse. Figure 5.3 illustrates this constraint visually. To see its connection to the previous formulation under spatial formulation, we rewrite constraint Eq.

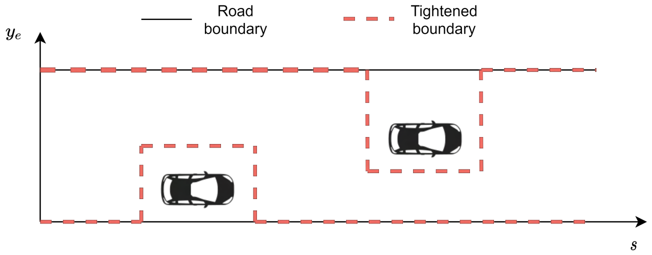


Figure 5.2: Collision avoidance by road boundary tightening

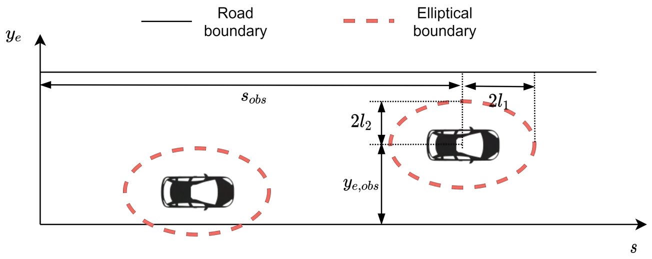


Figure 5.3: Collision avoidance using Euclidean norm

(5.27) in the following form given s is an independent clock state

$$y_e \in \begin{cases} [y_{e,L},\, y_{e,U}] & s \geq s_{obs} + 2l_1 \text{ or } s \leq s_{obs} - 2l_1 \\ \left[2l_2\sqrt{1 - \frac{(s - s_{obs})^2}{(2l_1)^2}} + y_{e,obs},\, y_{e,U}\right] \cup & \\ \quad \left[y_{e,L},\, y_{e,obs} - 2l_2\sqrt{1 - \frac{(s - s_{obs})^2}{(2l_1)^2}}\right] & s_{obs} - 2l_1 < s < s_{obs} + 2l_1 \end{cases} . \tag{5.28}$$

From Eq. (5.28), we can see that the Euclidean distance constraint is equivalent to lateral displacement constraints on y_e, which is similar to the road tightening method. However, it is also clear that it is a nonconvex constraint due to the fact that it creates two discrete feasible regions of y_e when $s_{obs} - 2l_1 \leq s \leq s_{obs} + 2l_1$. Nevertheless, this nonconvexity does not likely cause serious problems with optimization and control performance in practice, as local minimum is usually considered good enough for real-time control tasks.

We will use the Euclidean norm formulation in our study due to the lack of supervisory logic from the high-level decision making layer. The conventional way to embed collision avoidance constraints into NMPC is to simply enforce Eq. (5.27) on every discretized node of the prediction horizon, i.e. $h(\xi_k) \geq 0$ for $\forall k \in \{1, \ldots, N\}$. However, under this formulation, the collision avoidance constraints are always inactive if the obstacles are beyond the vehicle's forward reachable set along the prediction horizon. In other words, safety is only considered within the prediction horizon. On the other hand, it is crucial for vehicles act proactively when approaching to obstacles in some situations such as driving at high speed. This means a long horizon is required so that the NMPC is able to take the faraway obstacles into account, which adversely increases the computational load of optimization.

Motivated by the aforementioned reason, we will explore a new collision avoidance constraint formulation using control barrier function theory, which guarantees the system's safety in the sense of set invariance.

5.4.2 Control Barrier Function for Safety Critical System

Control barrier function (CBF) was firstly defined in [116]. A modern version of CBF is given in [114], and later refined in [117]. A nice overview of the theory and applications of CBF is given in [118]. For completeness, the theory of CBF is outlined here, which follows closely to those given in [118]. The only major difference is that CBF is stated in spatial domain in our study whereas the derivation in the reference is presented in the time domain.

We consider a nonlinear affine control system

$$\xi' = f\left(\xi\right) + g\left(\xi\right)u, \tag{5.29}$$

where $\xi \in \mathcal{D} \subset \mathbb{R}^n$ is the system state in some admissible set, functions $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are locally Lipschitz, and $u \in \mathcal{U} \subset \mathbb{R}^m$ is control input in some admissible set.

Let $u = k\left(\xi\right)$ be a feedback controller which is Lipschitz continuous, the closed-loop dynamics of Eq. (5.29) are

$$\xi' = f\left(\xi\right) + g\left(\xi\right)k\left(\xi\right). \tag{5.30}$$

Due to the locally Lipschitz condition, for any initial condition $\xi_0 \in D$ there exists a maximum interval of existence $I\left(\xi_0\right) = [s_0, s_f)$ such that $\xi\left(s\right)$ is the unique solution to Eq. (5.30) on $I\left(\xi_0\right)$.

The notion of safety is formalized in the context of enforcing invariance of a set. The safe set $\mathcal{C}$ is defined as the superlevel set of a continuously differentiable function $h : \mathcal{D} \subset \mathbb{R}^n \to \mathbb{R}$:

$$\mathcal{C} := \left\{\xi \in D \subset \mathbb{R}^n : h\left(\xi\right) \geq 0\right\}. \tag{5.31}$$

Definition 5.1. The set $\mathcal{C}$ is forward invariant if for every $\xi_0 \in \mathcal{C}$, $\xi\left(s\right) \in \mathcal{C}$ for $\xi\left(0\right) = \xi_0$ and all $s \in I\left(\xi_0\right)$. The system Eq. (5.30) is safe with respect to the set $\mathcal{C}$ if the set $\mathcal{C}$ is forward invariant.

Before defining CBF, we denote a function $\alpha : \mathbb{R} \to \mathbb{R}$ to be an extended class $\mathcal{K}_\infty$ function if it is strictly increasing and with $\alpha\left(0\right) = 0$. This allows us to define:

Definition 5.2. Let $C \subset \mathcal{D} \subset \mathbb{R}^n$ be the superlevel set of a continuously differentiable function $h : \mathcal{D} \to \mathbb{R}$, then h is a control barrier function (CBF) if there exists an extended class $\mathcal{K}_\infty$ function α such that for the control system Eq. (5.29):

$$\sup_{u \in \mathcal{U}} h' \left(\xi, u\right) := \sup_{u \in \mathcal{U}} \left(L_f h\left(\xi\right) + L_g h\left(\xi\right) u\right) \geq -\alpha\left(h\left(\xi\right)\right), \tag{5.32}$$

for all $x \in \mathcal{D}$.

Given a CBF h, we define the set consisting of all control values that satisfy:

$$K_{cbf}\left(\xi\right) := \left\{u \in \mathcal{U} : h'\left(\xi, u\right) + \alpha\left(h\left(\xi\right)\right) \geq 0\right\}. \tag{5.33}$$

The main result that relates the existence of a CBF and safety of the control system is given as

Theorem 5.1. *Let $\mathcal{C} \subset \mathcal{D} \subset \mathbb{R}^n$ be a set defined as the superlevel set of a continuously differentiable function $h : \mathcal{D} \subset \mathbb{R}^n \to \mathbb{R}$. If h is a control barrier function on $\mathcal{D}$ and $\frac{\partial h}{\partial \xi}\left(\xi\right) \neq 0$ for all $\xi \in \partial\mathcal{C}$, where $\partial\mathcal{C}$ denotes the boundary of $\mathcal{C}$, then any Lipschitz continuous controller $u\left(\xi\right) \in K_{cbf}\left(\xi\right)$ for the system (5.29) renders the set $\mathcal{C}$ safe.*

Theorem 5.1 enables intuitive controller synthesis based on optimization. Suppose we have a nominal feedback controller $u = k\left(\xi\right)$ for control system (5.29), which is not necessarily safe. Then a safety-guarantee controller can be constructed by the following minimum norm problem.

$$\begin{aligned} u\left(\xi_0\right) = \arg\min_{u \in \mathcal{U}} \quad & \frac{1}{2}\left\|u - k\left(\xi_0\right)\right\|_R^2 && \text{(CBF-OCP)} \\ \text{s.t.} \quad & L_f h\left(\xi_0\right) + L_g h\left(\xi_0\right) u + \alpha\left(h\left(\xi_0\right)\right) \geq 0 && (5.34) \end{aligned}$$

where ξ_0 is the current state measurement or estimation. Note that the constraint (5.34) is linear in control at a given system state. Therefore, in case $\mathcal{U}$ is a convex polytope, CBF-OCP becomes a QP, and can be solved quickly online.

However, constraint (5.34) can not be applied directly to our application. This is due to the fact that h defined in (5.27) has relative degree of 2 with respect to the vehicle model (5.7), so that u does not explicitly appears in h'. We state the following definition of relative degree given in [119].

Definition 5.3. The relative degree of a continuously differentiable function $h : \mathcal{D} \subset \mathbb{R}^n \to \mathbb{R}$ with respect to system (5.29) is the number of times we need to differentiate it along the dynamics of (5.29) until the control u explicitly shows.

It can be easily shown that the collision avoidance constraint (5.27) is of relative degree 2. Therefore, an alternative CBF formulation compatible with high relative degree systems must be used. In [120], an exponential control barrier function based on input-output linearization was first introduced for high relative degree systems. Then, a more generalized formulation was proposed in [119] in the form of high order CBF (HOCBF). In this study, we use the latter one and slightly modify it for the spatial domain.

For a continuously differentiable function h with relative degree of p, we define a series of functions $\psi_0 : D \to \mathbb{R}$, $\psi_1 : D \to \mathbb{R}$, $\psi_2 : D \to \mathbb{R}$, ..., $\psi_p : D \to \mathbb{R}$ such that

$$\psi_0 (\xi) := h (\xi) , \tag{5.35}$$

$$\psi_1 (\xi) := \psi_0' (\xi) + \alpha_1 (\psi_0 (\xi)) , \tag{5.36}$$

$$\psi_2 (\xi) := \psi_1' (\xi) + \alpha_2 (\psi_1 (\xi)) , \tag{5.37}$$

$$\vdots$$

$$\psi_p (\xi, u) := \psi_{p-1}' (\xi) + \alpha_p (\psi_{p-1} (\xi)) , \tag{5.38}$$

where $\alpha_1 (\cdot)$, $\alpha_2 (\cdot)$, ..., $\alpha_p (\cdot)$ denote extended class $\mathcal{K}_\infty$ functions.

A series of set $\mathcal{C}_1$, $\mathcal{C}_2$, ..., $\mathcal{C}_p$ associated with Eqns. (5.35)-(5.38) are defined:

$$\mathcal{C}_1 := \{\xi \in D : \psi_0 (\xi) \geq 0\} , \tag{5.39}$$

$$\mathcal{C}_2 := \{\xi \in D : \psi_1 (\xi) \geq 0\} , \tag{5.40}$$

$$\vdots$$

$$\mathcal{C}_p := \{\xi \in D : \psi_{p-1} (\xi) \geq 0\} . \tag{5.41}$$

Note that $\mathcal{C}_1 = \mathcal{C}$.

The HOCBF is defined using Eqns. (5.35)-(5.41).

Definition 5.4. A function $h : \mathcal{D} \subset \mathbb{R}^n \to \mathbb{R}$ is a high order control barrier function (HOCBF) of relative degree p for system (5.29) if there exist differentiable extended class $\mathcal{K}_\infty$ functions

α_1, α_2, …, α_p such that

$$\psi_p(\xi, u) := L_f^p h(\xi) + L_g L_f^{p-1} h(\xi) u + O\left(h^{(p-1)}(\xi)\right) + \alpha_p(\psi_{p-1}(\xi)) \geq 0, \tag{5.42}$$

for all $\xi \in \mathcal{C}_1 \cap \mathcal{C}_2 \cap \ldots \cap \mathcal{C}_p$. In (5.42), $O(\cdot)$ denotes the remaining Lie derivatives with degree less than or equal to $p-1$.

The set of control values that satisfy (5.42) is defined as

$$K_{hocbf}(\xi) := \{u \in \mathcal{U} : \psi_p(\xi, u) \geq 0\}. \tag{5.43}$$

We are now ready to give the following Theorem relating HOCBF to control system safety.

Theorem 5.2. *Given a HOCBF h from Def. 5.4 with the associated sets $\mathcal{C}_1$, $\mathcal{C}_2$, …, $\mathcal{C}_p$, if $\xi(s_0) \in \mathcal{C}_1 \cap \mathcal{C}_2 \cap \ldots \cap \mathcal{C}_p$, then any Lipschitz continuous controller $u(\xi) \in K_{hocbf}(\xi)$ renders the safe set $\mathcal{C}$ forward invariant for system (5.29).*

Theorem 5.2 allow us to synthesize an optimization based controller certifying the safety of high relative degree control system (5.29)

$$\begin{aligned} u(\xi_0) = \arg\min_{u \in \mathcal{U}} \quad & \frac{1}{2}\left\|u - k(\xi_0)\right\|^2 && \text{(HOCBF-QP)} \\ \text{s.t.} \quad & L_f^p h(\xi_0) + L_g L_f^{p-1} h(\xi_0) u + O\left(h^{(p-1)}(\xi_0)\right) + \alpha_p(\psi_{p-1}(\xi_0)) \geq 0 && (5.44) \end{aligned}$$

where constraint (5.44) is again linear in u.

5.4.3 HOCBF based Collision Avoidance Constraint for NMPC

The optimization based controller CBF-OCP or HOCBF-QP are often used together with control Lyapunov function to achieve stabilizing control with formal safety guarantee [117]. In these implementations, CBF or HOCBF is enforced only at current control step using state measurement ξ_0. This can potentially lead to over greedy control actions resulting in poor performance or infeasibility. On the contrary, NMPC optimizes the control actions over the whole future prediction horizon, which generally achieves long-term optimal performance.

Here we integrate HOCBF constraint to NMPC framework in a multi-step fashion, so that collision avoidance is certified not only at the current control step but in future steps.

We rewrite the definition of h here for convenience

$$h(\xi) := \frac{(s - s_{obs})^2}{(2l_1)^2} + \frac{(y_e - y_{e,obs})^2}{(2l_2)^2} - 1, \tag{5.45}$$

and the safe set is defined as

$$\mathcal{C} := \{\xi \in \mathbb{R}^n : h(\xi) \geq 0\}. \tag{5.46}$$

$h(\xi)$ is of relative degree 2 when we differentiate it along the vehicle dynamics. This means we need to define two extended class $\mathcal{K}_\infty$ functions $\alpha_1(\cdot)$ and $\alpha_2(\cdot)$ for HOCBF formulation. For the sake of simplicity, linear functions are adopted, which basically makes HOCBF become as the exponential CBF

$$\alpha_1(\psi_0) := \lambda_1 \psi_0, \tag{5.47}$$

$$\alpha_2(\psi_1) := \lambda_2 \psi_1, \tag{5.48}$$

where λ_1 and λ_2 are tuning parameters whose physical meaning will be discussed later. Furthermore, we assume that $\lambda_1 = \lambda_2 := \lambda$ for simplicity.

We rewrite the vehicle dynamics (5.7) affine in control.

$$\frac{d}{ds}\underbrace{\begin{bmatrix} y_e \\ \psi_e \\ v \\ s \\ t \end{bmatrix}}_{\xi} = \underbrace{\begin{bmatrix} \frac{v\sin\psi_e(1-y_e\kappa)}{v\cos\psi_e} \\ -\kappa \\ 0 \\ 1 \\ \frac{1-y_e\kappa}{v\cos\psi_e} \end{bmatrix}}_{f(\xi)} + \underbrace{\begin{bmatrix} 0 & 0 \\ \frac{1-y_e\kappa}{v\cos\psi_e} & 0 \\ 0 & \frac{1-y_e\kappa}{v\cos\psi_e} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}}_{g(\xi)} \underbrace{\begin{bmatrix} \omega \\ a \end{bmatrix}}_{u} \tag{5.49}$$

With Eqns. (5.47), (5.48) and (5.49), HOCBF then can be derived as

$$\psi_2(\xi) = L_f^2 h(\xi) + L_g L_f h(\xi) u + 2\lambda h'(\xi) + \lambda^2 h(\xi). \tag{5.50}$$

Using Theorem 5.2, the collision avoidance constraint is expressed as

$$L_f^2 h(\xi) + L_g L_f h(\xi) u + 2\lambda h'(\xi) + \lambda^2 h(\xi) \geq 0. \tag{5.51}$$

Unlike the single-point constraint used in HOCBF-QP, we consider (5.51) over multiple steps within prediction horizon.

$$L_f^2 h\left(\xi_k\right) + L_g L_f h\left(\xi_k\right) u_k + 2\lambda h'\left(\xi_k\right) + \lambda^2 h\left(\xi_k\right) \geq 0, \text{ for } k = 0, \ldots, N_{cbf}, \tag{5.52}$$

where $N_{cbf} \in \{0, \ldots, N\}$ is the length of constraint horizon, which can be different from the prediction horizon N. When $N_{cbf} = 0$, constraint (5.52) is linear in control given that ξ_0 is known, which can be handled by the optimization routine trivially. Whereas any constraint beyond the first step results in a state-input coupled path constraint requiring more computational effort. Therefore, N_{cbf} should be chosen carefully to balance control performance and computational complexity.

5.4.4 HOCBF with Vehicle Speed Scheduling

We have seen how space based NMPC facilitates simpler constraint formulations for motion planning. However, spatial formulation also creates an issue for motion planning due to the fixed prediction horizon in space. Figure 5.4 depicts the difference in prediction horizons of spatial and temporal formations at different vehicle speeds. In Fig. 5.4, h_s and t_s are respectively the sample distance and sample time of spatial and temporal formations, which are both constant values. In the case of time-based formulation, the total prediction length grows with higher vehicle speed. This behavior is similar to what a human driver does, and it is crucial for the stability and feasibility of collision avoidance at high speed. In the case of space based formulation, however, the total prediction length remains constant regardless of vehicle's speed. Therefore, if the controller is tuned for low speed, it might have difficulty avoiding obstacles when the vehicle is driving at a higher speed. While if tuned for high speed, it will have an unnecessary long prediction horizon at low speed.

A naive remedy is to adjust the sample distance h_s or the number of prediction steps N on the fly according to vehicle speed. However, this requires reformulating the structure of OCP in real-time, which not only creates additional computation overhead but also undermines the effectiveness of warm-starting.

The utilization of HOCBF allows us to propose a more elegant solution to this issue. Instead of changing the problem structure, we simply adapt the value of the hyperparameter λ in HOCBF according to vehicle speed. To see the effect of λ, let us write down the definitions of ψ_0, ψ_1, and ψ_2

$$\psi_0 = h, \tag{5.53}$$

$$\psi_1 = \psi_0' + \lambda\psi_0, \tag{5.54}$$

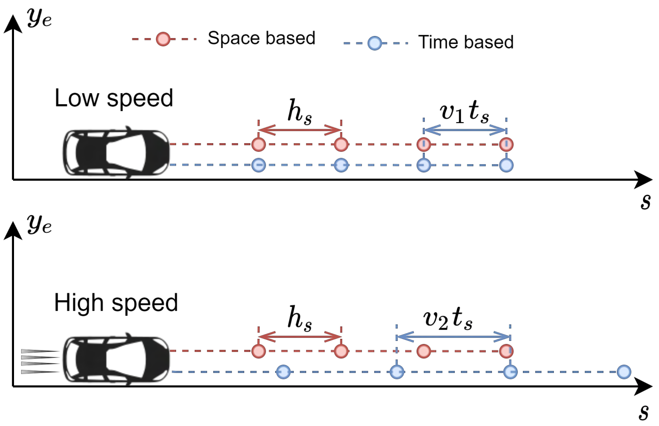


Figure 5.4: Comparison between space based and time based NMPC formulation

$$\psi_2 = \psi_1' + \lambda\psi_1. \tag{5.55}$$

Using above equations, HOCBF constraint $\psi_2 \geq 0$ can be expressed in terms of h

$$h'' + 2\lambda h' + \lambda^2 h \geq 0. \tag{5.56}$$

Equation (5.56) is a 2nd-order linear differential inequality of h. In order to find the lower bound of h, let $g := h' + \lambda h$ and (5.56) can be written as

$$g' + \lambda g \geq 0. \tag{5.57}$$

Then, by comparison lemma [121], the solution of g satisfies

$$g \geq Ae^{-\lambda s}, \tag{5.58}$$

where A is a constant value depending on initial condition. Now substituting h back to Eq. (5.58), we get

$$h' + \lambda h \geq Ae^{-\lambda s}. \tag{5.59}$$

Again, using comparison lemma, we obtain the bound of h as

$$h \geq (As + B)\, e^{-\lambda s} > 0, \tag{5.60}$$

where B is also a constant value depending on the initial condition. Equation (5.60) describes the lower bound of h, which decreases exponentially with the rate of λ. In other words, λ determines how fast a vehicle is allowed to approach an obstacle. Figure 5.5 shows intuitively

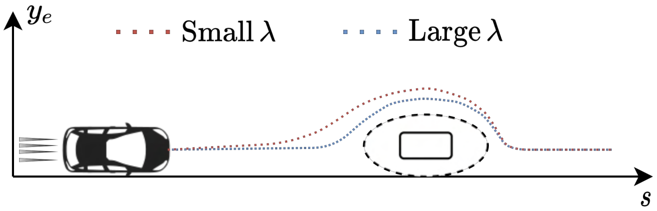


Figure 5.5: Effect of λ on collision avoidance

the effect of λ on collision avoidance trajectory planning. Large λ tolerates faster decreasing distance between vehicle and obstacle when the vehicle is far away, and starts evasion when the vehicle is relatively close to the obstacle. On the other hand, small λ confines the vehicle to approaching obstacles slowly, and also makes evasion earlier, resembling the effect of increasing prediction horizon.

With the above knowledge in mind, it becomes obvious how we can schedule λ in order to achieve consistent performance at different vehicle speeds. We want λ to be large at low speed, and gradually decrease at higher vehicle speeds. In practice, a set of λ is firstly tuned by simulations at several vehicle speeds, from low to high. Then a lookup table is generated based on linear interpolation using these values. During real-time control, λ is updated at each sample instant using the lookup table according to the current vehicle speed.

5.5 Summary of NMPC Formulation and Numerical Method

With all the ingredients at hand, we are ready to write down the NMPC formulation for vehicle motion planning.

$$
\begin{aligned}
\underset{\substack{\xi(s),u(s),\\ \sigma_f,\sigma_\xi,\sigma_{a_y},\sigma_{wp}}}{\text{minimize}} \quad & \frac{1}{2}\int_{s_0}^{s_f} q_1 y_e^2\left(s\right) + q_2 \psi_e^2\left(s\right) + q_3\left(v\left(s\right) - v_r\left(s\right)\right)^2 + \left\|u\left(s\right)\right\|_R^2 ds \\
& + \left\|\sigma_f\right\|_{q_f}^2 + \left\|\sigma_\xi\right\|_{q_\xi}^2 + \left\|\sigma_{a_y}\right\|_{q_{a_y}}^2 + \left\|\sigma_{wp}\right\|_{q_{wp}}^2 && (5.61) \\
\text{subject to} \quad & \xi\left(s_0\right) = \xi_0 && (5.62) \\
& \xi' = f\left(\xi\right) + g\left(\xi\right)u && (5.63) \\
& u_L \leq u\left(s\right) \leq u_U,\ \forall s \in \left[s_0, s_f\right] && (5.64) \\
& \left(v\left(s\right)\omega\left(s\right)\right)^2 + a\left(s\right)^2 \leq \mu g + \sigma_f,\ \forall s \in \left[s_0, s_f\right] && (5.65) \\
& \xi_L\left(s\right) - \sigma_\xi \leq \xi\left(s\right) \leq \xi_U\left(s\right) + \sigma_\xi,\ \forall s \in \left[s_0, s_f\right] && (5.66) \\
& a_{y,L}\left(s\right) - \sigma_{a_y} \leq v\left(s\right)\omega\left(s\right) \leq a_{y,U}\left(s\right) + \sigma_{a_y},\ \forall s \in \left[s_0, s_f\right] && (5.67) \\
& \xi_{wp,L} - \sigma_{wp} \leq \xi\left(s_{wp}\right) \leq \xi_{wp,U} + \sigma_{wp},\ \forall s_{wp} \in \mathcal{S}_{wp} && (5.68) \\
& L_f^2 h\left(\xi_k\right) + L_g L_f h\left(\xi_k\right)u_k + 2\lambda h'\left(\xi_k\right) + \lambda^2 h\left(\xi_k\right) \geq 0,\ \forall k \in \left\{0, \ldots, N_{cbf}\right\} && (5.69) \\
& \sigma_f \geq 0,\ \sigma_\xi \geq 0,\ \sigma_{a_y} \geq 0,\ \sigma_{wp} \geq 0. && (5.70)
\end{aligned}
$$

For feasibility consideration, constraints on friction circle, state, lateral acceleration and waypoint are softened by slack variables with 2-norm.

The NMPC presented above needs to be solved online in a receding horizon fashion. Real-time feasibility might be an issue due to nonlinear and nonconvex path constraints. In the previous chapter, we talked about how to solve trajectory tracking NMPC efficiently based on direct multiple shooting with RTI scheme. In principle, the same algorithm could be applied here for motion planning NMPC. However, there are two reasons why this algorithm is not the most suitable choice here.

(1) Multiple shooting transcription usually enforces path constraints only at discretization nodes,. Thus, the positions of the waypoints are also confined to these discrete points, i.e. $s_{wp} \in \{s_0, s_1, \ldots, s_N\}$, which puts great limitations on the high-level decision making algorithm if discretization is sparse.

(2) RTI scheme hinges on solving linearized OCP with only one QP iteration at each sample period. Therefore, constraints are not satisfied strictly at each sample period.

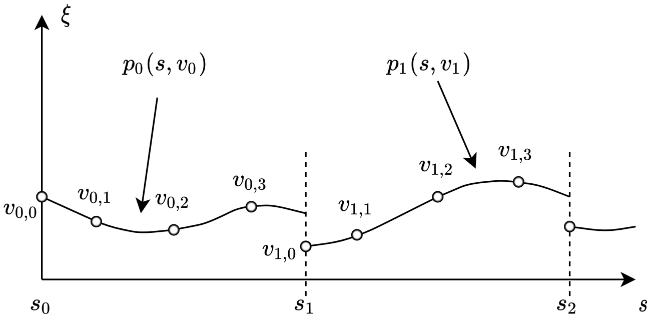


Figure 5.6: Direct collocation discretization

> Convergence to an exact solution is achieved if the sample rate is fast enough. However, for motion planning tasks with a relatively slow sample rate, exact solutions are preferable at every sample step so that the safety critical collision avoidance constraint is always ensured.

For the above two reasons, we choose to apply direct collocation transcription to the motion planning NMPC. A noticeable difference between direct collocation and direct multiple shooting is the discretization method. Direct collocation is based on the orthogonal collocation discretization method which approximates system state on each collocation interval by a dth-order polynomial $p_k\left(s, v_k\right)$ with $v_k \in \mathbb{R}^{n_x(d+1)}$ as coefficients [77]. Each polynomial consists of a set of d collocation points, whose positions are chosen according to Gauss-Legendre scheme in this study. Figure 5.6 shows the schematic of direct collocation discretization when $d = 3$.

The direct collocation allows us to possibly enforce the path constraints at arbitrary point s by simply evaluating the polynomial $p_k\left(s, v_k\right)$. Therefore, the waypoint can be chosen freely at any point within the prediction horizon, i.e. $s_{wp} \in [s_0, s_f]$. The NLP result form direct collocation is expressed as

$$\underset{\substack{v_k,\xi_k,u_k\\ \sigma\in\left\{\sigma_f,\sigma_\xi,\sigma_{a_y},\sigma_{wp}\right\}}}{\text{minimize}} \quad \sum_{k=0}^{N-1} l_k\left(v_k,\xi_k,u_k\right)+\left\|\sigma_f\right\|_{q_f}^2+\left\|\sigma_\xi\right\|_{q_\xi}^2+\left\|\sigma_{a_y}\right\|_{q_{a_y}}^2+\left\|\sigma_{wp}\right\|_{q_{wp}}^2 \tag{5.71}$$

$$\text{subject to} \quad \xi_0-q_0=0 \tag{5.72}$$

$$v_{k,0}-\xi_k=0,\ \forall k\in\{0,\ldots,N-1\} \tag{5.73}$$

$$p_k'\left(s_{k,1},v_k\right)-f\left(v_{k,1}\right)-g\left(v_{k,1}\right)u_k=0,\ \forall k\in\{0,\ldots,N-1\} \tag{5.74}$$

$$\vdots$$

$$p_k'\left(s_{k,d},v_k\right)-f\left(v_{k,d}\right)-g\left(v_{k,d}\right)u_k=0,\ \forall k\in\{0,\ldots,N-1\} \tag{5.75}$$

$$p_k\left(s_{k+1},v_k\right)-\xi_{k+1}=0,\ \forall k\in\{0,\ldots,N-1\} \tag{5.76}$$

$$\xi_{wp,L}-\sigma_{wp}\le p_m\left(s_{wp},v_m\right)\le\xi_{wp,U}+\sigma_{wp},\ \forall s_{wp}\in\mathcal{S}_{wp} \tag{5.77}$$

$$h\left(\xi_k,q_k,\sigma\right)\le 0,\ \forall k\in\{0,\ldots,N-1\} \tag{5.78}$$

where q_0 is the state measurement or estimation, $p_m\left(\cdot\right)$ is the collocation polynomial containing the waypoint. Equations (5.73)-(5.75) are collocation constraints. Equation (5.76) is continuity constraint. And Eq. (5.78) lumps all the other path constraints together.

Direct collocation typically results in a very large and sparse NLP with a dimension of $Nn_x\left(d+2\right)+\left(N-1\right)n_u$. They can be solved efficiently by structure-exploiting solvers. In this study, the open source nonlinear interior point based IPOPT solver is used [122].

5.6 Simulation Results

Implementation of the motion planning NMPC is done using Matlab and Casadi similar to that of tracking NMPC in the previous chapter. The model and controller parameters are given in Appendix A. In the simulations, the trajectory tracking NMPC developed in the previous chapter is used as a lower-level controller to track the planned trajectory given by the motion planning NMPC. For the sake of simplicity, we only consider static obstacles in the following simulations. The same control framework can be used for moving obstacles with only a few modifications, which we briefly discuss in Appendix C.

Figure 5.7 shows the top level schematic of the motion planning and tracking control system. The motion planning NMPC takes information and commands such as obstacle positions, waypoints, and road curvature from higher-level perception and decision making modules. Together with vehicle state feedback, it computes a reference trajectory for the

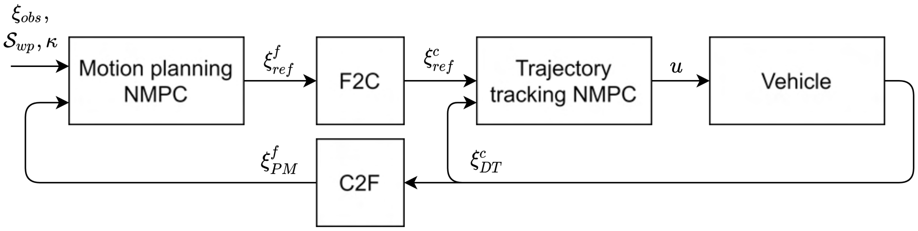


Figure 5.7: Motion planning and tracking control system

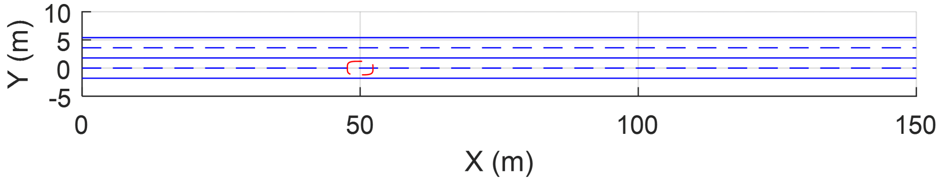


Figure 5.8: Road configuration for DLC. The blue solid line denotes lane condounary. The blue dashed line denotes lane center; red dashed line denotes obstacle.

tracking controller. The trajectory tracking NMPC computes steering and slip control inputs to the vehicle based on reference and state feedback. Note that the motion planning is formulated in the Frenet frame, while trajectory tracking is in Cartesian frame. Therefore, conversion blocks are needed to connect the planning and tracking loops.

5.6.1 Obstacle Avoidance with Double Lane Change

We consider the problem of avoiding a static obstacle sitting on a straight road segment with a double lane change maneuver. The road has two lanes where the obstacle is located on the bottom lane at 50m along the X direction. The lane width is 3.6m, and the obstacle is 4.6m long and 2.4m wide. The controlled vehicle enters the road in the bottom lane at $X = 0$. Figure 5.8 illustrates the road configuration of the simulation. We first simulate a vehicle with an entry speed of 70km/h. For comparison, we tuned two NMPC controllers, one with CBF based collision avoidance constraints and the other with the conventional distance constraint. Figure 5.9 shows the vehicle trajectory of the two controllers. We can see the two formulations are able to achieve very similar responses with proper tuning. However, there is a subtle difference in how they handle collision avoidance constraints. In Fig. 5.9 we denote the boundaries of two collision avoidance constraints, i.e. $h = 0$. It is shown that in order to achieve a similar trajectory, the constraint based on distance must be larger than that based on CBF. This is due to the fact that the trajectory based on distance constraint always tends to stay on the constraint boundary to achieve a smaller objective function value. While the trajectory based on CBF will leave a small buffer to the boundary since CBF defines

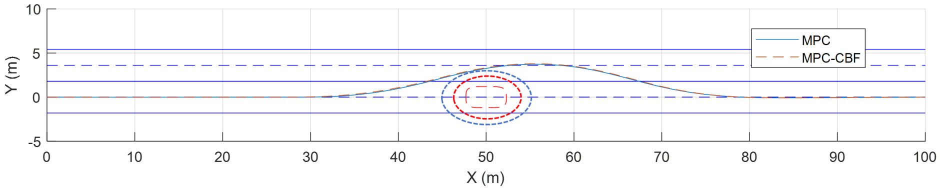


Figure 5.9: Collision avoidance at 70km/h ($\mu = 1$). The red dotted ellipse denotes CBF constraint. The blue dotted ellipse denotes distance constraint.

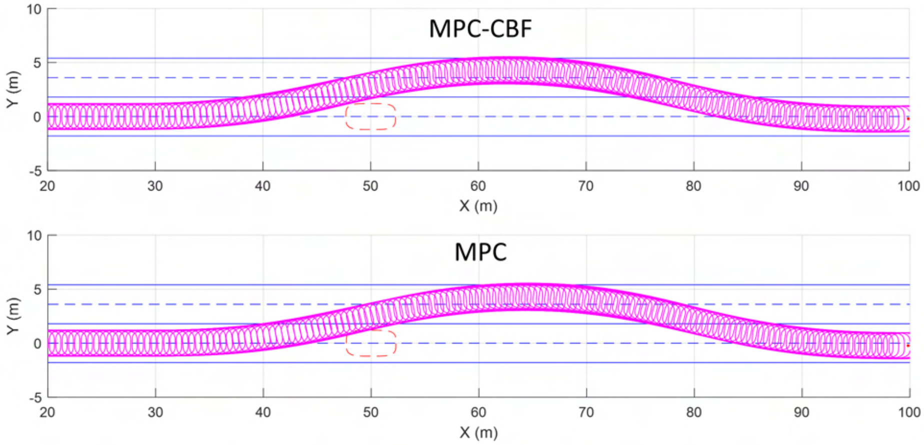


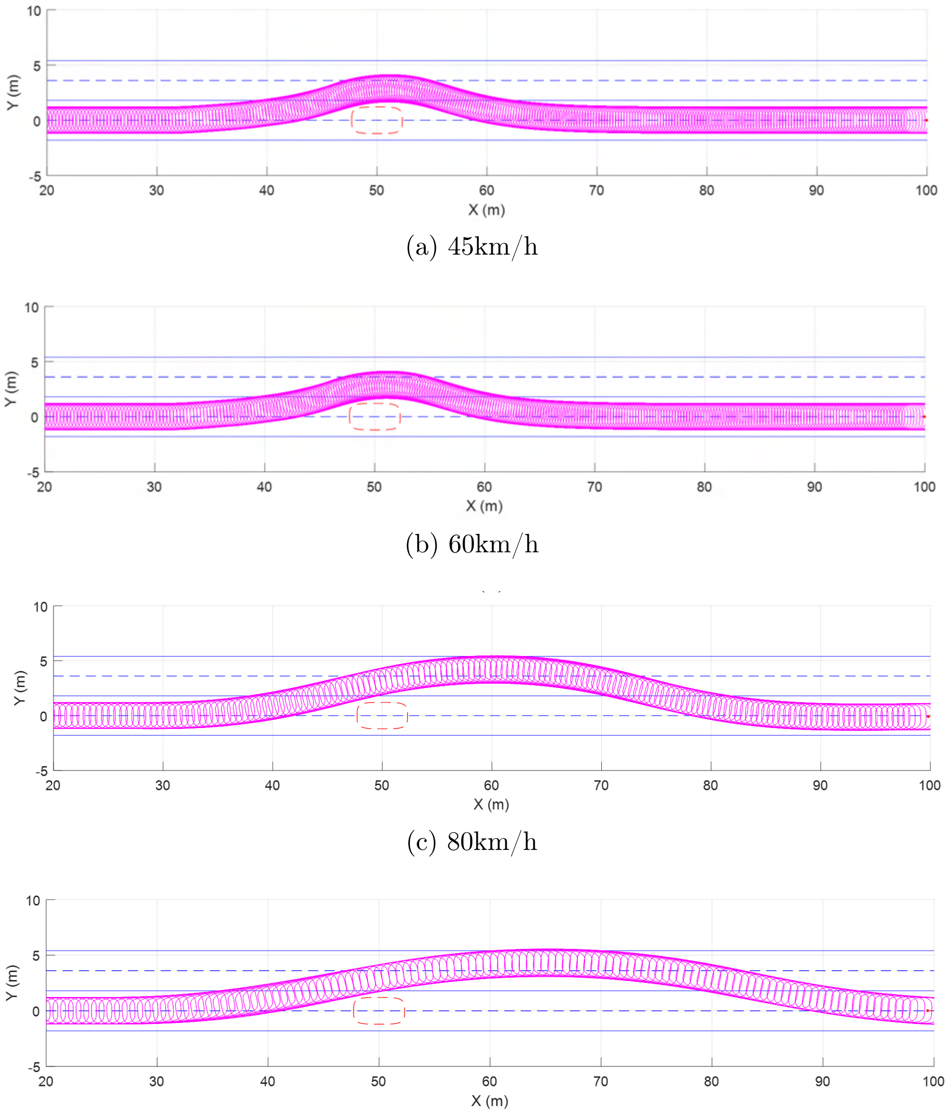
Figure 5.10: Collision avoidance at 85km/h ($\mu = 1$). The magenta rectangle denotes the shape of controlled vehicle.

not only the safe set but the way the vehicle is allowed to approach the set boundary by hyperparameter λ.

Using the same controller parameters, we now simulate the vehicle at an 85km/h entry speed. The results are shown in Fig. 5.10. In order to obtain feasible solutions, we have to formulate the collision avoidance constraint to soft constraint. We can see that the vehicle with CBF formulation is still able to avoid the obstacle, while the other one collides with the obstacle due to the soften constraint. This result shows that the CBF based collision avoidance is more robust against vehicle speed than the conventional distance-based constraint.

Figure 5.11 shows more simulations of NMPC with CBF at different vehicle speeds. It shows that without retuning the controller parameters, the vehicle is able to avoid the obstacle with a consistent distance at a wide range of speeds based on the scheduling of λ.

Next, we show the motion planning with additional constraints such as lateral acceleration



(a) 45km/h

(b) 60km/h

(c) 80km/h

(d) 100km/h

Figure 5.11: Collision avoidance at various vehicle speeds ($\mu = 1$)

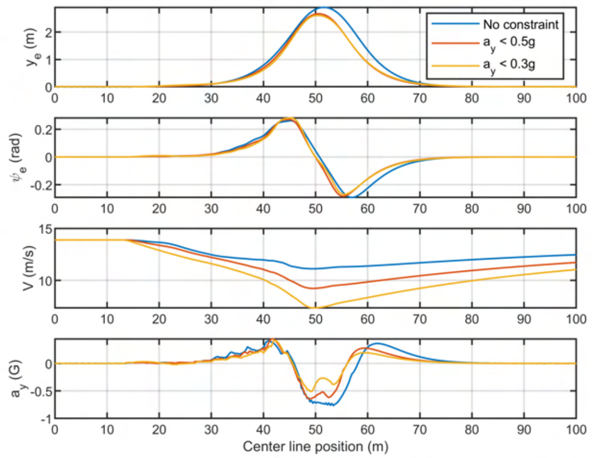


Figure 5.12: Collision avoidance with lateral acceleration constraints ($\mu = 1$)

and waypoints. Figure 5.12 shows the motion planning with different lateral acceleration constraints. The vehicle entry speed is 50km/h. We see that the motion planner is able to generate different trajectories according to the lateral acceleration requirements. It shows that the maximum lateral accelerations are reduced according to the constraints, although they are slightly over the limit due to the softened constraints. However, although the lateral acceleration does meet the limit, the controller tends to fulfill it by decelerating the vehicle speed, which is not very preferable in terms of the driver's comfort. To avoid an over decrease in speed, we simply increase the lower bound of the speed constraint. The lower bound of speed is set to be 80% of the entry speed. The result is shown in Fig. 5.13. Now, instead of reducing the vehicle speed, the motion planner actually modifies the shape of trajectory to have a smaller curvature such that the lateral acceleration limit is satisfied, thus really improving the driver's comfort.

We now show the motion planning with waypoint constraints. The two waypoints are set as $s_{wp} = \{30, 55\}$, $y_{e,wp} = \{3, 2.5\}$, and $t_{wp} = \{1.5, 2.7\}$. The simulation is presented in Fig. 5.14. We can see from the simulation that the motion planner generates a feasible trajectory that pass both the spatial and temporal points with only small deviations due to the soft constraint slackness. The vehicle speed is automatically scheduled by the motion planner to meet the temporal constraints.

5.6.2 Oval Circuit Tracking with Obstacles

We have shown the motion planning on a straight road with an obstacle. Now we will simulate the vehicle with the motion planner on an oval circuit with multiple obstacles. The

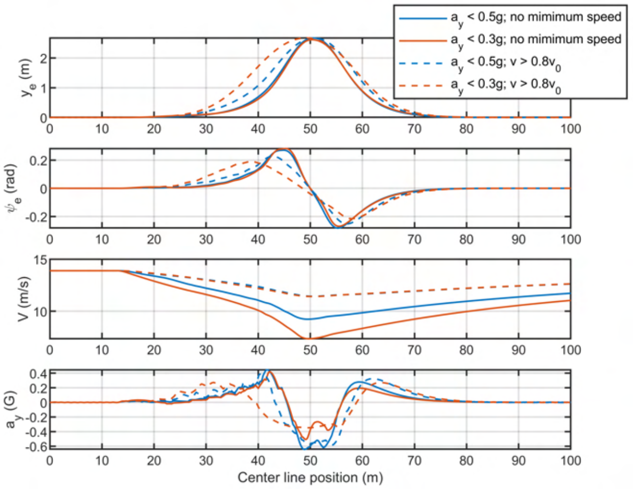
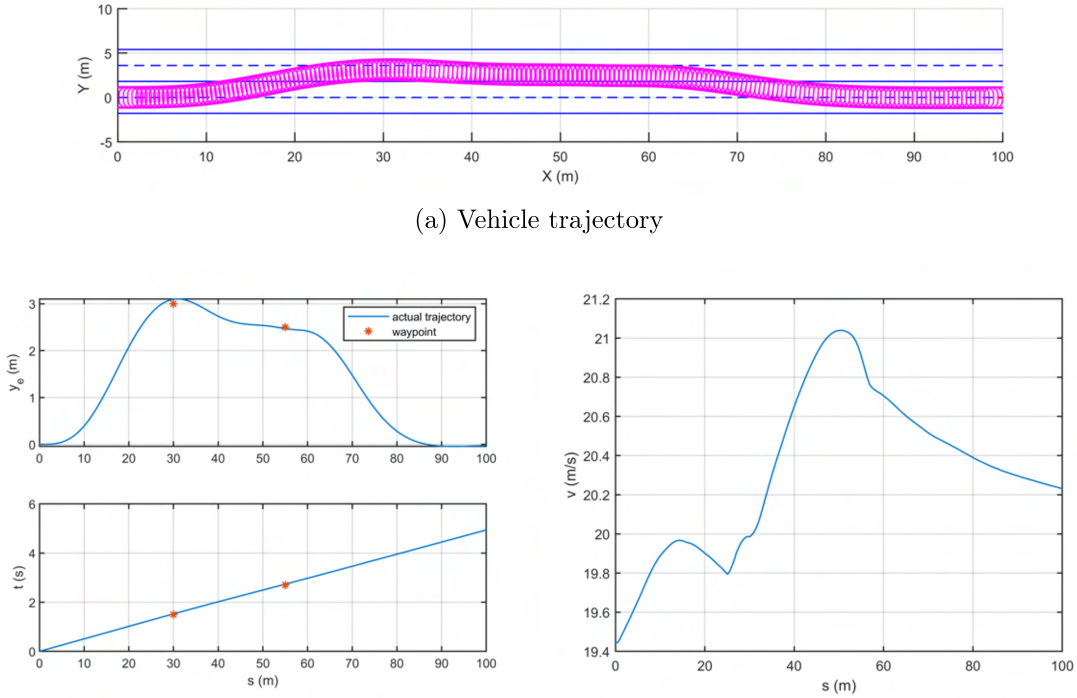


Figure 5.13: Collision avoidance with lateral acceleration constraints and additional speed constraints ($\mu = 1$)



(a) Vehicle trajectory

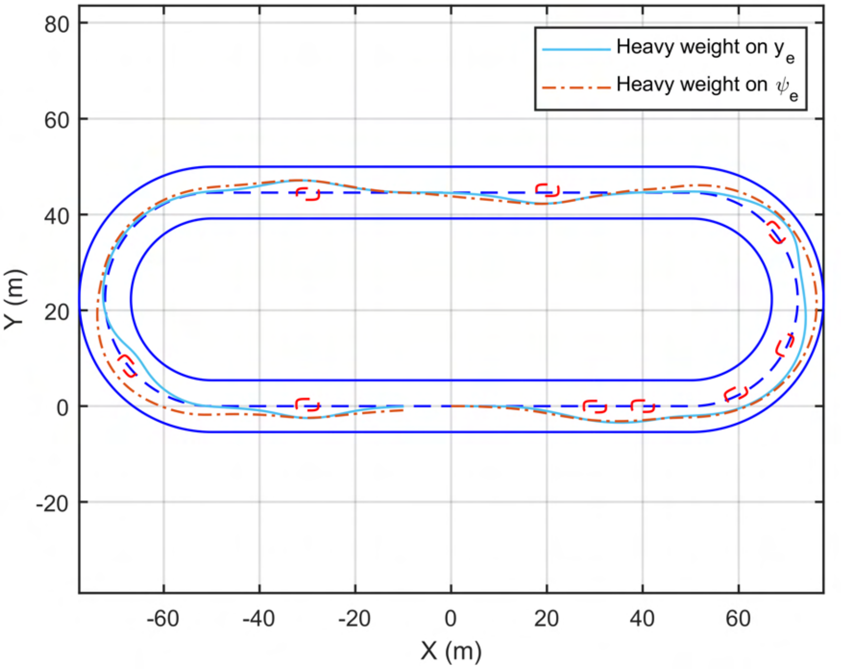
(b) Trajectory of lateral deviation and time

(c) Vehicle speed

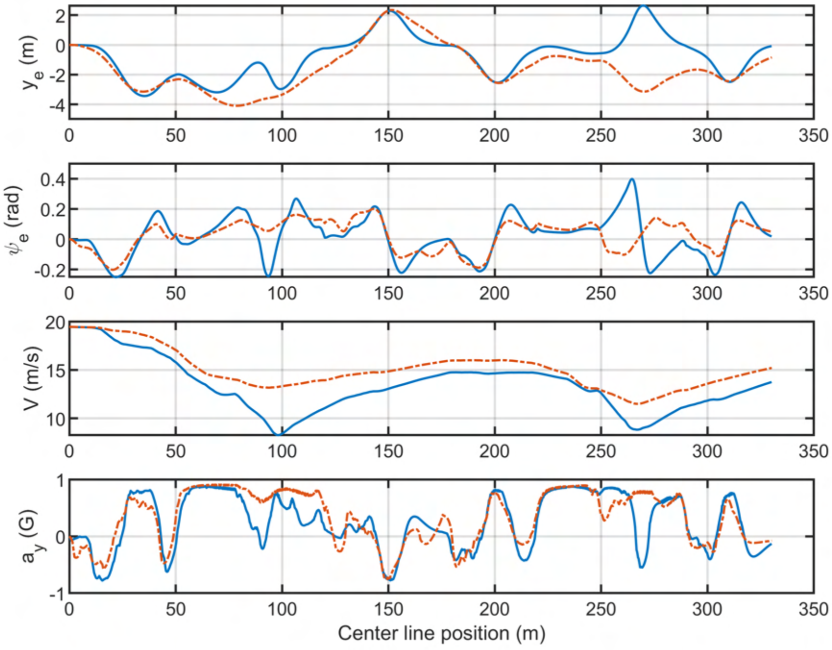
Figure 5.14: Motion planning with waypoint constraints ($\mu = 1$)

vehicle target speed is 70km/h. Figure 5.15 shows the simulation results with two different configurations of objective weighting. In Fig. 5.15, the blue solid line represents the trajectory based on heavy weighting on lateral deviation, while the red dash-dotted line is the result of the tuning that emphasizes yaw angle deviation. The values of weighting can be found in Appendix A. The rest of the controller parameters are the same in both cases. Lateral acceleration and waypoint constraints are not used in the simulations. We can see from the results that the vehicle is able to complete the circuit without collision in both setups. Both trajectories have maximum lateral accelerations of around $0.9g$, which demonstrates the capability of the proposed controller to handle the vehicle at the limit. However, the two set of trajectories have subtle differences. When the lateral deviation is weighted more, the vehicle always tries to return back to the center line whenever it is possible. Therefore, the lateral deviation is generally smaller, but the yaw angle deviation suffers relatively large fluctuations. It also shows a larger speed variation. On the other hand, the trajectory with larger weight on yaw angle deviation is much smoother with less speed variation despite bigger lateral deviation. Note that the two trajectories take different homotopies when passing the obstacle on the left half of the circuit, reflecting the nonconvex nature of the collision avoidance constraint.

We also investigated how the speed weighting affects the motion planning. Figure 5.16 shows the trajectories with two different speed weightings, those values can be found in Appendix A.. We see that the motion planner with large speed weighting tends to generate a trajectory with smaller curvatures in order to accommodate the constraints on the friction ellipse. While the controller with smaller weighting is able to make large curvatures by reducing the vehicle speed. It also shows a generally smaller lateral acceleration with a smaller weighting on vehicle speed.

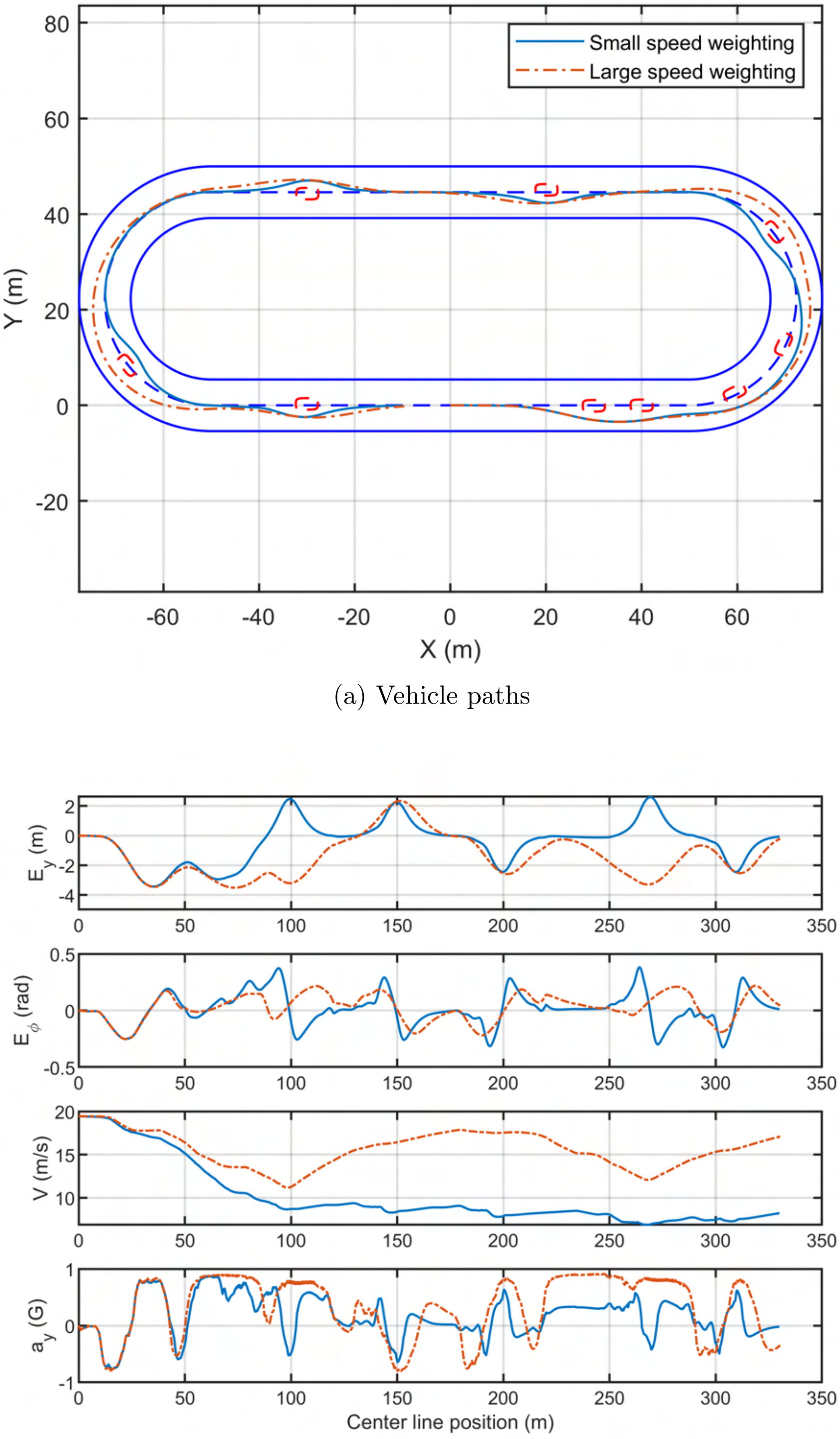


(a) Vehicle paths



(b) Vehicle states

Figure 5.15: Oval circuit tracking with heavy weighting on lateral deviation and heading angle deviation ($\mu = 1$)



(a) Vehicle paths

(b) Vehicle states

Figure 5.16: Oval circuit tracking with large and small weightings on speed ($\mu = 1$)

5.7 Conclusions

A motion planning algorithm based on the NMPC framework has been presented. The NMPC generates reference trajectories with numerical optimization based on the spatial domain point mass model and various input and state constraints. We incorporated CBF based collision avoidance constraints into the NMPC formulation so that it receives formal safety certification. Various simulations are conducted to evaluate the performance of the hierarchical motion planning and tracking control system. The simulation results have shown the great capability and flexibility of the proposed method for autonomous vehicle motion planning and control.

Chapter 6

Summary and Future Work

6.1 Summary

In this dissertation, the applications of model-based control design approaches for autonomous vehicles are presented. The four applications are divided into two categories according to the multi-layer schematic of the autonomous vehicle control system. For the low-level actuation layer, control development of EPAS and EMB systems are presented. While for the high-level motion control layer, trajectory tracking and motion planning controller design are discussed. The following summarizes the findings and contributions of these four cases.

6.1.1 EPAS Steering Feel Control

Drivers are sensitive to the steering feel of a vehicle and significant tuning efforts are placed to set the desired steering feel. Chapter 2 discusses a closed loop torque overlay control system to improve the steering feel of an electric power-assisted steering (EPAS) system. This system has a reference model, a rack force estimator, and a tracking controller. A target steering feel is generated from a reference model in which it uses an estimate of rack force determined by a Kalman filter based rack force estimator to reflect the actual vehicle operating conditions. A sliding mode controller is designed for tracking the target steering feel. The performance of the proposed control system is evaluated through simulation and a hardware-in-the-loop test. The simulation and experimental results show that the EPAS system with the proposed control method can successfully improve the steering feel under various operation conditions.

6.1.2 EMB Clamping Force Control

The EMB is a key component of the brake-by-wire (BBW) system, generating the clamping force between the brake pad and disk through an electric motor and mechanical transmission. Control of the clamping force has a significant impact on the brake performance. In Chapter 3, an EMB clamping force control system is developed that addresses several major challenges in practical implementation. Firstly, a nonlinear EMB model including the DC motor, planetary gear set, ball screw, and clamping force model is built. In particular, a novel clamping force model is introduced based on a linear transform of two polynomial functions. For the control system design, a clamping force estimation algorithm is proposed that requires only the existing measurements of motor torque, angular speed, and position. The contact point of the brake pad and the disk is also estimated using an internal model controller and a Rauch–Tung–Striebel smoother. The tracking controller is based on the disturbance observer structure, with a PI feedback controller and a zero-phase error tracking feedforward controller. Furthermore, a unified architecture is proposed such that a smooth transition between gap closing and clamping force tracking is realized. Finally, the performance of the entire control system under various conditions is evaluated based on simulation.

6.1.3 NMPC based Trajectory Tracking Control

Trajectory tracking is an essential component of the autonomous motion control system. The main objective of the controller is to control the vehicle to closely follow the reference trajectory by manipulating various control inputs. In Chapter 4, we proposed a NMPC controller for trajectory tracking. The subject vehicle is assumed to be equipped with 4WS and 4WID. A continuous-time NMPC is first formulated with a quadratic objective function balancing the tracking error and control effort. Constraints on control amplitude and rate are also embedded in the NMPC formulation. The standard formulation is extended by the frequency shaping technique to take into account different frequency domain requirements for actuators. In order to efficiently solve the NLP in real-time, an algorithm that incorporates the RTI scheme and control parameterization is proposed. The resulted QP problem is much smaller and its dimension is independent of the prediction horizon. Simulation results show that the NMPC effectively controls the vehicle to track the reference up to the handling limit under various road friction conditions. All the control inputs are smooth thanks to the frequency shaping. The control algorithm is also real-time feasible.

6.1.4 NMPC based Motion Planning

Motion planning is the high-level controller on the motion control layer of autonomous vehicles. It is responsible for generating a local trajectory as a reference to the tracking controller. The problem is formulated in the spatial domain, which entails simple representation of road boundary constraints. The trajectory is subjected to control input constraints presented by amplitude, rate, and friction circle. Different path constraints, such as road boundary, lateral acceleration, and waypoints are also taken into consideration by the planner. HOCBF is used to formulate the safety critical collision avoidance constraint. The numerical method is based on direct collocation transcription and an interior point solver, which enables efficient solving by sparsity exploitation and flexible choice of waypoint constraints. The motion planner is simulated together with the trajectory controller on a straight road and oval circuit with obstacles. The results show that the NMPC based motion planner successfully computes a dynamically feasible trajectory satisfying the control inputs and path constraints, and the tracking controller is able to control the vehicle to follow the trajectory closely.

6.2 Future Work

In this section we aim to give some directions for future work that are closely related to the topics in this dissertation.

6.2.1 EPAS Control

Chapter 2 discussed EPAS steering feel control which utilizes the reference model and rack force estimation. Based on this structure, a more flexible steering feel can be realized. The estimated rack force can be further modified according to the driver's preference and the vehicle's driving condition before it is used by the reference model. For example, the rack force estimation can be increased on low μ surface and decreased on high μ surface to achieve a more consistent steering feel. It can also be increased within a small steering angle and decreased within a larger steering angle to emphasize on-center steering feel while reducing steering effort within a large steering angle.

Furthermore, the same framework of steering feel control can be applied to steering position control in the future since the steering angle is used as the control variable. The compatibility of steering feel control and position control also enables a shared control strategy. For example, the control system can focus on the steering feel control when the vehicle is in the safe region, and gradually switch into position control when the vehicle approaches the unsafe region.

6.2.2 EMB Control

The clamping force estimation algorithm is presented in Chapter 3, and its performance is validated via experimental testing. However, in real-world operation, temperature variation and wear will affect the clamping force significantly. Therefore, mechanisms for adapting to the temperature and wear conditions should be developed. One possible way is to use a scaling factor to modify the the envelop of the clamping force. The values of the scaling factor must be calibrated using experimental tests. The machine learning based approaches are also promising for the modeling of nonlinear clamping force.

From the perspective of vehicle dynamics control, the EMB clamping force control system can be incorporated with wheel slip controller and simulated with vehicle model to validate its performance on ESC or torque vectoring. Different control algorithms can also be developed for the clamping force tracking. For example, H_∞ control or LPV control can be used to achieve robust performance against disturbance and uncertainty.

6.2.3 Trajectory Tracking Control

The trajectory tracking NMPC control structure can be easily extended to include more actuators, such as active suspension and active anti-roll bar to control the vehicle's vertical and roll motions. With more actuators considered, the computational demand will also increase. Therefore, the controller parameters should be tuned more carefully to balance the control performance and computation time. In particular, alternative control parameterization schemes can be explored, such as exponential parameterization, polynomial parameterization, and Fourier parameterization. Furthermore, based on the low-level actuators, different parameterizations can be used independently.

The deterministic NMPC framework can be robustified against model uncertainty and disturbance by using techniques such as tube MPC, stochastic MPC, or adaptive MPC. However, computational demand will become an issue for these methods. Although robust control within the NMPC framework might be challenging, it has been investigated in the context of other control methodologies, e.g. H_∞ control, μ synthesis, and sliding mode control. Therefore, the combination of NMPC control with other robust control techniques can be a promising approach. In particular, since the vehicle trajectory control is based on a over-actuated system, we can choose some actuators, such as steering, for trajectory tracking using the NMPC method, and other actuators, such as wheel slip, for disturbance rejection using robust control methods.

The frequency shaping method is only used for control input in our study. The same technique can be straightforwardly extended to the system state. For example, we can constrain the high frequency component of vehicle lateral and longitudinal acceleration to improve driving comfort. However, the constraints on state might affect the trajectory tracking performance, which could possibly cause safety issues since collision avoidance is not considered at the tracking level. Therefore, the controller tuning must be careful.

6.2.4 Motion Planning

The feasibility issue of NMPC based motion planning must be studied in the future. Due to the complexity of the environment and uncertainty of the measurement, a feasible solution might not always be found. Therefore, a feasibility condition for the motion planning NMPC should be studied. If a feasible solution cannot be found, a fail-safe mechanism must be implemented to avoid the collision.

Personalized motion planing is a possible direction for future research. Human drivers might have various preferences in terms of vehicle motion. For example, some may prefer to follow the preceding car rather than overtake. Personalization of motion planning is able to improve the driving experience and satisfy the driver's expectations. Learning-based approaches such as reinforcement learning techniques can be used for this purpose.

Only single vehicle motion planning is considered in this dissertation. To achieve better traffic safety and efficiency, multi-vehicle motion planning or cooperative driving should be investigated using frameworks such as distributed MPC and game theory. Formation control, also known as convoy control can also be applied to multi-vehicle motion planning problem. In formation control, the vehicles drive as a whole and change the formation structure adaptively according to the environment. It is particularly good at handling the scenarios such as lane-drop bottlenecks, on-ramp merging, and so on.

Finally, in order to implement the motion control system on a real vehicle, a companion vehicle state estimation pipeline must be incorporated alongside with the control system. Particle filtering and receding horizon estimation are two promising techniques that can be used for this purpose.

Appendix A

List of Model and Controller Parameters

The vehicle model and controller parameters that are used in Chapter 4 and 5 are listed as follows.

Table A.1: Vehicle parameters

Parameters	Meaning	Value
m	Vehicle mass	2010 [kg]
I	Vehicle yaw inertia	3300 [$^{kg}/_{m^2}$]
a	Distance from C.G. to front axle	1.05 [m]
b	Distance from C.G. to rear axle	1.45 [m]
c	Half track width	0.75 [m]
h	Height between C.G. and wheelbase	0.4 [m]
τ_L	Time constant of load transfer	0.01 [s]

Table A.2: Tire parameters

Road surface	Parameters	Longitudinal force	Lateral force
Asphalt	μ	1	1
	B	11.5	8.6
	C	1.6	1.1
	E	0.35	-1.2
Snowy	μ	0.4	0.4
	B	10	19
	C	2	0.55
	E	0.6	-2.1
Icy	μ	0.17	0.17
	B	31	28
	C	1.7	1.5
	E	0.7	-1.2

Table A.3: Parameters for the double lane change reference path

Parameters	d_{y1}	d_{y2}	d_{x1}	a_1	b_1	c_1	a_2	b_2	c_2
Value	3.7	3.7	25	0.096	19	1.2	0.1	71	1.2

Table A.4: NMPC tracking controller parameters

Parameters	Meaning	Value
T	Prediction horizon	1 [s]
T_s	Simple time	0.025 [s]
N_{cp}	Control parametrization nodes	$\{1, 11, 15\}$
Q	Reference tracking weighting	$diag\,(0.26, 300, 10, 3.2)$
R	Control inputs weighting	$diag\,(500, 500, 1250, 1250, 1250, 1250)$
$\delta_{f,U/L}$	Front steering angle limit	$\pm 30°$
$\dot{\delta}_{f,U/L}$	Front steering rate limit	$\pm 60°/s$
$\delta_{r,U/L}$	Rear steering angle limit	$\pm 10°$
$\dot{\delta}_{r,U/L}$	Rear steering rate limit	$\pm 30°/s$
$\dot{\lambda}_{U/L}$	Wheel slip rate limit	$\pm 1/s$
$R_{f,fws}$	Frequency shaping filter for front steering	$\frac{0.02s+1}{0.003s+1}$
$R_{f,rws}$	Frequency shaping filter for rear steering	$\frac{0.05s+1}{0.005s+1}$
$R_{f,slp}$	Frequency shaping filter for wheel slip	$\frac{0.01613s+1}{0.001613s+1}$

Table A.5: NMPC motion planning controller parameters

Parameters	Meaning	Value
M	Prediction horizon	30 [m]
h	Simple distance	1.5 [m]
Q_1	State weighting with large weight on y_e	$diag\,(1.2, 40, 0.01, 0, 0)$
Q_2	State weighting with large weight on ψ_e	$diag\,(0.4, 120, 0.01, 0, 0)$
Q_3	State weighting with large weight on v	$diag\,(1.2, 40, 0.05, 0, 0)$
Q_4	State weighting with small weight on v	$diag\,(1.2, 40, 0.001, 0, 0)$
R	Control inputs weighting	$diag\,(1, 0.1)$
$\omega_{U,L}$	Yaw rate limit	$\pm 15°/s$
$a_{U,L}$	Acceleration limit	$\pm 7m/s^2$
$\dot{\omega}_{U,L}$	Yaw rate limit	$\pm 30°/s^2$
$\dot{a}_{U,L}$	Acceleration rate limit	$\pm 15m/s^3$
v_i	Sample speed for λ scheduling	$\{0, 30, 50, 70, 85, 100, 150\}$ [kph]
λ_i	Value of λ at sample speed	$\{0.5, 0.5, 0.3, 0.3, 0.2, 0.15, 0.15\}$

Appendix B

Determining Rate Constraint of Wheel Slip

Since the wheel slip ratios are used as control inputs of the proposed trajectory tracking NMPC, their rate constraints should be chosen properly according to the psychical capability of the low-level actuators. The following presents how we can derive a realistic rate constraint on slip ratio.

The most deterministic factor that affects how fast slip ratio changes is the maximum brake torque available, which can be roughly calculated from tire force. Assuming pure slip condition, the maximum tire force is

$$F_{x,max} = \mu F_z, \tag{B.1}$$

where μ is road friction coefficient, and F_z is tire normal load. Considering load transfer, F_z on a single front wheel is approximately

$$F_z = F_{z0} + ma_x \frac{h}{2L}, \tag{B.2}$$

where F_{z0} is the static load, a_x is the deceleration, h is the C.G height, and L is the wheel base. Assuming full brake where $a_x = \mu g$ and $\mu = 1$, the maximum tire force obtained as

$$F_{x,max} = F_{z0} + mg\frac{h}{2L} \cong 7300N. \tag{B.3}$$

Using the relationship between brake torque and tire force, we have

$$T_{b,max} = F_{x,max} r_w \cong 2700Nm. \tag{B.4}$$

where $r_w = 0.37$m is the wheel radius. However, due to the tire relaxation length and brake system dynamics, $T_{b,max}$ cannot reached immediately. Therefore, a first-order lag with time

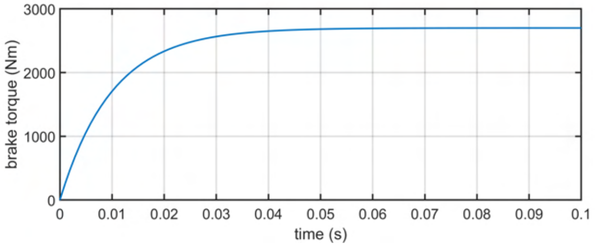


Figure B.1: Brake torque profile

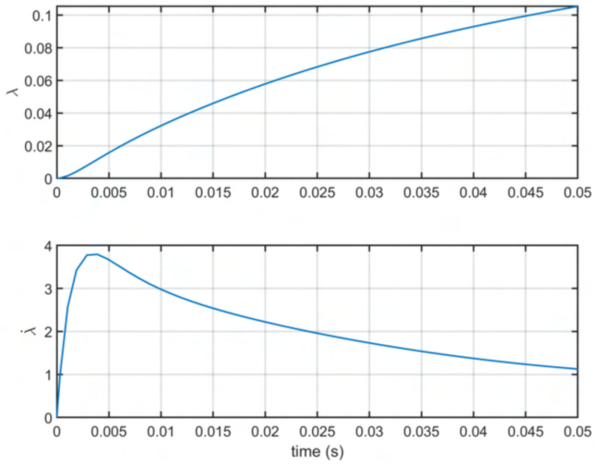


Figure B.2: Slip ratio response

constraint of 0.01s is used to generate the profile of $T_{b,max}$, as shown in Fig. B.1.

The brake torque trajectory in Fig. B.1 is then apply to a quarter car model to generate the slip ratio response, which is considered as the upper bound of achievable slip ratio by the actuator. The quarter car model is expressed as

$$m\dot{v} = -F_x, \tag{B.5}$$

$$J\dot{\omega} = r_w F_x - T_b \text{sgn}(\omega), \tag{B.6}$$

$$\lambda = \frac{v - \omega r_w}{v}, \tag{B.7}$$

where m is the mass of the quarter car, v is the longitudinal speed, and ω is the wheel rotational speed. Figure B.2 shows the response of the slip ratio. In Fig. B.2, we only show the response where λ is within the range of positive tire force slop, as the NMPC controller will not let the tire operate in negative slop. Neglecting the initial transition part, we can see that the $\dot{\lambda}$ approaches the maximum of 4s^{-1} and gradually reduces to 1s^{-1} at the end. Based on the simulation results, the rate constraint of λ are set as 1s^{-1}. For reference, a sliding mode based slip controller for passenger cars is designed in [69] and is able to reach $\lambda = 0.1$ in around 0.05 seconds, which means an equivalent rate of 2s^{-1}. Therefore, our choice should be totally achievable and pretty conservative.

Appendix C

Moving Obstacle Collision Avoidance Constraint

Let us first rewrite the HOCBF h for collision avoidance here

$$h\left(\xi\right) := \frac{\left(s - s_{obs}\right)^2}{\left(2l_1\right)^2} + \frac{\left(y_e - y_{e,obs}\right)^2}{\left(2l_2\right)^2} - 1, \tag{C.1}$$

where $\xi := \begin{bmatrix} y_e & \psi_e & v & s & t \end{bmatrix}^T$ is the state of vehicle model.

The constraint of HOCBF based collision avoidance is given as

$$L_f^2 h\left(\xi\right) + L_g L_f h\left(\xi\right) u + 2\lambda h'\left(\xi\right) + \lambda^2 h\left(\xi\right) \geq 0. \tag{C.2}$$

We can see that the constraint (C.2) is expressed in terms of the derivatives of h along the system dynamics. When obstacles are static, s_{obs} and $y_{e,obs}$ are constant values. Therefore, the derivatives can be calculated fairly straightforwardly. If obstacles are moving, however, s_{obs} and $y_{e,obs}$ become time varying, and thus their derivatives also become nonzero. In order to find obstacle's derivatives, a common way is to use a kinematic model that assumes constant speed and acceleration within the prediction horizon,

$$s_{obs}\left(t\right) = s_0 + v_s t + \frac{1}{2} a_s t^2, \tag{C.3}$$

$$y_{e,obs}\left(t\right) = y_{e,0} + v_y t + \frac{1}{2} a_y t^2, \tag{C.4}$$

where the initial position s_0 and $y_{e,0}$, speed v_s and v_y, and acceleration a_s and a_y are assumed known by perception system. Plug Eqns. (C.3) and (C.4) into (C.1), we obtain h for moving obstacle, which now depends on an additional state t. Now the derivatives of h can be calculated.

References

1. V. Ivanov, D. Savitski, and B. Shyrokau, “A Survey of Traction Control and Antilock Braking Systems of Full Electric Vehicles With Individually Controlled Electric Motors,” *IEEE Trans. Veh. Technol.*, vol. 64, no. 9, pp. 3878–3896, Sep. 2015. DOI: 10.1109/TVT.2014.2361860

2. L. Zhai, T. Sun, and J. Wang, “Electronic Stability Control Based on Motor Driving and Braking Torque Distribution for a Four In-Wheel Motor Drive Electric Vehicle,” *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 4726–4739, Jun. 2016. DOI: 10.1109/TVT.2016.2526663

3. M. Doumiati, O. Sename, L. Dugard, J.-J. Martinez-Molina, P. Gaspar, and Z. Szabo, “Integrated vehicle dynamics control via coordination of active front steering and rear braking,” *European Journal of Control*, vol. 19, no. 2, pp. 121–143, Mar. 2013. DOI: 10.1016/j.ejcon.2013.03.004

4. T. Gordon, M. Howell, and F. Brandao, “Integrated Control Methodologies for Road Vehicles,” *Vehicle System Dynamics*, vol. 40, no. 1-3, pp. 157–190, Jan. 2003. DOI: 10.1076/vesd.40.1.157.15877

5. K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner, “Three Decades of Driver Assistance Systems: Review and Future Perspectives,” *IEEE Intell. Transport. Syst. Mag.*, vol. 6, no. 4, pp. 6–22, 2014. DOI: 10.1109/MITS.2014.2336271

6. J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, Thao Dang, U. Franke, N. Appenrodt, C. G. Keller, E. Kaus, R. G. Herrtwich, C. Rabe, D. Pfeiffer, F. Lindner, F. Stein, F. Erbs, M. Enzweiler, C. Knoppel, J. Hipp, M. Haueis, M. Trepte, C. Brenk, A. Tamke, M. Ghanaat, M. Braun, A. Joos, H. Fritz, H. Mock, M. Hein, and E. Zeeb, “Making Bertha Drive—An Autonomous Journey on a Historic Route,” *IEEE Intell. Transport. Syst. Mag.*, vol. 6, no. 2, pp. 8–20, 2014. DOI: 10.1109/MITS.2014.2306552

7. W. Schwarting, J. Alonso-Mora, and D. Rus, “Planning and Decision-Making for Autonomous Vehicles,” *Annu. Rev. Control Robot. Auton. Syst.*, vol. 1, no. 1, pp. 187–210, May 2018. DOI: 10.1146/annurev-control-060117-105157

8. C. Katrakazas, M. Quddus, W.-H. Chen, and L. Deka, “Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research

directions," *Transportation Research Part C: Emerging Technologies*, vol. 60, pp. 416–442, Nov. 2015. DOI: 10.1016/j.trc.2015.09.011

9. Z. He and M. Gu, "Dynamic Research on Control Strategy of Electric Power Steering System," in *SAE 2012 World Congress & Exhibition*, Apr. 2012, pp. 2012–01–0212. DOI: 10.4271/2012-01-0212

10. I. A. Badiru, "Customer Focus in EPS Steering Feel Development," *SAE Int. J. Passeng. Cars - Mech. Syst.*, vol. 7, no. 3, pp. 1009–1015, Apr. 2014. DOI: 10.4271/2014-01-0148

11. K. D. Norman, "Objective Evaluation of On-Center Handling Performance," in *SAE International Congress and Exposition*, Feb. 1984, p. 840069. DOI: 10.4271/840069

12. M. Bröcker, "New control algorithms for steering feel improvements of an electric powered steering system with belt drive," *Vehicle System Dynamics*, vol. 44, no. sup1, pp. 759–769, Jan. 2006. DOI: 10.1080/00423110600885780

13. I. Yamazaki, I. Kushiro, and Y. Kunihiro, "Electronic Power Steering Compensating Control for Influence of Vehicle Dynamics on Steering Torque," *SAE Int. J. Passeng. Cars – Mech. Syst.*, vol. 2, no. 1, pp. 239–246, Apr. 2009. DOI: 10.4271/2009-01-0049

14. Tao Yang, "A New Control Framework of Electric Power Steering System Based on Admittance Control," *IEEE Trans. Contr. Syst. Technol.*, vol. 23, no. 2, pp. 762–769, Mar. 2015. DOI: 10.1109/TCST.2014.2325892

15. A. Marouf, M. Djemai, C. Sentouh, and P. Pudlo, "A New Control Strategy of an Electric-Power-Assisted Steering System," *IEEE Trans. Veh. Technol.*, vol. 61, no. 8, pp. 3574–3589, Oct. 2012. DOI: 10.1109/TVT.2012.2209689

16. C. C. de Wit, S. Guegan, and A. Richard, "Control design for an electro power steering system: Part I the reference model," in *2001 European Control Conference (ECC)*. Porto: IEEE, Sep. 2001, pp. 3611–3616. DOI: 10.23919/ECC.2001.7076494

17. M. Moradkhani, M. R. Hairi-Yazdi, and F. R. Salamsi, "Hinf loop-shaping control of an electric power steering system," in *2007 IEEE/ASME international conference on advanced intelligent mechatronics*. Zurich, Switzerland: IEEE, 2007, pp. 1–6. DOI: 10.1109/AIM.2007.4412513

18. D. Lee, K. Yi, S. Chang, B. Lee, and B. Jang, "Robust steering-assist torque control of electric-power-assisted-steering systems for target steering wheel torque tracking," *Mechatronics*, vol. 49, pp. 157–167, Feb. 2018. DOI: 10.1016/j.mechatronics.2017.12.007

19. M. Dohring, E. Lee, and W. Newman, "A load-dependent transmission friction model: theory and experiments," in *[1993] Proceedings IEEE International Conference on Robotics and Automation*. Atlanta, GA, USA: IEEE Comput. Soc. Press, 1993, pp. 430–436. DOI: 10.1109/ROBOT.1993.292210

20. F. Wilhelm, T. Tamura, R. Fuchs, and P. Mullhaupt, “Friction Compensation Control for Power Steering,” *IEEE Trans. Contr. Syst. Technol.*, vol. 24, no. 4, pp. 1354–1367, Jul. 2016. DOI: 10.1109/TCST.2015.2483561

21. K. Johanastrom and C. Canudas-de Wit, “Revisiting the LuGre friction model,” *IEEE Control Syst.*, vol. 28, no. 6, pp. 101–114, Dec. 2008. DOI: 10.1109/MCS.2008.929425

22. E. Garcia, P. Gonzalez de Santos, and C. Canudas de Wit, “Velocity Dependence in the Cyclic Friction Arising with Gears,” *The International Journal of Robotics Research*, vol. 21, no. 9, pp. 761–771, Sep. 2002. DOI: 10.1177/0278364902021009877

23. Y. Li, T. Shim, D. Wang, and T. Offerle, “Study on parameters affecting steering feel of column assist electric power steering,” *IJVD*, vol. 77, no. 3, p. 153, 2018. DOI: 10.1504/IJVD.2018.098941

24. Jose Velazquez Alcantar, “Improving Steering Feel in Electric Power Steering Systems: A Model Reference Feedback Controller Approach,” Master’s thesis, University of California, Davis, 2014.

25. A. Balachandran and J. C. Gerdes, “Designing Steering Feel for Steer-by-Wire Vehicles Using Objective Measures,” *IEEE/ASME Trans. Mechatron.*, vol. 20, no. 1, pp. 373–383, Feb. 2015. DOI: 10.1109/TMECH.2014.2324593

26. Y. Li, T. Shim, D. Wang, and T. Offerle, “Comparative Study of Rack Force Estimation for Electric Power Assist Steering System,” in *Volume 3: Vibration in Mechanical Systems; Modeling and Validation; Dynamic Systems and Control Education; Vibrations and Control of Systems; Modeling and Estimation for Vehicle Safety and Integrity; Modeling and Control of IC Engines and Aftertreatment Systems; Unmanned Aerial Vehicles (UAVs) and Their Applications; Dynamics and Control of Renewable Energy Systems; Energy Harvesting; Control of Smart Buildings and Microgrids; Energy Systems*. Tysons, Virginia, USA: American Society of Mechanical Engineers, Oct. 2017, p. V003T33A005. DOI: 10.1115/DSCC2017-5255

27. ——, “Enhancement of Steering Feel of Electric Power Assist Steering System Using Modeling Reference Control,” in *2018 Annual American Control Conference (ACC)*. Milwaukee, WI: IEEE, Jun. 2018, pp. 3257–3262. DOI: 10.23919/ACC.2018.8431923

28. T. Weiskircher, S. Fankem, and B. Ayalew, “Rack Force Estimation for Electric Power Steering,” in *Volume 3: 17th International Conference on Advanced Vehicle Technologies; 12th International Conference on Design Education; 8th Frontiers in Biomedical Devices*. Boston, Massachusetts, USA: American Society of Mechanical Engineers, Aug. 2015, p. V003T01A007. DOI: 10.1115/DETC2015-46228

29. Y. Li, T. Shim, D. Wang, and T. Offerle, “Effect of Load-Dependent Friction on the Estimation of Rack Force in Electric Power-Assisted Steering System,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 141, no. 11, p. 111005, Nov. 2019. DOI: 10.1115/1.4044181

30. C. Maron, T. Dieckmann, S. Hauck, and H. Prinzler, “Electromechanical Brake System: Actuator Control Development System,” in *SAE International Congress and Exposition*, Feb. 1997, p. 970814. DOI: 10.4271/970814

31. R. Schwarz, R. Isermann, J. Böhm, J. Nell, and P. Rieth, “Modeling and Control of an Electromechanical Disk Brake,” in *SAE International Congress & Exposition*, Feb. 1998, p. 980600. DOI: 10.4271/980600

32. C. F. Lee, “Brake force control and judder compensation of an automotive electromechanical brake,” Ph.D. dissertation, Department of Mechanical Engineering, The University of Melbourne, 2013.

33. C. Line, C. Manzie, and M. Good, “Control of an Electromechanical Brake for Automotive Brake-By-Wire Systems with an Adapted Motion Control Architecture,” in *SAE 2004 Automotive Dynamics, Stability & Controls Conference and Exhibition*, May 2004, pp. 2004–01–2050. DOI: 10.4271/2004-01-2050

34. E. Holweg, R. Klomp, J. Klaassens, and E. Lomonova, “Modeling and Inverse Model-Based Control of an Electro-Mechanical Brake Actuator,” *IFAC Proceedings Volumes*, vol. 33, no. 26, pp. 39–44, Sep. 2000. DOI: 10.1016/S1474-6670(17)39118-8

35. C. Line, C. Manzie, and M. Good, “ROBUST CONTROL OF AN AUTOMOTIVE ELECTROMECHANICAL BRAKE,” *IFAC Proceedings Volumes*, vol. 40, no. 10, pp. 579–586, 2007. DOI: 10.3182/20070820-3-US-2918.00078

36. G. Park and S. B. Choi, “Clamping force control based on dynamic model estimation for electromechanical brakes,” *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 232, no. 14, pp. 2000–2013, Dec. 2018. DOI: 10.1177/0954407017738394

37. C. Line, C. Manzie, and M. Good, “Electromechanical Brake Modeling and Control: From PI to MPC,” *IEEE Trans. Contr. Syst. Technol.*, vol. 16, no. 3, pp. 446–457, May 2008. DOI: 10.1109/TCST.2007.908200

38. C. F. Lee and C. M. Chris Line, “Explicit Nonlinear MPC of an Automotive Electromechanical Brake,” *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 10 758–10 763, 2008. DOI: 10.3182/20080706-5-KR-1001.01824

39. C. F. Lee and C. Manzie, “Near-time-optimal tracking controller design for an automotive electromechanical brake,” *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 226, no. 4, pp. 537–549, Apr. 2012. DOI: 10.1177/0959651811422164

40. S. Kwon, S. Lee, J. Lee, and D. Kum, “Accurate State Estimation for Electro-Mechanical Brake Systems,” *J. Electr. Eng. Technol.*, vol. 14, no. 2, pp. 889–896, Mar. 2019. DOI: 10.1007/s42835-019-00124-x

41. S. Formentin, G. Rallo, and S. M. Savaresi, "Data-Driven Clamping Force Control for an Electric Parking Brake Without Speed Measurement," in *2018 IEEE Conference on Decision and Control (CDC)*. Miami Beach, FL: IEEE, Dec. 2018, pp. 5128–5133. DOI: 10.1109/CDC.2018.8619170

42. R. Hoseinnezhad, A. Bab-Hadiashar, and T. Rocco, "Real-Time Clamp Force Measurement in Electromechanical Brake Calipers," *IEEE Trans. Veh. Technol.*, vol. 57, no. 2, pp. 770–777, Mar. 2008. DOI: 10.1109/TVT.2007.906374

43. C. Jo, S. Hwang, and H. Kim, "Clamping-Force Control for Electromechanical Brake," *IEEE Trans. Veh. Technol.*, vol. 59, no. 7, pp. 3205–3212, Sep. 2010. DOI: 10.1109/TVT.2010.2043696

44. Y. Li, T. Shim, D.-H. Shin, S. Lee, and S. Jin, "Effective Clamping Force Control for Electromechanical Brake System *," in *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. Boston, MA, USA: IEEE, Jul. 2020, pp. 643–648. DOI: 10.1109/AIM43001.2020.9158796

45. J. Brecht, A. Elvenkemper, J. Betten, U. Navrath, and J. B. Multhoff, "Elastic Properties of Friction Materials," in *21st Annual Brake Colloquium & Exhibition*, Oct. 2003, pp. 2003–01–3333. DOI: 10.4271/2003-01-3333

46. S. Saric, A. Bab-Hadiashar, and R. Hoseinnezhad, "Clamp-Force Estimation for a Brake-by-Wire System: A Sensor-Fusion Approach," *IEEE Trans. Veh. Technol.*, vol. 57, no. 2, pp. 778–786, Mar. 2008. DOI: 10.1109/TVT.2007.905251

47. G. Park, S. Choi, and D. Hyun, "Clamping force estimation based on hysteresis modeling for electro-mechanical brakes," *Int.J Automot. Technol.*, vol. 18, no. 5, pp. 883–890, Oct. 2017. DOI: 10.1007/s12239-017-0086-5

48. W.-F. Xie, "Sliding-Mode-Observer-Based Adaptive Control for Servo Actuator With Friction," *IEEE Trans. Ind. Electron.*, vol. 54, no. 3, pp. 1517–1527, Jun. 2007. DOI: 10.1109/TIE.2007.894718

49. R. Schwarz, R. Isermann, J. Böhm, J. Nell, and P. Rieth, "Clamping Force Estimation for a Brake-by-Wire Actuator," in *International Congress & Exposition*, Mar. 1999, pp. 1999–01–0482. DOI: 10.4271/1999-01-0482

50. S. Saric, "Development of an Intelligent Perception System for an Automotive Brake-by-Wire System," Ph.D. dissertation, Swinburne University of Technology, 2009.

51. Y. F. Fu, X. H. Hu, W. R. Wang, and Z. Ge, "Simulation and Experimental Study of a New Electromechanical Brake with Automatic Wear Adjustment Function," *Int.J Automot. Technol.*, vol. 21, no. 1, pp. 227–238, Feb. 2020. DOI: 10.1007/s12239-020-0022-y

52. Bohm, Jurgen, S. Martin, B. Jurgen, B. Karlheinz, and S. Peter, "System for controlling or adjusting an electromechanical brake," U.S. Patent 6 279 694.

53. B. Francis and W. Wonham, "The internal model principle of control theory," *Automatica*, vol. 12, no. 5, pp. 457–465, Sep. 1976. DOI: 10.1016/0005-1098(76)90006-6

54. D. Simon, *Optimal state estimation: Kalman, H [infinity] and nonlinear approaches.* Hoboken, N.J: Wiley-Interscience, 2006, oCLC: ocm64084871.

55. M. White, M. Tomizuka, and C. Smith, "Improved track following in magnetic disk drives using a disturbance observer," *IEEE/ASME Trans. Mechatron.*, vol. 5, no. 1, pp. 3–11, Mar. 2000. DOI: 10.1109/3516.828584

56. C. Kempf and S. Kobayashi, "Disturbance observer and feedforward design for a high-speed direct-drive positioning table," *IEEE Trans. Contr. Syst. Technol.*, vol. 7, no. 5, pp. 513–526, Sep. 1999. DOI: 10.1109/87.784416

57. Xu Chen and M. Tomizuka, "Optimal plant shaping for high bandwidth disturbance rejection in discrete disturbance observers," in *Proceedings of the 2010 American Control Conference*. Baltimore, MD: IEEE, Jun. 2010, pp. 2641–2646. DOI: 10.1109/ACC.2010.5531256

58. M. Tomizuka, "Zero Phase Error Tracking Algorithm for Digital Control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 109, no. 1, pp. 65–68, Mar. 1987. DOI: 10.1115/1.3143822

59. C. Kempf and S. Kobayashi, "Discrete-time disturbance observer design for systems with time delay," in *Proceedings of 4th IEEE International Workshop on Advanced Motion Control - AMC '96 - MIE*, vol. 1. Mie, Japan: IEEE, 1996, pp. 332–337. DOI: 10.1109/AMC.1996.509428

60. G. Raffo, G. Gomes, J. Normey-Rico, C. Kelber, and L. Becker, "A Predictive Controller for Autonomous Vehicle Path Tracking," *IEEE Trans. Intell. Transport. Syst.*, vol. 10, no. 1, pp. 92–102, Mar. 2009. DOI: 10.1109/TITS.2008.2011697

61. E. Kim, J. Kim, and M. Sunwoo, "Model predictive control strategy for smooth path tracking of autonomous vehicles with steering actuator dynamics," *Int.J Automot. Technol.*, vol. 15, no. 7, pp. 1155–1164, Dec. 2014. DOI: 10.1007/s12239-014-0120-9

62. P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive Active Steering Control for Autonomous Vehicle Systems," *IEEE Trans. Contr. Syst. Technol.*, vol. 15, no. 3, pp. 566–580, May 2007. DOI: 10.1109/TCST.2007.894653

63. P. Falcone, H. Eric Tseng, F. Borrelli, J. Asgari, and D. Hrovat, "MPC-based yaw and lateral stabilisation via active front steering and braking," *Vehicle System Dynamics*, vol. 46, no. sup1, pp. 611–628, Sep. 2008. DOI: 10.1080/00423110802018297

64. P. Falcone, M. Tufo, F. Borrelli, J. Asgari, and H. E. Tseng, "A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems," in *2007 46th IEEE Conference on Decision and Control*. New Orleans, LA, USA: IEEE, 2007, pp. 2980–2985. DOI: 10.1109/CDC.2007.4434137

65. N. Guo, X. Zhang, Y. Zou, B. Lenzo, and T. Zhang, “A Computationally Efficient Path-Following Control Strategy of Autonomous Electric Vehicles With Yaw Motion Stabilization,” *IEEE Trans. Transp. Electrific.*, vol. 6, no. 2, pp. 728–739, Jun. 2020. DOI: 10.1109/TTE.2020.2993862

66. Q. Cui, R. Ding, C. Wei, and B. Zhou, “Path-tracking and lateral stabilisation for autonomous vehicles by using the steering angle envelope,” *Vehicle System Dynamics*, pp. 1–25, Jun. 2020. DOI: 10.1080/00423114.2020.1776344

67. C. Xiang, H. Peng, W. Wang, L. Li, Q. An, and S. Cheng, “Path tracking coordinated control strategy for autonomous four in-wheel-motor independent-drive vehicles with consideration of lateral stability,” *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 235, no. 4, pp. 1023–1036, Mar. 2021. DOI: 10.1177/0954407020946884

68. R. de Castro, R. E. Araujo, and D. Freitas, “Wheel Slip Control of EVs Based on Sliding Mode Technique With Conditional Integrators,” *IEEE Trans. Ind. Electron.*, vol. 60, no. 8, pp. 3256–3271, Aug. 2013. DOI: 10.1109/TIE.2012.2202357

69. Taehyun Shim, Sehyun Chang, and Seok Lee, “Investigation of Sliding-Surface Design on the Performance of Sliding Mode Controller in Antilock Braking Systems,” *IEEE Trans. Veh. Technol.*, vol. 57, no. 2, pp. 747–759, Mar. 2008. DOI: 10.1109/TVT.2007.905391

70. H. B. Pacejka and I. Besselink, *Tire and vehicle dynamics*, 3rd ed. Oxford Waltham: Butterworth-Heinemann Elsevier, 2012.

71. C. S. Ahn, “Robust Estimation of Road Friction Coefficient for Vehicle Active Safety Systems,” Ph.D. dissertation, University of Michigan, Ann Arbor, Michigan, 2011.

72. M. Jonasson, J. Andreasson, B. Jacobson, and A. S. Trigell, “Global force potential of over-actuated vehicles,” *Vehicle System Dynamics*, vol. 48, no. 9, pp. 983–998, Sep. 2010. DOI: 10.1080/00423110903243232

73. H. G. Bock, M. M. Diehl, D. B. Leineweber, and J. P. Schlöder, “A Direct Multiple Shooting Method for Real-Time Optimization of Nonlinear DAE Processes,” in *Nonlinear Model Predictive Control*, F. Allgöwer and A. Zheng, Eds. Basel: Birkhäuser Basel, 2000, pp. 245–267. DOI: 10.1007/978-3-0348-8407-5$_1$4

74. N. K. Gupta, “Frequency-shaped cost functionals - Extension of linear-quadratic-Gaussian design methods,” *Journal of Guidance and Control*, vol. 3, no. 6, pp. 529–535, Nov. 1980. DOI: 10.2514/3.19722

75. H. Peng and M. Tomizuka, “Preview Control for Vehicle Lateral Guidance in Highway Automation,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 115, no. 4, pp. 679–686, Dec. 1993. DOI: 10.1115/1.2899196

76. R. Grandia, F. Farshidian, A. Dosovitskiy, R. Ranftl, and M. Hutter, “Frequency-Aware Model Predictive Control,” *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1517–1524, Apr. 2019. DOI: 10.1109/LRA.2019.2895882

77. L. T. Biegler, *Nonlinear programming: concepts, algorithms, and applications to chemical processes*, ser. MOS-SIAM series on optimization. Philadelphia: Society for Industrial and Applied Mathematics : Mathematical Programming Society, 2010.

78. D. P. Bertsekas, *Dynamic programming and optimal control. volume 1*, fourth edition ed. Belmont, Mass: Athena Scientific, 2017.

79. M. Grötschel, S. O. Krumke, and J. Rambau, Eds., *Online optimization of large scale systems*. Berlin ; New York: Springer, 2001.

80. M. Diehl, H. J. Ferreau, and N. Haverbeke, "Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation," in *Nonlinear Model Predictive Control*, M. Morari, M. Thoma, L. Magni, D. M. Raimondo, and F. Allgöwer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, vol. 384, pp. 391–417, series Title: Lecture Notes in Control and Information Sciences. DOI: 10.1007/978-3-642-01094-$1_3$2

81. G. Frison and M. Diehl, "HPIPM: a high-performance quadratic programming framework for model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6563–6569, 2020. DOI: 10.1016/j.ifacol.2020.12.073

82. E. M. Gertz and S. J. Wright, "Object-oriented software for quadratic programming," *ACM Trans. Math. Softw.*, vol. 29, no. 1, pp. 58–81, Mar. 2003. DOI: 10.1145/641876.641880

83. A. Zanelli, A. Domahidi, J. Jerez, and M. Morari, "FORCES NLP: an efficient implementation of interior-point methods for multistage nonlinear nonconvex programs," *International Journal of Control*, vol. 93, no. 1, pp. 13–29, Jan. 2020. DOI: 10.1080/00207179.2017.1316017

84. J. M. Maciejowski, *Predictive control: with constraints*. Harlow, England ; New York: Prentice Hall, 2002.

85. R. Cagienard, P. Grieder, E. Kerrigan, and M. Morari, "Move blocking strategies in receding horizon control," *Journal of Process Control*, vol. 17, no. 6, pp. 563–570, Jul. 2007. DOI: 10.1016/j.jprocont.2007.01.001

86. R. Quirynen, M. Vukov, and M. Diehl, "Multiple Shooting in a Microsecond," in *Multiple Shooting and Time Domain Decomposition Methods*, T. Carraro, M. Geiger, S. Körkel, and R. Rannacher, Eds. Cham: Springer International Publishing, 2015, vol. 9, pp. 183–201, series Title: Contributions in Mathematical and Computational Sciences. DOI: 10.1007/978-3-319-23321-5_7

87. D. Liao-McPherson, M. M. Nicotra, and I. Kolmanovsky, "Time-distributed optimization for real-time model predictive control: Stability, robustness, and constraint satisfaction," *Automatica*, vol. 117, p. 108973, Jul. 2020. DOI: 10.1016/j.automatica.2020.108973

88. T. Ohtsuka, "A continuation/GMRES method for fast computation of nonlinear receding horizon control," *Automatica*, vol. 40, no. 4, pp. 563–574, Apr. 2004. DOI: 10.1016/j.automatica.2003.11.005

89. K. Graichen and A. Kugi, "Stability and Incremental Improvement of Suboptimal MPC Without Terminal Constraints," *IEEE Trans. Automat. Contr.*, vol. 55, no. 11, pp. 2576–2580, Nov. 2010. DOI: 10.1109/TAC.2010.2057912

90. L. Grüne and J. Pannek, "Analysis of unconstrained NMPC schemes with incomplete optimization *," *IFAC Proceedings Volumes*, vol. 43, no. 14, pp. 238–243, Sep. 2010. DOI: 10.3182/20100901-3-IT-2016.00124

91. M. Diehl, H. G. Bock, and J. P. Schlöder, "A Real-Time Iteration Scheme for Nonlinear Optimization in Optimal Feedback Control," *SIAM J. Control Optim.*, vol. 43, no. 5, pp. 1714–1736, Jan. 2005. DOI: 10.1137/S0363012902400713

92. M. Diehl, H. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer, "Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations," *Journal of Process Control*, vol. 12, no. 4, pp. 577–585, Jun. 2002. DOI: 10.1016/S0959-1524(01)00023-3

93. J. Andersson, "A General-Purpose Software Framework for Dynamic Optimization," Ph.D. dissertation, Faculty of Engineering, KU Leuven, Leuven, Belgium, 2013.

94. M. Alamir, *Stabilization of Nonlinear Systems Using Receding-horizon Control Schemes*, ser. Lecture Notes in Control and Information Sciences. London: Springer London, 2006, vol. 339. DOI: 10.1007/978-1-84628-471-7

95. A. Murilo, M. Alamir, and D. Alberer, "A General NMPC Framework for a Diesel Engine Air Path," *International Journal of Control*, pp. 1–21, Apr. 2014. DOI: 10.1080/00207179.2014.905708

96. A. Murilo, R. Rodrigues, E. L. S. Teixeira, and M. M. D. Santos, "Design of a Parameterized Model Predictive Control for Electric Power Assisted Steering," *Control Engineering Practice*, vol. 90, pp. 331–341, Sep. 2019. DOI: 10.1016/j.conengprac.2019.07.010

97. M. Alamir, A. Murilo, R. Amari, P. Tona, R. Fürhapter, and P. Ortner, "On the Use of Parameterized NMPC in Real-time Automotive Control," in *Automotive Model Predictive Control*, M. Morari, M. Thoma, L. del Re, F. Allgöwer, L. Glielmo, C. Guardiola, and I. Kolmanovsky, Eds. London: Springer London, 2010, vol. 402, pp. 139–149, series Title: Lecture Notes in Control and Information Sciences. DOI: 10.1007/978-1-84996-071-7_9

98. K. M. Madhavan Rathai, M. Alamir, O. Sename, and R. Tang, "A Parameterized NMPC Scheme for Embedded Control of Semi-active Suspension System," *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 301–306, 2018. DOI: 10.1016/j.ifacol.2018.11.029

99. K. M. Madhavan Rathai, "Synthesis and real-time implementation of parameterized NMPC schemes for automotive semi-active suspension systems," Ph.D. dissertation, Université Grenoble Alpes, Grenoble, France, 2020.

100. J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “CasADi: a software framework for nonlinear optimization and optimal control,” *Math. Prog. Comp.*, vol. 11, no. 1, pp. 1–36, Mar. 2019. DOI: 10.1007/s12532-018-0139-4

101. H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, “qpOASES: a parametric active-set algorithm for quadratic programming,” *Math. Prog. Comp.*, vol. 6, no. 4, pp. 327–363, Dec. 2014. DOI: 10.1007/s12532-014-0071-1

102. S. Chang and T. J. Gordon, “A flexible hierarchical model-based control methodology for vehicle active safety systems,” *Vehicle System Dynamics*, vol. 46, no. sup1, pp. 63–75, Sep. 2008. DOI: 10.1080/00423110701882306

103. “Laboratory Test Procedure For FMVSS 126, Electronic Stability Control Systems,” U.S. Department Of Transportation, National Highway Traffic Safety Administration, Standard, 2008.

104. “ISO 3888-1:2018 Passenger cars — Test track for a severe lane-change manoeuvre — Part 1: Double lane-change,” International Organization for Standardization, Standard, 2018.

105. Y. Yoon, J. Shin, H. J. Kim, Y. Park, and S. Sastry, “Model-predictive active steering and obstacle avoidance for autonomous ground vehicles,” *Control Engineering Practice*, vol. 17, no. 7, pp. 741–750, Jul. 2009. DOI: 10.1016/j.conengprac.2008.12.001

106. S. Lee and H. E. Tseng, “Trajectory Planning with Shadow Trolleys for an Autonomous Vehicle on Bending Roads and Switchbacks,” in *2018 IEEE Intelligent Vehicles Symposium (IV)*. Changshu: IEEE, Jun. 2018, pp. 484–489. DOI: 10.1109/IVS.2018.8500498

107. J. Wurts, J. L. Stein, and T. Ersal, “Collision Imminent Steering at High Speed Using Nonlinear Model Predictive Control,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8278–8289, Aug. 2020. DOI: 10.1109/TVT.2020.2999612

108. S. Taherian, U. Montanaro, S. Dixit, and S. Fallah, “Integrated Trajectory Planning and Torque Vectoring for Autonomous Emergency Collision Avoidance,” in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. Auckland, New Zealand: IEEE, Oct. 2019, pp. 2714–2721. DOI: 10.1109/ITSC.2019.8917495

109. I. Batkovic, M. Zanon, M. Ali, and P. Falcone, “Real-Time Constrained Trajectory Planning and Vehicle Control for Proactive Autonomous Driving With Road Users,” in *2019 18th European Control Conference (ECC)*. Naples, Italy: IEEE, Jun. 2019, pp. 256–262. DOI: 10.23919/ECC.2019.8796099

110. R. Hajiloo, M. Abroshan, A. Khajepour, A. Kasaiezadeh, and S.-K. Chen, “Integrated Steering and Differential Braking for Emergency Collision Avoidance in Autonomous Vehicles,” *IEEE Trans. Intell. Transport. Syst.*, vol. 22, no. 5, pp. 3167–3178, May 2021. DOI: 10.1109/TITS.2020.2984210

111. R. Quirynen, K. Berntorp, K. Kambam, and S. Di Cairano, “Integrated Obstacle Detection and Avoidance in Motion Planning and Predictive Control of Autonomous Vehicles,” in *2020 American Control Conference (ACC)*. Denver, CO, USA: IEEE, Jul. 2020, pp. 1203–1208. DOI: 10.23919/ACC45564.2020.9147820

112. U. Rosolia, S. De Bruyne, and A. G. Alleyne, “Autonomous Vehicle Control: A Nonconvex Approach for Obstacle Avoidance,” *IEEE Trans. Contr. Syst. Technol.*, vol. 25, no. 2, pp. 469–484, Mar. 2017. DOI: 10.1109/TCST.2016.2569468

113. R. Verschueren, S. De Bruyne, M. Zanon, J. V. Frasch, and M. Diehl, “Towards time-optimal race car driving using nonlinear MPC in real-time,” in *53rd IEEE Conference on Decision and Control*. Los Angeles, CA, USA: IEEE, Dec. 2014, pp. 2505–2510. DOI: 10.1109/CDC.2014.7039771

114. A. D. Ames, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs with application to adaptive cruise control,” in *53rd IEEE Conference on Decision and Control*. Los Angeles, CA, USA: IEEE, Dec. 2014, pp. 6271–6278. DOI: 10.1109/CDC.2014.7040372

115. M. Graf Plessen, D. Bernardini, H. Esen, and A. Bemporad, “Spatial-Based Predictive Control and Geometric Corridor Planning for Adaptive Cruise Control Coupled With Obstacle Avoidance,” *IEEE Trans. Contr. Syst. Technol.*, vol. 26, no. 1, pp. 38–50, Jan. 2018. DOI: 10.1109/TCST.2017.2664722

116. P. Wieland and F. Allgöwer, “CONSTRUCTIVE SAFETY USING CONTROL BARRIER FUNCTIONS,” *IFAC Proceedings Volumes*, vol. 40, no. 12, pp. 462–467, 2007. DOI: 10.3182/20070822-3-ZA-2920.00076

117. A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, “Control Barrier Function Based Quadratic Programs for Safety Critical Systems,” *IEEE Trans. Automat. Contr.*, vol. 62, no. 8, pp. 3861–3876, Aug. 2017. DOI: 10.1109/TAC.2016.2638961

118. A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, “Control Barrier Functions: Theory and Applications,” in *2019 18th European Control Conference (ECC)*. Naples, Italy: IEEE, Jun. 2019, pp. 3420–3431. DOI: 10.23919/ECC.2019.8796030

119. W. Xiao and C. Belta, “Control Barrier Functions for Systems with High Relative Degree,” in *2019 IEEE 58th Conference on Decision and Control (CDC)*. Nice, France: IEEE, Dec. 2019, pp. 474–479. DOI: 10.1109/CDC40024.2019.9029455

120. Q. Nguyen and K. Sreenath, “Exponential Control Barrier Functions for enforcing high relative-degree safety-critical constraints,” in *2016 American Control Conference (ACC)*. Boston, MA, USA: IEEE, Jul. 2016, pp. 322–328. DOI: 10.1109/ACC.2016.7524935

121. H. K. Khalil, *Nonlinear systems*, 3rd ed. Upper Saddle River, N.J: Prentice Hall, 2002.

122. A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, no. 1, pp. 25–57, Mar. 2006. DOI: 10.1007/s10107-004-0559-y