Active Safety Control for Semi-Autonomous Teleoperated Road Vehicle

Smit Jaman Saparia

TU Delft

TUM

Active Safety Control for Semi-Autonomous Teleoperated Road Vehicle

by

Smit Jaman Saparia

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Monday August 31, 2020 at 10:00 AM.

Student number:	4835964	
Project duration:	September 1, 2019 – August 31, 2020	
Thesis committee:	M.Sc. Andreas Schimpe,	TU Munich, supervisor
	Dr. Laura Ferranti,	TU Delft, supervisor
	Dr. Barys Shyrokau,	TU Delft
	Dr. -Ing. Jens Kober,	TU Delft

This thesis is confidential and cannot be made public until August 31, 2021.

TU Delft

Abstract

The progress in technology has made vehicles safer and the quest to make them even more safe is never ending. Autonomous cars present the solution to make cars much more safer by eliminating the primary cause of road accidents, human error. However, autonomous cars tend to fail in decision making especially in complex traffic environment prompting human intervention, since the technology is not yet mature. The transition to autonomous cars can be achieved via teleoperated driving which allows human operator to remotely control the vehicle via mobile network. This thesis presents Model Predictive Control (MPC) based driver assistance system for semi-autonomous teleoperated road vehicle that helps avoid collision with static and dynamic obstacles. This system aims to mitigate some of the key challenges in teleoperated driving like reduced situational awareness and latency. The proposed system posses the ability to correct lateral and longitudinal motion of the vehicle and explicitly defines its authority to override human operator. To enhance trust in the human operator over the system, visual feedback of the vehicle behaviour is proposed as an additional set of information to the human operator. Simulations were done using high fidelity vehicle model and the results validate the expected behaviour of the system, as designed for teleoperated driving system setup of Institute of Automotive Technology, Technical University of Munich. The designed system is ready for implementation in the actual experimental vehicle and hence real tests can be conducted.

Smit Jaman Saparia
Munich, 5 August 2020

Acknowledgements

I would like to thank Andreas Schimpe for giving me this opportunity to work at Technical University of Munich. His support from the beginning made my transition to Munich easier and welcoming. I truly appreciate the flexibility he offered me while working on this thesis. I would also like to thank Laura Ferranti, my internal supervisor from Delft University of Technology, for her technical insights and guidance. It would not have been possible to accomplish this thesis without the support and guidance of both of my supervisors. I consider myself fortunate to have studied and worked at two of the best universities. I learnt a lot from all the people I met in the last two years in this journey that started in Delft. I would like to thank everyone who had a part in this journey.

Smit Jaman Saparia
Munich, 5 August 2020

Contents

List of Figures

List of Tables

1 Introduction

Safety in automobiles has held prime importance and with progress in technology, automobiles have become safer than before. Popular examples of safety features in cars include seat belt and Antilock Braking System (ABS). Despite the technology making cars safer and reliable, safety is still a persisting issue. Around 25,300 fatalities were reported in European Union in the year 2017 due to road accidents and 135,000 people got seriously injured [1]. It also reported that human error was responsible in almost 95% of all the road traffic accidents, a crucial conclusion that points to the direction in which the efforts should be directed to devise solutions for improving safety. Hence, eliminating human error holds the key to solving the problem of road traffic accidents.

Autonomous vehicles have the potential to solve this problem as they remove human driver from the loop completely. However, they are still far from reality. The perception systems are still incapable to work in all conditions, especially in complex urban traffic scenarios. Main causes of autonomous driving disengagements are owed to software limitations and failures [38]. This limits the use of autonomous vehicles at present. A human driver is generally expected to perform better than an autonomous car in understanding complex environments and taking safe driving decisions. Hence, it is still reasonable to have human in the loop. The transition to autonomous vehicles can be achieved via Teleoperated Driving (ToD). This allows a human operator to remotely control the vehicle in a complex traffic scenario and thus helps overcome the limitations of an autonomous vehicle.

1.1. Teleoperated Driving

ToD is another application of teleoperation, which is simply remote controlling of a device or a machine. It has several applications ranging from Mars rover for exploration, military drones, undersea exploration and telesurgery to name a few [48]. ToD can be deployed when an autonomous car fails in decision making and it has necessary perception sensors working. The potential applications of ToD can be (i) to bring the vehicle out of complex traffic scenario to a place where the vehicle can again work on its own or to a service center for repairs and (ii) to drive SAE Level 4 vehicle if it reaches the limits of operational design domain (ODD) [16]. Thus, ToD can act as a backup technology for autonomous cars and can leverage the transition to autonomous cars.

In ToD, a human operator controls the vehicle remotely from a workstation which is equipped with necessary input devices like steering wheel and pedals for acceleration and braking. The workstation has monitors that display the environment of the vehicle as captured by perception sensors installed on the vehicle. These sensors include camera and LIDAR (Light detection and ranging). The input commands given by the operator are transmitted over a wireless network to the vehicle which are then fed to the actuators on the vehicle. The information captured by the perception sensors is sent back to the operator's workstation for displaying on the monitors at workstation.

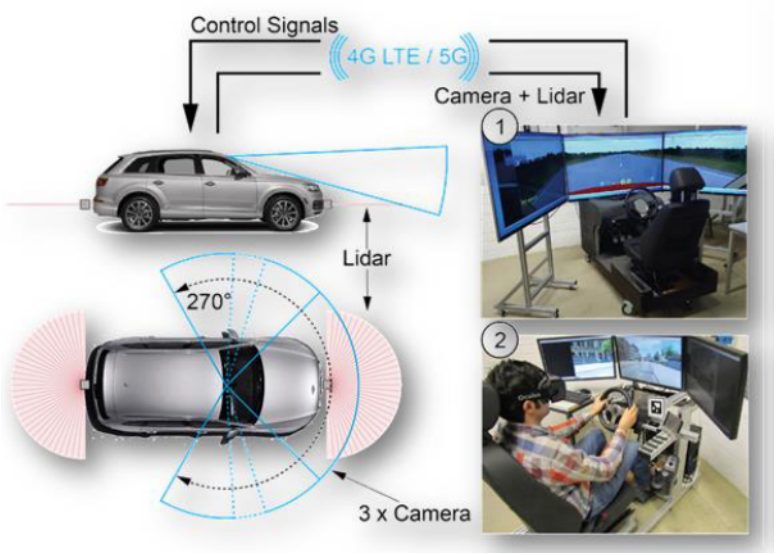


Figure 1.1: Teleoperated Driving using the experimental vehicle Audi Q7 at the TUM Automotive Institute of Technology. The insets in this figure show (1) Video stream displayed on the three screens (2) a human operator watching the video stream on Head Mounted Display (HMD).

The Institute of Automotive Technology or Lehrstuhl für Fahrzeugtechnik (FTM) at Technical University of Munich (TUM) has been involved in the research on teleoperated vehicle since 2010 [31]. The ToD system setup at FTM is illustrated in the Figure 1.1. In the following, the challenges faced in ToD are discussed. Thereafter, the system setup of FTM is described. It explains the necessary hardware installed on the experimental vehicle that helps to modify it to make it a teleoperated vehicle.

1.1.1. Challenges

ToD inherently has some challenges. The human operator manages Dynamic Driving Tasks (DDT) [16] based on the information he gets from the monitors while sitting in the workstation. This affects driving performance and is incomparable to managing DDT from within the vehicle itself. Therefore, ToD is challenging and some of these challenges are mentioned here:

1. **Potential Reduction in Situational Awareness**
   From the video stream only, it becomes difficult to judge velocity and location of the ego vehicle and other obstacles like cars and vulnerable road users (VRU). This problem aggravates in the peripheral region of the view. As a result, it adds to the mental effort needed to compensate for distortions in image and recreate missing information from the sensor data [14].

2. **Latency in the Network**
   It is the communication time delay which is induced due to the transmission of signals over a wireless network, typically a commercial 4G Long Term Evolution (LTE) network. Time taken to transmit signals from vehicle to workstation is more than that in the opposite direction since it involves larger data. However, it should be noted that, large fraction of this time is consumed in processing the signals and not the transmission over network. Increase in latency starts to affect driving performance. There are several techniques to mitigate the effects of latency. In section 1.1.2 the ToD system setup is described along with the hardware specifications which are carefully optimized to achieve minimum latency for a good quality of video stream.

3. **Connection Loss**
   Apart from communication time delays, complete connection loss is also a problem in teleoperation. There is always a possibility of connection loss due to reasons like bad weather, equipment faults etc. Hence, the teleoperated vehicle should be capable of dealing with connection loss and ensure safety of the passengers and other traffic participants.

All the factors discussed above increase the probability of collision with obstacles. For teleoperation to be successfully accepted for commercial use, it should guarantee safety and reliability especially in

the scenarios in which it is expected to be deployed. One solution to tackle this problem is to develop a driver assistance system that aids the human operator and helps avoid collision. This clearly highlights the objective of this work and lays down the features of the Active Safety Controller (ASC) to be designed.

1.1.2. System Setup at FTM

FTM has Audi Q7 as an experimental vehicle. The vehicle is equipped with necessary hardware like sensors, computer and communication units. As described in Hosseini [32], the description of the experimental vehicle is summarized here.

There are three front facing cameras above the windscreen which captures 240° of the surroundings. In addition to these, two cameras are mounted on the side mirrors and one camera is mounted on the rear of the car. The cameras used on the vehicle are BSF (FLIR BFS-U3-28S5C-C) from FLIR Systems. Apart from cameras, SICK LMS 291 LIDAR sensors are also mounted on the vehicle, one each on the front and rear. The LIDAR scans 180° of the environment in one layer with an angular resolution of 0.5° or 1°. The vehicle is also equipped with OxTS RT3003 precision inertial and GPS navigation systems, which measures motion, position and orientation of the vehicle.

The vehicle is upgraded with actuators to manipulate the steering wheel, acceleration and brake pedals, gear shifting using shift-by-wire and usual buttons for honking and switching on and off the headlights. All these can be also controlled directly by a safety driver from with the vehicle if needed.

Several computation units for specific tasks are also installed on the experimental vehicle. The main computational unit called CarPC is 16 core industrial PC which serves as the main interface between sensors, actuators, and communication unit. A MicroAutoBox from dSPACE [26] is used for rapid control prototyping (RCP) that translates the received commands from CarPC to the actuators.

A commercial LTE router is used for communication in the vehicle. LTE SIM card from Telekom is used during testing. Additionally, a secure connection using VPN is created between operator's workstation and the vehicle.

The operator's workstation is equipped with three 31 inches Acer Predator monitors that display the environment. SENSO-Wheel SD-LC [47] steering wheel and pedals from SENSODRIVE [30] are used as input devices. In addition to this set of input devices from SENSODRIVE, another set from Fanatec [2] is also available for use.

Latency influences driving performance in ToD [7]. Efforts are always made to reduce latency in teleoperation for best performance. The system latency in teleoperation can be classified as follows:

1. **Actuator Latency**
   It is the time difference between the operator input command and the corresponding actuator response inside the vehicle. The measurements of the FTM setup show that total actuator latency using commercial LTE is equal to 79.30 ms out of which large fraction is on the vehicle side of the chain. It takes 60.24 ms to convert Controller Area Network (CAN) signal form Autobox into and actual steering wheel angle.

2. **Glass Latency**
   It is the time difference between the sensor data collected from the environment, transmitted over the network to the operator workstation and then displayed on the monitors. Glass latency for the FTM setup using commercial network is equal to 121.05 ms.

Therefore, the total end-to-end latency for the FTM setup is equal to 200.35 ms. This accounts for the entire signal path including the transmission via network. The camera sensors, its framerate and resolution, encoders for encoding the camera images and the display monitors of the FTM setup are carefully chosen and optimized to achieve minimum latency for optimal quality of the video stream and presentation. The FTM latency can be considered as very low. To get an intuitive understanding of this number and its importance, a comparison can be made with reaction times of regular drivers and also

with sportsperson. As mentioned in the news article [50], Usain Bolt's reaction time of 155 ms in 2016 Olympics 100 m race was the second slowest and yet he won the gold medal. Valtteri Bottas' 201 ms reaction time at 2017 Austrian Grand Prix race start was considered too low. He won the that race and it was only his third win out of his more than 100 race starts until that race. A normal human driver would typically need half a second to respond. Although several other factors play a role in determining the final outcome of results in sports, this still presents a broad picture of reaction times and its significance. In comparison to these reaction times, the FTM latency can be states as low, and hence can be neglected for this work. Furthermore, to prove this proposition, simulation results are presented in 4.5 for validation.

It is also important to mention the limitations of the current FTM setup. The perception system can recognize the obstacles position in the environment, but it cannot estimate the boundaries of the lane and hence lane width and boundary information cannot be used for controller design. Moreover, the setup is not supported by sophisticated software to predict the obstacle velocity in future and hence this limits the use of complex obstacle avoidance strategies for dynamic obstacles.

The challenges of ToD and the limitations and capabilities of system setup highlights the need of a safety system to improve vehicle safety while being teleoperated. The safety system fundamentally constitutes of a controller that would be implemented in the CarPC in the vehicle. Since the human operator actively drives the vehicle, inclusion of the safety system makes it a case of shared control. Even though it is a case of shared control for ToD, ideas can be taken from literature that addresses the problems of collision avoidance under shared control for a regular setup.

1.2. Related Work

In this section, various literature on shared control strategies and collision avoidance for semi-autonomous road vehicle are discussed. In the course of the discussion, it becomes clear what strategy suits well for the design of safety system and also highlights the shortcomings of some of the strategies used in other work. These shortcomings are then addressed in the design of the controller of the safety system. The human operator controls the vehicle by setting steering wheel angle and velocity (via brake and accelerator pedals) of the vehicle. Thus, these become input to the controller which are then manipulated to meet the objective and then given as output which are then fed to the actuators on the vehicle.

1.2.1. Shared Control Strategy

The human operator is responsible for managing (i) DDT and (ii) Object and Event Detection and Response (OEDR) [16] all the time while the controller works in parallel to ensure collision avoidance. Therefore, it becomes a case of shared control. There are several ways in which shared control is realized. Some of the major ways of shared control are described here:

a. Model steering system and driver

This involves modelling steering system and human driver which are then incorporated in the controller formulation. This approach helps model the driver reaction or the intent and predict the future vehicle states accordingly. Therefore, it also helps to calculate the vehicle states due to collision causing uncertain driver inputs. The probability of such dangerous inputs increases due to lack of situation awareness or high workload. In this approach an electric power steering (EPS) actuator is used for torque intervention to the human input and hence it becomes essential to model the steering system [19]. The driver model and the steering model are coupled with the vehicle model where the input to the system is the controller input. The controller acts only when the driver is not able to steer the car without violating the safety constraints.

To model the human driver, some of its parameters are needed. This usually depends on the neuromuscular properties of the arm and the driving style of the driver [12]. In Falcone et. al. [21], it is shown how to estimate the gains of the driver model presented by using a non linear least squares problem. Thus, it involves collecting the data from actual driving tests for different types of driving behaviors like normal and aggressive driving styles. This additionally needs measurement setup and sensor fusion

algorithms as described in Falcone et. al. [21] making it a complex method to implement. Moreover, the driver model needs tuning and becomes specific to a particular driver and needs adjustments for a new driver.

b. Threat metric
To account for the driver and controller input, a threat metric can be devised that helps to blend the inputs from both the driver and controller. This approach is shown in Anderson et. al. [3] where a threat metric as a function of slip angles is devised. The idea is that when the vehicle is close to the obstacle and trying to evade it, the vehicle will have high slip angle values. Whereas, in the cases when the vehicle is relatively far from the obstacle, the slip angle values will be low. An intervention function translates this threat metric Φ (function of slip angles), into a scalar blending gain K in the range of 0 to 1. This is finally used to blend the driver and controller input. For the sake of completeness this is presented here,

$$u_v = K(\Phi)u_{MPC} + (1 - K(\Phi))u_{dr} \tag{1.1}$$

where, u_v is the blended input which is applied to the vehicle, u_{MPC} is the controller input, u_{dr} is the driver input and $K(\Phi)$ is the intervention function in the range 0 to 1. The extent of control authority is thus shifted between the driver and the controller based on the threat assessment with a simple idea of high threat means more controller intervention. The authors have also presented another threat metric as a function of an objective function and compared it with the one based on slip angle. The main idea of a blended input based on a threat metric remains the same.

It should be noted that such a technique needs tuning of the metric, which is usually not trivial. It also does not correct longitudinal motion of the vehicle, which is sometimes necessary when collision cannot be avoided only by steering. In situations when the vehicle steers to avoid an obstacle contrary to the driver input, he or she is not aware of the extent to which the controller has the authority. This can lead to problems of trust in the system.

c. Match the driver inputs
Another approach for shared control is to simply match the driver inputs. In this way, matching the driver commands becomes the control objective along with other objectives of collision avoidance and vehicle stability [20]. This eliminates the need to model a driver and steering system or creating a threat metric or a heuristic function to decide when to switch control between the human and the controller. The controller is always active in the background and intervenes only when the collision and safety constraints are violated. This approach is realized by using Model Predictive Control (MPC) as presented in Erlien et. al. [20].

1.2.2. Collision Avoidance Strategy
The controller needs to ensure collision avoidance with both static and dynamic obstacles and thus it needs to know the velocity as well as position of the obstacles. Some of the major methods for collision avoidance are presented here:

a. Velocity obstacles
Velocity obstacles represents the set of velocity vectors or the ”velocity obstacles” for a robot, which will result in a collision between the robot and the moving obstacles. Any velocity taken out of the set helps to avoid collision [49].There are several variants of the velocity obstacles such as Directive Circles for path planning of holonomic robots [39], Dynamic Window Approach (DWA) where dynamics of the robot are considered in choosing the set of collision free velocity that respects robot dynamics and obstacles [24] and Probabilistic Velocity Obstacle (PVO) that considers errors in the estimation of velocities and dimensions of the obstacles [25].

It should be noted that the above methods are more relevant for a fully autonomous, holonomic and slow moving robots and obstacles. A car is a non-holonomic robot where shared control will be employed for this work and will not necessarily be used for only low speed maneuvers. For shared control, where the operator manages DDT and doesn’t select goal position, it becomes considerably irrelevant to implement these techniques.

b. Potential field
Obstacles can be represented by potential field and this can be added into the problem formulation of controller. Such a flexibility is given by Model Predictive Control (MPC) and hence it becomes the preferred choice of control technique. Different obstacles can be represented with different potential fields according to their characteristics such as crossable and non-crossable as shown in Rasekhipur et al. [44]. Potential fields for collision avoidance can be used for semi-autonomous and fully autonomous vehicles. A similar concept called virtual force field, which fundamentally works on the same principle, is shown to work for slow speed robots sharing workspace with other obstacles and humans [55]. In this concept, several modifications were done to reduce path oscillations and time to reach goal.

c. Spatial constraint
In this method, obstacles are represented as a geometrical shape, typically an ellipse. These are added as constraint of the controller problem formulation for collision avoidance. Again, such a flexibility is provided by MPC and hence it becomes the preferred choice of control technique. Detection and localization of the obstacle depends on the perception system used on the vehicle. As shown in Ferranti et. al. [22], Dynamic Bayesian Networks (DBN) are used to predict the future position of the pedestrians which is a Mixture of Gaussians (MoGs) and then this is used in the constraints of the optimization problem. A DBN is a Switching Linear Dynamical System (SLDS) consisting of two Linear Dynamical Systems (LDSs). In Kooij et. al. [36], it is described in detail how DBN works. The system relies on the interactions between the perception and planning modules. The authors have shown this approach for full autonomous car moving at slow speed.

Instead of predicting the future position of the obstacles using DBN, obstacles can also be assumed to move with constant velocity for a particular prediciton horizon. This technique has been used for a shared control problem where collision avoidance is incorporated in the constraints [46]. The obstacles are approximated as ellipses and the ego vehicle is approximated by four circles. This approximation has been further improved in Brito et. al. [11] to avoid possible collisions due to inaccurate approximation of the geometrical shape. This method circumvents the need to use techniques like DBN to predict future obstacle position.

1.2.3. Conclusion from Related Work
The related work discussed above highlights the strengths and weaknesses of the methods deployed for shared control and collision avoidance. Based on this and the capabilities of the ToD system setup at FTM, the most suitable method for this work can be chosen.

For shared control, the controller should be designed such that its objective is to match the driver inputs. The advantage of this approach can be summarized as follows:

1. It eliminates the need to model driver and steering system.
2. No threat metric needs to be designed. This also helps eliminate extra tuning.
3. This method can be easily implemented in the current ToD setup of FTM.

For collision avoidance, the obstacles and ego vehicle can be represented as geometrical shapes and a collision avoidance constraint based on this information can be added in the problem formulation of the controller. Dynamic obstacles can be assumed to move with the same velocity for a desired prediction horizon. The advantages of this approach are:

1. It works for non-holonomic vehicle.
2. Works independent of the start and goal position of the vehicle.
3. No sophisticated prediction of future obstacle position is needed.
4. This method also can be easily implemented in the current ToD setup of FTM.

From the above choices of methods for shared control and collision avoidance, the type of control technique that suits clearly points to MPC. Therefore, it becomes necessary to use MPC for this work.

MPC can calculate optimal control input for the given optimization problem. It can consider model of the vehicle and can handle multitude of constraints.

Gaps identified in the related work
The approaches used in Anderson et. al. [3] and Erlien et. al. [20], correct steering input for control of lateral motion of the vehicle. The extent of steering correction is not restricted, and the driver does not actually know what the system is doing against his wish. In addition to this, the only scope of collision avoidance while heading towards an obstacle is steering, which might add infeasibility issues since it eliminates the scope of stopping via longitudinal motion control.

1.3. Proposed Approach

On the basis of the ToD challenges, capabilities and limitations of the FTM setup and the gaps identified in the literature on collision avoidance for shared control, the importance and requirements of a safety system needed to overcome these challenges has been highlighted. Therefore, this work proposes a driver assistance system called Active Safety Controller (ASC) that works along with human operator and helps avoid collision with static and dynamic obstacles.

The fundamental component of the ASC that is responsible to achieve the aforementioned objectives is MPC based controller. A general description of the system architecture given in the following clarifies how the overall system looks like.

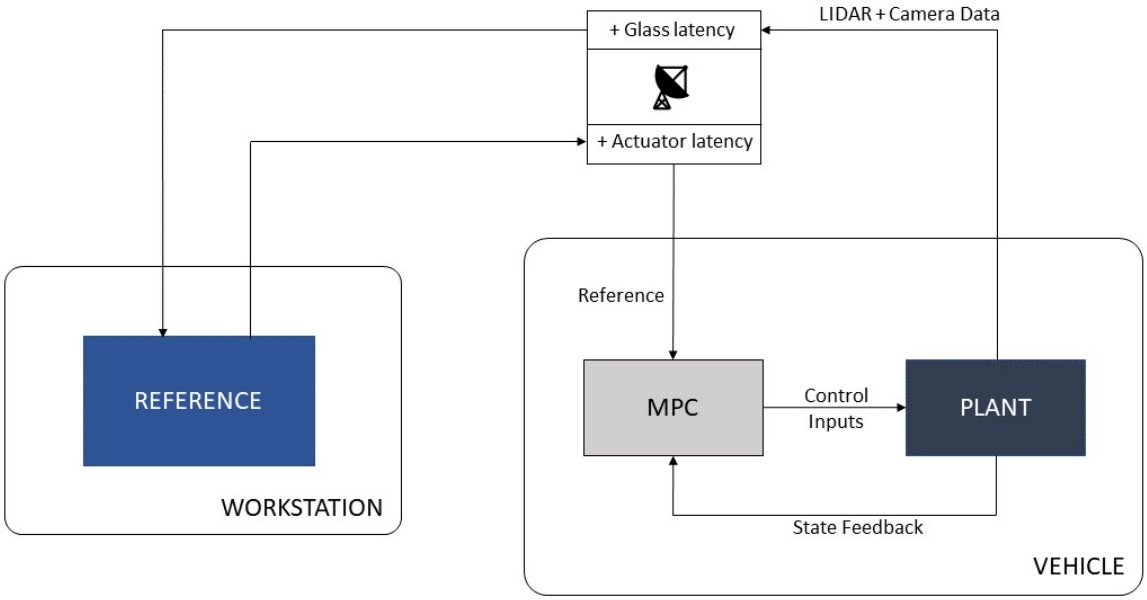


Figure 1.2: ToD system architecture showing the major components in closed loop.

Figure 1.2 illustrates the closed loop architecture of the ToD setup equipped with MPC based controller. The system is divided into (i) workstation where the human operator sits and generates driving inputs (ii) teleoperated vehicle, which is equipped with MPC based controller that finally controls the longitudinal and lateral motion of the vehicle to ensure collision avoidance. It is this controller that transforms the setup into a shared control type, which is designed to overcome the gaps identified in the literature as mentioned before. The workstation and the vehicle communicate via a commercial 4G LTE network and hence delays the signals by time equal to latency in the stream. As mentioned before, signals from the workstation to the vehicle are delayed by actuator latency and signals from vehicle to the workstation are delayed by glass latency. It should be again noted that the latency values account for the signal path from source to destination and not merely the additional time taken for signals to travel via

the network. The transmission of signals via a network makes shared control in ToD different from a regular shared control driving.

To realize the above objectives, the proposed features of the ASC can be listed. In addition to this, it becomes important to define the use cases of ToD in order to clarify capabilities of the ASC and the constraints under which it will operate. To summarise, the roles of both the controller and human operator can be established that helps clarify how will the shared control interface look like.

Before listing the proposed features of the ASC, it is important to mention the scenarios in which ToD will be deployed with the current system setup. The following are scenarios under which ToD is expected to be employed:

1. Vehicle fails in decision making and necessary perception sensors are working to accomplish ToD.
2. A SAE Level 4 vehicle [16] reaches its operational limits and needs human intervention to drive the vehicle further.

Based on the above scenarios, experimental scenarios are created for simulations that help check the expected behaviour of the ASC.

The following are features of the ASC that are included to help meet the objectives of this work and also help overcome the gaps identified in the literature.

1. **Lateral control with restriction on controller authority**
   Lateral control of the vehicle helps steer the vehicle away from obstacle with an additional constraint on steering wheel deviation from the reference input. This constraint aims to restrict the extent to which controller can deviate from human operator's steering wheel input. Therefore, it defines the controller's authority to manipulate the operator's inputs.
2. **Visual feedback to the operator**
   This feature aims to add extra information for the operator on the monitors of the workstation. The following illustrations are added to the operator screen:
   - **White Tracks**: The future trajectory as taken by the wheels of the ego vehicle due to complete sequence of controller steering and velocity inputs, as influenced by current operator inputs is shown as white tracks. This helps the operator to know, in which direction will the vehicle travel due to the combined effect of his current input and the complete sequence of controller steering and velocity inputs.
   - **Red Cone**: A red cone around the ego vehicle representing the controller authority. The limits of the red cone are formed by the trajectory followed by vehicle due to the maximum and the minimum controller deviation from the reference steering wheel input (as given by the operator). This helps the operator understand how much authority the controller has to override his inputs. It is helpful especially when the controller is already overriding operator inputs.
3. **Longitudinal control**
   A longitudinal motion control capability of the controller helps to stop the vehicle if needed for collision avoidance. It can also help avoid infeasibility issues.

It should be noted that the controller works along with the human operator. Therefore, it is important to list the roles of the human operator and of the controller to identify what to expect from each of the participant.

Roles of the Human Operator

1. The operator can be considered as a trained pilot whose job is to drive the vehicle out of the complex traffic environment to a safe place or to service station. Therefore, s/he cannot be expected to get distracted.

2. The operator is expected to manage DDT and OEDR all the time.

Roles of the Controller

1. The controller has to work in parallel all the time and intervene only momentarily for collision avoidance with dynamic or static obstacles.

2. The controller can control both lateral and longitudinal motion of the vehicle.

As mentioned in the section 1.1.2, the ToD setup at FTM has very low network latency and hence no latency mitigation techniques may be required for this particular setup. However, for setups working under high latency, latency mitigation techniques can be necessary. One such latency mitigation technique for ToD is predictive display. As a part of research, predictive display methods were explored and one unique method for predictive display was formulated. This proposed predictive display method was compared against the baseline predictive display method to investigate it's validity and highlight the usefulness for ToD setups working under high latency. It should be noted that this concept could be also useful for other applications of teleoperation. The simulation results for predictive display method are presented in the chapter 4

To conclude, the objectives of this thesis are identified in this section and the features of the proposed approach are laid.

1.4. Thesis Outline

The rest of the thesis presents the design of the MPC based ASC. Chapter 2 discusses background information on vehicle modelling. It gives an understanding of the system to be controlled, and presents choices of models that can be used as prediction model of MPC. It also describes how MPC works and further presents information on the solver used to solve the optimization problem. Chapter 3 presents the design of ASC and the complete problem formulation of MPC to achieve the objectives described above. It also covers the features of ASC that helps human operator better understand vehicle behaviour due to the combined actions of him and the MPC. Chapter 4 discusses the design and tuning of all the other components required for setting up the simulation setup. It also presents the performance comparison of ASC with the baseline approach and also verifies the ASC performance under the effect of latency. Predictive display method for ToD setups working under high latency is also presented in this chapter. Chapter 5 finally concludes the thesis and presents scope of future work.

2

Preliminaries

The design of Model Predictive Control (MPC) based Active Safety Controller needs several important components like model of the system and a suitable solver to find the optimal solution. MPC being a model based control technique uses a model of the system to be controlled as its prediction model. As the system here is the vehicle, two popular types of vehicle models are considered. Based on the final application and functionalities offered by each of the vehicle models, the most suitable for the prediction model is selected. Later in this chapter, a brief explanation of MPC and the solver used for solving the optimization problem is presented.

2.1. Vehicle Modelling

Vehicle and driver assistance systems development rely heavily on simulations. The reliability of the simulation results is affected by the choice of the vehicle model that captures the kinematic and dynamic behaviour of the vehicle. There are a variety of vehicle models available ranging from a simple bicycle model to a complex multi-body model with several degrees of freedom. A complex model helps accurately replicate the dynamic behaviour of the vehicle in reality. The vehicle model should be chosen depending on the application and the need to closely approximate the actual vehicle behaviour.

For the sake of completeness, two main vehicle models namely, kinematic model and dynamic model, are presented. In these models, front wheels are assumed to be represented by a single wheel (similarly for the rear wheels) [42]. This assumption exploits the advantage of symmetry in the vehicle. These models are only for a front wheel steered vehicle. Furthermore, steering wheel angle mismatch between left and right wheels is neglected and only planar motion is considered.

2.1.1. Kinematic Model

A kinematic bicycle model represents in general behaviour of the vehicle by using non-linear equations of motion and is based purely on the vehicle geometry. This approach is popular for study of lateral motion. It does not consider any tire slip angle which is defined as the angle between direction in which the tire is oriented and the orientation of velocity vector of the tire [43]. In addition to this, the kinematic bicycle model also does not consider longitudinal slip, the difference between tangential velocity of the tire and longitudinal velocity of the wheel axle [43]. This model is suitable in approximating vehicle behaviour for slow speed driving on dry road conditions.

The following equations describe the kinematic bicycle model using nonlinear continuous time equations in the inertial frame [35, 43] referring to the figure 2.1:

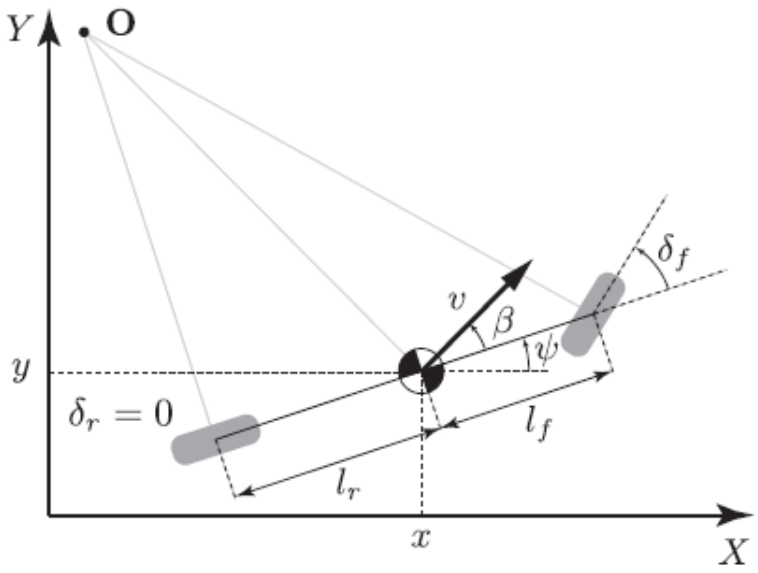


Figure 2.1: Kinematic Bicycle Model [35]

$$\dot{x} = v\cos(\psi + \beta) \tag{2.1a}$$
$$\dot{y} = v\sin(\psi + \beta) \tag{2.1b}$$
$$\dot{\psi} = \frac{v}{l_r}\sin(\beta) \tag{2.1c}$$
$$\dot{v} = a \tag{2.1d}$$
$$\beta = \tan^{-1}\left(\frac{l_r}{l_f + l_r}\tan(\delta)\right) \tag{2.1e}$$

where x and y are coordinates of Center of Mass (CoM) of the vehicle, ψ is the heading angle of the vehicle in the inertial frame, v is the vehicle velocity, l_f and l_r are the distances between CoM and front and rear axles respectively. β is the angle that the velocity vector makes with the longitudinal axis of the vehicle, a is the acceleration of the CoM and δ is the front road wheel angle.

2.1.2. Dynamic Model

Dynamic vehicle model considers the forces acting on the vehicle to describe motion. For designing systems that make the vehicle operate at limits of handling, it is necessary to use a dynamic vehicle model to accurately capture the vehicle behavior. Other than gravitational and aerodynamic forces, tires create all the forces acting on the vehicle. This is the reason why slip angles and slip ratios of the tires are considered in dynamic model in contrast to kinematic model. Method to determine tire forces relevant to the application in this work is presented later in this section.

The notations of the dynamic model are illustrated in the Figure 2.2, where the heading angle of the vehicle with respect to x-axis in the inertial frame is ψ, the road wheel angle of the front wheel with respect to vehicle longitudinal axis is δ, velocity v of the vehicle CoM is inclined to vehicle longitudinal axis by β, the body side slip angle and v_x and v_y are its longitudinal and lateral components in the vehicle frame. The lateral tire force is given by f_{y*} acting perpendicular to its direction and longitudinal tire force acting along its direction of inclination is given by f_{x*}, both for front f and rear r wheels. The distance of CoM to front axle is l_f and to rear axle is l_r.

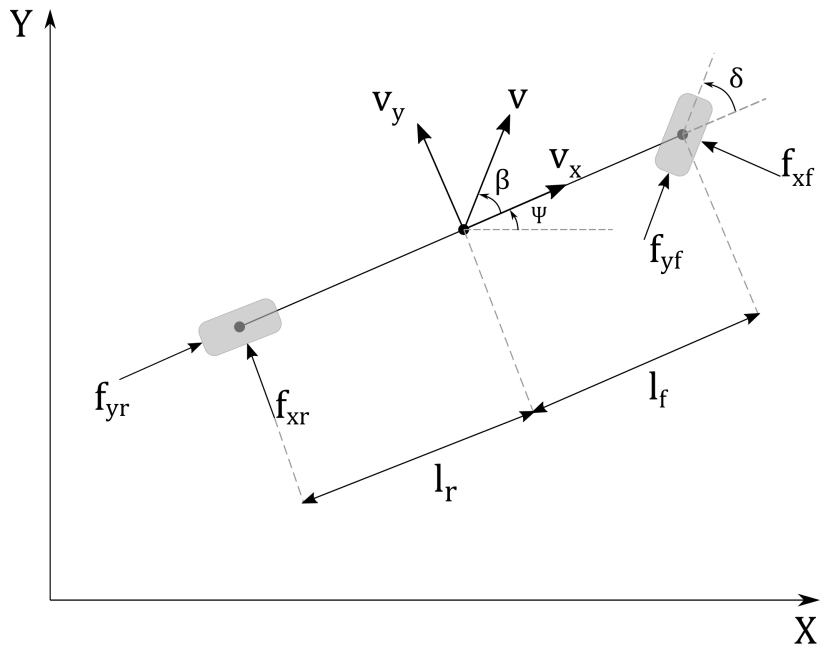


Figure 2.2: Dynamic Bicycle Model

2.1.2.1. Equations of Motion

As presented in Liniger et al.[37] and Velenis et. al.[52] the vehicle's equations of motion are:

$$ma_x = F_{xf} + F_{xr} \tag{2.2a}$$

$$ma_y = F_{yf} + F_{yr} \tag{2.2b}$$

$$I_z\ddot{\psi} = F_{yf}l_f - F_{yr}l_r \tag{2.2c}$$

where F_{x*} and F_{y*} are the net longitudinal and lateral forces acting on the vehicle in the vehicle frame, I_z is the moment of inertia around the normal axis with $\dot{\psi}$ as the heading rate. The lateral a_x and longitudinal a_y accelerations are given as:

$$a_x = \dot{v}_x - v_y\dot{\psi} \tag{2.3a}$$

$$a_y = \dot{v}_y + v_x\dot{\psi} \tag{2.3b}$$

The net longitudinal and lateral forces in terms of the tire forces in the vehicle frame are given as:

$$F_{xf} = f_{xf}\cos\delta - f_{yf}\sin\delta \tag{2.4a}$$

$$F_{yf} = f_{xf}\sin\delta + f_{yf}\cos\delta \tag{2.4b}$$

$$F_{xr} = f_{xr} \tag{2.4c}$$

$$F_{yr} = f_{yr} \tag{2.4d}$$

Substituting (2.3) and (2.4) in (2.2) the following differential equations are derived,

$$\dot{v}_x = v_y\dot{\psi} + \frac{1}{m}\left(f_{xf}\cos\delta - f_{yf}\sin\delta + f_{xr}\right) \tag{2.5a}$$

$$\dot{v}_y = -v_x\dot{\psi} + \frac{1}{m}\left(f_{xf}\sin\delta + f_{yf}\cos\delta + f_{yr}\right) \tag{2.5b}$$

$$\ddot{\psi} = \frac{1}{I_z}\left((f_{xf}\sin\delta + f_{yf}\cos\delta)l_f - f_{yr}l_r\right) \tag{2.5c}$$

The equations of motion in the inertial frame can be given as:

$$V_X = v_x \cos\psi - v_y \sin\psi \tag{2.6a}$$

$$V_Y = v_x \sin\psi + v_y \cos\psi \tag{2.6b}$$

2.1.2.2. Tire Forces

It is important to determine the tire forces and they can be highly nonlinear for high values of slip angles and slips ratios. Tire modelling is therefore an important topic in vehicle dynamics. For vehicles operating at slow speeds under moderate driving conditions for dry road surfaces, the slip angles and slips ratios are generally small and hence the tires are said to operate in the so called linear region. The vehicle in ToD is also expected to be operating under similar conditions and hence these approximations can be used to determine the tire forces.

Referring to Rajamani [43], slip ratio is the difference between the actual longitudinal velocity at the axle of the wheel V_X and the equivalent rotational velocity $r_{eff}\omega_w$ of the tire where, r_{eff} is the effective tire radius and ω_w is the rotational velocity of the tire. The longitudinal slip ratio is defined for the braking and accelerating conditions as,

$$\sigma_x = \begin{cases} \frac{r_{eff}\omega_w - V_X}{V_x} & \text{, during braking} \\ \frac{r_{eff}\omega_w - V_X}{r_{eff}\omega_w} & \text{, during accelerating} \end{cases} \tag{2.7}$$

For small slip ratios, the longitudinal tire force is proportional to the slip ratio and the following relationship can be used to determine longitudinal tire forces.

$$f_{xf} = C_{\sigma f}\sigma_{xf} \tag{2.8a}$$

$$f_{xr} = C_{\sigma r}\sigma_{xr} \tag{2.8b}$$

here, $C_{\sigma f}$ and $C_{\sigma r}$ are the longitudinal tire stiffness parameters for front and rear tires respectively. Likewise, the lateral tire forces are proportional to the small values of tire slip angles and thus the following relations can be obtained,

$$f_{yf} = C_\alpha \alpha_f \tag{2.9a}$$

$$f_{yr} = C_\alpha \alpha_r \tag{2.9b}$$

where, C_α is the cornering stiffness and the front α_f and rear α_r slip angles are

$$\alpha_f = \delta - \theta_{Vf} \tag{2.10a}$$

$$\alpha_r = -\theta_{Vr} \tag{2.10b}$$

where θ_{v*} is the angle made by the tire velocity vector with vehicle's longitudinal axis and can be calculated for the front and rear tires using

$$\tan\theta_{Vf} = \frac{v_y + l_f\dot{\psi}}{v_x} \tag{2.11a}$$

$$\tan\theta_{Vr} = \frac{v_y - l_r\dot{\psi}}{v_x} \tag{2.11b}$$

using small angle approximations, the final form of lateral forces can be written as,

$$f_{yf} = C_\alpha\Big(\delta - \frac{v_y + l_f\dot{\psi}}{v_x}\Big) \tag{2.12a}$$

$$f_{yr} = C_\alpha\Big(-\frac{v_y - l_r\dot{\psi}}{v_x}\Big) \tag{2.12b}$$

If the vehicle is travelling at high speeds pushing it to the limits of handling where the tires tend to operate in the non-linear regime then more accurate techniques need to be used to determine the corresponding tire forces such as Pacejka Magic Formula [5].

2.2. Model Predictive Control

Model Predictive Control is a model based optimal control approach. As described in Bemporad [6], it considers model of the open-loop process that abstracts the behaviour of the system. MPC can consider a multitude of constraints on the states and inputs of the system, usually dictated by the actuator limitations and desired range in which states and inputs are expected to be. Based on the constraints, model of the system and the desired performance specifications, an optimal control problem (OCP) is formulated that considers weights on tracking error and actuator efforts. This OCP is then translated into an equivalent optimization problem which depends on the initial state and reference signals. Then at each sampling time, the optimization problem is solved by setting the current state as the initial state. The OCP is formed for a particular time horizon starting from the current time to a finite time in the future and hence it predicts the future states of the system and then computes the optimal control inputs for the horizon. Therefore, it is also known as predictive approach. However, only the first control input from the horizon is executed on the system. The process is repeated at the next time step with new measurements and the problem is solved over a shifted prediction horizon and hence the name "receding horizon" control.

The novelty of MPC lies in the fact that it computes an optimal solution by considering model of the system and a set of constraints while predicting the future evolution of the states. On the downside, increasing the constraints and the length of prediction horizon increases the size of the problem and thus adds to the computational burden. If the model used is nonlinear or the cost function or constraints are also nonlinear, then it poses the challenge for both MPC stability theory and numerical solution [10, 18].

2.2.1. Optimization Problem Formulation

A general MPC optimization problem formulation in discrete time for a system with dynamics

$$x_{k+1} = f(x_k, u_k) \tag{2.13}$$

for a prediction horizon N_p at time t is given by:

$$\underset{x^{N_P}, u^{N_P}}{\text{minimize}} \quad m(x_{N_P}) + \sum_{k=0}^{N_P-1} l(x_k, u_k) \tag{2.14a}$$

$$\text{subject to} \quad x_{k+1} = f(x_k, u_k) \qquad k = (0, 1, \dots N_P - 1) \tag{2.14b}$$

$$x_0 = x(t) \tag{2.14c}$$

$$h(x_{k+1}, u_k) = 0 \qquad k = (0, 1, \dots N_P - 1) \tag{2.14d}$$

$$g(x_{k+1}, u_k) \leq 0 \qquad k = (0, 1, \dots N_P - 1) \tag{2.14e}$$

$$\tag{2.14f}$$

To summarize, the control input sequence u^{N_P} to be optimized and the sequence of predicted states x^{N_P} can be given as

$$u^{N_P} = [u_0, \quad u_1, \quad \dots \quad u_{N_P-1}]^T \tag{2.15a}$$

$$x^{N_P} = [x_1, \quad x_2, \quad \dots \quad x_{N_P}]^T \tag{2.15b}$$

The control objective is given as a sum of terminal cost m and stage cost l (2.14a). The optimization problem is minimized subject to a set of constraints such as the prediction model based on the system dynamics (2.14b), initial condition equal to the current measured states at time t (2.14c), equality constraints (2.14d) and inequality constraints (2.14e).

2.2.2. Solver

MPC is one of the most popular approaches in embedded optimal control which refers to a specific class of methods within embedded optimization. Since the MPC problem formulated is thought to be implemented in the experimental vehicle for testing, `acados` [53] is an ideal choice as the solver for

the nonlinear OCP formed.

The acados is an open source software package with a collection of solvers for fast embedded optimization intended for fast embedded applications. As mentioned in Verschueren et al. [53], the main feature of acados is efficient optimal control algorithms targeting embedded devices with the core written in a low-level language close to hardware i.e C. This helps avoid sacrificing computational efficiency. In addition to this, acados achieves flexibility by providing an interface with high level languages like Matlab and Python which enables users to design their algorithms with more convenience. ACADO Toolkit is another software environment which implements tools for automatic control and dynamic optimization of OCPs on embedded hardware [34]. Hence it is also popular with similar applications as presented in this thesis. The authors of acados claim it to be a successor of ACADO with the fact that acados is completely rewritten such that code generation is not needed anymore, a feature of ACADO Code Generation [33]. The disadvantages associated with ACADO Code Generation briefly includes loss of flexibility, challenging debugging process and unpredictable nature of automatic code optimization strategy. Interested reader can refer to Verschueren et al. [53] for more details. In addition to this, acados uses high performance linear algebra package BLASFEO for computationally intensive linear algebra operations. The choice of modelling language and automatic differentiation also affects flexibility, memory and runtime of embedded optimal control software. For this reason, acados uses CasADi [4]. It is based on expression graphs which involves shorter instruction sequences and a typically faster code, making it more suitable for embedded applications.

The nonlinear OCPs are discretized in acados with multiple shooting approach [8]. The resulting nonlinear programming (NLP) problem could be solved by any general purpose NLP solver like IPOPT [54] but acados uses Sequential Quadratic Programming (SQP) since it is an embeddable optimization method better suitable for embedded platform.

SQP is an iterative method to solve constrained nonlinear optimization problem. It solves a sequence of optimization subproblems or the Quadratic Programs (QP) which are the quadratic model of the objective function subject to a linearization of constraints [9]. Solving QP programs efficiently at each iteration is crucial. In acados, QP framework is based on HPIPM [41], a library defining three QP types with routines to create, manage and solve QPs.

The authors of acados have further highlighted its efficiency in comparison to other solvers like IPOPT, FalcOPT, VIATOC, ACADO Code Generation and GRAMPC in terms of relative cumulative suboptimality (RCSO) and computation times. acados performs better than other solvers and matches ACADO in RCSO and clearly beats all the other solvers in terms of computation times. Thus, acados proves its worth for the choice of the solver.

2.3. Summary

This chapter covered the background details of the important ingredients of ASC. Vehicle models namely, kinematic model and dynamic model were presented and their capabilities were discussed highlighting the operating conditions of the vehicle under which they preferred. Thereafter, a general description of MPC and corresponding optimization problem was discussed. Since the ASC will be working on a real time system, where fast computation times are a requirement, an efficient and fast solver – acados used for solving the optimization problem was presented. The elements that make acados unique when compared to other solver were highlighted.

3

Active Safety Controller

The objectives of the Active Safety Controller established earlier along with the conclusions drawn from related work, highlighted the choice of MPC as the control technique for designing ASC. Thus, MPC is the workhorse of the ASC designed in this work. This chapter discusses MPC problem formulation for achieving the primary objectives of the ASC namely – shared control and obstacle avoidance. The problem formulation follows the framework presented in the previous chapter.

The teleoperated vehicle for this work is driven at slow speeds up to 8 m/s since it is dangerous to remotely drive the vehicle at high speeds that too in complex traffic environments. This means that the vehicle is moderately driven with no intention of heavy acceleration and hence the tire slip angles and slip ratios stay considerably very small. As a result, the tires operate in linear region of the force curves and therefore a simple vehicle model that helps estimate the evolution of coordinates of CoM and the heading angle of the vehicle is sufficient for the prediction model of MPC. This requirement is easily fulfilled by non-linear kinematic bicycle model presented in 2.1.1 and hence it becomes the choice for prediction model of MPC. For obstacle avoidance criteria, a spatial constraint is formulated that considers obstacles approximated as ellipse with the vehicle bounded by circles.

By considering the non-linear kinematic bicycle model as prediction model and the lateral and longitudinal motion control requirements, the states z and input u to the system are chosen as:

$$z = [x \quad y \quad \psi \quad \delta \quad v]^T \tag{3.1}$$

$$u = [\dot{\delta} \quad a]^T \tag{3.2}$$

States and Inputs	Description
x	Position of the CoM of the vehicle along the x-axis in inertial frame
y	Position of the CoM of the vehicle along the y-axis in inertial frame
ψ	Heading angle of vehicle with respect to x-axis in inertial frame. Angle formed between longitudinal vehicle axis and the x-axis
δ	Road wheel angle (RWA). Angle formed between the direction in which front wheel is pointing and the longitudinal vehicle axis
v	Velocity vector of the CoM of the vehicle
$\dot{\delta}$	Rate of change of RWA
a	Acceleration of the CoM of the vehicle

Table 3.1: Description of the states and inputs of system

The objective of the controller is to match the human operator's commands which are referred as reference with the subscript ref. Therefore, the reference commands at time t are road wheel angle $\delta_{\mathrm{ref}}(t)$ and velocity $v_{\mathrm{ref}}(t)$. In reality, the operator gives reference steering wheel angle (SWA) input which is then converted to reference RWA δ_{ref} using a constant ratio of SWA to RWA. The length

of prediction horizon is N_P. In the following equations, N_P in the superscript means sequence of that quantity in the prediction horizon. The first term of the cost function penalizes the deviation of controller input from operator's $\delta_{\text{ref}}(t)$ with a penalty of W_δ:

$$J_\delta(\delta^N) = W_\delta \sum_{k=0}^{N-1} (\delta_{\text{ref}}(t) - \delta_k)^2 \tag{3.3}$$

The second term of the cost function penalizes deviation of controller input from operator's $v_{\text{ref}}(t)$ with a penalty of W_v:

$$J_v(v^N) = W_v \sum_{k=0}^{N-1} (v_{\text{ref}}(t) - v_k)^2 \tag{3.4}$$

To account for unavoidable constraint violation and avoid infeasibility issues, slack variables are introduced in some of the constraints, which will be explained later. These slack variables also become optimization variables and hence added as the third term of the cost function with penalty of W_s

$$J_S(S^N) = W_s \sum_{k=0}^{N-1} S_k^2 \tag{3.5}$$

Here, S is a vector of slack variables s_p for $p \in \{\delta, e\}$, of appropriate size. The overall cost function with the set of constraints making the complete optimization problem looks like:

$$\underset{\delta^N, v^N, S^N}{\text{minimize}} \quad J_\delta(\delta^N) + J_v(v^N) + J_S(S^N) \tag{3.6a}$$

$$\text{subject to} \quad z_{k+1} = f(z_k, u_k) \qquad k \in (0, N-1) \tag{3.6b}$$

$$z_0 = z(t) \tag{3.6c}$$

$$\delta_{\min} \le \delta_k \le \delta_{\max} \qquad k \in (1, N) \tag{3.6d}$$

$$v_{\min} \le v_k \le v_{\max} \qquad k \in (1, N) \tag{3.6e}$$

$$\dot{\delta}_{\min} \le \dot{\delta}_k \le \dot{\delta}_{\max} \qquad k \in (0, N-1) \tag{3.6f}$$

$$a_{\min} \le a_k \le a_{\max} \qquad k \in (0, N-1) \tag{3.6g}$$

$$\delta_{\text{dev min}} - s_\delta \le \delta_k - \delta_{\text{ref}}(t) \le \delta_{\text{dev max}} + s_\delta \qquad k \in (1, N) \tag{3.6h}$$

$$1 - s_e \le e_{k+1}^n(t) \qquad k \in (0, N-1) \text{ for } n^{th} \text{ obstacle} \tag{3.6i}$$

where,

$$e_k^n(t) = \frac{(x_{c_i,k} - x_{\text{obs}}^n(t))^2}{(\alpha_{\text{maj}})^2} + \frac{(y_{c_i,k} - y_{\text{obs}}^n(t))^2}{(\beta_{\min})^2} \qquad \text{for } i \in \{1, 2, 3, 4\} \tag{3.7}$$

$f(z_k, u_k)$ represents the kinematic bicycle model equations presented in equations (2.1) with the states and input corresponding to equation (3.1) and (3.2) respectively. It is discretized by `acados` itself using multiple shooting approach [8]. The equality constraint (3.6c) sets the current states as the initial condition. The maximum and minimum RWA $(\delta_{\min}, \delta_{\max})$ and the rate of change of RWA $(\dot{\delta}_{\min}, \dot{\delta}_{\max})$ defined by the actuator limitations are added in equations (3.6d) and (3.6f) respectively. Likewise, the bounds set on maximum and minimum vehicle velocity $(v_{\min}, v_{\max})$ and the acceleration $(a_{\min}, a_{\max})$ are added in the equations (3.6e) and (3.6g) respectively. It should be noted that no constraints on the position of the CoM i.e states x and y have been used. This is because the perception module of the experimental vehicle cannot judge the position of the lane boundaries and hence the controller cannot account for vehicle's position with respect to the lane boundaries.

In the following, the purpose of the constraints (3.6h) and (3.6i) is discussed:

Restriction on Controller Authority and Visual Feedback

The first term of the cost function $J_\delta(\delta^N)$ penalizes the deviation of RWA from the reference RWA. However, the controller can still choose to deviate from the reference as much it wants. If the controller overrides human operator by giving in larger than expected steering input, then this compromises the human operator's authority raising serious trust issues in the system. Fine tuning the penalties is not so trivial and finding the right set of penalties that helps ensure reasonable deviation of controller RWA from reference is difficult. Therefore, this cost function term acts as barrier with possibility of larger deviations.

The introduction of constraint (3.6h), defines a range within which the controller RWA can deviate from reference RWA. Therefore, it acts as a constraint with the benefit of imposing restrictions on the controller's authority to intervene and override human operator. This constraint can also help overcome trust issues in the system. Slack variable are introduced in this constraint to make it soft and help avoid infeasibility issues. There can be situations where it is unavoidable for the controller to violate this constraint for honouring obstacle avoidance criteria. In such cases, slack variables can help avoid such infeasibility issues.

In addition to this constraint, ToD offers the advantage of projecting information on the screens at the workstation. A visual feedback containing information on the region of future vehicle positions due to controller authority and future vehicle path due to complete sequence of steering wheel angle and velocity inputs by controller can be useful in knowing where will the vehicle end up. This concept is illustrated in Figure 3.1

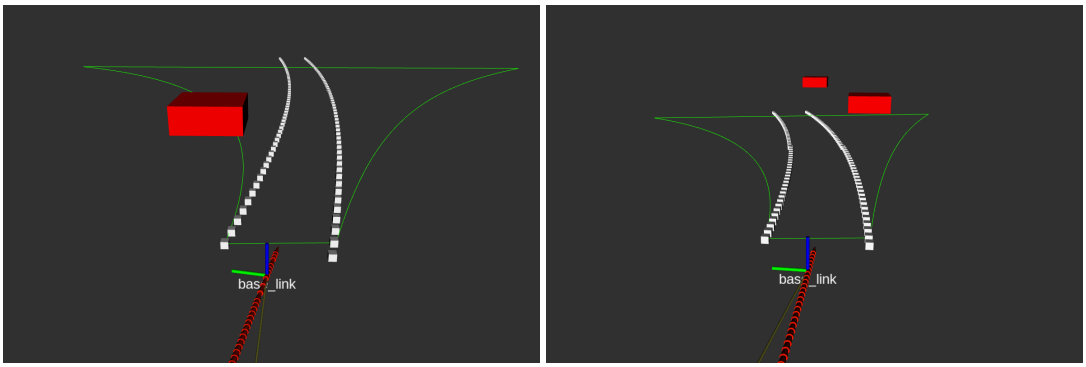


Figure 3.1: Illustration of visual feedback on the screens at workstation where the future vehicle trajectory due to complete sequence of steering and velocity inputs of the controller, as influenced by human operator's current inputs is shown by white tracks and the maximum and minimum vehicle trajectory due to the corresponding controller intervention is shown by the green controller authority cone. This illustration is created using rviz, a 3D visualization tool for ROS [23], where the obstacles are represented by the red cubes.

The future path taken by the vehicle due to complete sequence of steering wheel and velocity inputs by controller, as influenced by operator's current inputs is shown by white tracks in the visual feedback. If the controller intervenes by deviating the maximum from the reference RWA with the limit defined by (3.6h), then the corresponding vehicle path forms the right and left boundary of the authority cone. This is how the controller authority cone is formed for projecting on the screens. This visual feedback, helps the human operator to know the possible positions of the vehicle and hence avoids any surprises if the controller intervenes.

Obstacle Avoidance

Obstacle avoidance criteria is incorporated as a spatial constraint of the optimization problem. The obstacles are represented as ellipses and the vehicle is represented by four circles bounding it. The spatial constraint for n^{th} obstacle is added in equation (3.6i). The obstacle is represented as an ellipse as in equation (3.7) where the center coordinates of the n^{th} obstacle at time t in the inertial frame are given as $(x^n_{\text{obs}}(t), y^n_{\text{obs}}(t))$. The ego vehicle is represented by four circles longitudinally shifted along the longitudinal axis of the vehicle and the center coordinates of the i^{th} circle are $(x_{c_i,k}, y_{c_i,k})$ in the

inertial frame. Following the notations given in Figure 3.2

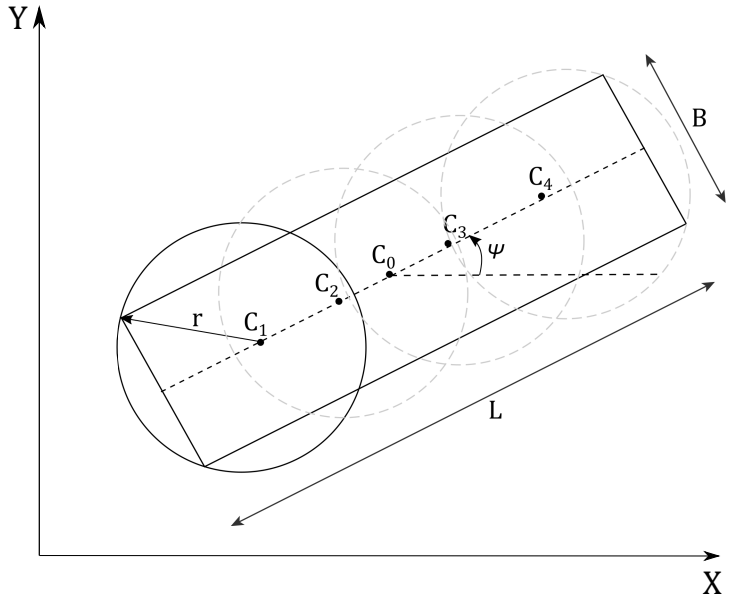


Figure 3.2: Vehicle bounded by four circles shifted along the vehicle longitudinal axis

the length of the ego vehicle is L and width is B. The four centers of the circles placed equidistant on the vehicle's longitudinal axis are named as C_i for $i \in \{1, 2, 3, 4\}$, such that the length of the vehicle is divided into five equal parts of each $L/5$. The $x - y$ coordinates of a particular circle center are referred to as x_{C_i} and y_{C_i}. The vehicle CoM is represented as C_0. The notation for the distance between two circle centers or a circle center and CoM is referred as C_{mn} for $m, n \in \{0, 1, 2, 3, 4\}$ and $m \neq n$.

The radius of the four bounding circles and the coordinates of the center of circles with respect to the ego vehicle's states (x, y) are calculated as follows:

$$C_{03} = \Delta L = \frac{L}{2} - \frac{2L}{5} = \frac{L}{10} \tag{3.8}$$

$$C_{04} = \Delta L + \frac{L}{5} = \frac{3L}{10} \tag{3.9}$$

Therefore,

$$\begin{aligned} x_{C_3} &= C_{03} \cos\psi = \frac{L}{10} \cos\psi \\ y_{C_3} &= C_{03} \sin\psi = \frac{L}{10} \sin\psi \end{aligned} \tag{3.10}$$

$$\begin{aligned} x_{C_4} &= C_{04} \cos\psi = \frac{3L}{10} \cos\psi \\ y_{C_4} &= C_{04} \sin\psi = \frac{3L}{10} \sin\psi \end{aligned} \tag{3.11}$$

Using symmetry,

$$\begin{aligned} x_{C_2} &= -x_{C_3} \\ y_{C_2} &= -y_{C_3} \end{aligned} \tag{3.12}$$

$$\begin{aligned} x_{C_1} &= -x_{C_4} \\ y_{C_1} &= -y_{C_4} \end{aligned} \tag{3.13}$$

The minimum radius of four circles that bound the ego vehicle is found using simple principles of geometry as:

$$r = \frac{B}{2\cos\left(\tan^{-1}\left(\frac{2\Delta L}{B}\right)\right)} \tag{3.14}$$

The semi-major and minor-axis of the ellipse in (3.7) are calculated as

$$\alpha = a + \lambda + r \tag{3.15a}$$

$$\beta = b + \lambda + r \tag{3.15b}$$

using the approach presented in Brito et al. [11] which ensures guaranteed collision avoidance instead of other approaches that approximate the Minkowsky sum of the ellipse with circle as an ellipse of semi-major $\alpha = a + r$ and semi-minor as $\beta = b + r$. Here (a, b) are the semi-major and minor axis of the vehicle calculated as $a = L/2$ and $b = B/2$. For reference, a brief explanation of the approach is presented in Appendix A

Summary

This chapter presented the problem formulation of MPC to meet the major criteria, namely shared control and obstacle avoidance of the ASC. It discussed reasons for the choice of kinematic bicycle model as the prediction model. The objective of the optimization problem is set to penalize the deviation of controller from the human operator's references of steering wheel angle and vehicle velocity. Therefore, the controller can handle lateral and longitudinal motion of the vehicle. Several constraints were incorporated to account for actuator limitations. To restrict controller's authority to intervene and override operator's steering inputs, a soft constraint was introduced that defines the range within which controller can deviate from reference steering input. To further enhance trust in human operator over the controller and present him an idea about the vehicle behaviour due to his and controller's inputs, the concept of visual feedback was presented. To meet the obstacle avoidance criteria, vehicle and the obstacle were represented by geometrical shapes and using the geometry principles, a spatial constraint was incorporated to ensure that the vehicle does not collide with the obstacle. Hence major contributions of this work, MPC for collision avoidance under shared control with defined restriction on controller authority and visual feedback to improve driving with enhanced trust on the system, were presented in this chapter.

4

Simulation Results

The ASC designed in the previous chapter is tested in closed loop simulations using IPG CarMaker [27]. The system architecture given in the Figure 4.1 shows the major components involved in closed loop simulation. For ToD, the setup consists of two main blocks, that are, (i) the workstation, where the human operator would give in the reference inputs for longitudinal and lateral motion control using steering wheel and gas and brake pedals (ii) the teleoperated vehicle, which is equipped with MPC based ASC. The operator increases/decreases the desired vehicle velocity using gas/brake pedals. For simulations, the reference SWA (or RWA to be precise) inputs are generated using a Feedback Linearized Path Tracking Controller (FBLC). Therefore, FBLC replicates the human operator behaviour. Along with the reference RWA, reference velocity inputs are also generated using preset velocity values as a function of time. Both the references generated in the reference block are transmitted to vehicle via a commercial 4G LTE network and hence it adds an actuator latency in signals transmitted. Based on these reference signals, MPC computes the optimal RWA and velocity control inputs to ensure collision avoidance.

Since the actual experimental vehicle and also the CarMaker plant cannot take velocity inputs directly, an Adaptive Cruise Control (ACC) is used that computes gas and brake pedal values to achieve the desired velocity computed by MPC. ACC acts as a medium to translate desired velocity into gas and brake pedal inputs for the vehicle. The RWA control input from MPC is translated to corresponding SWA using the steering ratio to convert from RWA to SWA. The inputs are then executed on the vehicle or the plant in the simulations. Meanwhile, the camera and LIDAR installed on the vehicle captures the environment of the vehicle and this information is sent as state feedback to the MPC and over the network back to the workstation to be displayed on the screens. This transmission again happens using the same network and hence adds glass latency in the signals transmitted. As mentioned earlier, these latencies account for signal path from source to destination and not merely the extra time added while traveling over the network. The average round trip latency for FTM setup is 200 ms. For simulations the actuator latency and glass latency are taken as 80 ms and 120 ms respectively. Latency delays the signals during transmission and hence the components receive old information. It should be noted that MPC receives the most recent state feedback but delayed reference signals from the reference block. This is how ToD differs from a regular shared control setup.

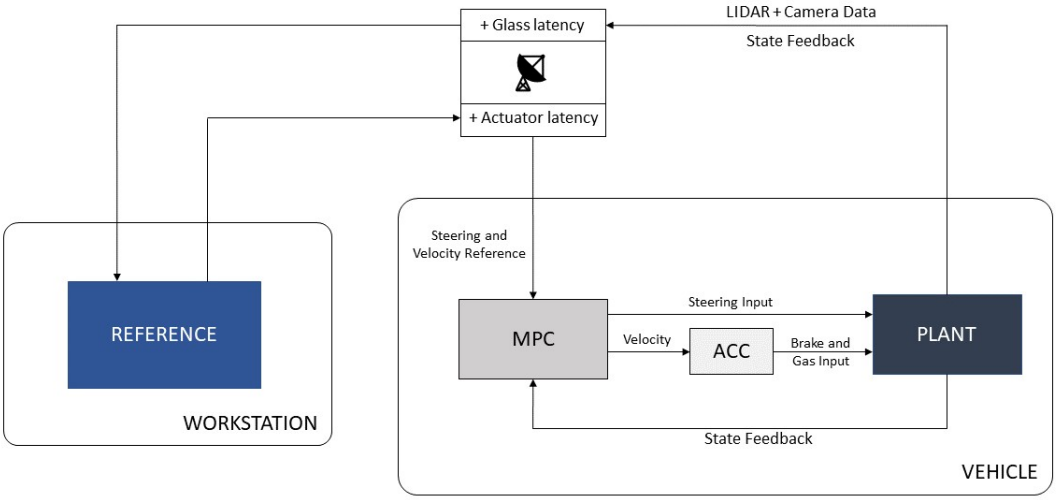


Figure 4.1: System architecture for Teleoperated Driving.

As mentioned in the above system architecture description, FBLC and ACC are also used in the complete control loop. Hence, the design and tuning of FBLC and ACC used in this work are also discussed in this chapter. To test the ASC for various scenarios where ToD is expected to be deployed under varied state of human operator, reference scenarios are designed and presented in this chapter. The ASC for ToD is compared with other baseline approach implemented in regular shared control setup. The simulations also check the effect of latency on driving performance as compared to no latency as in the case of a regular shared control setup.

The tuning parameters for all the system components like FBLC, MPC and ACC are summarized in B.1. The choices for tuning parameters are explained in the corresponding sections. It should be noted that the same set of tuning values are used for all the simulations unless otherwise specified.

All simulations were performed on MATLAB/Simulink® in parallel with IPG CarMaker on a Windows machine equipped with Intel Core i7-4600 2.10 GHz 2 core CPU with 8 GB of RAM. The average MPC solution time was around 40 to 50 ms and hence it can be expected to perform even faster when run on the powerful CarPC.

4.1. Scenarios for Simulations

Different scenarios are constructed to showcase the features of the designed ASC. These scenarios are then used to compare the performance of baseline approach controller and the ASC. The scenarios are classified according to the behaviour of the human operator. The reference trajectory thought to be followed by the human operator is formulated using simple or a sigmoid curve and a corresponding reference heading angle from the trajectory is computed. The scenarios are classified as follows:

4.1.1. Inattentive Human Operator

Although the human operator is required to be vigilant all the time, these scenarios consider cases where the operator is inattentive and his vehicle ends up very close to the obstacle which is very much blocking the vehicle's path. The scenarios considered in this case are:

Scenario 1a: Obstacle blocking part of the road

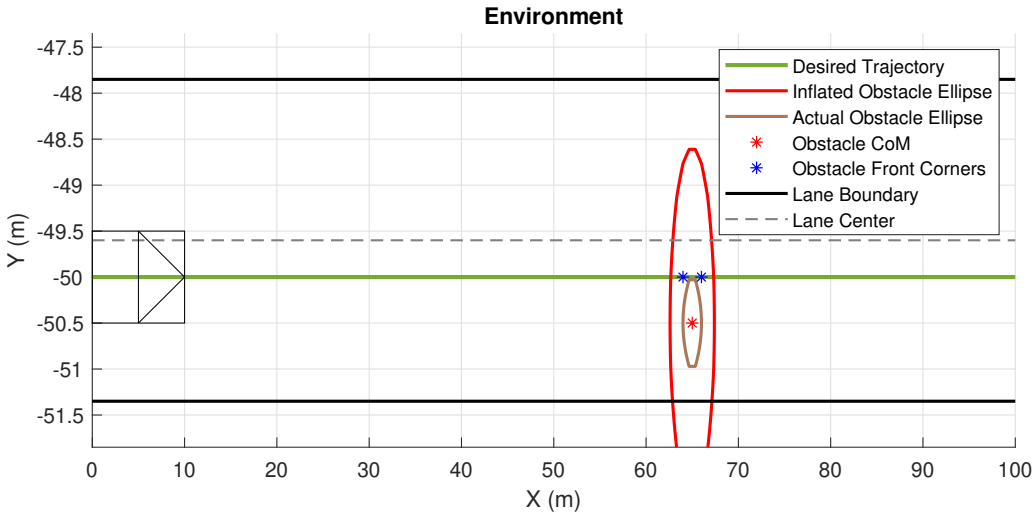


Figure 4.2: Desired trajectory for the CoM of vehicle when an obstacle partly blocks the road. The ego vehicle illustrated is not to scale.

In this scenario, the ego vehicle starts from $(0, -50)$ and continues along the positive x-axis. The desired trajectory is to be followed by the CoM of ego vehicle. The obstacle is placed with its longitudinal axis parallel to y-axis, with the front edge coinciding with desired trajectory of the ego vehicle. The actual obstacle is represented by a smaller ellipse and the inflated obstacle ellipse accounts for the circles representing ego vehicle.

Scenario 1b: Obstacle blocking full road

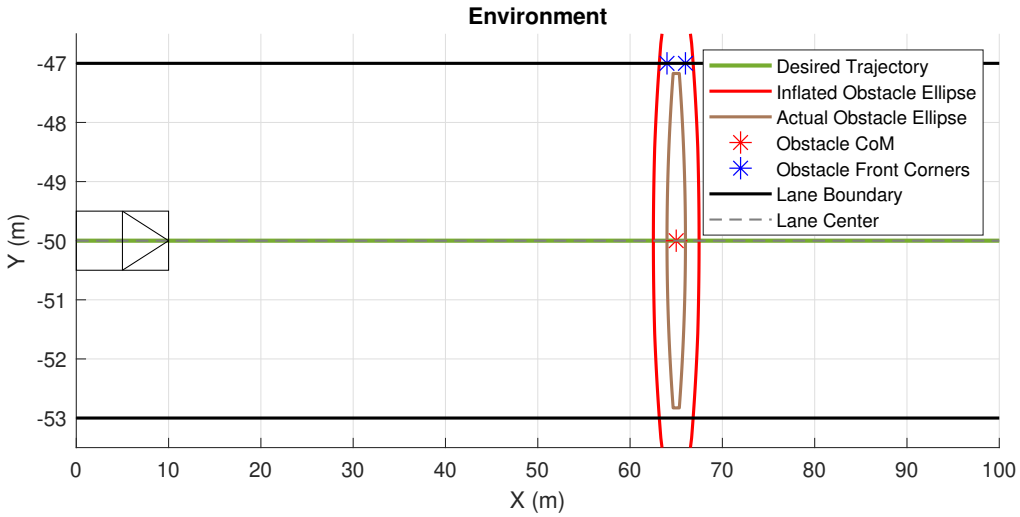


Figure 4.3: Desired trajectory for the CoM of vehicle when an obstacle completely blocks the road. The ego vehicle illustrated is not to scale.

In this scenario, obstacle blocks the entire road. This scenario creates a situation similar to ego vehicle running into a wall. The rest of the conditions are same as in Scenario 1a.

Scenario 1c: Two pedestrians crossing the road

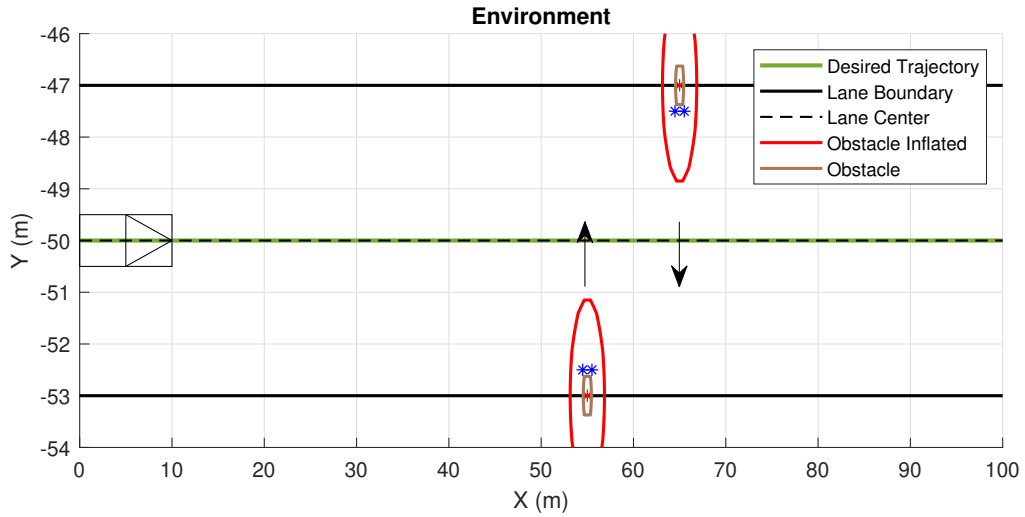


Figure 4.4: Desired trajectory for the CoM of vehicle while two pedestrians cross the road. The ego vehicle illustrated is not to scale.

In this scenario, two pedestrians starting from opposite sides of the road cross the road while the vehicle is travelling along the positive x-axis towards them. Therefore this scenario accounts for dynamic obstacles. The vehicle starts from $(0, -50)$. The human operator simply keeps on driving the vehicle and does not attempt to stop the vehicle, owing to the fact that he is inattentive. Same concept is used as before for representing the pedestrians as ellipses.

4.1.2. Inaccurate Human Operator

This case attempts to recreate a dense environment with obstacles closely placed making it difficult for the human operator to make judgement of location and size of the obstacles from the workstation.

Scenario 2: Obstacles closely spread along the vehicle path

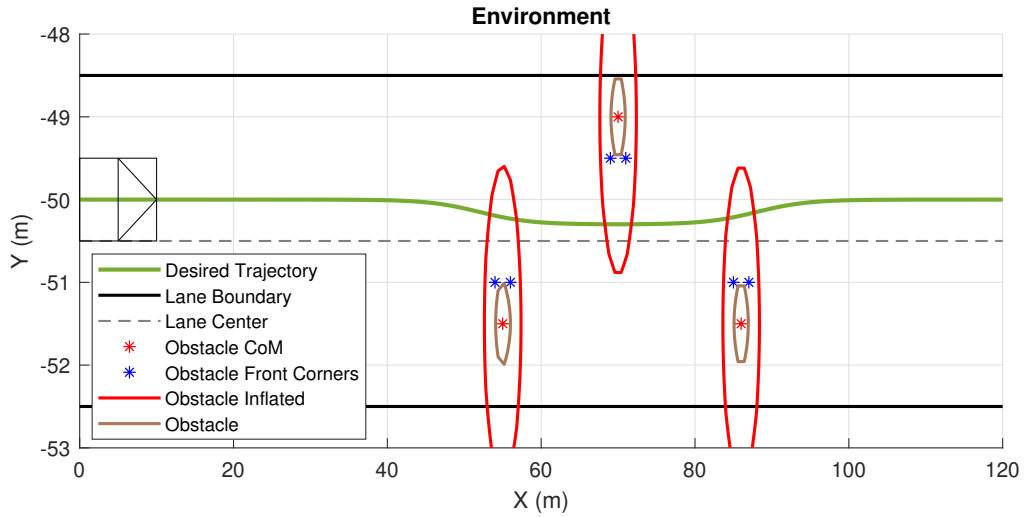


Figure 4.5: Desired trajectory for the CoM of vehicle while three obstacles are closely placed along the x-axis with limited space between them along the y-axis. The ego vehicle illustrated is not to scale.

In this scenario, three obstacles are placed with its longitudinal axis parallel to y-axis and equidistant from each other in the lateral direction. The space between them is enough for the vehicle to physically pass through them. Rest of the conditions are same as in Scenario 1a.

4.2. Simulation Setup

Various additional components had to be designed to test the MPC performance in simulations. The controller was formulated in MATLAB and the simulations were done using IPG CarMaker using the Simulink interface. CarMaker presents a complex vehicle model that depicts the actual vehicle behaviour quite accurately. Thus, it was used as the plant. As mentioned before, FBLC was designed to replicate the human operator to generate reference RWA for simulations. Moreover, ACC had to be designed to translate the desired velocity computed by MPC into corresponding brake and gas pedal values for the CarMaker. These elements are discussed in this section along with their tuning to achieve the desired performance.

4.2.1. Feedback Linearized Path Tracking Controller

For simulations, the human operator was replaced with FBLC to generate reference road wheel angle to follow a reference trajectory [13, 40]. Based on error model that considers heading error and lateral error with respect to the current vehicle states and the reference states, feedback linearized control law is formed [45]. It ensures generation of steering actions that helps track the reference trajectory based on current vehicle states and thus can correct any local deviations in path tracking. The reference trajectory is predesigned and it is a trajectory which the human operator is thought to follow. The control law is:

$$\delta_{\mathrm{FBL}}(t) = \arctan\left(\frac{-\gamma_1 e_L(t) - \gamma_2 v_{\mathrm{vref}} \sin\left(e_H(t)\right)}{v_{\mathrm{vref}}^2 \cos(e_H(t))}\right) \tag{4.1}$$

where, lateral error and heading error are e_L and e_H respectively and v_{ref} is the reference velocity as chosen by the operator. The correction for heading error and lateral error can be prioritized using the gains γ_1 and γ_2. The errors, e_L and e_H are computed with respect to the reference tracking point along the trajectory which is again a function of lookahead distance. It is the distance along the path at which tracking point is selected. A large lookahead distance results in stable transient behaviour while increasing the chances of corner cutting. Whereas a shorter lookahead distance results in accurate path tracking. To include the effect of visual feedback to the human operator, another feedback term is added. This term penalizes the deviation between the operator SWA (RWA) or the calculated δ_{FBL} and the current SWA (RWA), with the proportional gain γ_3. Thus, the overall simulated human operator's RWA input becomes:

$$\delta_{\mathrm{ref}}(t) = \delta_{\mathrm{FBL}}(t) + \gamma_3(\delta(t_0) - \delta_{\mathrm{FBL}}(t)) \tag{4.2}$$

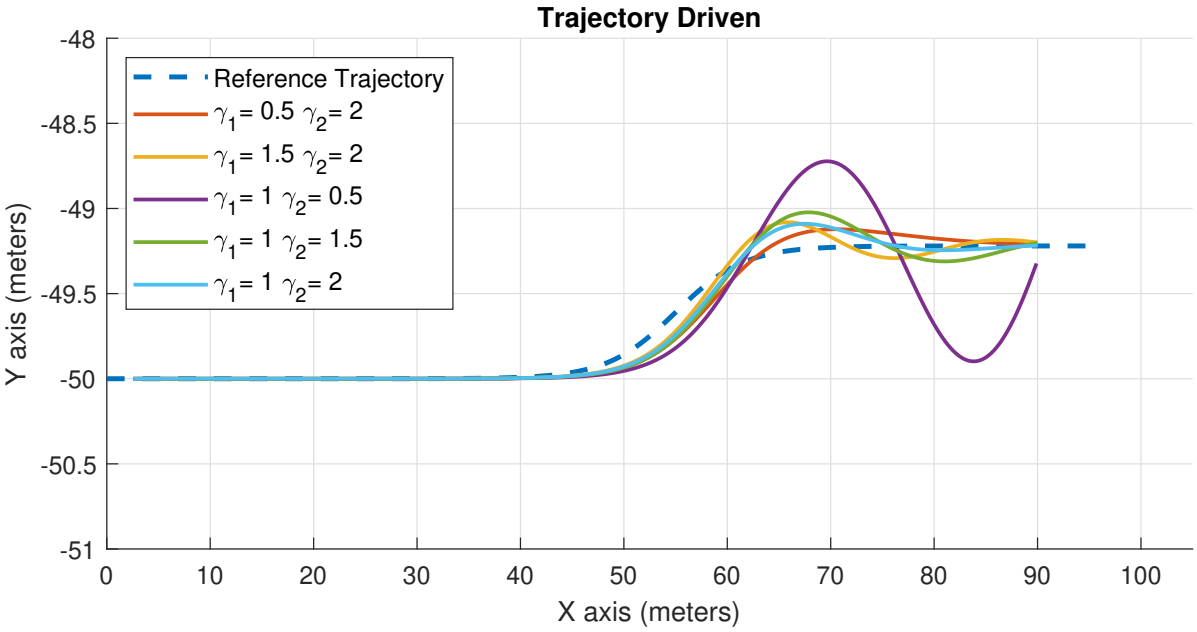


Figure 4.6: FBLC: Trajectory driven by vehicle for different combinations of γ_1 and γ_2 gains of FBLC for $\gamma_3 = 0.25$ and lookahead distance of 1 m

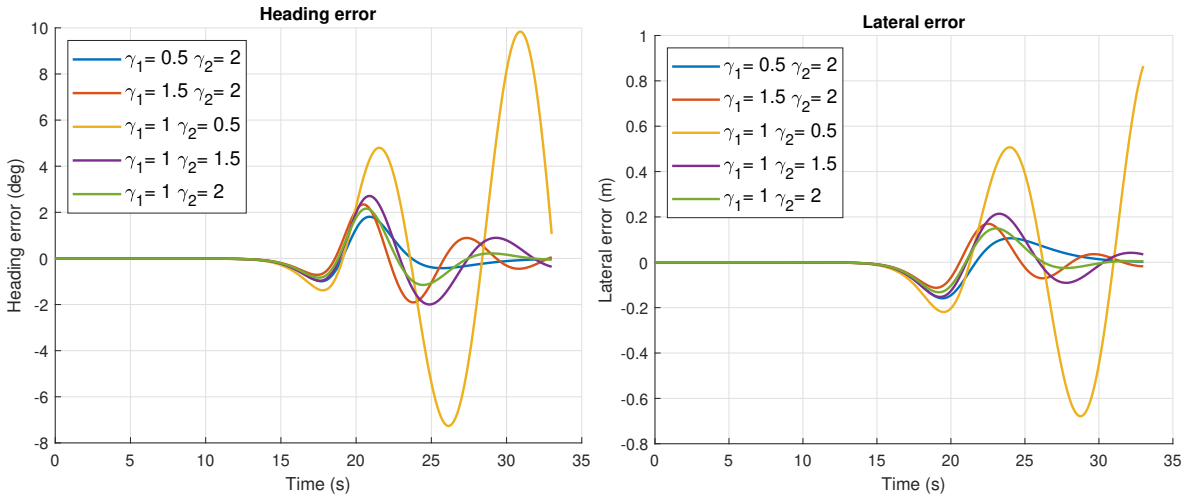


Figure 4.7: FBLC: Heading error e_H and Lateral error e_L for different combinations of γ_1 and γ_1 gains of FBLC for $\gamma_3 = 0.25$ and lookahead distance of 1 m

FBLC was tuned in closed loop by considering latency in the simulations. A constant lookahead distance of 1 m and γ_3 gain of 0.25 was chosen to iterate through different combinations of γ_1 and γ_2 gains. Figure 4.6 shows path tracking results for different tuning combinations. The reference path tries to replicate a simple lateral shift in straight line driving. This plot shows that the relation $\gamma_1 < \gamma_2$ should hold to avoid oscillations in the driven trajectory. The above result along with the heading and lateral error plots in the Figure 4.7 shows the favourable combination of the gains. A large difference in the gains satisfying the condition mentioned before, results in a smooth but a very moderate tracking. Hence penalties of $\gamma_1 = 1$ and $\gamma_2 = 2$ are chosen that results in a little aggressive but favourable combination for tight maneuvers. This tuning combination is used for all the simulations. It should be noted that, tuning determines aggressiveness of the FBLC in path tracking.

4.2.2. IPG CarMaker

CarMaker provides a complete package for virtual testing of systems on a whole vehicle in real-world test scenarios. It uses complex modelling of all the vehicle components ranging from steering and suspension systems, powertrain, brake, vehicle body and tire to name a few, with the flexibility to parameterize for customization of the simulation as per the need. Apart from the complex vehicle model, CarMaker provides driver model as well as elements of the environment such as road and traffic. This facilitates testing of the designed system in real world like scenario at the simulation level. Further information about vehicle model and environment can be found in the software reference manual [28]. This software presents additional flexibility by providing the scope of using it with other simulation softwares like MATLAB and Simulink.

For CarMaker simulations in this work, VOLVO XC90 T6 AWD was used which uses a RealTime Tire model. It is the realtime version of the IPGTire model which is an empirical tire model and as such based on measurements [29]. CarMaker also provides the scope of selecting driver parameters and setting the type of driver such as defensive, normal or aggressive via IPGDriver. For simulations, the type of driver chosen was normal and the default parameters were untouched. However, the choice of driver type should not matter since driving actions such as steering wheel input and acceleration/braking input are generated (using reasonable practical constraints) according to the reference trajectory, using FBLC and ACC respectively. A simple scenario is used with a tarmac road of friction coefficient 1. It is assumed that all the vehicle and sub-systems signals can be measured directly noise free as provided by CarMaker. The idea behind using CarMaker was to use a realistic plant for testing the designed controller and hence not much emphasis was given in creating the scenario. As mentioned earlier, CarMaker needs ACC to translate desired velocity into corresponding brake and gas pedal value to achieve the desired speed. The design and tuning of ACC is presented below.

4.2.2.1. Adaptive Cruise Control

The vehicle speed in CarMaker can be adjusted by changing the brake and gas pedal values which are in the range of 0 to 1. It cannot take a velocity value and achieve it automatically. Changing pedal values is an appropriate approach to achieve a desired speed as this is something similar to what would actually happen in a vehicle. The Cruise Controller on FTM setup takes in desired velocity to adjust the vehicle speed accordingly. Therefore, it is a wise decision to design an Adaptive Cruise Control (ACC) to achieve the same purpose for CarMaker simulations. The desired velocity is taken from the first predicted state of MPC. The ACC is formulated as a basic Proportional-Integral (PI) Controller as discussed here:

$$a_{\text{des}}(t) = \frac{v_{\text{MPC}}(t) - v_{\text{veh}}(t)}{\Delta a} \tag{4.3}$$

here, at time t, v_{MPC} is the desired velocity computed by MPC that is to be achieved, v_{veh} is the current vehicle velocity given by CarMaker and Δa is a tuning parameter that determines the magnitude of desired acceleration a_{des}. The error in the acceleration is calculated as:

$$e_a(t) = a_{\text{des}}(t) - a_{\text{veh}}(t) \tag{4.4}$$

where a_{veh} is the vehicle acceleration at time t. Based on this error, PI controller is formulated to find the control input u_a,

$$u_a(t) = K_P e_a(t) + K_I \sum_{i=1}^{n_t} e_a(t) \Delta T_s \tag{4.5}$$

K_P and K_I are the proportional and integral gains respectively and ΔT_s is the sampling time between two sampling instances and n_t is the number of sampling instances until time t. ΔT_s is equal to the sample time between two time step in the simulation. Since the pedal values are in the range of 0 to 1, the value of u_a is bounded between -1 and $+1$ using saturation block in Simulink. The gas G and brake B pedal values are computed from u_a based on the following:

$$\begin{aligned} B &= -u_a \qquad \textit{if } -1 < u_a < 0 \\ G &= u_a \qquad \textit{if } \quad 0 < u_a < 1 \end{aligned} \tag{4.6}$$

Tuning of ACC is important to achieve the desired speed in acceptable time satisfying the acceleration limits of the vehicle. Moreover, it was observed that ACC tuning heavily affects MPC performance. ACC was tuned for open loop scenario by iterating through the tuning parameters of Δa, K_P and K_I. The step response characteristics and the corresponding velocity plots as given in B.1, were then compared to find the best combination of tuning parameters for ACC. These results are compiled in Table 4.1.

		Δa for $K_P = 0.01$ $K_I = 0.001$			K_P for $\Delta a = 0.3$ $K_I = 0.001$			K_I for $\Delta a = 0.3$ $K_P = 0.01$		
Sr. No.	**Parameter**	**0.2**	**0.3**	**0.5**	**0.001**	**0.01**	**0.1**	**0.001**	**0.01**	**0.1**
1	Rise Time	0.8540	0.8800	1.0610	0.8730	0.8800	0.9360	0.8800	0.8990	0.9120
2	Settling Time	6.5950	6.3250	6.8660	6.3190	6.3250	6.3820	6.3250	6.5270	6.5210
3	Settling Min	4.5290	4.6230	4.5210	4.6310	4.6230	4.5730	4.6230	4.5130	4.5230
4	Settling Max	5.1520	5.0290	5.0000	5.0260	5.0290	5.0570	5.0290	5.0000	5.0550
5	Overshoot	3.0390	0.5800	0	0.5240	0.5800	1.1440	0.5800	0	1.1010
6	Undershoot	0	0	0	0	0	0	0	0	0
7	Peak	5.1520	5.0290	5.0000	5.0260	5.0290	5.0570	5.0290	5.0000	5.0550
8	PeakTime	6.4500	6.6500	14.9000	6.5500	6.6500	6.8000	6.6500	10.3000	15.0000

Table 4.1: Step response characteristics for different tuning items for ACC

From the above results, it can be seen that step response characteristics for $\Delta a = 0.3$ and $\Delta a = 0.2$ (for the given combination of K_P and K_I) are almost similar and better in comparison to characteristics for $\Delta a = 0.5$, however the overshoot for $\Delta a = 0.3$ is much lower than that for $\Delta a = 0.2$ and hence it becomes a preferred choice. The characteristics for K_P equals to 0.01 and 0.001 are almost equal for $\Delta a = 0.3$ and $K_I = 0.001$. $K_I = 0.001$ with $\Delta a = 0.3$ and $K_P = 0.01$ as constant, has the best step response characteristics as compared to other K_I values. Increasing K_I also increases oscillations in the velocity curve.

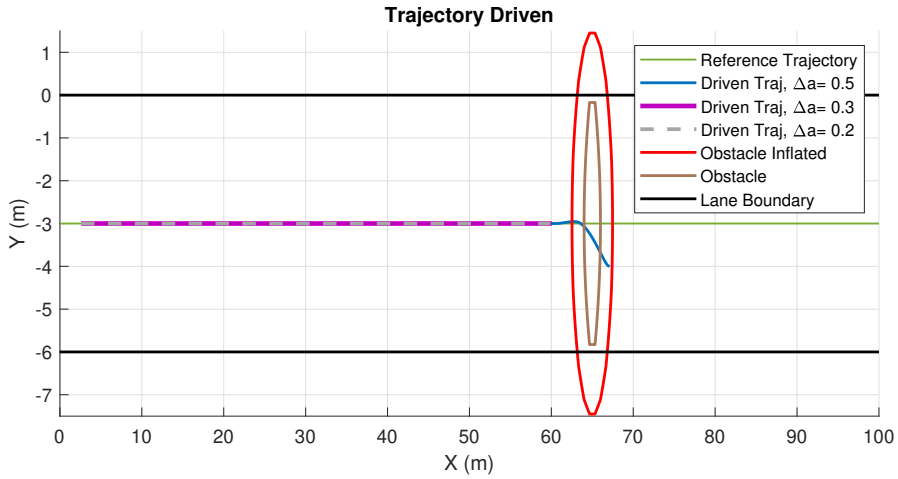


Figure 4.8: ACC: Trajectory driven by CoM of vehicle for different values of Δa where obstacle blocks the road completely and the vehicle is expected to stop.

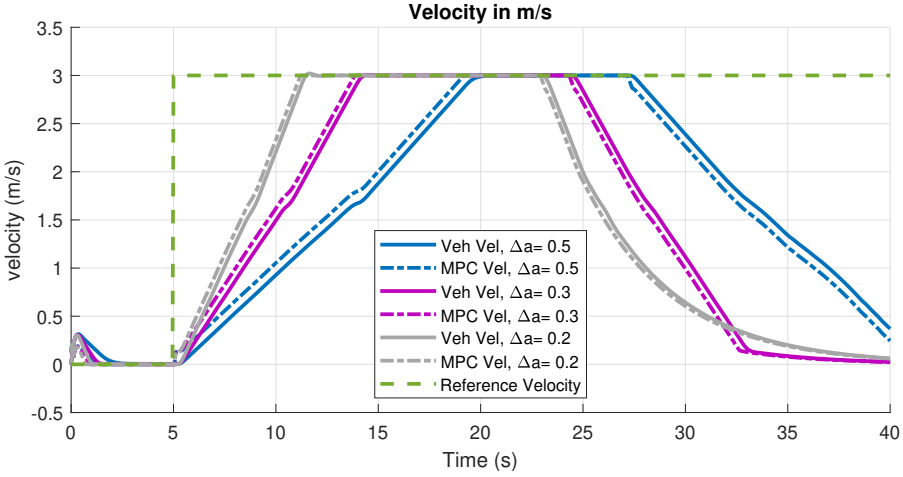


Figure 4.9: ACC: Velocity plot for different values of Δa where obstacle blocks the road completely. Vehicle velocity is represented by solid line and MPC velocity are represented by dashed line.

It was also observed that a good ACC tuning in open loop could have drastically different results for a closed loop case, especially when checking the stopping behaviour of MPC in Scenario 1b described in 4.1.1.2. Hence, tuned ACC was then checked for Scenario 1b in closed loop where latency in the simulation was considered. The trajectory driven for different Δa values is shown in Figure 4.8. Δa determines the aggressiveness of the ACC in accelerating to achieve the desired speed as well as decelerating in time to stop before a wall like obstacle (i.e Scenario 1b). It can be seen in the Figure 4.9 how fast the desired velocity is achieved for different Δa values and similarly how well in time the controller is able to stop the vehicle before the obstacle. Lousy ACC with higher Δa value cannot stop the vehicle in time and hence collides with the obstacle. In conclusion, the final tuning values of ACC are chosen as given in the Table 4.2:

Δa	K_P	K_I
0.3	0.01	0.001

Table 4.2: Final values of the tuning parameters for ACC

4.3. Constraint Bounds

The constraints of MPC problem listed in the equations (3.6d) to (3.6h) need to have their bounds fixed according to the actuator limitations. In the following, values of the bounds for various constraints are

discussed here

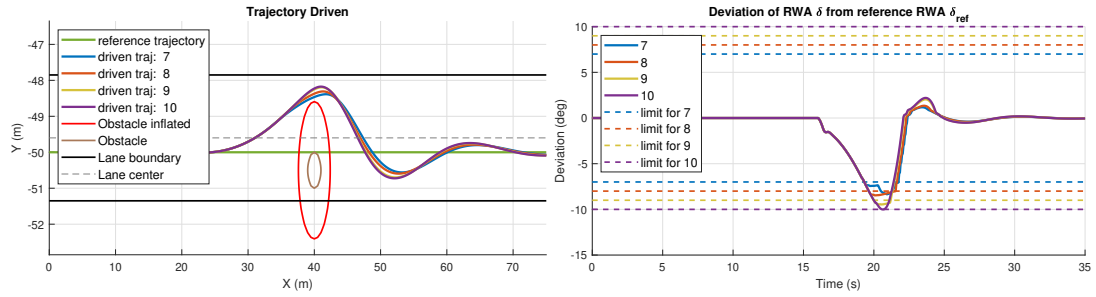
1. **Velocity** v
   The vehicle is expected to be driven at slow speeds in complex traffic environment. Therefore, the maximum speed is restricted to $v_{\max} = 8\,\text{m/s}$

2. **Acceleration** a
   VOLVO XC90 can accelerate from 0 to 100 km/h in 6.5 s. This indicates the maximum acceleration as 4.27 m/s^2. However, the maximum and minimum acceleration for the controller is bounded as $-2.5 \leq \dot{v}_{\max} \leq 2.5$ for a safe operation. It would be dangerous if the controller accelerates vehicle with its maximum potential against the wish of human operator.

3. **Road wheel angle** δ
   The number of steering wheel turns from end to end for VOLVO XC90 are three [17]. Therefore, the maximum steering wheel angle on one side becomes $\pm 540°$. Since the steering ratio (angel/angle) is 16.8 [17], the road wheel angle is bounded as $-32.14° \leq \delta \leq 32.14°$.

4. **Road wheel angle rate** $\dot{\delta}$
   The steering actuator installed on the FTM experimental vehicle can achieve maximum RWA rate of 30 °/s. This means that it can steer from neutral position to the maximum position within almost 1 s. To make the steering actions a little conservative, the maximum steering angle that can be achieved in 1 s is limited to 340° which translates to the bounds on RWA rate as $-20.23\,°/\text{s} \leq \dot{\delta} \leq 20.23\,°/\text{s}$.

5. **Deviation of road wheel angle RWA from reference RWA** $\delta - \delta_{\mathbf{ref}}$
   This constraint limits the controller authority or the degree of controller intervention to override the human operator's steering wheel inputs as discussed in chapter 3. A very high limit means controller has higher freedom over the operator's steering inputs and a very low limit means the controller is too conservative. A high limit also makes this constraint redundant. To find the appropriate value of the limits, iterations were done for different values of bounds for the Scenario 1a described in 4.1.1.1 with vehicle travelling at 3 m/s. The FBLC was tuned to make it little more aggressive since it then tries to follow the reference trajectory religiously and when the vehicle finally leaves the trajectory, it causes larger deviation from reference RWA. These conditions ensure the minimum length of predicted path and prompts the controller to make a heavy intervention to avoid collision. Figure 4.10a shows that for all the different bounds vehicle avoided collision, but the Figure 4.10b suggests that the controller violated narrower bounds to avoid collision. Therefore, bounds on this constraint were set as $-10 \leq \delta - \delta_{\text{ref}} \leq 10$. A relaxation of $10°$ also means that the controller has around 30% freedom to deviate from the reference RWA.



(a) Trajectory Driven

(b) Deviation of RWA from reference RWA

Figure 4.10: Plots for the trajectory driven and deviation in RWA δ from the reference RWA δ_{ref} for an aggressive FBLC with gains $\gamma_1 = 1.25$, $\gamma_2 = 2$, $\gamma_3 = 0.25$ and lookahead distance of 1 m

Constraint		Limits	Units
Velocity	v	± 8	m/s
Acceleration	$\dot{v}$	± 2.5	m/s
RWA	δ	± 32.14	degree
RWA rate	$\dot{\delta}$	± 20.23	degree/s
RWA deviation from reference RWA	$\delta - \delta_{\text{ref}}$	± 10	degree

Table 4.3: Summary of the physical constraints of the optimization problem and the value of the limits

4.4. Baseline vs ASC

In this section, the features of the MPC based ASC designed are showcased. The baseline approach has only lateral control capability with no restriction on the degree of controller intervention to override human operator's steering wheel angle inputs. As mentioned before, ASC can control lateral and longitudinal motion of the vehicle with a constraint or restriction on the controller's authority. These capabilities can be easily seen when the vehicle encounters a dead end with the driver deliberately trying to drive into the dead end. In this case, baseline controller would have problems in finding a collision free path if there is no free space on the sides to go around the obstacle. Since ASC is equipped with longitudinal control it is expected to stop in such a situation. This is illustrated in Figures 4.11 and 4.12

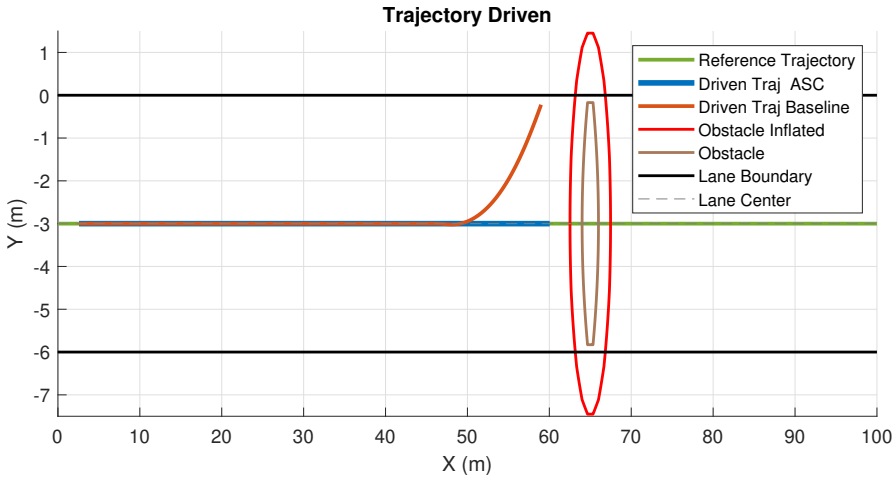
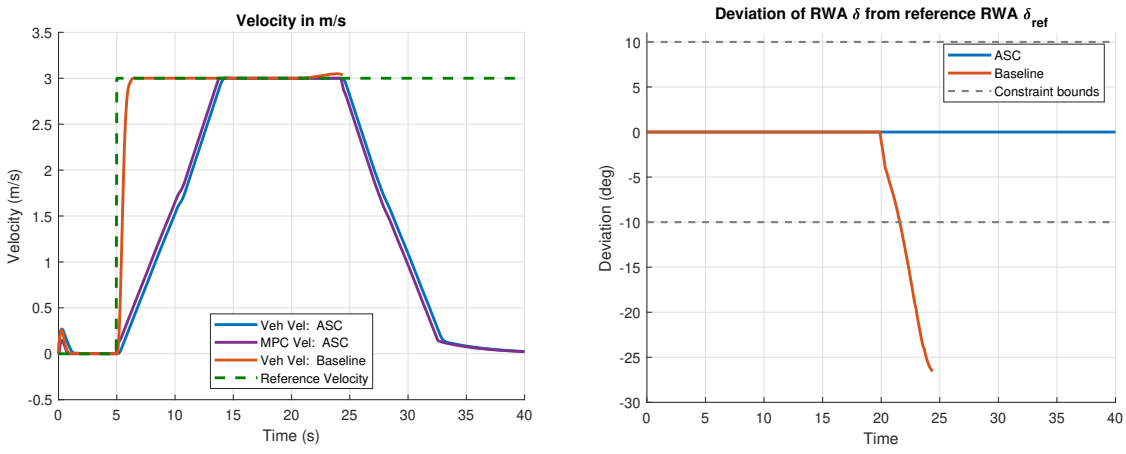


Figure 4.11: Baseline vs ASC: Trajectory driven by controller in baseline approach and ASC when encountering an obstacle blocking the entire path with no free space beyond the lane boundaries. The Baseline controller tries to drive the vehicle out of the lane and hence CarMaker aborts the simulation.



(a) Velocity

(b) Deviation of RWA δ from reference RWA δ_{ref}

Figure 4.12: Baseline vs ASC: Velocity and Deviation of RWA δ from reference RWA δ_{ref} plots for the Baseline Controller and ASC

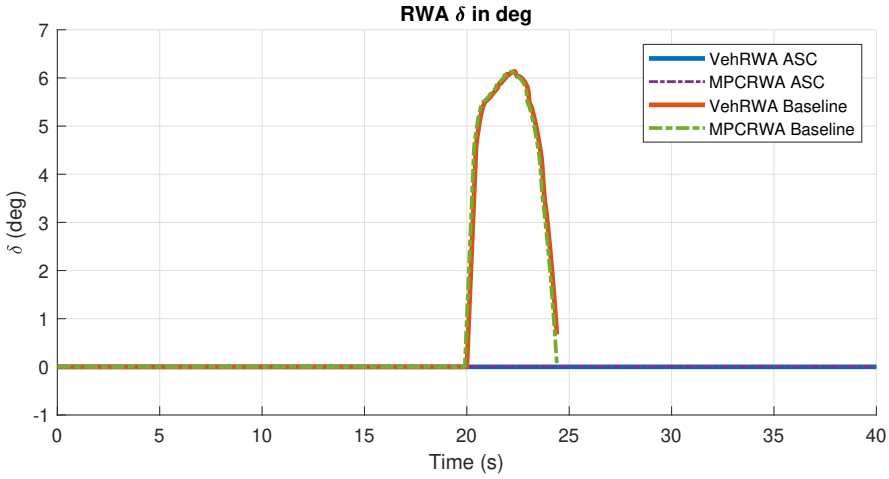


Figure 4.13: Baseline vs ASC: RWA δ

It can be seen that the baseline controller cannot make the vehicle stop and it keeps on travelling with the constant velocity. Baseline controller tries to drive the vehicle around the obstacle as evident from the RWA plot in Figure 4.13 and hence the vehicle attempts to leave the lane. The end of the trajectory represents the point where the vehicle crosses the lane and CarMaker aborts the simulation. On the contrary, vehicle equipped with ASC keeps on travelling along the positive x-axis and stops in front of the obstacle as can be seen from the velocity plot in Figure 4.12a. Figure 4.12b shows that the baseline controller does heavy RWA intervention close to $-25°$ or $-425°$ in steering wheel angle (SWA) opposed to what the human operator is demanding. When comparing this to maximum range of RWA $[-32.14° \quad 32.14°]$, this controller intervention is too high with the potential to raise trust issues in the human operator over the system. This is where constraint (3.6h), i.e. deviation of RWA from reference RWA helps.

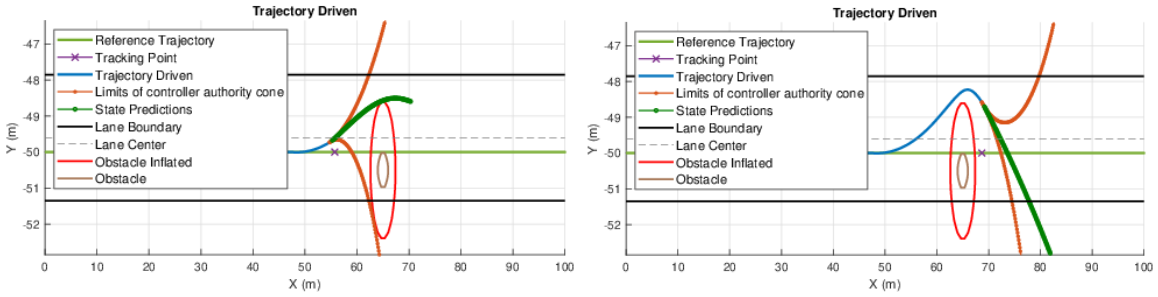


Figure 4.14: Snapshots of animated plot showing the trajectory driven by the vehicle chasing the tracking point with state predictions of MPC and the limits of the controller authority cone.

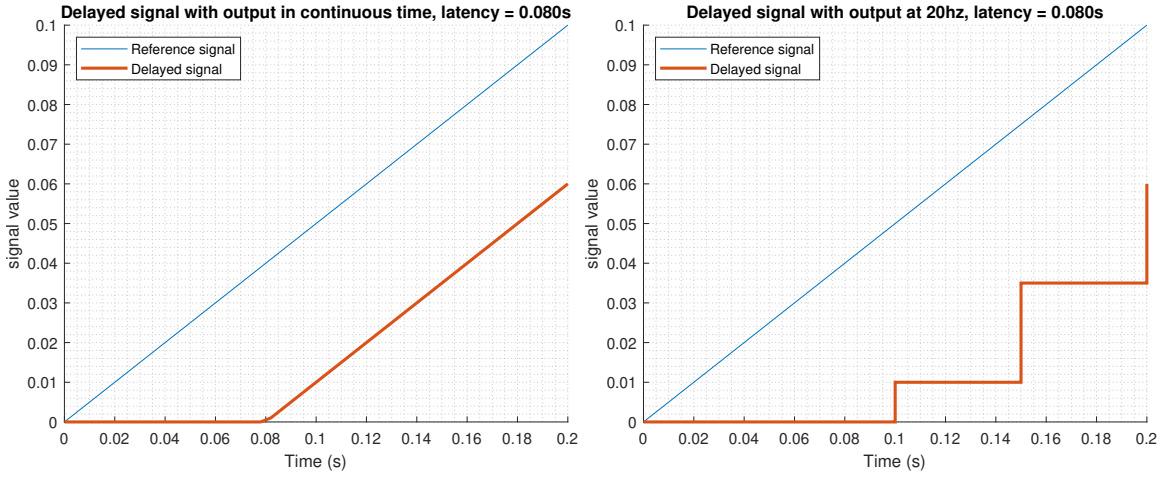
The inclusion of the constraint (3.6h) makes the controller conservative as well as reliable in terms of knowing how much the vehicle could deviate away from human operator's wish. This constraint explicitly defines the range within which the controller can deviate from the reference RWA δ_{ref}. The Figure 4.14 shows that the state predictions of the MPC always lie within the region defined by the trajectories that would have been taken by vehicle due to maximum and minimum controller intervention for RWA as defined by the constraint (3.6h). The visual feedback containing information on this controller authority as described in chapter 3 further infuses confidence on the controller behaviour.

4.5. Latency vs No Latency

The performance of ASC is expected to get affected due to latency in the network. As described earlier in 1.1.2, the latency of FTM is equal to 200.35 ms on an average and hence it is thought to be too low to affect the driving performance. To verify this proposition, simulations were done to compare the results for no latency in the system with latency in the system for environments with static and dynamic

obstacles.

Before discussing results for ASC performance due to network latency, it is important to get an intuitive idea on how latency affects signals. The Figure 4.15a illustrates how a reference signal gets delayed due to latency in the network. Since the delayed signal is fed to the MPC running at 20 Hz, MPC updates its input value with the latest reference signal value available and hence the reference values received meanwhile are discarded. This is clearly illustrated in Figure 4.15b.



(a) Delayed signal with output in continuous time

(b) Delayed signal with output at 20 Hz

Figure 4.15: Delayed Signal due to network latency equal to 80 ms with output at different frequency.

The same concept is used to delay the signals in the simulations to introduce the effect of network latency. The performance of ASC was checked for scenarios with the static and dynamic obstacles. For the environment with static obstacles, the human operator was assumed to be inaccurate in judgement of obstacle positions and hence Scenario 2 described in 4.1.2 was used. For environment with dynamic obstacles, Scenario 1c in 4.1.1.3 was used. The actuator latency equal to 80 ms and glass latency equal to 120 ms was taken for the simulations.

4.5.1. Static Environment

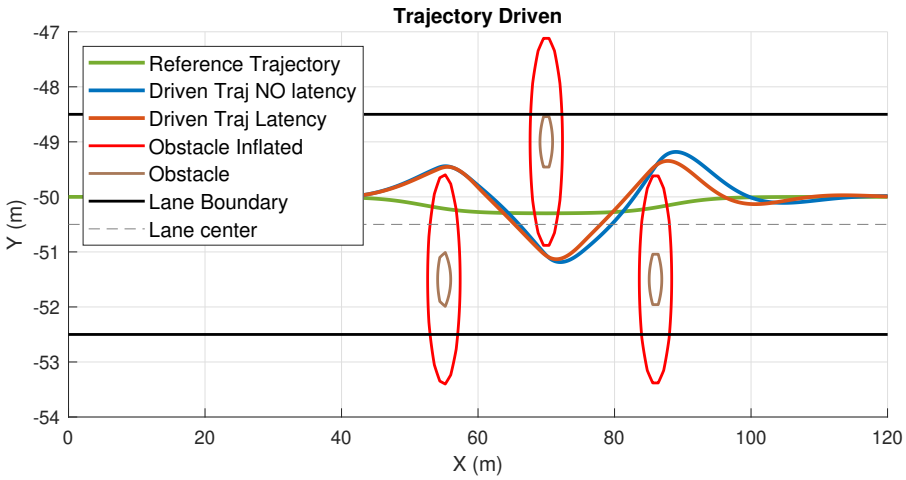
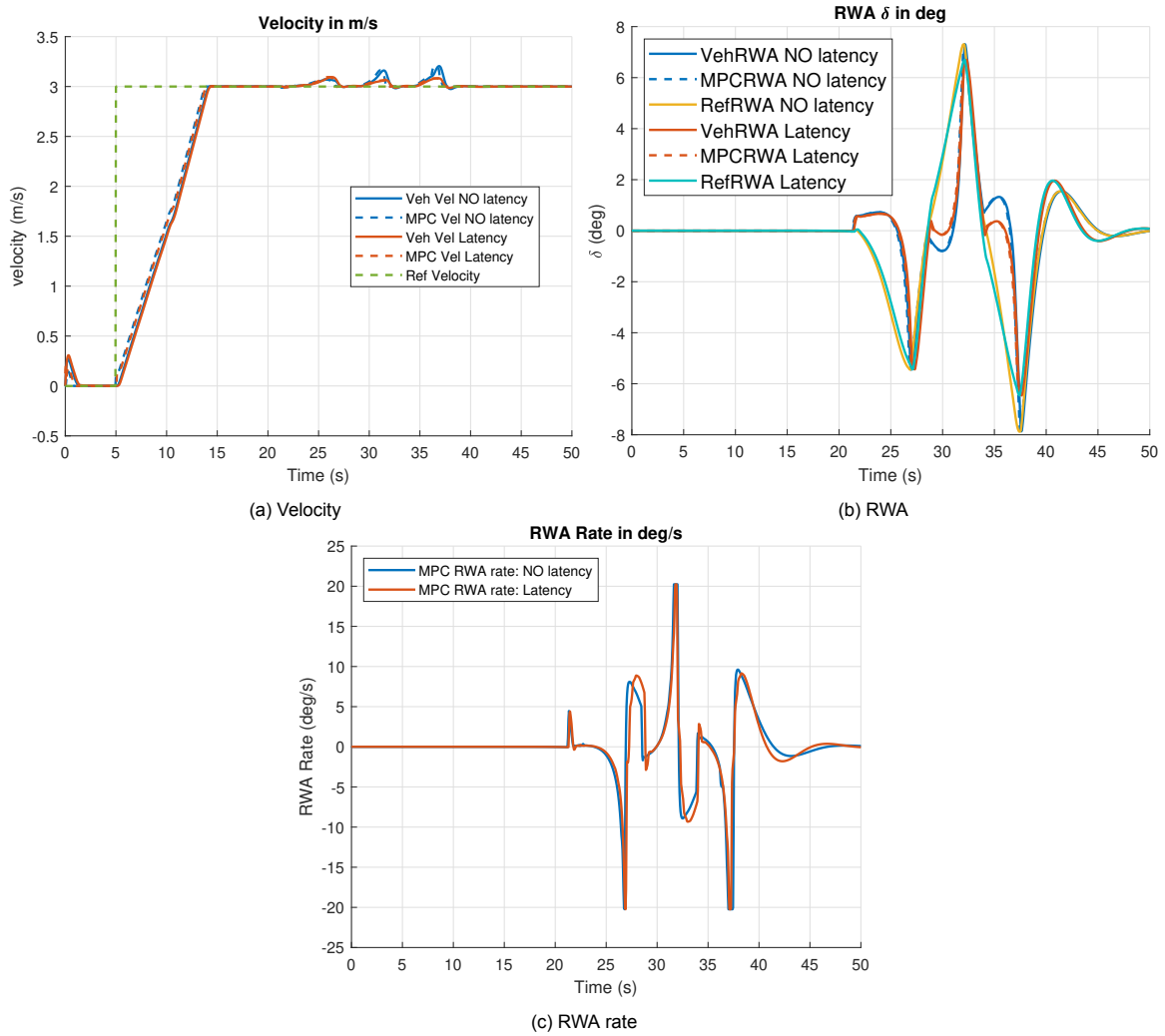


Figure 4.16: Latency vs No Latency: Trajectory Driven for static environment

In Figure 4.16, it can be observed that the driven trajectory is collision free for both the cases of latency and no latency. The driven trajectory for the latency case deviates from no latency case trajectory after crossing second obstacle and the deviation increases after obstacle three. The control inputs

generated by MPC are based on the references that are effectively generated for 200 ms old states. So by that time, the vehicle would have moved on by 0.6 m if it is travelling at 3 m/s. As mentioned earlier, MPC receives the most recent state feedback from the vehicle but reference inputs based on old states. This synchronisation mismatch in the state feedback and reference inputs can be accounted for the reason in the deviation of latency trajectory in the above figure.



(a) Velocity

(b) RWA

(c) RWA rate

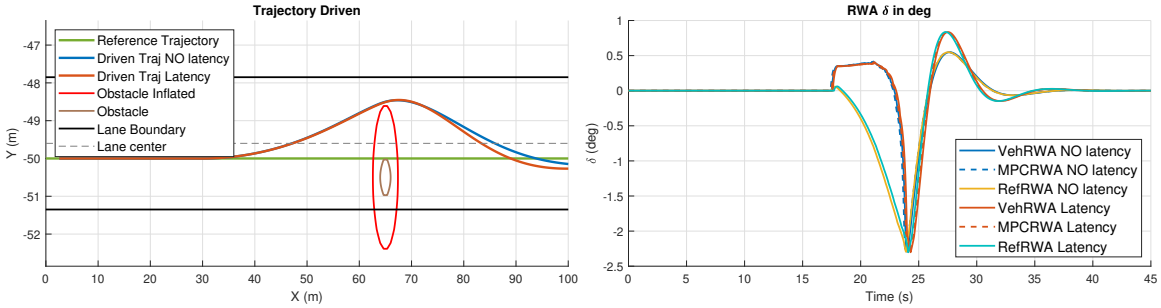
Figure 4.17: Latency vs No Latency: Velcoity v, RWA δ and RWA rate $\dot{\delta}$ performance for static environment

The vehicle velocity and computed MPC velocity for both the cases of latency and no latency are almost overlapping as evident from Figure 4.17a. Hence the velocity behaviour does not degrade significantly due to network latency. The steering behaviour also shows a similar result. Reference RWA for both the cases being similar, results in a similar evolution of vehicle and MPC RWA as can be seen in the Figure 4.17b. This is again reflected in the RWA rate in the Figure 4.17c. The deviation in trajectories for both the cases is reflected in the RWA in Figure 4.17b at time 30 s and 35 s. In both the cases, mean solution time was around 50 ms. Further plots on accelerations, deviation in RWA from reference and solver performance are presented in B.3.1

Performance at high speed

The above case considers vehicle travelling at a very slow speed of 3 m/s. The negative effect of latency is generally not expected to show at slow speed. Therefore, to check the effect of this low FTM latency for a vehicle travelling at 7 m/s, simulations were done using the Scenario 1a as given in 4.1.1.1. The simulations were not done for the same scenario as used above because with the same

set of tuning, the vehicle speed is too high to evade obstacles while travelling in between them, around the reference trajectory. Vehicle instead chooses to cross the obstacle three from below.



(a) Driven trajectory (b) RWA

Figure 4.18: Latency vs No Latency: Driven Trajectory and RWA plots for vehicle travelling at 7 m/s

The result from the Figure 4.18a is counter intuitive. High speed was expected to cause significant deterioration in ASC performance, however the trajectories driven for both the cases are collision free with some deviation in trajectories after crossing the obstacle. Even though it is noticeable, the overall performance is not unsafe. The deviation of RWA from the reference RWA for both the cases of latency and No latency shows similar and almost overlapping behaviour as can be seen in Figure 4.18b, with mismatches at 27 and 32 s which is again a reflection of deviations in driven trajectories. Since the vehicle is travelling at 7 m/s, the length of the predicted path increases to 35 m and hence controller can calculate collision free path well in advance. Therefore, high speed leads to longer predicted path which acts as a concealing effect for latency.

Performance under fluctuating latency

In the above simulations, constant latency was considered. To replicate the effect of variation in latency as one would expect in the real scenario, simulations were also done with latency fluctuating by $\pm 30\,\%$ around the mean latency values used before. A uniform distribution of fluctuations was used.

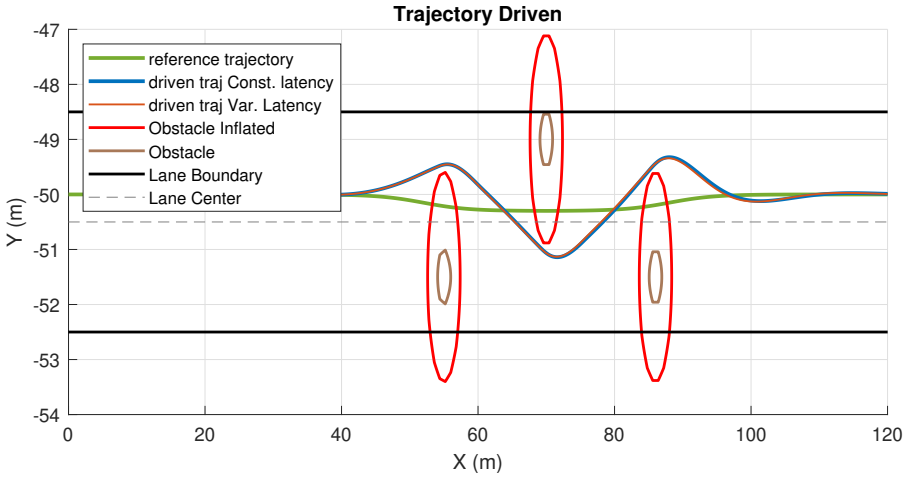


Figure 4.19: Trajectory Driven by vehicle equipped with ASC for two cases of constant latency and fluctuating latency.

Figure 4.19 shows that trajectories driven by vehicle in both the cases of constant latency and fluctuating latency are almost overlapping. This behaviour is seen in all the other plots for this comparison as presented in B.3.3. Therefore, fluctuating latency within the given range has almost the same result as that of constant latency.

4.5.2. Dynamic Environment

In the following, performance of ASC under the effect of latency when encountering dynamic obstacles is checked. The Scenario 1c given in 4.1.1.3 is used where two pedestrians cross the road while the human operator drives the vehicle towards their path. In these simulations, the obstacle position and velocity at each sampling time is taken and the obstacle is assumed to travel with the same velocity for the complete prediction horizon of MPC. With this assumption, a dummy obstacle is created for each MPC calculation, that covers exactly the same space as the obstacle would cover while travelling during the prediction horizon.

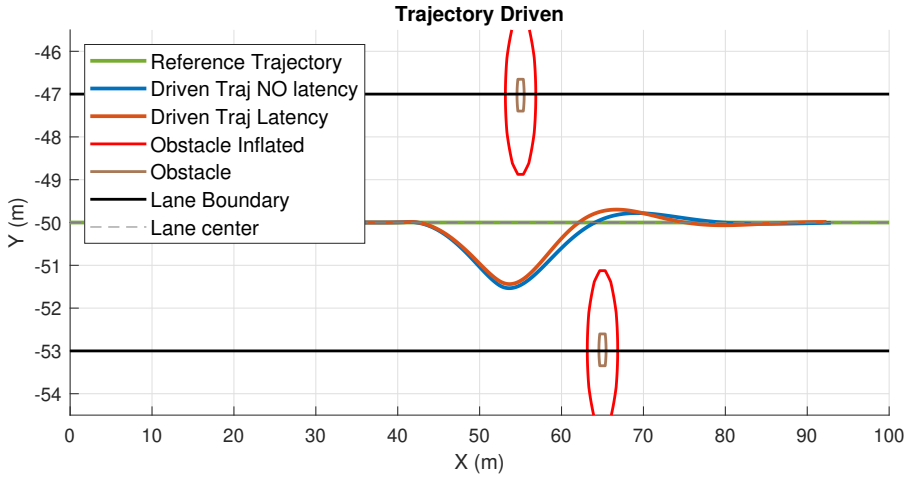


Figure 4.20: Latency vs No Latency: Trajectories driven for dynamic environment. The position of the obstacles is at the end of simulation where teh obstacles have crossed the road and the vehicle has travelled.

The Figure 4.20 shows the trajectories driven by the vehicle for both the cases of latency and no latency in the system. It shows the last frame of the animated plot where the obstacles have reached their goal position. The vehicle successfully avoided collision with both the obstacles and the trajectories for both latency cases are nearly similar with a little offset around x=65m.

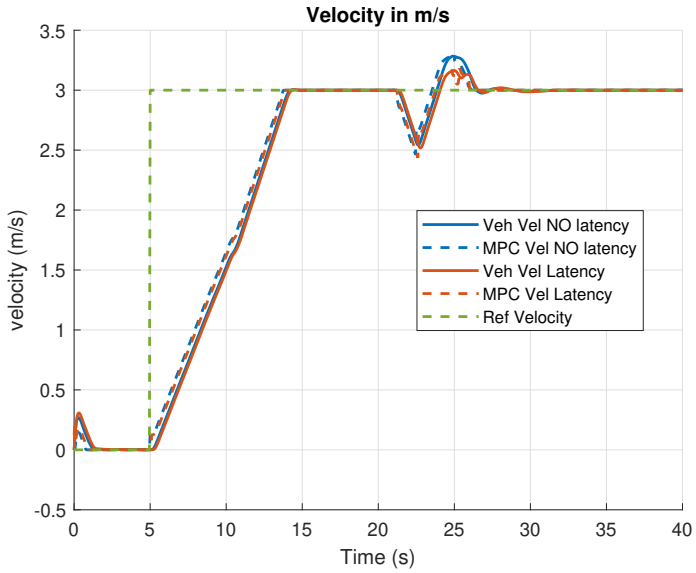
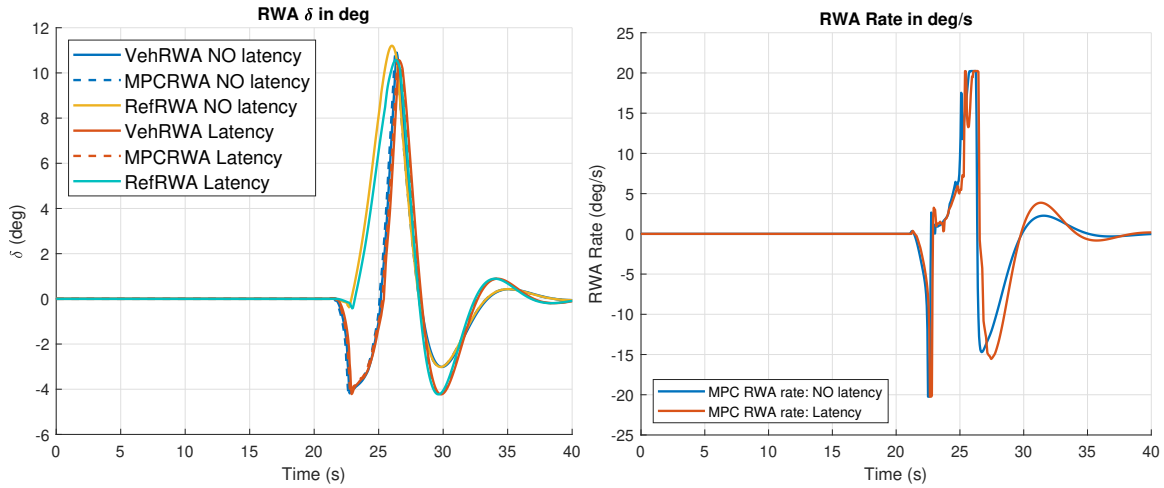


Figure 4.21: Latency vs No Latency: Velocity performance for dynamic environment

From the Figure 4.21 it can be seen that the controller reduces velocity for a while and then catches the reference velocity again. The reduction in velocity happens when the first obstacle is blocking the vehicle's path and thus controller tries to stop the vehicle. This also highlights the longitudinal control capability of ASC. Since the first obstacle is continuously traveling along the positive y-axis, free space

keeps on increasing behind it and makes way for the vehicle to go around and hence vehicle starts to accelerate once again to regain the reference speed. It attains a little more than the reference speed to stay away from the obstacle in an attempt to avoid collision.



(a) RWA (b) RWA rate

Figure 4.22: Latency vs No Latency: RWA δ and RWA rate $\dot{\delta}$ performance for dynamic environment

The reference RWA for both the cases stays similar until $\approx$ 27 s and then diverts from each other at 30 s and 33 s as can be seen in Figure 4.22a. This diversion is reflected in the RWA rate plot of Figure 4.22b. The vehicle and MPC RWA divert from their corresponding references from approximately 22 to 27 s owing to the fact that controller attempts to avoid collision during that time.

Thus, from the above results it can be concluded that low latency in the FTM setup does not have any detrimental effect on ASC performance and thus no serious latency mitigation techniques are needed.

4.6. Predictive Display for ToD under High Latency

This section discusses a latency mitigation technique, predictive display. As mentioned earlier, latency for the FTM setup is very low and no such mitigation techniques are needed and moreover, this has been shown in the previous section. However, for research purpose, predictive display has been explored for ToD working under high latency and some interesting results are presented here.

As mentioned in the beginning of this chapter with reference to the system architecture, reference block receives old states as feedback, which are delayed due to glass latency, and then corresponding references are generated. These reference signal are then further delayed due to actuator latency when transmitted to the MPC block. Meanwhile, the vehicle would have moved by time equal to the total round trip latency. In predictive display, vehicle position is predicted for the point in time when the vehicle receives control inputs, before information is given to the operator and hence time equal to the total round trip latency is taken to predict vehicle position. The camera images are enriched by this calculated predicted state and then displayed to the operator.

As described in Chucholowski et. al. [15], the Full Prediction method for predictive display is presented as a preferred method over the other methods shown in their work. In Full Prediction method, bicycle model (as presented in 2.1.1) is used to predict the vehicle position where vehicle velocity v and RWA δ are assumed to be constant. This method is taken as the baseline approach for predictive display.

The proposed predictive display method here, uses the predicted MPC states from the horizon as the state feedback to the reference block instead of using any prediction method at the reference block. This simplifies the process and needs no new set of calculations. MPC already uses model of the system and sets of constraints to predict the evolution of the states with reference to some objective. This method also eliminates the assumptions of constant velocity and RWA over the prediction horizon

and hence can be considered as more accurate than the baseline approach of Chucholowski.

In the following simulations, total round trip latency of 500 ms is used, more than double the FTM latency. Since the sampling time of MPC is 50 ms, 10^{th} predicted MPC state is used as state feedback for reference block. In this way, vehicle state in 500 ms is found.

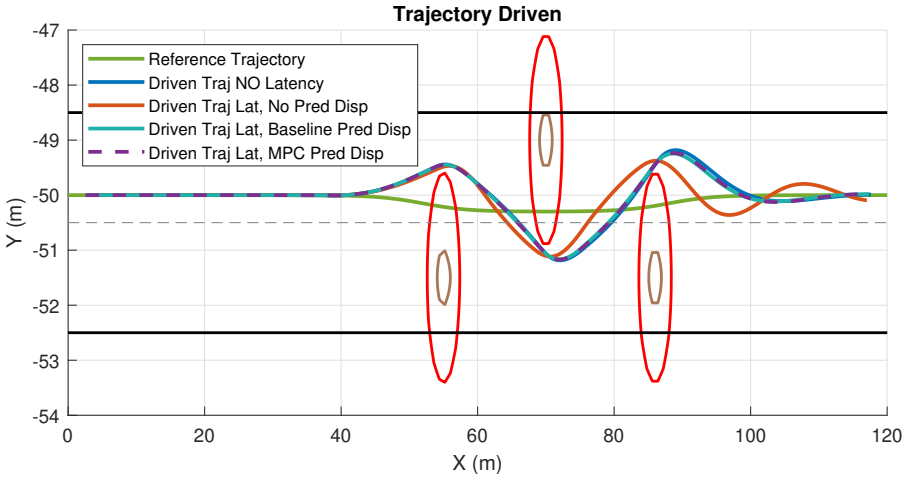
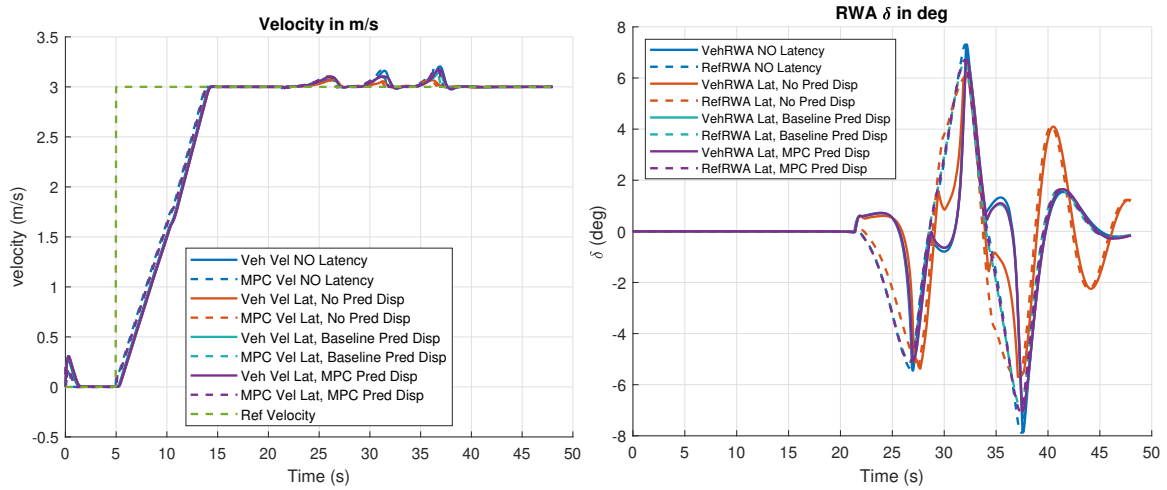


Figure 4.23: Predictive Display: Trajectory Driven for static environment

A good predictive display is expected to produce results equal to the case where there is no latency in the system. From the Figure 4.23 it can be observed that the driven vehicle trajectory for both the predictive display methods almost overlaps the vehicle trajectory for no latency in the system. Therefore, both the methods of predictive displays produce result equal to the case of no latency in the system and hence can be considered as effective. It can be also seen that the vehicle trajectory, for the case where no predictive display is used, stays collision free but not as smooth as in other cases.



(a) Velocity (b) RWA

Figure 4.24: Predictive Display: Velcoity v and RWA δ performance for static environment

From the figure 4.24a, the velocities for all the cases almost overlap each other. Since the driven vehicle trajectories almost overlap each other for the cases of no latency and the predictive display methods, the plots of RWA δ reflect the same information as can be seen in the Figure 4.24b.

Therefore, from the above results it can be concluded that predictive display can help mitigate the effect of high latency in the network and it becomes necessary to have such a mitigation technique as observed with the deterioration in the performance for the case where no such technique is used. Both

the predictive display methods show almost similar results. However, MPC based prediction method can be considered more accurate since it does not assume constant velocity and RWA for calculating predicted states. Further plots for this simulations are presented in B.4. It should be noted that, this concept should be further validated by performing actual tests on the ToD system setup. This concept of using MPC predicted state for predictive display could be also used in other teleoperation systems other than ToD.

4.7. Summary

This chapter discussed the simulation results that showcased the capabilities of ASC. Design and tuning of other components like FBLC and ACC was presented that were important to conduct simulations. FBLC was used to replicate human operator behaviour for simulations. Several scenarios of environment were used that were classified on the bases of human operator's driving inaccuracy and attentiveness. Before discussing the actual simulation results, the bounds on the constraints used for rest of the simulations were discussed.

The performance of controller for baseline approach was compared with that of ASC. The vehicle equipped with ASC behaved the way it was expected to and hence ASC turned out to be better than baseline approach. The performance of ASC was not exactly similar under the influence of network latency but was close to that for ASC under no latency. ASC performance for constant latency for high vehicle speed, fluctuating latency and dynamic environment was also checked. In no case, latency had a detrimental effect on the performance. Therefore, no special technique is required to compensate for the latency in the system. It should be noted that, these results considered the latency values as measured for the FTM setup. If the latency increases beyond these values, then the ASC performance can potentially degrade. To mitigate the effects of high latency, predictive display using MPC predicted states as the prediction method was also shown. However, this technique might not be needed for the FTM setup owing to the fact that it has low latency.

It should be also noted that, for all the simulations, unless specified, same set of tuning values for MPC, FBLC and ACC were used. The values of constraint bounds were also same for all the simulations. Hence, the ASC performance can be considered reliable for all the scenarios considered. Since these simulations were done using CarMaker, which provides an accurate vehicle model and simulation platform, the results are much trustworthy and almost similar results can be expected with the actual experimental vehicle.

5

Conclusion

This thesis explored the challenges in ToD and presented a safety system that helps make ToD safer. The primary difference between a regular shared control setup as found in road cars these days and shared control for ToD is that, the signals between human operator and the plant are delayed due to network latency. In addition to this, operating the vehicle from the workstation has its own challenges like, potential reduction in situational awareness and latency in the network. The ASC designed in this thesis helps overcome these challenges and also uses visual feedback to present additional information to the human operator that helps him understand the vehicle behaviour due to combined action of him and the controller.

The use of MPC as the control technique gave the benefit of predicting future vehicle states while considering multitude of constraints and model of the system, to compute optimal control input. The solver `acados` provided the luxury of solving non-linear optimization problem with low computation times and convenience to use the code for implementation in experimental vehicle. The simulation results confirmed the expected behaviour of ASC and highlighted its advantages over the baseline approach. The ASC performance was also checked under different cases of latency with respect to the FTM setup and it was found to be safe and quite similar to the performance in no latency case. However, predictive display for mitigating high latency in a ToD setup was also shown with a proposed approach of predicting vehicle position using MPC predicted states. By using CarMaker for simulations, the simulation results can be considered much reliable and similar results could be expected when tested on actual experimental vehicle. The implementation and testing of ASC on the experimental vehicle of FTM was part of the initial plan of this work. But unfortunately, it could not be realized due to the coronavirus pandemic.

Scope for Future Work

The ultimate test to check the performance of ASC would be to implement in the actual experimental vehicle and do real testing. Human factors investigation can also be carried out to determine the impact of latency on driving performance. The proposed predictive display method should also be implemented in the actual ToD system setup to check its performance in comparison with the baseline predictive display method. Another challenge in ToD, is connection loss. Path planning techniques like RRT* could be used to plan a collision free path for a safe parking maneuver in case of a complete connection loss. However, this is a conflicting solution owing to the fact that ToD was employed since the vehicle was not able to make decision by itself. Therefore, techniques need to be explored to address the challenge of connection loss.

A

Active Safety Control

To briefly summarize the appraoch, two ellipsoids $E_1 = Diag(\frac{1}{a^2}, \frac{1}{b^2})$ and $E_2 = Diag(\frac{1}{(a+\lambda)^2}, \frac{1}{(b+\lambda)^2})$ are considered. The semi-major and semi-minor axis of the E_1 ellipsoid are a and b respectively and E_2 is ellipsoid E_1 enlarged by λ in both the axis. To find the smallest ellipsoid that bounds the Minkowsky sum, a minimum value of λ needs to be found such that the distance between ellipsoid E_1 and E_2 is bigger than r^2, radius of the circle bounding the vehicle.

The lemma [51] used in the derivation to find λ is reproduced here as :

Lemma A.0.1 *Let $X^T A_1 X = 1$ and $X^T A_2 X = 1$ be two quadratics in R^n. Iff the matrix $A_1 - A_2$ is sign definite, then the square of the distance between the quadratic $X^T A_1 X = 1$ and the quadratic $X^T A_2 X = 1$ equals the minimal positive zero of the polynomial.*

$$f(z) = D_\lambda(\det \lambda A_1 + (z - \lambda A_2) - \lambda(z - \lambda A_1 A_2)) \tag{A.1}$$

where D stands for the discriminant of the polynomial treated with respect to λ.

By considering $A_1 = E_1$, $A_2 = E_2$ and $\{\lambda, a, b\} \in R^+$, $E_1 - E_2$ is ensured to be sign definite. Thus Lemma A.0.1 can be applied to determine the polynomial $F(z)$ and its roots. Square root of the minimal positive zero of $F(z)$ is the minimum distance equation. Hence, the minimum enlargement factor is found by solving for the value of λ that satisfies $\min_j \ \lambda(j) = r^2$. Interested reader can follow the detailed derivation as given in Brito et al. [11].

B

Simulation Results

B.1. Tuning Parameters

In the following tables, the vehicle parameters and different tuning parameters of the different system components used in the simulations are summarized. The same set of these values are used for all the simulations unless specified.

Vehicle Parameter	Values
l_f [m]	1.48
l_r [m]	1.504
Length [m]	4.950
Width [m]	1.9253
Steering to Road wheel angle ratio	16.8

Table B.1: Vehicle parameters for VOLVO XC90 T6 AWD

Parameter	Value
γ_1	1
γ_2	2
γ_3	0.25
Look ahead distance [m]	1

Table B.2: FBLC: Gains used for all the simulations

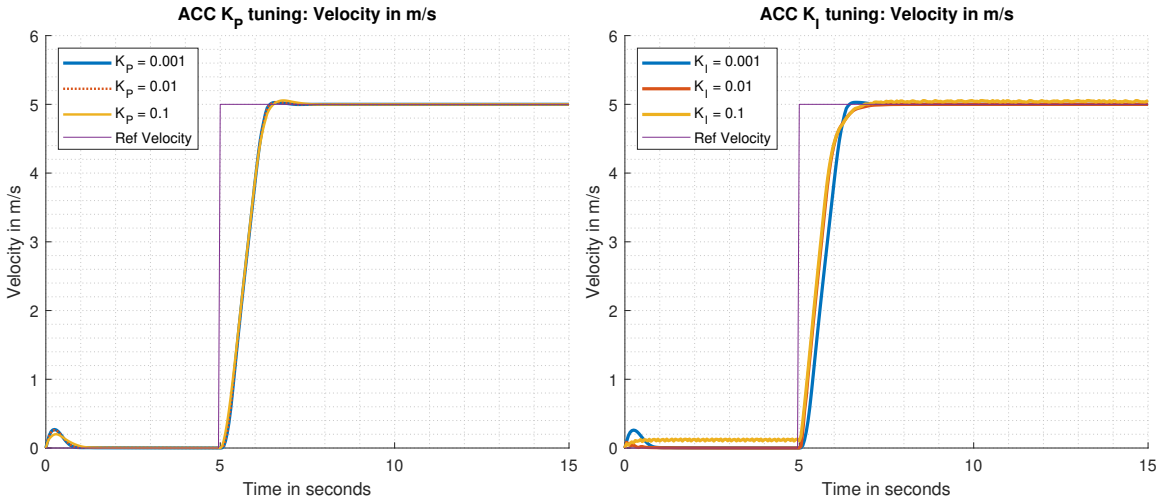
Parameter	Value
Δa	0.3
K_P	0.01
K_I	0.001

Table B.3: ACC: Gains used for all the simulations

Parameter	Value
Prediction Horizon	100
Sampling Time [s]	0.05
W_δ	100
W_v	1
W_S	10^8

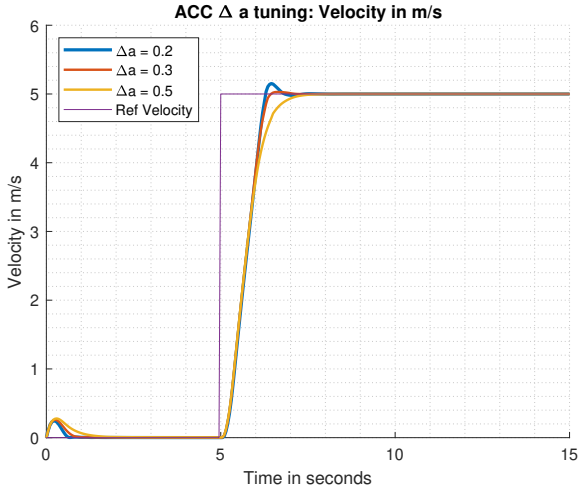
Table B.4: MPC: Penalties and tuning parameters used for all the simulations

B.2. Adaptive Cruise Control



(a) ACC: K_P tuning

(b) ACC: K_I tuning



(c) ACC: Δa tuning

Figure B.1: ACC: Step response plots for different tuning combinations

B.3. Latency vs No Latency

B.3.1. Static Environment: Comparison of No Latency with Constant Latency

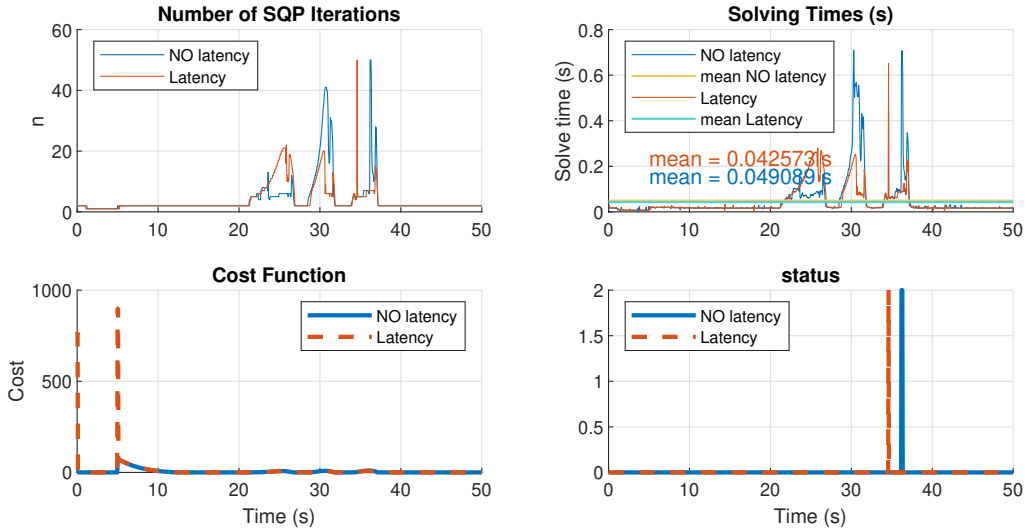
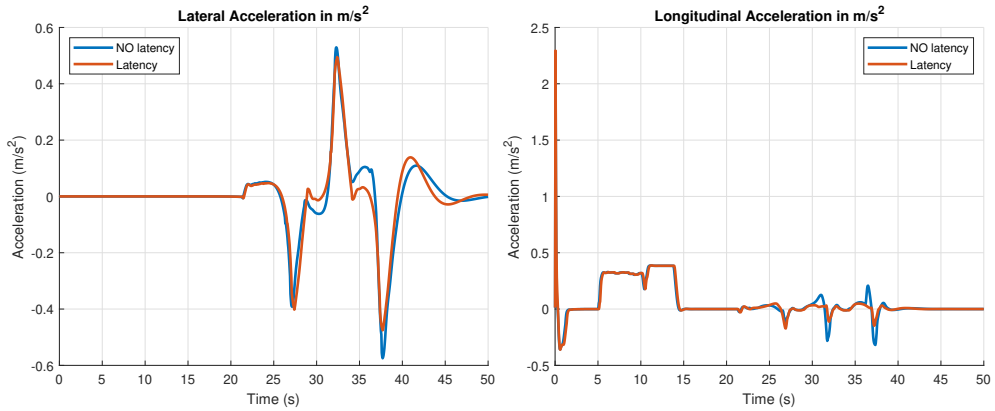


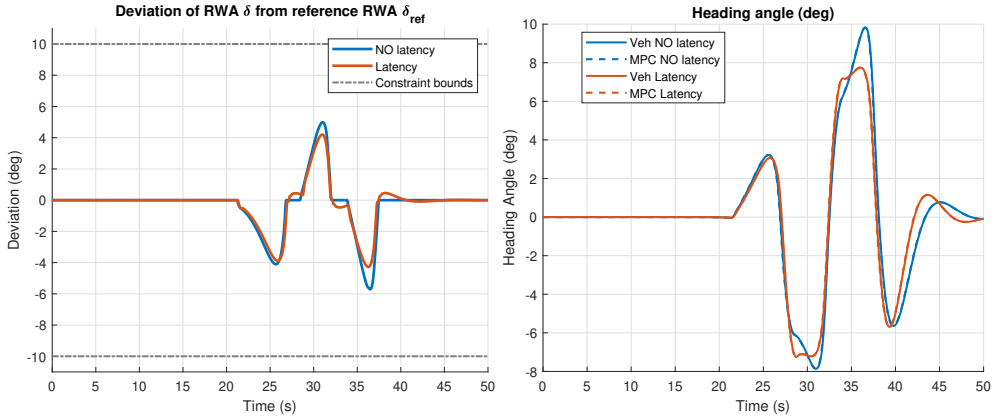
Figure B.2: Solver and cost function performance



(a) Lateral acceleration

(b) Longitudinal acceleration

Figure B.3: Acceleration performance



(a) Deviation of RWA from reference

(b) Heading Angle

Figure B.4: Deviation of RWA from reference RWA and heading angle performance

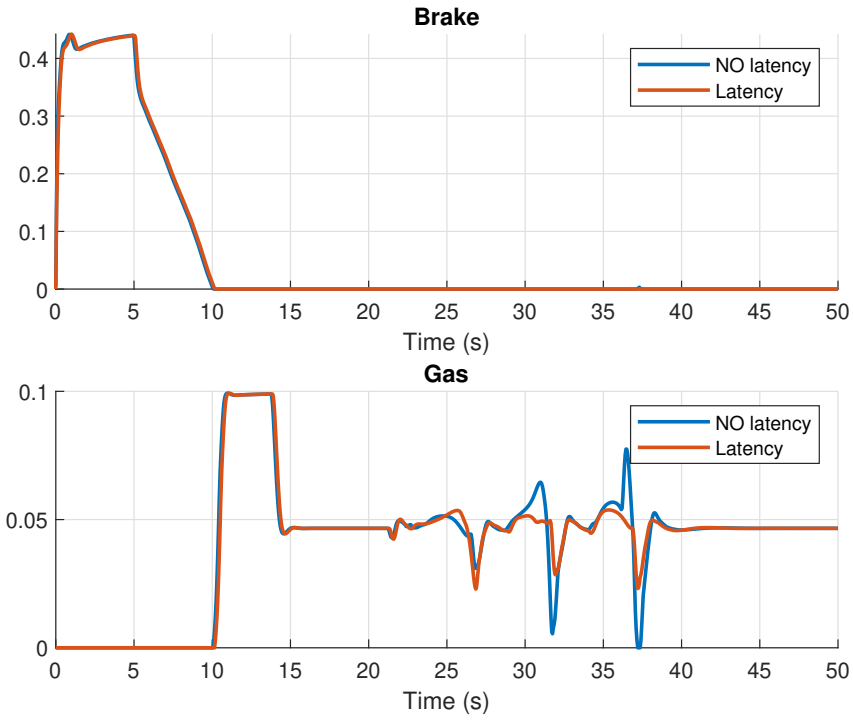
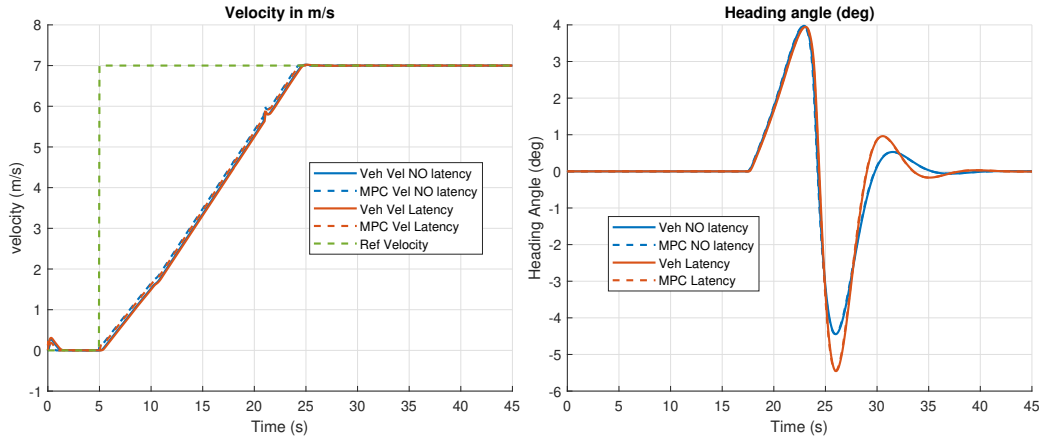


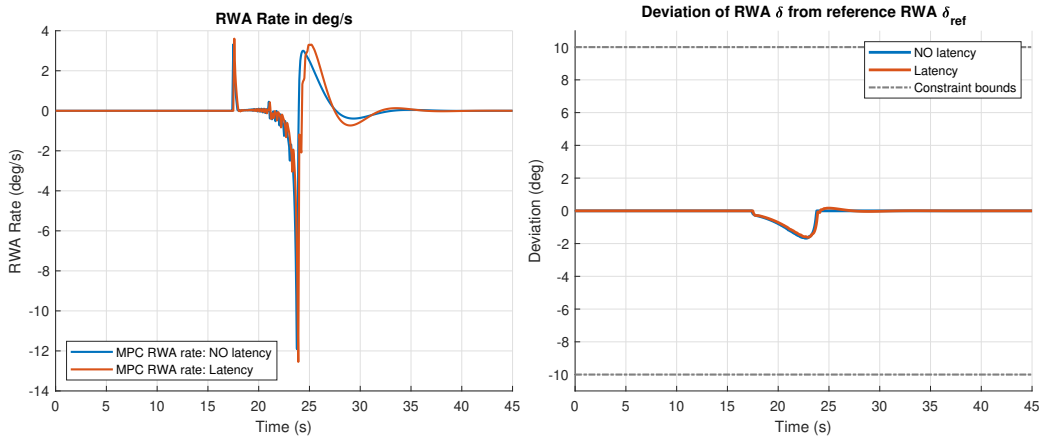
Figure B.5: Brake and Gas pedal performance

B.3.2. Static Environment: Comparison of No Latency with Constant Latency at High Speed



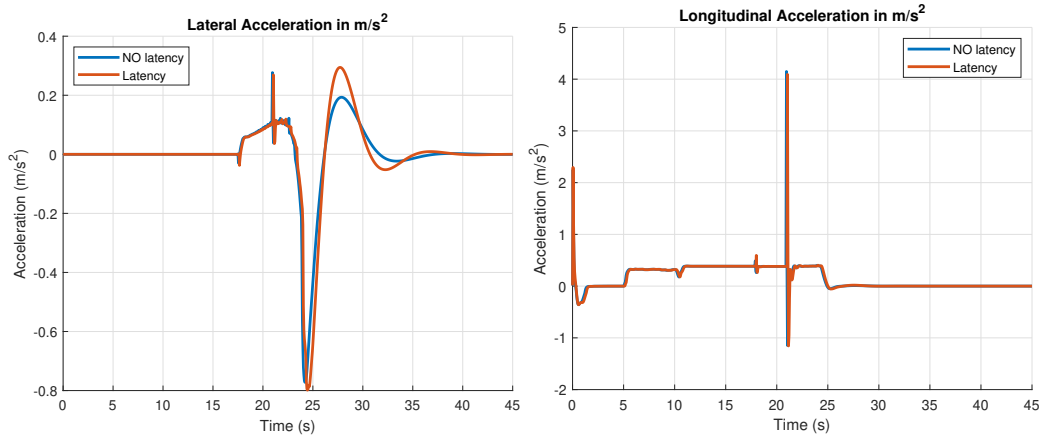
(a) Velocity (b) Heading angle

Figure B.6: Velocity and heading angle performance



(a) RWA rate (b) RWA

Figure B.7: RWA rate $\dot{\delta}$ and deviation of RWA from reference RWA performance



(a) Lateral acceleration (b) Longitudinal acceleration

Figure B.8: Acceleration performance

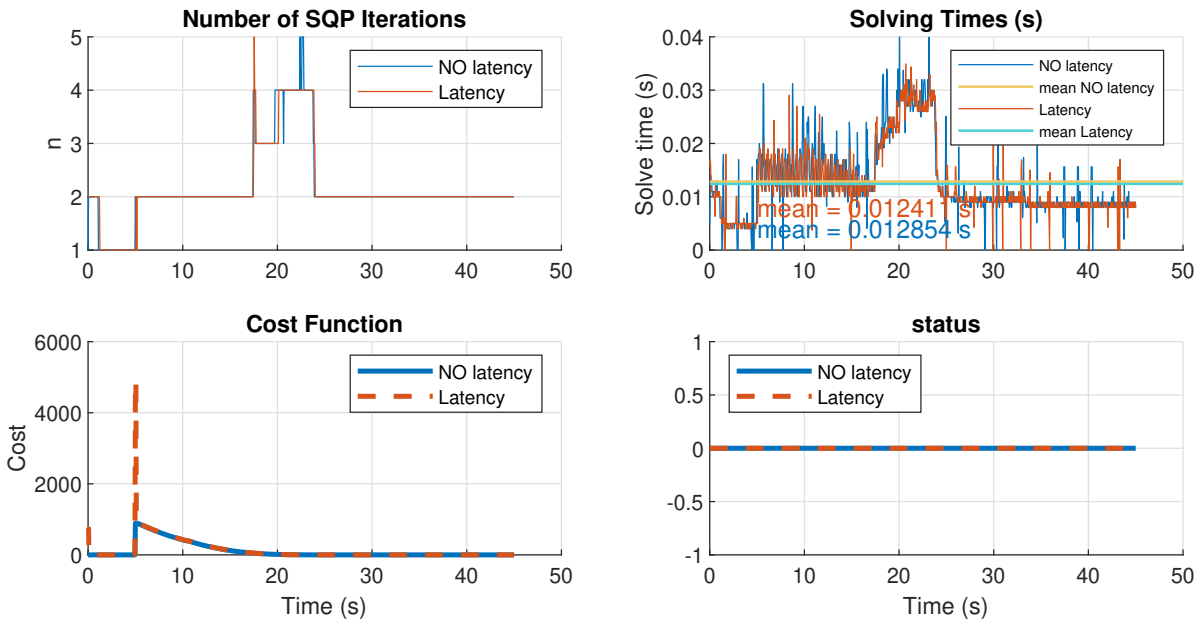


Figure B.9: Solver and cost function performance

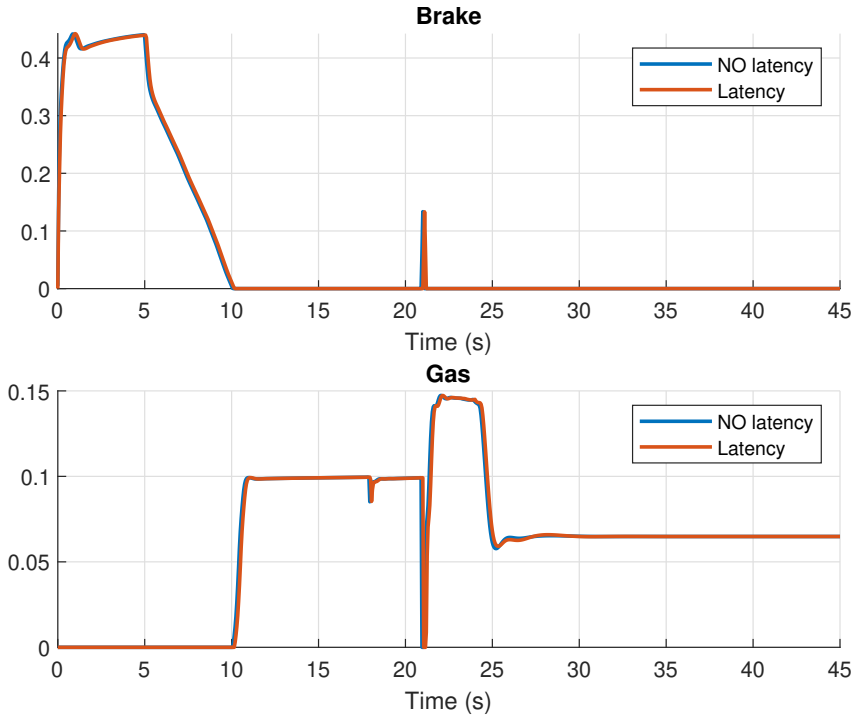
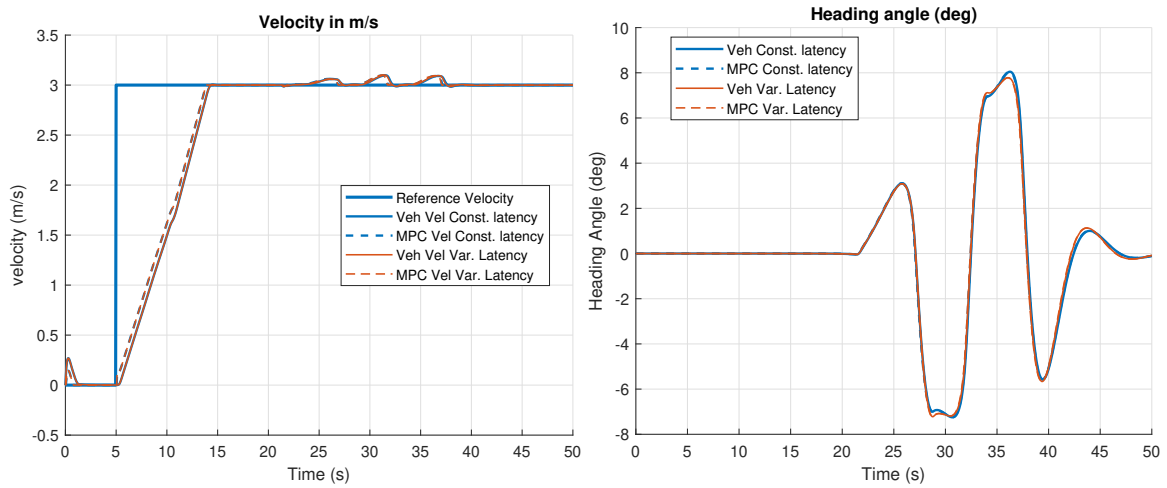


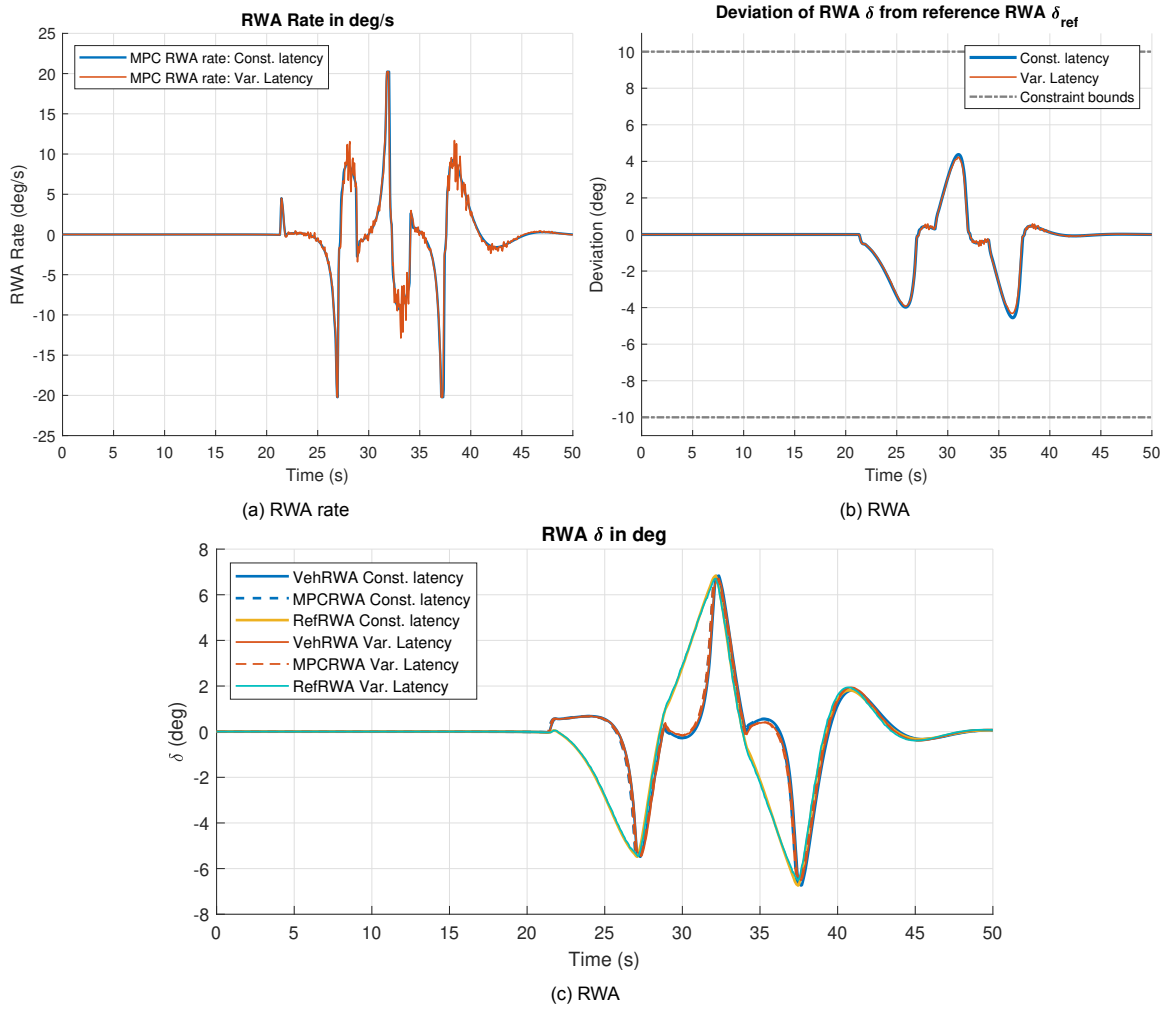
Figure B.10: Brake and Gas pedal performance

B.3.3. Static Environment: Comparison of Constant Latency with Fluctuating Latency



(a) Velocity (b) Heading angle

Figure B.11: Velocity and heading angle performance



(a) RWA rate (b) RWA

(c) RWA

Figure B.12: RWA δ, RWA rate $\dot{\delta}$ and deviation of RWA from reference RWA performance

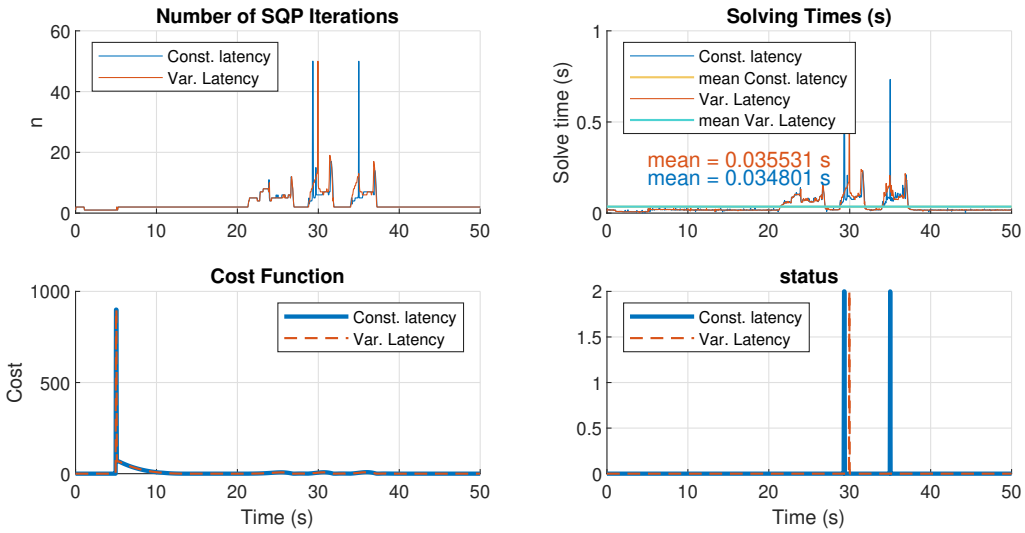
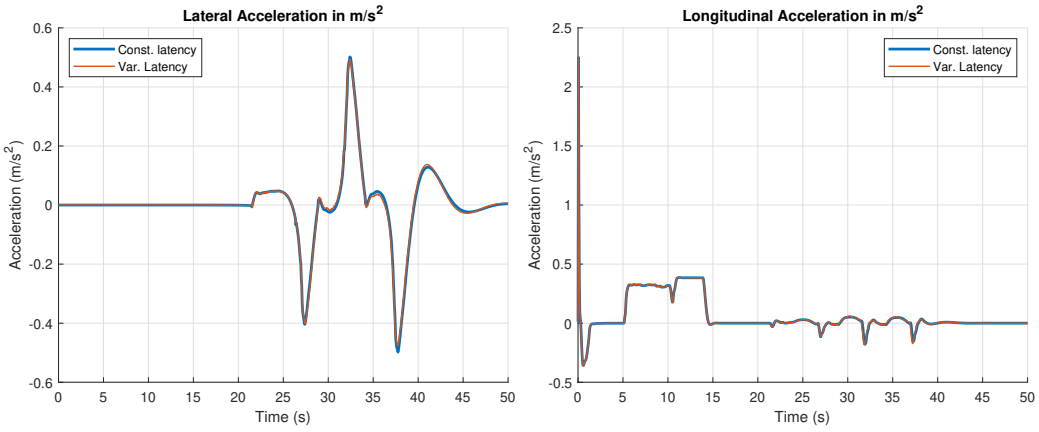


Figure B.13: Solver and cost function performance



(a) Lateral acceleration

(b) Longitudinal acceleration

Figure B.14: Acceleration performance

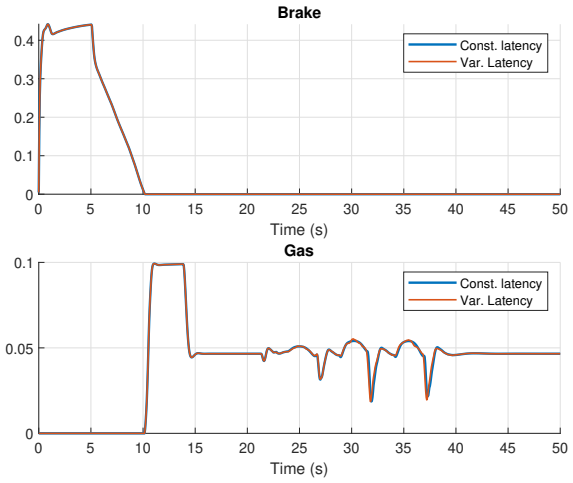


Figure B.15: Brake and Gas pedal performance

B.3.4. Dynamic Environment: Comparison of No Latency with Constant Latency

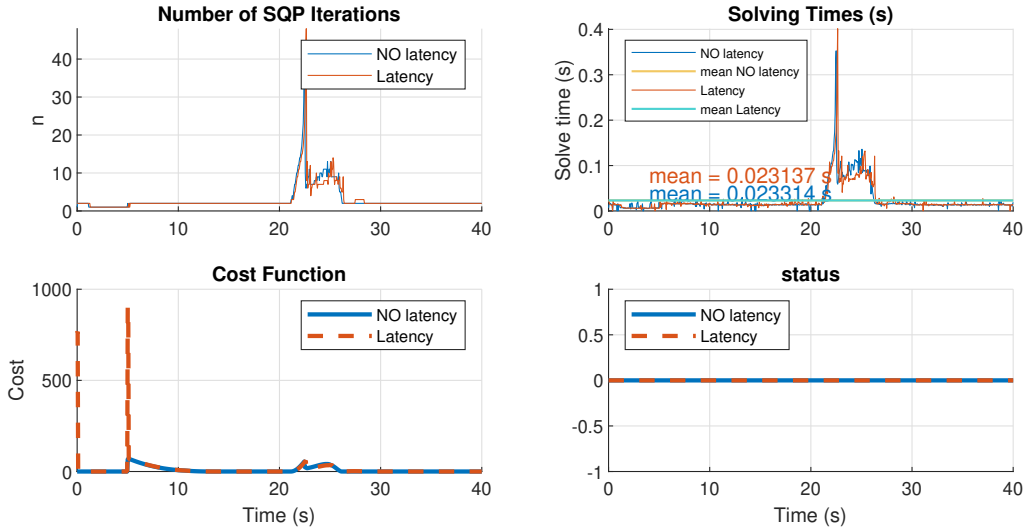
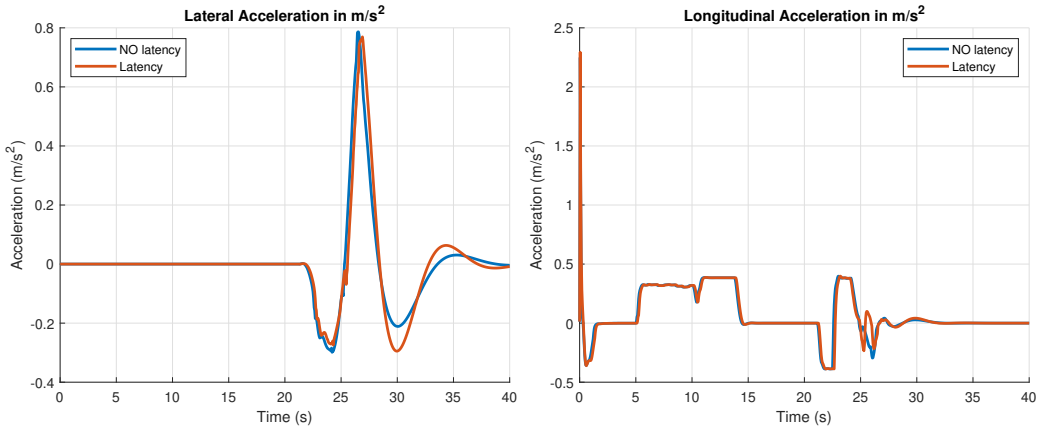


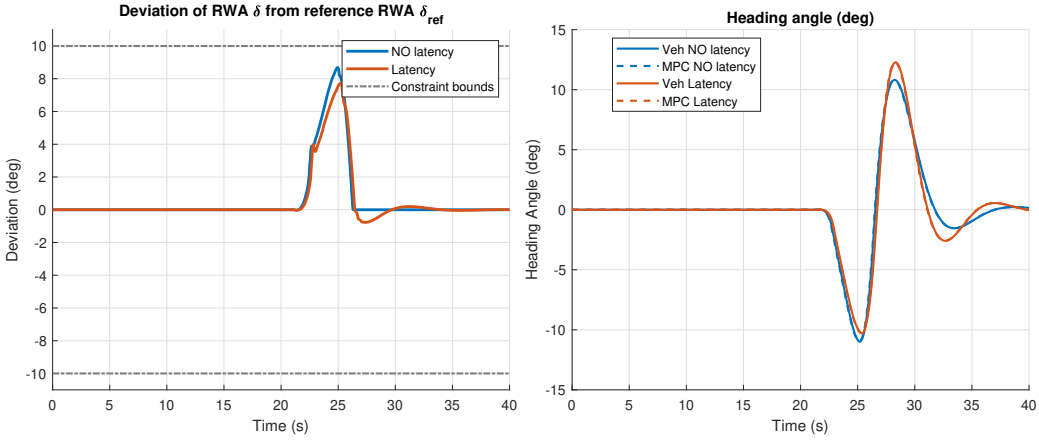
Figure B.16: Solver and cost function performance



(a) Lateral acceleration

(b) Longitudinal acceleration

Figure B.17: Acceleration performance



(a) Deviation of RWA from reference

(b) Heading Angle

Figure B.18: Deviation of RWA from reference RWA and heading angle performance

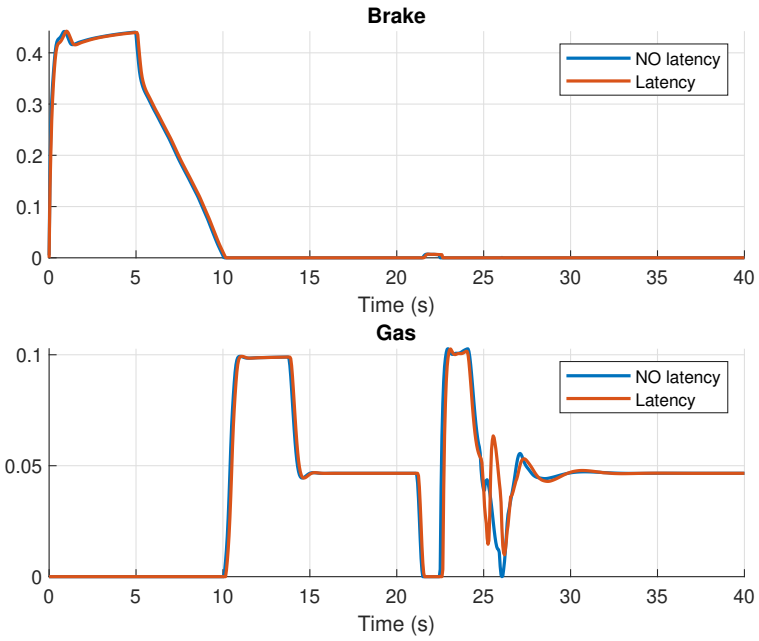


Figure B.19: Brake and Gas pedal performance

B.4. Predictive Display for ToD under High Latency

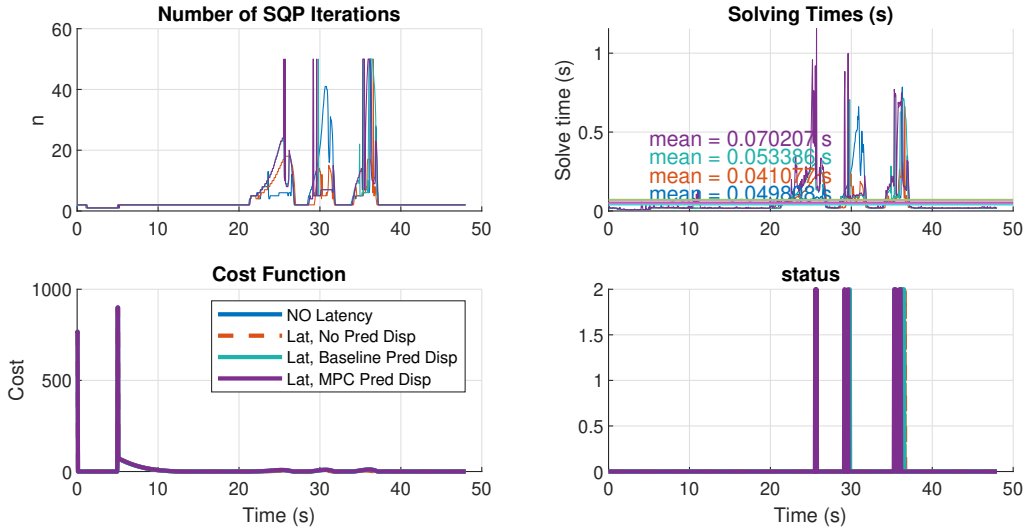
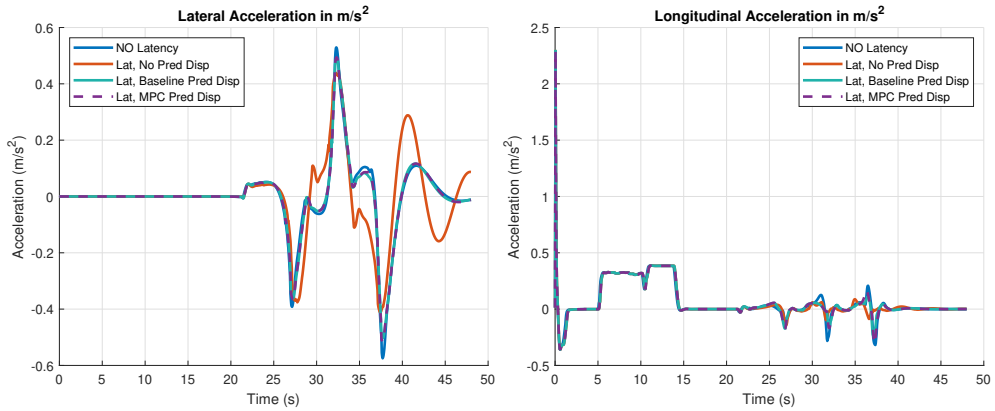


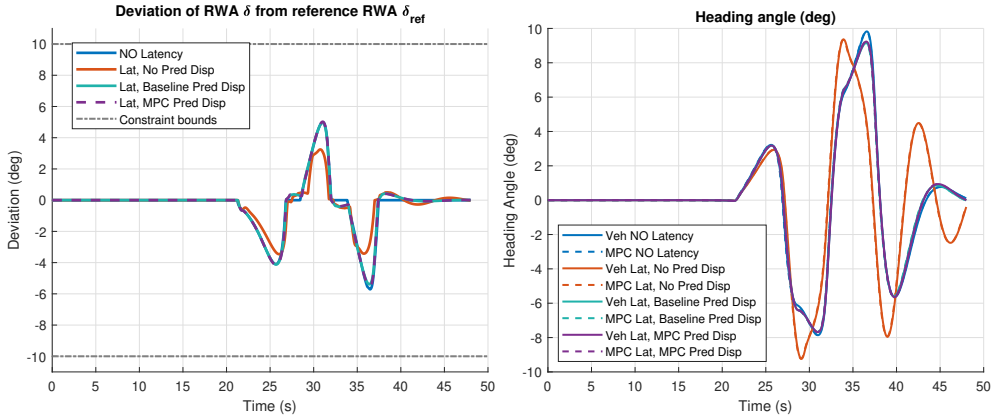
Figure B.20: Solver and cost function performance



(a) Lateral acceleration

(b) Longitudinal acceleration

Figure B.21: Acceleration performance



(a) Deviation of RWA from reference

(b) Heading Angle

Figure B.22: Deviation of RWA from reference RWA and heading angle performance

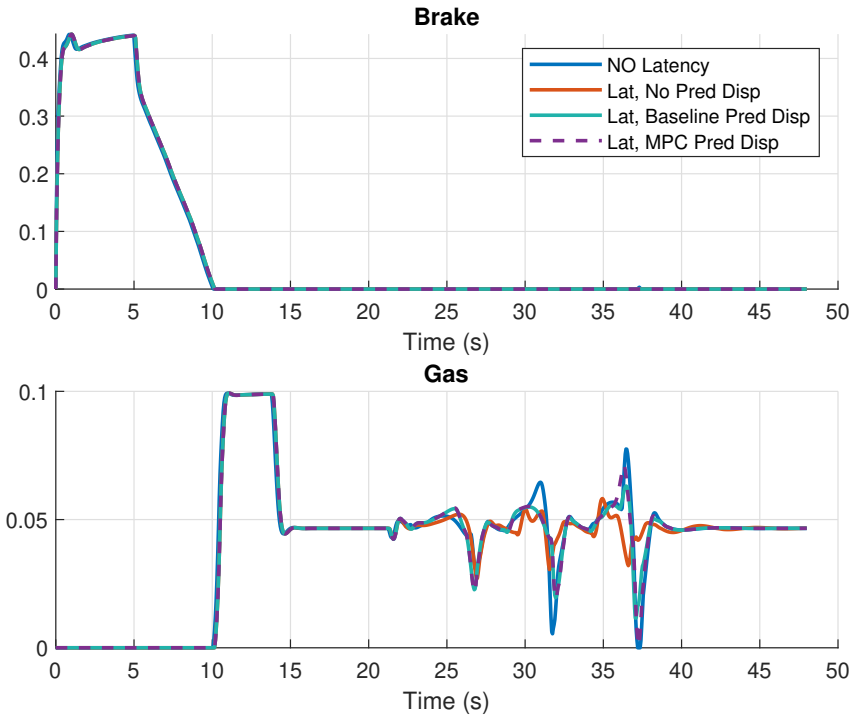


Figure B.23: Brake and Gas pedal performance

Bibliography

[1] EPRS European Commission 2017. Road fatality statistics in the eu (infographic), 2019. URL https://www.europarl.europa.eu/news/en/headlines/society/20190410STO36615/road-fatality-statistics-in-the-eu-infographic.

[2] Endor AG. URL https://fanatec.com/eu-en/.

[3] Sterling Anderson, Steven Peters, Thomas Pilutti, and Karl Iagnemma. An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios. *International Journal of Vehicle Autonomous Systems*, 8:190–216, 10 2010. doi: 10.1504/IJVAS.2010.035796.

[4] Joel A E Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019. doi: 10.1007/s12532-018-0139-4.

[5] Egbert Bakker, Lars Nyborg, and Hans B. Pacejka. Tyre modelling for use in vehicle dynamics studies. In *SAE Technical Paper*. SAE International, 02 1987. doi: 10.4271/870421. URL https://doi.org/10.4271/870421.

[6] A. Bemporad. Model predictive control design: New trends and tools. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 6678–6683, 2006.

[7] Björn Blissing, Fredrik Bruzelius, and Olle Eriksson. Effects of visual latency on vehicle driving behavior. *ACM Trans. Appl. Percept.*, 14(1), August 2016. ISSN 1544-3558. doi: 10.1145/2971320. URL https://doi-org.tudelft.idm.oclc.org/10.1145/2971320.

[8] H.G. Bock and K.J. Plitt. A multiple shooting algorithm for direct solution of optimal control problems*. *IFAC Proceedings Volumes*, 17(2):1603 – 1608, 1984. ISSN 1474-6670. doi: https://doi.org/10.1016/S1474-6670(17)61205-9. URL http://www.sciencedirect.com/science/article/pii/S1474667017612059. 9th IFAC World Congress: A Bridge Between Control Science and Technology, Budapest, Hungary, 2-6 July 1984.

[9] JF Bonnans, Jean Charles Gilbert, C Lemaréchal, and Claudia Sagastizabal. *Numerical Optimization – Theoretical and Practical Aspects*. 01 2006.

[10] Y.J. Reddy B.R. Mehta. *Industrial Process Automation Systems*. Butterworth-Heinemann, 2015. ISBN 9780128009390,.

[11] B. Brito, B. Floor, L. Ferranti, and J. Alonso-Mora. Model predictive contouring control for collision avoidance in unstructured dynamic environments. *IEEE Robotics and Automation Letters*, 4(4): 4459–4466, Oct 2019. ISSN 2377-3774. doi: 10.1109/LRA.2019.2929976.

[12] M. Bujarbaruah, Z. Ercan, V. Ivanovic, H. E. Tseng, and F. Borrelli. Torque based lane change assistance with active front steering. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, Oct 2017. doi: 10.1109/ITSC.2017.8317624.

[13] K. Burnett, A. Schimpe, S. Samavi, M. Gridseth, C. W. Liu, Q. Li, Z. Kroeze, and A. P. Schoellig. Building a winning self-driving car in six months. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9583–9589, 2019.

[14] Ryad Chellali and Khelifa Baizid. What maps and what displays for remote situation awareness and rov localization? In Gavriel Salvendy and Michael J. Smith, editors, *Human Interface and the Management of Information. Interacting with Information*, pages 364–372, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-21669-5.

[15] Frederic Chucholowski, Stefan Büchner, Johannes Reicheneder, and Markus Lienkamp. Prediction methods for teleoperated road vehicles. 01 2013. doi: 10.13140/2.1.1470.3048.

[16] On-Road Automated Driving (ORAD) committee. Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. *SAE*, 2018.

[17] Volvo Car Corporation. URL https://www.media.volvocars.com/global/en-gb/models/xc90/2017/specifications.

[18] C.V. Rao P.O.M. Scokaert D.Q. Mayne, J.B. Rawlings. Constrained model predictive control: Stability and optimality. *Automatica,*, pages 789–814, 2000. doi: https://doi.org/10.1016/S0005-1098(99)00214-9.

[19] Ziya Ercan, Ashwin Carvalho, Stéphanie Lefèvre, Francesco Borrelli, H. Eric Tseng, and Metin Gokasan. Torque-based steering assistance for collision avoidance during lane changes. 2016.

[20] S. M. Erlien, S. Fujita, and J. C. Gerdes. Shared steering control using safe envelopes for obstacle avoidance and vehicle stability. *IEEE Transactions on Intelligent Transportation Systems*, 17(2): 441–451, Feb 2016. ISSN 1558-0016. doi: 10.1109/TITS.2015.2453404.

[21] P. Falcone, M. Ali, and J. Sjoberg. Predictive threat assessment via reachability analysis and set invariance theory. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1352–1361, Dec 2011. ISSN 1558-0016. doi: 10.1109/TITS.2011.2158210.

[22] L. Ferranti, B. Brito, E. Pool, Y. Zheng, R. M. Ensing, R. Happee, B. Shyrokau, J. F. P. Kooij, J. Alonso-Mora, and D. M. Gavrila. Safevru: A research platform for the interaction of self-driving vehicles with vulnerable road users. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1660–1666, June 2019. doi: 10.1109/IVS.2019.8813899.

[23] Open Source Robotics Foundation. URL http://wiki.ros.org/rviz.

[24] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics Automation Magazine*, 4(1):23–33, March 1997. ISSN 1558-223X. doi: 10.1109/100.580977.

[25] C. Fulgenzi, A. Spalanzani, and C. Laugier. Dynamic obstacle avoidance in uncertain environment combining pvos and occupancy grid. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 1610–1616, April 2007. doi: 10.1109/ROBOT.2007.363554.

[26] DPSACE GMBH. URL https://www.dspace.com/en/pub/home.cfm.

[27] IPG Automotive GmbH, . URL https://ipg-automotive.com/products-services/simulation-software/carmaker/.

[28] IPG Automotive GmbH. *Reference Manual Version 8.1 CarMaker*. 2019.

[29] IPG Automotive GmbH. *User's Guide Version 8.1 CarMaker*. 2019.

[30] SENSODRIVE GmbH, . URL https://www.sensodrive.de/.

[31] A. Hosseini, D. Bacara, and M. Lienkamp. A system design for automotive augmented reality using stereo night vision. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 127–133, 2014.

[32] Amin Hosseini. *Conception of Advanced Driver Assistance Systems for Precise and Safe Control of Teleoperated Road Vehicles in Urban Environments*. PhD thesis, 01 2018.

[33] Boris Houska, Hans Joachim Ferreau, and Moritz Diehl. An auto-generated real-time iteration algorithm for nonlinear mpc in the microsecond range. *Automatica*, 47(10):2279 – 2285, 2011. ISSN 0005-1098. doi: https://doi.org/10.1016/j.automatica.2011.08.020. URL http://www.sciencedirect.com/science/article/pii/S0005109811003918.

[34] Boris Houska, Joachim Ferreau, and Moritz Diehl. Acado toolkit–an open source framework for automatic control and dynamic optimization. *Optimal Control Applications and Methods*, 32:298 – 312, 05 2011. doi: 10.1002/oca.939.

[35] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli. Kinematic and dynamic vehicle models for autonomous driving control design. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 1094–1099, 2015.

[36] Flohr F. Pool E.A.I. Gavrila D.M. Kooij, J.F.P. Context-based path prediction for targets with switching dynamics. *Int J Comput Vis*, page 239–262, 2019. doi: https://doi-org.tudelft.idm.oclc.org/10.1007/s11263-018-1104-4.

[37] Alexander Liniger, Alexander Domahidi, and Manfred Morari. Optimization-based autonomous racing of 1:43 scale rc cars. *Optimal Control Applications and Methods*, 36(5):628–647, Jul 2014. ISSN 0143-2087. doi: 10.1002/oca.2123. URL http://dx.doi.org/10.1002/oca.2123.

[38] C. Lv, D. Cao, Y. Zhao, D. J. Auger, M. Sullman, H. Wang, L. M. Dutka, L. Skrypchuk, and A. Mouzakitis. Analysis of autopilot disengagements occurring during autonomous vehicle testing. *IEEE/CAA Journal of Automatica Sinica*, 5(1):58–68, 2018.

[39] Ellips Masehian and Yalda Katebi. Robot motion planning in dynamic environments with moving obstacles and target. *World Academy of Science, Engineering and Technology*, 29, 01 2007.

[40] C. J. Ostafew, A. P. Schoellig, and T. D. Barfoot. Visual teach and repeat, repeat, repeat: Iterative learning control to improve mobile robot path tracking in challenging outdoor environments. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 176–181, 2013.

[41] High performance interior-point qp solvers. URL https://github.com/giaf/hpipm.

[42] P. Polack, F. Altché, B. d'Andréa-Novel, and A. de La Fortelle. The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles? In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 812–818, 2017.

[43] Rajesh Rajamani. *Vehicle Dynamics and Control*. Springer US, USA, 2 edition, 2012. ISBN 978-1-4614-1433-9.

[44] Y. Rasekhipour, A. Khajepour, S. Chen, and B. Litkouhi. A potential field-based model predictive path-planning controller for autonomous road vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 18(5):1255–1267, May 2017. ISSN 1558-0016. doi: 10.1109/TITS.2016.2604240.

[45] C. Samson and K. Ait-Abderrahim. Feedback control of a nonholonomic wheeled cart in cartesian space. In *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pages 1136–1141 vol.2, 1991.

[46] W. Schwarting, J. Alonso-Mora, L. Paull, S. Karaman, and D. Rus. Safe nonlinear trajectory generation for parallel autonomy with a dynamic vehicle model. *IEEE Transactions on Intelligent Transportation Systems*, 19(9):2994–3008, Sep. 2018. ISSN 1558-0016. doi: 10.1109/TITS.2017.2771351.

[47] SENSODRIVE GmbH SENSO-Wheel SD-LC. URL https://www.sensodrive.de/products/senso-wheel-sd-lc_e.php.

[48] T.B. Sheridan. Teleoperation, telerobotics and telepresence: A progress report. *Control Engineering Practice*, pages 205–214, 1995,.

[49] Z. Shiller, F. Large, and S. Sekhavat. Motion planning in dynamic environments: obstacles moving along arbitrary trajectories. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, volume 4, pages 3716–3721 vol.4, May 2001. doi: 10.1109/ROBOT.2001.933196.

[50] Sky Sports. Are a tennis player's reactions better than an f1 driver's?, 2018. URL https://fanatec.com/eu-en/.

[51] Alexei Yu. Uteshev and Marina V. Yashina. Metric problems for quadrics in multidimensional space. *Journal of Symbolic Computation*, 68:287 – 315, 2015. ISSN 0747-7171. doi: https://doi.org/10.1016/j.jsc.2014.09.021. URL http://www.sciencedirect.com/science/article/pii/S0747717114000893.

[52] E. Velenis, E. Frazzoli, and P. Tsiotras. On steady-state cornering equilibria for wheeled vehicles with drift. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 3545–3550, 2009.

[53] Robin Verschueren, Gianluca Frison, Dimitris Kouzoupis, Niels van Duijkeren, Andrea Zanelli, Branimir Novoselnik, Jonathan Frey, Thivaharan Albin, Rien Quirynen, and Moritz Diehl. acados: a modular open-source framework for fast embedded optimal control. *arXiv preprint*, 2019. URL https://arxiv.org/abs/1910.13753.

[54] Andreas Wächter and Lorenz Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106:25–57, 03 2006. doi: 10.1007/s10107-004-0559-y.

[55] L. Zeng and G. M. Bone. Collision avoidance for nonholonomic mobile robots among unpredictable dynamic obstacles including humans. In *2010 IEEE International Conference on Automation Science and Engineering*, pages 940–947, Aug 2010. doi: 10.1109/COASE.2010.5584243.