SAFE MOTION CONTROL AND PLANNING FOR AUTONOMOUS RACING VEHICLES

by

ALIASGHAR ARAB

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Mechanical and Aerospace Engineering

Written under the direction of

Jingang Yi

and approved by

_____

_____

_____

_____

New Brunswick, New Jersey

May, 2021

ABSTRACT OF THE DISSERTATION

Safe Motion Control and Planning for Autonomous Racing Vehicles

By Aliasghar Arab

Dissertation Director:

Jingang Yi

In the future of the autonomous car industry saving lives might depend on more demanding maneuvers than what the average drivers know how to do. Professional race car drivers, as well as some skilled stunt drivers, are able to exteremely push the limits of a vehicle's capabilities and safety features which are useful to avoid hazardous situations. By understanding these human-inspired driving abilities, a motion planning and autonomous vehicle control system can be developed for enabling Aggressive Maneuvering as a Safety Feature (AMSF). Similar to existing and widely used vehicle safety features, such as anti-lock braking and electronic stability control, AMSF can be utilized to increase overall safety. AMSF system will push the limits of maneuverability and motion stability to enable the next generation of accident-free vehicle systems. The design of an AMSF requires advanced theoretical tools and algorithms to guarantee safety assurance under a dynamically changing environment. Current AMSF design methods incorporate a variety of elements such as analytical vehicle dynamics model-based control, machine learning-based methods to mimic expert human drivers, integrated physical model-based knowledge, and experience-based skills to enhance vehicle maneuverability with guaranteed motion stability.

The main goal of this dissertation focuses on the safety-guaranteed motion planner and controller design for AMSF. This dissertation proposes methods and algorithms that enable the integration of data-driven approaches based on machine learning techniques and physical model analysis that will achieve parity with and even exceed the driving skills of the most skilled drivers. To achieve these goals, several new modeling and control approaches are developed in this dissertation. First, in order to improve the physical model accuracy under a dynamic environment, Gaussian Process on Polynomial Basis (GPPB) method is proposed to learn from human expert driving data. A Sum Of Square (SOS) method is used to estimate the safety boundary of the nonlinear vehicle dynamics to enable the use of Nonlinear Model Predictive Controller (NMPC) for AMSF design. It is also shown that the safety of the aggressive maneuvering can be considered as a safety brier applied as constraints to the vehicle's motion similar to a control barrier function. The main rationale of these designs comes from observations that many physical models cannot effectively capture dynamic changes or the uncertainties of the vehicle/environment interactions, such as tire-road contact properties. Machine learning-based methods generally provide an effective means to obtain the system dynamics changes in real time and incorporate them into the control design.

One important feature of the above integrated GPPB and SOS approach lies in the guaranteed safety and stability under the control design for aggressive vehicle maneuvers. The motion safety assurance while tracking a planned trajectory has been successfully resolved by integrating the data-driven methods with physical model-based controllers under dynamic changing environments. Theoretical analyses are presented in the dissertation for the vehicle dynamics model and also extendable to other well-understood dynamics structure of the car-like robots. The Lyapunov stability method is used to show safety assurance of a learning model-based control framework. A scaled race car-like robot is used as an experimental testbed to demonstrate the proposed control of the AMSF design. The experimental results demonstrate superior agility and fast traveling time performance under the proposed control design than those under the physical model-based control design in literature.

Another important aspect of this dissertation focuses on the real time motion planning for AMSF design. Planning the trajectory for an autonomous vehicle for agile maneuvers in a dynamic environment is complex and challenging, especially when the autonomous vehicles tend to maximize the driving capabilities to achieve certain metrics such as traveling time, etc. The proposed motion planner in the dissertation takes advantage of the Sparse Stable Trees (SST), the Star Rapidly-exploring Random Tree (RRT^*) algorithm, and the NMPC design. The use of the sparsity property helps to reduce the computational burden of the RRT^* method by removing non-used nodes in each iteration and therefore to render the algorithm to converge to optimal path quickly. A heuristic quality function is used to guide the search to achieve faster convergence, and the NMPC is used for rewiring feasibility among nodes. The motion planner is tested experimentally and also compared with the existing benchmarks to demonstrate superior performance.

Finally, the last part of the dissertation discusses and proposes an extension of the above-mentioned methods to further incorporate the newly development of machine learning techniques. Another goal is to test Gaussian Process and Reinforcement Learnings for direct and indirect controller design. Our goal is to test the learning methods for both the off-line and the online applications and design stable adaptive rules for real time stability and safety assurance and performance enhancement. Taking advantage of policy search approaches alongside nonlinear programming methods gives strong optimization power for overall safety and agility in real time implementation. The works in this part will be experimentally validated and demonstrated using two different car-like robot platforms: one is racing car-like and the other is a scaled autonomous truck used for the minimum time lap and stunt maneuverings experiments.

Acknowledgements

From the endless opportunities that have been available to me as a graduate student at Rutgers University, without a doubt, the greatest opportunity of all has been the chance to work with my advisor Professor Jingang Yi. What inspires me most about Jingang, aside from his well-known traits like superior intelligence, humble, and professional attitude is his never-ending passion for understanding things deeply.

I am grateful to my Ph.D. adviser Professor Jingang Yi for his support and guidance. Besides the knowledge I learned from him, I have always been encouraged by his enthusiasm for our work and enlightened by his vision on research topics. Without him, my work can never go as far as this. I would like to thank my committee members: Professor Haym Benaroya, Professor Laurent Burlion, and Professor Kostas Bekris for their valuable feedback on my work.

During these years of study, I work extensively with my fellow lab mates. I am grateful to Dr. Kaiyan Yu. We collaborate in the race-car project and faced the challenges of our adviser together. In the beginning of my Ph.D., she gave me a lot of help including my daily life and research. I have been lucky to work with Dr. Mitja Trkov and always admire him as a meticulously organized person and his effort to balance family and work.

I particularly want to express my gratitude to Dr. Pengcheng Wang for always being there when I needed someone to talk and his insightful comments. His talents are not limited to engineering and physics but also in Ping Pong, Badminton, Rocket Ball, and Ice Skating which we used to go a lot before he graduated and left Rutgers. If I want to name a humble and smart researcher I would mention Dr. Kuo Chen. He always inspired me with his dedication and hard work. I am thankful to Tarik Yigit and Merill Edmonds and they always showed the supports anytime I needed any during

our works in the Robotics, Automation, and Mechatronics (RAM) Lab. I would like to thank Yiran Wang, we prepared for our qualifications exams which it was one of the most challenging exams during my whole academic studies. Through our studies, she provides useful guidelines.

I am grateful to Nuwan Fernando, Umut Yuce, who helped to build the race track and RURacer-1. I would like to thank Alborz Jelvani and Dimitri Duma. Together, we redesigned RURacer and built RURacer-2 with 10 times more accurate sensing capability and higher computational power and solid structure. I am grateful to Jiaxing Yu for his tremendous help in conducting experimental results and initializing Robotics Operating System (ROS) for the race-car project. I am grateful to Mahsa Biglow for proofreading my dissertation and giving invaluable feedback. I would like to thank Siyu Chen, Chaoke Guo, Yongbin Gong, and Marko Mihalec for their discussion and help on multiple research works.

Despite the dificulites of working full-time and persuing a PhD degree, this made a very unique opportunaty for me that I could grow on both wings industry and academic simultaneously. I was the first one in our group to join Bell Labs completely remotely due to the lockdowns of COVID19 pandemic. However, it was a hard year for everyone I had a chance to work with wonderful researchers and learn from them. Iam speccially thankful to Dr. Ilija Hadzic, Dr. Mike Coss, Dr. Larry O'gorman and Akshay Gupta. I worked for Verizon for 4 years, which beside making a great atmosphere for employees it was where I meet my amazing colleagues and now friends. I have to say I am very gratefull to my everyone I have a chance to work with at Verizon speccially, Joe Cozzarelli, Ruben Cuadrat, Jeremy Nacer, Shiva Dandu, Gabor Illes, Isaac Calvo, Parissa Pandkhouh, Hossein Barough, Maryam Kabiri and many more who I enjoyed working with and always beign supportive about my studies.

A lot of friends at Rutgers were not directly related to my research but they made my journey much more enjoyable. I am grateful to Arastou, Taha, Sajjad, Saman, and Hessam. We had courses together and had discussions. I am thankful to Alireza Abbaspour, Yashar Musavi, Mehdi Souzanchi, and Soroush Arabshahi for many remote bits of help. I would like to thank all my friends with whom we went ice-skating, played

badminton, racketball, soccer, and volleyball. Winning the championships of soccer tournament at Rutgers Intramural with my friends, Parsa Rastin, Navid Arab, and Mohammadamin Mahmoudabadbozchelou.

Last but not the least, I am grateful to my wife, my parents, and my family. They always give me unconditional love and support.

Dedication

To my family!

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Replacing human-operated vehicles, which are today the primary mode of transportation across the world, with autonomous vehicles (also called self-driving, driverless or robotic vehicles), would inevitably have a profound impact on nearly every aspect of future human society [2]. Like any new technology, Autonomous Vehicles (AV) will become accepted by society if perceived as safe. This can only happen if automotive companies can ensure that AVs are safe, boost productivity, and reduce the environmental impact of traditional human-operated vehicles [3]. AVs need not only perform under normal driving conditions but also under specific safety scenarios ranging from routine police interactions to unexpected emergency situations that may require fast reactions (e.g., dodging an unexpected obstacle, saving valuable road users, or performing quick detours and fast evacuations). To this end, it is essential to look to professional driving skills in order to help design safe autonomous aggressive maneuvers for hazard avoidance [4].

About 94% of the approximately 1.25 million annual automotive deaths worldwide and the $40,000$ annual automotive deaths in the United States are attributed to drivers' behavior. According to the World Health Organization, 49% of road traffic deaths involve pedestrians (22%), cyclists (4%), motorcycles (23%) in addition to vehicle-animal crashes and accidents due to natural disasters [5]. The role of human drivers in vehicle accidents will not dramatically change unless advances in technology and science will make building AVs more reliable than human beings possible [6]. As with technologies such as Anti-Lock Braking Systems (ABS) and Electronic Stability Control (ESC), AVs aim to reduce fatalities and severe injuries in valuable road users by improving the safety of vehicle operation during emergency maneuvers [7].

Humans' natural and unconscious response to unpredictable hazards, such as an oncoming object, is to dodge rather than block. A motion planner and controller for safe agile maneuvers may be designed inspired by this natural evasive human reaction. Agile maneuvers performed by race-car drivers demonstrate that vehicles are capable of such evasive steering. In turn, these behaviors can be used by AVs in emergency situations to avoid collisions [1]. Despite the current conventional prohibitions against aggressive maneuvering on the road, AVs can learn from professional race-car drivers to perform stable yet agile motions [8]. Such actions increase AV maneuverability in avoiding unanticipated obstacles at high speed while maintaining vehicular control and ensuring safety across all operating environments [9].

Advancements in machine perception such as high precision sensors, computer vision, the use of LIDAR, enhanced embedded computers, and robust connectivity have enabled AVs to implement accurate cognitive algorithms such as motion planning and predictive controllers to pursue self-driving and obstacle avoidance autonomously. Advances in computational methods can make planning a motion safely by taking advantage of the vehicle's full dynamic capabilities possible. Motion planning algorithms are able to find all the physically feasible paths that the vehicle can take to avoid a collision. In contrast, the motion control system would be able to take advantage of the vehicle's full capabilities. Aggressive maneuvers for emergency situations require operating the vehicle up to its stability limits. These limitations are affected by various uncertainties such as modeled dynamics and the friction force between the tires and different road surfaces. Extensive driving experience and solid knowledge about the vehicle's operation enable race-car drivers to perform maneuvers at the limits of the vehicle's handling. This dissertation seeks to go beyond modeling the skills of a normal car driver in order to achieve the performance of a race-car driver in certain scenarios for autonomous driving.

A motion planning algorithm utilizing a high accuracy model of the car dynamics is able to find a path for AVs to compete with professional car drivers (and possibly win in a race). A predictive motion controller emulating the behavior of professional car drivers maximizes vehicle performance while maintaining stability in dynamic circumstances.

This dissertation proposes motion planning and controller machinery inspired by these techniques to safely maximize the autonomous vehicle's maneuverability in emergency situations. Aggressive maneuvering techniques similar to ESV and ABS can serve as emergency hazard avoidance for the next generation of safety features to keep vulnerable road users safe. This dissertation aims to boost the understanding of safe aggressive maneuvers such that the comprehension gained can be applied to motion planning and controller design for future AV safety features.

1.1 Background

Aggressive vehicle maneuvers are commonly used by professional race-car drivers to achieve fast and agile performance. Understanding these skilled maneuvers can help design autonomous driving capability and active safety features under extreme conditions such as emergency situations [1, 10, 11]. Aggressive maneuvering performed by race-car drivers is usually planned in advance as a feasible and safe path while they are aware of the limits for the vehicle's maximum maneuverability and ability to maintain the vehicle's motion still safe. The act of overseeing the agile maneuver by race-car drivers can be replaced by a motion planner, which is embedded the physical properties of the vehicle into consideration. Then, the motion controller performs the action by assuring the vehicle is able to follow the planned trajectory. Motion planner and motion controller design for aggressive maneuvering are analytically challenging, safety-critical, and computationally expensive.

Using professional driver testing data for pendulum-turn maneuvers shows the maneuver stability and agility performance demonstrated by more experienced race-car drivers out-performs normal drivers [10, 12]. Understanding professional driving skills will aid in designing human-inspired autonomous vehicle aggressive maneuvers intended to handle unstable vehicle motion in such scenarios as high-speed obstacle avoidance. Although significant efforts and progress have been made in recent years, systematic analysis and motion planning, and control system design are still needed to guarantee the performance and safety of these autonomous aggressive maneuvers. Thus, this dissertation proposes that insights gained from studying skilled drivers in highly dynamic

maneuvers and basic control theories with numerical methods for stability and safety analysis can be applied to guarantee the safety of vehicle control systems during aggressive maneuvering. Using agile maneuvers to autonomously avoid hazardous accidents and improve vulnerable road users safety is called Aggressive Maneuvering as a Safety Feature (AMSF). Figure 1.1 shows an example of the AMSF.

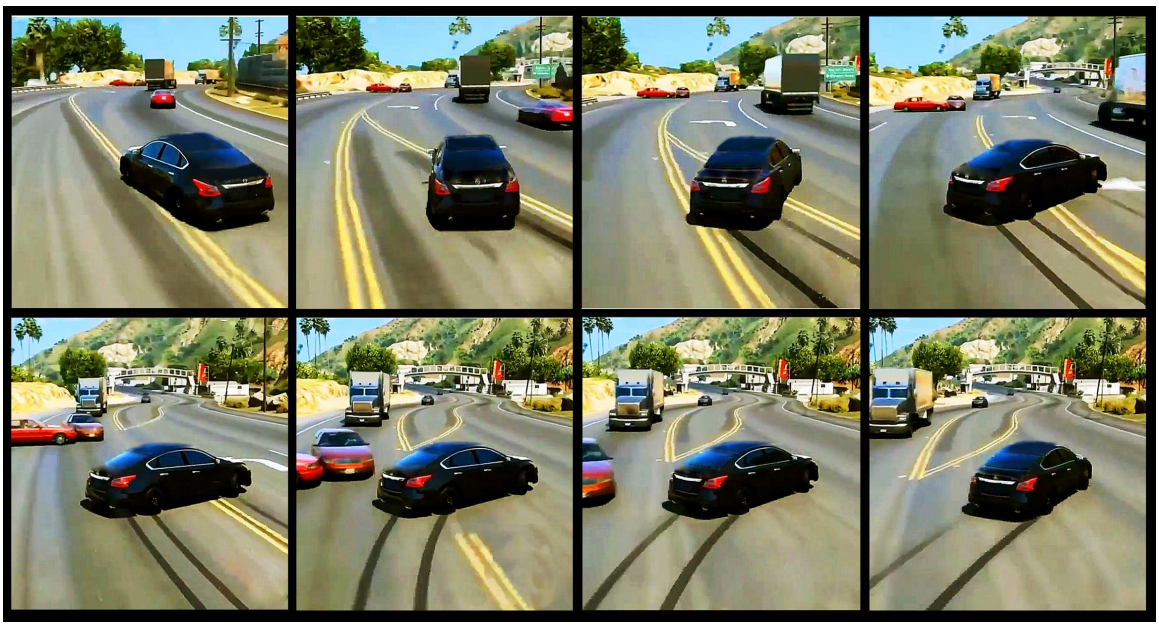


Figure 1.1: Frames captured from a simulations shows the car avoids a hazardous accident using aggressive maneuverings. (from top left). This image is extracted from video games.

1.1.1 Aggressive Maneuvering as a Safety Feature

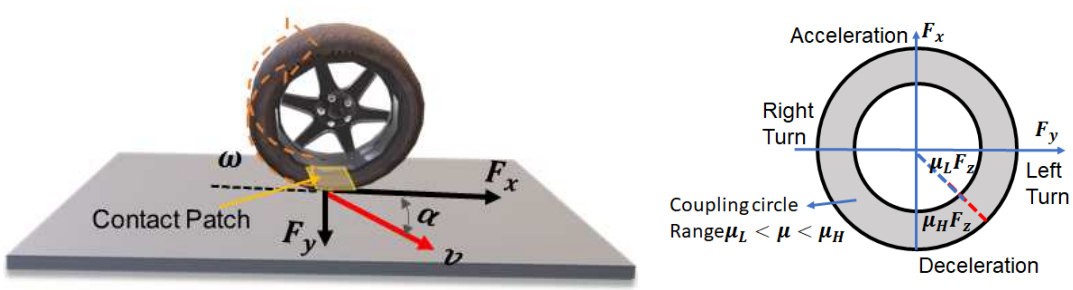
Studying aggressive maneuvers of robotic vehicles and autonomous cars has drawn academic and industry attention in recent years. However, most existing work, focuses on maneuvering and motion analysis [10,13–19]. Few works, use these analyses to increase safety and stability [20]. Investigating the stable and safe region as an attraction regions in the state space under certain control laws is a challenging problem [21]. Safety assurance in autonomous aggressive maneuvering is highly complicated, and vehicle dynamics are dependent on only traction or braking forces that are generated by four small contact patches between the tire and the road, as shown in Figure 1.2(a). Due to the complexity of modeling, sensing, and tire/road interactions, stability analysis

of ground vehicles for aggressive maneuvering becomes particularly challenging. The longitude and lateral stability of the vehicle for trail-braking and pendulum-turn cornering maneuvers by rally-racing drivers are studied, and an optimization approach is formulated to parameterize professional drivers' behavior [22–26].

Studying extremely aggressive maneuvers can be used for designing motion planning and controllers for handling vehicles under unstable motion, particularly in emergency situations such as high-speed obstacle avoidance and skidding. This approach requires a rethinking of the definition of vehicles' stability. Sliding and slipping are considered to potentially cause unstable motion in regular vehicle maneuvers. However, for aggressive maneuvering, they are instead considered useful to contribute to vehicles' safety under certain scenarios. The well-controlled utilization of the tire force capacity and intentional sliding and slipping by expert human drivers improves maneuvering agility. When attempting to drive collision-free paths in emergency situations, automobiles are limited by the amount of tire forces that can be produced through the contact patch. The limit, which is commonly shown as a frictional circle in Figure 1.2(b), is defined by the product of the tire-road friction coefficient and the vertical load of on the tire. The friction circle is captured as

$$\sqrt{F_x^2 + F_y^2} \leq F_z \mu, \tag{1.1}$$

where F_x, F_y, and F_z are the longitude, lateral and normal forces, respectively, and μ is the total friction coefficient that is highly uncertain and its accurate value is difficult to find in real time.



(a) (b)

Figure 1.2: (a) The schematic view of a tire. (b) Coupling circle of the friction with the friction range.

The tire-road friction model is highly nonlinear and consists model uncertainty, and the vehicle model is subject to nonholonomic constraints, which makes the control design more challenging since classic control methods are not able to solve this nonlinear control problem under motion constraints for aggressive maneuvering.

1.1.2 Stability of aggressive maneuvering

Researchers have made incredible advances in active safety control of the autonomous vehicle, with potentially hazard avoidance features that commonly limit the vehicle's motion [4, 27]. In contrast, the next generation of active safety control for autonomous vehicles should not restrict the vehicle's motion. Taking advantage of the vehicle's motion's maximum capacity is not possible without a proper understanding of the vehicle's dynamic. Hence, autonomous aggressive maneuvers can be planned in advance by understanding its dynamic and predicting maximum capability. Afterward, the predictive controller is able to follow the planned aggressive maneuvers for unexpected situations to avoid possible injuries to vulnerable road users.

Onboard vehicle sensors can measure the vehicle's current states in real time. Knowing the vehicle model and using a predictive controller, we can oversee the vehicle's future conditions, such as safety and stability, with certain accuracy. Predicting the vehicle's future condition and assuring its safety allows the controller to maximize the present performance while sustaining long-term stability. A stable autonomous controller for aggressive maneuvering under rapidly changing conditions should be able to predict the vehicle's handling limits and learn in a manner similar to professional drivers [28, 29]. Experienced drivers skim the road differently and predict the next scenarios and possible actions to avoid accidents. Classic analytical methods have been used to find and analyze dynamical systems' stability and safety. There is a lack of conversance to utilize these methods for complex problems such as autonomous aggressive maneuvering. Wherein this dissertation, we propose using various numerical methods for the safety-guaranteed control design. Constrained predictive controller, numerical stability analysis methods, and control barrier functions are methods capable of finding the safe subset of control commands. Applying the safety constraints to

the autonomous systems has received more attention in the recent years [30–34].

Motion stability is one of the key features of vehicle maneuvers. In [35], longitudinal motion stability is analyzed through tire slip dynamics under purely braking maneuvers. Lateral vehicle stability analyses in [15, 36] consider the dynamics of the vehicle yaw rate and the mass center side slip angle under a constant longitudinal velocity and zero tire slip ratio. The works in [10,12] define and illustrate the maneuver stability and agility performance. Agility metrics of vehicle motion are also proposed using professional driver testing data for a pendulum-turn maneuver. During aggressive maneuvers, vehicles operate under rapidly changing conditions, therefore we must consider transient motion stability as well. Race-car drivers can offer insights into vehicle control during extreme maneuvers; however, little data from race teams are publicly available for analysis. Researchers have studied race-car drivers' maneuvers during live racing events and collected vehicle's physical data acquired by sensors attached to the vintage race-cars and addressed the stability of aggressive maneuvers [28].

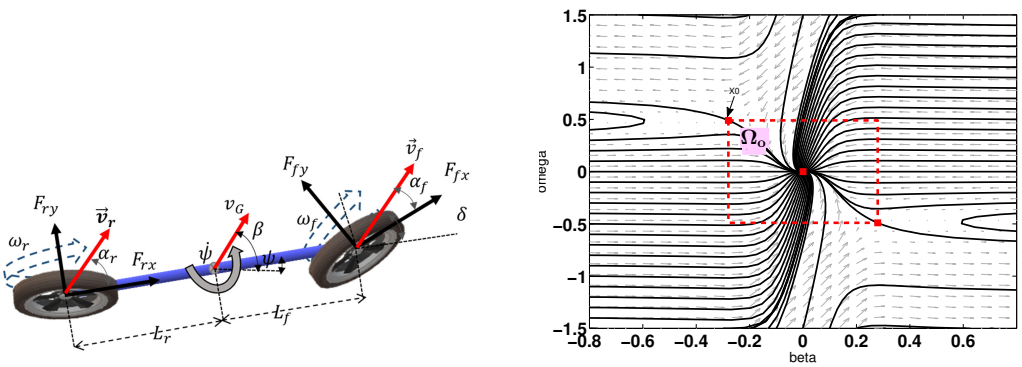
Understanding motion stability would be further helpful to design autonomous controllers to prevent the vehicles unsafe actions and improve hazard avoidance [4]. The motion stability defines the stable regions or envelopes and these stability envelopes are used in most autonomous driving controllers, for example, [1,22], etc. Model Predictive Control (MPC) is attractive to design control systems for the autonomous aggressive maneuvers because of its capability to systematically handle states and input constraints [37, 38]. Many other aggressive control designs, for example, [38–40] neglect the motion stability requirement, and no stability or safety is guaranteed when the controlled vehicles follow the desired path.

For simplicity and a better understanding of the overall complexity of the problem, two wheels on each axis (front and rear) are assumed to be a single wheel located at the midpoint of the vehicle's axis as shown in Figure 1.3(a). The two-wheel model or called bicycle model, reduces the dynamical system's complexity, which leads to less complicated stability analysis and control design, while the accuracy might be compromised. Instead, this dissertation uses a four-wheeled model for the higher accuracy of the control design. For the 2D bicycle model, the front wheel can be steered, and we

denote the steering angle as δ. The generalized coordinates are $q = [x, y, \theta]$, where (x, y) are the Cartesian coordinates of the center of gravity C_G, θ measures the orientation of the car body with respect to the x-axis. The dynamic model can be written as

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\beta} \\ \dot{\omega}_\psi \end{bmatrix} = \begin{bmatrix} \frac{F_{fy}+F_{ry}}{mV} - \omega_\psi \\ \frac{L_1 F_{fy} - L_2 F_{ry}}{I_z} \end{bmatrix} =: \boldsymbol{g}(\boldsymbol{x}, u), \tag{1.2}$$

where $\boldsymbol{x} = [\beta \ \omega_\psi]^T$, $u = \delta$. As shown in Figure 1.3(b), the *motion stability region* is defined as the closed region formed by the separatrices of the saddle points $\boldsymbol{x}_e^0$ of (1.2) under zero steering angle, $u = \delta = 0$. The stability region is considered as the maneuvering envelopes inside i.e., the vehicle motion should be kept.



(a) (b)

Figure 1.3: (a) The schematic view of a simplified bike model used to study the motion stability of a vehicle. (b) The open-loop phase portrait with a constant speed and zero steering wheel, the rectangle shows the utilized stability region by [1].

In [1], static stability regions are offered by neglecting the longitudinal dynamics. The stability regions used in the methods mentioned above are obtained through problem simplifications, such as considering steering wheel angle or longitude velocity constant. The vehicle's steering angle and longitude speed are not constant during the aggressive maneuvering, observing race-car drivers. Due to the vehicle and tire models' complexity, determining the stability region for safe aggressive maneuvering is not straight forward, and classic methods can't be used. This dissertation proposes a closed-loop stability region to maximize maneuverability without abdicating the safety.

Aggressive maneuvering is considered risky and is not socially and legally accepted.

However, it can be taken into consideration in designing targeted accident prevention policies [5]. Aggressive maneuvering might be the only feasible way to avoid unexpected dangerous situations such as a bicycle crossing the road, a pedestrian jaywalking the highway, or an animal jumping on the road, or a natural disaster blocking the road. These unexpected situations may cause making poor decisions and actions by human drivers. Average car drivers cannot react properly when aggressive maneuvering is the only feasible way to avoid danger. On-Road evaluation of hazard perception and reaction for human drivers and among the newly licensed teen drivers demonstrates poor skills, especially where unexpected hazards might materialize [41]. An autonomous aggressive maneuvering safety feature can deafest average human drivers, overcoming severe time constraints, limited knowledge about the vehicle's dynamic, and insufficient maneuvering skills.

This work aims to provide a provable stable, AMSF for future autonomous vehicles. We consider a car-like robotic platform in this study so that our results can be directly applied for vehicle active safety control. Provable stability is one of the critical characteristics of aggressive autonomous maneuvers as a safety feature. To keep the aggressive maneuvering stable, most control designs take the approach to restrict the motion within a restricted stability region [1, 4]. Our studies in this dissertation reveals that autonomous vehicles can perform similar or sometimes better than professional drivers to operate the vehicles outside the restricted stability regions to achieve superior agility. Therefore, to achieve aggressive maneuvering as a safety feature, it is required to design a stable controller with fewer restrictions to allow autonomous aggressive vehicle maneuvers for protecting vulnerable road users.

Envelope control is widely used in the aircraft industry. These systems allow pilots to freely operate the aircraft within a safe operating regime based on the aircraft load, pitch, bank, and speed limitations. The control system intervenes to prevent aircraft instability near and beyond the edges of this safe envelope. Both Airbus and Boeing implement envelope control in their aircraft, but their implementations differ on the extent of human versus computer control [9].

1.1.3 Motion planner and motion controller

A passenger car model is considered a nonholonomic dynamical system [42]. A nonholonomic constraint will not reduce the Degrees of Freedom (DoFs) of a mechanical system, but it adds a motion constraint. Nonholonomic constraints cannot be integrated and then will not reduce the number of DoFs. A vehicle, as a nonholonomic mechanical system, has only two control inputs, but there are 3-DoFs. Nonholonomic constraints in robotic are particularly interesting because implies that the mechanism can be completely controlled with a reduced number of actuators but it is subjected to particular paths taken to reach [43]. This has drawn the attention of dynamic control designers and engineers in the past decades. They have looked to such models as adaptive sliding-mode observer designed for a selective catalytic reduction system of a vehicle [44] and robust H_{inf} approach utilized to estimate the side-slip angle of the vehicle, a gain-scheduled robust control for lateral stability via linear parameter-varying technique [45], fuzzy sliding mode controller [46] and MPC [47–51]. Controlling the autonomous robot in the presence of uncertainty increases the complexity of control design [52, 53]. Therefore, designing a robust controller must overcome changes and uncertainties of the friction, weight shifting, and parameters for lane change maneuvers using adaptive sliding mode control with fuzzy boundary layer [54]. To address the estimation of uncertainties, we turn to learning methods such as neural networks [55], fuzzy systems [56], adaptive Taylor series [57] to ultimately create robust control of robot vehicles to enhance controller performance.

Nonholonomic constraints, the nonlinearity of the model, and uncertainties of the robotic vehicle bring additional complexity to the design of motion planning. It is significantly challenging to encode feasible and safe aggressive maneuvers into our proposed motion planner. Different tools for the stability and safety analysis of nonlinear control systems are available but not yet enough. Due to these complexities, the representation of vehicle stability is commonly defined as the stability of simplified dynamical systems (bike model) that consist of the rotational rate and the vehicle sideslip angle under a constant velocity and steering wheel angle [1, 4, 10]. This dissertation will show that

this method can't apply to a closed-loop control system with varying steering angle and vehicle's speed.

Understanding motion stability would offer profound insights in designing autonomous controllers to prevent risky actions in autonomous vehicles and to improve AVs' hazard avoidance capabilities. Motion stability determines stable regions or envelopes safe for the vehicle's maneuvers [1, 22]. Model Predictive Control (MPC) is an invaluable tool for designing control systems for autonomous aggressive maneuvers because it is capable of handling constraints subjected to the system and controller [37, 38, 50]. The stability boundaries or envelopes can be incorporated as constraints into NMPC's optimization engine. Many other aggressive control designs, for example, [38–40] neglect the motion stability requirement, and no stability or safety is guaranteed when the controlled vehicles follow the desired path.

More recent studies have introduced valuable methods to address the safety and stability of agile vehicle motions. Sensitivity analysis of path tracking at the vehicle's limit handling has been tested on the vehicle on a low friction surface [58]. Another research focused on minimizing lateral path tracking error [59]. Game theory has been utilized to design a planner for autonomous racing cars by representing the trajectory as a piecewise-polynomial and incorporating bicycle kinematics. This model has significantly out-performed a basic planner [60] as did a model combining of game-theoretic and a model predictive control algorithm designed for autonomous agile vehicles [32,61].

Similar to the envelope control widely used in the aircraft industry, autonomous vehicles allow the vehicle to freely maneuver within a safe, dynamic region determined by the vehicle's side-slip angle and yaw-rate. The model predictive controller prevents vehicle instability near and beyond the edges of the safe region [9]. Although some of the methods mentioned above show the intention to control autonomous aggressive maneuvers, they still restrict the vehicle's motion within a restricted region, which is not proper for a closed-loop control system. Studying professional driver performance [10, 12] reveals that the professional drivers sometimes operate the vehicle outside the car's stable regions to achieve high motion agility. Therefore, it is desirable to design autonomous driving strategies that deliver such unstable yet safe aggressive

maneuvers akin to professional drivers. A safety boundary is proposed in [12] extend the conservative stability region in the phase plane.

The most common approach to maintaining vehicle stability is to restrict the motion within the stability region of vehicle dynamics [1,4]. In [62], the learning model predictive control uses the data from previous motion to improve its performance; however, there is no proof that this satisfies safety requirements. The studies in [63] reveal that autonomous vehicles can perform similar to professional drivers in operating vehicles outside, rather than within, the stability regions to achieve superior agility. Therefore, in order to make highly aggressive maneuvers into a safety feature, we must perfect a provable stable controller to relax restrictions, allowing aggressive vehicle maneuvers for protecting vulnerable road users [59]. This easing of restrictions will allow vehicles to perform aggressive maneuvers [10, 64]. Existing obstacle avoidance systems, as well as future designs (including models incorporating aggressive maneuvering as safety features) must behave ethically in realistic scenarios to protect its passengers and vulnerable road users [65].

In the above-mentioned controller design for aggressive maneuvers, optimal trajectories are assumed to be known $a - prior$. Relaying on a given initial path simplifies the Nonlinear Programming (NP) hard problem of path finding to becomes a convex optimization [66, 67]. More generalized motion planning algorithms, such as sampling-based planners [68], are applied to find optimal motions and paths for aggressive maneuvers [4, 27, 69, 70]. Various versions of RRT^* as an anytime computation framework are exploit to find the optimal maneuvers. Although optimality and completeness might be guaranteed by these motion planners, the computational cost is one of the major drawbacks in implementing these methods for real time applications.

To address the computational efficacy issue of motion planning algorithms different methods have been implemented. Reusing the information from the previous computation cycle for online replanting will update the effected portion of the search trees by the change in a dynamic environment [71, 72]. Even though implementation of RRT^* is not amenable to real time computation on embedded computers for high-speed robotic applications, it can be implemented on future dedicated parallel-computation resources

on the cloud [70].

Aggressive vehicle maneuvers autonomously can be divided into three different methods,

- *Based on the analytical dynamic model [50, 73, 74]*
- Optimizing the inputs-output date mimicking expert human driver performance [75, 76]
- Combining the analytical and numerical stability, safety and agility knowledge with the learning and predictive methods [24, 77]

Instead of mimicking or using only analytical methods, this dissertation will use the human's expertise to design a motion planner and motion controller for safe autonomous aggressive maneuvering.

1.2 Dissertation outline and contributions

This dissertation presents the development, analysis, and design of a motion planner and controller for the safe autonomous aggressive and stunt vehicle maneuverings in six chapters. Chapter 1 is the introduction. In Chapter 2, we provide an overview of vehicle's model and define the safety region for aggressive maneuverings for a 1/7 scaled race-car. Chapter 3 introduces a predictive controller framework and addresses the feedback linearization integrated with a nonlinear model predictive controller to follow the center line of the race-track when is allowed to perform agile maneuvers. Chapter 4 describes how a successive optimal motion planning technique achieves the minimum lap times on the race track to defeat the professional driver for a lap time. Chapter 5 demonstrates how using numerical stability analyzing methods can help us to bring additional safety and robustness to the controller design for aggressive maneuverings combining analytical and numerical methods. In Chapter 6, we explain how to utilize reinforcement learning and heuristic optimization algorithms for a safe policy search based on the analytical model of the vehicle in simulations for autonomous stunt maneuvers in real world.

This dissertation aims to advance the understanding of aggressive maneuvers which can be used as a safety feature for future autonomous vehicles. To achieve our goal, a combination of expert human driver's experience, analytical methods, numerical stability analysis, learning machinery and policy search algorithms have been used. This dissertation demonstrates how safe agile maneuvers can be performed similar to professional drivers which can be used to increase the safety of vulnerable road users in future autonomous vehicles. Instead of previously used, open-loop safety region, a closed-loop safety region is introduced for the first time in Chapter 2. The safe control problem of the race-car is defined as a real time nonlinear programming problem subjected to the vehicle's model and the safety region. In Chapter 3, both simulation and experimental results imply that real time NMPC with an extended stability region improves the performance of autonomous aggressive maneuvering without losing stability. NMPC is able to follow the the trajectory and perform safe autonomous aggressive maneuvers but to defeat a professional race-car driver for a minimum-time lap challenge a global optimization algorithm is required.

Finding the shortest path or the fastest path while maintaining motion stability is necessary for developing a safe aggressive maneuvering. Finding the minimum time lap is a non-convex problem and NP-hard to solve. The optimal solution may not be unique and can take the form of a trade-off between different objectives. It is shown by observing measurably distinct driving styles that achieve similar results [8]. The main purpose of the discussed motion planning algorithm in Chapter 4 is to find the fastest feasible trajectory while maximizing motion stability. Contrary to the common implicit assumption that an optimal path always needs the maximum vehicle capability, the data compare skilled drivers and autonomous vehicles with similar outcomes but different car performance.

Optimal motion planning methods such as RRT^* require reconfiguration in the search trees to optimize trajectories. The new optimal node replaces the non-optimal node if NMPC can connect the new optimal node to the search tree. Chapter 4 explains how to use the NMPC described in Chapter 3 to find the minimum-time trajectory on the race track. NMPC and RRT^* have an extended stability region explained in

Chapter 3 to find the optimal trajectory. NMPC and *RRT** have extended stability region explained in Chapter 2 to maximize the capability of vehicle maneuvers without compromising safety.

The nonlinear multi-input multi-output nature of vehicle dynamics poses challenges in planning a predictable behavior path for autonomous vehicles in real time. *RRT* based algorithms are able to incorporate nonlinear dynamical model of the vehicle, range of uncertainties and constraints to find a path with guaranteed feasibility and safety. Despite all of the advantages, basic *RRT* methods are not optimal and there is no generalized method to rewire the nodes such as what was proposed by *RRT** to find the optimal solution. Enabled by the proposed motion planning method, NMPC is used to solve the complex problem of rewiring for *RRT**. The proposed spars *RRT** is an optimal path planning method which satisfies the safety constrains and uses NMPC for *rewiring* for the first time.

Chapter 5 presents how to use a numerical stability analyzing machinery to find the vehicle's stability region with modeling uncertainty. Chapter 5 describes the nature of the optimization and learning methods, including SOS and Polynomial Gaussian Process (GP) for the safety-guaranteed motion control design. The safety region can enhance safe aggressive autonomous driving using observed data in real time. Taking advantage of SOS as a numerical tool for stability assurance and GP to learn more about the system's uncertainty can improve the performance and robustness of the method.

The research community has been paying greater attention to expert race-car drivers in designing motion control and planning strategies for autonomous vehicles. Designers and engineers have looked to expert drivers for inspiration in developing new ways of thinking about stunt autonomous maneuvers. Chapter 6 presents that instructions from an expert stunt driver can be used to initiate the policy search for autonomous stunt vehicle maneuvers. In the final chapter we use reinforcement learning to find control policies with guaranteed safety for autonomous stunt maneuvering by knowing the range of uncertainties. We demonstrate how the control policy can be taken from simulation to the real world for stunt maneuvering and path following examples.

Chapter 2

Autonomous Vehicles and Aggressive Maneuvering

2.1 Introduction

This chapter introduces the state-space model of a vehicle dynamics, including the tire-road interactions. Aggressive maneuvering for the vehicles and the stability of these maneuvers are studied in this chapter. After defining the stability of the aggressive maneuvering, a safety region inside the stable region is defined. The safety region for the vehicle control is studied in both closed-loop and open-loop vehicle's control systems. The studies show the advantages of using the closed-loop safety region to increase the robot's motion capability. Linear Quadratic Regulator (LQR) controller presented in this chapter enables developing a comprehensive safety region analysis for the autonomous vehicle with the closed-loop control system. The stability region discovery framework presented in this chapter will serve as the foundation for the safety guaranteed motion planning and controller of aggressive autonomous maneuvers.

The last few years have witnessed technical competence adopted by autonomous and semi-autonomous vehicles in the car industry. Thus, it is essential to ensure that the vehicle dynamics are precisely achievable for high accuracy motion control and planning design. Although we are using a 1/7 scaled vehicle to lower the cost of our experiments, the proposed methods in this dissertation can be generalized for real size autonomous. AMSF can be designed as a unique equipment beside the myriad of electrical and mechanical subsystems and millions of lines of programming codes in them to bring more safety to the future autonomous vehicles. Despite current safety subsystems, such as drive dynamics control and anti-lock brake systems, which serve to increase the safety by limiting the maneuverability of the vehicle, AMSF takes advantage of the maximum capability of the vehicle for to increase the safety. To design the motion control and

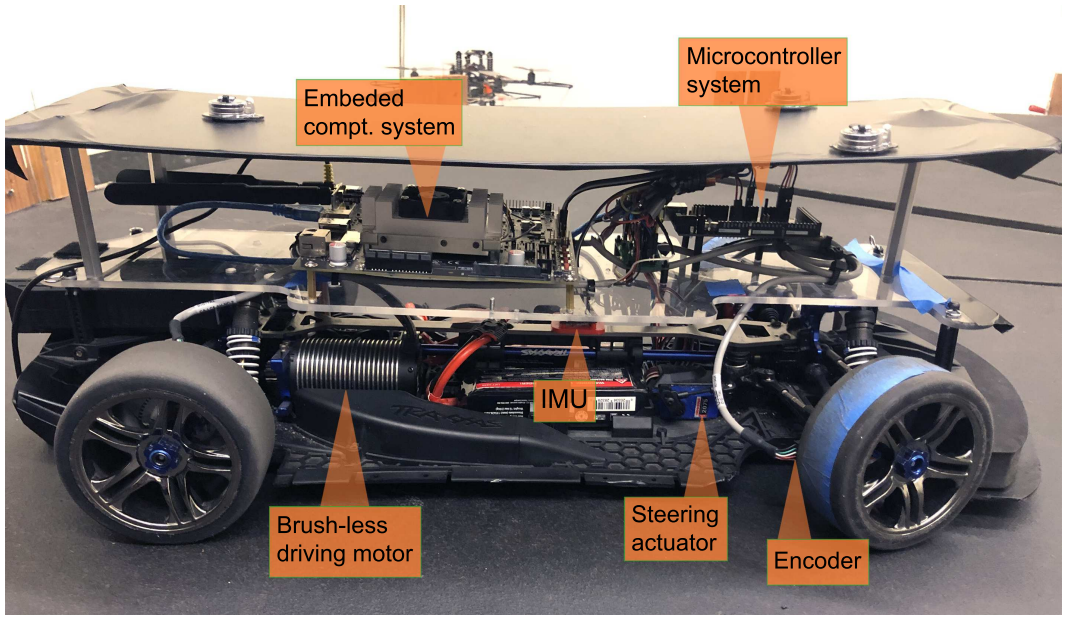
planning algorithms, the dynamical model must be known or approximated, and the physical interactions between the vehicular subsystems must be written in the form of state-space systems. The aim of this chapter is to produce a precise model; the methods of theoretical mechanics such as Lagrange or Euler principles are used.

As a result of the increase in computing power and advances in machine learning methods, comprehensive vehicle models can be used to describe autonomous vehicle behavior to design more reliable motion control and motion planning algorithms. We illustrate and demonstrate the experimental performance through a scaled robotic vehicle test-bed. A comparison with an existing control design in literature is also included and discussed. An integrated state-space model is introduced here using the car-like mobile robot's dynamics and kinematics that includes actuator modeling.

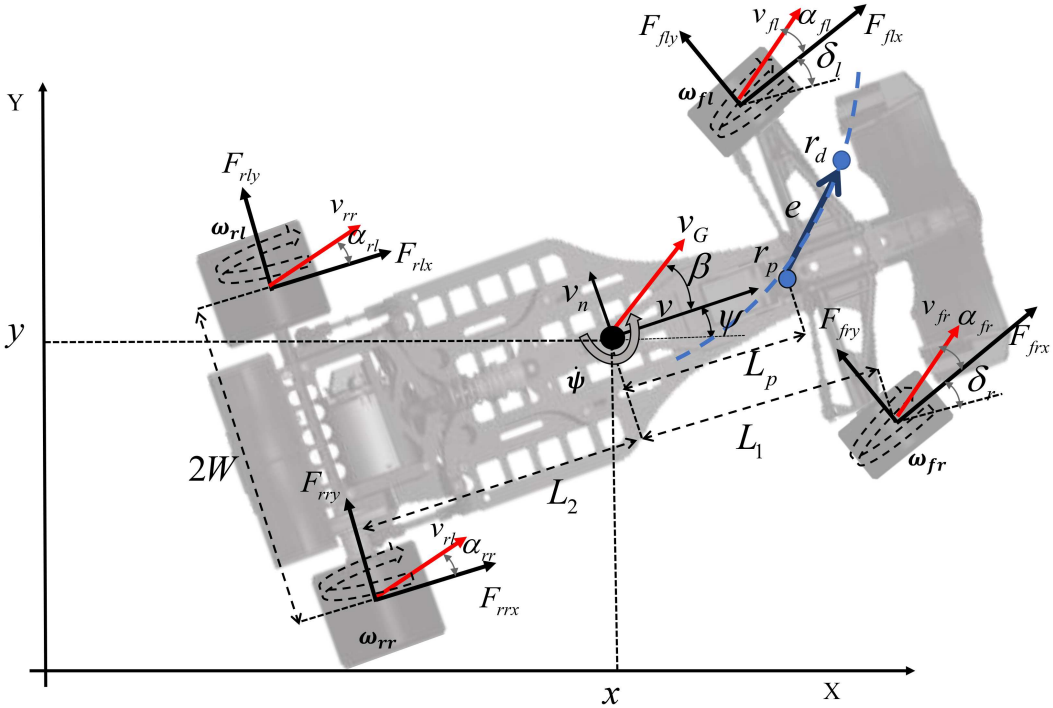
2.2 Vehicle model

There are numerous degrees of freedom associated with vehicle dynamics. The most simplified vehicle dynamic model is a 2-DOF bicycle model, representing the lateral and yaw motions described in Eq. (1.2). This model's drawbacks are that it doesn't include the dynamics for the longitudinal direction by assuming it does not affect the vehicle's lateral or yaw stability, which is not accurate due to the coupling effect of tires in Eq. (1.1) and load shifting on the side wheels. It's challenging to develop a dynamic model to capture vehicle motion in all directions and generally the model also requires computation cost. This section describes the derivation of the 3-DOF four-wheeled model used in this study. It also explains how to drive the equations for the front and rear tire of the right and left side slip ratio and side-slip angles. This section first describes the vehicle dynamics and then the dynamics of the tire-road interactions and then the motors' dynamics. Finally, an integrated state-space representation of the vehicle model is presented.

Figure 2.1(a) shows the scaled robotic vehicle used in the experiments. The scaled vehicle is a front-wheel steering mobile robot similar to the passenger car. More details of the experimental setup of the scaled robotic vehicle will be discussed in Section 2.4.1.



(a)



(b)

Figure 2.1: (a) Autonomous scaled vehicle. (b) A schematic of the robotic vehicle motion.

Figure. 2.1(b) illustrates the schematic of the kinematic and kinetic diagram. We consider the vehicle motion in the horizontal plane. A ground-fixed frame $\mathcal{N}(X, Y)$ and a body-fixed frame $\mathcal{B}(x, y)$ are used to build the model. The origin of $\mathcal{B}$ is located at the vehicle mass center (denoted as G). As shown in Figure 2.1(b), the location of G is at the middle point along with the robot lateral direction (but not along the longitudinal direction). The vehicle pose in $\mathcal{N}$ is denoted by $\boldsymbol{q} = [x\ y\ \psi]^T$, where $\boldsymbol{p} = [x\ \ y]^T$ is the position vector of G and ψ is the yaw angle with respect to the X-axis. The velocity vector is then $\mathbf{v} = [v_x\ v_y\ \omega_\psi]^T$ in $\mathcal{B}$ where $\omega_\psi = \dot{\psi}$. The distances from G to the front and the rear wheels are denoted as L_1 and L_2, respectively, and $L = L_1 + L_2$ is the wheelbase. The vehicle's width is denoted as $2W$, and the mass and moment of inertia about the z-axis in $\mathcal{B}$ are denoted as m and I_z, respectively.

The following list defines relevant definitions for the variables and the physical explanation associated with them in this dissertation.

Longitudinal force: The tire-road interaction forces toward the forward moving direction of each tire are named longitudinal forces. We denote these forces as F_{ijx}, where $i = f, r$, $j = l, r$, at the front (rear), left (right) wheels of the robotic vehicle, respectively.

Lateral force: Sideways direction of the forces applied to each wheel are lateral forces. These forces caused by the wheel's side-slip angles can control the rotation of the vehicle, but due to the configuration of the vehicle, we cannot avoid the lateral forces. Lateral forces can also be unexpected, similar to external disturbances. We denote the lateral forces as F_{ijy} where $i = f, r$, $j = l, r$, at the front (rear), left (right) wheel of the robotic vehicle, respectively.

Tire slip angle: This is equivalent to heading in a given direction but walking at an angle to that direction. Similarly, the tire might move in a direction while the heading direction is in a different direction, as it is shown by α in Figure 1.2(a). Each tire may have a different slip angle at the same instant in time, and we denote them as α_{ij} where $i = f, r$, $j = l, r$, at the front (rear), left (right) wheels of the robotic vehicle, respectively.

Tire slip ratio: Slip ratio describes the normalized relative velocity of the tire

center to the ground. The tire might have a faster or slower angular velocity at the moment comparing to the equivalent longitude velocity of the tire at the contacting point. Each tire may have a different slip ratio λ at the same instant in time, and we denote them as λ_{ij} where $i = f, r$, $j = l, r$, at the front (rear), left (right) wheels of the robotic vehicle, respectively.

Body-slip angle: β is the angle between the X-axis and the vehicle's velocity vector at the center of gravity point, as shown in Figure 2.1(b). It should be emphasized that this is different from the slip angle associated with the tire.

2.2.1 Vehicle dynamics model

The front steering angle is denoted as δ. It straightforward to obtain the motion equations of the vehicle system in Figure 2.1(b) as follows [78]:

$$m\dot{v}_x = mv_y\dot{\psi} + (\mathrm{c}_{\delta_l} F_{flx} + \mathrm{c}_\delta F_{frx}) + (\mathrm{s}_\delta F_{fly} + \mathrm{s}_\delta F_{fry}) + (F_{rlx} + F_{rrx}), \tag{2.1a}$$

$$m\dot{v}_y = -mv_x\dot{\psi} + (\mathrm{s}_\delta F_{flx} + \mathrm{s}_\delta F_{frx}) + (\mathrm{c}_\delta F_{fly} + \mathrm{c}_\delta F_{fry}) + (F_{rly} + F_{rry}), \tag{2.1b}$$

$$\begin{aligned} I_z\ddot{\psi} =& L_1[(\mathrm{s}_\delta F_{flx} + \mathrm{s}_\delta F_{frx}) + (\mathrm{c}_\delta F_{fly} + \mathrm{c}_\delta F_{fry})] - L_2(F_{rly} + F_{rry}) + \\ & W[(-\mathrm{c}_\delta F_{flx} + \mathrm{c}_\delta F_{frx}) + (\mathrm{s}_\delta F_{fly} - \mathrm{s}_\delta F_{fry}) + (-F_{rlx} + F_{rrx})]. \end{aligned} \tag{2.1c}$$

In the above and subsequent equations, we use $\delta \simeq \delta_l \simeq \delta_r$ and notation $\mathrm{s}_\delta := \sin\delta$ and $\mathrm{c}_\delta := \cos\delta$ for δ and other angles. The dynamic model (2.1) is then re-written in the matrix form as

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{B}_x(\delta)\mathbf{F}_x + \mathbf{B}_y(\delta)\mathbf{F}_y, \tag{2.2}$$

where $\mathbf{F}_x = [F_{flx}\ F_{frx}\ F_{rlx}\ F_{rrx}]^T$, $\mathbf{F}_y = [F_{fly}\ F_{fry}\ F_{rly}\ F_{rry}]^T$. and matrices $\mathbf{M}$, $\mathbf{C}$, $\mathbf{B}_x$ and $\mathbf{B}_y$ are given as

$$\boldsymbol{M} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_z \end{bmatrix}, \boldsymbol{C} = \begin{bmatrix} -mv_y\dot{\psi} \\ mv_x\dot{\psi} \\ 0 \end{bmatrix},$$

$$\boldsymbol{B}_x = \begin{bmatrix} \mathrm{c}_\delta & \mathrm{c}_\delta & 1 & 1 \\ \mathrm{s}_\delta & \mathrm{s}_\delta & 0 & 0 \\ L_1\,\mathrm{s}_\delta - W\,\mathrm{c}_\delta & L_1\,\mathrm{s}_\delta + W\,\mathrm{c}_\delta & -W & W \end{bmatrix},$$

$$\boldsymbol{B}_y = \begin{bmatrix} -\,\mathrm{s}_\delta & -\,\mathrm{s}_\delta & 0 & 0 \\ \mathrm{c}_\delta & \mathrm{c}_\delta & 1 & 1 \\ L_1\,\mathrm{c}_\delta - W\,\mathrm{s}_\delta & L_1\,\mathrm{c}_\delta + W\,\mathrm{s}_\delta & -L_2 & -L_2 \end{bmatrix}. \tag{2.3}$$

2.2.2 Tire/road friction modeling

The longitudinal and lateral friction forces depend on tire slip ratios and slip angles. The vehicle uses soft-rubber tires, and the motion heavily depends on the tire-road friction forces. Vehicle motion heavily depends on tire/road friction forces. The tire/road friction forces are related to longitudinal slip ratios, lateral slip angles, and normal loads at each tire. The longitudinal slip ratios are defined as:

$$\lambda_{ij} = \frac{v_{ij}\cos(\alpha_{ij}) - r_{ij}\omega_{ij}}{\max\{v_{ij}\cos(\alpha_{ij}), r_{ij}\omega_{ij}\}}, \quad i = f, r, \;\; j = l, r, \tag{2.4}$$

where r_{ij} is the tire's effective radius and v_{xij} and ω_{ij} are the relative velocity of the tire and the ground at the center of the tires and tire angular velocities, respectively. The lateral friction forces are functions of the slip angles, where α_{ij}, the front and rear tire slip angles, are defined as:

$$\alpha_{fj} = \tan^{-1}\left(\frac{v_y + L_1\dot{\psi}}{v_x}\right) - \delta, \quad \alpha_{rj} = \tan^{-1}\left(\frac{v_y - L_2\dot{\psi}}{v_x}\right), j = l, r. \tag{2.5}$$

The body-slip angle used in the bike model Eq. (1.2), is defined as:

$$\beta = \tan^{-1}\frac{v_{Gy}}{v_{Gx}}, \tag{2.6}$$

where we can write the kinematics relationship of the above calculations as $v := v_G\cos(\beta) = v_{Gx}$ and $v_n := v_G\sin(\beta) = v_{Gy}$. We define the variable $\sigma := \tan\beta = \frac{v_{Gy}}{v_{Gx}}$.

Due to the lateral and longitudinal accelerations, the load shifting can be obtained as

$$F_{fiz} = mg\frac{L_2}{2L} - \frac{mh}{2}\left(\frac{a_{Gx}}{L} \pm \frac{a_{Gy}}{2W}\right), \tag{2.7}$$

$$F_{riz} = mg\frac{L_1}{2L} + \frac{mh}{2}\left(\frac{a_{Gx}}{L} \mp \frac{a_{Gy}}{2W}\right), \quad i = l, r, \tag{2.8}$$

Using the *Pacejka's magic formula* from [79], we are able to estimate the longitudinal and the lateral friction forces caused by slip ratios (2.4) and slip angles (2.5) and loads of each tire (2.8) as

$$F_{ijx} = F_{ijz}\mu_{ijx}(\lambda_{ij}), F_{ijy} = F_{ijz}\mu_{ijy}(\alpha_{ij}), \quad i = f, r, \quad j = l, r, \tag{2.9}$$

The coupling effect between the longitudinal and the lateral friction forces is modeled as the friction circle shown in Eq. (1.1) for each wheel as

$$\sqrt{F_{ijx}^2 + F_{ijy}^2} \leq F_{ijz}\mu,$$

where μ is the total friction coefficient. For both $\mu_{ijx}(\lambda_{ij})$ and $\mu_{ijy}(\alpha_i)$, we use the *Pacejka's magic formula* [79] as

$$\mu(\alpha) = D\sin\left(C\tan^{-1}\left(B(1-E)\alpha + E\tan^{-1}(B\alpha)\right)\right) \tag{2.10}$$

where the model parameters B, C, D, and E are obtained through matching with experimental results that will be discussed in later sections. For both the longitude and lateral forces, we use the Pacejka's magic formula in this dissertation [79]. The values for the model parameters of the magic formula can be obtained by matching with experimental results [80, 81] or using estimation methods [82, 83].

2.2.3 Motor and transmission modeling

A model is briefly represented for the motor and transmission system in this section for the consistency of the robotic vehicle modeling. The longitude force commands can be assumed as the inputs of a robotic vehicle in Eq. (2.2); however, the robot is driven by an electric brushless servo motor shown in Figure 2.2(a). The use of motor dynamics and transmission system in the model leads us to a computationally fast

realistic controller. One electric motor provides the torques to shaft of the wheels via the dynamics

$$J_m\ddot{\varphi}_m + B_m\dot{\varphi}_m + \mathbf{r}_g^T\boldsymbol{\tau}_w = \tau_m \tag{2.11}$$

where $\tau_m \in \mathbb{R}$ is the torque provided by a brushless electric motor. J_m and B_m are the motor inertia and damping, respectively, and φ_m and $\ddot{\varphi}_m$ are the motor angular velocity and acceleration, respectively. $\boldsymbol{\tau_w} \in \mathbb{R}^4$ is the vector of wheel torques and $\mathbf{r}_g \in \mathbb{R}^4$ is the vector of the transmission ratio of the motor to each wheel. Electric motor's are usually controlled by the level of the voltage trough a Pulse Width Modulation (PWM) technique. Assuming all phases of the brushless DC motor have the same parameters, in order to obtain the motor voltage as the input of the system, we consider the electrical equation of geared brushless dc motors with transfer function deduction in below form

$$L_a\dot{I}_a + R_aI_a + K_b\dot{\varphi}_m = \zeta, \tag{2.12}$$

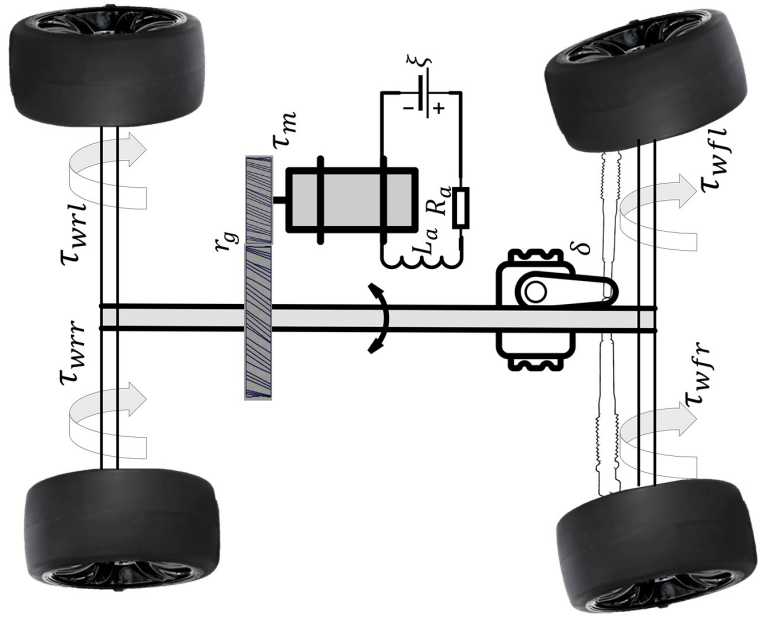
where $\zeta \in \mathbb{R}$ and $I_a \in \mathbb{R}$ are motor voltages and currents, respectively. Also $R_a, L_a, K_b \in \mathbb{R}$ represent the equivalent coefficients of the winding resistance, inductance, and back electromotive force (EMF) constant, successively. The motor torque τ_m is produced by the motor current

$$\tau_m = K_mI_a \tag{2.13}$$

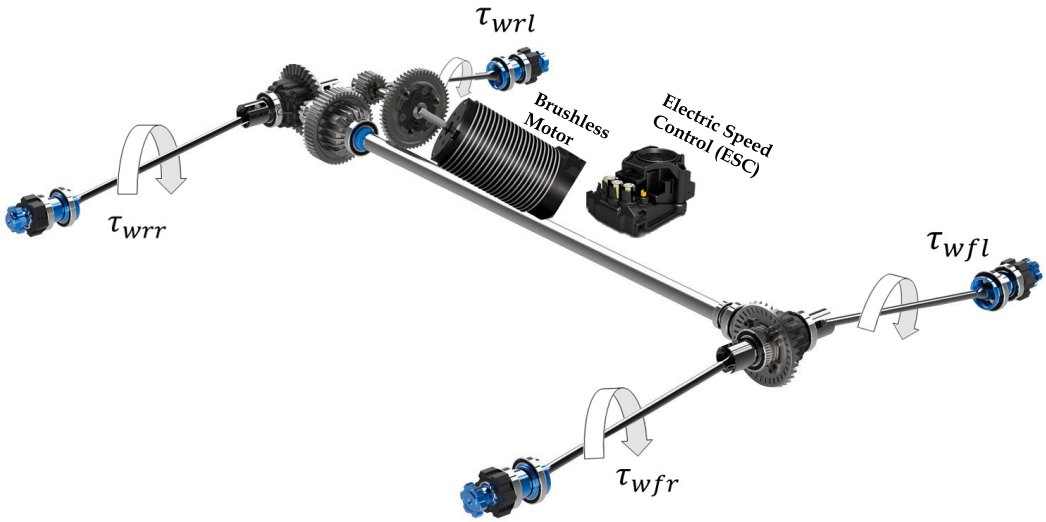
where K_m is the motor torque coefficient. The generated torque by the electric motor from Eqs. (2.11)-(2.12) is transferred to the wheels through the transmission system shown in Figure 2.2(b). This transmission system and differentials of the vehicle have been designed to work with 2WD or 4WD drivetrain where simplified modeling splits the motor's power attached to the driveshafts between the tires evenly [84]. The provided torque to each wheel rotates each wheel via the dynamics

$$\mathbf{J}_w\dot{\omega} + \mathbf{B}_w\omega + \mathbf{r}_w\mathbf{F}_x + \boldsymbol{\tau}_B = \boldsymbol{\tau}_w \tag{2.14}$$

where $\omega = [\omega_{fl}\ \omega_{fr}\ \omega_{rl}\ \omega_{rr}]^T$, $\mathbf{J}_w, \mathbf{B}_m$, and $\mathbf{r}_w$ are the 4×4 diagonal matrices for the wheel coefficients, namely, the inertia, damping, and radius, successively. Also, $\tau_w \in \mathbb{R}^4$ and $\tau_B \in \mathbb{R}^4_{\geq 0}$ is the vector of the breaks torques applied to each wheel's shaft independently via electric servos for RURacer-1 shown in Figure 2.5. Substituting



(a)



(b)

Figure 2.2: (a) Schematic of a DC electric brushless motor driven scaled vehicle and the electric servo for steering control. (b) A schematic of the transmission and reduction gears of the robotic vehicle.

Eqs. (2.13) and (2.12) into Eq. (2.11) by neglecting the impudence $L_a \approx 0$, torques of the four wheels can be calculated knowing the motor's voltage as

$$\boldsymbol{\tau}_w = (\mathbf{r}_g \mathbf{r}_g^T)^{-1} \mathbf{r}_g \left[R_a^{-1} K_m (\zeta - K_b \dot{\varphi}_m) - J_m \ddot{\varphi}_m - B_m \dot{\varphi}_m \right] \tag{2.15}$$

2.2.4 State-Space vehicle models

State-Space model without the motors dynamic

Defining the state variable $\mathbf{z} = [\mathbf{q}^T \ \dot{\mathbf{q}}^T]^T$ and inputs $\mathbf{u} = [\delta \ \mathbf{F}_x]^T$, we can rewrite the robot dynamics using (2.3) and (2.2) as

$$\dot{\mathbf{z}} = \mathbf{f}(\mathbf{z}, \mathbf{u}). \tag{2.16}$$

The represented model in Eq. (2.16) uses the steering wheel angle and the longitude force of the wheels as input. This model is not considered the wheel's motion dynamics, the motor's model, and the coupling effect on forces. This model's drawback is that there is no guarantee that we can generate the longitude forces accurately since there is no low-cost and precise method to measure the interaction force between the wheel and ground [85]. A significant number of researchers has used it control design [38, 86], however, this model is not complete and but it can be used for the study of the stability of the vehicle and control design purposes. A comprehensive state-space model combined with machine learning methods is introduced in Chapter 5.

2.3 Safety region for aggressive maneuvering

In this section, we first define the vehicle maneuver stability in an open-loop manner and then discuss how to determine the stability under a set of given closed-loop control systems. We also discuss the maneuver agility and present several agility metrics we use in our experimental comparisons. The safety region for the represented system is analyzed, and the closed-loop safety region is introduced. The dynamical model must reflect the vehicle's motion status and the frequency and duration of control actions to perform safe maneuvers adequately. We define the open-loop vehicle maneuver stability as follows.

Definition 2.1 *The* vehicle maneuver stable region *is defined as the domain of attraction of the nonlinear dynamical systems (2.1) and (1.2) ($\dot{\psi}$ and β as the state variables) under the given vehicle motion and steering angle. The vehicle maneuver is* stable *if the vehicle motion states are located within the stable maneuver region the states will converge to the* stable *equilibrium point.*

As shown in Figure 2.3(a), the *motion stability region* is defined as the closed region formed by the separatrix of the saddle points $\boldsymbol{x}_e^0$ of (1.2) under zero steering angle, $u = \delta = 0$. The stability region is considered as the maneuvering envelopes inside which the vehicle motion should be kept.

Precisely computing the stability region is expensive due to the complex nonlinear dynamics in (2.1). In [1, 74, 87], a simplified treatment takes the saddle points to construct a rectangular (or diamond) shape region $\boldsymbol{\Omega}_o$ in the β-$\dot{\psi}$ plane; see Figure 2.3(b). Denoting $\boldsymbol{x}_e^0 = [\beta_e^0 \;\; \dot{\psi}_e^0]^T$ as the saddle point of (1.2) under $\delta = 0$, namely, $\boldsymbol{g}(\boldsymbol{x}_e^0, 0) = \boldsymbol{0}$ in Eq. (1.2), we define simplified *open-loop stability region*

$$\boldsymbol{\Omega}_o = \{(\beta, \dot{\psi}) : |\beta| \leq |\beta_e^0|, |\dot{\psi}| \leq |\dot{\psi}_e^0|\}. \tag{2.17}$$

The region $\boldsymbol{\Omega}_o$ defined by (2.17) is shown in Figure 2.3(a). The results in [10] reveals that in aggressive maneuvers, the professional drivers operate the vehicles out of region $\boldsymbol{\Omega}_o$ and yet still maintain safety. Without clearly describing how to obtain the safety boundary, a safety region has been utilized for safe control of autonomous vehicles in [12]. In the following, we present a closed-loop safety region.

Definition 2.2 *The* vehicle maneuver stable region *is defined as the domain of attraction of the nonlinear dynamical systems (2.1) The* vehicle maneuver stable region *is defined as the domain of attraction of the nonlinear dynamical systems (2.1) and (1.2) ($\dot{\psi}$ and β as the state variables) under the given vehicle motion and feedback controller. The vehicle's maneuver is considered as* safe *if the vehicle's state at any given time during the maneuver is inside the safety region.*

Under optimal closed-loop steering wheel control, namely, LQR, we define *closed-loop stability region*

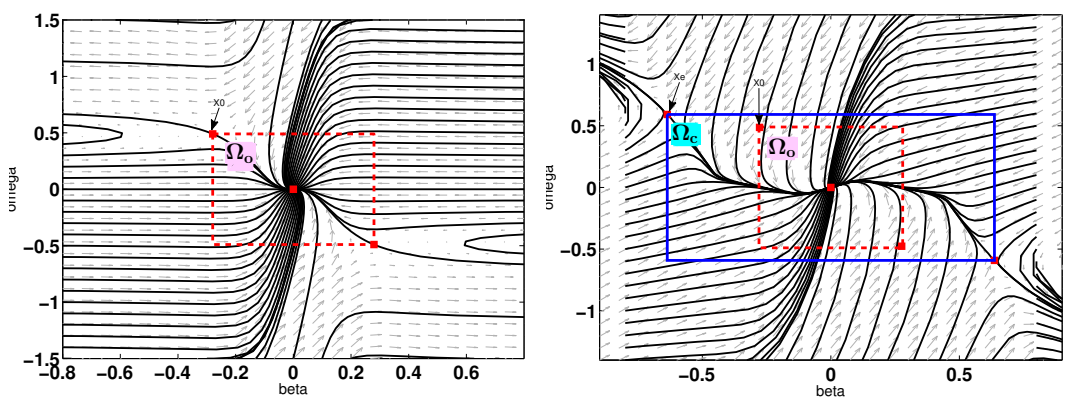
$$\boldsymbol{\Omega}_c = \{(\beta, \dot{\psi}) : |\beta| \leq |\beta_{ec}^0|, |\dot{\psi}| \leq |\dot{\psi}_{ec}^0|\}. \tag{2.18}$$

We augment $\boldsymbol{\Omega}_o$ and compute the safety region by considering vehicle motion with possibly maximum steering actuation. LQR method is used to find the state feedback control δ_{LQR} to obtain the augmented $\boldsymbol{\Omega}_o$.

$$\begin{aligned} \delta_{LQR} &= \arg\min_u \int_0^\infty \left(k_\beta \beta^2 + k_\psi \dot{\psi}^2 + Ru^2\right) dt, \\ \dot{\boldsymbol{x}} &= \boldsymbol{D}\boldsymbol{x} + \boldsymbol{E}u, \end{aligned} \tag{2.19}$$

where $\boldsymbol{D} = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\big|_{\boldsymbol{x}}$ and $\boldsymbol{E} = \frac{\partial \boldsymbol{g}}{\partial u}\big|_{\boldsymbol{x}}$ are the Jacobian linearization of nonlinear function $\boldsymbol{g}(\boldsymbol{x}, \delta)$ in (1.2). Parameters k_β, k_ψ and R are positive constant gains. The solution of the LQR design is given as $\delta_{LQR} = -\boldsymbol{K}\boldsymbol{x}$, where $\boldsymbol{K} \in \mathbb{R}^2$ is the feedback gain.

Lemma 2.1 *The safety region of vehicle without a controller as the open-loop region is a subset or equal to the safety region for the given closed-loop stabilizer controller of the vehicle that is* $\boldsymbol{\Omega}_o \subseteq \boldsymbol{\Omega}_c$.



(a) (b)

Figure 2.3: Stability regions and boundaries calculation. (a) The open-loop phase portrait with $\delta = 0$ and region $\boldsymbol{\Omega}_o$. (b) The open-loop phase portrait with steering feedback $\delta_{LQR} = -\boldsymbol{K}\boldsymbol{x}$ and region $\boldsymbol{\Omega}_c$.

Proof of Lemma 2.1

To prove the lemma, we need to compute and compare the non-zero equilibra $\boldsymbol{x}_e^0$ and $\boldsymbol{x}_e^c$ of dynamics (2.1) under $\delta = 0$ and $\delta = \delta_{LQR}$, respectively. To compute and analyze $\boldsymbol{x}_e^0$ and $\boldsymbol{x}_e^c$, we simplify the lateral friction forces in (2.1b). A piecewise linear friction

model [88] is used to approximate (2.10). The piecewise linear friction force function $F_y(\alpha)$, namely,

$$F_y(\alpha) = C(a_1 + a_2\alpha), \tag{2.20}$$

where C is the tire cornering stiffness coefficient. $a_1 = 0$, $a_2 = 1$ if $0 \le \alpha \le \alpha_m$; $a_1 = a_1^* = \frac{(\alpha_{\max}-\gamma_\alpha\alpha_m)\alpha_m}{\alpha_{\max}-\alpha_m}$, $a_2^* = \frac{-(1-\gamma_\alpha)\alpha_m}{\alpha_{\max}-\alpha_m}$ if $\alpha_m < \alpha \le \alpha_{\max}$, where α_m is the slip angle with the maximum $F_{\max}$, $\alpha_{\max}$ is the maximum slip angle, and $0 < \gamma_\alpha \le 1$ is a constant to denote the ratio of F_y at α_m and $\alpha_{\max}$. Note that $a_1^* > 0$ and $a_2^* < 0$.

Using (2.20) for the front and rear tires, model (2.1) is reduced to

$$\begin{aligned}\dot{\boldsymbol{x}} &= \underbrace{\begin{bmatrix} -\frac{C_f a_{2f}+C_r a_{2r}}{mV} & -1-\frac{L_{c2}}{mV^2} \\ -\frac{L_1 C_f a_{2f}-L_2 C_r a_{2r}}{I_z} & -\frac{L_1^2 C_f a_{2f}+L_2^2 C_r a_{2r}}{I_z V} \end{bmatrix}}_{\boldsymbol{D}} \boldsymbol{x} \\ &\quad + \underbrace{\begin{bmatrix} \frac{C_f a_{2f}}{mV} \\ \frac{L_1 C_f a_{2f}}{I_z} \end{bmatrix}}_{\boldsymbol{E}} \delta + \begin{bmatrix} \frac{C_f a_{1f}+C_r a_{1r}}{mV} \\ \frac{L_1 C_f a_{1f}-L_2 C_r a_{1r}}{I_z} \end{bmatrix}\end{aligned} \tag{2.21}$$

and the equilibrium $\boldsymbol{x}_e$ is obtained as $\boldsymbol{x}_e = \begin{bmatrix} \frac{D_1+D_2\delta}{\Delta} & \frac{D_3+D_4\delta}{\Delta} \end{bmatrix}^T$, where $D_1 = LC_FC_r(L_1a_{2f}a_{1r}+L_2a_{2r}a_{1f})-mV^2(L_1C_fa_{1f}-L_2C_ra_{1r})$, $D_2 = C_fa_{2f}(L_2LC_ra_{2r}-mV^2L_1)$, $D_3 = -LVC_fC_r(a_{2f}a_{1r}-a_{1f}a_{2r})$, $D_4 = -LVC_fC_ra_{2f}a_{2r}$, and $\Delta = L^2C_fC_ra_{2f}a_{2r} - mV^2(L_1C_fa_{2f} - L_2C_ra_{2r})$. Under $\delta = 0$, it is straightforward to obtain

$$\boldsymbol{x}_e^0 = \begin{bmatrix} \frac{D_1}{\Delta} & \frac{D_3}{\Delta} \end{bmatrix}^T. \tag{2.22}$$

With $\delta_{LQR} = -\boldsymbol{K}\boldsymbol{x} = -k_1x_1 - k_2x_2$, $k_1, k_2 > 0$, we obtain

$$\boldsymbol{x}_e^c = \begin{bmatrix} \frac{D_1\Delta+k_2(D_1D_4-D_2D_3)}{\Delta(\Delta+k_1D_2+k_2D_4)} & \frac{D_3\Delta+k_1(D_2D_3-D_1D_4)}{\Delta(\Delta+k_1D_2+k_2D_4)} \end{bmatrix}^T \tag{2.23}$$

From (2.22) and (2.23), the difference $\Delta\boldsymbol{x}_e = \boldsymbol{x}_e^c - \boldsymbol{x}_e^0$ is obtained as

$$\Delta\boldsymbol{x}_e = -\begin{bmatrix} \frac{D_2(k_1D_1+k_2D_3)}{\Delta(\Delta+k_1D_2+k_2D_4)} & \frac{D_4(k_1D_1+k_2D_3)}{\Delta(\Delta+k_1D_2+k_2D_4)} \end{bmatrix}^T. \tag{2.24}$$

We now show the results by the following four cases.

Case 1: $|\alpha_f| \le \alpha_{fm}$ and $|\alpha_r| \le \alpha_{rm}$. In this case, $a_{1i} = 0$, $a_{2i} = 1$, $i = f, r$, and therefore, $D_1 = 0$ and $D_3 = 0$. From (2.24), we have $\boldsymbol{x}_e^c = \boldsymbol{x}_e^0$ and by definition, $\boldsymbol{\Omega}_c = \boldsymbol{\Omega}_o$.

Case 2: $|\alpha_f| \leq \alpha_{fm}$ and $|\alpha_r| > \alpha_{rm}$. In this case, $a_{1f} = 0$, $a_{2f} = 1$, $a_{1r} = a_{1r}^* > 0$, and $a_{2r} = a_{2r}^* < 0$. Thus, $D_1 = LC_fC_rL_1a_{1r}^* + mV^2L_2C_ra_{1r}^* > 0$, $D_2 = C_f(L_2LC_ra_{2r}^* - mV^2L_1) < 0$, $D_3 = -LVC_fC_ra_{1r}^* < 0$, $D_4 = -LVC_fC_ra_{2r}^* > 0$, and $\Delta = L^2C_fC_ra_{2r}^* - mV^2(L_1C_f - L_2C_ra_{2r}^*) < 0$.

Note that $\beta_e^0 = \frac{D_1}{\Delta} < 0$ and $\dot{\psi}_e^0 = \frac{D_3}{\Delta} > 0$. To show $\mathbf{\Omega}_o \subseteq \mathbf{\Omega}_c$, we need to prove that $\beta_e^c - \beta_e^0 \leq 0$ and $\dot{\psi}_e^c - \dot{\psi}_e^0 \geq 0$. From (2.23), we find the conditions for the above inequalities to be held as $k_1D_1 + k_2D_3 \geq 0$, $\Delta + k_1D_2 + k_2D_4 > 0$ and these are satisfied if k_1 and k_2 are chosen as

$$-\frac{\Delta}{D_4} - \frac{D_2}{D_4}k_1 \leq k_2 \leq \frac{LC_fL_1 + mV^2L_2}{LVC_f}k_1. \tag{2.25}$$

Case 3: $|\alpha_f| > \alpha_{fm}$ and $|\alpha_r| \leq \alpha_{rm}$. In this case, $a_{1r} = 0$, $a_{2r} = 1$, $a_{1f} = a_{1f}^* > 0$, and $a_{2f} = a_{2f}^* < 0$. $D_1 = LC_fC_rL_2a_{1f}^* - mV^2L_1C_fa_{1f}^*$, $D_2 = C_fa_{2f}*(L_2LC_r - mV^2L_1)$, $D_3 = LVC_fC_ra_{1f}^* > 0$, $D_4 = -LVC_fC_ra_{2f}^* > 0$, and $\Delta = (L^2C_fC_r - mV^2L_1C_f)a_{2f}^* + mV^2L_2C_r$. For high vehicle velocity $v \geq \sqrt{\frac{LL_2C_r}{mL_1}}$, $\Delta > 0$, $D_1 < 0$, $D_2 > 0$. Similar to the previous case, if $k_1D_1 + k_2D_3 > 0$, then $\beta_e^c \leq \beta_e^0 \leq 0$ and $\dot{\psi}_e^c \geq \dot{\psi}_e^0 \geq 0$. Therefore, if choosing gains

$$k_2 > \frac{mL_1V^2 - LL_2C_r}{LVC_r}k_1, \tag{2.26}$$

then $\mathbf{\Omega}_o \subseteq \mathbf{\Omega}_c$.

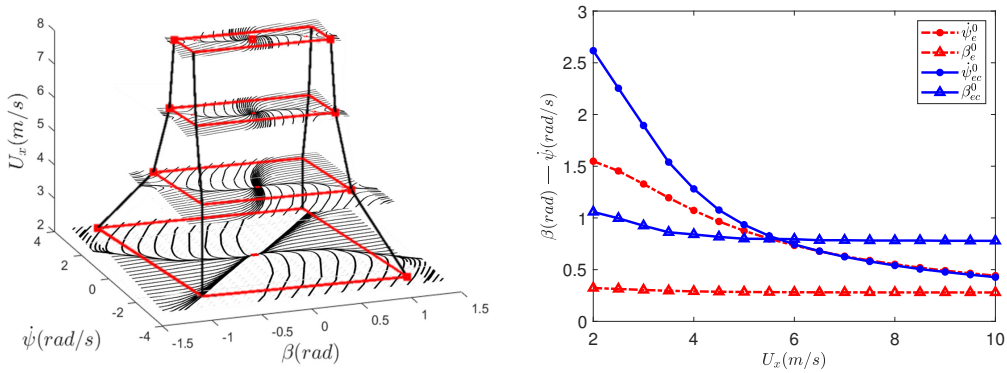
Case 4: $|\alpha_f| > \alpha_{fm}$ and $|\alpha_r| > \alpha_{rm}$. We have $a_{1r} = a_{1r}^* > 0$, $a_{1f} = a_{1f}^* > 0$, $a_{2f} = a_{2f}^* < 0$, and $a_{2r} = a_{2r}^* < 0$. For simplicity, we assume the same properties for the front and rear tires by $a_{1f}^* = a_{1r}^*$ and $a_{2f}^* = a_{2r}^*$. Then, $D_1 = L^2C_fC_ra_{1f}^*a_{2f}^* - mV^2a_{1f}^*(L_1C_f - L_2C_r)$, $D_2 = C_fa_{2f}^*(L_2LC_ra_{2f}^* - mV^2L_1) > 0$, $D_3 = 0$, $D_4 = -LVC_fC_r{a_{2f}^*}^2 < 0$, $\Delta = L^2C_fC_r{a_{2f}^*}^2 - mV^2a_{2f}^*(L_1C_f - L_2C_r)$.

If $L_1C_f \geq L_2C_r$ (i.e., oversteering), $\Delta > 0$, $D_1 < 0$, we need $\Delta + k_1D_1 + k_2D_4 < 0$, namely,

$$k_2 \quad > \quad \frac{mL_1V^2 - LL_2C_ra_{2f}^*}{LVC_ra_{2f}^*}k_1 + \frac{L^2C_fC_ra_{2f}^* - mV^2(L_1C_f - L_2C_r)}{LVC_fC_ra_{2f}^*}, \tag{2.27}$$

for $\beta_e^c \leq \beta_e^0 \leq 0$ and $\dot{\psi}_e^c \geq \dot{\psi}_e^0 = 0$. If $L_1C_f < L_2C_r$ (i.e., understeering), for high velocity $v \geq \sqrt{\frac{L^2C_fC_ra_{2f}^*}{L_1C_f - L_2C_r}}$, $\Delta < 0$, $D_1 > 0$, then if (2.27) is satisfied, similarly we obtain $\mathbf{\Omega}_o \subseteq \mathbf{\Omega}_c$.

In summary, we choose gains to satisfy (2.25), (2.26) or (2.27) and then $\mathbf{\Omega}_o \subseteq \mathbf{\Omega}_c$ and this completes the proof. These safety regions can be calculated for different speed as it is demonstrated in Figure 2.4.



(a) (b)

Figure 2.4: (a) Stability regions and the phase portraits for different speeds for a vehicle with the closed-loop controller. (b) The equilibrium points in Eqs. (2.17)-(2.18) for different longitude speeds.

2.4 Experimental setup

Two different versions of a scaled race-car are used for the tests, and their setups are explained in this section. The value of the model parameters of both versions of the scaled vehicles are the same and it is shown in Table 2.1.

Table 2.1: Testing robotic vehicle parameters

m (kg)	L_1 (m)	L_2 (m)	W (m)	h (m)	I_z (kg m^2)
6.0	0.2	0.2	0.15	0.05	0.25

2.4.1 Scaled car-like robot

The first version of the robotic vehicle (RURacer-1) shown in Figure 2.5 is a modified Traxxas XO-1 model with various added onboard sensors and actuators. This car has

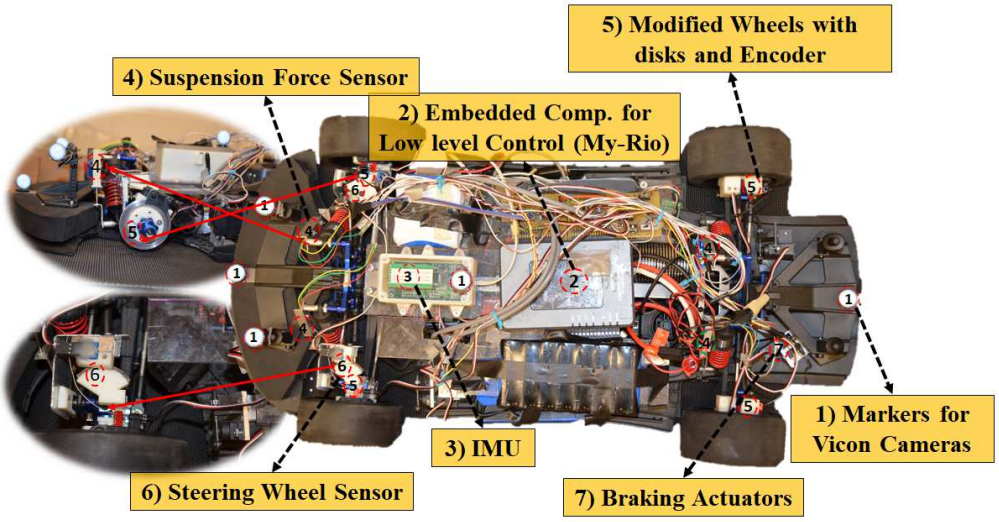
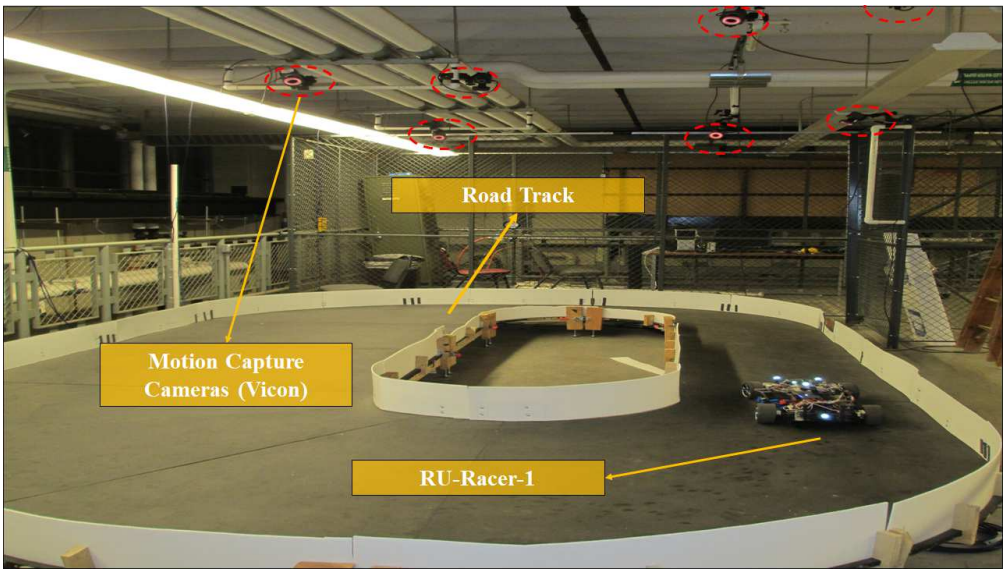


Figure 2.5: Autonomous scaled vehicle with four independent breaks and load sensors named RURacer-1.

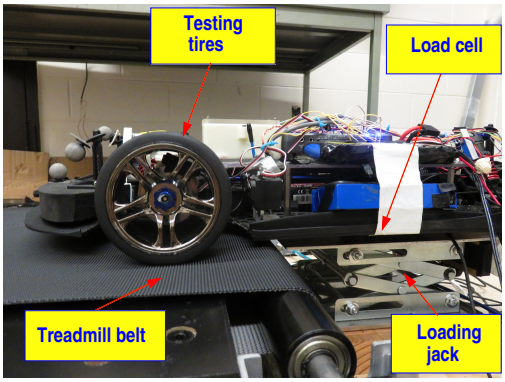
been prepared for autonomous agile maneuvering tests in the Robotics and Mechatronics (RAM) laboratories in the Department of Mechanical and Aerospace Engineering at Rutgers.

RURacer-1 is a four-wheel driven 1/7 scaled race-car powered by a powerful brushless electric motor with the front and rear differentials. At each wheel, an electric disc brake is added to provide the active braking capability. The steering and the brake actuators are operated by the servo motors controlled by the onboard embedded system. Four potentiometers are added to the suspensions to measure the deflection and the normal forces $\mathbf{F}_z$ at each tire. Two rotating potentiometers are added and used to measure the steering angle. All onboard sensors and actuators are connected to an embedded real time system (myRIO from National Instruments) with a sampling and control frequency of 1 kHz.

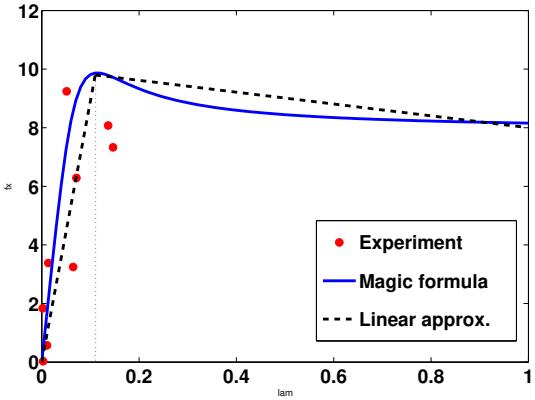
The robotic vehicle is tested on a wooden track, as shown in Figure 2.6(a). The track is specially designed to test the aggressive vehicle maneuvers with straight-line and U-turn shapes. A set of reflective markers are attached to the robotic vehicle. The vehicle's real time position is provided by the motion capture systems (8 Bonita cameras from Vicon Motion Systems Ltd).



(a)



(b)



(c)

Figure 2.6: (a) RURacer-1 on the testing race track and the motion capture cameras. (b) Tire force characteristics test setup. (c) Longitudinal tire force F_x testing results.

Tire force testing setup

We tested the tire characteristics and obtained the tire model. Figure 2.6(b) shows the tire force testing setup. The vehicle is supported by a loading jack, and the front two tires are pressed against a treadmill belt. The contact between the tire and belt surface has a similar friction property as the tire-track contact. A 6-DOF load cell is mounted rigidly between the vehicle and the jack. The normal force is kept around half of the vehicle's weight. The treadmill and the tire moving velocities are controlled to generate various slip ratios λ. Meanwhile, the corresponding longitudinal friction forces $\mathbf{F}_x$ are measured. Figure 2.6(c) shows the comparison of the magic formula, experimental results, and piecewise linear approximation for $\mathbf{F}_x$. Tables 2.2 and 2.3 represent the parameters for experimental results shown in Figure 2.6(c).

Table 2.2: Parameters of the magic formula

B	C	D	E
7.2	1.52	0.68	-1.999

Table 2.3: Parameters of the piecewise linear friction model

α_m	$\alpha_{\max}$	γ_a	Stiffness
0.135	2.2	0.5	76

2.4.2 Enhanced scaled car-like robot

The goal to build RURacer-2 was to create improve platform performance, such as enhanced wheel encoders and advanced embedded processors. Compared to RURacer-1, the new platform can reach four times lower latency (less than $7ms$) and twenty times faster computation power. To achieve this, an hardware architecture was designed along with the custom fitting of wheel encoders to monitor individual wheel speed and easily accessible and flexible hardware, software, and interfaces. An upgraded IMU system

was also added to the new version capable of estimating the vehicle's rotations in three dimensions in addition to 3D angular velocities and linear accelerations. This platform allows researchers to take advantage of the latest software and hardware platforms, such as the Robotic Operating System (ROS) and the NVIDIA Jetson TX2 system on module (SOM).

Hardware and software architecture

The scaled vehicle chosen for this project is a similar 1/7th scale Traxxas XO-1. The XO-1 shown in Figure 2.1(a) is highly capable of drifting as it features an optional rear-wheel drive system and can reach up to 100 miles per hour (MPH). To monitor individual wheels' angular velocity, optical wheel encoders (EM1 from US Digital) were custom fit and assembled on each side of the axles.

We chose NVIDIA Jetson-TX2 to control the vehicle because of it's high computational power for real time processing. TX2 is running the Linux kernel with an Ubuntu operating system created by NVIDIA called L4T. Also, TX2 features onboard WiFi, which we set up to connect to the router and the host computer for real time communications. The host computer provides high computational power at the edge for resource-hungry algorithms such as motion planning or image processing. The communication framework is built on top of ROS to receive and send data from different processing nodes.

A microcontroller controls the steering wheel's servo and the speed of the brushless motor. Also, optical encoders are connected to the microcontroller, communicating to TX2 by Universal Asynchronous Receiver/Transmitter (UART) using *rosserial*. Using UART, we achieve a $100Hz$ frequency for real time sensor data acquisition and actuator control. This allows running the NMPC motion controller on TX2 while the lower level programming part is on the microcontroller.

The IMU used for RURacer-2 is provided by Adafruit-BNO055, providing 9 DOF sensing information, that it has been difficult before. We used a 3.2 Mega Pixels (MP) FLIR Blackfly GigE camera at 30 frames per second, resulting in up to $2.6mm$ of accuracy to localize the vehicle. The camera is connected to the network, and the image

processing node is running on the edge processing unit, which is a powerful desktop computer. The architecture used for RURacer-2 is shown in Figure 2.7.

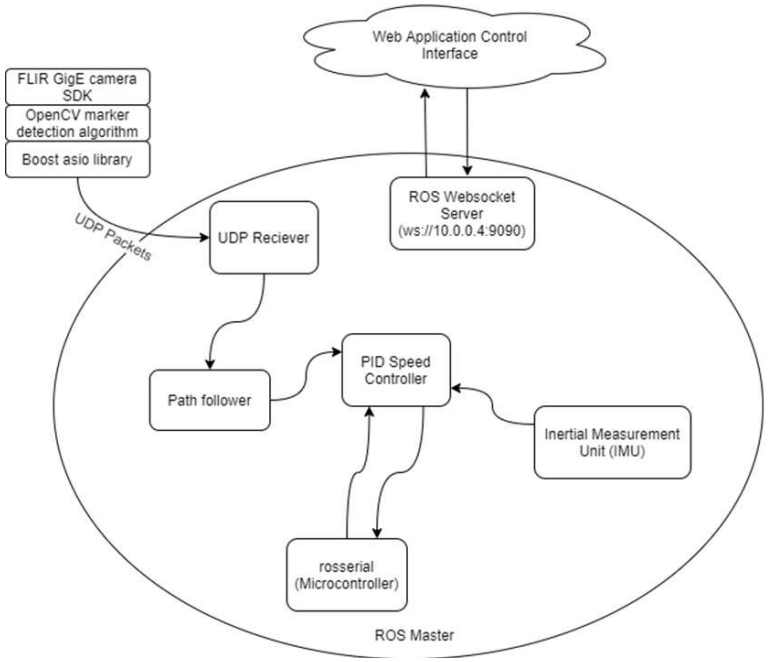


Figure 2.7: Software and communications architecture of RURacer-2 platform using camera for localization.

2.4.3 Existing platforms

RURacer has been specifically designed for the motion control and planning research purposes. There are existing open-source test-beds that can be used for similar applications.

- *MIT racecar: A The platform uses variety of sensors and computing hardware, placed on top of a 1/10-scale mini race car as a multi-purpose open-source platform for robotics research and education [89].*

- Georgia Tech Autorally: AutoRally is a scaled truck platform as a self-driving vehicle testbed for research purposes. This platform contains all the sensing or computing units designed for aggressive autonomous off-road driving tests [90].

- *UPENN F1/10: F1TENTH was designed as an open-source autonomous vehicle test-bed by the University of Pennsylvania for autonomous systems research and*

education purposes [91].

The contributors of RURacer platform include the materials to share this system as an open-source robotic vehicle test-bed for autonomous agile maneuvering research and education, specifically for motion planning and motion control. The main differences of RURacer platforms are the embedded accurate sensors and independent breaks. Load sensors, steering wheel angle sensor on the wheel, and independent breaks are unique differences of RURacers.

2.5 Summary

We introduced a closed-loop safety region instead of the previously used open-loop safety region for the aggressive vehicle maneuverings. LQR controller was used to find the safety for the autonomous vehicles when the maneuverability is not sacrificed. The proposed safety region increased the agility of the vehicle for hazard avoidance situations. The safety region will be then used as a convex constraint for the nonlinear model predictive controller. Also, RURacer-1 and RURacer-2, the car-like robots used for the experiments were presented. We explained the the localization and communications methods, attached sensors and actuators. We represented the tire testing setup which is utilized to measure the tire-road friction model characteristics. The dynamical model of the vehicle knowing it's physical properties is presented.

Chapter 3

Hybrid Nonlinear Model Predictive and Feedback Linearization Motion Controller

3.1 Introduction

While controlling autonomous vehicles has been well investigated in the past several decades, there are considerably fewer studies that address a path tracking controller that allows the car to perform safe aggressive maneuvers. Designing an envelope motion controller in order to operate the vehicle on the edges of the motion stability (which can be called aggressive maneuvering) will maximize the car's maneuverability for hazard avoidance situations. The proposed aggressive motion controller can also be used for other urgent delivery applications, agile and safe maneuvers, scaping from hazardous conditions, and high-speed obstacle avoidance.

Although there have been valuable efforts in developing control methods for autonomous vehicles, much of this research has focused on controlling the vehicle under regular driving conditions with moderate maneuvers [87, 92]. Studies on controlling vehicles in more critical maneuvers near the stability limits have decorated many associated challenges, such as sustainability and safety [10, 93–97].

Like the experienced drivers, particularly those with racing experience, the predictive controller can safely control the vehicle at the full limits of its maneuverability. Maintaining the vehicle's motion states inside the safety region is as challenging as determining the stability's margins for the predictive control design. Race-car drivers have demonstrated aggressive maneuvering capabilities by achieving minimum-time rounds on a race track. To prove this, the performance of autonomous aggressive maneuvers should be similar or defeat professional drivers. Nonetheless, the proposed controller is allowed to perform to the limits of the closed-loop stability region approximated in

Section 2.3.

In this dissertation, a hybrid nonlinear model predictive and feedback linearization controller is proposed for a path-tracking purpose. Using the proposed method, the vehicle is allowed to maximize the vehicle's maneuverability as thoroughly as a race-car driver. In this chapter, using a proper dynamic model for FLC and NMPC, the autonomous vehicle car could follow the trajectory designed on the race track with similar performance to an expert RC race-car driver.

3.2 Feedback linearization control

The vehicle control design consists of two parts: the first part is a Feedback Linearization Controller (FLC) to compute the desired tire friction forces. The second part is an NMPC controller to calculate the controlled inputs around the needed tire forces within the safety region. The two-part design is inspired by the tube-based MPC in [98].

Feedback linearization control is a common approach used in controlling a class of nonlinear systems. This method involves transforming the nonlinear system into a linear system through a change of variables and suitable control inputs. It was observed that the nonlinearity of saddle-point bifurcation would appear in vehicle dynamics with respect to the variation of the front wheel steering angle. The vehicle's model is not feedback linearization similar to what can be applied to an affine-type nonlinear system as given by a general nonlinear form

$$\begin{aligned} \dot{\mathbf{z}} &= \mathbf{f}_g(\mathbf{z}) + \mathbf{g}_g(\mathrm{z})\mathbf{u} \\ \mathbf{y} &= \mathbf{h}_g(\mathbf{z}) \end{aligned} \tag{3.1}$$

where $z \in \mathbb{R}^n$ is a vector of state variables, $\mathbf{u} \in \mathbb{R}^m$ is a vector of manipulated input variables, $\mathbf{y} \in \mathbb{R}^m$ is a vector of outputs, $\mathbf{f}_g(\cdot) \in \mathbb{R}^n$ is a general vector of nonlinear functions, $\mathbf{g}_g(\cdot) \in \mathbb{R}^{n\times m}$ is a matrix of nonlinear functions, and $\mathbf{h}_g(\cdot) \in \mathbb{R}^m$ is also a vector of nonlinear functions. The system (3.1) is said to be input-state feedback linearizable if there exists a diffeomorphism $T : D_z \in \mathbb{R}^n$ such that $D_x = T(D_z)$ contains the origin and the nonlinear mapping $\mathbf{x} = T(\mathbf{z})$ transforms the system (3.1) into a stable linear form by canceling the nonlinear as it is explained in Chapter 13

of [99]. Considering the vehicle dynamics in (2.1) as partially feedback linearizable by rewriting Eq. (2.16) in the form of

$$\begin{aligned} \dot{\mathbf{z}} &= \mathbf{f}_{FL}(\mathbf{z}, \delta) + \mathbf{g}_{FL}(\mathbf{z}, \delta)\mathbf{u}_2 \\ \mathbf{y} &= \mathbf{h}_{FL}(\mathbf{z}) \end{aligned} \tag{3.2}$$

where $\mathbf{z} = [\mathbf{q}\ \dot{\mathbf{q}}]$ the input vector $\mathbf{u}$ is spitted into the steering wheel angle δ and longitude force of the wheels $\mathbf{u}_2 = \mathbf{F}_x$, and $\mathbf{y} = [\mathbf{h}_1(\mathbf{z})\ \mathbf{h}_2(\mathbf{z})] = [\boldsymbol{r}_P\ \dot{\boldsymbol{r}_P}]^T$. We consider $\boldsymbol{r}_P$ as the center of the front wheels (center point between C_1 and C_2 in Figure 2.1(a)) and can be calculated

$$\boldsymbol{r}_P = \mathbf{h}_1(\mathbf{z}) = [x + L_1 c_\psi \quad y - L_1 s_\psi]^T, \tag{3.3}$$

and the time derivative of $\boldsymbol{r}_P$ is

$$\dot{\boldsymbol{r}}_P = \mathbf{h}_2(\mathbf{z}) = \boldsymbol{\Lambda}\mathbf{v},\ \boldsymbol{\Lambda} = \begin{bmatrix} \mathrm{c}_\psi & -\mathrm{s}_\psi & -L_1\,\mathrm{s}_\psi \\ \mathrm{s}_\psi & \mathrm{c}_\psi & L_1\,\mathrm{c}_\psi \end{bmatrix}. \tag{3.4}$$

$\mathbf{f}_{FL}$ and $\mathbf{g}_{FL}$, nonlinear models used in (3.2) for FLC design are given as

$$\mathbf{f}_{FL} = \begin{bmatrix} \dot{\mathbf{q}} \\ \mathbf{M}^{-1}\left[\mathbf{B}_y\mathbf{F}_y - \mathbf{C}(\mathbf{z})\right] \end{bmatrix},\ \mathbf{g}_{FL} = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1}\mathbf{B}_x \end{bmatrix}. \tag{3.5}$$

Using the vehicle dynamic equation (2.1) and the new variable definitions in (3.3) and (3.4), the time derivative of Eq. (3.4) is

$$\ddot{\boldsymbol{r}}_P = \boldsymbol{\Lambda}\boldsymbol{M}^{-1}(\boldsymbol{B}_x\boldsymbol{F}_x + \boldsymbol{B}_y\boldsymbol{F}_y - \boldsymbol{C}\boldsymbol{v}) + \dot{\boldsymbol{\Lambda}}\mathbf{v}, \tag{3.6}$$

where the model matrices are presented in (2.3). For a given desired trajectory $\boldsymbol{r}_d$, considering $\boldsymbol{F}_x$ as the controlled input $\boldsymbol{u}_2$, we have

$$\boldsymbol{F}_x = \boldsymbol{\Gamma}^{-1}[\ddot{\boldsymbol{r}}_d + \boldsymbol{K}_d(\dot{\boldsymbol{r}}_d - \dot{\boldsymbol{r}}_P) + \boldsymbol{K}_p(\boldsymbol{r}_d - \boldsymbol{r}_P) + \boldsymbol{\vartheta}], \tag{3.7}$$

where $\boldsymbol{\Gamma} = \boldsymbol{\Lambda}\boldsymbol{M}^{-1}\boldsymbol{B}_x$, $\boldsymbol{\vartheta} = \boldsymbol{\Lambda}\boldsymbol{M}^{-1}\boldsymbol{B}_y\boldsymbol{F}_y + \dot{\boldsymbol{\Lambda}}\mathbf{v}$, and $\boldsymbol{K}_p = \mathrm{diag}([k_{p1}; k_{p2}])$, $k_{p1}, k_{p2} \in \mathbb{R}^n_{\geq 0}$ and $\boldsymbol{K}_d = \mathrm{diag}([k_{d1}; k_{d2}])$, $k_{d1}, k_{d2} \in \mathbb{R}^n_{\geq 0}$ are positive diagonal matrices. Substituting the controller input (3.7) into (3.6) while defining error $\boldsymbol{e} = \boldsymbol{r}_d - \boldsymbol{r}_P$, we obtain

the closed-loop equation of

$$\ddot{\boldsymbol{e}} + \boldsymbol{K}_d\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e} = 0 \tag{3.8}$$

and the above error dynamics are stable for any given initial conditions. The longitude forces of each wheel $\boldsymbol{F}_x$ are controllable using independent break for RURacer1. Road-tire friction forces and brushless motor torques are causing to have bounded inputs and saturation effect. To resolve the input and constraints of the control system and safety region, NMPC, a constrained numerical optimization algorithm, will be introduced.

FLC is employed to construct the stabilizing control laws for the vehicle model. The stability of the overall closed-loop control system is then guaranteed by applying the stability region and NMPC. NMPC finds the steering angle to minimize the tracking error while maintaining the system stability using numerical optimization. It also ensures the motion is inside the safety region (2.18) by penalizing the sets of control inputs for potentially unstable behavior of the vehicle's dynamic.

3.2.1 Analysis of FLC for the vehicle control with model uncertainty

Using Lyapunov stability analysis can find the stable funnel for the controller in (3.7). The stable funnel is not the same as the safety region in Section 2.3 and it's basically the domain of attraction for the closed-loop control system in (3.8). Developing two control loops, namely, FLC and NMPC, helps us to form the integrated closed-loop system to find the region of attraction with the guaranteed asymptotic convergence. Due to difficulties of lateral forces precise estimation, we assume that there is an uncertainty in the part of the model $\boldsymbol{\vartheta}$ and we have an approximation of it as $\hat{\boldsymbol{\vartheta}}$. Concerning the closed-loop systems FLC with uncertainty in (3.7), let us formulate the following state-space equations as

$$\dot{\boldsymbol{E}}_L = \boldsymbol{A}_L\boldsymbol{E}_L + \boldsymbol{B}_L U_L \tag{3.9}$$

where

$$\boldsymbol{E}_L = \begin{bmatrix} \mathbf{e}(t) \\ \dot{\mathbf{e}}(t) \end{bmatrix}, \boldsymbol{A}_L = \begin{bmatrix} \mathbf{0}_{2\times2} & \mathbf{I}_{2\times2} \\ -\mathbf{k}_p & -\mathbf{k}_d \end{bmatrix}, \boldsymbol{B}_L = \begin{bmatrix} \mathbf{0}_{2\times2} \\ \mathbf{I}_{2\times2} \end{bmatrix}, \mathbf{U}_L = \hat{\boldsymbol{\vartheta}} - \boldsymbol{\vartheta}, \tag{3.10}$$

$\boldsymbol{E}_L \in \mathbb{R}^4$ is the vector of error and it's derivative, $\boldsymbol{A}_L \in \mathbb{R}^{4\times 4}$ and $\boldsymbol{B}_L \in \mathbb{R}^{4\times 2}$ represents the state-space matrix of the closed-loop control system and $\boldsymbol{U}_L \in \mathbb{R}^4$ is the approximation error. Before analyzing the stability of FLC, the following assumptions are required.

Assumption 3.1 *The desired trajectory, $\boldsymbol{r}_d$, and its time derivatives $\dot{\boldsymbol{r}}_d$ are smooth and uniformly bounded.*

Assumption 3.2 $\hat{\boldsymbol{\vartheta}} - \boldsymbol{\vartheta} = \boldsymbol{\Lambda}\boldsymbol{M}^{-1}\boldsymbol{B}_y(\mathbf{F}_y - \hat{\mathbf{F}}_y)$ *is bounded, since the lateral force $\mathbf{F}_y$, and the estimation of the lateral force $\hat{\mathbf{F}}_y$, and the parameters for the model $\boldsymbol{M}$, $\boldsymbol{\Lambda}$, $\boldsymbol{B}_y$ are also bounded.*

Let $V_L(t)$ be a continuously differentiable function defined over domain $D \subset R^n$

$$V_L(t) = \frac{1}{2}\boldsymbol{E}_L^T\boldsymbol{P}\boldsymbol{E}_L \tag{3.11}$$

Stability region $\Omega \subset D$ as a compact set that is positively invariant with respect to $\dot{\boldsymbol{E}}_L$ such that $\dot{V}_L(t) \leq 0$. Let's evaluate the time derivative of $V(t)$ to find the attraction region of Ω

$$\dot{V}_L(t) = \frac{1}{2}\boldsymbol{E}_L^T\left(\boldsymbol{A}_L^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}_L\right)\boldsymbol{E}_L + \boldsymbol{E}_L^T\boldsymbol{P}\boldsymbol{B}_L\boldsymbol{U}_L \tag{3.12}$$

where matrix $\boldsymbol{Q}$ is positive definite and by solving $\boldsymbol{A}_L^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}_L = -\boldsymbol{Q}$ we will find a symmetric positive solution for $\boldsymbol{P}$ to guarantee the stability of the closed-loop system in (3.9) for every state starting in Ω. One can rewrite Eq. (3.12) as

$$\dot{V}_L = -\frac{1}{2}\boldsymbol{E}_L^T\boldsymbol{Q}\boldsymbol{E}_L + \boldsymbol{E}_L^T\boldsymbol{P}\boldsymbol{B}_L\boldsymbol{U}_L, \tag{3.13}$$

The derivative of Lyapunov equation Eq. (3.13) has the unique symmetric positive definite solution. By selecting $k_{pi} > 0$, $k_{di} > 0$, all eigenvalues of $\boldsymbol{A}_L$ are in the open left-half of the complex plane. To make it clear, one can calculate a characteristic equation as $\lambda_{1,2} = -k_{d1}/2 \pm \sqrt{k_{d1}^2 - 4k_{p1}}/2$, $\lambda_{3,4} = -k_{d2}/2 \pm \sqrt{k_{d2}^2 - 4k_{p2}}/2$ to show the Asymptotically Stability is feasible when $k_{di} > 0$ and $k_{di}^2 > 4k_{pi}$, $i = 1, 2$.

Based on the above discussion and Eq. (3.13), the derivative of the Lyapunov function should be negative, $\dot{V}_L(t) < 0$ in Ω to guarantee the stability of the closed-loop

control system. The attraction region is positively invariant if we can satisfy the inequality

$$\boldsymbol{E}_L^T \boldsymbol{P} \boldsymbol{B}_L \boldsymbol{\Lambda} \boldsymbol{M}^{-1} \boldsymbol{B}_y (\boldsymbol{F}_y - \hat{\boldsymbol{F}}_y) < \frac{1}{2} \boldsymbol{E}_L^T \boldsymbol{Q} \boldsymbol{E}_L \implies \dot{V}_L(t) < 0 \tag{3.14}$$

Knowing the vehicle dynamics are locally Lipschitz functions, we can use NMPC to find robust and stable steering wheel commands around a neighborhood of the domain of attraction Ω by knowing the range of accuracy for lateral forces approximations $\inf_\delta \|\boldsymbol{F}_y - \hat{\boldsymbol{F}}_y\| \leq \rho$. Using the range of lateral forces approximation and triangle inequality we can find the upper bound of the error as

$$\|\boldsymbol{E}\| < \frac{2\rho \|\boldsymbol{P} \boldsymbol{B}_L \boldsymbol{\Lambda} \boldsymbol{M}^{-1} \boldsymbol{B}_y\|}{\lambda_{Qmax}}, \tag{3.15}$$

where λ_{Qmax} is the largest eigenvalue for $\boldsymbol{Q}$ and ρ is the maximum approximation error for the tire lateral force modeling.

Using Lyapunov function to define the domain of attraction and integrate it with NMPC to find the steering wheel angle to follow a desired trajectory while it's subjected to safety region constraints is discussed in Chapter 2. The stability results are given for the nominal model, and the effects of uncertainties are briefly studied. Knowing the range of uncertainties, we can find the domain of attraction which the closed-loop controller is asymptotically stable and using NMPC will allow us to apply safety and motion constraints.

3.3 Nonlinear Model Predictive Control

In this section, we introduce the NMPC method for constrained control of dynamic systems such as car-like robot. First, we define the classic NMPC algorithm and then the constrained control problem is formalized to control an autonomous vehicle's aggressive maneuvering within the safety region. After introducing NMPC in a special setting, we describe various extensions of the basic algorithm to apply the safety region as a constraint for the dynamic system. Finally, we investigate the experimental challenges and present the results using the NMPC.

MPC calculates control actions for a prediction horizon at each control interval, using a combination of model-based prediction and constrained optimization. The key difference between linear MPC and NMPC is that the model can be nonlinear and include time-varying parameters. Nonlinear model predictive control is an optimization-based method for the feedback control of nonlinear systems. Its primary applications are stabilization and tracking problems. NMPC is flexible to integrate various state/input boundaries by defining customized penalty functions. NMPC originated from the optimal control theory with fundamental optimization contributions, like the numerical optimizations for convex and non-convex problems, dynamic programming, and the work of Lev Pontryagin and Richard Bellman in the 1950s [100, 101].

3.3.1 NMPC formulation

NMPC, at each controlling instant runs a dynamic programming optimizer to predict future actions of the system over a finite time horizon, $k = 0, \ldots, N-1$, of length $N \geq 2$. The first element of the resulting optimal control sequence is used and implemented the next step of the system. In this section, we give a mathematical description of using NMPC for constrained control systems to follow the desired trajectory $\mathbf{z}_*$ inside the feasible subset of states $\mathbf{z}_\mathrm{d} \equiv \mathbf{z}_* \in Z$ consists aggressive maneuvers. $Z \in \mathbb{R}^n$ and $U \in \mathbb{R}^m$ represent the overall states and inputs space for the vehicle's model. The definitions and assumptions and theoretical development of NMPC in this section are obtained from [102]. The numerical simulations and experimental results compared with a professional race-car driver will then be presented.

A prerequisite for being able to find a control policy to converge to a desired state $\mathbf{z}_*$ is to make $\mathbf{z}_*$ an equilibrium of the nominal closed-loop system. Also, NMPC is able to find the optimal control policy properly if $\mathbf{z}_*$ and the initial states of the system are inside the permitted state space Ω_p and they are not violating control system constraints. $\Omega_p \equiv \mathbb{Z}$ is a subset of states which are not violating any constraints. Assuming that $\mathbf{u}_*$ is the control command that minimizes the trajectory error, where $\mathbf{u}_* \in U$ where $[\mathbf{z}(t) \in \Omega_p \forall\, t > 0,\ \mathbf{z}(t) \to \mathbf{z}_*]$ subjected to the nonlinear model equation

$\mathbf{f}(\cdot)$ will lead us to

$$\dot{\mathbf{z}}_* = \mathbf{f}\left(\mathbf{z}_*, \mathbf{u}_*\right), \tag{3.16}$$

which it will be predicatively computed it in a sequel for the prediction horizon. The cost function to be used in our optimization should penalize the distance of an arbitrary state $\mathbf{z} \in Z$ to $\mathbf{z}_*$ and the violation cost of the control system form the constraints. One of the main reasons for the success of NMPC is its ability to take safety and motion constraints into account explicitly. Here, we consider mixed constraints for the safety of the control system, states and inputs. To this end, we introduce a nonempty set $\mathbb{Z} \subseteq Z$ that doesn't violate the constraints and for each $\mathbf{z} \in \mathbb{Z}$ we introduce a nonempty control constraint set $\mathbb{U}(\mathbf{z}) \subseteq U$. Of course, $\mathbb{U}$ may also be chosen independent of $\mathbf{z}$. The idea behind introducing these sets is that we want the trajectories to lie in $\mathbb{Z}$ and the corresponding control values to lie in $\mathbb{U}(\mathbf{z})$. Given such a cost function $J(z, u, \mathbb{Z}, \mathbb{U})$ and a prediction horizon length $N \geq 2$, we can now formulate the basic NMPC scheme in Algorithm 3.1.

Algorithm 3.1: (Basic NMPC algorithm for constrained inputs and states)
At each sampling time $t_n, n = 0, 1, 2 \ldots$

1 Measure the state $\mathbf{z}(n) \in Z$ of the system;
2 Set $\mathbf{z}_0 := \mathbf{z}(n)$, solve the optimal control problem
minimize $\quad J_N\left(\mathbf{z}_0, \mathbf{u}(\cdot)\right) := \sum_{k=0}^{N-1} \ell\left(\mathbf{z}_u\left(k, \mathbf{z}_0\right), \mathbf{u}(k)\right)$ with respect to
$\mathbf{u}(\cdot) \in \mathbb{U}^N\left(\mathbf{z}_0\right), \quad$ subject to
$\mathbf{z}_u\left(0, \mathbf{z}_0\right) = \mathbf{z}_0, \quad \mathbf{z}_u\left(k+1, \mathbf{z}_0\right) = \mathbf{z}\left(\mathbf{z}_u\left(k, \mathbf{z}_0\right), \mathbf{u}(k)\right)$ and denote the obtained optimal control sequence by $\mathbf{u}^\star(\cdot) \in \mathbb{U}^N\left(\mathbf{z}_0\right)$.;
3 Define the NMPC-feedback value $\mu_N(\mathbf{z}(n)) := \mathbf{u}^\star(0) \in U$ and use this control value in the next sampling period.

We use NMPC to explicitly take constraints on the control inputs and states into account. The input and state constraints are chosen independently where the safety constraints are mixed. These are made precise in the following definitions:

Definition 3.1 *The* Admissibility consider a control system and the state and control constraint sets $\mathbb{Z} \subseteq Z$ and $\mathbb{U}(\mathbf{z}) \subseteq U$. *The states* $\mathbf{z} \in \mathbb{Z}$ *are called admissible states and the control values* $\mathbf{u} \in \mathbb{U}(\mathbf{z})$ *are called admissible control values for* $\mathbf{z}$. *The elements of the set* $\mathbb{Y} := \{(\mathbf{z}, \mathbf{u}) \in Z \times U | \mathbf{z} \in \mathbb{Z}, \mathbf{u} \in \mathbb{U}(\mathbf{z})\}$ *are called admissible pairs.*

Assumption 3.3 *The* Viability for each $\mathbf{z} \in \mathbb{Z}$ there exists $\mathbf{u} \in \mathbb{U}(\mathbf{z})$ such that $f(\mathbf{z}, \mathbf{u}) \in \mathbb{X}$ holds. *The property defined in this assumption is called viability or controlled forward invariance of* $\mathbb{Z}$. *It excludes the situation that there are states* $\mathbf{z} \in \mathbb{Z}$ *from which the trajectory leaves the set* $\mathbb{Z}$ *for all admissible control values. Hence, it ensures* $\mathbb{U}^N(\mathbf{z}_0) \neq \emptyset$ *for all* $\mathbf{z}_0 \in \mathbb{X}$ *and all* $N \in \mathbb{N}_\infty$. *This property is important to ensure the safety and feasibility of Safe Optimal Control Problem at* N^{th} *step,* $(\mathrm{SOCP_N})$: *the safe optimal control problem (SOCP* $_{\mathrm{N}}$*) is called feasible and safe for an initial value* $\mathbf{z}_0$ *if the set* $\mathbb{U}^N(\mathbf{z}_0)$ *over which we optimize is non empty. Viability of* $\mathbb{Z}$ *thus implies that* $(\mathrm{SOCP_N})$ *is feasible yet safe for each* $\mathbf{z}_0 \in \mathbb{Z}$ *and hence ensures that* $\mu_N(\mathbf{z})$ *is well defined for each* $\mathbf{z} \in \mathbb{Z}$.

Definition 3.2 [102] Mixed constraints *are given functions* $G_i^S : Z \times U \to \mathbb{R}, i \in \mathscr{E}^S = \{1, \ldots, p_g\}$ *and* $H_i^S : Z \times U \to \mathbb{R}, i \in \mathscr{I}^S = \{p_g + 1, \ldots, p_g + p_h\}$ *with* $p_g, p_h \in \mathbb{N}_0$, *we define the constraint sets* $\mathbb{Z}$ *and* $\mathbb{U}(\mathbf{z})$ *via*

$$\mathbb{Z} := \left\{ \mathbf{z} \in Z \middle| \begin{array}{l} \textit{there exists } \mathbf{u} \in U \textit{ with } G_i^S(\mathbf{z}, \mathbf{u}) = 0 \textit{ for all } i \in \mathscr{E}^S \\ \\ \textit{and } H_i^S(\mathbf{z}, \mathbf{u}) \geq 0 \textit{ for all } i \in \mathscr{I}^S \end{array} \right\}$$

and for $\mathbf{z} \in \mathbb{Z}$

$$\mathbb{U}(\mathbf{z}) := \left\{ \mathbf{u} \in U \middle| \begin{array}{l} G_i^S(\mathbf{z}, \mathbf{u}) = 0 \textit{ for all } i \in \mathscr{E}^S \textit{ and} \\ \\ H_i^S(\mathbf{z}, \mathbf{u}) \geq 0 \textit{ for all } i \in \mathscr{I}^S \end{array} \right\}$$

Here, the functions G_i^S *and* H_i^S *do not need to depend on both arguments. The functions* G_i^S, H_i^S *not depending on the input* $\mathbf{u}$ *are called pure state constraints, the functions* G_i^S H_i^S *not depending on* $\mathbf{z}$ *are called pure control constraints and the functions* G_i^S, H_i^S *depending on both* $\mathbf{z}$ *and* $\mathbf{u}$ *are called mixed constraints.*

The idea behind introducing these definitions and assumptions is that we want the trajectories to lie in inside the admissible sets and ensuring the safety based on the viability assumption when the corresponding control values are permissible at all time.

3.3.2 NMPC for aggressive maneuvering

The aim of the NMPC controller is to find the control policy where the states and inputs of the policy are within the admissible pairs in Definition 3.1 during the a task of aggressive maneuvering. Also, to ensure the safety of the maneuver, Assumption 3.3 require the vehicle to ensure there is an admissible input that can prevent the vehicle falling outside of the safety region in (2.18). There are 3 different types of constraints applied to the proposed NMPC which are defined in Definition 3.2. The main goal of model predictive control is to control the state of the vehicle system toward a reference trajectory and then keep it close to this reference with the vehicle's motion inside the safe region. An NMPC is designed on top of numerical optimization to solve the path following and maintain the vehicle motion within the computed safety envelope. To formulate the safe optimal control problem for the aggressive maneuvering, a discretized system of (2.16) with a fixed sampling period T_s is used

$$\boldsymbol{z}(k+1) = \boldsymbol{f}(\boldsymbol{z}(k), \boldsymbol{u}(k)),\ \boldsymbol{u}(k) = \boldsymbol{u}(k-1) + \Delta\boldsymbol{u}(k), \tag{3.17}$$

where $\mathbf{u}(k)$, $k \in \mathbb{N}$, is the input and $\Delta\mathbf{u}(k)$ is the changing amount of the input at the kth step. The following cost is minimized in the NMPC design.

$$J_\delta = \sum_{i=k}^{k+H_p} (l_1\|\boldsymbol{e}_i\|^2 + l_2\|\dot{\boldsymbol{e}}_i\|^2 + l_3\|\Delta\boldsymbol{u}_i\|^2) + l_4 J_m, \tag{3.18}$$

where l_j, $j = 1, \cdots, 4$, are constant weights, H_p is the predicting horizon, and cost J_m penalizes the possible robot motion outside $\boldsymbol{\Omega}_o$, namely,

$$J_m = \begin{cases} k_\beta \beta^2 + k_\psi \omega_\psi^2 & \text{if } \mathbf{z} = [\beta\ \omega_\psi]^T \notin \boldsymbol{\Omega_o} \\ 0 & \text{if } \mathbf{z} = [\beta\ \omega_\psi]^T \in \boldsymbol{\Omega_o} \end{cases}, \tag{3.19}$$

where $k_\beta, k_\psi > 0$ are the constant gains. The NMPC is then formulated as

$$\begin{aligned} \min_{\Delta u(t)} \quad & J_\delta \\ \text{subject to } \ & \mathbf{z}_{k+1,t} = \boldsymbol{f}(\mathbf{z}_{k,t}, \mathbf{u}_{k,t}) \\ & u_{\min} \le |\mathbf{u}_{k,t}| \le u_{\max}, \Delta u_{\min} \le |\Delta\boldsymbol{u}_{k,t}| \le \Delta u_{\max} \\ & \mathbf{u}_{k+1,t} = \Delta\boldsymbol{u}_{k+1,t} + \mathbf{u}_{k,t},\ k = 1, \ldots, H_p, \\ & (\beta_{k,t+H_p}, \omega_{\psi_{k,t+H_p}}) \in \boldsymbol{\Omega}_c. \end{aligned} \tag{3.20}$$

where $\mathbf{z}(t) = [\mathbf{z}_{t,t}\ \mathbf{z}_{t+1,t}\ \cdots \mathbf{z}_{t+H_p,t}]$ with $\mathbf{z}_{t,t} = \mathbf{z}(t)$ is the sequence of $\mathbf{z}(t)$ over the prediction horizon H_p at time t, and updated according to (3.17). Terms $\mathbf{u}_{k,t}$, $\Delta\mathbf{u}_{k,t}$, and $\mathbf{u}_{t,t} = \mathbf{u}(t)$ are the kth input sequence $\mathbf{u}_t$ and $\Delta\mathbf{u}_t$ respectively. At each time step t, the performance index (3.18) is minimized to obtain the optimal control input.

The use of J_m in the NMPC is inspired by a strategy to penalize the motion stability similar taken by the professional drivers. If the motion is within $\mathbf{\Omega}_o$, the control design follows a regular model predictive control. If the motion is outside $\mathbf{\Omega}_o$, the modified NMPC design tries to bring the motion back to the stable equilibrium at the origin.

3.3.3 Stability of NMPC

Nonlinear model predictive control, also known as model receding horizon, is expedient to deal with nonlinearities and constraints. Many of the stability analysis techniques which later turned out to be useful for NMPC, like Lyapunov function-based stability proofs or stabilizing terminal conditions were in fact first developed for linear MPC and later carried over to the NMPC. Also, for computational reasons the prediction horizon should be as short as possible, compatible with the expected performance and hardware, software limitations of the robotic vehicle. Although the main challenge is to guarantee closed-loop stability and performance also for small values of prediction horizon by an appropriate design of the constrained optimization engine it is possible to ensure closed-loop stability [103]. The NMPC proposed in this chapter the perturbation and uncertainties are explicitly taken into account using a domain of attraction and applying that to the optimization problem as constraints. Similarly, min–max MPC schemes building on game theoretic ideas or tube-based MPC schemes [104] have been developed to deal with model uncertainties. Later in Chapter 5 a learning based NMPC will be introduced for uncertainty approximations.

Convergence of NMPC

The proposed NMPC uses Nonlinear Conjugate Gradients (NCG) as a numerical dynamic programming algorithm to solve the constrained optimization problem of safe aggressive maneuvering. NCG, comparing to other dynamic programming methods,

needs lower memory requirements, and it has a superior local and global convergence rate [105]. The development of efficient NCG for constrained nonlinear systems is an active research area [106, 107].

In our abstract formulations of the NMPC Algorithm (3.20) only the first element $\Delta \boldsymbol{u}_{0,t}$ of the respective minimizing control sequence is used in each step, and the remaining entries $\Delta \boldsymbol{u}_{1,t}, \ldots, \Delta \boldsymbol{u}_{N-1,t}$ are discarded. In the practical implementation however, these entries play an important role because numerical optimization algorithms for solving the cost function usually work iteratively, starting from an initial guess $\Delta \boldsymbol{u}(\cdot)^0_{k,t}$ an optimization algorithm computes iterates $\Delta \boldsymbol{u}(\cdot)^i_{k,t}, i = 1, 2, \ldots$, converging to the minimizer $\Delta \boldsymbol{u}(\cdot)^*_{k,t}$ and a good choice of $\Delta \boldsymbol{u}(\cdot))^0_{k,t}$ is crucial in order to obtain fast convergence for this iteration, or even to ensure convergence, at all. Here, the minimizing sequence from the previous time step or using an initial guess by a simple kinematic base controller can be efficiently used to construct such a good initial guess.

Consider the finite horizon counterpart optimal control problem at N^{th} step of control system (3.20). For this method one can show that the optimal value function similar to Eq. (3.11) is of the form $V_N(\mathbf{E}) = \mathbf{E}^\top \mathbf{P}_N \mathbf{E}$ and that the matrix $\mathbf{P}_N$ has a unique symmetric and positive definite form discrete time algebraic Riccati equation where $\mathbf{P} \in \mathbb{R}^{4\times 4}$. In [102], it has been shown that as $N \to \infty$ the convergence implies that for each $\varepsilon > 0$ there exists $N_\varepsilon > 0$ such that the inequality

$$\left| \mathbf{E}^\top P_N \mathbf{E} - \mathbf{E}^\top P \mathbf{E} \right| \leq \varepsilon \|\mathbf{E}\|^2$$

holds for all $N \geq N_\varepsilon$. The convergence property of (3.20) and stability analysis in Section 3.2 delivers an stable feedback law from (3.7) and (3.20) the closed-loop control system is asymptotically stable.

Computational complexity and real time challenges of NMPC

NMPC solves the optimal control problems in real time using iterative nonlinear programming optimization methods, which are computationally heavy and practically challenging. Iterative methods use the gradient of the cost function to find the convergence direction during the optimization; however, to avoid falling into local minimum traps,

we will use the following assumptions.

Assumption 3.4 (Equilibrium endpoint constraint) *(i) The point* $\mathbf{z}_* \in \mathbb{Z}$ *is an equilibrium for an admissible control value* $\mathbf{u}_*$*, i.e., there exists a control value* $\mathbf{u}_* \in \mathbb{U}(\mathbf{z}_*)$ *with* $f(\mathbf{z}_*, \mathbf{u}_*) = \mathbf{z}_*$*. (ii) The stage cost* $\ell : Z \times U \to \mathbb{R}_0^+$ *satisfies* $\ell(\mathbf{z}_*, \mathbf{u}_*) = 0$ *for* $\mathbf{z}_*$ *from (i). Each admissible control sequence on the horizon* $N-1$ *can be extended to an admissible control sequence on the horizon* N *in real time.*

The statistical analysis for the NCG method in simulations shows the capability of the real time convergence of the algorithm. Figure 3.1 demonstrates the average convergence rate of the proposed NCG. The statistical comparison of NCG with Steepest Descent (SD), Particle Swarm Optimization (PSO), and modified Memetic Algorithms (MA) [108] proves that NCG can be used for real time purposes since the process time for an applicable convergence is acceptable.

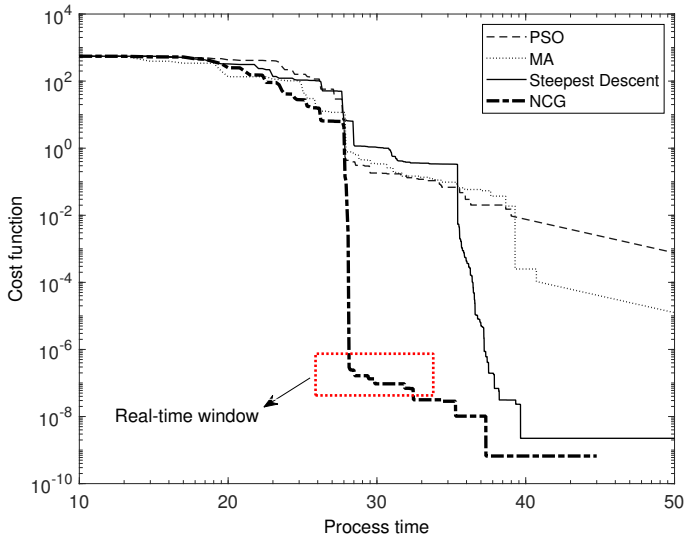


Figure 3.1: Statistical convergence comparison of NCG for a trajectory following problem over 100 different tries for path agile maneuvering in simulations.

3.4 Experimental results

RURacer-1 shown in Figure 2.5 was tested on a wooden track as shown in Figure 2.6(a). Details of the experimental setup and physical models are explained in Section 2.4.1. A

skilled RC vehicle driver was invited to conduct aggressive maneuvers. The driver has more than ten years of experience running RC race-cars. The human driver tried to follow any arbitrary trajectory and run the vehicle as fast as possible without losing the stability to achieve the minimum time lap on the race track shown in Figure 2.6(a). We collected the data from his experiments to compare with autonomous driver tests. The controller design then uses a vehicle's velocity profiles as one of the tracking targets, and the results are also compared with the human driver performance. For comparison purposes, we also implemented the Restrictive Boundary NMPC (RBNMPC) in [87] that uses the open-loop stability boundary in the MPC design.

We set up the desired trajectory $\boldsymbol{r}_d$ as the center-line of the track as shown in Figure 3.2. Under the control systems design, the vehicle tries to follow the desired trajectory within the shortest time while maintaining a safe motion. Figure 3.2 shows the performance under the proposed NMPC and the RBNMPC along with the trajectory under the human expert driver. The tracking errors under the NMPC are smaller than those under the RBNMPC and even the human driver. The traveling times under these two controllers and the human driver are listed in Table 3.1. We also list and compare the maneuver agility metrics under the two controllers and the human expert driver. The first agility metric is the accumulated lateral jerk, and the second is the accumulated relative lateral acceleration of the robotic vehicle [10]. Both metrics are calculated over the traveling distances. These comparisons demonstrate the higher agility and the shorter traveling time under the NMPC than those under the RBNMPC design. It is also clearly demonstrated that the human expert driver outperforms both autonomous controllers.

Table 3.1: Tracking performance comparison

Controller	Travel time (s)	Agility metric 1 (m/s^3)	Agility metric 2
NMPC	4.85	6.31	0.58
RBNMPC	5.21	4.83	0.46
Human	4.58	7.36	0.65

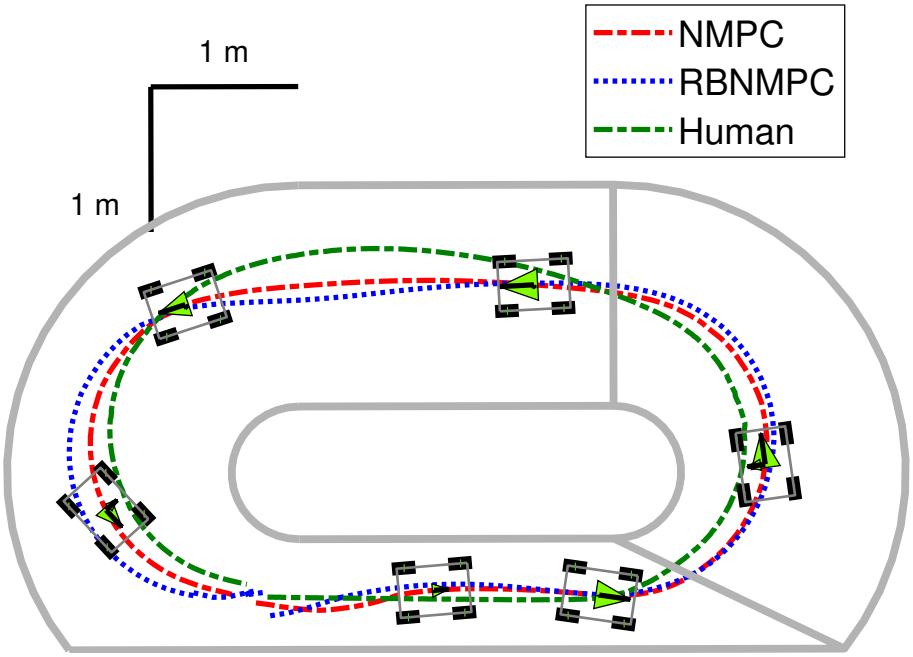


Figure 3.2: Trajectory comparison under two different autonomous control designs and human test.

Figure 3.3 further shows the vehicle motion comparisons under the two control designs along with the performance of the human expert driver. Figure 3.3(a) shows the longitudinal and lateral velocities (v_x and v_y) and Figure 3.3(b) demonstrates the yaw-rate (ω_ψ) comparisons. From both figures, we observe that under the NMPC, the magnitudes of the velocity and the yaw-rate are in general larger than those under the RBMPC design, while under the human expert driver, these values are then even larger than those of the NMPC. To clearly see such a trend, Figure 3.3(c) shows the slip angle comparisons under the two control designs and the human expert driver. Under the NMPC, the slip angle β reaches and keeps large values (around more than 10 degrees) over a long period, while the slip angle under the RBNMPC maintains relatively smaller values. Under the human expert driver, the slip angle values reach even larger magnitudes than those under the NMPC for some periods. This observation is similar to the comparison between the professional racing drivers, and the typical human drivers presented in [10].

It is also interesting that from the results shown in Figure 3.3(d), the NMPC generates larger steering inputs than those of RBNMPC (e.g., over 1.5-2.5 s and 3.2-4.8 s

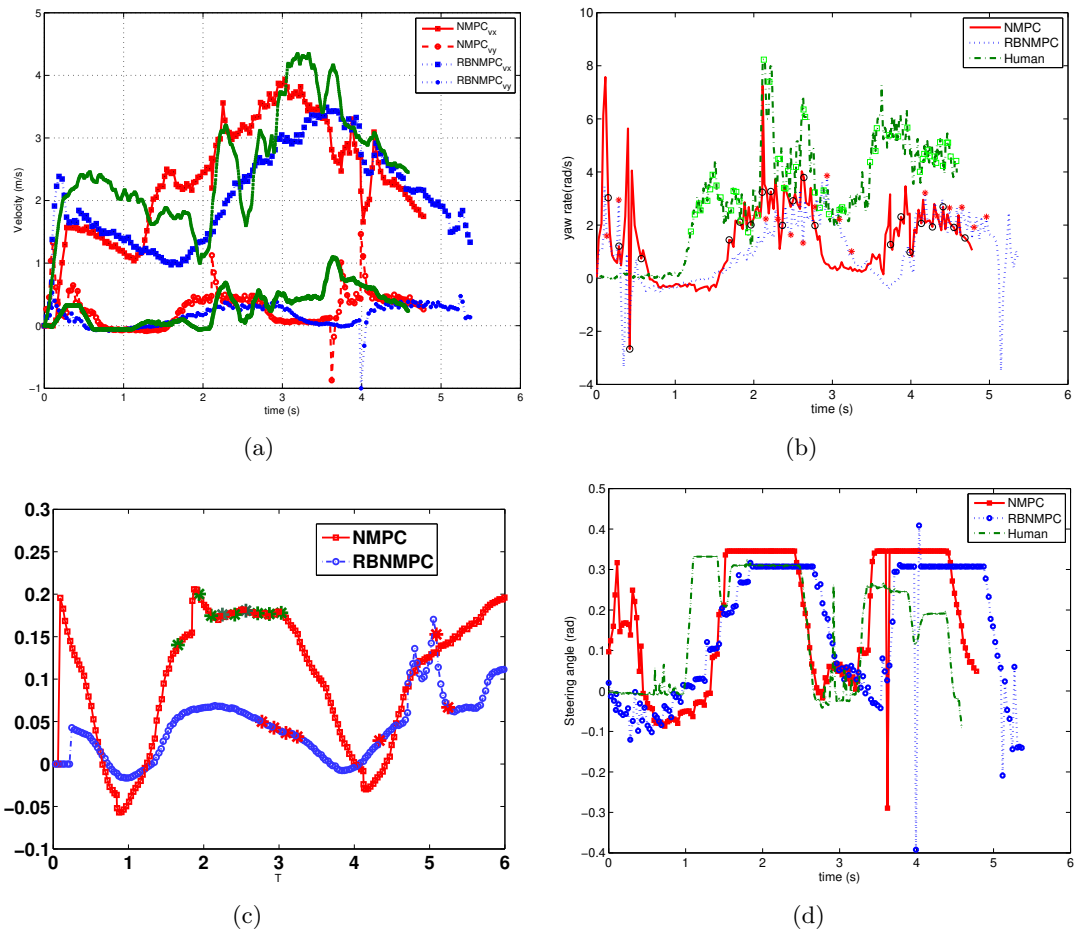
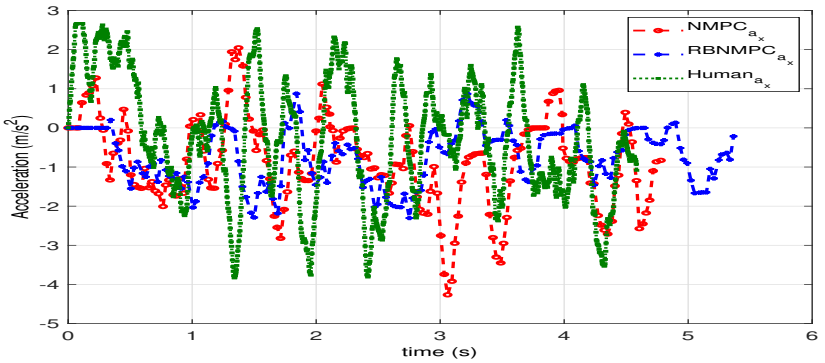
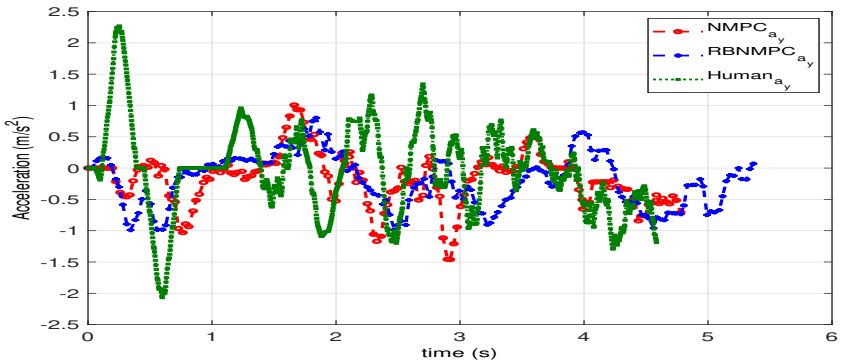


Figure 3.3: Comparison results under two autonomous controllers. (a) Vehicle velocity profiles. (b) Yaw rate ($\dot{\psi}$) profiles. The large circle, square, and star markers indicate the out of open-loop stability region at these moments. (c) The center slip angle β profiles. (d) Steering angle δ profiles.



(a)



(b)

Figure 3.4: Comparison results under two autonomous controllers. (a) Longitude acceleration profiles. (b) Lateral acceleration profiles.

periods). On the other hand, the human expert driver did not use any larger steering angles but instead, the duration of steering action is longer than that of the two controllers. Moreover, the human driver uses rapid changes in steering the vehicle. The saturated steering angle differences under the two controllers shown in Figure 3.3(d) are due to the augmented, large stability regions, that is, $\mathbf{\Omega}_o \subseteq \mathbf{\Omega}_c$. Figure 3.3(b) also shows the time instances (as large circle, square, and star markers) at which the vehicle motion is outside the stability region of the open-loop vehicle dynamics. The longitude and lateral acceleration profiles are also represented in Figure 3.4(a) and Figure 3.4(b), respectively.

3.5 Summary

In this chapter, a nonlinear model predictive controller on top of a feedback linearization controller was proposed to follow the desired trajectory and aggressive maneuvers within the safety region. NMPC was modeled as a constrained nonlinear programming problem and was solved in real time. NMPC useed the complete analytical model of the vehicle subjected to safety region and input limits. The controller allowed the vehicle to maximize the vehicle's maneuverability as thoroughly as a race-car driver. A proper dynamic model of the vehicle, tire, and its actuators provided more accurate prediction for the vehicle's future state for the NMPC. Experimental results proved that a proper dynamic model for FLC and NMPC, and the trajectory designed on the race track, the autonomous vehicle car can follow the trajectory with similar performance to an expert RC race-car driver.

Chapter 4

Motion Planning for Safe Aggressive Maneuvering

4.1 Introduction

During a aggressive or stunt maneuvering, professional drivers follow a certain sequence of motions for specific maneuvers. These motions are usually planned ahead of time, however, they are usually ready for any unexpected situation to reduce the risk. We can divide professional race car driving into two steps, planing and performing. A motion planning algorithm can be compared to professional driver's action schedule when the motion controller behaves as a synchronizing engine to keep the actions precise and safe. The problem of safe optimal motion planning requires computation of control commands and vehicle states to find risk free agile vehicle maneuvers. The resulting problem can be formulated as a constrained optimization process. This optimal control problem is subjected to the dynamical model of the vehicle and it's safety regions. Motion planning is computationally expensive and commonly unachievable in real time.

Optimal path planning for racing have been studied in the past a few years. A sequential optimization algorithm can generate race trajectories using an initial feasible trajectory [66]. However, this method has limitations since the initial path is not always accessible and it is not guaranteed to find the globally optimal trajectory due to using convex optimization methods. Using professional driving techniques to generate the trajectories similar to race car drivers take on the race track for autonomous vehicle maneuvering [109]. It is still challenging to find the time-optimal trajectory in real time even by assuming the initial feasible trajectory is accessible [110–112].

Sampling-based motion planning algorithms, such as Rapidly-exploring Random Trees (*RRT*) [113] and Sampling-Based Model Predictive Control (SBMPC) [114], efficiently find feasible trajectories for complex dynamical systems with constraints. These

algorithms have demonstrated the ability to produce feasible open-loop trajectories, but they cannot guarantee the produced trajectory's optimality. The development of the *RRT** algorithm ensures the asymptotically optimality [68] through the nearest and rewiring strategies in the tree node expansion process. However, since solving the boundary-value problem for a complex dynamic system is challenging, the *RRT** algorithm's rewiring mechanism cannot be implemented efficiently.

Recently, a modified *RRT** algorithm is implemented to find a minimum-time trajectory for autonomous high-speed driving of vehicles [70]. A simple particle model is used to transform the steering problem for the half-car model to reduce computations [27]. The sparse-RRT method does not use the rewiring process and instead provides asymptotic near-optimality for planning without access to a steering function [115]. Motion primitives are used to generate a feasible and fast trajectory for vehicle maneuvers, and optimality of the trajectory is, however, not guaranteed [38]. Randomized motion planning algorithms are also proposed to generate open-loop sequences of minimum-time motions for successful simulation [27, 70, 116]. A closed-loop RRT-based motion planning algorithm is implemented in [71] for real time autonomous vehicles, but the optimality of the trajectory is not considered and guaranteed. Feedback-based motion planning algorithms are developed in [117] with guaranteed control stability and safety.

This chapter's primary aim is to design an optimal motion planning algorithm for aggressive autonomous maneuvering for vehicle dynamics with drifting capabilities on a race track with physical constraints and track boundaries. Our motion planning method is based on the Rapidly-exploring Random Trees (RRT) [113], *RRT** [68], and Sparse-RRT [115]. The principal motives for this choice are: (i) Sampling-based algorithms apply to various type of nonlinear dynamic systems; (ii) the incremental optimization nature of *RRT** and complexity reduction using *Sparse-RRT* algorithm complies with near real time optimal path planning for the real world and implementation; (iii) sampling-based methods allow trajectory planning while imposing possibly very complex constraints on the states and inputs of the control system.

4.1.1 Rapidly-exploring random trees

(*RRT*) is a heuristic search method that builds up a tree where each branch is one iteration of a numerical simulation of a dynamical system [118, 119]. The simulated system at each iteration intends to generate a motion towards random nodes in the search space. The safety and feasibility of the dynamical system are ensured at each step of the simulation. Before adding the node to a tree, the safety and feasibility of the simulated branch are checked. Unlike the original *RRT* [113], which looks for the input sequence leading to the goal, this dissertation will find the most desired trajectory using an integrated controller. In order to plan the trajectory in real time, this method is required between each decision step to generate the vehicle control motions, which are possible to regenerate using NMPC. The advantage of using the controller integrated to *RRT* is its ability to apply to kinodynamic planning problems, which include both system and environmental uncertainties. The integrated NMPC furthers the development of this method to reconfigure the tree for rewiring in *RRT** [68]. Finding an asymptotically optimal solution using *RRT** for complex dynamical systems is not straight forward since it needs solving boundary value problems. Since there is no general solution for rewiring, this dissertation uses NMPC to generate local closed-loop simulations for the vehicle on the race track. It was implemented using a full vehicle model to ensure the vehicle's maneuver's precision and safety, where the safety region is found analytically. The proposed method reconnects the nodes together using NMPC. A drain function is implemented to remove extra nodes to reduce process burden, called Sparse-RRT* (*SRRT**). However, the implementation was not fast enough. For real time implementation; it is used in a near real-time test scenario on the edge computer.

4.2 Constrained optimal motion planning design

In this section, we present a motion planning approach that takes the advantages of the local steering algorithm [27] and the sparse search method with the integration of the NMPC [116]. One attractive property of the proposed motion planner is less computation difficulty and thus is suitable for real time applications. Indeed, we implemented

and demonstrated the motion planning algorithm using RURacer-1 and RURacer-2, on the race track. We also compare the performance under the autonomous driving control and the expert human driver. The main contribution lies in the design and implementation of a reliable motion planning scheme for autonomous vehicle aggressive maneuvers. The proposed sparse-based motion planner has adopted from [116], and the use of the MPC for the steering function is inspired by from [27]. We extend the work in [116] to take advantage of the steering function and that the rewiring process is used only for the dynamically reachable nodes to reduce the computational cost. The combined feedback linearization and NMPC design are used as the lower-level controller for robust motion performance that was explained in Chapter 3.

4.2.1 Motion planning formulations

Let compact sets $\boldsymbol{Z} \subset \mathbb{R}^n$ and $\boldsymbol{U} \subset \mathbb{R}^m$ denote the trajectories and the controls for system (2.16), respectively. Let $\boldsymbol{Z}_o \subset \boldsymbol{Z}$ and $\boldsymbol{Z}_g \subset \boldsymbol{Z}$ denote the obstacle and goal regions, respectively. The feasible region is then $\boldsymbol{Z}_f := \boldsymbol{Z}/\boldsymbol{Z}_o$. The vehicle is first located at $\boldsymbol{z}_0 \in \boldsymbol{Z}$ and must maintain inside feasible region $\boldsymbol{Z}_f$ and reach $\boldsymbol{Z}_g$. A feasible trajectory $\boldsymbol{\omega} \in \boldsymbol{Z}_f$ over duration T is a set of dynamic states of the vehicle system over $[0, T]$ which connect $\boldsymbol{z}_0$ to $\boldsymbol{Z}_g$ with corresponding controls $\boldsymbol{u}$. $\boldsymbol{\Omega}$ is the set of all feasible trajectories in $\boldsymbol{Z}_f$. The motion planning for given $\{\boldsymbol{Z}_f, \boldsymbol{U}, \boldsymbol{z}_0, \boldsymbol{Z}_g\}$ is to find trajectory $\boldsymbol{\omega} \in \boldsymbol{\Omega}$ while minimizing a cost function (i.e., traveling time). The cost function $c(\boldsymbol{z}) \neq 0$ for trajectory $\boldsymbol{z} \in \boldsymbol{Z}$ is defined as

$$J_c(\boldsymbol{z}) = \int_0^T c(\boldsymbol{z}(t))dt.$$

The RRT^* algorithm is a sampling-based motion planner that guarantees optimality [68]. The main approach of RRT^* is the use of the `Nearest` and `Rewire` procedures that need `Steering` function to connect two different nodes in the configuration space. `Nearest` checks the distance or cost between each vertex and its parent to find the closest node in the a neighborhood of vertices in a fixed radius from the new node. In `Rewire`, after a vertex has been connected to the cheapest neighbor, the neighbors are again examined for a accumulative cost reduction if the are being rewired to the

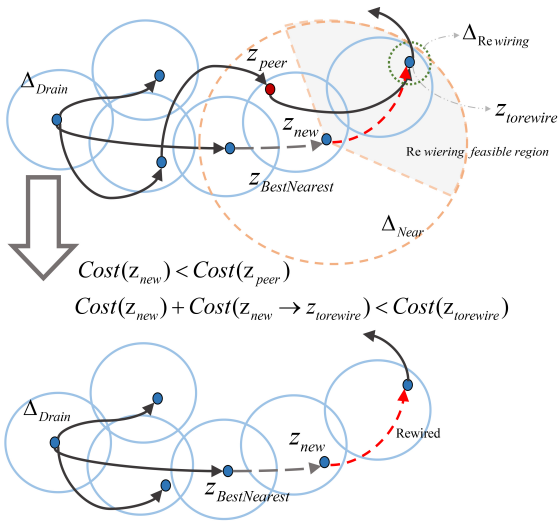


Figure 4.1: Drain and Rewiring methods after generating a new node is graphically presented in a selected picture.

newly added vertex. In the Sparse-RRT [116] algorithm, the `Nearest` function is used but `Rewire` is not used because of the difficulty to solve boundary-value problem. A `Drain` function is also used to remove the nodes with higher costs in neighboring sets and keeps the lowest-cost node. In our approach, we instead use the NMPC controller to connect the nodes.

We extend the *RRT** algorithm described in [70] and make modifications to add the attractive properties of the Sparse-*RRT* in [116] as a `Drain` function. Unlike the approach in [27] to connect the nodes in the rewiring process, the NMPC design is used to connect the nodes for the vehicle dynamic model. The *SRRT** is illustrated in Algorithm 4.1. Figure 4.1 also illustrates the construction of the `BestNearest`, `Drain` and `Rewire` processes in the *SRRT** design and these processes are described in the following discussions. Also, there are further modifications to the `Sample` and `Drain` algorithms to improve the convergence and reduce the computation complexity comparing to original *RRT* and *Sparse-RRT* methods.

`Sample:` Inspired from the MPC-tree method [114], we take the goal-directed samples in the steering function design. The goal-directed sampling process's main idea is to take a small percentage (e.g., 5%) of the samples biased toward the goal. Such a process would help to converge to the goal quickly. The sampled point is presented in

Algorithm 4.1: SRRT*$(\boldsymbol{Z}_g, \boldsymbol{U}, \boldsymbol{z}_0, \boldsymbol{Z}_f, i_{\max})$

```
1 𝕍 ← z_0, 𝔼 ← ∅, i ← 0;
  for i < i_max do
2     z_rand ← Sample(Z_f, Z_g);
3     z_near ← BestNearest(𝕍, z_rand, Δ_near);
4     z_new ← Steering(z_near, z_rand, Z_f);
      if z_new ≠ ∅ then
5         𝕍 ← 𝕍 ∪ {z_new}, 𝔼 ← 𝔼 ∪ {(z_near, z_new)};
6         𝕍, 𝔼 ← Rewire(𝕍, z_near, z_new);
7         𝕍, 𝔼 ← Drain(𝕍, z_new, Δ_Drain);
      end
8     i ← i + 1;
  end
```

Figure. 4.1 with an orange circle as $\boldsymbol{z}_{\text{rand}}$.

`BestNearest:` The main concept of the `BestNearest` function is illustrated in Figure 4.1. The circle around $\boldsymbol{z}_{\text{rand}}$ with a radius of Δ_{Near} defines the area that the algorithm searches for the neighbors. Similar to the RRT* [27] and Sparse-RRT [115] algorithms, the `BestNearest` function used in Algorithm 4.1 tries to find a node with the minimum cost within a vicinity Δ_{near} of $\boldsymbol{z}_{\text{new}}$. The vicinity is limited only to the nodes that are dynamically feasible to be steered from $\boldsymbol{z}_{\text{near}}$ to $\boldsymbol{z}_{\text{new}}$. The `BestNearest` function is illustrated in Algorithm 4.2. The `Near` function returns all the nodes in $\mathbb{V}$ that are in vicinity of $\boldsymbol{z}_{\text{rand}}$ within a radius of Δ_{near}. The nodes in $\mathbb{Z}_{\text{Near}}$ are restricted to those that are dynamically feasible through `FeasibleNear` function in the algorithm.

`Steering:` Unlike random propagation in [115], in the proposed *SRRT** algorithm, the nearest node $\boldsymbol{z}_{\text{Near}}$ is driven toward $\boldsymbol{z}_{\text{rand}}$ (usually close to $\boldsymbol{z}_{\text{new}}$) using also the `Steering` function, as illustrated in Figure 4.1. The advantage of using the `Steering` function is that when the system has an obstacle-free path to $\boldsymbol{Z}_g$, the goal-directed sampling helps a fast convergence to obtain the initial trajectory for real time applications.

`Rewire:` To find the nodes $\boldsymbol{z}_{\text{torewire}}$ that need to be rewired, the vicinity of $\boldsymbol{z}_{\text{new}}$ within a radius of Δ_{near} is used to restrict to the nodes that are dynamically feasible, as the gray circle area illustrated in Figure 4.1. We also use a vicinity around $\boldsymbol{z}_{\text{new}}$ within a radius of Δ_{rewire} as the converging area from node $\boldsymbol{z}_{\text{torewire}}$. The main difference and

advantage of the *SRRT** algorithm with the Sparse-RRT in [115] is the added `Rewire` function. The reason for not using the `Rewire` function in [115] lies in the difficulty to solve the boundary value problems. At each iteration, after sampling a new state and extending the tree towards the new state, the `Rewire` function attempts to re-assign the parent of each nearby node. The nearby nodes' vicinity is restricted to the ranges the NMPC can steer to reduce the number of attempts. In the *RRT** algorithm [27], the number of times to invoke the `Steering` function is $O(n \log n)$ and the collision check is implemented after the rewiring by the `Steering` function. To reduce the computational burden, the collision check is enforced while propagating. We also use the closed-loop NMPC to steer the system from one node to another with its inherent optimization formulation.

`Drain`: The main idea of the `Drain` function is to divide the search space into subsections and then use search methods to connect the subsections and find the trajectory. Comparing with other algorithms, the main difference in this work is that the subsections are chosen dynamically while the tree is growing. Unlike [116], the drain region Δ_{Drain} is selected regarding the attraction region of the dynamic system. In Figure 4.1, drain region Δ_{Drain} is illustrated as the blue circles where all the nodes in the vicinity of $\boldsymbol{z}_{\text{new}}$ within Δ_{Drain} are called $\boldsymbol{z}_{\text{peer}}$. The `Drain` function is illustrated in Algorithm 4.3. In Algorithm 4.3, $z.\texttt{children}$ is the number of the children of node $\boldsymbol{z}$ and $\mathbb{Z}_{\text{Useless}}$ is the set of the nodes which have a higher cost than the best-reached node and $\mathbb{Z}_{\text{Reached}}$ is the set of the nodes that reach the goal.

Algorithm 4.2: `BestNearest`$(\mathbb{V}, \boldsymbol{z}_{\text{rand}}, \Delta_{\text{near}})$

```
1 Z_Near ← Near(V, z_rand, Δ_near);
2 Z_Near ← FeasibleNear(Z_Near, z_rand);
  if Z_Near = ∅ then
3 |  Return : Nearest(V, z_rand);
  end
  else
4 |  Return : argmin_{z∈Z_Near} c(z) ;
  end
```

Algorithm 4.3: `Drain`$(\mathbb{V}, \boldsymbol{z}_{\text{new}}, \Delta_{\text{Drain}})$

```
1 Z_Drain ← Near(V, z_new, Δ_Drain);
  if c(z_new) ≥ argmin_{z∈Z_Drain} c(z) then
2     remove(z_new);
      else
          for z ∈ Z_Drain do
              if z.children = 0 then
3                 remove(z);
              end
          end
      end
  end
  if Z_Reached ≠ ∅ then
4     Z_Useless ← {z ∈ V | c(z) ≥ argmin_{z∈Z_Reached} c(z)};
      for z ∈ Z_Useless do
5         remove(z);
      end
  end
```

4.3 Complexity analysis and comparisons

4.3.1 Analysis of proposed motion planning

This section illustrates desirable properties of *SRRT** including the near-optimality property and the convergence rate. Then some simulations are given to show these performances of *SRRT** by comparing *RRT**, *SST* and *SRRT**.

Convergence rate and optimal properties

For the analysis proposes, some notions are introduced. $A_k^{(n)}$denotes a δ-similar trajectory to the kth segment is generated, where k is the number of the segment in the optimal trajectory and n is the number of iteration. $E_k^{(n)}$ denotes whether at least one optimal trajectory to the kth segment is generated from iteration 1 to n. [115] mentioned *BestNearest* eventually generates a δ-similar trajectory to any optimal trajectory and gave the following boundary condition:

$$P(E_k^{(n)}) \geq 1 - \prod_{j=1}^{n} P(E_{k-1}^{(j)}) \cdot \gamma \rho_{\delta \to \delta_{BN}} \tag{4.1}$$

where γ is a positive constant and $\rho_{\delta\to\delta_{BN}}$ denotes the probability for *Steering* function to generate a trajectory. Assume that the kth segment of the trajectory always meets the lower boundary of the probability function $P(E_k^{(n)})$. The ratio of the probabilities can be obtained between iteration n and $n-1$ while n approaches infinity.

$$\begin{aligned}\lim_{n\to\infty}\frac{|1-P(E_k^{(n)})|}{|1-P(E_k^{(n-1)})|} &= \frac{\prod_{j=1}^{n}(1-P(E_{k-1}^{(j)})\cdot\gamma\rho_{\delta\to\delta_{BN}})}{\prod_{j=1}^{n-1}(1-P(E_{k-1}^{(j)})\cdot\gamma\rho_{\delta\to\delta_{BN}})} \\ &= \lim_{n\to\infty}(1-P(E_{k-1}^{(n)})\cdot\gamma\rho_{\delta\to\delta_{BN}}).\end{aligned} \tag{4.2}$$

In addition, we have the following equation if a δ-similar trajectory to any optimal trajectory is generated.

$$\lim_{n\to\infty}P(E_{k-1}^{(n)}) = 1. \tag{4.3}$$

Applying these two equations (4.2) and (4.3), we can obtain:

$$\lim_{n\to\infty}\frac{|1-P(E_k^{(n)})|}{|1-P(E_k^{(n-1)})|} = 1-\gamma\cdot\rho_{\delta\to\delta_{BN}}. \tag{4.4}$$

The equation (4.4) states that *BestNearest* converges linearly to near-optimal solutions. [116] explained the experimental performance of *BestNearest* and *Drain* function in *SRRT** guarantees the near-optimality properties. This approach's complexity is mainly determined by two nearest neighbor queries, *BestNearest* and *Drain*. A sparse data structure is maintained in *SRRT**, so the computational burden to perform these queries is much smaller than other methods. The statistical analysis of the complexity and convergence are discussed for different test-beds in simulations.

4.3.2 Simulation results of SRRT*

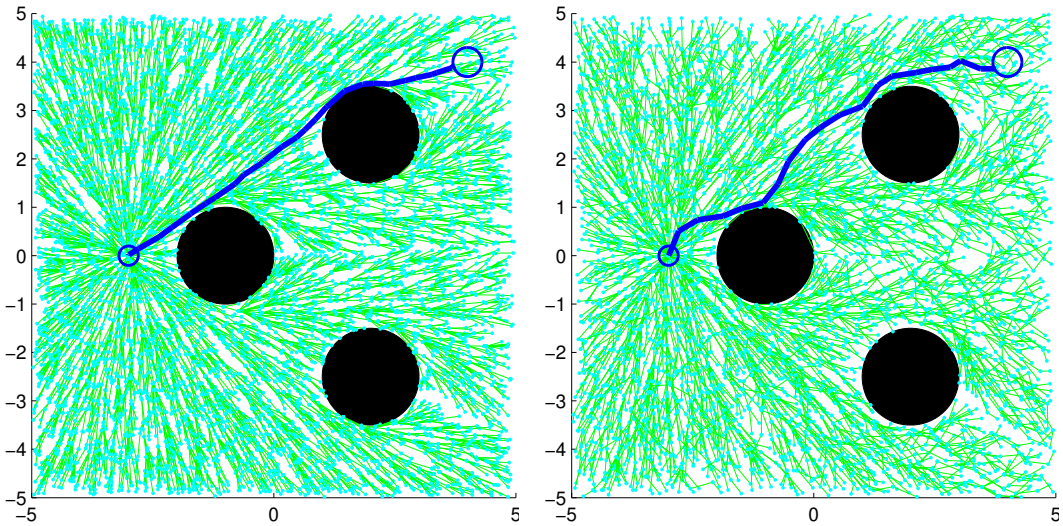
We conducted simulation comparison of the running time, a number of the nodes, and the convergence rate of the cost functions (i.e., traveling time or distance) among the *RRT**, *SST* and *SRRT** algorithms. The vehicle dynamics model and 2D model (i.e., dubins vehicle dynamics) were tested in simulations. Figure 4.3(a)-4.3(c) show the comparison results (average of 20 random runs) of these algorithms with the vehicle dynamics: the running time, the number of the nodes, and the convergence rate of the cost function. For the computation load, the *SRRT** outperforms slightly than the *SST*

and the *RRT**. The number of the nodes for the *RRT** is increasing linearly with the iterations, which reaches to 26740 nodes. The *SST* and the *SRRT** algorithms reach to 6535 nodes and 4024 nodes respectively after 8×104 iterations due to the sparse properties. The Cost of *SST* drops rapidly in the beginning, but it maintains at 3.34 seconds. The costs of the *SRRT** and the *RRT** finally maintain at the same lower-level, smaller than that of the *SST*. For $2D$ model, although the running time of *SST* is smaller than that of *SRRT**, the cost of *SST* remains much higher level than that of *SRRT**. Similar results can be drawn from Figure 4.2 that the distance of *SRRT** and *RRT** are very close while *SST* has a longer distance than other algorithms.

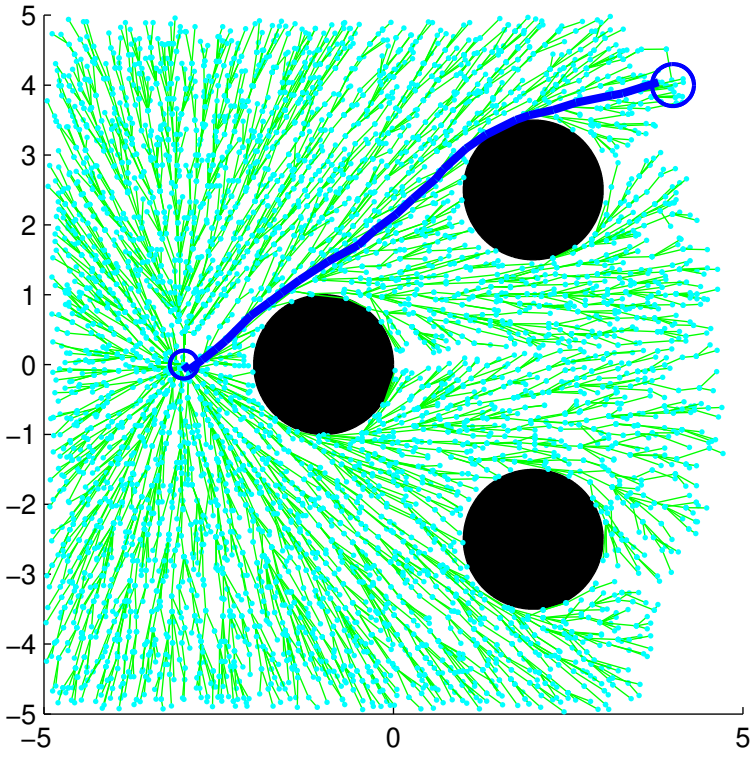
4.1 shows the simulation results under different algorithms for simple $2D$ and the U-turn vehicle maneuver. Similarly, the number of the nodes for the *SRRT** is less than the other two methods. The cost functions are the distance from the initial states to the goal and the traveling time. The length of the best path found by *SRRT** is slightly shorter than that of *SST* in the simulation of simple $2D$. As for vehicle dynamics, the traveling time of *SRRT** converges similar to that of the *RRT** while the traveling time of the *SST* is much larger than the other algorithms. Similar to that shown in Figure 4.3(a), the *SRRT** algorithm is faster than both the other two methods. All of these results demonstrate the attractive properties of the *SRRT** algorithm.

Table 4.1: Average results for simulating different algorithms foe same problem and 50000 iterations.

Problem	Method	Nodes	Computation Time	Cost
Simple 2D (Distance)	RRT*	45605	234 (s)	7.942 (m)
	SST	6543	154 (s)	8.098 (m)
	Spare-RRT*	5954	161 (s)	7.961 (m)
U-turn (Time)	RRT*	27350	412 (s)	2.87 (s)
	SST	6105	323 (s)	3.18 (s)
	SRRT*	4573	291 (s)	2.93 (s)



(a) (b)



(c)

Figure 4.2: Comparing the minimum distance motion planning results for a simple $2D$ particle model with 10000 iterations. (δ_{Near}=0.5, δ_{Drain}=0.05) (a) The standard *RRT**. 7820 nodes in 38 seconds while the cost is 8.006. (b) The *Sparse-RRT*. 4831 nodes in 31 seconds while the cost is 8.64. (c) The proposed method as *SRRT**. 4505 nodes in 33 seconds and 4523 times of rewiring while the cost is 8.016.

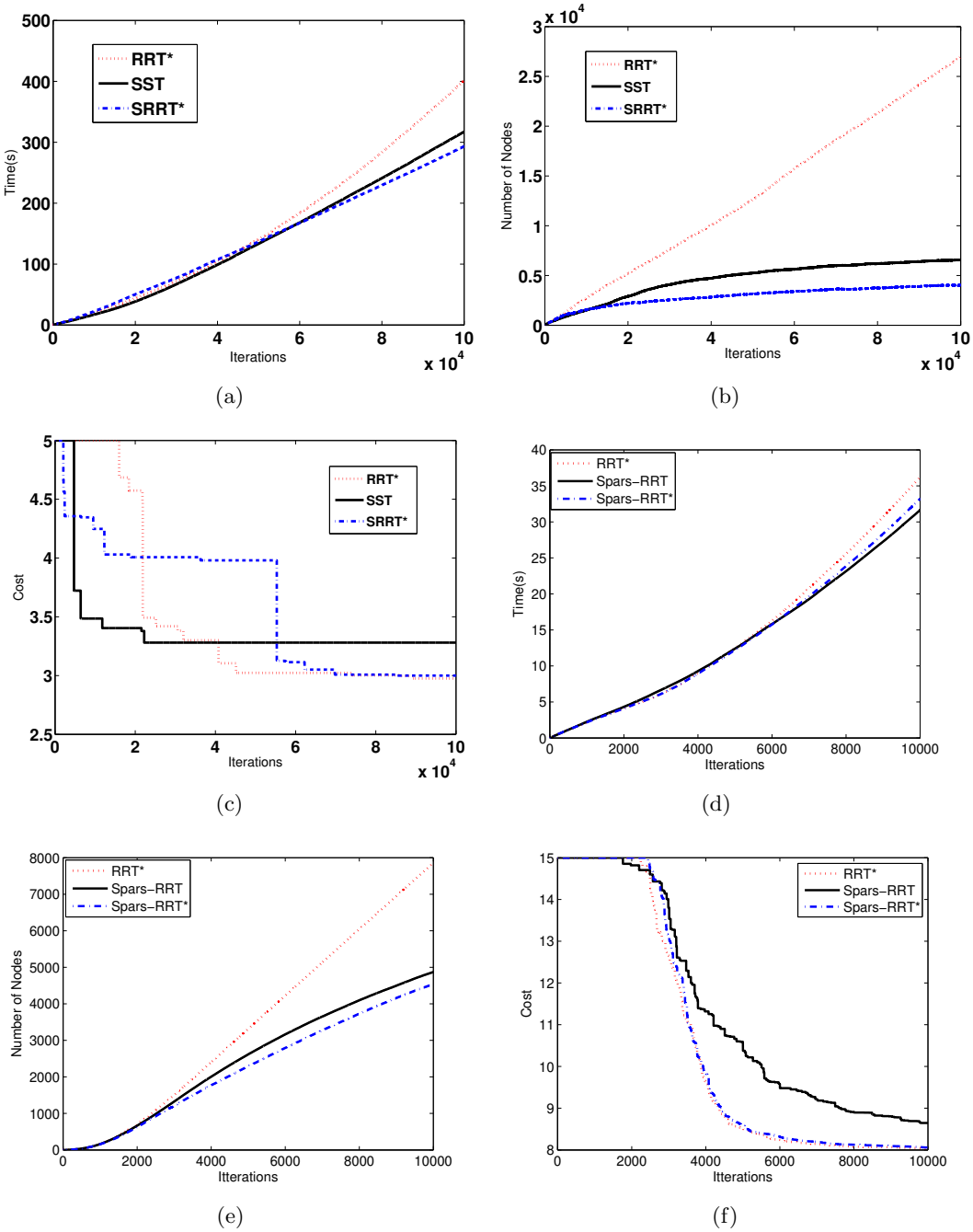


Figure 4.3: Comparing the number of the nodes, computational time and convergence of the cost function for vehicle motion planning. (a) The elapsed time of different methods for the car mode. (b) Number of the generated nodes for the car model. (c) Convergence of the cost function (Time) for the car model. (d) The elapsed time of different methods for the 2-D mode. (e) Number of the generated nodes for the 2-D model. (f) Convergence of the cost function (Distance) for the 2-D model.

4.4 Experimental results

Test 1: Minimum-time half of a Lap

Similar to Chapter 3, RURacer-1 in Figure 2.5 was used to test the race-car to follow the optimal trajectory on the track was used for the experiments which presented in Figure 2.6(a). The time-optimal trajectory computation by the *SRRT** was not fast enough for online motion planning. In our experiments, the optimal motion planning was first computed once. The robot will modify the plan when an unknown random obstacle obstructs the path, which was not applicable to this scenario. The NMPC was implemented to follow the minimum time trajectory computed by the *SRRT** off-line. Figure 4.4 shows the time-optimal trajectory generated by the *SRRT** algorithm (i.e., the solid line). The blue-dot line in the figure indicates the NMPC tracking results after the optimal trajectory was generated by *SRRT**. The main reason for conducting the NMPC in simulation was to check that the rewiring process's small tolerances do not cause any problems.

We again invited the expert RC vehicle driver to the campus and conducted driving experiments, and collected his test data for comparison purposes. His tests were conducted several times, and all motion data were recorded using the sensors and cameras. Taking the same starting and ending points, we ran the autonomous driving tests. Then the motion controller was implemented to track the optimal trajectory generated by the motion planner.

Figure 4.5 shows the comparison results of the motion planning under human expert driving and autonomous control. The *SRRT** was used to generate the time-optimal trajectory for the autonomous driving test $\boldsymbol{r}_d$ as shown in the figure. Under the motion control design, the vehicle followed the desired trajectory in the shortest time. As shown in Figure 4.5, the NMPC minimized the tracking error, and the trajectories under the motion controller and the human driver do not follow the motion planner result. Furthermore, we also computed and listed the traveling times under these two cases and the results are listed in Table 4.2. For comparison purposes, we also include the maneuver agility metrics in the table. These agility metrics include the accumulated

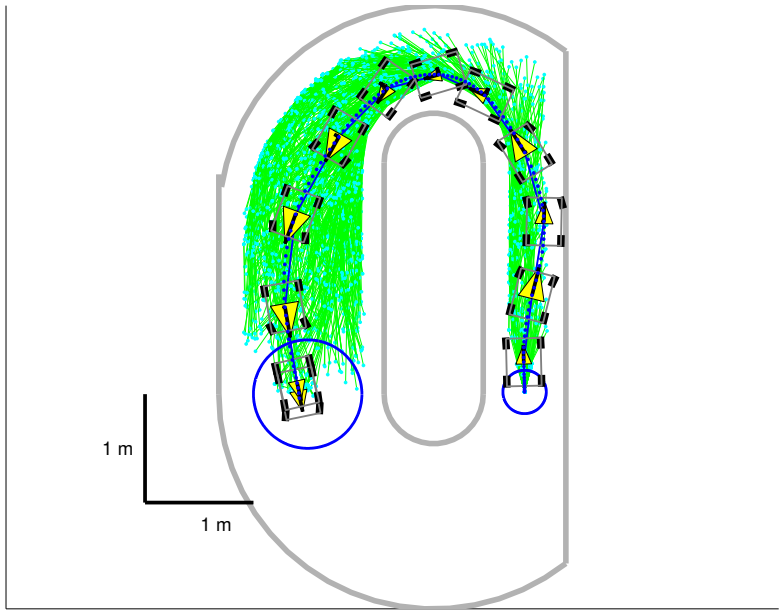


Figure 4.4: The time-optimal trajectory using the SRRT*. The results are obtained under $500,000$ iterations, $2,573$ nodes, $1,570$ rewiring and $601,260$ of drained nodes in $1,063.97$ seconds. ($\Delta_{\text{Near}} = 0.5$ and $\Delta_{\text{Drain}} = 0.1$). The optimal traveling time was found as 2.93 s.

lateral jerk and the accumulated relative lateral acceleration of the vehicle [10]. Both metrics listed in the table are calculated over the traveling distances. From those results, the autonomous controller achieves a shorter traveling time and higher agility than those by the human expert driver.

Table 4.2: Tracking performance comparison under two controllers and the human driver

Controller	Traveling time (s)	Agility metric 1 (m/s^3)	Agility metric 2
Proposed	3.18	5.31	0.58
Human	3.26	7.36	0.65

Figure 4.6 further shows the vehicle motion comparisons under the control design

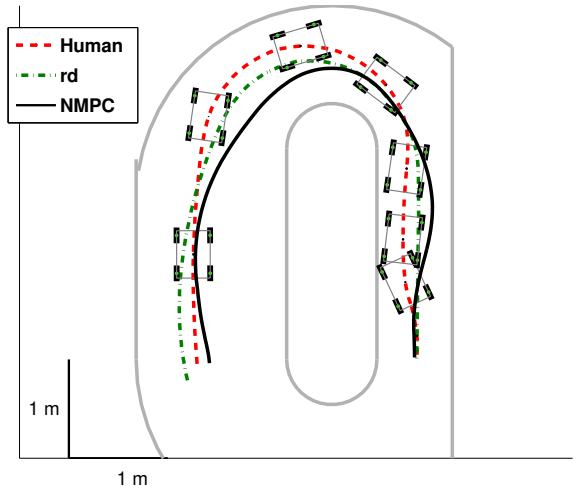


Figure 4.5: Trajectory comparison under two different autonomous control designs and human test.

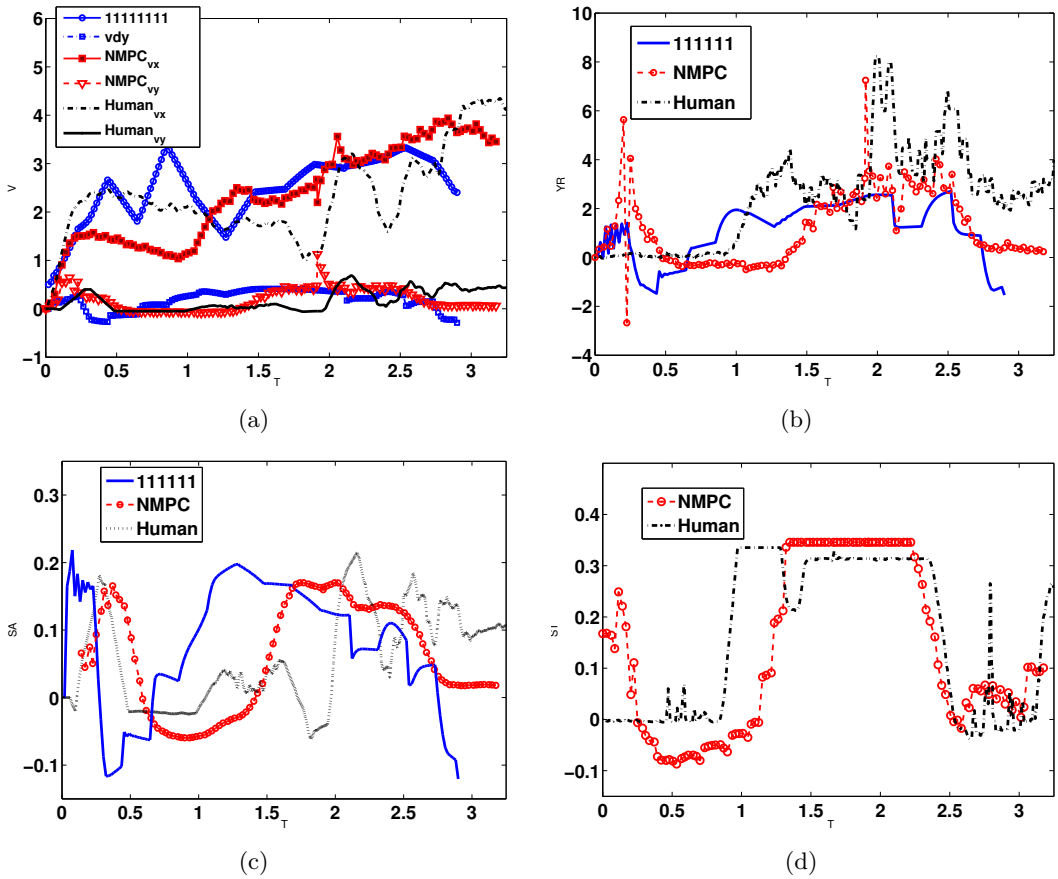


Figure 4.6: Experimental comparison results under the autonomous controller and the human expert driver. (a) Vehicle velocity profiles. (b) Yaw-Rate ($\dot{\psi}$) profiles. (c) The body center slip angle (β) profiles. (d) Steering angle (δ) profiles.

and human's test. Figure 4.6(a) shows the longitudinal and lateral velocities, Figure 4.6(b) demonstrates the yaw-rate ($\dot{\psi}$) comparisons, Figure 4.6(c) shows the slip angle comparisons and finally, Figure 4.6(d) shows the steering angle comparison. These figures clearly show that under the NMPC, the vehicle runs smoothly without quickly changing the steering and slip angles, which is observed by the performance of the human expert driver. Both the autonomous and human expert driving show large slip angles (around more than 10 degrees). This observation is similar to the comparison between the professional racing drivers, and the typical human drivers presented in [10].

Test 2: Minimum-Time Lap

We asked the same RC vehicle driver to conduct driving tests for the fastest lap for comparison purposes. We collected the results from several tests he performed and chose the best result for comparison. In this test the vehicle follows the optimal trajectory generated by the proposed *SRRT**. Then the proposed motion controller in Chapter 3 is implemented to track the optimal trajectory generated by the motion planner.

The time-optimal trajectory computation by the *SRRT** is not fast enough for online motion planning. In our experiments, the optimal motion planning was first computed off-line, and the motion controller was implemented to follow the trajectory. The NMPC was implemented to follow the minimum time trajectory computed by the *SRRT** off-line. Figure 4.7 shows the time-optimal trajectory generated by the *SRRT** algorithm (i.e., the solid line). The blue-dot line indicates the NMPC tracking results after the optimal trajectory was generated by *SRRT**.

The RC vehicle driver conducted new driving tests to reach the fastest lap time. The starting point of the vehicle was from one side of the track, and the goal was to reach the same area again. The human expert driving tests were conducted several times, and all motion data were recorded. The best result from his tests was compared with the autonomous controller test of NMPC following the optimal path found by *SRRT** shown in Figure 4.7.

Figure 4.8 shows the comparison results of the motion planning, human expert

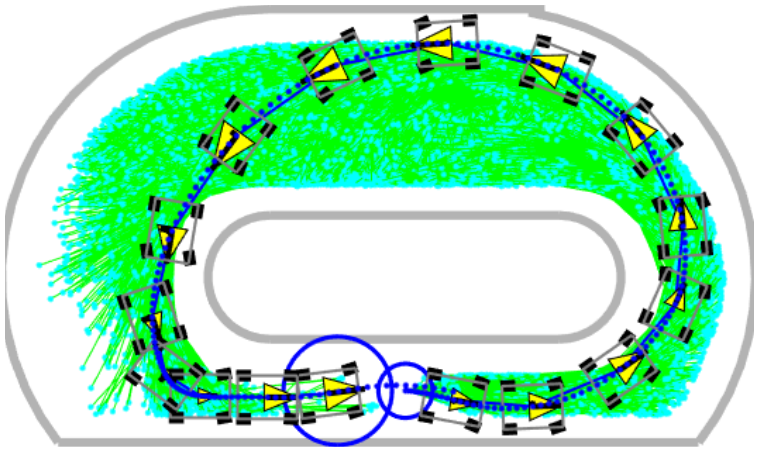


Figure 4.7: The time-optimal trajectory using the *SRRT**. The results are obtained under $500,000$ iterations, $2,573$ nodes, $1,570$ rewiring and $601,260$ of drained nodes in 9073.97 seconds. ($\Delta_{\text{Near}} = 0.5$ and $\Delta_{\text{Drain}} = 0.1$). The optimal traveling time is found as 4.31 s.

driving, and autonomous control. The *SRRT** was used to generate an initial time-optimal trajectory for the autonomous driving test $\boldsymbol{r}_d$ as shown in the figure. Under the motion control design, the vehicle tries to follow the desired trajectory within the shortest time. As shown in Figure 4.8, it is clear that the NMPC is based on minimizing the tracking error. The human expert driving trajectory is also plotted in the same figure. The trajectories under the motion controller and the human driver control do not follow the exact trajectory designed by the motion planner results. Furthermore, we computed and listed the traveling times under these two cases and the results are included in Table 4.3. For comparison purposes, we also listed the maneuver agility metrics in the table. These two agility metrics include the accumulated lateral jerk and the accumulated relative lateral acceleration of the vehicle [10]. Both metrics listed in the table are calculated over the traveling distances. It is clear from Table 4.3 that the autonomous controller achieves a shorter time and higher agility than those by the human expert driver.

Figure 4.9 further shows the vehicle motion comparisons under the control design and human control experiments. Figure 4.9(a) shows the longitudinal and lateral velocities, Figure 4.9(b) demonstrates the yaw-rate ($\dot{\psi}$) comparisons, Figure 4.9(c) shows the slip angle comparisons and finally, Figure 4.9(d) shows the steering angles comparison. These figures clearly show that under the NMPC, the vehicle runs *SRRT** smoothly

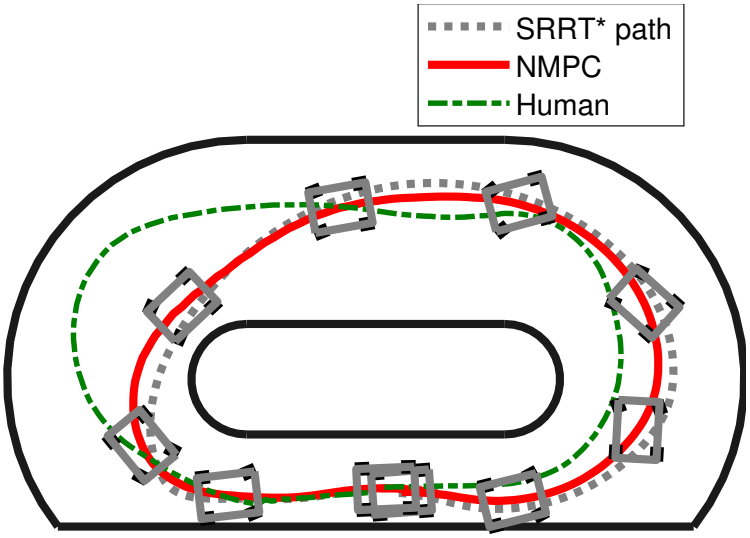


Figure 4.8: Test 2-Trajectory comparison under two different autonomous control designs and human test.

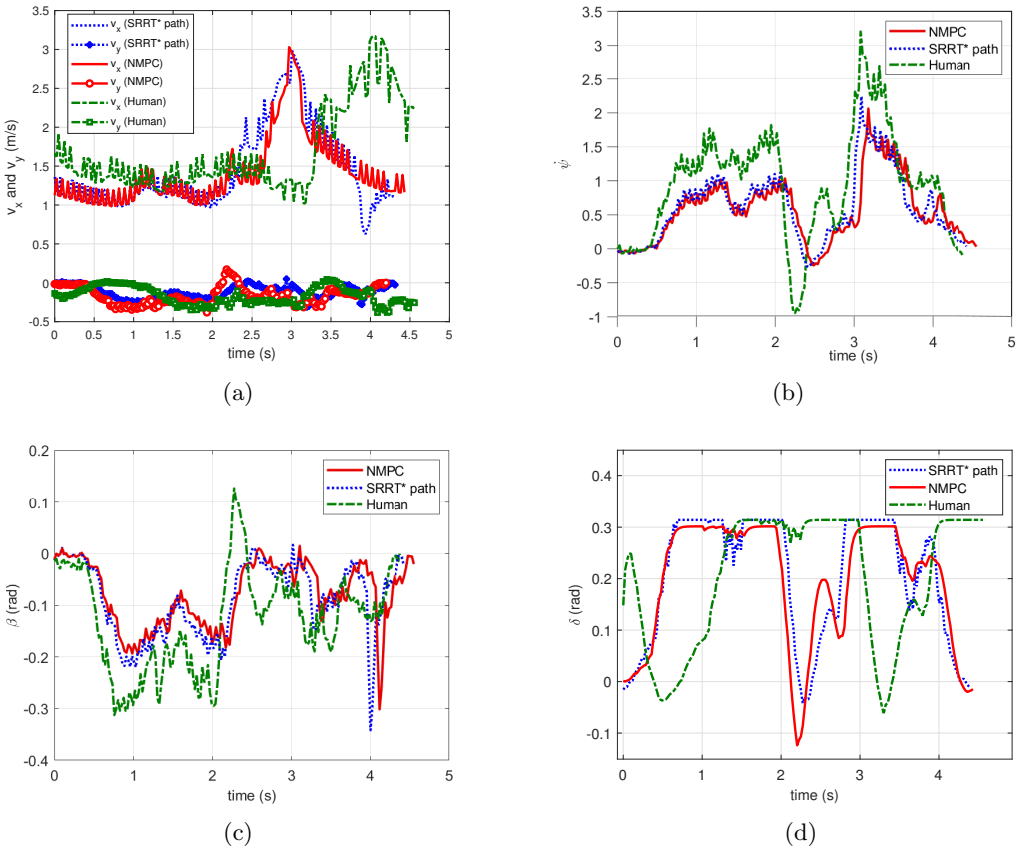


Figure 4.9: Test 2-Experimental comparison results under the autonomous controller and the human expert driver. (a) Vehicle velocity profiles. (b) Yaw-Rate ($\dot{\psi}$) profiles. (c) The body center slip angle (β) profiles. (d) Steering angle (δ) profiles.

Table 4.3: Test 2: Tracking performance comparison with planned trajectory, proposed controllers and human's control experiments

Controller	Finishing time (s)	Agility metric 1 (m/s^3)	Agility metric 2
SRRT* $\boldsymbol{r}_d$	4.31	7.51	0.59
Proposed	4.36	7.44	0.53
Human	4.49	7.68	0.65

without quickly changing the steering and slip angles, which can be seen by the performance of the human expert driver. Both the autonomous and human expert driving show large slip angles (around more than 12 degrees). This observation is similar to the comparison between the professional racing drivers, and the human drivers presented in [10].

4.5 Summary

We presented the *SRRT** motion planning algorithms for autonomous aggressive vehicle maneuvers. The *SRRT** motion planner took advantages of the *Sparse-RRT* and the *RRT** algorithms. The advantage of sparse property for motion planning helped to reduce the computational burden by removing un-useful nodes in the searching process. The attractive property of the *SRRT** lies in fast convergence to the optimal solution with less amount of computations. We implemented and compared our proposed algorithm with *Sparse-RRT* and *RRT** both in simulations and experiments. Our experimental goal was to find the minimum time trajectory on the race track. Comparison with the human expert driver, we could achieve a better lap time using *SRRT** for motion planning and NMPC as the motion controller. The experimental results have demonstrated the high agility maneuvering performance under the autonomous driving control with the motion planner.

Chapter 5

Learning Methods for Vehicle Safety Enhancement

5.1 Introduction

This chapter aims to provide a safe nonlinear predictive-learning controller to enable AMSF. The safety region for aggressive maneuverings introduced in Chapter 2 was utilized by the motion controller in Chapter 3 to safely follow the minimum time trajectory found by the motion planner discussed in Chapter 4. These methods work successfully unless there is an unexpected change in the vehicle model or the tire-road parameters. The uncertain essence of the tire-road friction can cause instability and dangerous behaviors unless these uncertainties are compensated for the real world experiments. This chapter provides an algorithmic approach that employs Lyapunov's stability theory for finding a stable search region for a real time learning predictive controller for AMSF.

Gaussian Process on Polynomial Basis (GPPB) learning method is used to improve the vehicle and tire model accuracy for enhancing the controller performance and safety region estimation proposed in Chapter 3. Sum Of Square (SOS) method is used to find a stable region considering uncertainties and disturbances to improve safety region estimations for AMSF. In [120, 121], the GP learning approaches are used to improve its dynamics model from data to safely increasing racing performance while the safety boundary for NMPC is not considered. Despite existing learning based methods which used GP for training the model, we used GPPB and integrated it with SOS to update the stability region. The estimation of the stability region is then used as safety constraints by NMPC for real time control of the vehicle.

In [122], the learning model predictive control uses the data from previous motions to improve its performance, while there is no proof for satisfying safety requirements. The studies in [63] reveal that autonomous vehicles can perform similar to professional

drivers to operate the vehicles outside, rather than within, the stability regions to achieve superior agility. Therefore, to achieve highly aggressive maneuvers as a safety feature, it is required to design a stable controller to relax the restrictions and allow the aggressive vehicle maneuvers for protecting vulnerable road users [123].

However, learning and analytical methods have proven to successfully control autonomous vehicles for aggressive maneuvering, ensuring a learning policy's safety and stability, and the scalability of an analytical method is still questionable. Due to the vehicle dynamics' complexity, verifying the stability and effect of uncertainties is typically inefficient if not infeasible. SOS is used to implement a verifiable predictive control [124, 125] and learning policy [126]. Numerical and data-driven methods are the only approaches for stability verification capable of dealing with complex control policies, such as learning-based controllers with model and environment uncertainties. SOS techniques provide an algorithmic method based on Semi-Definite Programming (SDP) to guarantee non-negativity of a constrained polynomial function such as the Lyapunov function to design control policies with stability analysis and estimation of the region of attraction.

SOS has provided a tractable algorithmic way to deal with constraints appearing in many control problems such as uncertainty compensation and determining the region of attraction. SOS has been recently applied to various control problems such as H_{inf} optimal control [127], sliding mode control [128], nonlinear robust stabilization [129], and the nonlinear SOS-MPC [124]. A two-step strategy utilizes SOS to propose MPC for input constrained nonlinear systems, which defines a set of policies and an on-line optimization [124]. SOS is experimentally validated for tracking a given trajectory on a severe torque limited underactuated double pendulum [130].

Similarly, it is demonstrated that drifting puts the system's states very close to an Unstable Equilibrium Condition (UEC) of the vehicle dynamic [74] and controlling the vehicle to maneuver. In contrast, the system's states are very close to a UEC requires a reliable and robust control policy with provable guaranteed stability. Due to the complexity of model and maneuvers, we are using a numerical optimization method specifically designed for stability verification called SOS instead of analytical Lyapunov

algorithm to find a controller with guaranteed stability.

5.2 Finding the stability region using SOS

This section explains the SOS-based design to estimate the safety region and calculate the state-space of the system where the learning-predictive control can guarantee the system's states will stay inside the safety region. The main goal of this method is to extend the definition of the *"safety"* region estimated by existing approaches [74] and maximize the feasible state space to design a safe motion controller to perform autonomous aggressive maneuvers.

Similar to the stability region discussed in Chapter 2, the SOS-based method is utilized to estimate the feasible and safe actuation space for the aggressive maneuverings. Similarly, the phase portrait of *yaw-rate/side-slip* ($\dot{\psi}/\beta$) is used to demonstrate the safety region of the closed-loop control systems [10]. The pseudo-code in Algorithm 5.1 describes how AMSF uses a tractable policy iteration method using SOS to find a feasible actuation space based on the nonlinear closed-loop system in (5.10) and use NMPC to enhance the performance and guaranteed safety.

The advantage of using the proposed numerical attraction funnel instead of a rigid boundary in [74] is that the funnel can take the system's uncertainty into account while using a closed-loop control system to maximize the feasible maneuvers for autonomous vehicle systems. Figure 5.1 compares the stable funnel found by the SOS method with the LQR based region found in Chapter 2 which is shown by the rectangle.

5.2.1 Reassert Vehicle Dynamics for Learning Predictive Control

The scaled vehicle, RURacer-2 prototype shown in Figure 2.1 is used for the experimental tests in this chapter. A new state-space representation of the vehicle model is formulated as an explicit distinct of a nominal system model $\boldsymbol{f}_n(\cdot)$ which represent the known part of the dynamical model and the unknown part of the model which is approximated using learning methods. These longitude and lateral forces are considered as uncertain part of the model which are identified using GPPB. Let's define the

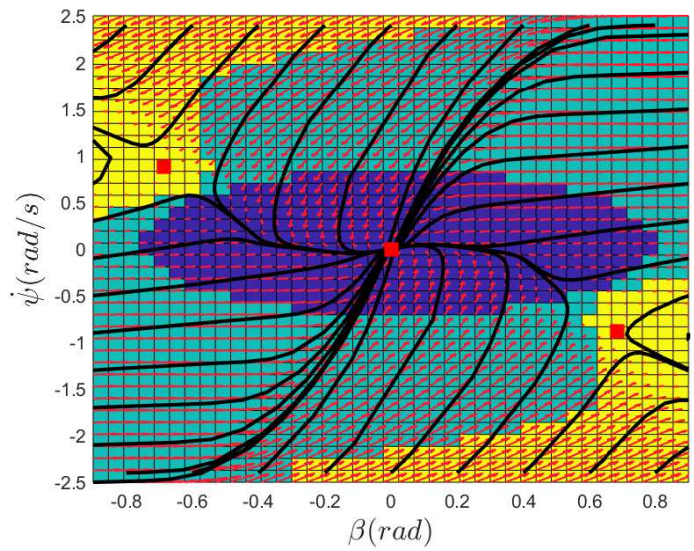


Figure 5.1: Stable region found by SOS and a GP model.

nominal model as

$$\dot{\mathbf{z}}(t) = \boldsymbol{f}_n(\mathbf{z}(t), \mathbf{u}(t)), \tag{5.1}$$

where $\mathbf{z} = [\mathbf{q}^T \ \mathbf{v}^T \ \boldsymbol{\omega}^T]^T$ is the state vector where $\mathbf{v} = [v_x \ v_y \ \dot{\psi}]$ and $\boldsymbol{\omega}$ is the vector of wheels velocity. $\mathbf{u} = [\delta \ \zeta]^T$ is the input vector, where δ is the steering wheel angle and ζ is the throttle rate which controls the motor's voltage. Using (2.2)-(2.15), the state space of the nominal model included with the electric motor in (5.1) is expressed as

$$\boldsymbol{f}_n = \begin{bmatrix} \mathbf{z}_2 \\ \mathbf{M}^{-1}\left[\mathbf{B}_x\mathbf{F}_x(\mathbf{z}_2, \mathbf{z}_3) + \mathbf{B}_y\mathbf{F}_y(\mathbf{z}_2, \delta) - \mathbf{C}(\mathbf{z}_1, \mathbf{z}_2)\right] \\ K_\zeta \zeta - K_m(\boldsymbol{\omega} + r_w\mathbf{F}_x) \end{bmatrix}. \tag{5.2}$$

More details about state-space model $\boldsymbol{f}_n$ in (5.1) is represented in Chapter 2. The complexity of the model has been a serious challenge for a precise and safe control design for agile maneuvers. In addition, the vehicle uses soft-rubber tires, and the motion heavily depends on the tire-road interactions and friction conditions. The longitudinal and lateral friction forces can be calculated using tire slip ratio and slip angle using Magic formula in Eq. (2.10). Still, the parameters are estimated and are not always constant during a trip and can cause uncertainties and disturbances of the system.

Algorithm 5.1: Safe nonlinear learning predictive control.

```
DATA: Initialize the Polynomial models, Gaussian process and SOS
  decomposition;
Results: Finds the initial stability region;
Initialization;
while AMSF==1 do
    Gaussian Process updates the Polynomial models;
    if Collected data is enough then
        Run SOS and evaluate the new safe aggressive maneuvering region Ω_s;
        Update the current model and safe aggressive maneuvering region Ω_s;
        else
            Continue Use NMPC with the current model;
        end
    end
end
```

Using GPPB learning algorithms to estimate longitudinal and lateral friction models is advantageous to enhance NMPC and design a nonlinear learning predictive controller.

$$\begin{cases} \mathcal{GPPB}_{F_{ijx}} : F_{ijx} = f_{F_{ijx}}(\mathbf{v}, \boldsymbol{\omega}), \ \mathcal{GPPB}_{F_{ijy}} : F_{ijy} = f_{F_{ijy}}(\mathbf{v}, \delta) \end{cases} \tag{5.3}$$

To formulate the GPPB we are making the following assumptions

Assumption 5.1 *The prior and the measurement noise distributions are Gaussian.*

Assumption 5.2 *Based on the universal approximation theory, the dynamical models have an estimation with special polynomial form. The accuracy of the polynomial function relates to the amount of data used to train the model.*

The proposed control design consists of four parts:

- A model-based feedback linearization to compute the desired tire friction forces using nominal and learning models.
- An inverse learning model to estimate the calculate the steering wheel angle and the throttle rate using the inverse of the tires-road models in (5.3).
- NMPC controller uses Nonlinear Conjugate gradients to predict and guarantee the controller's safety.

- SOS finds the safety region.

From Eq. (2.2), we formulate the vehicle's dynamical model in the matrix form as

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{q}, \mathbf{v}) = \mathbf{B}_x(\delta)\mathbf{F}_x(\mathbf{v}, \omega) + \mathbf{B}_y(\delta)\mathbf{F}_y(\mathbf{v}, \delta), \tag{5.4}$$

where the longitude force is a function of $\mathbf{F}_x(\mathbf{v}, \omega)$ and the lateral force is also a function of $\mathbf{F}_y(\mathbf{v}, \delta)$. GPPB is used to approximate the longitude and lateral forces shown in Eq. (5.3). We obtain the motor voltages as the inputs of the system to design a direct learning-based controller. A simplified model for the electric motor and the power transmission system in Eqs. (2.11)-(2.15) is written in the below matrix form as

$$\dot{\boldsymbol{\omega}} + K_m(\boldsymbol{\omega} + r_w\mathbf{F}_x) = K_\zeta \zeta, \tag{5.5}$$

where K_ζ is the scaling factor for the throttle rate to motor voltage and K_m is a nominal constant matrix obtained from motor and transmission system parameters. Suppose that the initial training data set for input-output controller uses the learning data pairs and updates during the experimental tests. F_{ijx}, and F_{ify} are estimated using the GPPB in (5.3) to enhances the model and the controller performance.

5.2.2 Redesign feedback linearization

Similar to Section 3.2, a modified version of FLC is outlined in this section. The new design of FLC finds longitude and lateral forces where in Section 3.2 FLC is designed to find longitude forces. The hybrid FLC-NMPC control design uses the GPPB to estimates the tire-road friction model based on the Assumption 5.3.

Assumption 5.3 *The system's states, $\boldsymbol{z}$ are observable using sensors or any estimation method and the robot's dynamical model is available. The motor model is evaluated with a fair amount of accuracy, and the only uncertain part of the system is the tire-road interaction model.*

Let's rewrite the model in (3.6) as

$$\ddot{\mathbf{r}}_P = \mathbf{\Gamma}_x\mathbf{F}_x + \mathbf{\Gamma}_y\mathbf{F}_y + \boldsymbol{\vartheta} \tag{5.6}$$

where $\mathbf{\Gamma}_x = \mathbf{\Lambda}\mathbf{M}^{-1}\mathbf{B}_x$ and $\mathbf{\Gamma}_y = \mathbf{\Lambda}\mathbf{M}^{-1}\mathbf{B}_y$ and $\boldsymbol{\vartheta} = (\dot{\mathbf{\Lambda}} - \mathbf{\Lambda}\mathbf{M}^{-1}\mathbf{C})\mathbf{v}$. Using FLC The force vectors $\mathbf{F}_x$ and $\mathbf{F}_y$ can be obtained as

$$\begin{cases} \mathbf{F}_x = \mathbf{\Gamma}_x^{-1}[\ddot{\mathbf{r}}_d + \boldsymbol{K}_d\dot{\mathbf{e}} + \boldsymbol{K}_p\mathbf{e} - \hat{\boldsymbol{\vartheta}}_x] \\ \mathbf{F}_y = \mathbf{\Gamma}_y^{-1}[\ddot{\mathbf{r}}_d + \boldsymbol{K}_d\dot{\mathbf{e}} + \boldsymbol{K}_p\mathbf{e} - \hat{\boldsymbol{\vartheta}}_y] \end{cases} \tag{5.7}$$

where error is $\mathbf{e} = \mathbf{r}_d - \mathbf{r}_p$ and $\dot{\mathbf{e}}$ and $\ddot{\mathbf{e}}$ are the first and second derivative of the error. $\hat{\boldsymbol{\vartheta}}_x$ and $\hat{\boldsymbol{\vartheta}}_y$ are the estimation of $\boldsymbol{\vartheta}_x = \boldsymbol{\vartheta} + \mathbf{\Gamma}_y\mathbf{F}_y$ and $\boldsymbol{\vartheta}_y = \boldsymbol{\vartheta} + \mathbf{\Gamma}_x\mathbf{F}_x$, respectively. Based on the Assumption 5.3, $\mathbf{\Gamma}_x$ and $\mathbf{\Gamma}_y$ are known and the longitude and lateral forces can be estimated using (5.3). $\hat{\boldsymbol{\vartheta}}_x$ and $\hat{\boldsymbol{\vartheta}}_y$ can be written as

$$\begin{cases} \hat{\boldsymbol{\vartheta}}_x = \dot{\mathbf{\Lambda}}\mathbf{v} + \mathbf{\Lambda}\mathbf{M}^{-1}\mathbf{B}_y f_{F_y}(\mathbf{v}, \delta), \;\; \hat{\boldsymbol{\vartheta}}_y = \dot{\mathbf{\Lambda}}\mathbf{v} + \mathbf{\Lambda}\mathbf{M}^{-1}\mathbf{B}_y f_{F_y}(\mathbf{v}, \delta) \end{cases} \tag{5.8}$$

$\mathbf{F}_x$ and $\mathbf{F}_y$ can be controlled implicitly using the the steering wheel δ and throttle ζ using an inverse GPPP model as

$$\begin{cases} \mathcal{GPPB}_\delta : \delta = f_\delta^{-1}(\mathbf{v}, \boldsymbol{F}_y), \;\; \mathcal{GPPB}_\zeta : \zeta = f_\zeta^{-1}(\mathbf{v}, \delta, \boldsymbol{\omega}, \boldsymbol{F}_x) \end{cases} \tag{5.9}$$

Substituting the control law (5.7) in to (5.6) and using (5.9) forms the dynamic of the closed-loop control system as,

$$\ddot{\mathbf{e}} + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e} = \eta(t) \tag{5.10}$$

where $\eta(t) = \boldsymbol{\vartheta} - \hat{\boldsymbol{\vartheta}}$ is called lumped uncertainty and can also be approximated by the polynomials to any degree of accuracy using GPPB. As it is discussed in Chapter 3, the lumped uncertainty $\eta(t)$ is bounded and the error will converge to a small value in Eq. (3.15).

Assumption 5.4 *$\eta(t)$ is bounded with an upper limit as $||\eta(t)|| \leq \epsilon$.*

Longitude and lateral forces are controller indirectly using inverse learning control method in (5.9). The steering wheel and throttle control commands found by inverse learning method are subjected to the FLC control law in (5.7) as constraints. Knowing the upper bound of the uncertainty $\eta(t)$ and using SOS we are able to find the stable

region numerically for the worst-case scenario of the uncertainties. Similar to Chapter 3, NMPC can predict the future states of the system and guarantee the system will stay inside the safety region approximated by SOS.

5.2.3 SOS estimation of the safety region

This section briefly describes how SOS enables us to estimate the attraction region for the closed-loop control system with a bounded uncertainty region. In order to compare the SOS numerical method with the analytical method, we propose the use of composite Lyapunov functions. Considering the upper bound of the uncertainties, a funnel of attraction region, instead of a rigid attraction region, is calculated by defining invariant sets of variables. We consider a fixed Lyapunov function during each SOS optimization step. In the case of $\boldsymbol{x}_{sos} = [\dot{\mathbf{e}}; \mathbf{e}]$ in (3.10) and using similar equations for the closed-loop system (3.9), we can write

$$\dot{\boldsymbol{x}}_{sos} = \boldsymbol{A}_{sos}\boldsymbol{x}_{sos} + \boldsymbol{b}_{sos}\boldsymbol{u}_{sos} \tag{5.11}$$

where $\boldsymbol{A}_{sos} = \boldsymbol{A}_L$, $\boldsymbol{b}_{sos} = \boldsymbol{B}_L$ from Eq. (3.10) and $\boldsymbol{u}_{sos} = \eta(t)$. The closed loop form of the system in (5.10) demonstrates that $\eta(t)$ is a function of $\boldsymbol{x}_{sos}$ a and $\dot{\boldsymbol{x}}_{sos}$ and can be estimated as $\mathcal{GPPB}_\eta : \eta = f_\eta(\dot{\boldsymbol{x}}_{sos}, \boldsymbol{x}_{sos})$ using GPPB and we consider:

Assumption 5.5 *The origin is an equilibrium point of f_η with $f_\eta(0,0) = 0$, which means $\{\eta(t) \to 0$ if $\mathbf{e}, \dot{\mathbf{e}}, \ddot{\mathbf{e}} \to 0\}$ from Eq.(5.10).*

Lemma 5.1 *A two-times continuously differentiable Lyapunov function $\mathcal{V}(x_{sos})$ is given. Moreover, there exists a constant $c_s > 0$ such that $\frac{\partial V(x_{sos})}{\partial x_{sos}}(\boldsymbol{A}_{sos}\boldsymbol{x}_{sos} + \boldsymbol{b}_{sos}f_\eta) < 0$ for all $\boldsymbol{x}_{sos} \in \Omega_s = \mathcal{V}(c_s)$, where $\mathcal{V}(c) = \{\boldsymbol{x}_{sos} \in \mathcal{X}_{sos} | \mathcal{V}(\boldsymbol{x}_{sos}) \leq c\}$.*

Knowing that driving motor's input voltage is bounded and it capable of generating bounded torques we can assume tire forces will be bounded at all times. Also, the tire models presented in Section 2 shows that the tire forces are bounded to the coupling circle's limits which the limited force will generate finite acceleration and speed for the wheels and the vehicles. Knowing the states of the system are limited and using the BIBO theory, $\mathcal{V}(x_{sos})$ should be bounded in a finite time horizon [53].

Lemma 5.2 *The function f_η is Lipschitz continuous and bounded by η_{max}, where η_{max} the upper limit of the uncertainties winch can be considered as the worst-case scenario for the stability region estimations. This can be proved using the boundlessness of the system inputs and states.*

The safety region Ω_s is the inner region of that, which is larger than the open-loop stability rectangle shown in Figure 2.3(a). Using Assumptions 5.4 and 5.5, it is guaranteed that any initial condition starts inside the safe region $\boldsymbol{x}_{sos}(0) \in \Omega_s$ would stay inside the $\{\boldsymbol{x}_{sos}(t) \in \Omega_s \ \forall \ t > 0\}$ by using controller (5.9) subjected to FLC rules in (5.7). SOS is a nonparametric optimization method from machine learning, where the goal is to find an approximation of a nonlinear map from a state $\boldsymbol{x}$ to the function value $\dot{\boldsymbol{x}}_{sos} = \boldsymbol{y}(t) = f(\boldsymbol{x}_{sos}, \boldsymbol{u}_{sos})$ in (5.11). For any bounded initial state and bounded input value the system states will be bounded which there will be a finite value $\alpha > 0$ such that the signal magnitude never exceeds $\|\boldsymbol{y}(t)\| < \alpha$.

Assumption 5.6 *Domain $\mathcal{X}_{sos}$ is a sub-domain of the safety region $\mathcal{X}_{sos} \subseteq \Omega_s$ if $\{\|\boldsymbol{y}(t)\| < \alpha \ \forall \ t > 0, \ \boldsymbol{x}_{sos}(0) \in \mathcal{X}_{sos}, \boldsymbol{u}_{sos}(t) \in \mathcal{U}_{sos}\}$, where $\mathcal{X}_{sos}$ and $\mathcal{U}_{sos}$ are bounded domains for the states and input signals.*

To formulate Assumption 5.6 using the attractive feature of SOS that are optimizing over it to cast as a tractable semi-definite programming of nonlinear systems estimated by polynomials. Indeed, it is known that a polynomial function discussed later in Eq. (5.14) of degree d is even more instrumental for control applications. The fact that when a polynomial function is not precisely determined but its coefficients are otherwise affinely parameterized in terms of some unknowns, the search for coefficient values exploited for algorithmically constructing Lyapunov functions for an approximated nonlinear systems using SOS by

$$\mathcal{V} = \left\{ \mathcal{V}(\boldsymbol{x}_{sos}) : \mathcal{V}(\boldsymbol{x}_{sos}) = v_0(\boldsymbol{x}_{sos}) + \sum_{i=1}^{m} c_i v_i(\boldsymbol{x}_{sos}) \right\} \tag{5.12}$$

where, $c_l, ..., c_m \in \mathrm{R}$ are real coefficients and $v_i(x_{sos})$ are some monomials. The search for a $\mathcal{V}(x_{sos}) - \phi(x_{sos}) > 0$ and $-\dot{\mathcal{V}}(x_{sos}) > 0$ performed using semi-definite programming where $\phi(x_{sos})$ is some positive definite polynomial. GP approximates the system's

polynomials as an estimator for the closed-loop control system and SOS designs the tractable convex optimization technique to find the safety region Ω_s for the predictive control of the vehicle.

$$\begin{aligned} &\text{minimize} \quad \textstyle\sum_{i=1}^{m} a_i c_i \\ &\text{subject to} \qquad v^T \left(F_0(x_{sos}) + \textstyle\sum_{i=1}^{m} c_i F_i(x_{sos})\right) v \text{ is a sum of squares,} \end{aligned} \tag{5.13}$$

where $a_i \in \mathrm{R}$ are fixed coefficients, and $c_i \in \mathrm{R}$ are decision variables, and $F_i(x_{sos})$ are some symmetric matrix functions of the state variables $\mathbf{x}_{sos} \in \mathrm{R}^n$. In Algorithm 5.2, semi-definite programming is used in Matlab and *SOSTOOLS* to find the a subset of state space with guaranteed stability $\dot{\mathcal{V}} = \frac{\partial \mathcal{V}(x_{sos})}{\partial x_{sos}} \left(\boldsymbol{A}_{sos}\boldsymbol{x}_{sos} + \boldsymbol{b}_{sos} f_\eta\right) < 0$.

Algorithm 5.2: Safe region exploration

```
Star: Initialize the model (5.11) and the polynomial model for f_η(·);
Results: Initialize safety region Ω_s = Ω_c (Ω_c is he closed-loop safety region in
  Section 2.3);
Initialization;
while (Update Ω_s)==1 do
    Choose an initial state x_sos ∈ X_sos;
    Construct a Lyapunov function using (5.12);
    if Assumption 5.6 is true and V̇ < 0 ∀ t > 0 then
        Update Ω_s ← Ω_s ⋃ X_sos.;
        else
        end
        Continue Choose a new sob-domain X_sos.;
    end
end
```

5.3 Learning framework for predictive controller

According to Assumption 5.2, we can take advantage of Bayesian learning to actively learn the unknown part of the model-dynamics from collected data in experiments. The learning predictive controller is subjected to the SOS-based safety region as a constraint to guarantee the stability and safety of the vehicle's aggressive maneuvering with uncertain tire models. Figure 5.2 shows the diagram of the proposed safe learning-predictive controller. Gaussian Process is used to directly calculate the tire forces

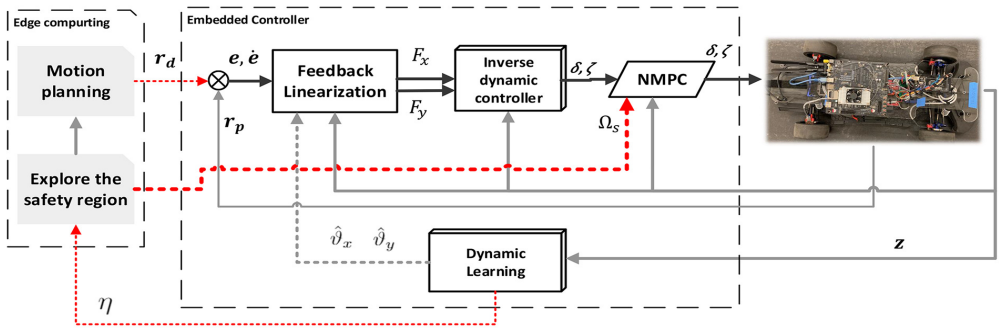


Figure 5.2: Proposed safe predictive control for aggressive autonomous maneuvering.

and then calculate the wheel's speed using the tire's inverse model. The uncertainties from the closed-loop control process can be propagated to the final GP polynomial conjecture. GPs are a nonparametric regression method from machine learning, where the goal is to find an approximation of a nonlinear map from a given state vector $\mathbf{x}_{gp}$ to the approximation function value $f(\mathbf{x}_{gp})$. Then, SOS uses the upper limit of the approximation error to find the safety region Ω_s for the worst-case scenario, as it is demonstrated in Figure 5.2.

5.3.1 Polynomial Gaussian processes

Initially, the longitude and lateral tire-road interaction model in (5.3) estimate the tires models to enhance the vehicle modeling for FLC and NMPC control design. Then, GPPB estimates the steering wheel δ and throttle rate ζ in (5.9) using longitude and lateral forces calculated by FLC in (5.7) by inverse models. However, due to the computational complexity of GP, which scales cubically with the number of training points N (i.e $\left(O\left(N^3\right)\right)$). The learning is performed initially offline based on simulations. Then the trained model is used to initialize the online learning. Online learning is achieved by implementing local learning, using only a fraction of the real time data.

We can obtain the posterior distribution of a polynomial Gaussian function to estimate the control commands in (5.9) at an arbitrary state $\mathbf{x}_{gp} \in \mathrm{R}^{n_x}$, where n_x is the length of state vector $\mathbf{x}$. The steering wheel $\delta = f_\delta(\mathbf{v}, \boldsymbol{F}_y) + \epsilon_\delta$ where $\{f_\delta(\mathbf{x}_{gp}) \in \Omega_\delta \to \mathbb{R} : \mathbf{x}_{gp} = [\mathbf{v}, \boldsymbol{F}_y]\}$ is the real function underneath and $\epsilon_\delta \sim \mathcal{N}\left(0, \sigma^2_{\epsilon_\delta}\right)$. We consider a set of $\mathcal{D}$ input and output data pairs $(\mathbf{x}_i \;\; f_i)$ with length m forming the

dictionary where, $\mathcal{D} = \{\mathbf{Y} = \left[\mathbf{y}_0^T \ \dots \ \mathbf{y}_m^T\right] \in \mathbb{R}^{m\times n_f},\ \mathbf{X} = \left[\mathbf{x}_0^T \ \dots \ \mathbf{x}_m^T\right] \in \mathbb{R}^{m\times n_x}\}$. In the context of GPPB, we aim to seek a representation of the model f_δ and similarly $\{f_\zeta(\mathbf{x}_{gp}) \in \Omega_\zeta \to \mathbb{R} :\ \mathbf{x}_{gp} = [\mathbf{v}\ \boldsymbol{F}_x\ \delta\ \boldsymbol{\omega}]\}$, and $\{f_{\boldsymbol{F}_{ijx}}(\mathbf{x}_{gp}) \in \Omega_{fx} \to \mathbb{R} :\ \mathbf{x}_{gp} = [\mathbf{v}\ \boldsymbol{\omega}]\}$, and $\{f_{\boldsymbol{F}_{ijy}}(\mathbf{x}_{gp}) \in \Omega_{fy} \to \mathbb{R} :\ \mathbf{x}_{gp} = [\mathbf{v}\ \delta]\}$, and $\{\boldsymbol{f}_\eta(\mathbf{x}_{gp}) \in \Omega_\eta \to \mathbb{R} :\ \mathbf{x}_{gp} = [\boldsymbol{e}\ \dot{\boldsymbol{e}}\ \ddot{\boldsymbol{e}}]\}$ as expansions of a series of orthogonal and normalized polynomials of Gaussian process $\boldsymbol{\phi}_\alpha(\mathbf{x}_{gp})$.

$$\boldsymbol{f}_\kappa(\mathbf{x}_{gp}) = \sum_{\alpha\in\mathbb{N}^d} \beta_\alpha \boldsymbol{\phi}_\alpha,\ \alpha = \{\alpha_1, \dots, \alpha_d\} \tag{5.14}$$

where α is the multi-index and $\boldsymbol{\phi}_\alpha(\mathbf{x}_{gp}) = \prod_{i=1}^d \phi_{\alpha_i}^{(i)}(\mathbf{x}_i)$ and $\kappa \in \{\zeta, \delta, \eta, F_{ijx}, F_{ijy}\}$. The computational complexity of GPPB depends on the number of data points, which motivates using a limited number of points. The goal of these approximations is to calculate the control inputs using GPPB and enhance the model of the system for model-based FLC (5.7), safety region estimation in (5.13) and NMPC.

The main idea supporting this method is related to the existence of a suitable polynomial function of the generalized inputs $\mathbf{x}_{gp}$ with a certain accuracy. Inspired by this property, we propose a GPPB approximation method, introduced in [131], which is enabling us to use SOS for algorithmic stability and safety assurance for learning algorithms. The resulting multivariate polynomial GP approximation is given by

$$f(\mathbf{x}) \sim \mathcal{N}\left(\boldsymbol{\mu}^d(\mathbf{x}), \boldsymbol{\Sigma}^d(\mathbf{x})\right) \tag{5.15}$$

where $\mu(\mathbf{x})$ and $\Sigma(\mathbf{x})$ are the posterior mean and covariance functions as

$$\begin{aligned} \mu(\mathbf{x}) &= \mathbf{k}_{\mathbf{xZ}}\left(\mathbf{K}_{\mathbf{XX}} + \mathbf{I}\sigma^2\right)^{-1}[\mathbf{Y}]. \\ \Sigma(\mathbf{x}) &= k_{\mathbf{xx}} - \mathbf{k}_{\mathbf{xX}}\left(\mathbf{K}_{\mathbf{XX}} + \mathbf{I}\sigma^2\right)^{-1}\mathbf{k}_{\mathbf{Xx}} \end{aligned} \tag{5.16}$$

To estimate the unknown part of the systems, GPPB requires collecting a set of data pairs. We have used nonlinear model of the vehicle and the closed-loop form of the system from (5.6). Let's assume that we can create certain maneuvers with no lateral forces $\mathbf{F}_y = \mathbf{0}$, and then we can obtain data pairs for longitude forces as

$$\mathbf{F}_x = \boldsymbol{\Gamma}_x^\dagger\left(\ddot{\mathbf{r}}_P - \boldsymbol{\vartheta}\right) \tag{5.17}$$

Similarly, let's assume that we can perform certain maneuvers that longitude forces are negligible $\mathbf{F}_x = \mathbf{0}$, and then we can obtain data pairs for longitude forces as

$$\mathbf{F}_y = \mathbf{\Gamma}_y^{\dagger} \left(\ddot{\mathbf{r}}_P - \boldsymbol{\vartheta} \right) \tag{5.18}$$

These test scenarios are conducted in simulations for preparing the initial data pairs for the longitude and lateral forces. The collected data in experiments should satisfy Eq. (5.17) and (5.17) to be selected as a training data pair in real time. For example, the straight line trajectories satisfy the conditions for collecting data pairs for longitude forces.

5.3.2 Learning-predictive control

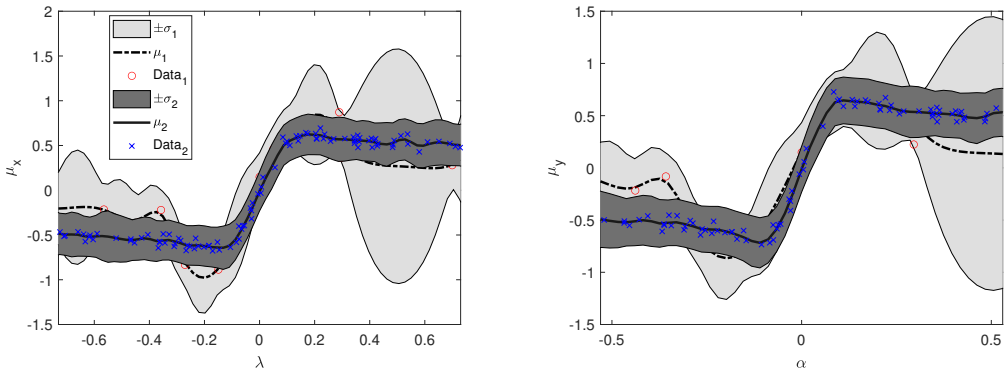
For a given desired trajectory r_p, a model-based nonlinear conjugate gradient optimization is used to enhance the controller commands found by Eqs. (5.7) and (5.9). The implemented NMPC, similar to Chapter 3, guarantees the vehicle will not violate safety region, Ω_s found in (5.13), during the aggressive maneuverings. We formulate the NMPC control problem by using model regression for the tire-road friction learning model estimated by (5.3) as

$$\begin{aligned}
\min_{\Delta u(t)} \quad & \sum_{i=k}^{k+H_p} \left(l_1 \|\mathbf{e}_i\|^2 + l_2 \|\dot{\mathbf{e}}_i\|^2 + l_3 \|\Delta \mathbf{u}_i\|^2 \right) + l_4 J_s \\
\text{subject to} \quad & \mathbf{z}_{k+1,t} = \mathbf{f}(\mathbf{z}_{k,t}, \mathbf{u}_{k,t}) \\
& u_{\min} \leq |\mathbf{u}_{k,t}| \leq u_{\max}, \Delta u_{\min} \leq |\Delta \mathbf{u}_{k,t}| \leq \Delta u_{\max} \\
& \mathbf{u}_{k+1,t} = \Delta \mathbf{u}_{k+1,t} + \mathbf{u}_{k,t},\ k = t, \ldots, t + H_p, \\
& (\beta_{k,t+H_p}, \omega_{\psi_{k,t+H_p}}) \in \Omega_s.
\end{aligned} \tag{5.19}$$

where l_j, $j = 1, \cdots, 4$, are constant weights, $\mathbf{z}(t) = [\mathbf{z}_{t,t}\ \mathbf{z}_{t+1,t}\ \cdots \mathbf{z}_{t+H_p,t}]$ with $\mathbf{z}_{t,t} = \mathbf{z}(t)$ is the sequence of states $\mathbf{z}(t)$ over the prediction horizon H_p at time t. Terms $\mathbf{u}_{k,t}$, $\Delta \mathbf{u}_{k,t}$, and $\mathbf{u}_{t,t} = \mathbf{u}(t)$ are the kth input sequence $\mathbf{u}_t$ and $\Delta \mathbf{u}_t$ respectively. Cost J_s penalizes the possible robot motion outside Ω_s, determines using SOS as represented in Figure 5.1.

5.4 Experimental results

RURacer-2 shown in Figure 2.1 is used for the experiments on the race track shown in Figure 5.4. The controller performance is tested for the aggressive vehicle maneuvers to follow a minimum-time desired trajectory on the track. In Figure 5.4, $SRRT^*$ explained in Chapter 4 is used to find the minimum time trajectory, and learning-predictive controller in Figure 5.2 is used to perform safe-aggressive maneuvers. Thin lines in Figure 5.4 represent the explored trajectories, and the thick multi-color line shows the controlling performance funnel. It is evident during the drifting we have a massive funnel due to uncertainty of modeling for the nonlinear part of the tire that have demonstrated the "*friction-slip ratio*" and "*friction-side slip angle*" relationships shown in Figure 5.3. The controller was tested for different and variable surface conditions to validate the GP learning method and updated safety region. The vehicle's phase portrait as one of the behavior and performance outcomes is compared with other methods without learning and/or safety features. For comparison purposes, we also implemented the Cautious NMPC (CNMPC) in [120] that uses the GP in the design without an approximated safety assurance. The GP can approximate the tire model and enhance the model by collecting more information, $Data_1$ is only 30 data pairs, and $Data_2$ is 110 data pairs and can improve the model by collecting more data in real time as is presented in Figure 5.3.



(a) (b)

Figure 5.3: GPPB based estimation of tire model. (a) Longitude model

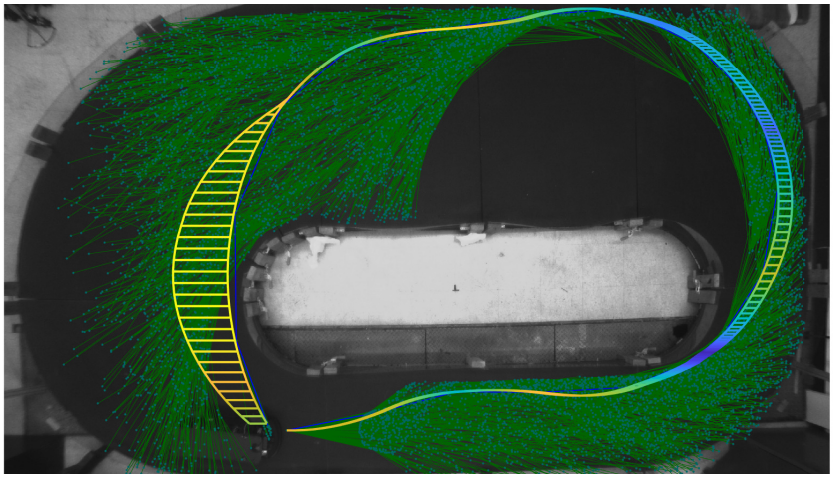


Figure 5.4: Minimum-time feasible trajectory found by *SRRT** and the tracking funnel for the vehicle on the race track.

*SRRT** is used to find the desired trajectory and NMPC is executed to follow this trajectory for the worst-case scenario of the uncertainties at the time to calculate the funnel presented in Figure 5.4. The experimental results shown in Figure 5.5 demonstrates that the vehicle controlled with NMPC stayed inside the funnel discovered in Figure 5.4. Under the control systems design, the vehicle tries to follow the desired trajectory within the shortest time while maintaining the safe motion inside the safety region $\left(\beta(t), \dot{\psi}(t)\right) \in \Omega_s$. Figure 5.6 further shows the vehicle motion comparisons under the two other control designs along with the performance of the proposed control method. Figure 5.6(a) shows the longitudinal and lateral velocities (v_x and v_y) and Figure 5.6(b) demonstrates the yaw-rate $\dot{\psi}$ comparisons. From both figures, we observe that under the AMSF, the magnitudes of the velocity and the yaw-rate are in general larger than those under the CNMPC design, while under the RBNMPC driver, these values are even smaller than those of the CNMPC. To clearly see such a trend, Figure 5.6(c) shows the slip angle comparisons under the three control designs. Under the

Table 5.1: Tracking performance of the best lap for 10 round of the tests.

Controller	Time (s)	$\|\|e\|\|$	Agility metric 1 $\left(\mathrm{m/s^3}\right)$	Agility metric 2
AMSF	4.37	0.11	6.28	0.62
CNMPC [120]	4.61	0.12	5.76	0.57
RNMPC [26]	5.08	0.78	4.65	0.32

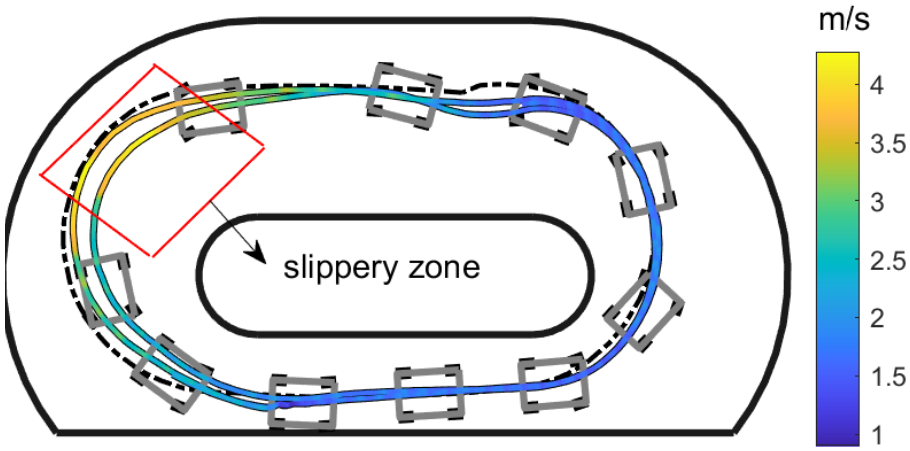


Figure 5.5: Tracking performance for U-turn which the change of the color shows the change in vehicle's speed on the track and the black line is the desired trajectory. (yellow is the fastest and blue is the slowest)

AMSF, the slip angle β reaches and keeps large values (around more than 15 degrees) over a longer period, while the slip angle under the RBNMPC and CNMPC maintains at relatively smaller values.

It is also interesting that from the results shown in Figure 5.6(d), the AMSF generates larger steering inputs than those of RBNMPC (e.g., over 2.5-5s and 10-15 s periods). On the other hand, the RBNMPC did not use any larger steering angles, and the changes are smoother. Moreover, the AMSF uses rapid changes in steering the vehicle to drift while turning in Figure 5.5. The saturated steering angle differences under the two controllers shown in the figure are due to the augmented, larger stability regions of AMSF, $\Omega_o \subset \Omega_s$. Figures 5.6(c) and 5.6(b) also show the vehicle motion inside the safe stability region Ω_s of the vehicle dynamics in Figure 5.1.

5.5 Summary

This chapter presented a safe learning-optimization based motion control method for autonomous aggressive vehicle maneuvers with tire model uncertainties. The AMSF motion controller took advantage of the polynomial GP learning and the SOS machinery to approximate the vehicle's model while the stability of the closed-loop controller is guaranteed. The flexibility of NMPC for approximating the model uncertainties, or

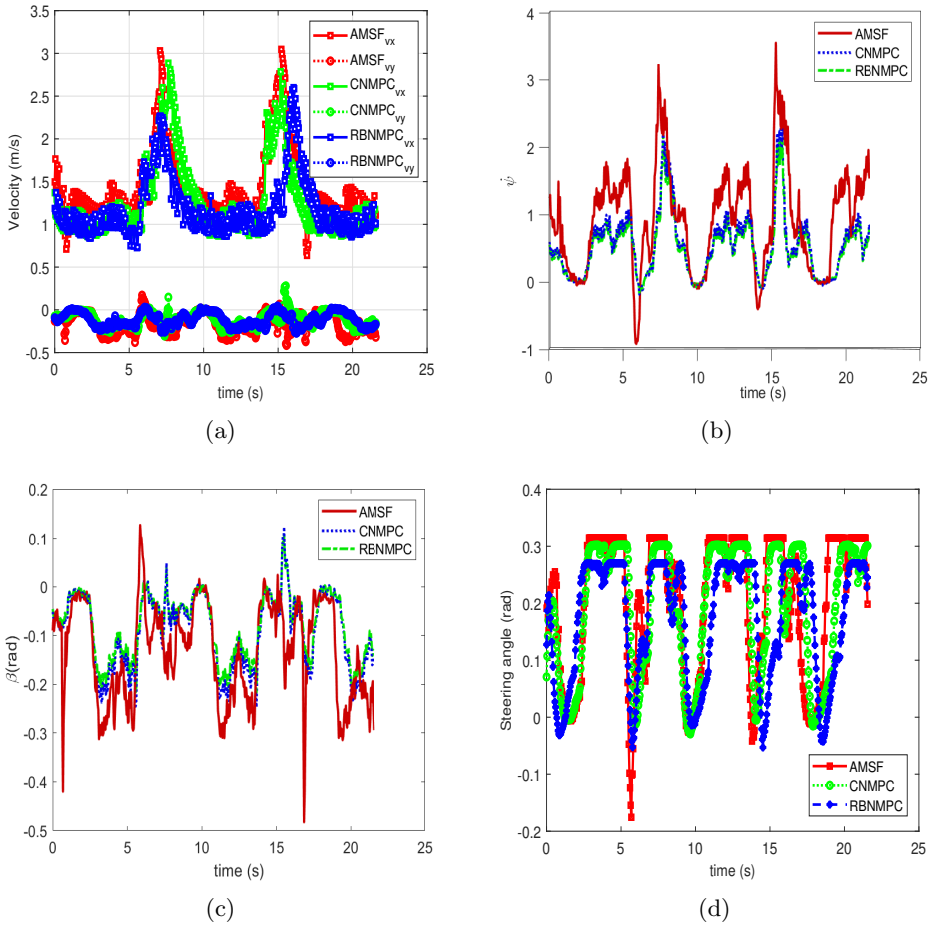


Figure 5.6: Comparison results under three autonomous controllers. (a) Vehicle velocity profiles. (b) Yaw rate $\dot{\psi}$ profiles. The profile indicates the areas outside the safety region Ω_s at these moments. (c) The center slip angle β profiles. (d) Steering angle δ profiles.

part of it, in real time helped to enhance the controller performance. The attractive property of the method in this chapter lies in adaptive guaranteed stability with a proper tracking performance capable of aggressive maneuvers. We implemented and tested the motion controller using a RURacer-2 autonomous vehicle. A comparison with other predictive and learning methods was presented in the experimental results section of this chapter. The comparison results have demonstrated the higher agility and performance under the proposed learning-predictive motion controller.

Chapter 6

Autonomous Stunt Maneuvering: From Simulations to Real-World

Since simulations are inexpensive and more accessible, they have been extensively used for policy learning and controller test purposes. A control policy designed or tested in simulation often does not perform exactly the same when it is deployed in the real world. In this chapter, we focus on using simulations to generate a safe control policy utilizing Constrained Markov Decision Processing (CMDP). The simulation to real world performance and safety are analyzed using the worst-case scenario of uncertainties. The learning method in Chapter 5 is also used to approximate the uncertain part of the dynamic model and disturbances. We demonstrate experimentally that this method permits the development of control policy design in simulations to enable safe autonomous stunt maneuvering for the RURacer2 with disturbances and uncertainties. The proposed method can be used for the safe control policy design for other autonomous stunt maneuvers.

6.1 Introduction

Similar to other aggressive vehicle maneuvers studied in the previous chapter, stunt maneuverings can be used as an active emergency feature to achieve the vehicle's maximum maneuverability. This chapter proposes a novel method for integrating the expert's knowledge with a safe reinforcement learning algorithm. Compared with previous works, the proposed approach formulates a J-turn as a sequence of continuous policies with safety constraints. Constrained Markov Decision Processing (CMDP) solves an optimization problem to find a safe control policy for these maneuvers. Instructions provided by professional stunt drivers construct sequences actions with fuzzy values for

the commands. We studied the problem of taking numerical simulations to real world with guaranteed safety. The Proposed policy search finds an autonomous J-turn maneuver similar to what stunt car drivers would do. Taking the model and enlivenment uncertainties into consideration during the policy search enables us to find a safety-guaranteed control design for the real world test. We demonstrate experimentally that this method permits the development of policy design for the safe stunt J-turn using simulations to test in real world.

The instructions provided by a professional stunt driver are itemized steps without explicit details. In this chapter, we want to achieve an Autonomous Stunt J-turn Vehicle Maneuver (ASVM) using a Safe Reinforcement Learning (SRL) approach. Those astonishing maneuvers by professional drivers motivate the realm of autonomous vehicles to mimic these highly agile motions in emergency situations. To address this, we develop a method that interprets natural language instructions into a sequence of actions with fuzzy representations of the vehicle actions and states. For example, one can tell his or her future autonomous vehicle to stay on the right line and do not go beyond the speed limit; however, it will be more complicated when an emergency car driver wants to perform a quick turn and possibly uses ASVM. Constrained nonlinear optimization uses the sequence of actions to find a safe control policy. The constraints represent physical limitations and safety of humans and the vehicle where the CMDP framework is used to model such circumstances [132]. Worst-case criterion under bounded uncertainty is integrated into our policy search to take the simulation results into the real world with guaranteed safety.

Professional stunt vehicle operations usually occur at the maximum limit of the vehicle's maneuverability on the stability limits' edge. In recent years, RL is used for aggressive maneuvering of aerial [133] and ground vehicles [75], but the safety of the maneuvers are compromised. Safe control policy search attracts more attention in recent years [134, 135], where natural language can be used to add safety conditions to the policy search algorithms [136]. SRL enables us to find feasible and safe stunt maneuvers viable for future emergency autonomous vehicles.

Random exploration for a specific control policy is often highly inefficient and may

(a) (b)

Figure 6.1: Stunt vehicle maneuvers performed by professional drivers at stunt driving events. (a) Driving on two wheels (also known as ski-stunt driving) and drifting of two cars very closely. (Ken Block Does the Ultimate Playground; https://oldtripod.com/ken-block-does-the-ultimate-playground) (b) Two cars are jumping while a third car is performing ski-stunt maneuvers.

completely fail. Expert suggestions are incorporated with policy search algorithms to find a rewarding policy with less number of learning iterations [137]. It is shown that good advice can reduce the amount of exploration required to learn a control policy [138]. Policy search approaches can convert human instructions into synthetic training experiences to scaffold the basic representations of RL [139]. Also, natural language can be directly translated into safety conditions for the policy search with time and sequential ordering [136]. The external knowledge can initialize a policy search to avoid unnecessary random exploration. Still, the safety and effect of uncertainties are not part of the instructions since these methods are tested only in simulations [137,138]. However, the instructed RL control policy search might perform well in simulations but poorly in the real world, and the safety of the policy in real world is not guaranteed [139]. Control policies trained in simulation should satisfy the safety criteria for successful deployment in the real world to avoid any possible damages to the vehicle and the environment.

To address the problem of the safe control policy search, various methods are incorporated safety features into policy learning algorithms. A safe learning algorithm is subjected to inequality and equality constraints to find a competent control policy with model uncertainty [31, 134, 140]. The global safety of an RL during training via a set of local linear constraints is guaranteed by constructing Lyapunov functions to systematically transform optimizations and RL algorithms into their safe counterparts [135].

Similarly, Control Barrier Functions (CBF) are used to guarantee safety with high probability during the learning process [141]. A reinforcement learning framework combines the different simulation levels for steady-state drifting of a car without considering the safety criteria [75]. Also, a similar framework for RL with multiple simulators of a target task with varying levels of fidelity is tested on a remote-controlled car [142]. Vehicle's behavior during extreme maneuvers is less predictive, and finding a safe control policy for a stunt J-turn is more challenging without understanding the vehicle's dynamics during agile maneuvering.

Autonomous vehicles attempting to perform these types of maneuvers are confronted with many new challenges, most notably the unstable nature of stunt maneuvering, the difficulty of accurate numerical and analytical modeling, and unpredictability of the performance in a new environment [10, 74, 143]. A small change such as the road-tire property can completely change the behavior of the vehicle specifically for agile maneuvers [76]. Analytical methods successfully have controlled the vehicle for agile maneuvering or even drifting, but they cannot be generalized for stunt maneuvers [24,144,145]. The majority of state-of-the-art works in vehicle control focuses on normal maneuvers that either does not need a precise model for a control design where the difference between the simulation and real world is negligible [146]. Model-based reinforcement learning requires a precise analytical model to find a control policy for stunt maneuvering using simulations. Then, the safety of the control policy in real world can be guaranteed for the worst-case situations by knowing the range of the uncertainties [75].

Motivated by these observations, an example is shown in Figure 6.1. This chapter proposes an instructed learning method to find safe stunt J-turn vehicle maneuvers with bounded model uncertainties. A safety factor based on the probability of violating the worst-case scenario's constraints is embedded into the admissible action sets. This instructed learning formulation initializes the policy search that is restricted to the admissible set of actions for any control sequence. Also, the performance and safety of the policy will be improved by implementing a learning-based NMPC in Chapter 5.

We propose approaching the problem of learning stunt maneuverings in their native

form, i.e., as sequential instructed learning, while evaluating the safety for the worst-case scenarios during autonomous stunt J-turn maneuvers. The proposed model-based RL algorithm finds a safe control policy with both continuous and discrete action spaces while it incorporates the uncertainties of the tire-road friction model. Being able to initiate the control policy from an expert's instruction and handling both discrete and continuous actions allows us to learn a natural policy similar to the instructions given by a professional stunt driver. We demonstrate our approach's guaranteed-safety and effectiveness on a J-turn stunt maneuvering, focusing on using a policy trained from simulation for real world experiments. The native hybrid policy search initiated by a professional stunt's instruction ensures the safety and performance for the worst-case scenario of the uncertainties in simulations. The selected control policy should achieve the maximum rewards for the nominal model of the system. Simultaneously, it satisfies the safety and minimum requirements of the performance for the different simulated scenarios of uncertainty and disturbances.

During the learning process, the vehicle must try different actions for each sequence in order to satisfy the requirements of the provided instructions. The control policy should address the effect of the uncertainties for real world implementation to guarantee the safety of the maneuver for a collision-free stunt J-turn. The learning algorithm searches within the instructed parts of the state-action space to converge with fewer samples. A complete and rewarded J-turn stunt maneuver is the one with the maximum energy efficiency and no risk of a collision to maintaining safety. More specifically, the purpose of this algorithm is to find a control policy that can perform a complete J-turn without losing much speed and violating the safety lines. In this method, we might scarify the performance to the safety, which it means we might lose speed during the J-turn maneuver while minimizing the risk to collide.

6.2 Safe Autonomous Stunt Vehicle Maneuvers

A professional stunt driver incorporates a precise set of actions to maximize the vehicle's maneuverability, shown in Figure 6.1(a). A complete and rewarded autonomous maneuver has the maximum energy efficiency in the shortest amount of time and distance

(a)

(b)

Figure 6.2: (a) Stunt J-turn performed by professional drivers at stunt driving events. (Cropped from youtube.com/watch?v=IlaDkxFKD9U) (b) Ski-stunt maneuver tested by RC truck.

without risk of collision or failure. This chapter demonstrates the feasibility of using simulation-based learning for performing autonomous stunt J-turn vehicle maneuvers in real world using the proposed safe policy search method. A professional stunt driver should knows how to perform a 180° (J-turn), 180° handbrake turn (bootleg-turn), and driving on two wheels (Ski-stunt) and over or under-steer. The same strategy can be developed for other stunt maneuvers in future works.

6.2.1 Stunt J-turn maneuvering

A J-turn also called a "reverse 180°" is a stunt maneuver in which a reversing vehicle is spun 180 degrees until facing forward, without changing the direction of travel. A perfect J-turn reaches a large yaw-rate for a short amount of time without losing much speed. Also, the maneuver's performance might change dramatically with a small change in speed, tire, or road's property. We have received these instructions from a professional stunt driver.

1. Check your surroundings and make sure there is enough space and nothing the car might hit.

2. Move reverse until you get enough speed.

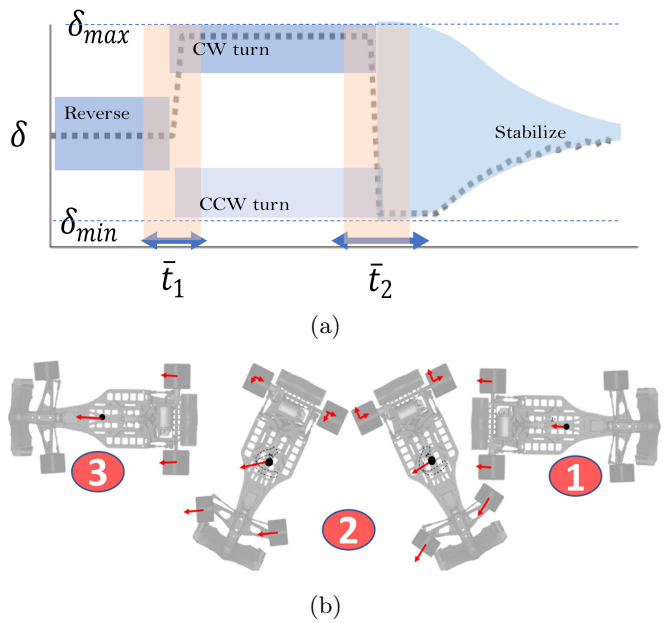


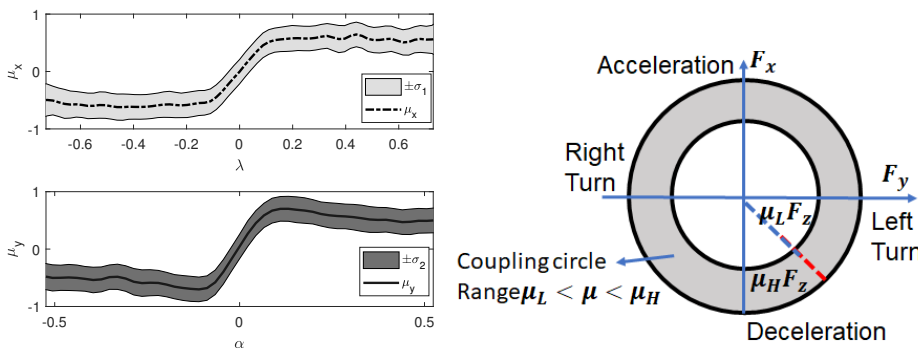
Figure 6.3: Initiate the learning to J-turn based on professional paradigms. The figure shows (1) three sequence of motions (2) two switching windows between different motion sequences, and (3) a bounding box of the safe subset of the inputs. The objective is to minimize the time and distance for J-turn as total reward without violating safety constraints for the WC friction uncertainty.

3. Spin the steering wheel all the way, be careful of not to roll the car.

4. Once your turn is around 90°, start to straighten up the wheel.

5. Control the steering wheel until you're facing the right direction.

Taking advantage of the instructions, our proposed method finds the policy control for an autonomous J-turn maneuver in three steps; 1) The instruction is interpenetrated to a set of maneuvers as it is shown in Figure 6.3, 2) RL finds the controller policy that works in simulation with verified safety for the Worst-Case (WC) scenario using upper bound of the uncertainty, 3) test the controller on the vehicle and improve the performance by increasing the model's precision. A safe control policy will not violate any safety criteria for any of the situations inside the uncertainty range during the J-turn maneuver $\{\boldsymbol{z}_{t+1}(\boldsymbol{z}_t, \boldsymbol{u}_t) \notin \mathcal{U} \ \forall \ \mu_t \in [\mu_L, \mu_H], \ t > 0\}$, where $\boldsymbol{z}$ and $\boldsymbol{u}$ are the states and inputs vector of the vehicle and $\mathcal{U}$ the unsafe subset of states.



(a) (b)

Figure 6.4: (a) The schematic view of a tire. (b) Coupling circle of the friction with the friction range.

6.3 Model uncertainties for safe policy design

The tire's force model caused by the contact patch between the tire and ground is commonly uncertain and difficult to measure in real time [83]. However, the boundedness nature of the tire friction model can help to find the upper limit of the model uncertainties, shown as a frictional circle in Figure 6.4(b). The friction circle is captured as $\sqrt{F_x^2 + F_y^2} \leq F_z \mu_t$, where F_x, F_y, and F_z are the longitude, lateral and normal forces, and μ is the total friction coefficient. However, knowing the upper limit μ_H and the lower limit μ_L of the friction models shown in Figure 6.4(a), can be utilized to find a control policy for a stunt J-turn by assuring the safety with an unknown friction model.

In this chapter we are using the same scaled car platform shown in Figure 2.1(a) for experimental tests. The vehicle's model is considered as a mechanical system with non-ideal nonholonomic constraints, due to existence of slipping and skidding of the tires during a stunt maneuver. However, knowing the range of the system actuation, a funnel of the states of the system is predictable for safety assurance of the control policy.

Remark 6.1 *An uncertain constrained mechanical system with limited inputs subjected to non-ideal nonholonomic constraints evolves in time in such a way that its acceleration at each instant of time has overall possible accelerations and forces at that instant of time [147]. The overall limits on accelerators and forces can be used to calculate the*

WC scenario within these limits for safety checks.

In a closed-loop control system of an autonomous vehicle, we should consider that the magnitude of the tire/road forces acting to accelerate, decelerate and rotate the vehicle are bounded $\{||F_{x,y,z}(t)|| < F_{x,y,z_{\max}}, \forall\, t > 0\}$, the generated torques and speed by the electric motor are bounded $\{||\tau_m(t)|| < \tau_{m_{\max}}, ||\omega_m(t)|| < \omega_{m_{max}}, \forall\, t > 0\}$ and steering wheels rotations in limited as $\delta(t) \in [\delta_{\min}, \delta_{\max}]$. The tire-road friction model is highly nonlinear and uncertain but limited where $\mu \in [\mu_L, \mu_H]$. Knowing the maximum and minimum capability of the vehicle's actuators and range of model uncertainties, we can predict states the vehicle will possibly reach.

6.3.1 Vehicle model with model uncertainties

To formulate our safe policy search method, we use a representation of the nominal and uncertain part of the vehicle dynamics is given as

$$\boldsymbol{z}_{t+1} = \boldsymbol{f}_n(\boldsymbol{z}_t, \boldsymbol{u}_t) + \boldsymbol{f}_u(\boldsymbol{z}_t, \boldsymbol{u}_t), \tag{6.1}$$

where the nominal model $\boldsymbol{f}_u$ is the vehicle's presented in Eq. (5.1) and $\boldsymbol{f}_u$ is a bounded function for unknown part of the vehicle model. The detail of the model matrices in (6.1) can be found in (5.2). Similarly, tire forces $\boldsymbol{F}_x$ and $\boldsymbol{F}_y$ in (5.2) are controlled implicitly using the steering wheel δ and throttle ζ. Using the GP for tire model estimations in (5.3), the upper and lower bound of the tire model uncertainty, μ_L and μ_H can be approximated, as it is demonstrated in Figure 6.4. The range of the uncertainties in tire modeling will be used for safety assurance during the control policy search.

The design consists of four parts: interpret an expert's instruction to compute a fuzzy range of steering angle and the throttle rate, estimating the upper bound of uncertainties for the worse-case scenario, a model-based RL to search for the safe policy, and an NMPC controller tracks the desired trajectory in real time outlined in Chapter 3.

6.4 Safe Reinforcement Learning

The proposed SRL procedure involves two decision-making scenarios, choosing the control sequence and action signals which are used by the controllers. We propose to treat

the stunt maneuvering problems as instructed by an expert and solving them as hybrid constrained optimization problems. This section briefly explains how to use a modified CMDP to find the sequence of safe maneuvers and actions simultaneously for an autonomous stunt J-turn maneuver.

6.4.1 Preliminaries

Let's formulate our CMDP as a tuple of $\mathcal{T} = \{\mathcal{Z}, \mathcal{A}, , f_n, f_u, P, r, \gamma, \mathcal{U}, \mathcal{C}\}$, where $\mathcal{Z}$ is a finite set of states, $\mathcal{A}$ is a hybrid set of continuous and discrete action which contains the J-turn stage shown in Figure 6.3(a) and the inputs vector $\boldsymbol{u}$. $f_n : \mathcal{Z} \times \mathcal{A} \to \mathcal{Z}$ is the nominal dynamics model of the system, $f_u : \mathcal{Z} \to \mathcal{Z}$ is a bounded function for approximations of the unknown system dynamics, $\mathcal{U}$ is a set of unsafe states. Also, $r(\boldsymbol{z}, \boldsymbol{a}) : \mathcal{Z} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $\gamma \in (0, 1)$ is a discount factor, and $P(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t, \boldsymbol{a}_t) : \mathcal{Z} \times \mathcal{A} \times \mathcal{Z} \to [0, 1]$ denotes the probability distribution for the transition kernel. The set of safety penalty cost functions $\mathcal{C} : \mathcal{Z} \times \mathcal{A} \to \mathbb{R}$, where $c(\boldsymbol{z}, \boldsymbol{a})$ is the immediate risk for state $\boldsymbol{z}$ with action $\boldsymbol{a}$. We assume the set of unsafe states is initially empty $\mathcal{U}_{t=0} = \varnothing$, where $\mathcal{Z} = \{\mathcal{Z}' \cup \mathcal{U}\}$.

A policy $\pi : \mathcal{Z} \to \mathcal{M}(\mathcal{A})$ maps each state to a probability distribution over the actions. The value of a state $\boldsymbol{z}$ under policy π is denoted $V^{\pi,P}(\boldsymbol{z})$ and represents the expected sum of discounted returns when starting from an initial state and executing policy π,

$$V^{\pi,P}(\boldsymbol{z}) = \mathbb{E}^{\pi,P}\left[\sum_{t=0}^{\infty} \gamma^t r\left(\boldsymbol{z}_t, \boldsymbol{a}_t\right) \mid \boldsymbol{z}_0 = \boldsymbol{z}\right] \tag{6.2}$$

where the value function $V^{\pi,P}(\boldsymbol{z})$ for policy π relates to action-value function for the expected discounted return according to policy π when choosing action $\boldsymbol{a}$ in state $\boldsymbol{z}$ as

$$V^{\pi,P}(\boldsymbol{z}) = \mathbb{E}^{\pi,P}\left[Q^{\pi}(\boldsymbol{z}, \boldsymbol{a})\right] \tag{6.3}$$

which both value function and action-value function both satisfy the recursive expression as

$$Q(\boldsymbol{z}_t, \boldsymbol{a}_t) = \mathbb{E}^{P}_{(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t, \boldsymbol{a}_t)}\left[r(\boldsymbol{z}_t, \boldsymbol{a}_t) + \gamma V(\boldsymbol{z}_{t+1})\right] \tag{6.4}$$

At each state in any episode, the state will move to the next state if the next simulated state is in the safe region. A state will move to the set of unsafe states if the

Algorithm 6.1: Safe IRL-ASVM initialized by a sequence of fuzzy instructions (6.6)

```
Initialize RL Policy using the instructions provided by a professional stunt driver π^0_sim;
Initialize the analytical model knowing uncertainty and safety criteria in simulation;
Approximate the initial policy for simulations with no uncertainty as π^f_sim using π^0_sim;
Collect Episode Reward, (the fastest full 180 degree turn is the most rewarded one);
Start: Safe RL search with π^f_sim while safe policy for real world is not found do
    Evolve the policy search through constrained MDP for simulations;
    Run the simulations and evaluate safey criteria for the worst case scenario;
    if Policy is safe then
        Update the policy π_rw for real world experiments
    end
    i ← i + 1;
end
Initialize the controller for the real world;
Apply learning rules for safety and performance improvements;
```

safety penalty cost violates the upper bound of the uncertainty criteria $||c(\boldsymbol{z}, \boldsymbol{a})||_{\inf} > \sigma$, where the upper bound of safety criteria is a limited positive value $\sigma(\boldsymbol{z}, t) \in \mathbb{R}_{\geq 0}$. We study safe exploration as a constrained policy optimization to find the optimal policy in the subset of safe states.

6.4.2 Hybrid optimization Procedure

We consider a stunt J-turn maneuvering as a hybrid policy of a simple class in which the action spaces are represented with both continuous and discrete dimensions. A generalized state-dependent distribution of action policy $\pi(\boldsymbol{a}|\boldsymbol{z})$ models both discrete and continuous random variables as

$$\begin{aligned} \pi(\boldsymbol{a} \mid \boldsymbol{z}) = \pi^{\varsigma}\left(\boldsymbol{a}^{\varsigma} \mid \boldsymbol{z}\right) \pi^{d}\left(\boldsymbol{a}^{\mathcal{D}} \mid \boldsymbol{z}\right) = \\ \prod_{\boldsymbol{a}^{i} \in \boldsymbol{a}^{\varsigma}} \pi^{\varsigma}\left(\boldsymbol{a}^{i} \mid \boldsymbol{z}\right) \prod_{\boldsymbol{a}^{i} \in \boldsymbol{a}^{\mathcal{D}}} \pi^{d}\left(\boldsymbol{a}^{i} \mid \boldsymbol{z}\right) \end{aligned} \tag{6.5}$$

where ς and $\mathcal{D}$ represent the continuous and sequential discrete action spaces. The subset of discrete actions for a J-turn, $i = 1, 2, 3$ discrete choices of action sequences.

The subset of continuous actions is a normal distribution continuous range of values based on the expert's instruction $\boldsymbol{u}_k \in \boldsymbol{U}_k$, shown in Figure 6.3(a). Then the optimal policy will be discovered inside this region instructed by an expert.

6.4.3 Worst-case scenario

Instead of maximizing the expectation of the return for the WC policy over all possible models, the WC scenario of all safety functions should be limited to the upper bound of uncertainty criteria [148]. We formulate the WC scenario for any given set of state and action as an optimization problem subjected to the system's dynamic to maximize the cost for spatial and temporal constraints within the range of uncertainties. By utilizing the proposed CMDP, the safe control policy problem is solved without violating the robot motor capability. In general, the CMDP problem we wish to solve is given as follows.

$$\begin{aligned} \min_{\pi} \quad & \mathbb{E}^{\pi,P}\{\textstyle\sum_{t=0}^{\infty} \gamma^t r\left(\boldsymbol{z}_t, \mu^{\pi}(\boldsymbol{z}_t)\right) \mid \boldsymbol{z}_0 \in \mathcal{Z}'\} \\ \text{s.t.} \quad & [max_{\sigma} \quad ||c(\boldsymbol{z}_t, \boldsymbol{a}_t)||_{\text{inf}} \quad] < \epsilon_t^{safe}, \quad \forall t \in [0, T], \end{aligned} \tag{6.6}$$

where $\mathbb{E}^{\pi,p}(\cdot)$ stands for the expectation with respect to the policy π and the transition model P. The proposed policy search for a safe stunt J-turn as a constrained optimization can lead to a risk-free desirable maneuver with imprecise modeling. This imprecise modeling is a result of two types of uncertainties: a) the inherent uncertainty related to the tire-road interactions, and b) the parameter uncertainty and sensing noise related to the vehicle. The maximum safety penalty cost associated with the WC scenario is limited to the upper bound of safety criteria ϵ_t^{safe}. Knowing the limits of the unknown part of the physical model of the vehicle, we can assume that all reachable states for the vehicle with a certain set of actions from a given state are predictable. The policy's safety is ultimately guaranteed by the proposed pseudo-code for a modified CMDP in Algorithm 6.1.

However, using Eq. (6.1) for numerical simulations of the vehicle's motion, and the model parameters such as the friction coefficients are still not measured precisely or we only have a bounded estimations using GPPB in Chapter 5. Hence, the range of

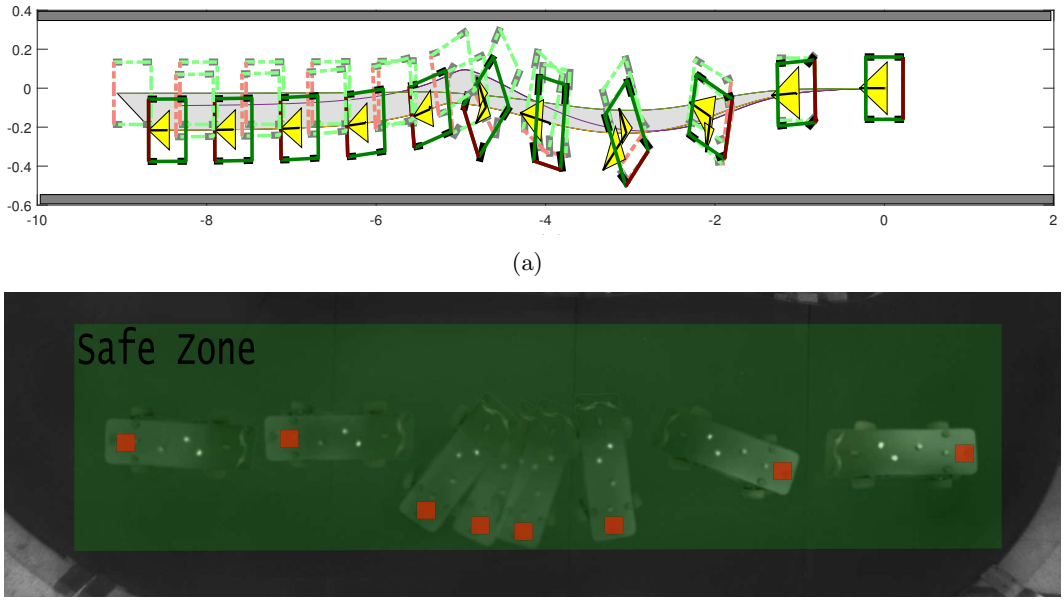
uncertainties for the system in real world are mapped into the simulations as $f_u(\cdot)$ to find a risk-free control policy for autonomous stunt maneuvers. The initial control policy π_s^{init} is based on the nominal model $f_n(\cdot)$ when the optimal controller is designed for a nominal system, and then the safe control policy is explored whiteout violating the safety criteria for the worst-case scenario of uncertainties.

6.5 Simulations and real world tests

The main difference between our method and RL-CBF is that we are using a hybrid policy search. Using a hybrid policy search enables us to incorporate a professional stunt driver's instruction as a discrete seance of continuous actions. The vehicle will not enter the unsafe sub-set of state space for the future prediction horizon H_p, $\{s_t^k \notin \mathcal{D} \forall t > 0, k = 1, ..., H_p\}$ using the NMPC. Unlike RL, NMPC is able to solve convex constrained optimizations in real time to provide the minimal control deviation from the safe control policy. The proposed controller is compared with the safe control policy without NMPC, and with RL-CBF method [141]. Safe RL policy with NMPC provides higher probability of safety guaranteed J-turn maneuvers during the learning process and can maintain the performance-guaranteed policy for a wider range of model uncertainties. If we have no dynamics uncertainties, then safety is always guaranteed for the learned policy using RL. The deployed NMPC is able to drive the vehicle to the safe region in real time.

6.5.1 Experimental Results

To find the desired control policy, the CMDP computes the sequence of input commands and predicts trajectory of states using numerical simulations. The learned safe control policy and the NMPC are embedded to the robot and it communicates to the edge computer for the heavy computation such as the learning process and image processing for robotic vehicle's localization. In the experiment, safe RL finds the optimal solution within the safe sub-set of the feasible controllers for the desired stunt maneuvering with the guaranteed stability for the worst-case scenario. The proposed methods has



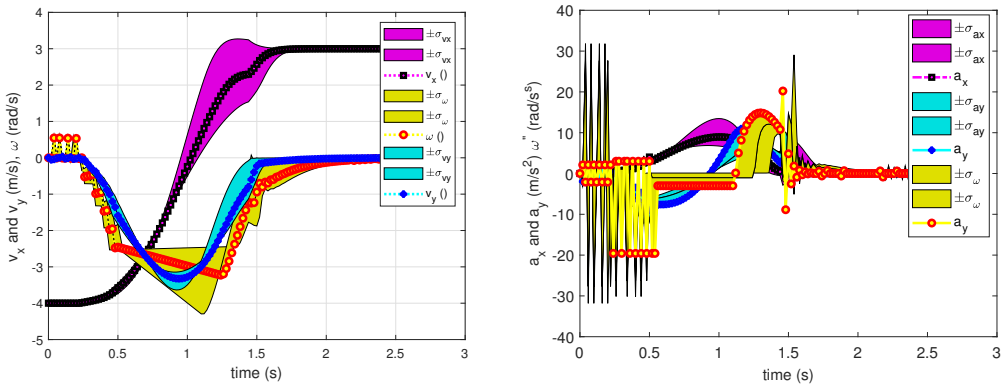
(a)

(b)

Figure 6.5: Simulations for a J-turn in a narrow road. (a) The control policy in simulations is safe for the WC scenario of the uncertainties during a J-turn Maneuver. (b) Experimental results for the J-turn using the same control policy found in simulations (red squires show the front of the vehicle).

the highest probability of successful and safe maneuvers shown in Table 6.1.

Feasibility funnel approximation is a popular tool to approximate all the future states that the robot might go in the future prediction horizon with existing uncertainties. We have implemented the simulations and calculated the feasibility range for the vehicle's state such as location, speed and acceleration for the worst-case scenarios of the uncertainties. The feasibility funnel of the vehicle's speed and accelerations are wider during the spin and thinner during the straight motions, shown in Figure 6.6(a) and Fig 6.6(b). The large chattering seen during $0 - 0.5s$ in Figure 6.6(b) are the measurement noises. Knowing the range of the uncertainties for the tire model and assuring the safety for the worst-case scenario during RL policy search, we are able to guarantee the vehicle will perform the J-turn maneuver in real world inside the safe zone as it is shown in Figure 6.5(b). Figure 6.5(a) demonstrates the policy never passes the safety lines even for the worst-case scenarios during the simulations.



(a) (b)

Figure 6.6: Showing the range of feasible states for the worst-case scenarios of a simulated episode with policy π_{rw}. (a) Feasible range of longitude, lateral and the rotational velocities of the vehicle. (b) Feasible range of longitude, lateral and the rotational accelerations of the vehicle.

6.6 Summary

Adding the information about the model uncertainties and safety criteria into the model-based RL framework allows us to explore a safety-guaranteed control policy for stunt vehicle maneuvering. The proposed safe RL algorithm was initiated with an expert's instruction to improve exploration efficiency. The control policy was explored based on the vehicle model in the simulation where the safety-guaranteed was implemented in real world. This framework, which combines the analytical model of the vehicle and range of uncertainty with the expert's instruction, has the advantages of starting the safe real world maneuvering and incorporating a better uncertainty estimation from measurements for real time improvement. A significant assumption in this

Table 6.1: Comparisons results for autonomous stunt J-turn maneuvers under different control policies (for 50 simulations with various random uncertainty situations and 8 real world tests).

Method	**WC scenario**	**real world**	**J-turn time**
Safe IRL	96.00%	87.50%	1.72ms
RL-CBF [141]	96.00%	75.00%	1.82ms
Safe IRL-NMPC	100.00%	100.00%	1.51ms

work was that the physical model of the vehicle can be represented by the nominal and unknown functions. However, computing the valid safe set of control policies was based on the boundedness of the uncertainties.

Chapter 7

Conclusions and Future Work

This dissertation studied constrained motion planning and control approaches for safe autonomous aggressive vehicle maneuverings. In the future of the autonomous car industry, saving lives might depend on more demanding maneuvers than what the average drivers know how to do. Studying the professional driver performance reveals that these drivers are safely operating the vehicle outside the commonly used stable regions to achieve high motion agility [10]. To benefit aggressive maneuverings as a safety feature, an autonomous car should be able to disable ESC and ABS and step beyond what is defined as a safety boundary [4]. This dissertation takes a safety viewpoint and approach for aggressive maneuverings and designs the constrained planner and controllers for autonomous vehicles.

7.1 Conclusions

In Chapter 2, a closed-loop safety region was introduced for aggressive vehicle maneuverings. Using a closed-loop safety region instead of a previously used open-loop one, we designed a safety-guaranteed motion controller when without scarifying the maneuverability. Hence, this safety region can be used during real time optimizations of an iterative constrained controller such as NMPC. The car-like robots, RURacer-1 and RURacer-2, were used for the experiments. The vehicle was localized by a motion capture system and communicated with a local computer through WiFi. There were multiple sensors, motor controllers, and independent beak systems attached to the robots. A tire testing setup was utilized to measure the tire-road friction model characteristics. An results, state-space models of the vehicle's dynamic based on its physical properties were presented for the model based motion planning and control

purposes.

In Chapter 3, we presented a control design by formulating the NMPC on top of the FLC for aggressive vehicle maneuvers. The stability of the closed-loop controller was analyzed using Lyapunov stability theories. The convergence of NMPC was improved using nonlinear Conjugate gradient as our constrained nonlinear programming algorithm. The safety of the maneuvers were guaranteed with NMPC's capabilities of handling different type of constraints. The control system design was built on the extended safety region concept in the phase plane of the robotic vehicle dynamics. We used an LQR-based feedback design to compute the safety boundary that was less conservative than the stability region used in the existing controller design. The safety region was then used in a modified nonlinear model predictive control design to achieve high agility maneuvers. The control design was tested and validated on a scaled vehicle. Finally, a comparison with human expert driving was also presented to demonstrate the controller performance.

Chapter 4 presented a sparse and optimal motion planning algorithm *SRRT**. This planner was able to find trajectories for autonomous aggressive vehicle maneuvers such as the race-line for a minimum time lap or agile turns. The *SRRT** motion planner took advantage of the *Sparse-RRT* and the *RRT** algorithms. This advantage of the sparse property for motion planning helped to reduce the computational burden by removing useless nodes in the searching process. The most attractive property of the *RRT** lies in fast convergence to the optimal solution. We tested the motion planner using RURacer-1 for minimum-time scenarios. The experimental results have demonstrated a high agility maneuvering performance under the autonomous driving control with the motion planner.

Chapter 5 presented a learning-optimization-based motion control method for autonomous aggressive vehicle maneuvers. The AMSF motion controller took advantage of the polynomial GP learning and the SOS machinery to approximate the vehicle's model while the closed-loop controller's stability was guaranteed. The flexibility of NMPC for using model approximation functions can improve the performance in real time. The attractive property of the method in this chapter lies in adaptive guaranteed

stability with optimal performance. We implemented and tested the motion controller using a RURacer-2 autonomous vehicle. The experimental results have demonstrated the high agility maneuvering performance under the autonomous driving control with the motion controller. However it is demonstrated that SOS and PBGP can be used to address a safety-guaranteed control design, this method was limited due since SOS algorithm works only for polynomial based systems. If there was no accurate polynomial based estimation for the system, this method will not be applicable.

In Chapter 6, a model-based RL framework was introduced to explore a safety-guaranteed control policy search for stunt vehicle turns. Tire model uncertainties were addressed during the control policy search using the worst-case scenario approach. A safe policy search approach initialized with an expert's instruction to improve exploration efficiency. A model-based control policy search used simulations to find a safe policy for stunt J-turn maneuvers in real world implementations.

7.2 Future Work

There are two main directions to extend this dissertation's work, practically and theoretically. The dissertation's work can be extended practically by optimizing the motion planner and motion controller's architecture for real ime performance. On the other hand, a real time integrated motion planning and control framework for safe, agile vehicle maneuvers can be used to minimize the risk of accidents in future autonomous vehicles. The next step can be addressing more demanding maneuvers such as those performed by professional stunt drivers in Figure 6.1(a).

Theoretically, there are a few research directions that can follow this dissertation's work. Chapter 2 has introduced an extended safety region for aggressive maneuvering, which was improved using SOS-based approaches to estimate this region dynamically for a class of learning-based controller proposed in Chapter 5. It was expected that determining maneuver's safety region could be extended to learning-based controllers and vehicle's model with higher degrees of freedom to prevent roll-over and other preventable dangers during aggressive maneuverings. Even though the vehicle's roll-over

problems have been studied in [149], preventing roll-over for autonomous aggressive maneuvering problems still needs to be solved to achieve a fully safe autonomous aggressive maneuvering goal.

In Chapter 3, a model-based was NMPC subjected to a safety region as a constraint was proposed. The model-based NMPC was also used in Chapter 4 for in the *steer* function of RRT^* during the *rewiring* process. Using the NCG dynamic programming and applying the safety region constraints, we can determine a configuration for the vehicle's state-space with guaranteed safety. In Chapter 4, the sparse motion planning algorithm was able to find a minimum time trajectory on the race-track off-line and used NMPC to follow that trajectory in real time to compete with a professional driver. A real time motion planning for agile maneuvers with guaranteed safety for systems with model uncertainty needs to be implemented to achieve a fully autonomous algorithm in a dynamic environment with unknown parameters.

In Chapter 5 a GPPB-based learning-predictive controller was proposed to improve the model-based NMPC in Chapter 3. GPPB was used to estimate the tire-road model and SOS was used to estimate the safety region. The proposed method was a partially learning-predictive controller since we assume we had known part of the vehicle model. In Cha pter 6, the safety guaranteed control policy was studied using a safe IRL approach for stunt J-turn maneuvers. However the learning-based controllers have been studied, a predictive controller with a provable guaranteed safety with model uncertainties is still a challenging problem.

Professional stunt drivers have been trained and practiced for thousands of hours to achieve the maximum maneuverability of the vehicle safely. Understanding these maneuvers better can help us to design a safe control policy which can be used as an automated safety feature in emergency situations for the future driver-less vehicles. Due to the differences between different stunt maneuvers and the uncertain nature of tire-road model, the autonomous aggressive maneuvering cannot be directly integrated as a safety feature and must be customized for any maneuver and condition. Generalizing aggressive maneuvering as a safety feature and all of these research topics can be the future works of this dissertation.

References

[1] R. Y. Hindiyeh, "Dynamics and control of drifting in automobiles," Ph.D. dissertation, Dept. Mech. Eng., Stanford Univ., Stanford, CA, 2013.

[2] T. Litman, "Autonomous vehicle implementation predictions." Victoria Transport Policy Institute Victoria, Canada, 2020.

[3] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations," *Transportation Research Part A: Policy and Practice*, vol. 77, pp. 167–181, 2015.

[4] S. C. Peters, "Optimal planning and control for hazard avoidance of front-wheel steered ground vehicles," Ph.D. dissertation, Dept. Mech. Eng., Mass. Inst. Tech., Cambridge, MA, 2012.

[5] E. Constantinou, G. Panayiotou, N. Konstantinou, A. Loutsiou-Ladd, and A. Kapardis, "Risky and aggressive driving in young adults: Personality matters," *Accid. Anal. Prev.*, vol. 43, no. 4, pp. 1323–1331, 2011.

[6] R. Sparrow and M. Howard, "When human beings are like drunk robots: Driverless vehicles, ethics, and the future of transport," *Transportation Research Part C: Emerging Technologies*, vol. 80, pp. 206–215, 2017.

[7] C. J. Kahane and J. N. Dang, "The long-term effect of abs in passenger cars and ltvs," Tech. Rep., 2009.

[8] J. C. Kegelman, "Learning from professional race car drivers to make automated vehicles safer," Ph.D. dissertation, Dept. Mech. Eng., Stanford Univ., Stanford, CA, 2018.

[9] S. M. Erlien, "Shared vehicle control using safe driving envelopes for obstacle avoidance and stability," Ph.D. dissertation, Dept. Mech. Eng., Stanford Univ., Stanford, CA, 2015.

[10] J. Yi, J. Li, J. Lu, and Z. Liu, "On the dynamic stability and agility of aggressive vehicle maneuvers: A pendulum-turn maneuver example," *IEEE Trans. Contr. Syst. Technol.*, vol. 20, no. 3, pp. 663–676, 2012.

[11] D. Tavernini, E. Velenis, R. Lot, and M. Massaro, "The Optimality of the Handbrake Cornering Technique," *ASME J. Dyn. Syst., Meas., Control*, vol. 136, no. 4, pp. 1–11, 2014.

[12] J. Li, Y. Zhang, J. Yi, and Z. Liu, "Understanding agile-maneuver driving strategies using coupled longitudinal/lateral vehicle dynamics," in *Proc. ASME Dyn. Syst. Control Conf.*, Arlington, VA, 2011, Paper # DSCC2011-6152.

[13] E. Frazzoli, "Discussion on 'Optimality properties and driver input parameterization for trailing-braking cornering'," *Europ. J. Control*, vol. 14, no. 4, pp. 321–324, 2008.

[14] C. Canudas de Wit, P. Tsiotras, E. Velenis, M. Basset, and G. Gissinger, "Dynamic friction models for road/tire longitudinal interaction," *Veh. Syst. Dyn.*, vol. 39, no. 3, pp. 189–226, 2003.

[15] S. Shen, J. Wang, P. Shi, and G. Premier, "Nonlinear dynamics and stability analysis of vehicle plane motions," *Veh. Syst. Dyn.*, vol. 45, no. 1, pp. 15–35, 2007.

[16] M. Yamakado and M. Abe, "An experimentally confirmed driver longitudinal acceleration control model combined with vehicle lateral motion," *Veh. Syst. Dyn.*, vol. 46, no. Suppl., pp. 129–149, 2008.

[17] E. Velenis, P. Tsiotras, and J. Lu, "Aggressive maneuvers on loose surface: Data analysis and input parametrization," in *15th Medit. Control Conf.*, Athens, Greece, June 2007, pp. 1–6.

[18] ——, "Modeling aggressive maneuvers on loose surface: The cases of trail-braking and pendulum-turn," in *Proc. Europ. Control Conf.*, Kos, Greece, July 2007, pp. 1233–1240.

[19] V. Gavrilets, E. Frazzoli, B. Mettler, M. Piedmonte, and E. Feron, "Aggressive maneuvering of small autonomous helicopters: A human-centered approach," *Int. J. Robot. Res.*, vol. 20, no. 10, pp. 795–807, 2002.

[20] C. G. Bobier, S. Joe, and J. C. Gerdes, "Sliding surface envelope control: Keep the vehicle within a safe state-space boundary," in *Proc. ASME Dyn. Syst. Control Conf.*, Cambridge, MA, 2010, Paper #DSCC2010-4144.

[21] M. W. McConley, B. D. Appleby, M. A. Dahleh, and E. Feron, "Computational complexity of lyapunov stability analysis problems for a class of nonlinear systems," *SIAM J. Control Optim.*, vol. 36, no. 6, pp. 2176–2193, 1998.

[22] T. Chung and K. Yi, "Design and evaluation of side slip angle-based vehicle stability control scheme on a virtual test track," *IEEE Trans. Contr. Syst. Technol.*, vol. 14, no. 2, pp. 224–234, 2006.

[23] V. Cossalter, A. Aguggiaro, D. Debus, A. Bellati, and A. Ambrogi, "Real cases motorcycle and rider race data investigation: Fall behavior analysis," in *20th Enhanced Safety of Vehicles Conference*, Lyon, France, 2007, pp. 1–13.

[24] A. Heilmeier, A. Wischnewski, L. Hermansdorfer, J. Betz, M. Lienkamp, and B. Lohmann, "Minimum curvature trajectory planning and control for an autonomous race car," *Vehicle System Dynamics*, vol. 58, no. 10, pp. 1–31, 2019.

[25] A. Arab, K. Yu, J. Yi, and D. Song, "Motion planning for aggressive autonomous vehicle maneuvers," in *Proc. IEEE Conf. Automat. Sci. Eng.* Fort Worth, Texas: IEEE, Aug 2016, pp. 221–226.

[26] A. Arab, K. Yu, J. Yi, and Y. Liu, "Motion control of autonomous aggressive vehicle maneuvers," in *Proc. IEEE/ASME Int. Conf. Adv. Intelli. Mechatronics*. Banff, Canada: IEEE, July 2016, pp. 1663–1668.

[27] J. H. Jeon, R. V. Cowlagi, S. C. Peters, S. Karaman, E. Frazzoli, P. Tsiotras, and K. Iagnemma, "Optimal motion planning with the half-car dynamical model for autonomous high-speed driving," in *Proc. Amer. Control Conf.*, Washington DC, 2013, pp. 188–193.

[28] J. C. Kegelman, L. K. Harbott, and J. C. Gerdes, "Insights into vehicle trajectories at the handling limits: analysing open data from race car drivers," *Veh. Syst. Dyn.*, vol. 55, no. 2, pp. 191–207, 2017.

[29] J. Subosits and J. C. Gerdes, "Autonomous vehicle control for emergency maneuvers: The effect of topography," in *Proc. Amer. Control Conf.* Chicago, USA: IEEE, 2015, pp. 1405–1410.

[30] J. Zeng, B. Zhang, and K. Sreenath, "Safety-critical model predictive control with discrete-time control barrier function," *arXiv preprint arXiv:2007.11718*, 2020.

[31] A. Arab and J. Yi, "Safety-guaranteed learning-predictive control for aggressive autonomous vehicle maneuvers," in *Proc. IEEE/ASME Int. Conf. Adv. Intelli. Mechatronics*. Virtual, Boston, USA: IEEE, 2020, pp. 1036–1041.

[32] G. Notomista, M. Wang, M. Schwager, and M. Egerstedt, "Enhancing game-theoretic autonomous car racing using control barrier functions," in *Proc. IEEE Int. Conf. Robot. Autom.* Virtual, Paris, France: IEEE, 2020, pp. 5393–5399.

[33] T. D. Son and Q. Nguyen, "Safety-critical control for non-affine nonlinear systems with application on autonomous vehicle," in *Proc. IEEE Conf. Decision Control*. Nice, France: IEEE, 2019, pp. 7623–7628.

[34] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, "Learning-based model predictive control for safe exploration," in *Proc. IEEE Conf. Decision Control*. Miami, FL: IEEE, 2018, pp. 6059–6066.

[35] B. Olson, S. Shaw, and G. Stépán, "Stability and bifurcation of longitudinal vehicle braking," *Nonlinear Dyn.*, vol. 40, no. 4, pp. 339–365, 2005.

[36] E. Ono, S. Hosoe, H. D. Tuan, and S. Doi, "Bifurcation in vehicle dynamics and robust front wheel steering control," *IEEE Trans. Contr. Syst. Technol.*, vol. 6, no. 3, pp. 412–420, 1998.

[37] P. Falcone, F. Borrelli, H. E. Tseng, J. Asgari, and D. Hrovat, "Low complexity MPC schemes for integrated vehicle dynamics control problems," in *Proc. Int. Symp. Adv. Veh. Control*, Kobe, Japan, 2008.

[38] A. Gray, Y. Gao, T. Lin, J. K. Hedrick, H. E. Tseng, and F. Borrelli, "Predictive control for agile semi-autonomous ground vehicles using motion primitives," in *Proc. Amer. Control Conf.*, Montreal, Canada, 2012, pp. 4239–4244.

[39] J. Yoon, J.-H. Oh, J.-H. Park, S. Kim, and D. Lee, "Autonomous dynamic driving control of wheeled mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Hong Kong, China, 2014, pp. 5274–5279.

[40] H. Nakano, K. Okayama, J. Kinugawa, and K. Kosuge, "Control of an electrica vehicle with a large sideslip angle using driving forces of four independently-driven wheels and steer angle of front wheels," in *Proc. IEEE/ASME Int. Conf. Adv. Intelli. Mechatronics*, France, 2014, pp. 1073–1078.

[41] R. Agrawal, M. Knodler, D. L. Fisher, and S. Samuel, "Virtual reality headset training: Can it be used to improve young drivers' latent hazard anticipation and mitigation skills," *Transportation research record*, vol. 2672, no. 33, pp. 20–30, 2018.

[42] R. Rajamani, *Vehicle dynamics and control.* Springer Science & Business Media, 2011.

[43] A. D. Luca and G. Oriolo, "Modelling and control of nonholonomic mechanical systems," in *Kinematics and dynamics of multi-body systems.* Springer, 1995, pp. 277–342.

[44] H. Zhang and J. Wang, "Adaptive sliding-mode observer design for a selective catalytic reduction system of ground-vehicle diesel engines," *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 4, pp. 2027–2038, 2016.

[45] X. J. Jin, G. Yin, and N. Chen, "Gain-scheduled robust control for lateral stability of four-wheel-independent-drive electric vehicles via linear parameter-varying technique," *Mechatronics*, vol. 30, pp. 286–296, 2015.

[46] L. Li, J. Lian, M. Wang, and M. Li, "Fuzzy sliding mode lateral control of intelligent vehicle based on vision," *Advances in Mechanical Engineering*, vol. 5, pp. 1–7, 2013.

[47] M. Choi and S. B. Choi, "Mpc for vehicle lateral stability via differential braking and active front steering considering practical aspects," *Proceedings of the Institution of Mechanical Engineers, Part D: Int. J. Automot. Eng.*, vol. 230, no. 4, pp. 459–469, 2016.

[48] H. Yoshida, S. Shinohara, and M. Nagai, "Lane change steering manoeuvre using model predictive control theory," *Veh. Syst. Dyn.*, vol. 46, no. S1, pp. 669–681, 2008.

[49] J. M. Park, D. W. Kim, Y. S. Yoon, H. J. Kim, and K. S. Yi, "Obstacle avoidance of autonomous vehicles based on model predictive control," *Proceedings of the Institution of Mechanical Engineers, Part D: Int. J. Automot. Eng.*, vol. 223, no. 12, pp. 1499–1516, 2009.

[50] M. Brown, J. Funke, S. Erlien, and J. C. Gerdes, "Safe driving envelopes for path tracking in autonomous vehicles," *Contr. Eng. Pract.*, vol. 61, pp. 307–316, 2017.

[51] G. Cesari, G. Schildbach, A. Carvalho, and F. Borrelli, "Scenario model predictive control for lane change assistance and autonomous driving on highways," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 3, pp. 23–35, 2017.

[52] M. M. Fateh and A. Arab, "Robust control of a wheeled mobile robot by voltage control strategy," *Nonlinear Dynamics*, vol. 79, no. 1, pp. 335–348, 2015.

[53] M. Souzanchi-K, A. Arab, M. Akbarzadeh-T, and M. M. Fateh, "Robust impedance control of uncertain mobile manipulators using time-delay compensation," *IEEE Trans. Contr. Syst. Technol.*, vol. 26, no. 6, pp. 1942–1953, 2017.

[54] A. Norouzi, R. Kazemi, and S. Azadi, "Vehicle lateral control in the presence of uncertainty for lane change maneuver using adaptive sliding mode control with fuzzy boundary layer," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 232, no. 1, pp. 12–28, 2018.

[55] A. Arab, J. Yi, M. M. Fateh, and S. Arabshahi, "Robust control of a low-cost mobile robot using a neural network uncertainty compensator," in *Proc. ASME Dyn. Syst. Control Conf.*, 2014.

[56] A. Arab and M. M. Fateh, "An uncertainty compensator for robust control of wheeled mobile robots," *Advanced Robotics*, vol. 29, no. 20, pp. 1303–1313, 2015.

[57] A. Haqshenas-M, M. M. Fateh, and S. M. Ahmadi, "Adaptive control of electrically-driven nonholonomic wheeled mobile robots: Taylor series-based approach with guaranteed asymptotic stability," *International Journal of Adaptive Control and Signal Processing*, vol. 34, no. 5, pp. 638–661, 2020.

[58] E. Wachter, A. Schmeitz, F. Bruzelius, and M. Alirezaei, "Path control in limits of vehicle handling: A sensitivity analysis," in *The IAVSD International Symposium on Dynamics of Vehicles on Roads and Tracks.* Springer, 2019, pp. 1089–1095.

[59] E. Wachter, M. Alirezaei, F. Bruzelius, and A. Schmeitz, "Path control in limit handling and drifting conditions using state dependent riccati equation technique," *Proceedings of the Institution of Mechanical Engineers, Part D: Int. J. Automot. Eng.*, vol. 234, no. 2-3, pp. 783–791, 2020.

[60] M. Wang, Z. Wang, J. Talbot, J. C. Gerdes, and M. Schwager, "Game theoretic planning for self-driving cars in competitive scenarios," in *Robotics: Science & Systems*, 2019.

[61] G. Williams, B. Goldfain, P. Drews, J. M. Rehg, and E. A. Theodorou, "Best response model predictive control for agile interactions between autonomous ground vehicles," in *Proc. IEEE Int. Conf. Robot. Autom.* IEEE, 2018, pp. 2403–2410.

[62] U. Rosolia and F. Borrelli, "Sample-based learning model predictive control for linear uncertain systems," *arXiv preprint arXiv:1904.06432*, 2019.

[63] J. Y. Goh, T. Goel, and J. C. Gerdes, "Towards automated vehicle control beyond the stability limits: Drifting along a general path," *ASME J. Dyn. Syst., Meas., Control*, vol. 142, no. 2, pp. 1–10, 2019.

[64] E. Velenis, E. Frazzoli, and P. Tsiotras, "Steady-state cornering equilibria and stabilization for a vehicle during extreme operating conditions," *Int. J. Veh. Autonom. Syst.*, vol. 8, no. 2/3/4, pp. 217–241, 2010.

[65] N. J. Goodall, "Ethical decision making during automated vehicle crashes," *Transportation Research Record*, vol. 2424, no. 1, pp. 58–65, 2014.

[66] N. R. Kapania, J. Subosits, and J. Christian Gerdes, "A sequential two-step algorithm for fast generation of vehicle racing trajectories," *ASME J. Dyn. Syst., Meas., Control*, vol. 138, no. 9, 2016.

[67] E. Velenis, P. Tsiotras, and J. Lu, "Optimal properties and driver input parameterization for trail-braking cornering," *Europ. J. Control*, vol. 14, no. 4, pp. 308–320, 2008.

[68] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011.

[69] A. Dobson and K. E. Bekris, "A study on the finite-time near-optimality properties of sampling-based motion planners," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Tokyo, Japan, 2013, pp. 1236–1241.

[70] J. hwan Jeon, S. Karaman, and E. Frazzoli, "Anytime computation of time-optimal off-road vehicle maneuvers using the rrt*," in *Proc. IEEE Conf. Decision Control*, Dec 2011, pp. 3276–3282.

[71] Y. Kuwata, S. Karaman, J. Teo, E. Frazzoli, J. P. How, and G. Fiore, "Real-time motion planning with applications to autonomous urban driving," *IEEE Trans. Contr. Syst. Technol.*, vol. 17, no. 5, pp. 1105–1118, 2009.

[72] M. Zucker, J. Kuffner, and M. Branicky, "Multipartite rrts for rapid replanning in dynamic environments," in *Proc. IEEE Int. Conf. Robot. Autom.* Roma, Italy: IEEE, 2007, pp. 1603–1609.

[73] C. Voser, R. Y. Hindiyeh, and J. C. Gerdes, "Analysis and control of high sideslip manoeuvres," *Vehicle System Dynamics*, vol. 48, no. S1, pp. 317–336, 2010.

[74] R. Y. Hindiyeh and J. C. Gerdes, "A controller framework for autonomous drifting: Design, stability, and experimental validation," *ASME J. Dyn. Syst., Meas., Control*, vol. 136, no. 5, 2014, Paper #051015.

[75] M. Cutler and J. P. How, "Autonomous drifting using simulation-aided reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom.* Stockholm, Sweden: IEEE, 2016, pp. 5442–5448.

[76] M. Acosta and S. Kanarachos, "Teaching a vehicle to autonomously drift: A data-based approach using neural networks," *Knowledge-Based Systems*, vol. 153, pp. 12–28, 2018.

[77] A. Arab and Y. Mousavi, "Optimal control of wheeled mobile robots: From simulation to real world," in *Proc. Amer. Control Conf.* Denver, USA, Virtual: IEEE, 2020, pp. 583–589.

[78] U. Kiencke and L. Nielsen, *Automotive Control Systems.* New York, NY: Springer-Verlag, 2000.

[79] H. B. Pacejka, *Tire and Vehicle Dynamics*, 2nd ed. Warrendale, PA: SAE International, 2006.

[80] H. Shiobara, T. Akasaka, and S. Kagami, "Two-dimensional contact pressure distribution of a radial tire in motion," *Tire Sci. Technol.*, vol. 24, no. 4, pp. 294–320, 1996.

[81] S.-L. Koo and H.-S. Tan, "Dynamic-deflection tire modeling for low-speed vehicle lateral dynamics," *ASME J. Dyn. Syst., Meas., Control*, vol. 129, no. 3, pp. 393–403, 2007.

[82] K. Yi, K. Hedrick, and S.-C. Lee, "Estimation of tire-road friction using observer based identifiters," *Veh. Syst. Dyn.*, vol. 31, no. 2, pp. 233–261, 1999.

[83] J. Yi, L. Alvarez, X. Claeys, and R. Horowitz, "Tire/road friction estimation and emergency braking control using a dynamic friction model," *Veh. Syst. Dyn.*, vol. 39, no. 2, pp. 81–97, 2003.

[84] M. Mohammadpour, S. Theodossiades, and H. Rahnejat, "Multiphysics investigations on the dynamics of differential hypoid gears," *ASME J. Vib. Acoust.*, vol. 136, no. 4, pp. 1–15, 2014.

[85] G. Erdogan, L. Alexander, and R. Rajamani, "Estimation of tire-road friction coefficient using a novel wireless piezoelectric tire sensor," *IEEE Sensors J.*, vol. 11, no. 2, pp. 267–279, 2010.

[86] J. Li, J. Yi, Z. Liu, and J. Lu, "On the dynamic stability and agility of aggressive vehicle maneuvers: A pendulum-turn maneuver example," in *Proc. ASME Dyn. Syst. Control Conf.*, Cambridge, MA, 2010, Paper #DSCC2010-4032.

[87] C. E. Beal and J. C. Gerdes, "Model predictive control for vehicle stabilization at the limits of handling," *IEEE Trans. Contr. Syst. Technol.*, vol. 21, no. 4, pp. 1258–1269, 2013.

[88] J. Li and J. Yi, "Vehicle motion stability with two vehicle dynamics models," in *Proc. ASME Dyn. Syst. Control Conf.*, Arlington, VA, 2011, Paper # DSCC2011-6195.

[89] A. Anders, A. Fishberg, S. Karaman, and A. Agarwal, "Rapid autonomous complex-environment competing ackermann-steering robot," https://github.com/mit-racecar, 2016.

[90] B. Goldfain, P. Drews, C. You, M. Barulic, O. Velev, P. Tsiotras, and J. M. Rehg, "Autorally: An open platform for aggressive autonomous driving," *IEEE Control Systems Magazine*, vol. 39, no. 1, pp. 26–55, 2019.

[91] M. O'Kelly, H. Zheng, D. Karthik, and R. Mangharam, "F1tenth: An open-source evaluation environment for continuous control and reinforcement learning," in *Proceedings of the NeurIPS 2019 Competition and Demonstration Track*, ser. Proceedings of Machine Learning Research, vol. 123. PMLR, Dec 2020, pp. 77–89.

[92] P. Falcone, F. Borrelli, J. Asgari, E. H. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Trans. Contr. Syst. Technol.*, vol. 15, no. 3, pp. 566–580, 2007.

[93] M. Gerdts, S. Karrenberg, B. Müller-Beßler, and G. Stock, "Generating locally optimal trajectories for an automatically driven car," *Optim. Eng.*, vol. 10, no. 4, p. 439, 2009.

[94] E. Velenis, E. Frazzoli, and P. Tsiotras, "On steady-state cornering equilibria for wheeled vehicles with drift," in *Proc. IEEE Conf. Decision Control*, Shanghai, China, pp. 3545–3550.

[95] Y. Gao, T. Lin, F. Borrelli, E. Tseng, and d. Hrovat, "Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads," in *Proc. ASME Dyn. Syst. Control Conf.* Boston, USA: American Society of Mechanical Engineers Digital Collection, 2010, pp. 265–272.

[96] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1: 43 scale rc cars," *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 628–647, 2015.

[97] V. A. Laurense, J. Y. Goh, and J. C. Gerdes, "Path-tracking for autonomous vehicles at the limit of friction," in *Proc. Amer. Control Conf.* Seattle, USA: IEEE, 2017, pp. 5586–5591.

[98] Y. Gao, A. Grayb, H. E. Tseng, and F. Borrelli, "A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles," *Veh. Syst. Dyn.*, vol. 52, no. 6, pp. 802–823, 2014.

[99] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2002.

[100] L. S. Pontryagin, "Optimal regulation processes," *Uspekhi Matematicheskikh Nauk*, vol. 14, no. 1, pp. 3–20, 1959.

[101] R. Bellman, "The theory of dynamic programming," Rand Corp, Santa Monica, CA, Tech. Rep., 1954.

[102] F. Allgöwer and A. Zheng, "Nonlinear model predictive control." Birkhäuser, 2012, vol. 26.

[103] G. De Nicolao, L. Magni, and R. Scattolini, "Stability and robustness of nonlinear receding horizon control," in *Nonlinear Model Predictive Control*. Springer, 2000, pp. 3–22.

[104] D. M. Raimondo, D. Limon, M. Lazar, L. Magni, and E. F. ndez Camacho, "Min-max model predictive control of nonlinear systems: A unifying overview on stability," *European Journal of Control*, vol. 15, no. 1, pp. 5–21, 2009.

[105] W. W. Hager and H. Zhang, "A survey of nonlinear conjugate gradient methods," *Pacific journal of Optimization*, vol. 2, no. 1, pp. 35–58, 2006.

[106] ——, "An active set algorithm for nonlinear optimization with polyhedral constraints," *Science China Mathematics*, vol. 59, no. 8, pp. 1525–1542, 2016.

[107] ——, "A new active set algorithm for box constrained optimization," *SIAM Journal on Optimization*, vol. 17, no. 2, pp. 526–557, 2006.

[108] A. Arab and A. Alfi, "An adaptive gradient descent-based local search in memetic algorithm applied to optimal controller design," *Information Sciences*, vol. 299, pp. 117–142, 2015.

[109] P. A. Theodosis and J. C. Gerdes, "Generating a racing line for an autonomous racecar using professional driving techniques," in *Proc. ASME Dyn. Syst. Control Conf.*, 2011, pp. 853–860.

[110] N. R. Kapania, J. Subosits, and J. C. Gerdes, "A sequential two-step algorithm for fast generation of vehicle racing trajectories," in *Proc. ASME Dyn. Syst. Control Conf.*, Columbus, Ohio, USA, October 2015, pp. 1–12.

[111] T. Lipp and S. Boyd, "Minimum-time speed optimisation over a fixed path," *International Journal of Control*, vol. 87, no. 6, pp. 1297–1311, 2014.

[112] J. P. Timings and D. J. Cole, "Minimum maneuver time calculation using convex optimization," *ASME J. Dyn. Syst., Meas., Control*, vol. 135, no. 2, pp. 1–9, 2013.

[113] S. LaValle and J. Kuffner, J.J., "Randomized kinodynamic planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, 1999, pp. 473–479.

[114] D. D. Dunlap, E. G. Collins Jr, and C. V. Caldwell, "Sampling based model predictive control with application to autonomous vehicle guidance," in *Florida Conference on Recent Advances in Robotics*, 2008.

[115] Y. Li, Z. Littlefield, and K. E. Bekris, "Asymptotically optimal sampling-based kinodynamic planning," *Int. J. Robot. Res.*, vol. 35, no. 5, pp. 528–564, 2016.

[116] Z. Littlefield, Y. Li, and K. E. Bekris, "Efficient sampling-based motion planning with asymptotic near-optimality guarantees for systems with dynamics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Tokyo, Japan, 2013, pp. 1779–1785.

[117] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts, "Lqr-trees: Feedback motion planning via sums-of-squares verification," *Int. J. Robot. Res.*, vol. 29, no. 8, pp. 1038–1052, 2010.

[118] J.-C. Latombe, "Robot motion planning." Springer Science & Business Media, 2012, vol. 124.

[119] S. M. LaValle, *Planning Algorithms*. New York, NY: Cambridge University Press, 2006, also available at http://planning.cs.uiuc.edu/.

[120] L. Hewing, A. Liniger, and M. N. Zeilinger, "Cautious nmpc with gaussian process dynamics for autonomous miniature race cars," in *Proc. Europ. Control Conf.*, Limassol, Cyprus, 2018, pp. 1341–1348.

[121] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger, "Learning-based model predictive control for autonomous racing," *IEEE Robot. Automat. Lett.*, vol. 4, no. 4, pp. 3363–3370, 2019.

[122] U. Rosolia and F. Borrelli, "Learning how to autonomously race a car: a predictive control approach," *IEEE Trans. Contr. Syst. Technol.*, 2019.

[123] M. M. Baars, H. Hellendoorn, and M. Alirezaei, "Control of a scaled vehicle in and beyond stable limit handling," in *26th IAVSD Symposium on Dynamics of Vehicles on Roads and Tracks*, Gothenburg, Sweden, 2019, pp. 1121–1128.

[124] G. Franze, "A nonlinear sum-of-squares model predictive control approach," *IEEE Transactions on Automatic Control*, vol. 55, no. 6, pp. 1466–1471, 2010.

[125] E. Harinath, L. C. Foguth, J. A. Paulson, and R. D. Braatz, "Nonlinear model predictive control using polynomial optimization methods," in *Proc. Amer. Control Conf.*, Boston, USA, 2016, pp. 1–6.

[126] O. Bastani, Y. Pu, and A. Solar-Lezama, "Verifiable reinforcement learning via policy extraction," in *Advances in Neural Information Processing Systems*, 2018, pp. 2494–2504.

[127] Y. Zhu, D. Zhao, X. Yang, and Q. Zhang, "Policy iteration for h_∞ optimal control of polynomial nonlinear systems via sum of squares programming," *IEEE Transactions on Cybernetics*, vol. 48, no. 2, pp. 500–509, 2017.

[128] S. Sanjari and S. Ozgoli, "Sliding mode control design: a sum of squares approach," *arXiv preprint arXiv:1601.02190*, 2016.

[129] S. P. Khesal and I. Mohammadzaman, "Nonlinear robust roll autopilot design using sum-of-squares optimization," *ASME J. Dyn. Syst., Meas., Control*, vol. 140, no. 11, p. 111005, 2018.

[130] A. Majumdar, A. A. Ahmadi, and R. Tedrake, "Control design along trajectories with sums of squares programming," in *Proc. IEEE Int. Conf. Robot. Autom.*, Karlsruhe, Germany, 2013, pp. 4054–4061.

[131] A. Dalla Libera, R. Carli, and G. Pillonetto, "A novel multiplicative polynomial kernel for volterra series identification," *arXiv preprint arXiv:1905.07960*, 2019.

[132] A. HasanzadeZonuzy, D. Kalathil, and S. Shakkottai, "Learning with safety constraints: Sample complexity of reinforcement learning for constrained mdps," *arXiv preprint arXiv:2008.00311*, 2020.

[133] P. Abbeel, A. Coates, M. Quigley, and A. Ng, "An application of reinforcement learning to aerobatic helicopter flight," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. Platt, and T. Hoffman, Eds. Cambridge, MA: MIT Press, 2007, pp. 1–8.

[134] J. Garcıa and F. Fernández, "A comprehensive survey on safe reinforcement learning," *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 1437–1480, 2015.

[135] Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh, "A lyapunov-based approach to safe reinforcement learning," *arXiv preprint arXiv:1805.07708*, 2018.

[136] T.-Y. Yang, M. Hu, Y. Chow, P. J. Ramadge, and K. Narasimhan, "Safe reinforcement learning with natural language constraints," *arXiv preprint arXiv:2010.05150*, 2020.

[137] A. Khadke, A. Agarwal, A. Mohseni-Kabir, and D. Schwab, "Exploration with expert policy advice," 2019.

[138] R. T. Icarte, T. Q. Klassen, R. A. Valenzano, and S. A. McIlraith, "Advice-based exploration in model-based reinforcement learning," in *Canadian Conf. Artif. Intell.* Springer, 2018, pp. 72–83.

[139] E. Yeh, M. Gervasio, D. Sanchez, M. Crossley, and K. Myers, "Bridging the gap: Converting human advice into imagined examples," *Advances in Cognitive Systems*, vol. 6, pp. 1168–1176, 2018.

[140] T. M. Moldovan and P. Abbeel, "Safe exploration in markov decision processes," *arXiv preprint arXiv:1205.4810*, 2012.

[141] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," in *Proc. AAAI Conf. Artific. Intell.*, vol. 33, no. 01, 2019, pp. 3387–3395.

[142] M. Cutler, T. J. Walsh, and J. P. How, "Real-world reinforcement learning via multifidelity simulators," *IEEE Trans. Robot.*, vol. 31, no. 3, pp. 655–671, 2015.

[143] E. Velenis, D. Katzourakis, E. Frazzoli, P. Tsiotras, and R. Happee, "Steady-state drifting stabilization of rwd vehicles," *Contr. Eng. Pract.*, vol. 19, no. 11, pp. 1363–1376, 2011.

[144] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou, "Information theoretic mpc for model-based reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom.*, Marina Bay, Singapore, 2017, pp. 1714–1721.

[145] F. Zhang, J. Gonzales, K. Li, and F. Borrelli, "Autonomous drift cornering with mixed open-loop and closed-loop control," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 1916–1922, 2017.

[146] D. Isele, "Interactive decision making for autonomous vehicles in dense traffic," in *Proc. IEEE Conf. Intell. Transport. Syst.* Suntec, Singapore: IEEE, 2019, pp. 3981–3986.

[147] F. E. Udwadia, "Fundamental principles of lagrangian dynamics: mechanical systems with non-ideal, holonomic, and nonholonomic constraints," *J. Math. Anal. Appl.*, vol. 251, no. 1, pp. 341–355, 2000.

[148] G. Dalal, K. Dvijotham, M. Vecerik, T. Hester, C. Paduraru, and Y. Tassa, "Safe exploration in continuous action spaces," *arXiv preprint arXiv:1801.08757*, 2018.

[149] I. Gwayi and M. S. Tsoeu, "Rollover prevention and path following of autonomous vehicle using nonlinear model predictive control," in *Open Innovations Conference*. Jyvaskyla, Finland: IEEE, 2018, pp. 13–18.