

# MOTION BOUNDARIES FORMATION FOR AUTONOMOUS VEHICLES

by

Joel Bannis

BSc. Geomatics, University of the West Indies, 2011

Master of Spatial Analysis, Ryerson University, 2012

An MRP

presented to Ryerson University

in partial fulfillment of

the requirements for the degree of

Master of Engineering

in the program of

Civil Engineering

Toronto, Ontario, Canada, 2020

© Joel Bannis, 2020

## **Author's Declaration For Electronic Submission Of An MRP**

I hereby declare that I am the sole author of this MRP. This is a true copy of the MRP, including any required final revisions.

I authorize Ryerson University to lend this MRP to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this MRP by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my MRP may be made electronically available to the public.

# MOTION BOUNDARIES FORMATION FOR AUTONOMOUS VEHICLES

Joel Bannis

Master of Engineering

Civil Engineering

Ryerson University, 2020

## **Abstract**

In this paper, the application of Model Predictive Control to perform curvilinear motion planning is explored. More specifically, nonlinear MPC will be focused on because of its proven efficiency in the modeling of uncertainties as well as in nonlinear model dynamics. The main objective of this report is to show that with proper modeling and formulation of motion constraints, curvilinear motion planning can be achieved with nonlinear MPC. The trajectory of the vehicle will be tracked with the least error while satisfying constraints such as speed and steering angles. Simulations are presented which demonstrate the ability of the suggested models to successfully perform curvilinear motion staying safely within the bounds, while simulations of several models validate its performance. A deterministic sensitivity analysis was conducted in order to determine the impact of the prediction horizon time. Experimental results show that a critical prediction horizon time approximately 10 to 13 seconds was identified as the ideal range for optimal results of the model.

## **Acknowledgments**

First and foremost, I would like to thank God, whose blessing I owe everything to. I wish to thank my supervisor Dr. Said Easa without whose guidance this MRP would not have been completed. I am grateful to Dr. Maxym Diachuk for making this research idea possible. Your time, direction and timely support are what has allowed me to successfully complete this report.

## **Dedication**

*I dedicate this to my wife and daughter. Their encouragement and support have been the inspiration that aided me towards the completion of this report.*

# Table Of Contents

<b>DECLARATION</b>	<b>ii</b>
<b>ABSTRACT</b>	<b>iii</b>
<b>ACKNOWLEDGEMENTS</b>	<b>iv</b>
<b>DEDICATION</b>	<b>v</b>
<b>LIST OF TABLES</b>	<b>viii</b>
<b>LIST OF FIGURES</b>	<b>ix</b>
<b>NOMENCLATURE</b>	<b>xi</b>
<b>1 INTRODUCTION</b>	<b>1</b>
1.1 Problem Statement . . . . .	2
1.2 Scope . . . . .	2
1.3 Objectives . . . . .	2
<b>2 LITERATURE REVIEW</b>	<b>3</b>
2.1 Path Planning . . . . .	3
2.2 MPC Theory . . . . .	5
2.2.1 Linear MPC . . . . .	5
2.2.2 Nonlinear MPC . . . . .	6
2.3 MPC Formulation . . . . .	10
2.3.1 Prediction Model . . . . .	12
2.3.2 Cost Function Description . . . . .	13
2.3.3 Constraint Function Description . . . . .	14
2.4 Solution of the Optimization Problem . . . . .	14

2.4.1	Sequential Quadratic Programming . . . . .	15
2.4.2	Linear Quadratic Regulator . . . . .	16
2.5	Vehicle Models . . . . .	16
<b>3</b>	<b>METHODOLOGY</b>	<b>19</b>
3.1	Vehicle Modeling . . . . .	21
3.1.1	Single Vehicle . . . . .	22
3.1.2	Single Long Truck . . . . .	27
3.1.3	Tractor Semitrailer Vehicle . . . . .	32
3.1.4	Tractor-Semitrailer Vehicle with Semitrailer's Steered Axle . . . . .	36
3.2	Optimization . . . . .	40
<b>4</b>	<b>SIMULATION RESULTS</b>	<b>43</b>
4.1	Simulation Environment . . . . .	43
4.2	Motion Boundary Simulation . . . . .	43
4.2.1	Scenario 1: Circle Motion . . . . .	43
4.2.2	Scenario 2: No Space Constraints . . . . .	54
4.3	Sensitivity Analysis . . . . .	61
<b>5</b>	<b>CONCLUSIONS</b>	<b>65</b>
5.1	Summary of Results . . . . .	65
5.2	Future Work . . . . .	66
	<b>REFERENCES</b>	<b>67</b>

# LIST OF TABLES

4.1	Control parameters for passenger car vehicle model . . . . .	44
4.2	State and constraint parameters for passenger car vehicle model . . . . .	45
4.3	Control parameters for single truck vehicle model . . . . .	47
4.4	State and constraint parameters for single truck vehicle model . . . . .	47
4.5	Control parameters for conventional tractor-semitrailer vehicle model . . . . .	49
4.6	State and constraint parameters for conventional tractor-semitrailer vehicle model .	50
4.7	Control parameters for articulated vehicle with semitrailer steered axle model . . .	52
4.8	State and constraint parameters for articulated vehicle with semitrailer steered axle model . . . . .	52
4.9	No space state and constraint parameters for passenger car vehicle model . . . . .	54
4.10	No space state and constraint parameters for single truck vehicle model . . . . .	56
4.11	No space state and constraint parameters for conventional tractor-semitrailer vehi- cle model . . . . .	57
4.12	No space state and constraint parameters for articulated vehicle with semitrailer steered axle model . . . . .	59

# LIST OF FIGURES

2.1	Schematic representation of the model predictive control algorithm . . . . .	11
2.2	Kinematics of lateral vehicle motion . . . . .	17
3.1	Flowchart of nonlinear MPC algorithm as it is used in model simulation . . . . .	20
3.2	Kinematics of curvilinear motion of single passenger vehicle (Kong et al., 2015) . .	23
3.3	Kinematics of curvilinear motion of single truck with steered rear axle (Espinosa et al., 1998) . . . . .	28
3.4	Kinematics of curvilinear motion of an articulated vehicle (Ritzen et al., 2016, Werner et al., 2012) . . . . .	33
3.5	Kinematics of curvilinear motion of an articulated vehicle with semitrailer’s steered axle (Ritzen et al., 2016, Werner et al., 2012) . . . . .	37
4.1	Predicted trajectory for a passenger car in autonomous mode at the roundabout arc	45
4.2	Results of states and controls for passenger car in autonomous mode at the roundabout arc: . . . . .	46
4.3	Predicted trajectory for a single truck in autonomous mode at the roundabout arc .	48
4.4	Results of states and controls for a single truck in autonomous mode at the roundabout arc . . . . .	48
4.5	Predicted trajectory for a tractor-semitrailer in autonomous mode at the roundabout arc . . . . .	50
4.6	Results of states and controls for a tractor-semitrailer in autonomous mode at the roundabout arc: . . . . .	51
4.7	Predicted trajectory for an articulated vehicle with steered semitrailer’s axle in autonomous mode at the roundabout arc . . . . .	53
4.8	Results of states and controls for an articulated vehicle with steered semitrailer’s axle in autonomous mode at the roundabout arc . . . . .	53

4.9	Predicted trajectory for a passenger car in autonomous mode for parking place changing . . . . .	55
4.10	Results of states and controls for passenger car in autonomous mode for parking place changing . . . . .	55
4.11	Predicted trajectory for a single truck in autonomous mode for parking place changing	56
4.12	Results of states and controls for a single truck in autonomous mode for parking place changing . . . . .	57
4.13	Predicted trajectory for a tractor-semitrailer in autonomous mode for parking place changing . . . . .	58
4.14	Results of states and controls for a tractor-semitrailer in autonomous mode for parking place changing . . . . .	58
4.15	Predicted trajectory for an articulated vehicle with steered semitrailer's axle for parking place changing . . . . .	60
4.16	Results of states and controls for an articulated vehicle with steered semitrailer's axle in autonomous mode for parking place changing . . . . .	60
4.17	Computing time in seconds and mean acceleration with different prediction horizon times . . . . .	62
4.18	Results of sensitivity analysis for a passenger car in autonomous mode at the round-about arc with different prediction horizon times . . . . .	64

# NOMENCLATURE

$\omega_\phi$	Vehicle yaw rate
$\omega_\psi$	Semitrailer articulation rate
$\omega_\theta$	Vehicle front steering rate
$\omega_\zeta$	Vehicle rear axle steering rate
$\phi$	Vehicle yaw angle
$\psi$	Semitrailer articulation angle
$\theta$	Vehicle front steering angle
$\zeta$	Vehicle rear axle steering angle
$a$	Vehicle longitudinal acceleration
$e_1$	Fifth wheel shift relative to the tractor's rear axle
$H$	Lane width
$L$	Wheelbase distance
$L_1$	wheelbase distance
$L_2$	wheelbase distance
$R$	Outer boundary radius
$r$	Inner boundary radius
$v$	Vehicle velocity
$v_x$	Vehicle longitudinal velocity along global x-coordinate

$v_y$  Vehicle lateral velocity along global y-coordinate

$x$  Vehicle x-displacement

$y$  Vehicle y-displacement

# 1 INTRODUCTION

The research topic of autonomous vehicles (AV) and its applications has attracted an overwhelming interest over the years and is continually growing. The rapid developments and advancements in technology have allowed researchers to realize the possibility of having fully AV seamlessly operating on transportation networks around the world. As such, a fully AV is expected to perform all aspects of the dynamic driving task under all roadway and environmental conditions that can be managed by a human driver. These systems aim at driving the vehicle autonomously by controlling the steering, braking and throttling in real-time according to established planned path and controls. There are numerous research studies focused on active steering methods to reduce the number of motor vehicle accidents and make the driving experience safer. One aspect of concern is obstacle avoidance. Obstacle avoidance can be summed up as the task of satisfying positional constraints with the goal of preventing object collisions. This is different from simple path planning. Obstacle avoidance involves implementation of reactive control on the dynamic environments, whereas, planning implies pre-computation within path boundaries. As such, collision avoidance systems make up the core aspect of advanced active safety systems. The importance of obstacle avoidance in fully AV is emphasized by its potential for increasing road safety and the overall reduction of occurring accidents. Vehicles equipped with these systems can identify obstacles on the road such as pedestrians, cyclists, other vehicles as well as lane makers. They are intelligent enough to perform emergency maneuvers to avoid collisions or perform parking procedures independently. This suggests that a reliable and accurate obstacle avoidance system is paramount for fully AV.

MPC has been highlighted to be an effective means used various areas of automotive control solutions including obstacle avoidance. This is because MPC has been adopted to handle nonlinearities and uncertainties as well as constraints exceptionally well. This report presents such an application of model predictive control applied to motion planning for AV. A detailed methodology is presented on the various components required for the optimal design of an MPC such as cost function, mathematical model and constraints.

## **1.1 Problem Statement**

In this research, the application of Model Predictive Control to perform curvilinear motion planning is explored. Nonlinear MPC will be focused on because of its proven efficiency in the modeling of uncertainties as well as in nonlinear model dynamics. The trajectory of the vehicle will be tracked with the least error while satisfying constraints such as speed and steering angles. The optimal solution of the NMPC allows explicit constraints for the controls and states and minimizes a cost function, which ensures for a vehicle arriving at its destination point avoid any obstacles.

## **1.2 Scope**

Nonlinear mathematical models for four (4) autonomous design vehicles are developed and tested for curvilinear motion. The simulations are performed in MATLAB's Model-Based Design Environment, Simulink. Robustness testing is performed by varying vehicle parameters including speed and steering angles, yaw angle vs. folding angle and acceleration/steering rate controls.

## **1.3 Objectives**

This report is focused on motion planning using NMPC for AV. The main objective of this paper is to show that with proper modeling and formulation of motion constraints, curvilinear motion planning can be achieved with nonlinear MPC. The second objective is to test the NMPC precision and adequacy and its sensitivity to changes in the vehicle state parameters.

# 2 LITERATURE REVIEW

A fully autonomous vehicle is expected to perform all aspects of the dynamic driving task under all roadway and environmental conditions that can be managed by a human driver. To work effectively, AV systems need the capability of path planning, obstacle avoidance, object detection and object classification.

Obstacle avoidance is one of the critical technologies that ensure human and vehicle safety. In order to avoid collision with fixed obstacles such as lane boundaries, a system must be able to detect the boundaries and recalculate an optimal path. As it relates to an AV, after detecting a boundary, the system will need to steer itself towards a safe and efficient path in real-time. This highlights the two main characteristics which define this problem; the nonlinearity of the vehicle dynamics and the presence of time-varying state and input constraints while navigating in a dynamic environment. Considering how complex this is to accomplish, there are a number of control methods that have been proposed in literature for such an intricate system. Some of these include path planning (Du and Tan, 2015, Saback et al., 2019) and model predictive control (MPC) (Frasch et al., 2018, Falcone et al., 2007) which has shown to be a more effective method for such applications.

## 2.1 Path Planning

In the process of development of AV, advanced technologies in planning and control are required. More specifically, planning algorithms allow an AV to determine the behavior of the vehicle by itself (Chu et al., 2012). Planning algorithms for AV have been divided into two stages. The global planning stage involves plotting the global route using waypoints. This is usually generated from sources such as digital maps and a localization system. The local planning stage is where the local path is generated. This is done from information gathered based on the global route and information obtained from sensors such as radar or Lidar. Researchers have used this approach to achieve some form of obstacle avoidance, however, some limitations are apparent. The method

presented by Shiller and Gwo (1991), combines various aspects of motion planning of AV in a unified way on general terrains. Vehicle dynamics and kinematics were taken into consideration as well as topography, obstacles and regions with various mobility. In this study, the local optimal path was obtained by using a parameter optimization procedure. Using the motion time as the cost function, the initial guess to the local optimization problem is obtained by a global graph search (Shiller and Gwo, 1991). Obstacle avoidance was demonstrated by placing obstacles to block the unconstrained local optimal solution. It was found that the best solution obtained was in the vicinity of a local optimal path. Huang et al. (2014), presented a real-time optimization method for AV motion planning using differential dynamic programming. Motion planning includes trajectory generation and trajectory optimization. Conventional methods of optimization involve trajectory smoothing however, Huang et al. (2014), decided to integrate differential dynamic programming to solve the optimization problem. The motion planner was tested for obstacle avoidance, however, as this was not the focus of the study the algorithm was not modified with that in mind. Results show that the planner chose a trajectory to avoid the obstacle. Once the obstacle was avoided the planner was able to choose the trajectory which returns to the center of the road. Differential dynamic programming considers the vehicle model in the optimization and ensures a kinematically feasible trajectory for the vehicle. Although obstacle avoidance was not the focus of the study, the algorithm was able to avoid the obstacle and navigate back on track.

Marino et al. (2009), presented a nested proportional integrating derivative (PID) steering control for lane-keeping in vision-based AV. This system was designed to perform path following in the case of roads with an uncertain curvature. The researchers compared the results from simulations using the nested PID system with the MPC used in in the CarSim environment and found that MPC performed slightly better. According to Du et al. (2016), the predictive ability of MPC distinguishes itself from the other control methods. One of the major drawbacks of typical approaches such as path planning is that these methods leave the vehicle dynamics out of consideration. On the other hand, Chu et al. (2012), presented local path planning for off-road autonomous driving with avoidance of static obstacles. Using a real-time path-planning algorithm, the optimal path

for off-road autonomous driving with static obstacles avoidance was achieved. This provides a design framework for local path planning that uses the global route and local environmental data. It illustrates how with a predefined route, path planning is capable of obstacle avoidance, however, because the path is predefined, there is no dynamic capability of the system to respond to new obstacles on its own.

## **2.2 MPC Theory**

The general concept of receding horizon control and model predictive control (MPC) can be traced back to the 1960s (Morari and Lee, 1997). Over the years interest in the field has been growing and expanding attracting many researchers around the world. MPC has applications in many areas of engineering from chemical engineering to mechanical engineering even spanning across to civil engineering (Guay et al., 2017). At its core, MPC is a framework for implanting existing tools of optimal control. MPC is considered a practical approach that is suited to control dynamical constrained systems. It can to handle input and output constraints by directly incorporating it into the optimization problem. It is also considered a control method which explicitly uses a model of the system to predict its state at future time instant. One of the major advantages of MPC is that it is considered straight forward to formulate the models based on well-understood concepts. It also inherently explicitly handles constraints; this allows for the development time to be much shorter than for competing advanced control methods. Overall, it is considered easier to maintain since that changing the model or specs does not require complete redesign. As such, MPC is generally chosen because of its capability of systematically incorporating nonlinearities. These qualities are extremely important for motion boundary formation for AV.

### **2.2.1 Linear MPC**

When the predicted state and output trajectory can be expressed as a linear function of the current state this is a case of a linear MPC. Furthermore, the resulting optimization problem can either be

a case of linear or quadratic programming. This is determined by the state of the cost function. As such, both linear programming and quadratic programming problems can be efficiently solved using high-performance solvers. There have been many research studies which have utilized this method of linear MPC as it relates to obstacle avoidance.

Jiang et al. (2017), presents an approach for real-time obstacle avoidance of AV. An MPC scheme based on convex quadratic programming (CQP) is developed to generate safe trajectories. To reduce the computational burden in optimizing the performance index of MPC, linear time-varying MPC is adopted and a unique single dimension artificial potential fields (SDAPF) method to utilize the obstacle information is proposed. AV with this proposed method can track the desired path if there is no obstacle on it and avoid both static and dynamic obstacles if the path is occupied (Jiang et al., 2017). Simulation results show the validity of the approach and its superior real-time performance, which is critical to AV. Turri et al. (2013), presents a linear MPC formulation that specifically addresses the lane-keeping and obstacle avoidance problems for a passenger car driving on low curvature roads. The linearity of the model and convexity of the constraints is used to change the MPC problem to resemble a set of quadratic subproblems. The low computational complexity of each subproblem allows the researchers to solve the MPC problem in real-time while using long prediction horizons. Tests for obstacle avoidance were performed and results showed that the vehicle avoids the obstacle and returns to the road centerline with a low overshoot (Turri et al., 2013). The researchers also note that the performance is like that seen in simulations. However, it is important to note that the simulation scenario with multiple obstacles was not considered in the experiments due to the lack of testing time.

### **2.2.2 Nonlinear MPC**

Nonlinear MPC makes it impossible to express exact prediction as a linear function of current state and input trajectory. As a result, even if the cost function was linear or quadratic, the resulting optimization problem would be nonlinear. The cost function should be formulated in such a way that its minimization leads to the fulfillment of control objectives. In the case of

obstacle avoidance, for example, the cost function would need to be formulated so that when it is optimized the objective of tracking of reference variables is achieved. Compared to linear MPC, nonlinear MPC is computationally demanding and as expected a completely different approach to the optimization must be taken. Depending on the target application, nonlinear MPC has many use cases in the area of AV, more specifically, motion boundary formation for AV. Huang and Panagou (2017), presented research that supports using nonlinear model predictive control approach for automated vehicles turning and merging automatically. Quirynen et al. (2018), outlines embedded optimization algorithm for steering in AV based on nonlinear model predictive control. The purpose of the research by Rafaila and Livint, (2015), is to present a nonlinear model-based predictive control of autonomous steering for ground vehicles. These examples of research using nonlinear MPC pave the way for its application in motion planning research.

As an expansion of motion boundary formation and motion planning, we can consider obstacle avoidance of AV. Using similar principles, nonlinear MPC controllers are used to solve obstacle avoidance problems. Abbas et al. (2017), focused on implementing an MPC based trajectory controller for obstacle avoidance in an autonomous ground vehicle. Interestingly, the model uses a simplified bicycle model within the control, but the researchers validated the model by implementing it in a fully nonlinear CarSim model where obstacle avoidance was tested. This controller is tested under several constrained scenarios including static obstacle avoidance and avoidance of obstacles with more complex constraints. The results show that the proposed method was able to handle the dynamic trajectory changes and unanticipated obstacles at normal road speeds. Similarly, Liu et al. (2018), presents the formulation of nonlinear MPC for obstacle avoidance in high speed, large size autonomous ground vehicles (AVG). These vehicles have a high center of gravity that operates in unstructured environments. Unstructured environments refer to the fact that there are no lanes or traffic rules to follow. This presented a challenge for the researchers because other MPC formulations which are specific to autonomous passenger vehicles could not be used in this context. As such, the researchers needed to develop a new MPC formulation to navigate an AVG and avoid obstacles while doing so safely.

The main goal of this study was to develop a model that will allow the AVG to move from its initial position to a specified target location as fast and as safe as possible. As this was expected to be used for military applications, travel time was a priority and was expected to be minimized. The nonlinear MPC obstacle avoidance algorithm developed for this research is based on two parts. The Lidar data processor and the control command generator (Liu et al., 2018). Lidar was used to locate the obstacles of varying size and shapes within the prediction horizon. The algorithm together with the plant forms a closed-loop system. The control command generator consists of a multiphase optimal control problem for each feasible opening which is solved in parallel. Two different vehicle models were used to predict vehicle trajectories. Simulation results show that the method developed can yield a satisfactory performance in a variety of scenarios. As this is a specific model created for and AVG and the unstructured environments it was tested in, this specific model will maybe not be applicable for autonomous passenger vehicles. As noted by the researchers, future research directions include moving obstacles and pursuing real-time implantation of the algorithm. It would be interesting to see this model applied to autonomous passenger vehicles to determine how results are affected based on the environment in which the vehicles are tested.

The dynamic vehicle model that is used in the nonlinear MPC method can accommodate varying levels of complexity. This complexity can range based on the type of vehicle model utilized, from a simplified single-track bicycle models to being combined with a four-wheel model. This combined with the choice of vehicle dynamics and environment can increase the overall complexity of the model. Yoon et al. (2009), presents a model predictive control approach for trajectory generation of unmanned ground vehicles (UGVs) combined with a tire model. An optimal tracking problem while avoiding collision with obstacles is formulated in terms of cost minimization under constraints. Information on obstacles is incorporated online in the nonlinear model-predictive framework as they are sensed within a limited sensing range. The overall problem is solved online with nonlinear programming. For the local path regeneration upon detecting new obstacles, the cost function is augmented using the obstacle information in two methods. The first method uses the distance from the UGV to the nearest detected obstacle, and the second method uses the parallax

information from the vehicle about the detected obstacles. Simulation results in cluttered and dynamic environments show that the modified parallax method effectively reflects the threat of the obstacles to the UGV considering the dimension and state variables of the vehicle, showing clear improvements over the distance-based methods. Similarly, Park et al. (2009), focused on an obstacle avoidance algorithm for ground vehicles based on nonlinear MPC. However, the approach taken by the researchers was to separate the MPC based trajectory generation from the tracking controller. This decision was made because of the difficulties in the calculation of optimal solutions using complex nonlinear models. The trajectories were generated so that they were compatible with the vehicle dynamics. The obstacle avoiding trajectories were calculated using the steepest gradient descent method. The actual plant used in the simulations was the CarSim vehicle model. Simulation results show that the generated trajectories are compatible with the full non-linear vehicle dynamics. Furthermore, satisfactory performance in terms of collision avoidance was shown in both static and dynamic environments.

Over the last several decades, MPC technology has evolved into a mature state (Stewart et al., 2010). Principles such as closed-loop properties are well documented and understood. The use of nonlinear models in MPC has motivated the possibility of improving control by refining the quality of the forecasting ability. This has encouraged the use of novel techniques such as implement new ways to approach and solve the optimization problem. Du et al. (2016), proposes a nonlinear MPC controller which controls the vehicle velocity and steering simultaneously. The optimization solver is based on generic algorithms which allow for a more flexible structure for MPC formulation. According to the researcher, the cost function and constraints can be designed in a more accurate meaningful and direct way when using the genetic algorithm. Results from both simulation and on-field test showed that nonlinear MPC developed allowed the vehicle to follow the road centerline accurately and consistently. Moreover, the results also indicate that passenger's safety and comfort are taken into consideration. Using the genetic algorithm as the optimization solver and improve the results of the MPC revealed that the proposed GA based nonlinear MPC can be suitable. It would be interesting to see how a GA based nonlinear MPC can be used for obstacle

avoidance and how much of an effect it can have on the overall results. On the other hand, Cho et al. (2018), presented a different method to avoid obstacles in real-time using an obstacle dependent gaussian potential field (ODG-PF). This method works by first receiving distance data from the range sensors and then creating a Gaussian potential field from objects within the threshold range. The approach used by the researchers differs from the classical potential field-based methods used in other research studies. One major difference is that ODG-PF defines obstacles from range sensor data and then calculates the repulsive fields of the obstacles instead of putting the range sensor data into the equation.

Another difference can be found in the way ODG-PF treat the calculation of the repulsive and attractive fields. The repulsive and attractive fields are functions of the angle and as such are not vectors. This allows the system to be robust and adaptive to small changes in the environment. As a result, the system can avoid both static and moving obstacles without any adjustments. Based on results from simulation, the researchers conclude that vehicle movements using ODG-PF were stable. In the performed moving obstacle scenarios, the vehicle using it did not collide with any of the static or moving obstacles (Cho et al., 2018). Another advantage of the system is that it does not need to perform activities such as image processing or computer vision processing which is considered time consuming. Based on this, the researchers suggest that ODG-PF is relatively light and as a such can be easily implanted in a real-time system. This is an interesting approach used by the researchers in their novel approach to obstacle avoidance. It is important to point out that the testing vehicle used was an unmanned solution ERP42v1 without a braking system. This was an ideal vehicle for the purposes of ODG-PF however, more testing would be required using an actual AV to determine if the same results can be achieved.

## **2.3 MPC Formulation**

As mentioned earlier, an MPC controller is based on an iterative, constrained optimization of the mode. This basically means that at each discrete sampling time the model is sampled and at that

point the actual state is measured or estimated. At this stage the performance of the controller is expressed by the cost function. According to Takács and Rohal'-Ilkiv (2012), the predicted cost function gives a numerical indicator of the quality of control. To obtain the optimal sequence of inputs the cost function must be minimized. This is performed at each sampling interval using a numerical optimization algorithm. This sequence of repeated measure, predict and optimize is known as receding horizon control.

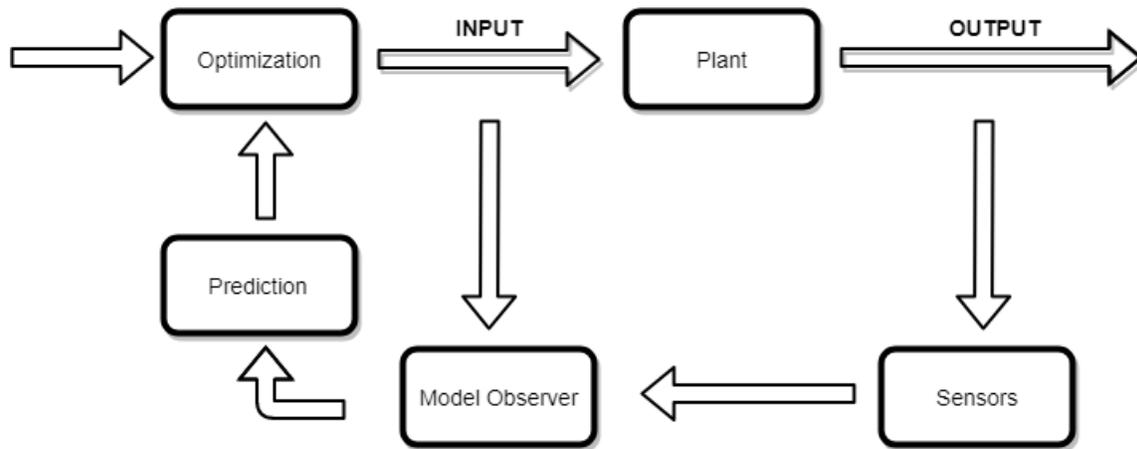


Figure 2.1: Schematic representation of the model predictive control algorithm (Takács and Rohal'-Ilkiv, 2012)

Fig 2.1, describes the typical MPC algorithm. In the diagram an MPC is illustrated as a form of feedback control algorithm where a dynamic optimization process determines inputs based on the measurements. With respect to motion boundary formation for AV, the portion of the road which is seen while driving is represented by the prediction horizon in the MPC model. With a limited amount of the road ahead, the MPC will continuously update the decisions in real time using the latest prediction horizon. According to Takács and Rohal'-Ilkiv (2012), the MPC horizon is receding forward in time. Maintaining a trajectory within a boundary or avoiding obstacles can be considered a constraint the MPC would need to overcome. As such the addition of such constraints can complicate the situation and affect the ability of the MPC to function properly. On the other hand, using nonlinear MPC with boundaries as constraints can allow for successful application of motion boundary formation.

The development of an MPC is generally centered around three main areas:

1. The predictive model
2. The optimal control problem
3. The cost function

### 2.3.1 Prediction Model

The quality of the model largely determines the success of the MPC. If the model is not accurate in its parameters or structure the actual performance of the system will behave incorrectly. A predictive controller iterates the state-space model several steps ahead in time to determine how the system will behave. Adjustments are made to the inputs at the current time accordingly at each sampling instant. If we consider nonlinear continuous-time state-space models, this model class is given by a set of ordinary differential equation.

$$\dot{x}(t) = f(x(t), u(t)) \quad (2.1)$$

With an algebraic output equation

$$y(t) = g(x(t), u(t)) \quad (2.2)$$

Where  $x(t)$  is system state at time  $t$ ,  $u(t)$  is control input at time  $t$  and  $y(t)$  is system output. Both equations are given in the most general form which allows many real-world systems to be described.

The Jacobian matrix of continuous time state space models given by equation Eq 2.1 and Eq 2.2 is useful in the formulation of optimization problem that must be solved. The Jacobians are

defined as:

$$\nabla_x f = \frac{\partial f}{\partial x} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix} \quad (2.3)$$

And

$$\nabla_u f = \frac{\partial f}{\partial u} = \begin{pmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_m} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial u_1} & \cdots & \frac{\partial f_n}{\partial u_m} \end{pmatrix} \quad (2.4)$$

### 2.3.2 Cost Function Description

As mentioned earlier, the importance of the cost function lies in its ability to be an indicator of the degree of optimality of a dynamic model. A cost function needs to be minimized or maximized depending on the intended use. According to Takács and Rohal'-Ilkiv (2012), the degree of optimality can express how close the model performance is to the desired output levels. The most basic form of a cost function is the Time-Optimal cost function in which the system moves from some initial state  $x(t_0)$  to final state  $x(t_f)$  in the minimum amount of time. This generally expressed as:

$$J = \int_{t_0}^{t_f} dt = t_f - t_0 = t^* \quad (2.5)$$

The equation, Eq 2.5 , represents the most basic form of a cost function, however, the cost function can become more complex and be expressed as a quadratic function. in this form the terms of states and control are squared. the structure of the quadratic cost function signifies the type of problem being solved. An optimal solution depends on the values of states, values of control, variables and form of the cost function.

### 2.3.3 Constraint Function Description

The real power of MPC lies in computing optimal control actions for systems, which incorporate constraints. Instead, the optimization procedure must be performed, in between samples. Simplest constraints are linear and have the following form:

$$\begin{aligned} U_{\min} &\leq u(t) \leq U_{\max} \\ X_{\min} &\leq x(t) \leq X_{\max} \end{aligned} \tag{2.6}$$

Where  $U_{\max}$  and  $U_{\min}$ ,  $X_{\max}$  and  $X_{\min}$  are the upper and the lower limits on control input and states respectively.

## 2.4 Solution of the Optimization Problem

According to Guay et al. (2017), there has been a significant development in the area of numerical methods for the online solution of dynamic optimization problems. Early MPC implementations generally made use of sequential quadratic programming solvers. Solving generally involves two elements: the search for the optimal trajectory and the solution to generate the corresponding state trajectory. Carvalho et al. (2013), presents the design of a controller for an autonomous ground vehicle. Their goal was to have a system which tracks the lane centerline while avoiding collisions with obstacles. To achieve this, the researchers relied on a nonlinear MPC framework together with a four-wheel vehicle model. The inputs used from the four-wheel vehicle model were from the front steering angle and the braking torques at each of the four wheels. In their proposed approach, the researchers tested a customized sequential quadratic programming (SQP) algorithm. SQP involves the iterative solution of a convex approximation to the original problem (Carvalho et al., 2013), the nonlinear vehicle dynamics are linearized analytically and convex approximation to the collision avoidance constraints are obtained. The researchers found that the proposed approach allows for the use of longer prediction horizons when compared to those that are used with general nonlinear

solvers. Results from simulation experiments show the ability of the controller to keep the vehicle safe and avoiding multiple stationary and dynamic obstacles.

### 2.4.1 Sequential Quadratic Programming

SQP is an iterative method for solving nonlinearly constrained optimization problems. As with most optimization methods SQP is not a single algorithm but rather a conceptual method from which numerous specific algorithms have evolved. It is expected to find the global optimum for convex problems, however, if the optimization problem is not convex, it converges to a locally optimal point.

Suppose we have a nonlinear optimization problem in a form:

$$\begin{aligned} \min_x \quad & f(x) \\ x \text{ subject to: } \quad & h(x) = 0 \\ & g(x) \geq 0 \end{aligned} \tag{2.7}$$

Where  $x$  is a vector of optimization variables of the objective functions and  $h$  equality constraint function and  $g$  inequality constraint function respectively.

Similar SQP problems can generally be found in the form of nonlinear problems and its applications. Its greatest strength is its ability to solve problems with nonlinear constraints. As such, it is expected that there is at least one constraint function. The basic idea of SQP is to model nonlinear problems at a given approximate solution by a quadratic programming subproblem and then to use the solution to the subproblem to construct a better approximation. This process is treated to create a sequence of approximations that it is hoped will converge to a solution.

## 2.4.2 Linear Quadratic Regulator

The linear quadratic regulator is a well-known method that provides optimally controlled feedback gains and is considered a special case of the generic linear-quadratic problem. One of the main problems with LQR is its inability to deal with constraints (Takács and Rohal'-Ilkiv, 2012). Compared to SQP, LQR optimizes in a fixed time window, whereas SQP, for example, optimizes in a receding time window. As a result, a new solution is computed often however, LQR uses the single optimal solution for the whole-time horizon. According to Takács and Rohal'-Ilkiv (2012), SQP typically solves the optimization problem in smaller time windows than the whole horizon and hence may obtain a suboptimal solution. However, because SQP makes no assumptions about linearity, it can handle hard constraints as well as migration of a nonlinear system away from its linearized operating point, both of which are downsides of LQR.

## 2.5 Vehicle Models

Nonlinear MPC strongly relies on a mathematical model of the vehicle. Having a descriptive model is hence fundamental to ensure good control and estimation of performance. According to Kong et al. (2015), the main components of a modern AV are localization, perception and control. The kinematic and dynamic models are the two general models which are used together with MPC to simulate AV model-based control design. The kinematic model provides a mathematical description of the vehicle motion without consideration for forces that affect motion. The most popular model is known as the bicycle mode Fig (2.2). The major assumptions used in the development of the kinematic model is that the velocity vectors are in the direction of the orientation of the front and rear wheels respectively. According to Rajamani (2005), this is equivalent to assuming that the slip angles at both wheels are zero.

This is acceptable for vehicles traveling at low speeds and is considered a useful model for very low-speed applications such as vehicle control for automated parking (Rajamani, 2005). At higher vehicle speeds the assumption that the velocity at each wheel is in the direction of the wheel can

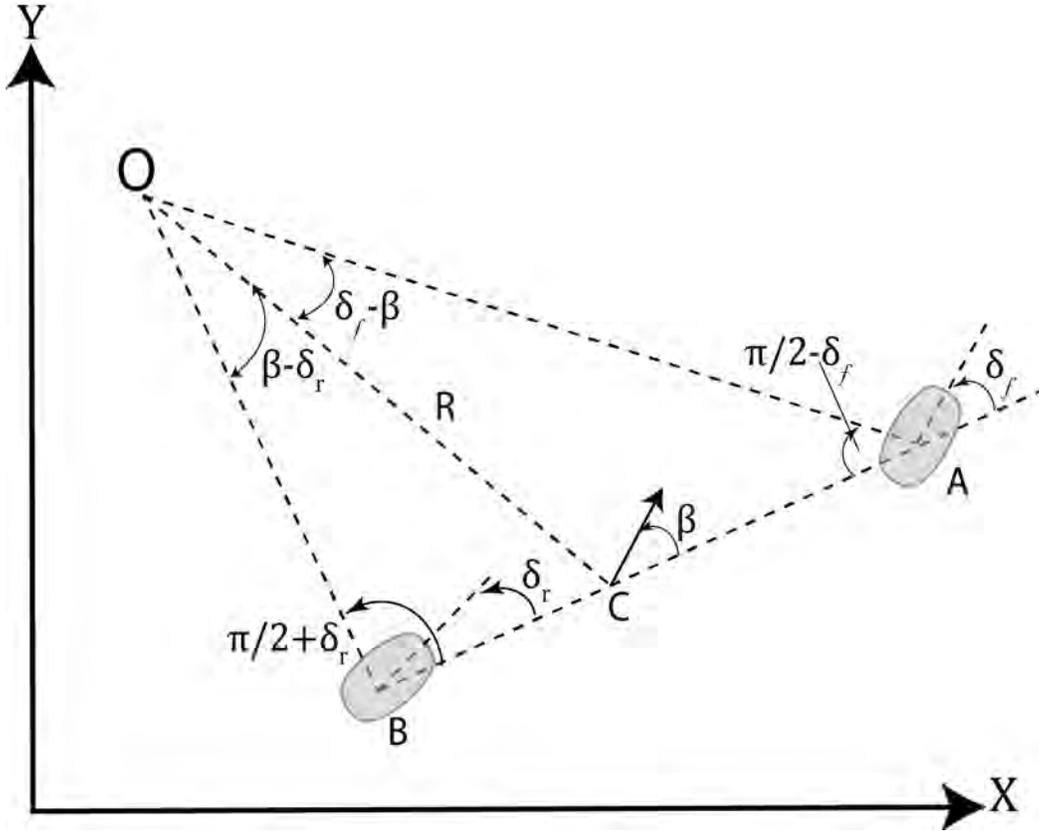


Figure 2.2: Kinematics of lateral vehicle motion (Rajamani, 2005)

no longer be made. As a result, a dynamic model for lateral vehicle motion must be developed. These dynamic models take into consideration the lateral tire forces and slip angles. Using the fundamentals and assumptions of these model's other vehicle models can be derived based on vehicle dynamics.

Kong et al. (2015), study the use of kinematic and dynamic vehicle modes for MPC design used in autonomous driving. According to Kong et al. (2015), published MPC controllers use dynamic vehicle models combined with linear tire models (Carvalho et al., 2013). This is in line with the literature review for this report. Kinematic vehicle models are seldom used. One of the glaring disadvantages of this approach is that it is considered computationally intensive and any tire model becomes singular at low vehicle speeds. This limits its application for stop and go scenarios. The authors propose to address both disadvantages by using the kinematic bicycle model to motivate MPC controller design. The results showed that the kinematic model has better forecast

errors compared to the dynamic model at varying times. Furthermore, experimental tests for the sinusoidal and winding track suggest that the kinematic model can track reference trajectories well at lower speeds. However, the performance dropped significantly at higher speeds where the tire model would be more appropriate.

### 3 METHODOLOGY

In this section, the formulation of the nonlinear MPC is introduced for curvilinear motion planning for AV. When dynamics cannot be accurately modeled by linear models, nonlinear models may be essential to capture the full dynamics of the vehicle model to be controlled. Nonlinear MPC is an advanced control technique that relies on the system model to predict the model trajectory and minimize its deviation from a given reference. The full nonlinearity of the model can be considered by the nonlinear MPC and constraints depending on both states and controls can be easily enforced in the problem formulation. The core strength of MPC is that it is considered a form of optimal control that uses a receding horizon structure.

Nonlinear MPC consists of solving at every time instant the dynamic optimization problem. The problem of estimating the current state given a set of measurements can be formulated as an optimization problem that can consider the full model nonlinearities and gives the opportunity to enforce constraints. At each sampling time step, starting at the current state, an open-loop optimal control problem is solved over a finite prediction horizon. The optimal control input is applied to the system to propagate the system dynamics. At the next time step, a new optimal control problem is solved over a shifted horizon based on the current state of the system. The optimal solution requires a dynamic model of the system, enables input and output constraints to be explicitly included and minimizes a cost function. This allows the vehicle to detect and avoid obstacles by adjusting the trajectory on the fly. A workflow diagram of the computation process implanted in the MPC framework is shown in Fig 3.1. The loop in the diagram corresponds to the iteration of the prediction horizon time. In this project the search for the optimal solution is accelerated using the Taylor series during the linearization of the functions.

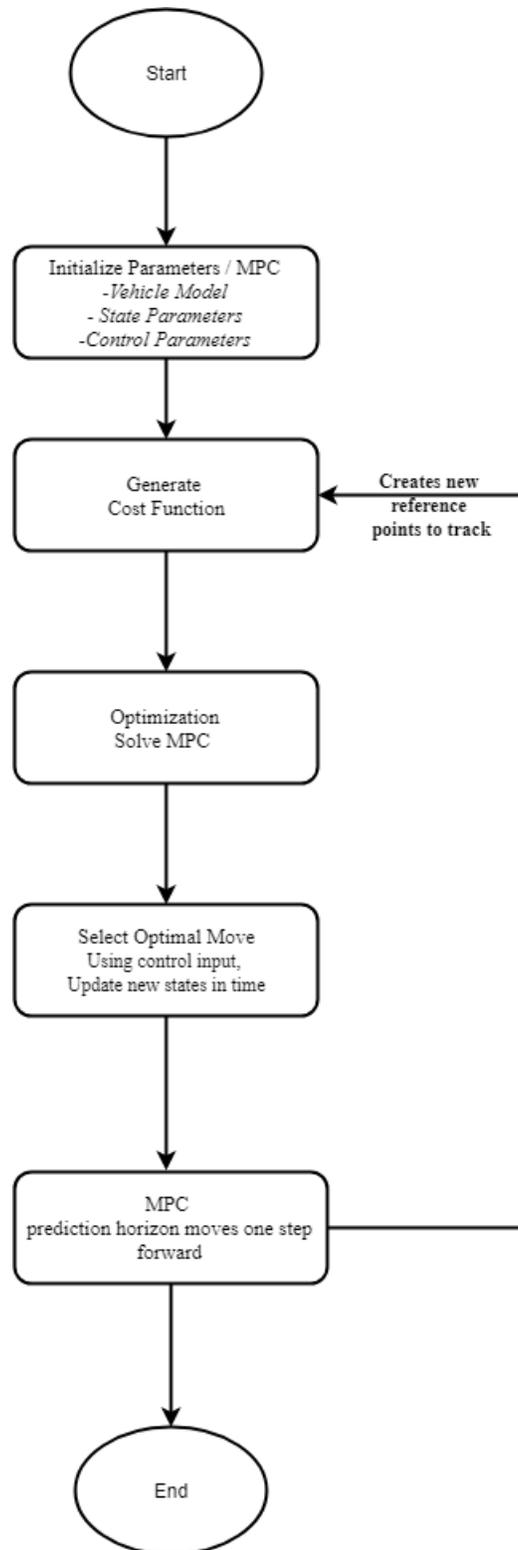


Figure 3.1: Flowchart of nonlinear MPC algorithm as it is used in model simulation

### 3.1 Vehicle Modeling

Nonlinear MPC strongly relies on a mathematical model of the vehicle. Having a descriptive model is hence fundamental to ensure good control and estimation of performance. This section introduces the vehicle models used for the control design. The kinematic models of four (4) vehicles are proposed; (i) single vehicle, (ii) single long truck, (iii) tractor-semitrailer vehicle and (iv) tractor-semitrailer vehicle with semitrailer's steered axle. The kinematic model of vehicle motion is developed to track the motion of the vehicle in the fixed coordinate system with considering the forces causing the motion regardless of forces and torque that causes it. In this project the kinematic bicycle model serves as the basis for the other models. The kinematic bicycle model is a classic model that is exceptional at capturing vehicle motion in normal driving conditions. As a result, it has long been used as a suitable representation for passenger vehicles because of its simplicity and nonholonomic constraints of a car. The bicycle model used in this project is based on the front wheel steering model. This was chosen because the front wheel orientation can be controlled relative to the heading of the vehicle in a 2D plane. The front-wheel represents the front right and left wheels of the vehicle. Similarly, the rear wheel represents the rear right and left wheels of the vehicle.

To analyze the kinematics of the bicycle mode, the reference point was placed at the center of the rear axle. The identification of the reference point is an important step, this is because depending on its placement the resulting kinematic equations are changed respectively. As a result this will influence the MPC formulation. Using the rear axle as the reference point, the heading of the bicycle will be considered to be  $\theta$ , and  $L$  will represent the length between the two wheel axis (Fig 3.2). The velocity is denoted by  $v$  and points in the same direction as each wheel. One of the assumptions made when using this model is the no slip condition. The no slip condition requires that the wheel cannot move laterally or slip longitudinally. As a result  $\omega$ , the rotation rate of the model is equal to the velocity over the instantaneous center of rotation radius  $R$ .

From the similar triangles formed by  $L$  and  $R$ , and  $v$  and  $\phi$  it can be assumed that  $\tan(\phi)$  is equal to  $L$  over the instantaneous turn radius  $R$ . By combining both equations we can find the relation between the rotation rate and the vehicle  $\omega$ , and the steering angle as  $\omega = v \cdot \tan(\theta)/L$  (Eq 3.3). Based on this model configuration, the velocity components of the reference point in the  $x$  and  $y$  direction are equal to the forward velocity  $v \cdot \cos(\theta)$  and  $\sin(\theta)$  respectively. Finally, it is not possible to instantaneously change the steering angle of a vehicle from one extreme of its range to another. The kinematic models can be formulated with four states:  $x$ ,  $\omega$ ,  $\theta$ , and the steering angle  $\phi$ . With that the kinematic bicycle model is now complete.

The kinematic bicycle model takes as inputs the velocity and the steering rate  $\phi$ . The state of the system, including the positions  $X_C$ ,  $Y_C$ , the orientation  $\theta$ , and the steering angle  $\phi$ , evolve according to our kinematic equations from the model, which satisfy the no slip condition. We can now use this model to design kinematic steering MPCs. As mentioned earlier, this serves as the base for the other vehicle models used in this project.

### 3.1.1 Single Vehicle

In the kinematic bicycle model of a biaxial vehicle, it is assumed that the center of rotation is formed by the intersection of the perpendiculars drawn to the planes of wheels' rotation. The dynamics of the model can be seen in (Fig. 3.2). It is assumed that the angular velocity of rotation is relative to the instantaneous center of velocity  $O$ :

$$\omega = \frac{v}{R} \tag{3.1}$$

If we consider that the minimum turning radius  $R$  can be determined by the right triangle with the

vertex  $O$  from the ratio of the steering wheel's rotation angle  $\theta$

$$\tan(\theta) = L/R \tag{3.2}$$

When

$$\omega = v \cdot \tan(\theta)/L \tag{3.3}$$

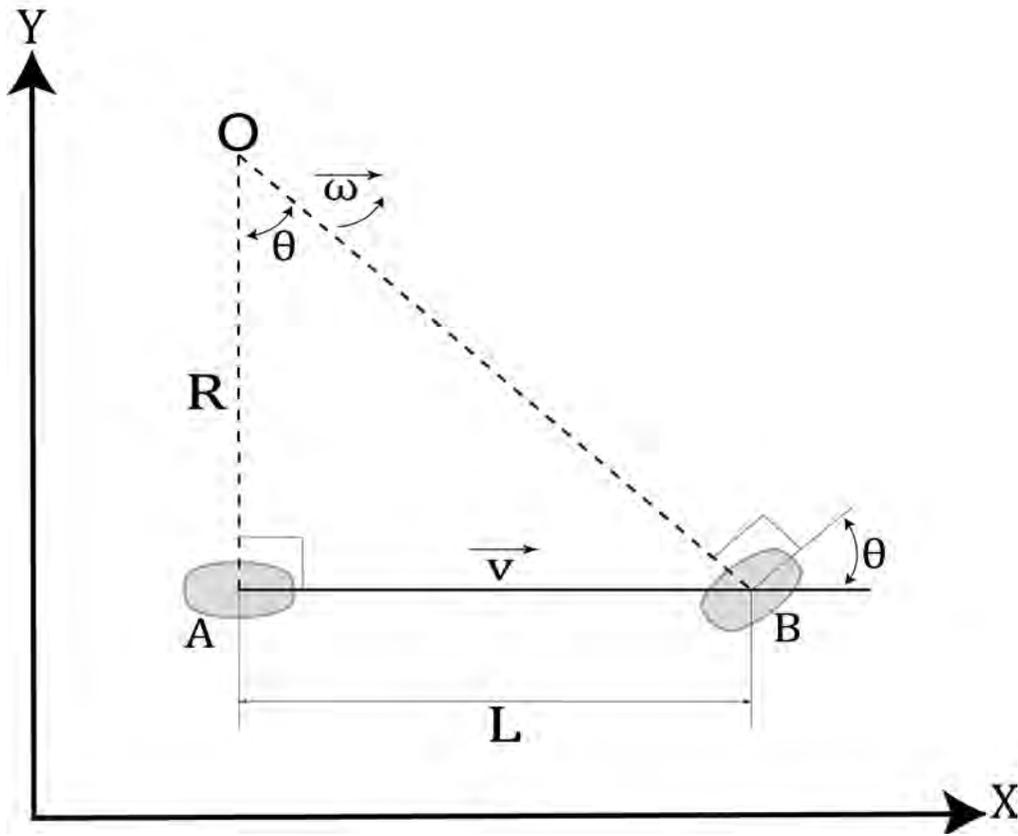


Figure 3.2: Kinematics of curvilinear motion of single passenger vehicle (Kong et al., 2015)

In this method, the bicycle model is linearized at the initial point and is used as the internal model for the MPC controller. To set up this problem we introduce the model state parameters. The model state parameters consist of:  $x$  - vehicle longitudinal displacement,  $y$  - vehicle lateral displacement,

$\phi$  - vehicle yaw angle,  $\theta$  - vehicle's front axle steering angle,  $v$  - vehicle velocity. The derivatives are given by  $v_x$  - vehicle longitudinal velocity along global x-coordinate,  $v_y$  - vehicle lateral velocity along global y-coordinate,  $\omega_\phi$  - vehicle yaw rate. The control parameters used in this model are  $\omega_\theta$  - vehicle's front axle steering rate,  $a$  - vehicle longitudinal acceleration. Thus, the control parameters are longitudinal acceleration and the angular velocity of the steered wheel rotation. Also input the vector of model parameters  $\mathbf{p}$  ( $\mathbf{p} = L$ ), where  $L$  is the vehicle wheelbase.

Then, in the vector form:

$$\mathbf{q} = \begin{pmatrix} x \\ y \\ \phi \\ \theta \\ v \end{pmatrix} \quad \dot{\mathbf{q}} = \begin{pmatrix} v_x \\ v_y \\ \omega_\phi \\ \dot{\theta} \\ \dot{v} \end{pmatrix} \quad \dot{\mathbf{q}} = \mathbf{f}(\mathbf{q}(t), \mathbf{u}(t), \mathbf{p}) \quad \mathbf{u} = \begin{pmatrix} \omega_\theta \\ a \end{pmatrix} \quad \mathbf{f}(\mathbf{q}, \mathbf{u}, \mathbf{p}) = \begin{pmatrix} v \cdot \cos(\phi) \\ v \cdot \sin(\phi) \\ v \cdot \tan(\theta)/L \\ \omega_\theta \\ a \end{pmatrix} \quad (3.4)$$

Reduce the nonlinear function  $\mathbf{f}(\mathbf{q}, \mathbf{u}, \mathbf{p})$  to a more convenient form, separating states and controls:

$$\mathbf{f}(\mathbf{q}(t), \mathbf{u}(t), \mathbf{p}) = \mathbf{f}(\mathbf{q}, \mathbf{u}, \mathbf{p}) = \varphi(\mathbf{q}, \mathbf{p}) + B \cdot \mathbf{u} \quad (3.5)$$

Where

$$\varphi(\mathbf{q}, \mathbf{p}) = v \cdot \begin{pmatrix} \cos(\phi) \\ \sin(\phi) \\ \tan(\theta)/L \\ 0 \\ 0 \end{pmatrix}, B = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (3.6)$$

Thus

$$\dot{\mathbf{q}} = \varphi(\mathbf{q}, \mathbf{p}) + B \cdot \mathbf{u} \quad (3.7)$$

Or

$$\frac{d}{dt} \begin{pmatrix} x \\ y \\ \phi \\ \theta \\ v \end{pmatrix} = v \cdot \begin{pmatrix} \cos(\phi) \\ \sin(\phi) \\ \tan(\theta)/L \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \omega_\theta \\ a \end{pmatrix} \quad (3.8)$$

Jacobian Matrix: To speed up the search for the optimal solution, as well as for the possibility of using adaptive MPC, consider the linearization of (Eq. 3.7) through the expansion in a Taylor

series with the first linear terms in the vicinity of point  $O$ . In vector form

$$\begin{aligned} \dot{\mathbf{q}}_0(t) + \Delta \dot{\mathbf{q}}(t) &= f(\mathbf{q}_0(t), \mathbf{u}_0(t), \mathbf{p}) + \\ &+ \left. \frac{\partial f(\mathbf{q}(t), \mathbf{u}(t), \mathbf{p})}{\partial \mathbf{q}} \right|_{\substack{\mathbf{q}_0 \\ \mathbf{u}_0}} \cdot \Delta \mathbf{q}(t) + \left. \frac{\partial f(\mathbf{q}(t), \mathbf{u}(t), \mathbf{p})}{\partial \mathbf{u}} \right|_{\substack{\mathbf{q}_0 \\ \mathbf{u}_0}} \cdot \Delta \mathbf{u}(t) + O^2 \end{aligned} \quad (3.9)$$

Given that

$$\dot{\mathbf{q}}_0(t) = \mathbf{f}(\mathbf{q}_0(t), \mathbf{u}_0(t), \mathbf{p}) \quad (3.10)$$

Obtain,

$$\Delta \dot{\mathbf{q}}(t) = \left. \frac{\partial \mathbf{f}(\mathbf{q}(t), \mathbf{u}(t), \mathbf{p})}{\partial \mathbf{q}} \right|_{\substack{\mathbf{q}_0 \\ \mathbf{u}_0}} \cdot \Delta \mathbf{q}(t) + \left. \frac{\partial \mathbf{f}(\mathbf{q}(t), \mathbf{u}(t), \mathbf{p})}{\partial \mathbf{u}} \right|_{\substack{\mathbf{q}_0 \\ \mathbf{u}_0}} \cdot \Delta \mathbf{u}(t) \quad (3.11)$$

Where

$$A(t) = \left. \frac{\partial \mathbf{f}(\mathbf{q}(t), \mathbf{u}(t), \mathbf{p})}{\partial \mathbf{q}} \right|_{\substack{\mathbf{q}_0 \\ \mathbf{u}_0}} \quad B(t) = \left. \frac{\partial \mathbf{f}(\mathbf{q}(t), \mathbf{u}(t), \mathbf{p})}{\partial \mathbf{u}} \right|_{\substack{\mathbf{q}_0 \\ \mathbf{u}_0}} \quad (3.12)$$

Then the linearized equation in increments

$$\Delta \dot{\mathbf{q}} = A \cdot \Delta \mathbf{q} + B \cdot \Delta \mathbf{u} \quad (3.13)$$

Matrix A is Jacobian, which is calculated as:

$$J = \begin{pmatrix} \frac{\partial \mathbf{f}_1}{\partial q_1} & \cdots & \frac{\partial \mathbf{f}_1}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{f}_n}{\partial q_1} & \cdots & \frac{\partial \mathbf{f}_n}{\partial q_n} \end{pmatrix} \quad (3.14)$$

Substituting Eq.(3.4) in Eq.(3.14), yield

$$A = \begin{pmatrix} 0 & 0 & -v \cdot \sin(\phi) & 0 & \cos(\phi) \\ 0 & 0 & v \cdot \cos(\phi) & 0 & \sin(\phi) \\ 0 & 0 & 0 & v \cdot (\tan^2(\theta) + 1)/L & \tan(\theta)/L \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}; B = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (3.15)$$

### 3.1.2 Single Long Truck

Unlike the kinematic bicycle model of a biaxial vehicle (Fig.3.2), it is assumed that the minimum turning radius of the vehicle with the additional rear steered axle is floating and depends on the ratio of the rotation angles of the front and rear axles' steered wheels (Fig.3.3). The angular velocity of rotation relative to the instantaneous center of rotation  $O$  is determined according to Eq.(3.1).

Furthermore, in this model, the minimum turning radius  $R$  can be determined in two ways via

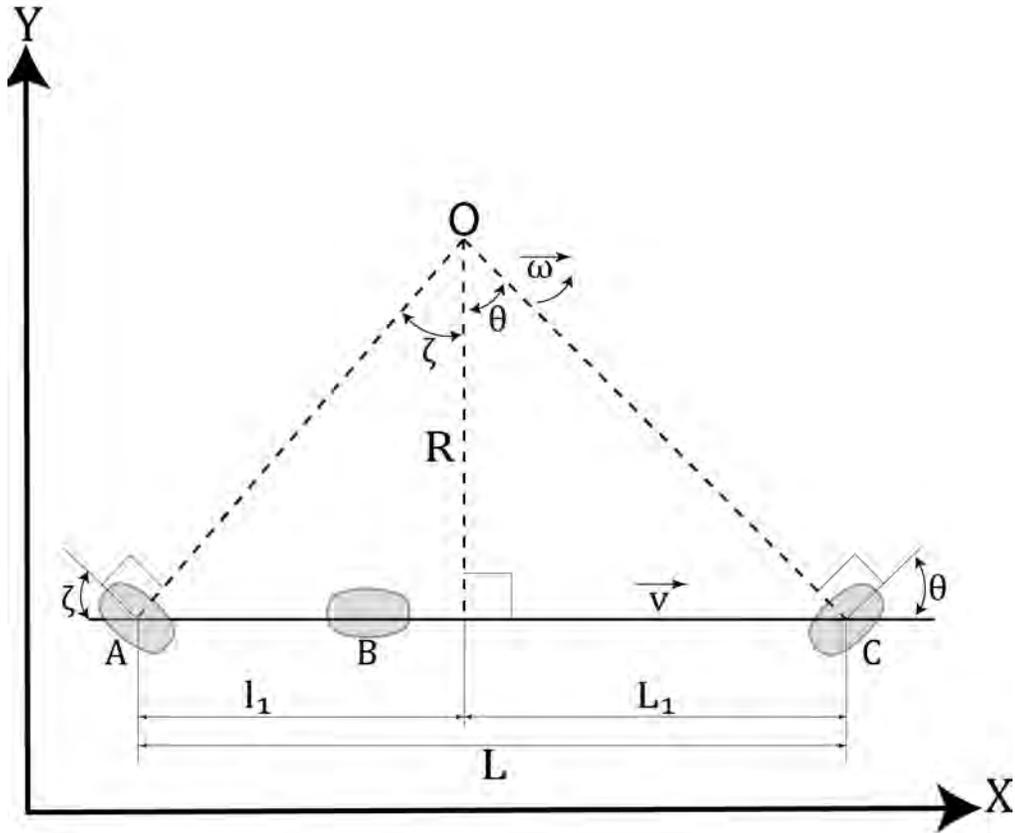


Figure 3.3: Kinematics of curvilinear motion of single truck with steered rear axle (Espinosa et al., 1998)

right angle triangles with vertex  $O$ . From the ratio of the steering angle  $\theta$  of the front steered wheel:

$$\tan(\theta) = L_1/R \quad (3.16)$$

From the ratio of the steering angle  $\zeta$  of the rear steered wheel:

$$\tan(\zeta) = l_1/R \quad (3.17)$$

Then

$$L_1 = R \cdot \tan(\theta)$$

$$l_1 = R \cdot \tan(\zeta) \quad (3.18)$$

$$L_1 - l_1 = R \cdot (\tan(\theta) - \tan(\zeta))$$

Given that the  $l_1$  coordinate is negative relative to the R radius entry point

$$L_1 - l_1 = L_1 - (-|l_1|) = L_1 + l_1 = L \quad (3.19)$$

Thus,

$$R = L / (\tan(\theta) - \tan(\zeta)) \quad (3.20)$$

The expression for the angular velocity can be written as:

$$\omega = \frac{v}{R} = \frac{v \cdot (\tan(\theta) - \tan(\zeta))}{L} \quad (3.21)$$

Similarly to the previous model, we can now introduce the state parameters, derivatives and control parameters. The state parameters are defined by,  $x$  - vehicle longitudinal displacement,  $y$  - vehicle lateral displacement,  $\phi$  - vehicle yaw angle,  $\theta$  - vehicle's front axle steering angle,  $\zeta$  -

vehicle's rear axle steering angle,  $v$  - vehicle velocity.

The derivatives are defined by,  $v_x$  - vehicle longitudinal velocity along global x-coordinate,  $v_y$  - vehicle lateral velocity along global y-coordinate,  $\omega_\theta$  - vehicle yaw rate.

Also, the control parameters are defined by,  $\omega_\theta$  - vehicle's front axle steering rate,  $\omega_\zeta$  - vehicle's rear axle steering rate,  $a$  - vehicle longitudinal acceleration. Thus, the control parameters are longitudinal acceleration and the angular rotation velocities of the steered wheels of the front and rear axles. Vector of model parameters  $\mathbf{p}$  ( $\mathbf{p} = L$ ), where  $L$  is the vehicle wheelbase. Then, in vector form:

$$\mathbf{q} = \begin{pmatrix} x \\ y \\ \phi \\ \theta \\ \zeta \\ v \end{pmatrix} \quad \dot{\mathbf{q}} = \begin{pmatrix} v_x \\ v_y \\ \omega_\phi \\ \dot{\theta} \\ \dot{\zeta} \\ \dot{v} \end{pmatrix} \quad \mathbf{u} = \begin{pmatrix} \omega_\theta \\ \omega_\zeta \\ a \end{pmatrix} \quad \dot{\mathbf{q}} = \mathbf{f}(\mathbf{q}(t), \mathbf{u}(t), \mathbf{p}) \quad (3.22)$$

$$\mathbf{f}(\mathbf{q}(t), \mathbf{u}(t), \mathbf{p}) = \mathbf{f}(\mathbf{q}, \mathbf{u}, \mathbf{p}) = \varphi(\mathbf{q}, \mathbf{p}) + B \cdot \mathbf{u} \quad (3.23)$$

Where

$$\varphi(\mathbf{q}, \mathbf{p}) = v \cdot \begin{pmatrix} \cos(\phi) \\ \sin(\phi) \\ (\tan(\theta) - \tan(\zeta))/L \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.24)$$

Thus,

$$\dot{\mathbf{q}} = \varphi(\mathbf{q}, \mathbf{p}) + B \cdot \mathbf{u} \quad (3.25)$$

Or

$$\frac{d}{dt} \begin{pmatrix} x \\ y \\ \phi \\ \theta \\ \zeta \\ v \end{pmatrix} = v \cdot \begin{pmatrix} \cos(\phi) \\ \sin(\phi) \\ (\tan(\theta) - \tan(\zeta))/L \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \omega_\theta \\ \omega_\zeta \\ a \end{pmatrix} \quad (3.26)$$

Jacobian. Similarly to Eqs.(3.9-3.14), get the linearization of the equation (3.26). Denote,

$$c_\phi = \cos(\phi) \quad s_\phi = \sin(\phi) \quad t_\theta = \tan(\theta) \quad t_\zeta = \tan(\zeta) \quad (3.27)$$

Then a  $6 \times 6$  matrix A has the following nonzero elements:

$$\begin{aligned} A_{1,3} &= -v \cdot s_\phi, \quad A_{2,3} = v \cdot c_\phi, \\ A_{3,4} &= v \cdot (t_\theta^2 + 1)/L, \quad A_{3,5} = -v \cdot (t_\zeta^2 + 1)/L, \\ A_{1,6} &= c_\phi, \quad A_{2,6} = s_\phi, \quad A_{3,6} = (t_\theta - t_\zeta)/L \end{aligned} \quad (3.28)$$

\*Note: Matrix B remains without changes.

### 3.1.3 Tractor-Semitrailer Vehicle

Now, consider the kinematic bicycle model of a two-unit vehicle. The rotation center is assumed to be formed by the intersection of perpendiculars drawn to the rotational planes of the wheels (Fig. 3.2). In this case (Fig. 3.4), the angular velocity of tractor's rotation relative to the instantaneous center of velocities O will be  $\omega_1$ , and the angular velocity of the semitrailer  $\omega_2$ :

$$\omega_1 = \frac{v_1}{R_1}, \quad \omega_2 = \frac{v_2}{R_2} \quad (3.29)$$

At this stage we can introduce the model state parameters, derivative parameters and control parameters. The state parameters are given by, x - vehicle longitudinal displacement, y - vehicle lateral displacement,  $\phi$  - vehicle yaw angle,  $\psi$  - vehicle articulation angle,  $\theta$  - vehicle's front axle steering angle, v - vehicle velocity. The derivatives are given by,  $v_x$  - vehicle longitudinal velocity



Considering that  $\psi = \phi_1 \phi_2$ ,  $d\psi/dt = \omega_1 \omega_2$ , in vector format:

$$\mathbf{q} = \begin{pmatrix} x \\ y \\ \phi \\ \psi \\ \theta \\ v \end{pmatrix} \quad \mathbf{p} = \begin{pmatrix} L_1 \\ e_1 \\ L_2 \end{pmatrix} \quad \dot{\mathbf{q}} = \begin{pmatrix} v_x \\ v_y \\ \omega_\phi \\ \dot{\psi} \\ \dot{\theta} \\ \dot{v} \end{pmatrix} \quad \mathbf{u} = \begin{pmatrix} \omega_\theta \\ a \end{pmatrix} \quad \dot{\mathbf{q}} = \mathbf{f}(\mathbf{q}(t), \mathbf{u}(t), \mathbf{p}) \quad (3.30)$$

The angular velocity of the leading unit (tractor) is determined similarly to Eq.(3.3) replacing L with  $L_1$ . For the semitrailer position it could be written:

$$\tan(\psi) = \frac{L_2 - e_1/\cos(\psi)}{R_2} \quad (3.31)$$

Then,

$$\omega_2 = \frac{v_2}{R_2} = \frac{v_2 \cdot \tan(\psi)}{L_2 - e_1/\cos(\psi)} = \frac{v_1 \cdot \cos(\psi) \cdot \tan(\psi)}{L_2 - e_1/\cos(\psi)} = \frac{v \cdot \sin(\psi)}{L_2 - e_1/\cos(\psi)} \quad (3.32)$$

As a result, similarly to Eq.(3.23),

$$\varphi(\mathbf{q}, \mathbf{p}) = v \cdot \begin{pmatrix} \cos(\phi) \\ \sin(\phi) \\ \tan(\theta)/L_1 \\ \tan(\theta)/L_1 - \frac{\sin(\psi)}{L_2 - e_1/\cos(\psi)} \\ 0 \\ 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (3.33)$$

Or

$$\frac{d}{dt} \begin{pmatrix} x \\ y \\ \phi \\ \psi \\ \theta \\ v \end{pmatrix} = v \cdot \begin{pmatrix} \cos(\phi) \\ \sin(\phi) \\ \tan(\theta)/L_1 \\ \tan(\theta)/L_1 - \frac{\sin(\psi)}{L_2 - e_1/\cos(\psi)} \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \omega_\theta \\ a \end{pmatrix} \quad (3.34)$$

Jacobian. Similarly to Eqs.(3.9-3.14), get the linearization of the equation (3.26). Denote,

$$c_\phi = \cos(\phi), \quad s_\phi = \sin(\phi), \quad c_\psi = \cos(\psi), \quad s_\psi = \sin(\psi), \quad t_\theta = \tan(\theta) \quad (3.35)$$

Then, the matrix A of 6×6 dimension has following nonzero elements:

$$\begin{aligned}
 A_{1,3} &= -v \cdot s_\phi, & A_{2,3} &= v \cdot c_\phi \\
 A_{4,4} &= -v \cdot \left( \frac{c_\psi}{L_2 - e_1/c_\psi} + \frac{e_1 \cdot s_\psi^2}{c_\psi^2 \cdot (L_2 - e_1/c_\psi)^2} \right) \\
 A_{3,5} &= A_{4,5} = v \cdot (t_\theta^2 + 1)/L_1 \\
 A_{1,6} &= c_\phi, & A_{2,6} &= s_\phi, & A_{3,6} &= t_\theta/L_1 \\
 A_{4,6} &= t_\theta/L_1 - s_\psi/(L_2 - e_1/c_\psi)
 \end{aligned} \tag{3.36}$$

\*Note: Matrix B remains without changes.

### 3.1.4 Tractor-Semitrailer Vehicle with Semitrailer's Steered Axle

This model is based on the previous one with the additional state parameter  $\zeta$  - semitrailer's middle axle steering angle, and additional control parameter  $\omega_\zeta$  - semitrailer's middle axle steering rate (Fig.3.5).

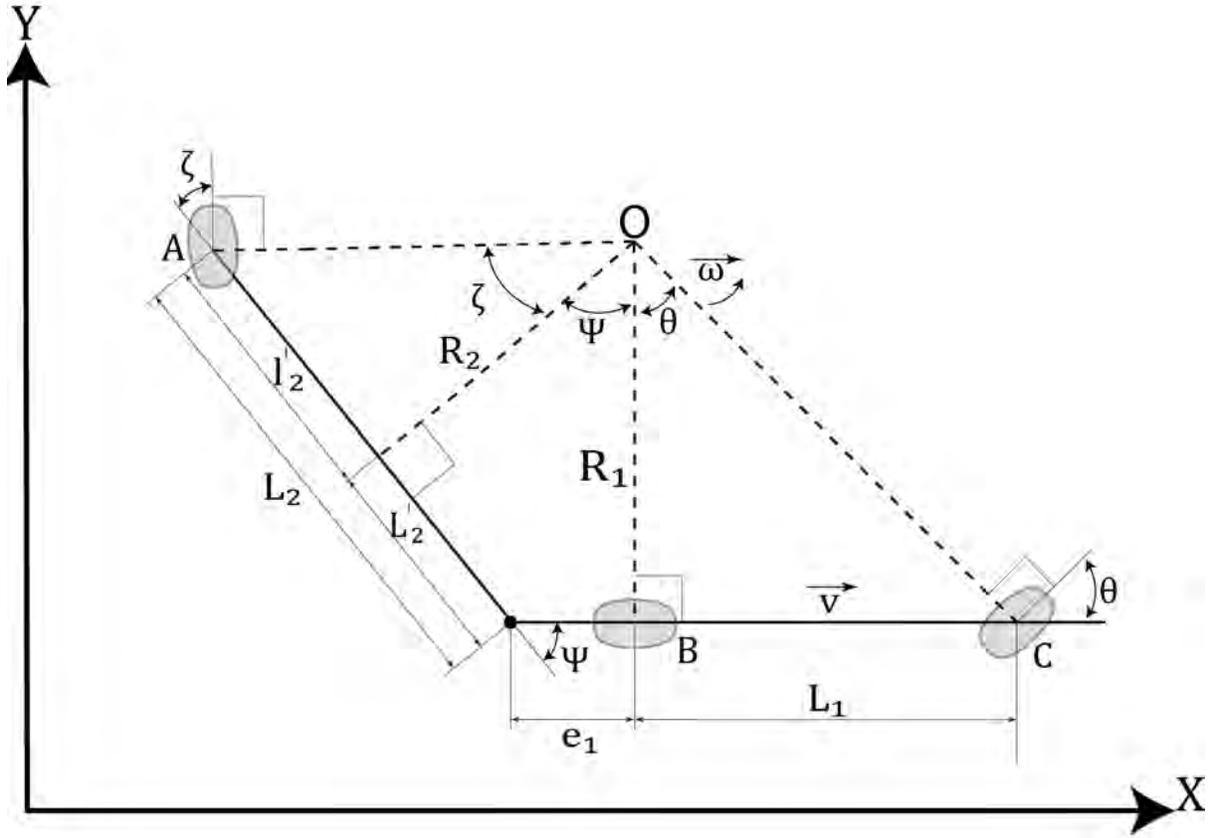


Figure 3.5: Kinematics of curvilinear motion of an articulated vehicle with semitrailer's steered axle (Ritzen et al., 2016, Werner et al., 2012)

State-space components:

$$\mathbf{q} = \begin{pmatrix} x \\ y \\ \phi \\ \psi \\ \theta \\ \zeta \\ v \end{pmatrix} \quad \mathbf{p} = \begin{pmatrix} L_1 \\ e_1 \\ L_2 \end{pmatrix} \quad \dot{\mathbf{q}} = \begin{pmatrix} v_x \\ v_y \\ \omega_\phi \\ \dot{\psi} \\ \dot{\theta} \\ \dot{\zeta} \\ \dot{v} \end{pmatrix} \quad \mathbf{u} = \begin{pmatrix} \omega_\theta \\ \omega_\zeta \\ a \end{pmatrix} \quad \dot{\mathbf{q}} = \mathbf{f}(\mathbf{q}(t), \mathbf{u}(t), \mathbf{p}) = \varphi(\mathbf{q}, \mathbf{p}) + \mathbf{B} \cdot \mathbf{u} \quad (3.37)$$

The radius  $R_2$  may be determined from two conditions, considering Eq.(3.31):

$$\begin{aligned} l'_2 &= R_2 \cdot \tan(\zeta) \\ L'_2 - e_1/\cos(\psi) &= R_2 \cdot \tan(\psi) \end{aligned} \quad (3.38)$$

Then

$$L'_2 - l'_2 - e_1/\cos(\psi) = R_2 \cdot (\tan(\psi) - \tan(\zeta)) \quad (3.39)$$

Considering the coordinate  $l'_2$  is negative relative to a cross point of radius  $R_2$ :

$$L'_2 - l'_2 = L'_2 - (-|l'_2|) = L'_2 + l'_2 = L_2 \quad (3.40)$$

Thus,

$$R_2 = \frac{L_2 - e_1/\cos(\psi)}{\tan(\psi) - \tan(\zeta)} \quad (3.41)$$

The expression for angular velocity  $\omega_2$  may be derived in a view:

$$\omega_2 = \frac{u_2}{R_2} = \frac{v_1 \cdot \cos(\psi)}{R_2} = \frac{v \cdot \cos(\psi) \cdot (\tan(\psi) - \tan(\zeta))}{L_2 - e/\cos(\psi)} = \frac{v \cdot \sin(\psi - \zeta)}{(L_2 - e/\cos(\psi)) \cdot \cos(\zeta)} \quad (3.42)$$

As a result, similarly to Eq.(3.33),

$$\varphi(\mathbf{q}, \mathbf{p}) = v \cdot \begin{pmatrix} \cos(\phi) \\ \sin(\phi) \\ \tan(\theta)/L_1 \\ \tan(\theta)/L_1 - \frac{\cos(\psi) \cdot (\tan(\psi) - \tan(\zeta))}{L_2 - e/\cos(\psi)} \\ 0 \\ 0 \\ 0 \end{pmatrix} B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.43)$$

Or

$$\frac{d}{dt} \begin{pmatrix} x \\ y \\ \phi \\ \psi \\ \theta \\ \zeta \\ v \end{pmatrix} = v \cdot \begin{pmatrix} \cos(\phi) \\ \sin(\phi) \\ \tan(\theta)/L_1 \\ \tan(\theta)/L_1 - \frac{\cos(\psi) \cdot (\tan(\psi) - \tan(\zeta))}{L_2 - e/\cos(\psi)} \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \omega_\theta \\ \omega_\zeta \\ a \end{pmatrix} \quad (3.44)$$

Jacobian. Similarly to Eqs.(3.9-3.14), get the linearization of the equation (3.26). Denote,

$$c_\phi = \cos(\phi), s_\phi = \sin(\phi), c_\psi = \cos(\psi), s_\psi = \sin(\psi), t_\theta = \tan(\theta), t_\psi = \tan(\psi), t_\zeta = \tan(\zeta) \quad (3.45)$$

Then, the matrix A of  $7 \times 7$  dimension has following nonzero elements:

$$\begin{aligned} A_{1,3} &= -v \cdot s_\phi, & A_{2,3} &= v \cdot c_\phi \\ A_{4,4} &= -v \cdot \frac{\frac{L_2 \cdot \cos(3 \cdot \psi - \zeta)}{4} + e_1 \cdot \cos(2 \cdot \psi - \zeta) + \frac{L_2 \cdot \cos(\psi + \zeta)}{4} + \frac{L_2 \cdot \cos(\psi - \zeta)}{2}}{c_\zeta \cdot (e_1 - L_2 \cdot c_\psi)^2} \\ A_{3,5} &= A_{4,5} = v \cdot (t_\theta^2 + 1) / L_1 \\ A_{4,6} &= \frac{v \cdot c_\psi^2}{c_\zeta^2 \cdot (L_2 \cdot c_\psi - e_1)} \\ A_{1,7} &= c_\phi, & A_{2,7} &= s_\phi, & A_{3,7} &= t_\theta / L_1 \\ A_{4,7} &= \frac{t_\theta}{L_1} - \frac{c_\psi \cdot (t_\psi - t_\zeta)}{L_2 - e_1 / c_\psi} \end{aligned} \quad (3.46)$$

\*Note: Matrix B remains without changes.

## 3.2 Optimization

Proceed from minimizing the functional to predict vehicle control,

$$\min_u J(u) = \int_0^{t_f} (y^T \cdot W_y \cdot y + u^T \cdot W_u \cdot u + \Delta u^T \cdot W_u \cdot \Delta u) \cdot dt \quad (3.47)$$

Subject to:

$$\dot{\mathbf{q}} = \mathbf{f}(\mathbf{q}(t), \mathbf{u}(t), \mathbf{p}), \quad t \in [t_0, t_f] \quad (3.48)$$

The system of constraints is written as

$$\begin{cases} y_{j,\min}(t) - \varepsilon \cdot h_{j,\min}^{(y)}(t) \leq y_j(t) \leq y_{j,\max}(t) + \varepsilon \cdot h_{j,\max}^{(y)}(t), & t \in [t_0, t_f] \\ u_{j,\min}(t) - \varepsilon \cdot h_{j,\min}^{(u)}(t) \leq u_j(t) \leq u_{j,\max}(t) + \varepsilon \cdot h_{j,\max}^{(u)}(t), & t \in [t_0, t_f] \\ \Delta u_{j,\min}(t) - \varepsilon \cdot h_{j,\min}^{(\Delta u)}(t) \leq \Delta u_j(t) \leq \Delta u_{j,\max}(t) + \varepsilon \cdot h_{j,\max}^{(\Delta u)}(t), & t \in [t_0, t_f] \end{cases} \quad (3.49)$$

where  $y_{j,\min}(t)$ ,  $y_{j,\max}(t)$  = minimum and maximum values of  $j$ th output at the  $i$ th prediction horizon step, respectively;  $\Delta u_{j,\min}(t)$ ,  $\Delta u_{j,\max}(t)$  = minimum and maximum values of  $j$ th input at the  $i$ th prediction horizon step, respectively;  $\Delta u_{j,\min}(t)$ ,  $\Delta u_{j,\max}(t)$  = minimum and maximum values of  $j$ th input rate at the  $i$ th prediction horizon step, respectively;  $h_{j,\min}^{(y)}(t)$ ,  $h_{j,\max}^{(y)}(t)$  = minimum and maximum values of  $j$ th output's hard constraints at the  $i$ th prediction horizon step, respectively;  $h_{j,\min}^{(u)}(t)$ ,  $h_{j,\max}^{(u)}(t)$  = minimum and maximum values of  $j$ th input's hard constraints at the  $i$ th prediction horizon step, respectively;  $h_{j,\min}^{(\Delta u)}(t)$ ,  $h_{j,\max}^{(\Delta u)}(t)$  = minimum and maximum values of  $j$ th input rates' hard constraints at the  $i$ th prediction horizon step, respectively;  $n_y$  = number of output parameters;  $n_u$  = number of input parameters;  $n_{\Delta u}$  = number of input rate parameters.

The functional Eq. (3.46) in discrete form has a view:

$$\min_u J(z_k) = \sum_{i=1}^{p-1} \left( y_i^T \cdot W_y \cdot y_i + u_i^T \cdot W_u \cdot u_i + \Delta u_i^T \cdot W_{\Delta u} \cdot \Delta u_i \right) \quad (3.50)$$

where  $y$  - vector of output parameters;  $W_y$ ,  $W_u$ ,  $W_{\Delta u}$  - weighting factors;  $y_i$  - output signals at the  $i$ th prediction horizon step;  $z_p = \left( u^T_0, u^T_{i+1}, \dots, u^T_{p-1}, \varepsilon_p \right)$   $u_i$  - control signals at the  $i$ th prediction horizon step; - solution;  $\varepsilon_p$  - scalar dimensionless slack variable used for constraint softening;  $p_\varepsilon$  - constraint violation penalty weight;  $i$  - current control interval;  $p$  - prediction horizon (number of intervals).

The system of constraints is written as

$$\begin{cases} y_{j,\min(i)} - \varepsilon \cdot h_{j,\min(i)}^{(y)} \leq y_{j,i} \leq y_{j,\max(i)} + \varepsilon \cdot h_{j,\max(i)}^{(y)}, & i = 1 \dots p, \quad j = 1 \dots n_y \\ u_{j,\min(i)} - \varepsilon \cdot h_{j,\min(i)}^{(u)} \leq u_{j,i-1} \leq u_{j,\max(i)} + \varepsilon \cdot h_{j,\max(i)}^{(u)}, & i = 1 \dots p, \quad j = 1 \dots n_u \\ \Delta u_{j,\min(i)} - \varepsilon \cdot h_{j,\min(i)}^{(\Delta u)} \leq \Delta u_{j,i-1} \leq \Delta u_{j,\max(i)} + \varepsilon \cdot h_{j,\max(i)}^{(\Delta u)}, & i = 1 \dots p, \quad j = 1 \dots n_{\Delta u} \end{cases} \quad (3.51)$$

where  $y_{j,\min(i)}$ ,  $y_{j,\max(i)}$  = minimum and maximum values of  $j$ th output at the  $i$ th prediction horizon step, respectively;  $u_{j,\min(i)}$ ,  $u_{j,\max(i)}$  = minimum and maximum values of  $j$ th input at the  $i$ th prediction horizon step, respectively;  $\Delta u_{j,\min(i)}$ ,  $\Delta u_{j,\max(i)}$  = minimum and maximum values of  $j$ th input rate at the  $i$ th prediction horizon step, respectively;  $h_{j,\min(i)}^{(y)}$ ,  $h_{j,\max(i)}^{(y)}$  = minimum and maximum values of  $j$ th output's hard constraints at the  $i$ th prediction horizon step, respectively;  $h_{j,\min(i)}^{(u)}$ ,  $h_{j,\max(i)}^{(u)}$  = minimum and maximum values of  $j$ th input's hard constraints at the  $i$ th prediction horizon step, respectively;  $h_{j,\min(i)}^{(\Delta u)}$ ,  $h_{j,\max(i)}^{(\Delta u)}$  = minimum and maximum values of  $j$ th input rates' hard constraints at the  $i$ th prediction horizon step, respectively;  $n_y$  = number of output parameters;  $n_u$  = number of input parameters;  $n_{\Delta u}$  = number of input rate parameters.

For a single passenger car and for a conventional tractor-semitrailer  $y = \theta$ , for a single truck and for an articulated vehicle with steered semitrailer's steered axles  $y = (\theta, \zeta)^T$ .

Using the state-space model and above parameters, MATLAB is used to program and simulate the vehicle model designs.

# 4 SIMULATION RESULTS

## 4.1 Simulation Environment

This section presents the results obtained from simulations performed using MATLAB/SIMULINK software package. MATLAB was selected for nonlinear MPC implementation and simulation because of the SIMULINK interface. The combined use of MATLAB/SIMULINK allows for text and graphic programming to design the MPC system in a simulation environment. The model predictive control toolbox provides functionality for designing and simulating MPCs. The kinematic vehicle models developed in the previous section are tested to demonstrate its performance. Finally, tests are presented to demonstrate how the models tolerate variation in vehicle parameters.

## 4.2 Motion Boundary Simulation

### 4.2.1 Scenario 1: Circle Motion

In this study, the idea of constraints is that the trajectories of  $k$  given points of the vehicle contour must lay within the considered boundaries. At each point, there is a distance by a radius  $r_{pk}$  from the mass center and compose an angle  $\alpha_{pk}$  with the longitudinal axis of the vehicle local coordinate system. Then, in the global coordinate system, the coordinates of the points will be  $x_{pki}$ ,  $y_{pki}$  for the  $i$ th prediction horizon step.

$$\begin{aligned}x_{p_{k,i}} &= x_i + r_{p_k} \cdot \cos(\phi_i + \alpha_{p_k}) \\y_{p_{k,i}} &= y_i + r_{p_k} \cdot \sin(\phi_i + \alpha_{p_k})\end{aligned}\tag{4.1}$$

Correspondingly, radii of controlling points

$$r_{k,i} = \sqrt{x_{p_{k,i}}^2 + y_{p_{k,i}}^2} \quad (4.2)$$

Then, the condition of nonlinear restrictions is

$$r \leq r_{k,i} \leq R \quad (4.3)$$

Where  $r$  and  $R$  – inner and outer radii, respectively.

The first simulation ran was the **Passenger Car**. The following parameters were used to define the model:

Table 4.1: Control parameters for passenger car vehicle model

Parameter	Value
L	2.8 m
R	20.0 m
H	3.75 m
Lane Width ( $r$ )	16.25 m
$\beta$	45°
Prediction Horizon Time ( $t_f$ )	10 s

Table 4.2: State and constraint parameters for passenger car vehicle model

Name	Parameter	Value
Initial State	$q_0$	$(0, -(R+r)/2, 0, \arctan(2 \cdot L/(R+r)), 0)^T$
Final State	$q_f$	$((R+r)/2 \cdot \cos(\beta), ((R+r)/2 \cdot \sin(\beta), (\pi/2+\beta), \arctan(2 \cdot L/(R+r)), 0)^T$
State Restrictions	$\theta$	$-0.7 \leq \theta \leq 0.7$
	Velocity	$-10 \leq v \leq 10$
Control Restrictions	$\omega_\theta$	$-0.5 \leq \omega_\theta \leq 0.5$
	Acceleration	$-2 \leq a \leq 2.5$

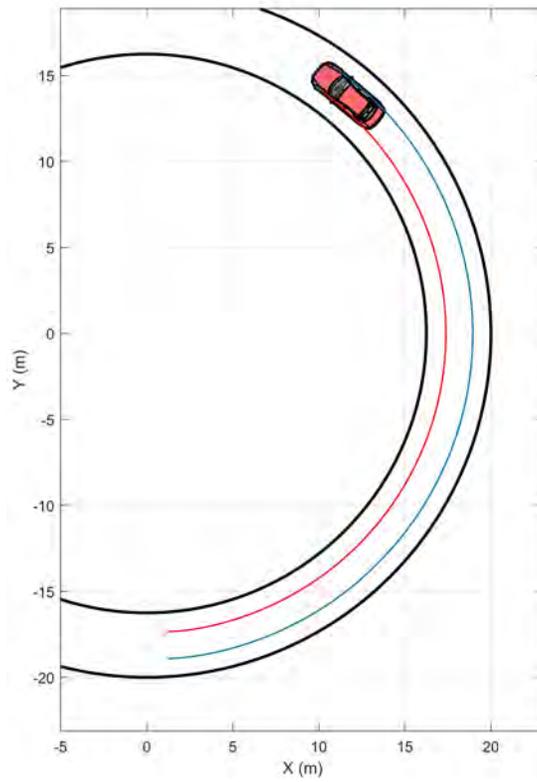


Figure 4.1: Predicted trajectory for a passenger car in autonomous mode at the roundabout arc

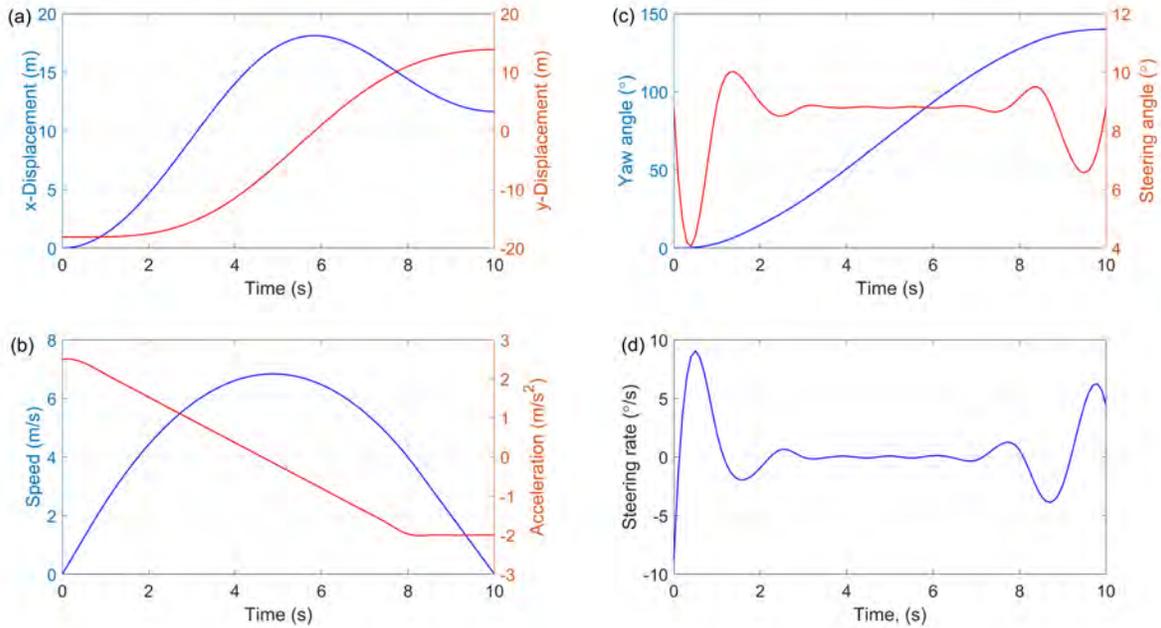


Figure 4.2: Results of states and controls for passenger car in autonomous mode at the roundabout arc:  
 (a) - mass center displacement, (b) - speed/acceleration, (c) - yaw angle vs. steering angle, (d) - steering rate control

The trajectory obtained by applying the proposed control scheme to this scenario is displayed in Fig 4.1 , where the controller is able to avoid the boundaries accurately. The model correctly maintains the same distance away from the boundary as it navigates along the path. In Fig 4.2, if we compare graph (c), and graph (d) we can see smooth curves between the steering angle and steering rate. This suggests that the system is fluid and smooth while avoiding the boundary. After the start of the simulation, the MPC slowly corrects the inaccuracy and find the optimal solution to maintain avoiding the lane boundary, this is shown by the rate of change of the Yaw angle Fig 4.2 graph (c).

The second model simulated was the **Single Truck**. The following constraint parameters were used to test the model:

Table 4.3: Control parameters for single truck vehicle model

Parameter	Value
$L_1$	6.65 m
$l_1$	1.4 m
R	30.0 m
H	6.0 m
Lane Width (r)	24.0 m
$\beta$	45°
Prediction Horizon Time ( $t_f$ )	10 s

Table 4.4: State and constraint parameters for single truck vehicle model

Name	Parameter	Value
Initial State	$q_0$	$(0, -(R+r)/2, 0, \arctan(2 \cdot L_1/(R+r)), -\arctan(2 \cdot l_1/(R+r)), 0)^T$
Final State	$q_f$	$((R+r)/2 \cdot \cos(\beta), (R+r)/2 \cdot \sin(\beta), (\pi/2+\beta), \arctan(2L_1/(R+r)), -\arctan(2 \cdot l_1/(R+r)), 0)^T$
State Restrictions	$\theta$	$-0.7 \leq \theta \leq 0.7$
	$\zeta$	$-0.5 \leq \zeta \leq 0.5$
	Velocity	$-10 \leq v \leq 10$
Control Restrictions	$\omega_\theta$	$-0.5 \leq \omega_\theta \leq 0.5$
	$\omega_\zeta$	$-0.5 \leq \omega_\zeta \leq 0.5$
	Acceleration	$-2 \leq a \leq 2.5$

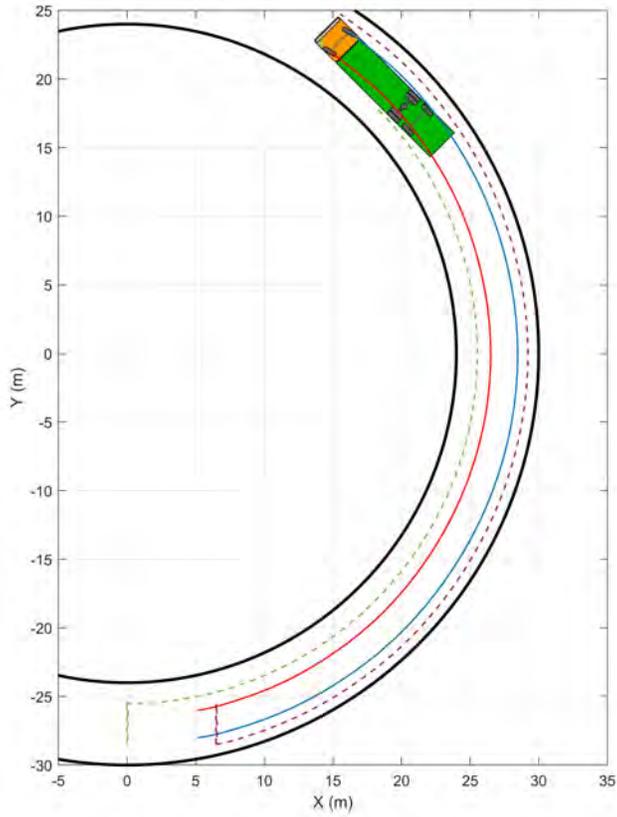


Figure 4.3: Predicted trajectory for a single truck in autonomous mode at the roundabout arc

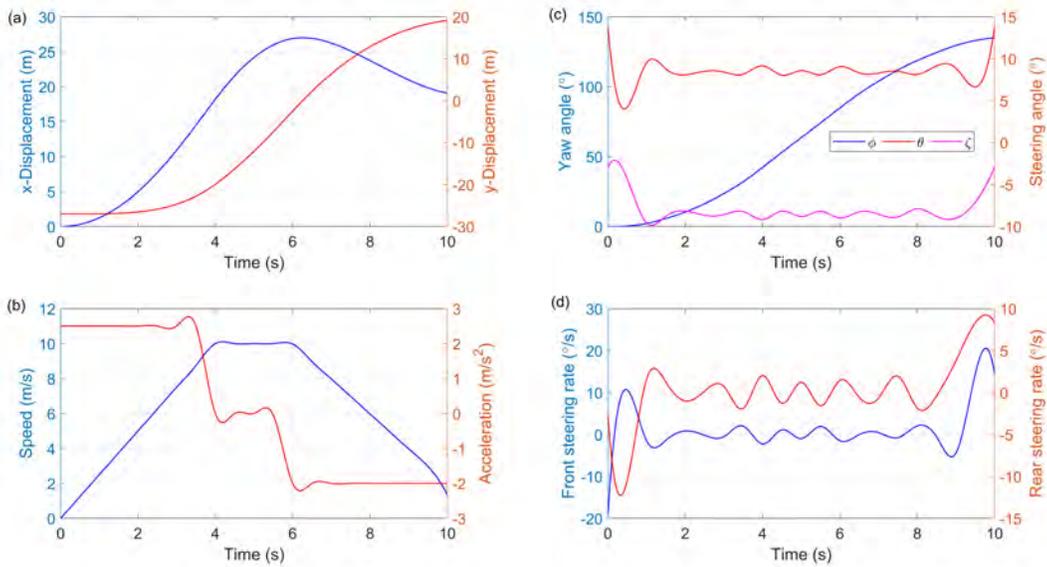


Figure 4.4: Results of states and controls for a single truck in autonomous mode at the roundabout arc:  
 (a) - mass center displacement, (b) - speed/acceleration, (c) - yaw angle vs. steering angle, (d) - steering rate control

Similarly to the results obtained from the passenger car simulation, the single truck results show that the model was successful in avoiding the boundary constraint. Looking at the front steering rate and rear steering rate (Fig 4.4 graph (d)), we can see that they begin inverse of one another. This is expected because it is assumed that the minimum turning radius of the vehicle with the additional rear steered axle is floating and depends on the ratio of the rotation angles of the front and rear axles' steered wheels.

The third model simulated was the **Conventional Tractor-Semitrailer**. The following constraint parameters were used to test the model:

Table 4.5: Control parameters for conventional tractor-semitrailer vehicle model

Parameter	Value
$L_1$	3.8 m
$L_2$	7.57 m
$e_{tr}$	0.47 m
$R$	30.0 m
$H$	6.0 m
Lane Width ( $r$ )	24.0 m
$\beta$	45°
Prediction Horizon Time ( $t_f$ )	10 s

Table 4.6: State and constraint parameters for conventional tractor-semitrailer vehicle model

Name	Parameter	Value
Initial State	$q_0$	$(0, -(R+r)/2, 0, \arctan(2L_1/(R+r)), -\arctan(2 \cdot l_1/(R+r)), 0)^T$
Final State	$q_f$	$((R+r)/2 \cdot \cos(\beta), ((R+r)/2 \cdot \sin(\beta), (\pi/2+\beta), \arctan(2L_1/(R+r)), -\arctan(2 \cdot l_1/(R+r)), 0)^T$
State Restrictions	$\theta$	$-0.7 \leq \theta \leq 0.7$
	$\psi$	$-\pi/2 \leq \psi \leq \pi/2$
	Velocity	$-10 \leq v \leq 10$
Control Restrictions	$\omega_\theta$	$-0.5 \leq \omega_\theta \leq 0.5$
	Acceleration	$-2 \leq a \leq 2.5$

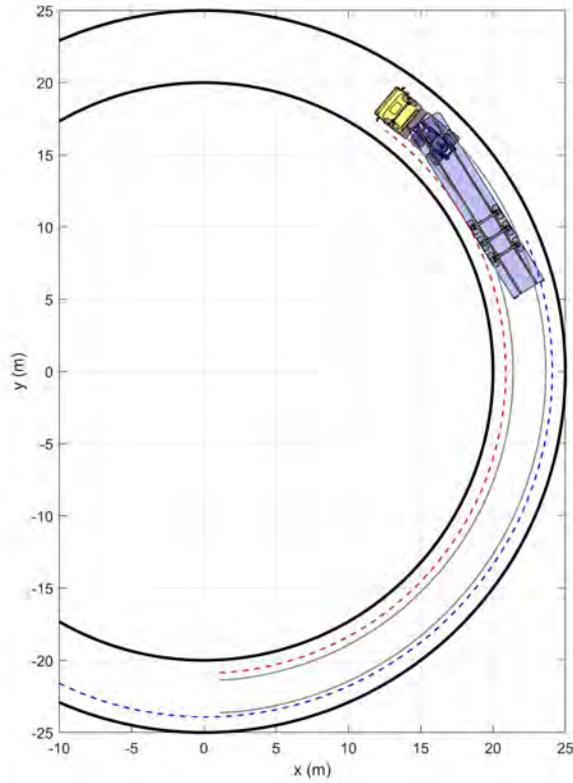


Figure 4.5: Predicted trajectory for a tractor-semitrailer in autonomous mode at the roundabout arc

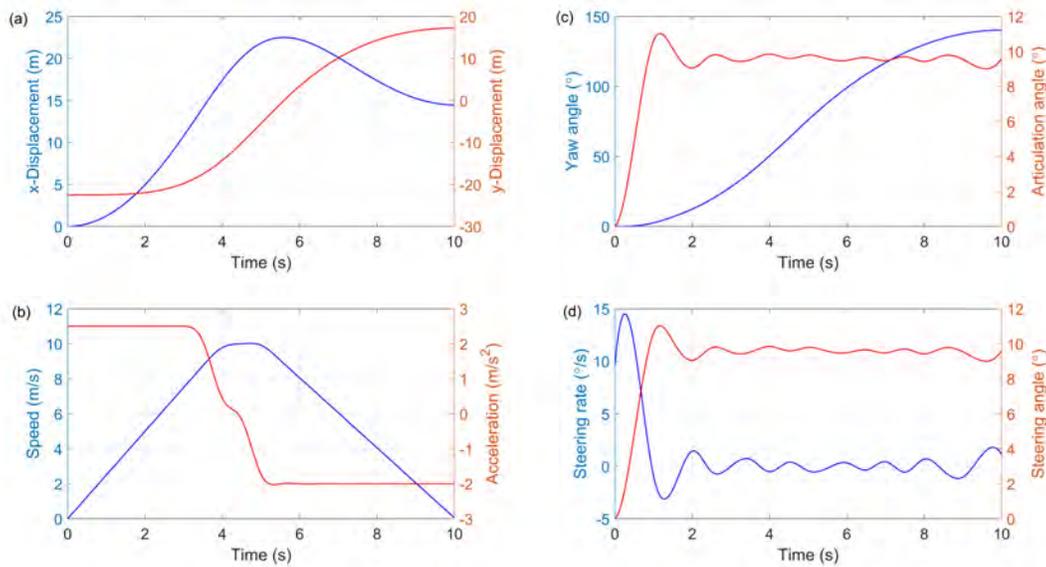


Figure 4.6: Results of states and controls for a tractor-semitrailer in autonomous mode at the roundabout arc: (a) - mass center displacement, (b) - speed/acceleration, (c) - yaw angle vs. folding angle, (d) - steering rate controls

The tractor semitrailer vehicle is based on a two-unit vehicle model. Considerations had to be made for the angular velocity of the tractor's rotation as well as the angular velocity of the semitrailer. The results suggest that the model was successful. The tractor semi-tractor was able to avoid colliding with the boundary and maintain its trajectory. Fig 4.6 graph (d) shows that the vehicle model was able to complete its trajectory smoothly considering both the angular velocity of the tractors rotation and the angular velocity of the semitrailer.

Finally, the last model simulated was the **Articulated vehicle with semitrailer steered axle**. The following constraint parameters were used to test the model:

Table 4.7: Control parameters for articulated vehicle with semitrailer steered axle model

Parameter	Value
$L_1$	3.8 m
$L_2$	7.57 m
$e_{tr}$	0.47 m
R	25.0 m
H	4.5 m
Lane Width (r)	21.5 m
$\beta$	45°
Prediction Horizon Time ( $t_f$ )	10 s

Table 4.8: State and constraint parameters for articulated vehicle with semitrailer steered axle model

Name	Parameter	Value
Initial State	$q_0$	$(0, -(R+r)/2, 0, 0.35, \arctan(2L_1/(R+r)), -\arctan(2L_2/(R+r)), 0)^T$
Final State	$q_f$	$((R+r)/2 \cdot \cos(\beta), ((R+r)/2 \cdot \sin(\beta), (\pi/2+\beta), 0.35, \arctan(2L_1/(R+r)), -\arctan(2L_2/(R+r)), 0)^T$
State Restrictions	$\theta$	$-0.7 \leq \theta \leq 0.7$
	$\psi$	$-\pi/2 \leq \psi \leq \pi/2$
	$\zeta$	$-0.61 \leq \zeta \leq 0.61$
	Velocity	$-10 \leq v \leq 10$
Control Restrictions	$\omega_\theta$	$-0.5 \leq \omega_\theta \leq 0.5$
	$\omega_\zeta$	$-0.61 \leq \omega_\zeta \leq 0.61$
	Acceleration	$-2 \leq a \leq 2.5$

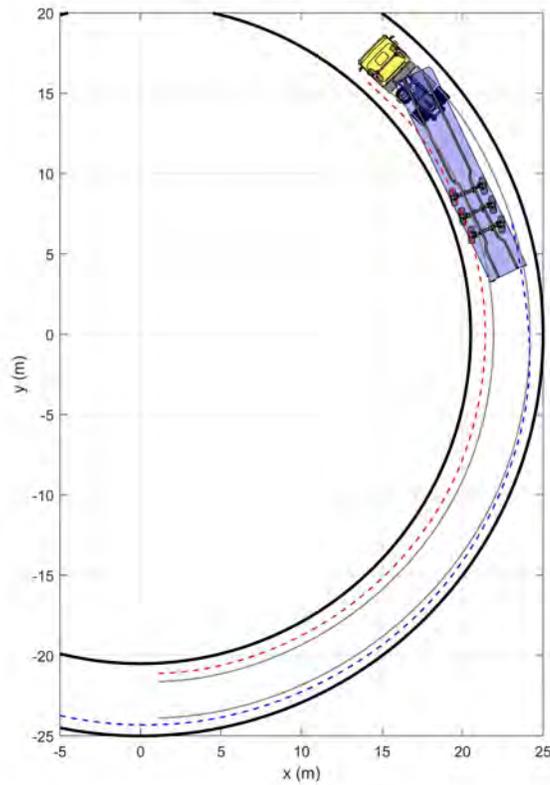


Figure 4.7: Predicted trajectory for an articulated vehicle with steered semitrailer's axle in autonomous mode at the roundabout arc

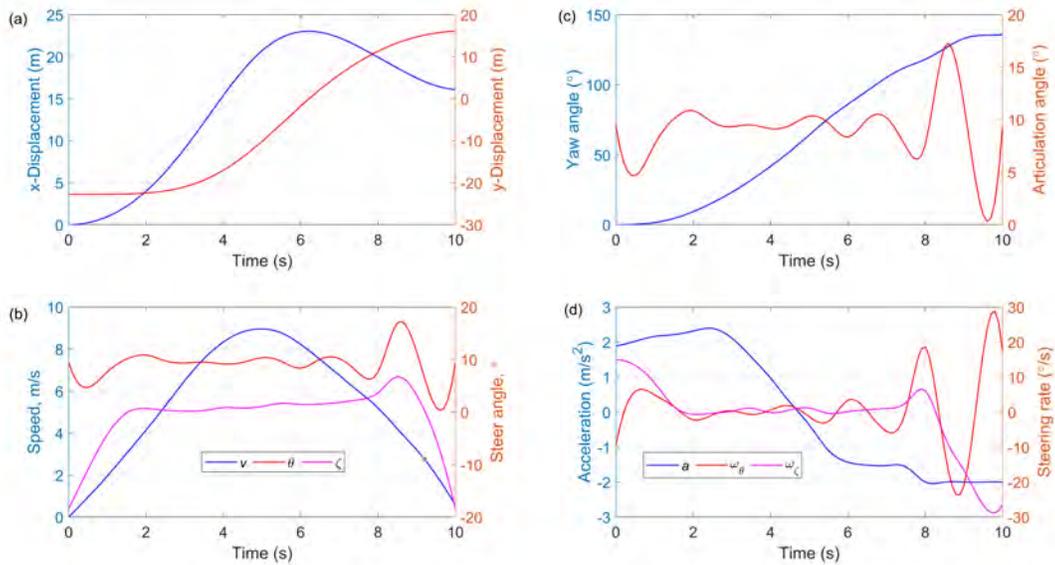


Figure 4.8: Results of states and controls for an articulated vehicle with steered semitrailer's axle in autonomous mode at the roundabout arc: (a) - mass center displacement, (b) - speed and steering angles, (c) - yaw angle vs. folding angle, (d) - acceleration / steering rate controls

The vehicle model used here is similar to the tractor-semitrailer previously mentioned with the addition of the middle angle steering rate. This model was successful in simulating curvilinear motion planning by avoiding the boundaries as it navigates along the path. Fig 4.8 graph (d) shows that the steering rate of the middle angle and the tractors steering angle are similar. This suggests that the model follows a smooth trajectory.

### 4.2.2 Scenario 2: No Space Constraints

As mentioned earlier, the second objective of this paper was to test the NMPC precision, adequacy and its sensitivity to changes in the vehicle state parameters. To accomplish this, the trajectory of motion is predicted in the absence of space restrictions, the initial and final values of the state vector are specified. The expectation of this simulation is to move and park the vehicle by determining the optimal path.

The first simulation ran was the **Passenger Car**. The following constraint parameters were used to test the model:

Table 4.9: No space state and constraint parameters for passenger car vehicle model

Name	Parameter	Value
	L	2.8 m
Prediction Horizon Time	$t_f$	10 s
Initial State	$q_0$	$(0, 0, 0, 0, 0, 0)^T$
Final State	$q_f$	$(-25, 0, \pi/2, 0, 0)^T$
State Restrictions	$\theta$	$-0.7 \leq \theta \leq 0.7$
	$\zeta$	$-0.5 \leq \zeta \leq 0.5$
	Velocity	$-10 \leq v \leq 10$
Control Restrictions	$\omega_\theta$	$-0.5 \leq \omega_\theta \leq 0.5$
	Acceleration	$-2 \leq a \leq 2.5$

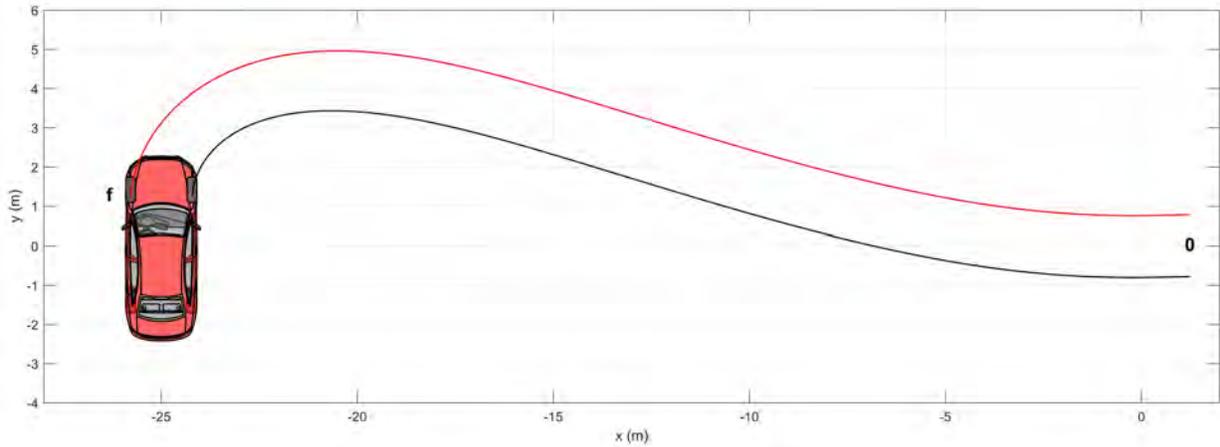


Figure 4.9: Predicted trajectory for a passenger car in autonomous mode for parking place changing

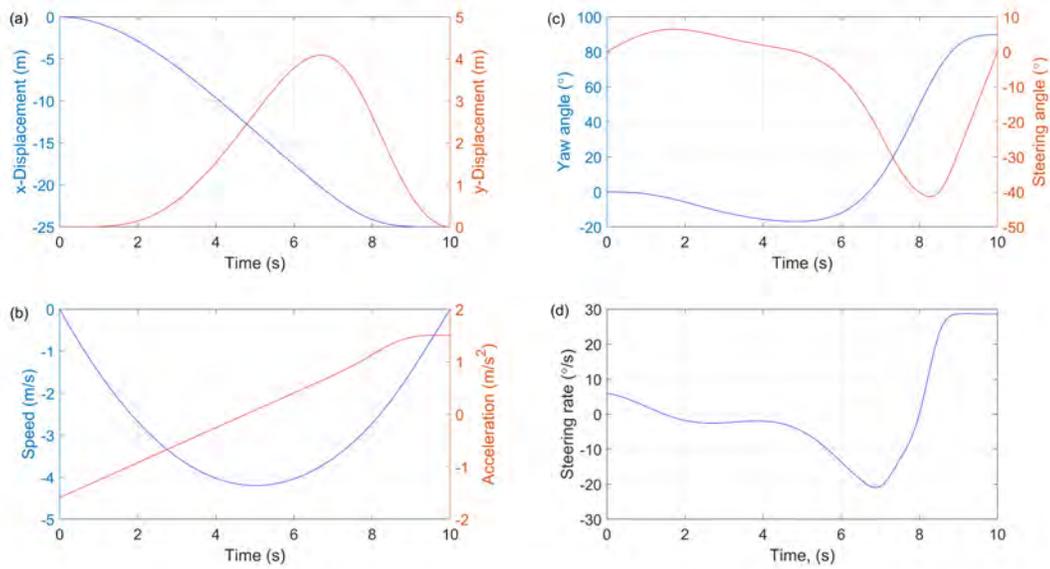


Figure 4.10: Results of states and controls for passenger car in autonomous mode for parking place changing: (a) - mass center displacement, (b) - speed/acceleration, (c) - yaw angle vs. steering angle, (d) - steering rate control

The second model simulated was the **Single Truck**. The following constraint parameters were used to test the model:

Table 4.10: No space state and constraint parameters for single truck vehicle model

Name	Parameter	Value
Prediction Horizon Time	$L_1$	6.65 m
	$l_1$	1.4 m
	$t_f$	10 s
Initial State	$q_0$	$(0, 0, 0, 0, 0, 0)^T$
Final State	$q_f$	$(-35, 10, \pi/2, 0, 0, 0)^T$
State Restrictions	$\theta$	$-0.7 \leq \theta \leq 0.7$
	$\zeta$	$-0.5 \leq \zeta \leq 0.5$
	Velocity	$-10 \leq v \leq 10$
Control Restrictions	$\omega_\theta$	$-0.5 \leq \omega_\theta \leq 0.5$
	$\omega_\zeta$	$-0.5 \leq \omega_\zeta \leq 0.5$
	Acceleration	$-2 \leq a \leq 2.5$

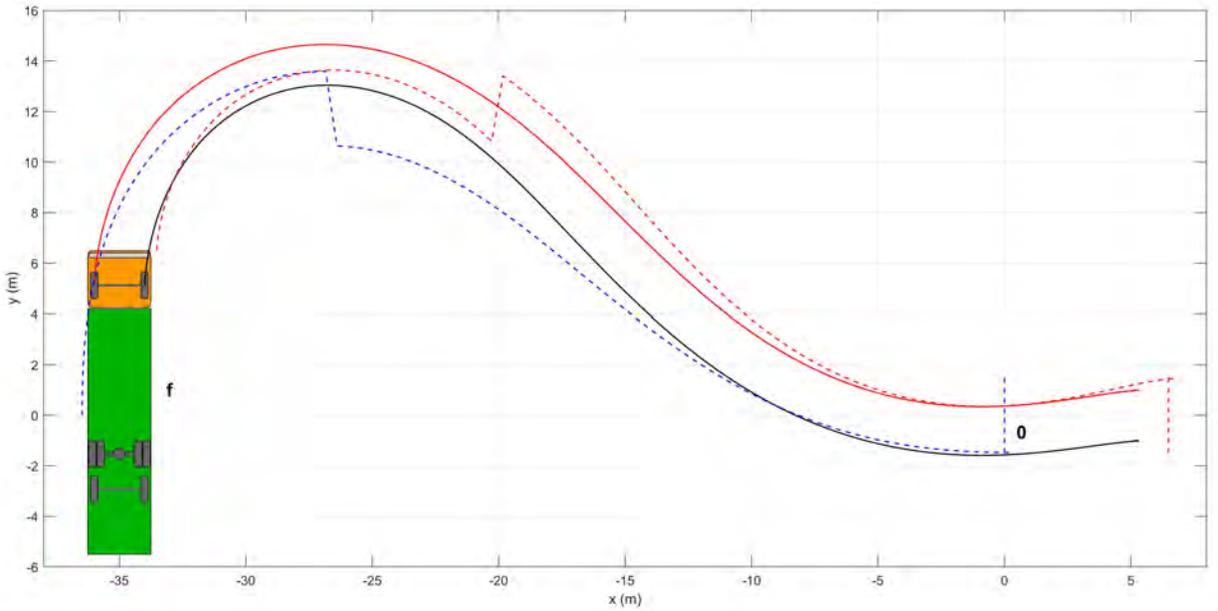


Figure 4.11: Predicted trajectory for a single truck in autonomous mode for parking place changing

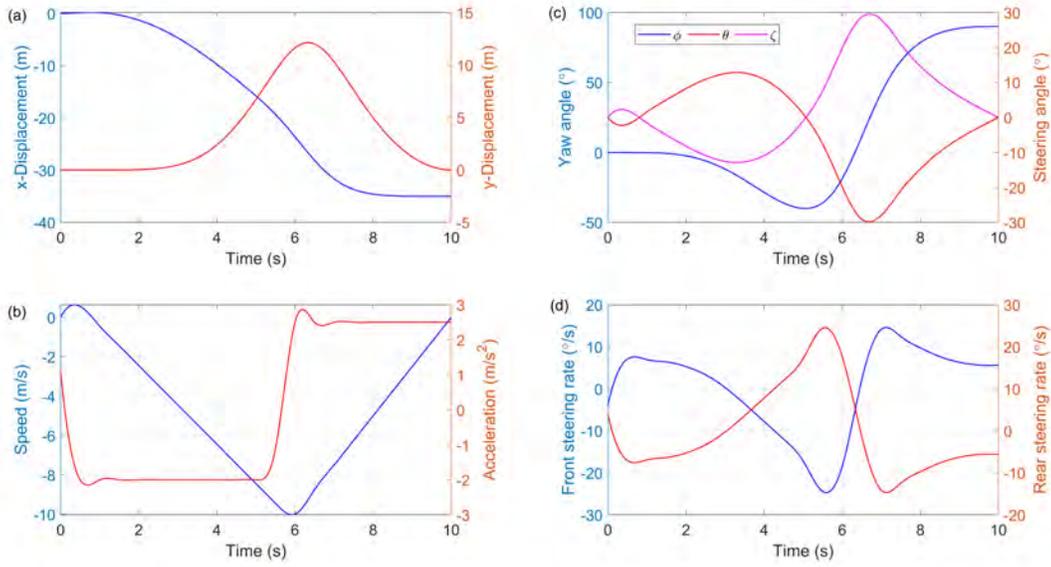


Figure 4.12: Results of states and controls for a single truck in autonomous mode for parking place changing: (a) - mass center displacement, (b) - yaw angle vs. steering angles, (c) - speed/acceleration, (d) - steering rate controls

The third model simulated was the **Conventional Tractor-Semitrailer**. The following constraint parameters were used to test the model:

Table 4.11: No space state and constraint parameters for conventional tractor-semitrailer vehicle model

Name	Parameter	Value
	$L_1$	3.8 m
	$L_2$	7.57 m
	$e_{tr}$	0.47 m
Prediction Horizon Time	$t_f$	10 s
Initial State	$q_0$	$(0, 0, 0, 0, 0, 0)^T$
Final State	$q_f$	$(-35, 10, \pi/2, 0, 0, 0)^T$
State Restrictions	$\theta$	$-0.7 \leq \theta \leq 0.7$
	$\psi$	$-\pi/2 \leq \psi \leq \pi/2$
	Velocity	$-10 \leq v \leq 10$
Control Restrictions	$\omega_\theta$	$-0.6 \leq \omega_\theta \leq 0.6$
	Acceleration	$-2 \leq a \leq 2.5$

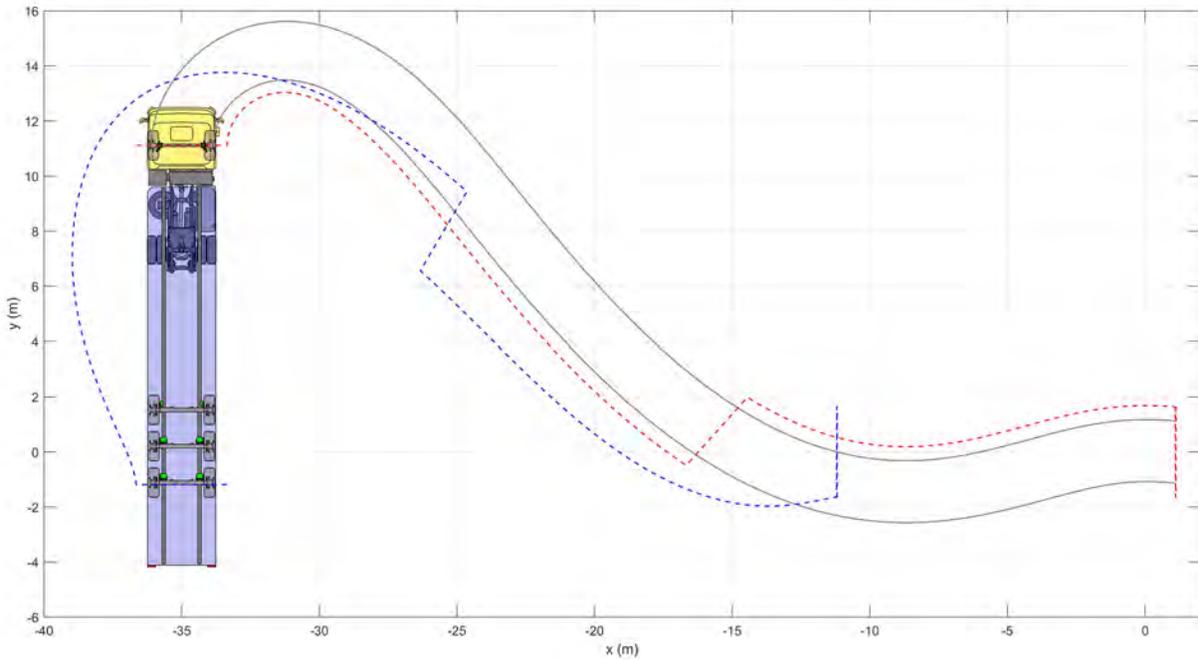


Figure 4.13: Predicted trajectory for a tractor-semitrailer in autonomous mode for parking place changing

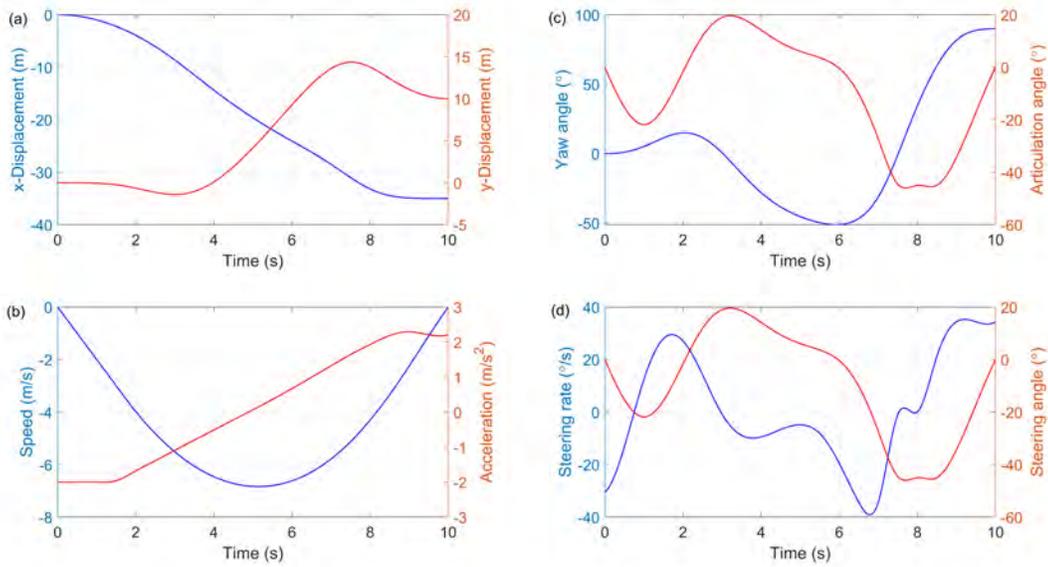


Figure 4.14: Results of states and controls for a tractor-semitrailer in autonomous mode for parking place changing: (a) - mass center displacement, (b) - speed/acceleration, (c) - yaw angle vs. folding angle, (d) - steering rate control and angle

Finally, the last model simulated was the **Articulated Vehicle with Semitrailer Steered Axle**.

The following constraint parameters were used to test the model:

Table 4.12: No space state and constraint parameters for articulated vehicle with semitrailer steered axle model

Name	Parameter	Value
Prediction Horizon Time	$L_1$	3.8 m
	$L_2$	7.57 m
	$e_{tr}$	0.47 m
	$t_f$	10 s
Initial State	$q_0$	$(0, 0, 0, 0, 0, 0)^T$
Final State	$q_f$	$(-35, 10, \pi/2, 0, 0, 0)^T$
State Restrictions	$\theta$	$-0.7 \leq \theta \leq 0.7$
	$\psi$	$-\pi/2 \leq \psi \leq \pi/2$
	$\zeta$	$-0.61 \leq \zeta \leq 0.61$
	Velocity	$-10 \leq v \leq 10$
Control Restrictions	$\omega_\theta$	$-0.7 \leq \omega_\theta \leq 0.7$
	$\omega_\zeta$	$-0.61 \leq \omega_\zeta \leq 0.61$
	Acceleration	$-2 \leq a \leq 2.5$

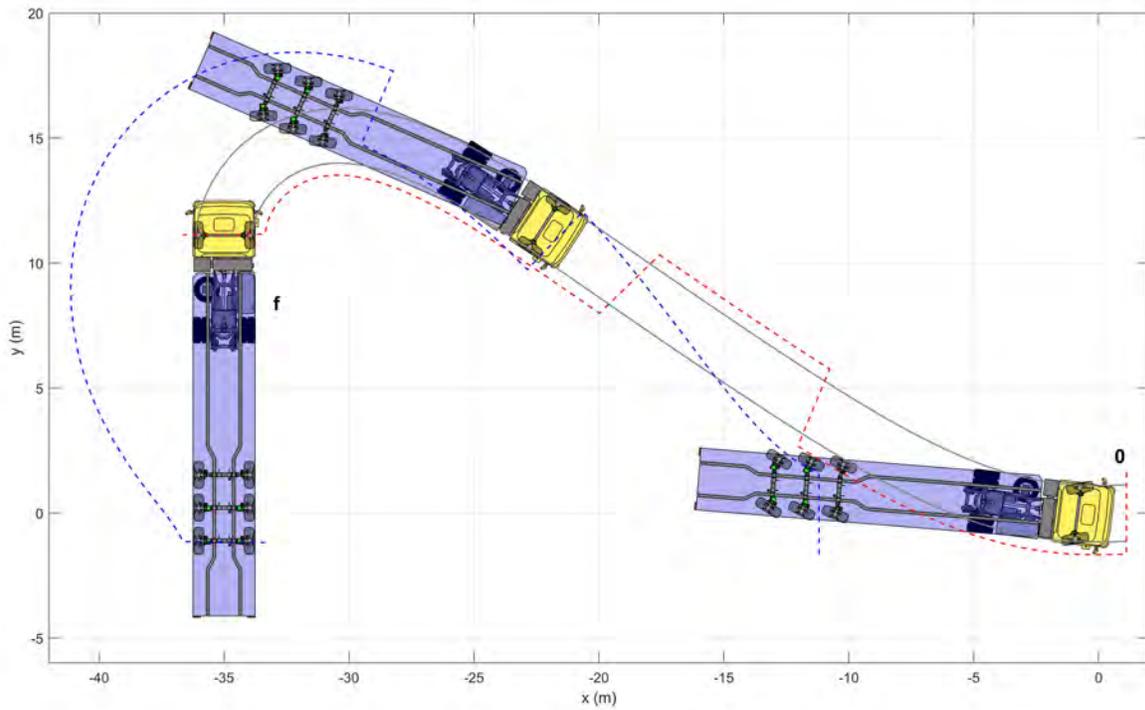


Figure 4.15: Predicted trajectory for an articulated vehicle with steered semitrailer's axle for parking place changing

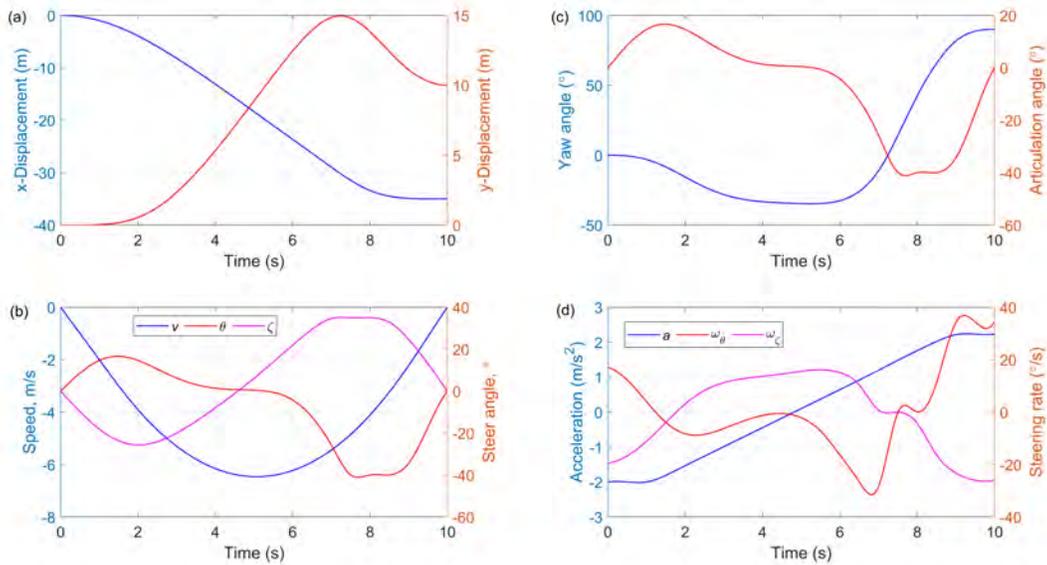


Figure 4.16: Results of states and controls for an articulated vehicle with steered semitrailer's axle in autonomous mode for parking place changing: (a) - mass center displacement, (b) - speed and steering angles, (c) - yaw angle vs. folding angle, (d) - acceleration / steering rate controls

The results show that the models are successful when implanted with no space restrictions. Although the initial and final values of the state vector are specified the model still has a lot to

accomplish. For each vehicle, the MPC must orient itself so that it can accurately arrive at the final position. This application is most suitable for the parking of AV. However, other applications include delivery trucks docking at dispatch centers autonomously.

### **4.3 Sensitivity Analysis**

The nonlinear MPC operates by solving the optimization problem at each sample time. As mentioned earlier, this sequence of repeated measure, predict and optimize is known as receding horizon control. For linear systems, there are several studies that have suggested selection criteria for the prediction horizon based on the applicable scenario. However, not much research has been conducted for nonlinear systems. Some of the results indicate that shortening the prediction horizon time tends to produce a slower system response. However, conventional MPC feedback suggests that longer horizon tend to produce aggressive control action, more overshoot and are more sensitive to disturbances. However, having a longer horizon time has been shown to result in faster responses from vehicle models. Conducting a sensitivity analysis on the prediction horizon time allows for insight into the effects it has on the model.

In order to determine the impact of varying the prediction horizon time, a deterministic sensitivity analysis was conducted using the single passenger car model. The only parameter that was varied was the prediction horizon time with the other state and constraint parameters remaining constant Table 4.1 and Table 4.2. In each case, the value of the prediction horizon parameter was incrementally changed while keeping other parameters constant. The overall performance of the model was investigated for each scenario.

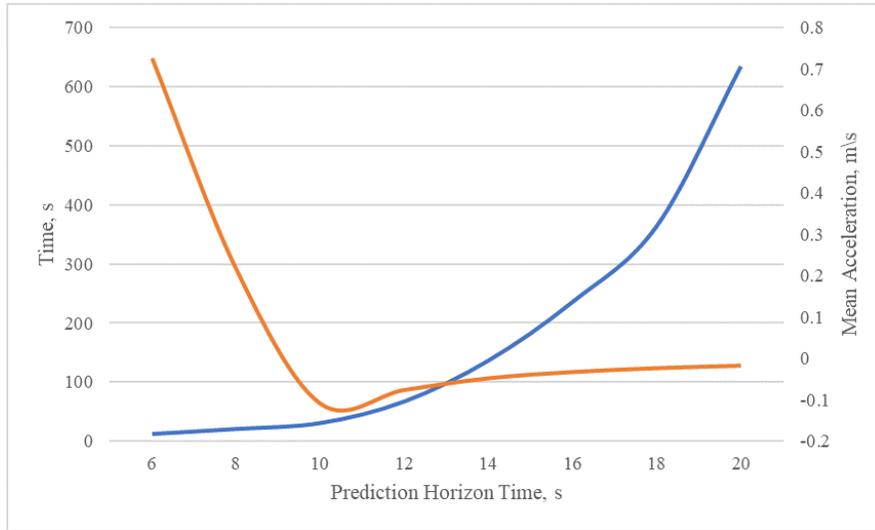
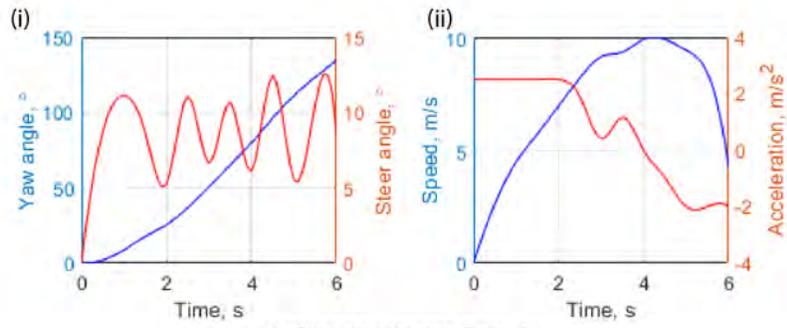


Figure 4.17: Computing time in seconds and mean acceleration with different prediction horizon times

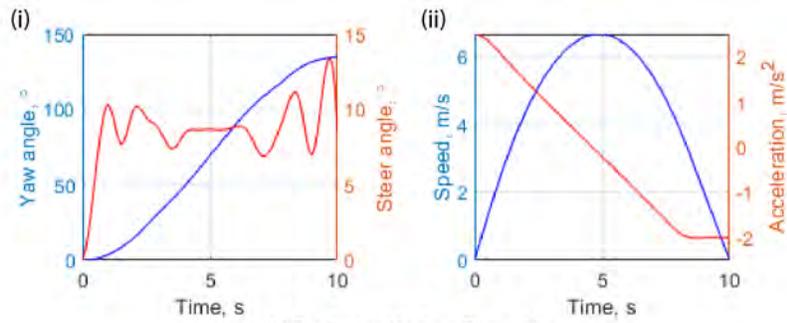
It was found that the nominal stability of the single car model was strongly affected by the prediction horizon time. A critical prediction horizon time approximately 10s to 13s was identified as the ideal range for optimal results of the model Fig 4.17. Prediction horizon times shorter than the critical range of values produced a slow and unresponsive situation while the mean acceleration was relatively high Fig 4.17. Based on the simulation results, the effect of increasing the prediction horizon time above the critical range of values resulted in a faster response in the model, however, the simulations were found to be unstable. This could be due to the additional computation time required by the hardware to run the simulation Fig 4.17. More results from the sensitivity analysis Fig 4.18 highlight the direct relationship between the speed and acceleration and yaw angle and steering angle as the prediction horizon time varies.

The importance of the prediction horizon time has been highlighted from the results of the sensitivity analysis. If the prediction horizon time is too great there was a faster response in the model, however, the model was not stable. This could likely be due to the inability of the optimization function to minimize the function. On the other hand, lower values of the prediction horizon time produced an unresponsive model. The critical values for the prediction horizon time were found to be ranging from 10s to 15s. It is important to note that the speed of execution of the

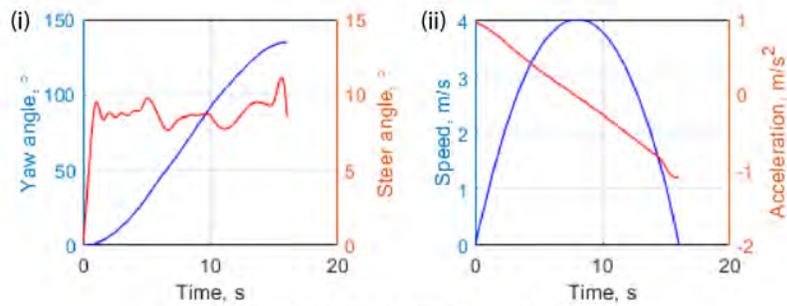
simulation can be affected greatly by the hardware which it is being run on. As a result, the results of the sensitivity analysis may differ if the simulations are being ran on more advanced hardware. It is also important to note that while the single car model was used in this sensitivity analysis it is easily capable of evaluating the impact of varying the prediction horizon time, this is because it is considered a simple model which the other models are built upon.



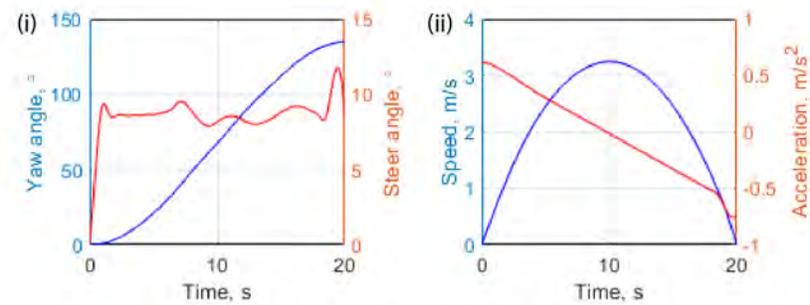
(a) Prediction Horizon Time - 6s



(b) Prediction Horizon Time - 10s



(c) Prediction Horizon Time - 16s



(d) Prediction Horizon Time - 20s

Figure 4.18: Results of sensitivity analysis for a passenger car in autonomous mode at the roundabout arc with different prediction horizon times showing (i) - speed and steering angles, (ii) - acceleration / steering rate controls

# 5 CONCLUSIONS

## 5.1 Summary of Results

In this project, the application of nonlinear MPC to perform curvilinear motion planning and lane-keeping for AV is presented. The main objective of this paper is to show that with proper modeling and formulation of motion constraints, curvilinear motion planning can be achieved with nonlinear MPC. The second objective is to test the NMPC precision and adequacy and its sensitivity to changes in the vehicle state parameters. Mathematical models for four (4) vehicle types were derived and then a nonlinear MPC framework was formulated. The framework formulates the problem as a nonlinear model predictive control problem. Simulations were performed using the Simulink/MATLAB environment using the fully nonlinear kinematic vehicle model to test the limits of the controller and its robustness to changes in the vehicle parameters. The models were tested in different types of simulation scenarios by varying vehicle parameters including (a) speed and steering angles, (b) yaw angle vs. folding angle and (c) acceleration/steering rate controls. The efficiency and the sensitivity of the proposed combined path planning and MPC framework have been evaluated under numerous simulated test cases.

Simulations are presented which demonstrate the ability of the suggested models to successfully perform curvilinear motion staying safely within the bounds, while simulations of several models validate its performance. The results show the effectiveness of the controller in the presence of uncertainty and utility of this approach. Further testing of the robustness of the models shows that they can perform predicted motions successfully in the absence of space restrictions where the initial and final values of the state vectors were specified. Moreover, a deterministic sensitivity analysis was conducted in order to determine the impact of the prediction horizon time on the model. More specifically, the trade-off between model performance and prediction horizon time has been highlighted. The minimum prediction horizon which provides acceptable performance and vehicle stability has been quantified at several vehicle speeds. Experimental results show that a

critical prediction horizon time approximately 10 to 13 seconds was identified as the ideal range for optimal results of the model. It was shown that the proposed models were successful in satisfying curvilinear motion planning by keeping within the constraints of the lane in a circle motion.

In conclusion, the computational complexity of the MPC framework has been highlighted and simulation results have shown the possibility of experimentally validating the models at varying prediction horizon times. It was found that with proper modeling and formulation curvilinear motion planning can be achieved with nonlinear MPC. It is evident that MPC can be systematically designed and implemented to control a nonlinear vehicle model with constraints on the states and input.

## **5.2 Future Work**

Possible extensions of this work include the implementation of the models to test obstacle avoidance with stationary objects or integrating other vehicles in the simulation to implement accident avoidance capability. This has proven to be computationally intensive and as such, more practical dedicated hardware may be required. The results from the sensitivity analysis further motivate efforts towards a real-time implementation to evaluate the system in real-world scenarios. An important design metric for realtime implantation of MPC is the computation time. It will be interesting to analyze the realtime effects of the MPC when the models are simulated on dedicated hardware.

In this project, vehicle models used were based on vehicle kinematics. Further expansion of this work can include comparing the performance of kinematic models with tirebased models to see the difference in performance and if there are extra computational requirements. Expanding the simulation scenarios to include more complex vehicle models can expand the scope of the research to include different classes of AV including aerial vehicles.

# REFERENCES

- Abbas, M. A., Milman, R., and Eklund, J. M. (2017). Obstacle Avoidance in Real Time with Nonlinear Model Predictive Control of Autonomous Vehicles. *Canadian Journal of Electrical and Computer Engineering*, 40(1):12–22.
- Carvalho, A., Gao, Y., Gray, A., Tseng, H. E., and Borrelli, F. (2013). Predictive control of an autonomous ground vehicle using an iterative linearization approach. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, (Itsc):2335–2340.
- Du, X., Htet, K. K. K., and Tan, K. K. (2016). Development of a Genetic-Algorithm-Based Nonlinear Model Predictive Control Scheme on Velocity and Steering of Autonomous Vehicles. *IEEE Transactions on Industrial Electronics*, 63(11):6970–6977.
- Du, X. and Tan, K. K. (2015). Autonomous reverse parking system based on robust path generation and improved sliding mode control. *IEEE Transactions on Intelligent Transportation Systems*, 16(3):1225–1237.
- Espinosa, F., García, R., Mazo, M., López, E., and Mateos, R. (1998). Modelling and Simulation of the Kinematic and Dynamic Behavior of a Fork-Lift-Truck. *IFAC Proceedings Volumes*, 31(3):51–55.
- Falcone, P., Borrelli, F., Asgariy, J., Tsengy, H. E., and Hrovat, D. (2007). A model predictive control approach for combined braking and steering in autonomous vehicles. *2007 Mediterranean Conference on Control and Automation, MED*.
- Frasch, J. V., Gray, A., Zanon, M., Ferreau, H. J., Sager, S., Borrelli, F., and Diehl, M. (2018). An auto-generated nonlinear MPC algorithm for real-time obstacle avoidance of ground vehicles. *2013 European Control Conference (ECC)*, pages 4136–4141.
- Guay, M., Adetola, V., and DeHaan, D. (2017). *Robust and Adaptive Model Predictive Control of Nonlinear Systems*, volume 37.

- Huang, L. and Panagou, D. (2017). Automated turning and merging for autonomous vehicles using a Nonlinear Model Predictive Control approach. *Proceedings of the American Control Conference*, pages 5525–5531.
- Huang, W., Wu, X., Zhang, Q., Wu, N., and Song, Z. (2014). Trajectory optimization of autonomous driving by differential dynamic programming. *2014 13th International Conference on Control Automation Robotics and Vision, ICARCV 2014*, 2014(December):1758–1763.
- Jiang, H., Wang, Z., Chen, Q., and Zhu, J. (2017). Obstacle avoidance of autonomous vehicles with CQP-based model predictive control. *2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016 - Conference Proceedings*, pages 1668–1673.
- Kong, J., Pfeiffer, M., Schildbach, G., and Borrelli, F. (2015). Kinematic and dynamic vehicle models for autonomous driving control design. *IEEE Intelligent Vehicles Symposium, Proceedings*, 2015-Augus:1094–1099.
- Marino, R., Scalzi, S., Orlando, G., and Netto, M. (2009). A nested PID steering control for lane keeping in vision based autonomous vehicles. *Proceedings of the American Control Conference*, pages 2885–2890.
- Morari, M. and Lee, J. (1997). Model Predictive Control: Past, Present and Future. *Computers & Chemical Engineering*, 23(4-5):667–682.
- Nayl, T., Nikolakopoulos, G., and Gustafsson, T. (2015). Effect of kinematic parameters on MPC based on-line motion planning for an articulated vehicle. *Robotics and Autonomous Systems*, 70:16–24.
- Nocedal, J., Wright, S. J., and Robinson, S. M. (2004). *Numerical Optimization*.
- Park, J. M., Kim, D. W., Yoon, Y. S., Kim, H. J., and Yi, K. S. (2009). Obstacle avoidance of autonomous vehicles based on model predictive control. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 223(12):1499–1516.

- Quiryneen, R., Berntorp, K., and Di Cairano, S. (2018). Embedded Optimization Algorithms for Steering in Autonomous Vehicles based on Nonlinear Model Predictive Control. *Proceedings of the American Control Conference*, 2018-June:3251–3256.
- Rafaila, R. C. and Livint, G. (2015). Nonlinear model predictive control of autonomous vehicle steering. *2015 19th International Conference on System Theory, Control and Computing, ICSTCC 2015 - Joint Conference SINTES 19, SACCS 15, SIMSIS 19*, pages 466–471.
- Rajamani, R. (2005). *Vehicle Dynamics and Control*. Springer, New York, NY, UNITED STATES.
- Ritzen, P., Roebroek, E., Van De Wouw, N., Jiang, Z. P., and Nijmeijer, H. (2016). Trailer Steering Control of a Tractor-Trailer Robot. *IEEE Transactions on Control Systems Technology*, 24(4):1240–1252.
- Saback, R. M., Conceicao, A. G. S., Santos, T. L. M., Albiez, J., and Reis, M. (2019). Nonlinear Model Predictive Control Applied to an Autonomous Underwater Vehicle. *IEEE Journal of Oceanic Engineering*, (0364):1–14.
- Shiller, Z. and Gwo, Y. R. (1991). Dynamic Motion Planning of Autonomous Vehicles. *IEEE Transactions on Robotics and Automation*, 7(2):241–249.
- Takács, G. and Rohal'-Ilkiv, B. (2012). *Model Predictive Vibration Control*. Springer London, London.
- Turri, V., Carvalho, A., Tseng, H. E., Johansson, K. H., and Borrelli, F. (2013). Linear model predictive control for lane keeping and obstacle avoidance on low curvature roads. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, (Itsc)*:378–383.
- Werner, R., Mueller, S., and Kormann, G. (2012). Path Tracking Control of Tractors and Steerable Implements Based on Kinematic and Dynamic Modeling. *International conference on precision agriculture*.