TECHNISCHE UNIVERSITÄT
ILMENAU

Technische Universität Ilmenau

Fakultät für Informatik und Automatisierung

Master Thesis

Reliable Autonomous Vehicle Control – a Chance Constrained Stochastic MPC Approach

To achive the Degree of:

Master of Science (M. Sc.)
in Mechatronic

Submitted by: Luis Felipe Poma Aliaga
Date and Place of Birth: February 15, 1992 Huanuco - Perú

Supervisor (TU Ilmenau): Dr. rer. nat. Abebe Geletu W. Selassie
Supervisor (PUCP): Dr. Julio C. Tafur
Date and Place: May 11, 2017, Ilmenau, Germany

Statement of Authorship

The present work has been made independently without use other than those specified sources. All points were taken literally or in accordance with their published sources are identified as such. The work has not been submitted in the same or similar form, or in part, under one or other tests.

Ilmenau, May 11, 2017

Luis Felipe Poma Aliaga

Abstract

In recent years, there is a growing interest in the development of systems capable of performing tasks with a high level of autonomy without human supervision. This kind of systems are known as autonomous systems and have been studied in many industrial applications such as automotive, aerospace and industries. Autonomous vehicle have gained a lot of interest in recent years and have been considered as a viable solution to minimize the number of road accidents. Due to the complexity of dynamic calculation and the physical restrictions in autonomous vehicle, for example, deterministic model predictive control is an attractive control technique to solve the problem of path planning and obstacle avoidance. However, an autonomous vehicle should be capable of driving adaptively facing deterministic and stochastic events on the road. Therefore, control design for the safe, reliable and autonomous driving should consider vehicle model uncertainty as well uncertain external influences. The stochastic model predictive control scheme provides the most convenient scheme for the control of autonomous vehicles on moving horizons, where chance constraints are to be used to guarantee the reliable fulfillment of trajectory constraints and safety against static and random obstacles. To solve this kind of problems is known as chance constrained model predictive control. Thus, requires the solution of a chance constrained optimization on moving horizon. According to the literature, the major challenge for solving chance constrained optimization is to calculate the value of probability. As a result, approximation methods have been proposed for solving this task.

In the present thesis, the chance constrained optimization for the autonomous vehicle is solved through approximation method, where the probability constraint is approximated by using a smooth parametric function. This methodology presents two approaches that allow the solution of chance constrained optimization problems in inner approximation and outer approximation. The aim of this approximation methods is to reformulate the chance constrained optimizations problems as a sequence of nonlinear programs. Finally, three case studies of autonomous vehicle for tracking and obstacle avoidance are presented in this work, in which three levels probability of reliability are considered for the optimal solution.

Zusammenfassung

In den letzten Jahren gibt es ein wachsendes Interesse an der Entwicklung von Systemen, die in der Lage sind Aufgaben mit einem hohen Maß an Autonomie und ohne menschliche Aufsicht durchzufhren. Dies Art der Systeme werden als autonome Systeme bezeichnet, und wurden in vielen industriellen Anwendungen der Automobil und Luftfahrtindustrie untersuch. Autonome Fahrzeuge (engl. autonomous vehicle) haben in den letzten Jahren stark an Interesse gewonnen und gelten als eine tragfähige Lösungen um die Anzahl der Unfälle im Straßenverkehr zu minimieren. Aufgrund der Komplexität der dynamischen Gleichungen und die physikalishen Einschränkungen in autonomen Fahrzeug, ist zum Beispiel deterministische Modellprädiktive Regelung (engl. model predictive control) eine attraktive Steuerungsmethode Technik, um das Problem der Bahnplanung und die Vermeidung von Hindernis zu lösen. Dennoch sollte ein autonomen Fahrzeug die Möglichkeit besitzen trotz deterministischer und stochastischer Ereignisse auf der Straße fahren zu können. Ein Regelungsdesign für sicheres und zuverlässiges Fahren sollte neben der fahrzeugtechnischen Modellen Sicherheit auch externe Einflüsse berücksichtigen. Die stochastische modellprädiktive Regelung ist als Kontrollschema für die Bewegung von Fahrzeuge mit beweglichen Horizonten am besten geeignet, wenn Mönglichkeitsbegrenzung (engl. chance constraint) benötigt wird um die Zuverlässigkeit auf dem Fahrweg und beim Ausweichen von Hindernissen zu garantieren. Um diese Problemstellungen zu lösen, bieten sich möglichkeitsbegrenzende modellprädiktive Regelungen an. Für einen beweglichen Horizont kann eine möglichkeitsbegrenzende Optimierung (engl. chance constrained optimization) genutzt werden. Laut Literatur ist die große Herausforderung für die Lösung von möglichkeitsbegrenzende Optimierung die korrekte Berechnung der Wahrscheinlichkeiten. Infolgedessen sind Näherungsverfahren vorgeschlagen worden, für die Lösung dieses schwierige Aufgabe. In der vorliegenden Masterarbeit ist die wahrscheinlichkeitsbeschränkte Optimierung für das autonome Fahrzeug durch entsprechende Näherungsverfahren für die Wahrscheinlichkeitsbeschränkungen unter Verwendung von glättenden Parameterfunktionen. Diese Methode beruht auf zwei Ansätzen zur Lösung des wahrscheinlichkeitsbeschränkte Optimierung Problems: der inneren und der äußeren Annäherung. Das Ziel dieser Näherungsverfahren ist die

Formulierung die wahrscheinlichkeitsbeschränkte Optimierungsprobleme als eine Folge von nichtlinearen Programmen. Zum Abschluss wurden drei Fallstudien der autonome Fahrzeuge für das Tracking und das Vermeiden von Hindernissen vorgestellt. Dafür würde eine dreitstufige Wahrscheinlichkeit als optimale Lösung ermittelt.

Acknowledgment

My deep gratitude goes first to my advisor Dr. Abebe Geletu for his valuable support and his continuous guide. I also thank Prof. Pu Li for the interest shown and the support.My sincere thanks also to Dr. Tafur Sotelo for being my co-advisor and for providing worthy comments and advice.

I would like to thank my family and all my friends with whom I spent this year in Ilmenau for the support provided.

My deeply thanks to CONCYTEC for the financial support and the opportunity of living this experience.

Contents

List of Figures

List of Tables

List of Acronyms

AFS	Active Front Steering
ALV	Autonomous Land Vehicle
ASM	Active set method
IPM	Interior point method
SQP	Sequential quadratic programming
QP	Quadratic programming
NLP	Nonlinear programming
IpOpt	Interior point optimizer
ODE	Ordinary differential equation
KKT	Karush-Kuhn-Tucker
BFGS	Broyden-Fletcher-Goldfarb-Shanno (method)
MPC	Model predictive control
SMPC	Stochastic model predictive control
DMPC	Deterministic model predictive control
NMPC	Nonlinear model predictive control
ABS	Anti-lock brake system
DCS	Dynamic stability control
ECM	Engine control unit
OCP	Optimal control problem

IA	Inner approximation
OA	Outer approximation
CCOPT	Chance constrained optimization
SAA	Sample Average Approximation
QSM	Quasi-Monte Carlo
ACC	Adaptive Cruise Control
AV	Autonomous vehicle
LP	Linear programming
LTV	Linear time varying
HJB	Hamilton Jacobian Bellman
PDE	Partial differential equation
BVP	Boundary value problem
IVP	Initial value problem
SAE	Society of Automotive Engineers

Chapter 1

Introduction and motivation

1.1 Motivation

The first experiments in the autonomous vehicles field started between the decades 1920 and 1980 where car companies and universities were the first pioneers. One of the important works was the Navlab vehicles based on vision algorithms and tested in a realistic outdoor environment [71]. Another interesting application was Autonomous Land Vehicle (ALV) project at the Carnegie Mellon University where a mobile robot was developing, which is capable of operating in the real world outdoor [45]. Since then, numerous companies and research organizations have aroused their interest in developing prototypes for Autonomous vehicle (AV) such as Mercedes-Benz, General Motors, Continental Automotive Systems, IAV, Autoliv Inc., Bosch, Nissan, Renault, Toyota, Audi, Hyundai Motor Company, Volvo, Tesla Motors, Peugeot, Local Motors, AKKA Technologies . Currently, with advanced computers, the incorporation of the complicated dynamic of AV and tire force model to the algorithm of control can be accounted for active steering, in comparison to the previous technologies. Since the early 2000's, car companies have been working on improving vehicle autonomy. Google, Uber and Tesla are the main companies that have had success with autonomous vehicles. Improvements are still being made today to get vehicles operate fully autonomously. However, in the case of uncertainties that can arise due to the error in the motion equation, measurement error or an external perturbation, the trajectory tracking or obstacle avoidance may fail. It can be a serious and interesting issue due to the fact that in real application the uncertainties should be considered.

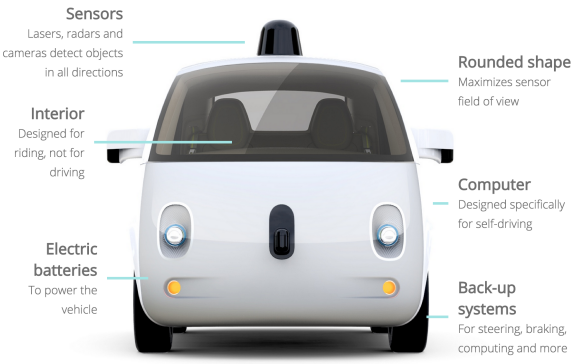

Figure 1.1 – Google's driverless project car [2]

1.2 Objective of the thesis

The major objectives of this master thesis are:

- to identify a mathematical model of vehicle dynamics for autonomous drive
- to study the source of uncertainty in the model parameters and external disturbances
- to study the chance constrained MPC scheme for the reliable, safe, and optimal autonomous vehicle drive
- to study appropriate MPC algorithms and to implement on case-studies

1.3 Organization of the thesis

The content presented in this thesis is organized as follows:

- *Chapter 2* briefly summarizes the state of art of the AV using Deterministic model predictive control (DMPC) and SMPC for the solution of the tracking problem and obstacle avoidance. Moreover, Sequential quadratic programming (SQP) and Interior point method (IPM) are introduced as methods for solving nonlinear. optimization problems.
- *Chapter 3* describes the concept of linear and nonlinear stochastic MPC with chance constraint, where different formulations are presented. Then, approximation method are introduced as an better alternative for the Chance constrained optimization (CCOPT). Finally, IA and OA are presented and formulated.

- *Chapter 4* describes the implementation of the CCOPT in IpOpt [73] with C++ interface. To obtain the analytical expressions of the gradient vectors and Jacobian matrices and even the Hessian matrix, CASADI[6] is employed that is an Automatic Differentiation software with interface C++.
- *Chapter 5* presents the simulation and experimental tests results of the three case studies for AV, formulated as CCOPT problems, to verify the computational efficiency for various reliability levels and approximation parameter τ for IA and OA.
- *Chapter 6* concludes this thesis with a summary of the main results and discusses the outlook for possible future research.

Chapter 2

State of the art

2.1 Autonomous Vehicles

Recent advances in the autonomous vehicle are facilitated due to advances in computing technology and the high performance algorithms. An autonomous ground vehicle is mechanical equipment which is capable to operate,with a certain level of autonomy, across the surface on the ground. According to the Society of Automotive Engineers (SAE), there exist 6 scales for grading vehicle automation [35, 46]. More details about the mean of each level can be found in [1].

- Level 1: with driver intervention.
- Level 2: driver Assistance.
- Level 3: partial automation
- Level 4: conditional automation
- Level 5: high automation.
- Level 6: full automation.

In recent years, the autonomous vehicles have been proposed as a promising solution for the reduction of road accidents, improved safety, reduced congestion, lower emissions, reduction of driving stress. According to [61] , the human error is the cause of 93% of the road accidents. Therefore, Active Safety is a viable solution for addressing this issue. This approach has been employed to improve the vehicle controllability and stability of the vehicle [25]. Some examples of Active Safety System are the following:

- Anti-lock brake system (ABS) designed to avoid the sliding of the wheel, through improving the control and reducing the stopping distance [47].

- Dynamic stability control (DCS) that avoid the car rollout [47].
- Cruise control with lane departure avoidance and anticollision brake system [65].
- Adaptive Cruise Control (ACC) is an intelligent way of cruise control that slows down and speeds up automatically to keep pace with the car driving [66].
- Engine control unit (ECM) is preconfigured with timing-valve maps that allow a better performance in different atmospheric conditions while reducing contaminant emissions.

In general, the steps of the decision-making the architecture of a typical self-driving are employed as references in several applications for AV. The decision making system of a self-driving car is divided into four components as follow [60].

1. Route planning
2. Adaptive decision making
3. Motion planning
4. Vehicle control

On the other hand, autonomous ground vehicles systems normally must work in presence of stochastic scenarios. This represents a new challenge for the research. Although several works have been developed in AV, the feasible trajectory calculated may fail due to the fact that the uncertainties are not taken into account in the control algorithm. Therefore, the path planing under the uncertainties should be considered. The uncertainties in autonomous vehicles may arises due to the modeling errors, measurement errors and external perturbation.

1. Modeling errors: errors due to the linearization or simplification of the vehicle model.
2. Measurement errors: depends on the accuracy of the sensors.
3. External Perturbation: influence of the weather, road surface, air drag coefficient, wind force, friction coefficient, human mistake, etc.

To address this issues, the receding horizon concept provides the solution to deal with the dynamic and uncertain nature of the vehicle planning [46]. In particular, stochastic MPC provides the most convenient scheme the control of autonomous vehicles on moving horizons, where chance constraints are to be used for guaranteeing the reliable

satisfaction of trajectory constraints and safety against uncertainties. Stochastic MPC will be presented in the chapter 3 while a briefly review of deterministic MPC will described next.

2.2 Nonlinear MPC for Autonomous Vehicle

Model predictive control (MPC) is a modern technique control based on the receding horizon approach [67], in which the idea is to predict and optimize the future system behavior based on the model of process. The MPC has the advantage that introduces a feedback controller, which allows system stability and good performance in the presence of constraints.
At each instant time (t_k) is realized a set of steps, which are shown in Algorithm 1. Therefore, at each iteration an OCP is solved.

Algorithm 1: MPC Strategy[20]

Input : Initial value of the states
Output : Optimal control variables, Optimal state variables

1 **repeat**
2 Measure or estimate the process state x at time t_k;
3 Formulate the MPC problem for the optimization time $t \in [t_k, t_{k+N}]$;
4 Compute the optimal control sequence $\bar{u}^*(t), t \in [t_k, t_{k+N}]$ by solving the MPC problem;
5 Apply optimal control $\bar{u}(t) = \bar{u}^*(t_k), t \in [t_k, t_{k+1}]$ to the system until $t = t_{k+1}$;
6 Set $k = k + 1$;
7 **until** *Stopped by user;*;

There is no standard rule for choosing "N"(Prediction horizon). However, the value of "N" is determinant since it increase or decrease the complexity in solving the optimization problem. ($\bar{u}^* = \{u_k^*, u_{k+1}^* ... u_{k+N}^*\}$).
The MPC approach requires a model of the plant, which should depict the significant dynamic. Moreover, this model should not be too complex, nor should pursue an extremely high accuracy, because this causes large order models which sometimes have a very similar dynamic than more simple models with lower order.
The MPC approach operation is illustrated in 2.1:

Advantages and disadvantages of MPC

- Constraints can be specified for the state vector $\bar{x}$ and the control variables u_k.
- Management of multiple-input-multiple output (MIMO) for large-scale systems.

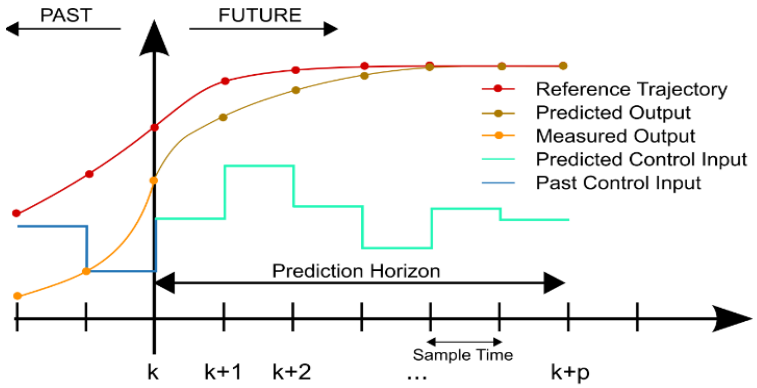

Figure 2.1 – A basic working principle of model predictive control [11]

- Minimization of trajectory tracking error.
- Propagation error, due to noise, is reduced.
- Compensation of disturbances is achieved by the feedback feature that is characteristic of the receding horizon approach[67].
- The sampling time can be determinant in the performance for the optimization algorithms. In particular,the sampling time and the length of the prediction horizon are critical for autonomous vehicle.

2.2.1 Deterministic MPC for Autonomous Vehicle

Autonomous vehicle is a challenging task that requires an advanced control scheme. Due to fact that the different vehicle dynamic models have been widely studied, MPC is an attractive tool for precise trajectory planning and obstacle avoidance. Moreover, MPC is a modern technique based on the system model to predict the model trajectory and minimize the deviation from a given reference for AV. There exist in the literature a variety of MPC approaches to AV, differing in the dynamic model, applications, robust, etc.

Although MPC presents a good control scheme the computational burden is a challenging task for real-time applications. Nevertheless, there exist approaches based on Linear time varying (LTV) systems that consists in the successive linearization of the vehicle model and they have been employed in [17, 23, 26, 28, 76]. Therefore, the computation time is reduced and can be applied in a real-time application. On other hand, the tractability of nonlinear MPC for autonomous vehicle has also been studied. In [25] a rapid prototyping system is implemented but it was tested only at low velocity, because of computational burden. In [20, 22] efficient solvers for nonlinear MPC are implemented in a real-time

application for tracking and avoidance of obstacles. To test the perfomance of the solvers proposed SUMMIT mobile robot [3] was used. There exist also works where LTV systems and Nonlinear model predictive control (NMPC) have been widely in order to compare the computational time [47, 59]. AFS system in an autonomous vehicle has

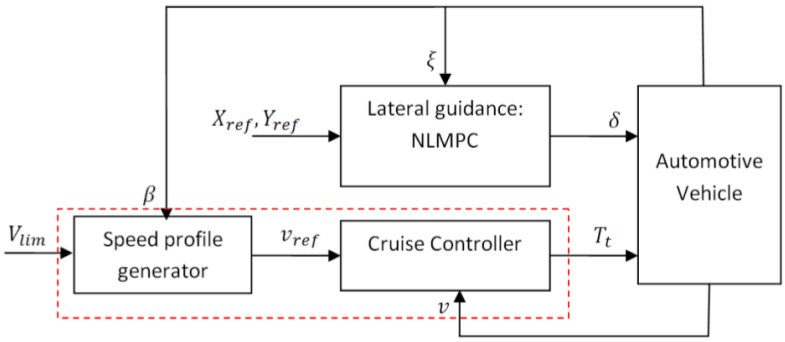

Figure 2.2 – Global architecture for longitudinal and lateral control [10]

been severally studied and implemented to perform a sequence of double lane changes [25, 47]. The integration of AFS with braking at the four wheel has been added in [24, 27, 28] to achieve a better yaw and lateral stabilization. However, when the system braking is considered, the main difficulty is the modeling of the longitudinal dynamics. Other interesting work developed in consider the integration of the longitudinal and lateral control, where the longitudinal dynamic is modeled with the speed profile and a cruise controller [9, 10]. The idea of coupled and decoupled longitudinal and lateral dynamics have been considered in [44], where very detailed modeling of the vehicle, including engine and power train modeling are taken into account.

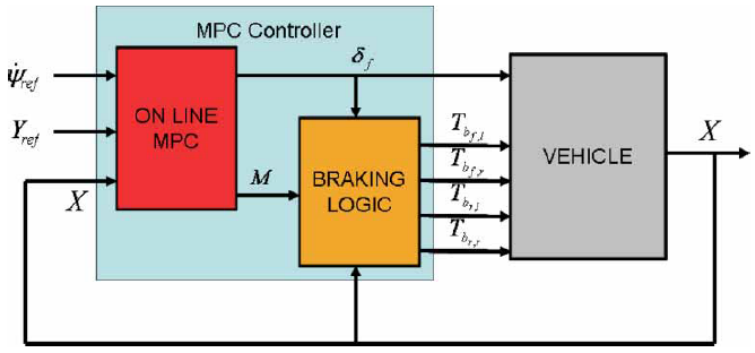

Figure 2.3 – AFS with braking system formulated in [28]

2.2.2 Obstacles Avoidance Approaches

As has been mentioned before, obstacle avoidance methods have been widely studied for Autonomous vehicle (AV). A method to solve is to define a path constraint that determines the feasibility of the state trajectory for the vehicle. Another interesting method is to use a potential function to penalize the distance between the obstacles defined and the vehicle. The two approaches will be presented next.

Path Constraint Approach

This approach is based on defining a safety region for vehicle using the path constraint. This region can be defined as an ellipse due to the smooth characteristics [20, 21, 22]. The path constraint is formulates as follow.

$$\frac{(x_{pos} - x_{obs})^2}{r_a^2} + \frac{(y_{pos} - y_{obs})^2}{r_b^2} \geq 1, \tag{2.1}$$

Where r_a and r_b are the axes of the ellipse, the position of the obstacle is (x_{obs}, y_{obs}) that represents the center of the ellipse and the position of vehicle is depicted by (x_{pos}, y_{pos})

Potential Function Approach

Another important approaches to assure obstacle avoidance is the formulation of a potential function, which can be added to the objective function. Therefore, the terms

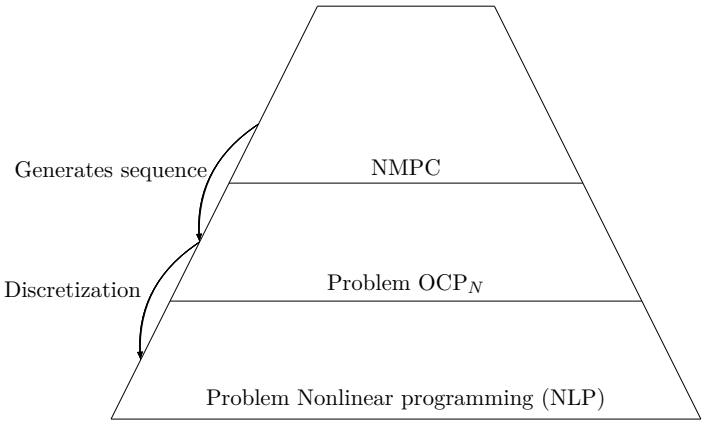

Figure 2.4 – Architecture of MPC [39]

P_{obs} penalizes the relative distance between the obstacle detected and the vehicle. The potential function can be formulated as follow.

$$P_{obs} = \frac{K_{obs}}{(x_{pos} - x_{obs})^2 + (y_{pos} - y_{obs})^2 + \epsilon} \geq 1, \tag{2.2}$$

Where K_{obs} is the gain value and ϵ is a parameter used to avoid non singularity. This approach has been studied in the literature [4, 70] . However, the drawback of this method is that the dimensions of the safety region cannot be defined as in the case of the path constraint approach [20].

2.3 Optimal Control Problem

As has been mentioned before, MPC is a control technique that solves an OCP on each prediction horizon. the architecture of NMPC is described in Figure 2.4, where the NMPC generates a sub-problem of OCP. The OCP can be formulated as follow.

$$\min_{u(t)} \quad J = \overbrace{\Psi(x(t_f), tf)}^{\text{Lagrange term}} + \overbrace{\int_{t_0}^{t_f} L(x(t), u(t), t)dt}^{\text{Mayer term}} \tag{2.3a}$$

$$\text{subject to} \quad x(t_0) = x_0 \tag{2.3b}$$

$$\dot{x} = f(x(t), u(t), t) \quad t \in [t_0, t_f] \tag{2.3c}$$

$$u_{min} \leq u(t) \leq u_{max} \tag{2.3d}$$

$$g(x(t), u(t), t) \leq 0 \tag{2.3e}$$

Where $x(t)$ and $u(t)$ are the state and control variables, respectively x_0 is the initial state, t_0 and t_f are the initial and final time. In the OCP problem (2.3), J is the objective function. The dynamic constraint (3.3c) is represented by ODE in the time horizon $[t_0, t_f]$. The constraints (2.3e) and (2.3d) represented by inequality constraint are the bound limits on the control and state variable.

2.3.1 Solution methods for OCP

The solution of an OCP is quite challenging but it has great advantage since it can be applied in the field of industry as chemical, mechanical, electrical, etc. In the literature, solution approaches can be divided into three categories [5],[20]. Dynamic programming, indirect methods and direct methods has been employed for the solution of OCP as shown in Figure 2.5.

- **Dynamic Programming:** This method uses the Hamilton Jacobian Bellman (HJB), which is a Partial differential equation (PDE) by using the principle of optimality.
- **Indirect methods:** This method is known as first optimization and then discretization. The main idea is to solve the OCP using the optimality conditions that allows to transform the OCP into nonlinear Boundary value problem (BVP).
- **Direct Methods:** these approaches are the most used and studied, according to the literature. In comparison to the indirect methods are known as first discretization an then optimization. The idea of these methods is to solve the OCP by transforming it into an NLP as shown in the figure 2.4, which can be solved using numerical optimization methods. There exist different discretization techniques as Runge Kutta, Euler discretization,etc. Currently, there exist several direct methods, for instance, direct single shooting, multiple shooting and collocation methods that have been widely studied [5, 20, 22, 39]. In the following, a review of the methods mentioned are presented.

a) Direct Single Shooting

The direct single shooting formulates the OCP into a finite NLP problem. The time interval is divided into N intervals, where the control variable is piecewise constant in each interval $[t_k, t_k + 1]$.

$$t_0 < t_1 < t_2 < \ldots < t_N = tf \tag{2.4}$$

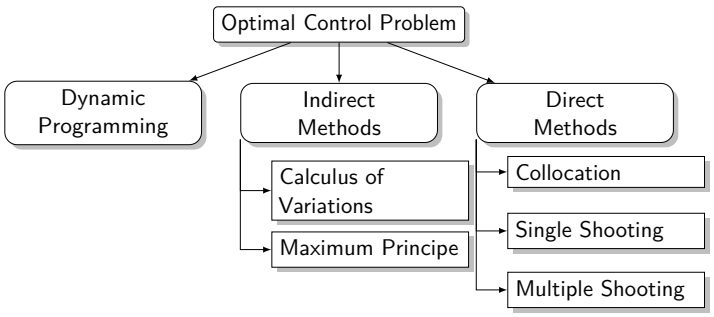

Figure 2.5 – Solution Methods for OCP

The solver NLP provides at each iteration a given control input sequence $u = [u_0, u_1, \ldots, u_{N-1}]$. Hence, the following Initial value problem (IVP) is solved.

$$\dot{x} = f(x(t), u(t), t), \quad x(t_0) = x_0, \quad t \in [t_0, t_f] \tag{2.5}$$

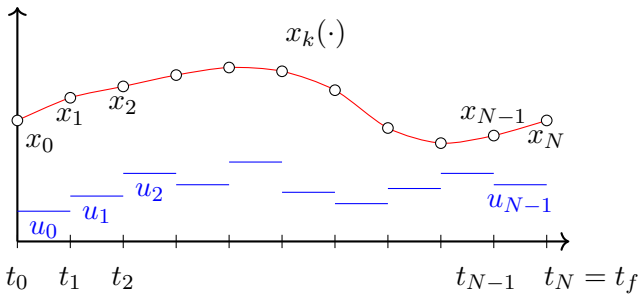

Figure 2.6 – Piecewise constant of the control variables

The idea of this method is to decouple the dynamics of the control system (3.3c) from the OCP. Hence, the NLP problem is formulated in terms of the shooting vector u_k.

$$\min_{u_k} \quad J = \Psi(x_N(u_{N-1})) + \sum_{k=0}^{N-1} L(x_k(u_k), u_k) \tag{2.6a}$$

$$\text{subject to} \quad u_{min} \leq u_k \leq u_{max} \tag{2.6b}$$

$$g(x_{k+1}(u_k), u_k) \leq 0, \quad k = 0, 1, \ldots, N-1 \tag{2.6c}$$

b) Multiple Shooting

The formulation of this method is based on [15]. The reformulation of the OCP into NLP problem starts in the same way as the single shooting method. The time interval is divided into N intervals, where the control variable is piecewise constant in each interval $[t_k, t_k + 1]$ and the state variables are discretized in the same point grinds.

$$t_0 < t_1 < t_2 < \ldots < t_N = tf \tag{2.7}$$

Therefore, in this method the ODE is solved independently for each interval considering an initial value states proportioned by the solver. However, to assure the continuity of the state trajectory, the equality constraints is to be added to the formulation of NLP, where the terminal value of state variable in each interval is equal to initial value of the

state variable in the next subinterval [5]. Therefore, the state variables are considered as part of the optimization variables.

$$w = \begin{bmatrix} x_0 \\ u_0 \\ x_1 \\ u_1 \\ \vdots \\ u_{N-1} \\ x_N \end{bmatrix} \tag{2.8}$$

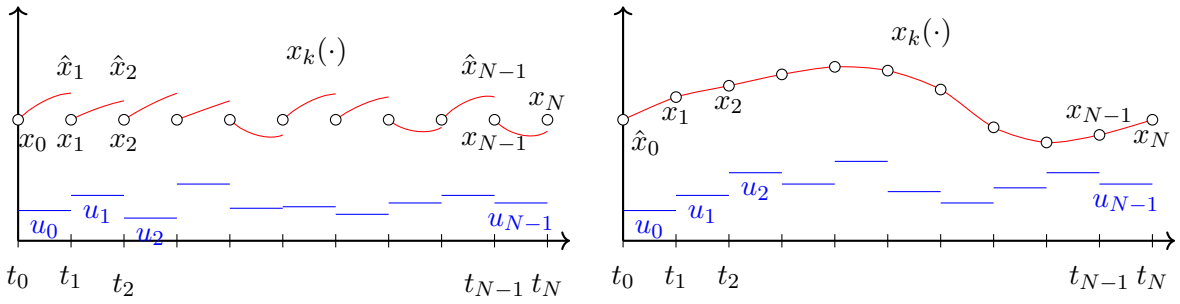

Figure 2.7 – (left) Single trajectories obtained through the solution of the ODE. (right) Convergence of state and control profiles for the direct multiple shooting method [20].

Hence, the NLP problem is formulated as follow.

$$\min_{u_k, x_k} \quad J = \Psi(x_N) + \sum_{k=0}^{N-1} L(x_k, u_k) \tag{2.9a}$$

$$\text{subject to } x(t_0) - x_0 = 0 \tag{2.9b}$$

$$\hat{x}_{k+1} - f(x_k, u_k) = 0, \quad k = 0, 1, \ldots, N-1 \tag{2.9c}$$

$$u_{min} \leq u_k \leq u_{max}, \quad k = 0, 1, \ldots, N-1 \tag{2.9d}$$

$$g(x_{k+1}, u_k) \leq 0, \quad k = 0, 1, \ldots, N-1 \tag{2.9e}$$

c) Collocation Methods

This method is based on the collocation solution to solve ODE, which is a generalization of the implicit Runge Kutta method. First, the time horizon is divided into N intervals and the control variables are considered as piecewise constant in each interval $[t_k, t_k + 1]$. These are known as collocation intervals. Then, each collocation interval is divided

into n internal points $t_{k,i}$, for $i = 1, 2, \ldots, n$, these points are referred as the collocation points. According to the Weiertrass theorem the state trajectory can be approximated at each collocation interval using a time dependent polynomial. Therefore, to assure the continuity of the initial state variable of each interval to be equal to the final state of the previous collocation interval. As a result, the OCP is formulated as very large NLP.

2.4 Methods for constrained optimization Problem

In this the section the most relevant solution methods for NLP will be presented. In an optimization problem, a function is to be minimized or maximized that depends on real variables, with unconstraint or constraint at all on the values of these variables. The function is referred as an objective function $f(x)$ that can be formulated as a quantity of cost, profit, efficiency, size, shape, weight or output [7].

2.4.1 Constrained minimizations problems

In real application, the optimization problems carry constraint. Hence, the feasible region is restricted due to the presence of constraint. The standard formulation of a constrained optimization problem is given by.

$$\min_{x \in R^n} \quad f(x) \tag{2.10a}$$

$$\text{subject to} \quad h_i(x) = 0, \quad i = 1, \ldots, m < n \tag{2.10b}$$

$$g_j(x) \leq 0, \quad j = 1, \ldots, p < n \tag{2.10c}$$

Where, f is the objective function to be minimize, $\mathbf{h(x)} = [h_1(x), ..., h_m(x)]^T$ is the set of equality constraint and $\mathbf{g(x)} = [g_1(x), ..., g_p(x)]^T$ is the set of the inequality constraint, respectively and $x \in R^n$ is the vector of optimization variables. The functions f, h_i and g_j are all differentiable [54].

2.4.1.1 Optimality conditions

There exist different solution methods to address the constrained optimization problems. The main solution techniques are summarized next.

a) Lagragian method

The Lagrange function is employed for the constrained optimization problems (2.10).

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^{m} \lambda_i h_i(x) + \sum_{j=1}^{p} \mu_j g_j(x) \tag{2.11}$$

The optimality conditions are formulated as follow.

$$\bigtriangledown_x L = \mathbf{0} \tag{2.12a}$$

$$\bigtriangledown_\lambda L = \mathbf{0} \tag{2.12b}$$

$$\mathbf{g(x)} \leq \mathbf{0} \tag{2.12c}$$

$$\mu_j g_j(x) = 0, \quad j = 1, \ldots, p \tag{2.12d}$$

$$\mu_j \geq 0, \quad j = 1, \ldots, p \tag{2.12e}$$

Where λ_i and μ_j are the Lagrange multipliers. The optimality conditions are known as Karush-Kuhn-Tucker (KKT) conditions [7]. Other important condition for $f(x)$ to be minimum is that $\bigtriangledown^2_{xx} L$ must be positive definite.

b)Penalty function method

The main idea of the penalty function method is to added a penalty function to the objective function. Therefore, the constrained optimization problem can be solved using algorithms for unconstrained optimization problem mentioned in the previous section. The new objective function with the penalty functions is presented as follow.

$$F(x) = f(x) + r_k \sum_{i=1}^{m} h_i^2(x) + \sum_{j=1}^{p} < g_j(x) >^2 \tag{2.13}$$

Where r_k is the penalty parameter and the function $< g_j(x) >= max(0, g_i(x))$. At each iteration the penalty parameter can be computed as follow [7].

$$r_k = max(1, \frac{1}{\|(< g_j(x) >, h_i(x))\|}) \tag{2.14}$$

The main advantage of this method is that the starting point can star from an infeasible point and the unconstrained optimization methods can be employed.

c) Augmented Lagrangian method

This method combine the Lagrange multipliers and the penalty functions [7]. The augmented Lagrangian functions can be formulated as follow.

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^{m} \lambda_i h_i(x) + \sum_{j=1}^{p} \mu_j \alpha_j(x) + r_k \sum_{i=1}^{m} h_i^2(x) + \sum_{j=1}^{p} < \alpha_j(x) >^2 \quad (2.15)$$

Where λ_i and μ_j are the Lagrange multipliers, r_k is the penalty parameter. These terms can be computed at each iteration as follow [7].

$$\alpha_j = max(g_j(x), \frac{-\mu_j}{2r_k}) \quad (2.16a)$$

$$\lambda_i^{k+1} = \lambda_i^k + 2r_k h_i(x) \quad (2.16b)$$

$$\mu_j^{k+1} = \mu_j^k + 2r_k max(g_j(x), \frac{-\mu_j}{2r_k}) \quad (2.16c)$$

d) Method of feasible directions

There exist special cases where the optimization problem require constraints to be accomplished at each iteration. Therefore, this methods were formulated. The most popular methods are Zoutendijk's method and Rosen's gradient projection method [7] .

2.4.1.2 Methods for Nonlinear Optimization problem

The most popular methods for the nonlinear optimization problem are Interior Point Methods and Sequential Quadratic Programming [39].

Active Set SQP Method

The fundamental idea in this method is to define at each step of the algorithm a set of constraints (named the "Working Set") as the Active Set. For each iteration, since the point x_k is fixed, the problem is reduced to an Linear programming (LP) with equality constraints. Also, the KKT conditions must hold for each solution. During the execution, the algorithm will eliminate or add constraints when it is possible to keep decreasing the objective function [12].

a) Quadratic programming (QP) algorithm

In this section, a typical QP problem of the following form is considered.

$$\min_{x} \quad \frac{1}{2}x^T Q x + c^T x \tag{2.17a}$$

$$\text{subject to} \quad a_i^T x = b_i, \quad i = 1, \ldots, m \tag{2.17b}$$

$$a_j^T x <= b_j, \quad j = m+1, \ldots, p \tag{2.17c}$$

The first m constraints are linear equalities and the remaining are less than type inequalities. Matrix Q will be assumed positive definite. The basic iteration has the following form:

$$x_{k+1} = x_k + \alpha_k d_k \tag{2.18}$$

The set of active constraint is defined as.

$$W^k = \{1, \ldots, m\} \cup \{i \in I \quad | a_i^T x = b_i \quad i = m+1, \ldots, p\} \tag{2.19}$$

To solve the direction d_k finding problem. The optimization problem (2.17) is reformulated as equality constrained QP and is to be solve in each iteration k. The new reformulation is presented as.

$$\min_{d} \quad \frac{1}{2}d_k^T Q d_k + g_k d_k \tag{2.20a}$$

$$\text{subject to} \quad a_i^T (x_k + d_k) = b_i, \quad i \in W^k \tag{2.20b}$$

Where $g_k = Qx_k + c$ and the KKT conditions can be written in a matrix form.

$$\begin{pmatrix} Q & \mathbf{A^T} \\ \mathbf{A} & \mathbf{0} \end{pmatrix} \begin{pmatrix} d_k \\ \lambda_\mathbf{k} \end{pmatrix} = \begin{pmatrix} -g_k \\ \mathbf{0} \end{pmatrix} \tag{2.21}$$

Where $A = [a_1, a_2, \ldots, a_i]^T \quad , \quad i \in W^k$ and $\lambda_\mathbf{k} = [\lambda_1, \lambda_2, \ldots, \lambda_i]^T \quad , \quad i \in W^k$ is the vector of Lagrange multipliers associated with the active constraint. The complete algorithm for the Active set method (ASM) for QP problems is detailed in the algorithm 2.

b) SQP algorithm

The SQP algorithm is one of the most effective methods for solving nonlinearly constrained optimization problems. The advantage of SQP is its ability to deal with nonlinear cost functions and constraints in small or large problems. At each iteration, the algorithm is linearized by using a Taylor series approximation around the point

Algorithm 2: Quadratic programming (QP) algorithm [12]

Input : Choose feasible point x_0 and the set of active constraint W_0 according to the x_0

Output : Solution point close x^*

Set k= 0
repeat
  Compute d_k and $\lambda_\mathbf{k}$ by solving (2.21)
  if $d_k = 0$ **then**
    if $\lambda_i \geq 0 \bigvee i \in W_k \cap I$ **then**
      STOP
    else
      $\lambda_j = min\{\lambda_i,\, i \in W_k\}$;
      $W_k = W_k - \{j\}$;
      GO TO step 3
  else
    Compute the step length α

$$\alpha_k = min\left[1, \frac{b_j - a^T x_k}{a_j d_k} \text{ for } a_j d_k > 0,\, j \notin W_k\right]$$

  if $\alpha_k < 1$ **then**
    Include restriction to the active working set
    $W_k = W_k + \{j\}$
  Set $x_{k+1} = x_k + \alpha d_k$;
  Set $W_{k+1} = W_k$;
  Set $k = k + 1$;
until *stopping criterion*;

(x_k, λ_k) . Then, it generates a QP sub-problem, which can be solved efficiently with the methods shown in the previous section(See Algorithm 2). In a similar way as The IPM, the general optimization problem (2.10) is considered. Therefore, at each iteration step k the search direction d_k is obtained by computing the following QP problem.

$$\min_d \quad \frac{1}{2} d_k^T H(x_k, \lambda_k) d_k [\nabla f(x_k)]^T \tag{2.22a}$$

$$\text{subject to} \quad h_i(x_k) + [\nabla h_i(x_k)]^T d_k = 0, \quad i = 1, \ldots, m \tag{2.22b}$$

$$g_j(x_k) + [\nabla g_j(x_k)]^T d_k \leq 0, \quad j = 1, \ldots, p \tag{2.22c}$$

Where H is the Hessian matrix of the Lagrangian that can be approximated using Broyden-Fletcher-Goldfarb-Shanno (method) (BFGS). The SQP algorithm is presented in (3).

Algorithm 3: Sequential quadratic programming (SQP) algorithm [12]

Input : Initial values x_0
Output : Optimal solution for x^*

Set $k = 0$.
repeat
  Compute d_k by solving the QP_k subproblem (2.22) using the algorithm 2.
  Find the step length α_k by a line search.
  Set $x_{k+1} = x_k + \alpha d_k$ and $\lambda_{k+1} = \lambda_k$.
  Set $k = k + 1$.
until *stopping criterion*;

Interior Point Methods

Interior Point Methods (IPM) are certain class of algorithm to solve to linear or non-linear convex optimization problems. This methods arises from the search of algorithm with better properties than simplex method, due to the simplex method can be inefficient on certain pathological problems [42]. The main idea of this approach is that IPM move inside the feasible region to reach the optimal solution. There exist diverse classes of Interior Point that depend on problem type for solving. In the literature, there are two main approaches the primal dual and primal barrier [39] that will be presented in the next section.

a) Primal-dual Methods

The main objective of primal-dual methods is to combine the simplicity in the formulation of the inequality constraints with an efficient computational performance. To solve optimization problems through the IPM, a perturbation parameter is introduced in the complementarily KKT [54]. Therefore, a modification in the KKT conditions are required to solve by using primal dual methods. The equations (2.28) are reformulated in equality constraints by using the slack variables $s \geq 0$ and the complementary condition (2.12d) is relaxed through the perturbation parameter $\tau \leq 0$ [19]. The modification of equations (2.12) are expressed as follow.

$$\bigtriangledown f(x) + \bigtriangledown \mathbf{h(x)}^T \lambda + \bigtriangledown \mathbf{g(x)}^T \mu = \mathbf{0} \tag{2.23a}$$

$$\mathbf{h(x)} = \mathbf{0} \tag{2.23b}$$

$$\mathbf{g(x)} + s = \mathbf{0} \tag{2.23c}$$

$$Zs - \tau e = \mathbf{0} \tag{2.23d}$$

$$(\mu, s, \tau) \geq \mathbf{0} \tag{2.23e}$$

Where $Z = diag(\mu_1, \mu_2, \ldots, \mu_p)$ and $e = \overbrace{[1, 1, \ldots, 1]}^{p}{}^T$. The equations (2.23) are iteratively solved by Newton method. Define the following function.

$$F_\tau(x, \lambda, \mu, s) = \begin{bmatrix} \bigtriangledown f(x) + \bigtriangledown \mathbf{h(x)}^T \lambda + \bigtriangledown \mathbf{g(x)}^T \mu \\ \mathbf{h(x)} \\ \mathbf{g(x)} + s \\ Zs - \tau e \end{bmatrix} = \begin{bmatrix} r_d \\ r_p \\ r_c \\ r_{sz} \end{bmatrix}$$

Therefore, the search direction d_k is calculated through the following equation for a fixed τ.

$$J_{F_\tau}(x_k, \lambda_k, \mu_k, s_k) d_k + F(x_k, \lambda_k, \mu_k, s_k) = \mathbf{0} \tag{2.24}$$

Where $d = [\triangle x_k, \triangle \lambda_k, \triangle \mu_k, \triangle s_k]^T$ and J is the Jacobian operator. The algorithm stops when the norm of the residual vector is less than a given small tolerance $||r_d|| \leq \epsilon_1$, $||r_p|| \leq \epsilon_2$, $||r_c|| \leq \epsilon_3$ and $||r_{sz}|| \leq \epsilon_4$.
Moreover, the parameter of perturbation τ must converge to zero during the iterations. This parameter can be computed through the primal-dual distance defined as m, this distance is defined by (2.23d) and can be formulated as a function of slack variables s. Moreover, the variable σ is added that is known as parameter of the direction combination, which defines the trajectory of the optimal solution [19]. Therefore, τ is expressed as follow.

$$\tau = \sigma m \tag{2.25}$$

Where $m = \sigma \frac{\mu^T s}{p}$. In order to select the value of σ. Two cases must be analyzed. In the case value of $\sigma = 0$, which corresponds to the affine-scaling direction. In this approach, the solution is obtained thorough the non-perturbed solution of the KKT conditions. On other hand, when $\sigma = 1$, which corresponds to the centralization direction. In this

approach, the optimal point is solved, with a primal dual distance m equal to the initial value of τ.
In order to decrease the value of τ during each iteration, it is necessary that $0 \leq \sigma \leq 1$ [8]. The primal-dual method is summarized in Algorithm 4.

Algorithm 4: Primal-Dual Interior Point Algorithm [75]

Input : Strictly feasible choose the starting point x_0, λ_0, μ_0, s_0 and $\sigma \in (0, 1)$
Output : Solution point close x^*

Set k= 0
repeat
  Set $\tau_k = \sigma \frac{\sum_{i=0}^{p} \mu_{ik} s_{ik}}{p}$
  Compute $\triangle x_k$, $\triangle \lambda_k$, $\triangle \mu_k$, and $\triangle s_k$ by solving (2.24)
  Find the step length α_k by a line search such that $s_{k+1} \leq 0$ and $\mu \leq 0$
  Set $(x_{k+1}, \lambda_{k+1}, \mu_{k+1}, s_{k+1}) = (x_k, \lambda_k, \mu_k, s_k) + \alpha_k(\triangle x_k, \triangle \lambda_k, \triangle \mu_k, \triangle s_k)$
  Set $k = k + 1$
until $||r_d|| \leq \epsilon_1 \& ||r_p|| \leq \epsilon_2 \& ||r_c|| \leq \epsilon_3 \& ||r_{sz}|| \leq \epsilon_4$;

b) Primal-Barrier Methods

As mentioned earlier, the main idea of a barrier method is to use a barrier function $B(x, \mu)$ for removing the inequality constraint (4.2c). A common approach is to use a logarithmic barrier function, which has the property of being continuous, differentiable and convex [20], [39].

$$B(x, r_k) = f(x) + r_k[\sum_{j=0}^{p} Ln(s_j)] \tag{2.26}$$

Where r_k is the barrier parameter. the terms Ln are defined at the points x for which $x \geq 0$ and $g(x) \leq 0$. Thereby, consider now the following parametric problem.

$$\min_{x} \quad B(x, r_k) \tag{2.27a}$$

$$\text{subject to} \quad h_i(x) = 0 \quad i = 1, \ldots, m \tag{2.27b}$$

$$g_j(x) + s_j = 0 \quad j = 1, \ldots, p \tag{2.27c}$$

Now, the objective is to find the optimal solution x for a fixed value of r_k. The problem is solved by the KKT conditions. To find the minimum point of vector x is necessary to solve the next system.

$$\bigtriangledown f(x) + \bigtriangledown \mathbf{h(x)}^T \lambda + \bigtriangledown \mathbf{g(x)}^T \mu = \mathbf{0} \tag{2.28a}$$

$$r_k \mathbf{S}^{-1} e + \mu = \mathbf{0} \tag{2.28b}$$

$$\mathbf{h(x)} = \mathbf{0} \tag{2.28c}$$

$$\mathbf{g(x)} + \mathbf{S} = \mathbf{0} \tag{2.28d}$$

Where $e = (\overbrace{1, \ldots, 1}^{p})$ and $S = diag(s1, \ldots, s_p)$, λ and μ are the Lagrange multiplier. The equations system is commonly solved by Newton-Method for a fixed r_k.

$$F_{r_k}(x, s, \lambda, \mu) = \begin{bmatrix} \bigtriangledown f(x) + \bigtriangledown \mathbf{h(x)}^T \lambda + \bigtriangledown \mathbf{g(x)}^T \mu \\ r_k \mathbf{S}^{-1} e + \mu \\ \mathbf{h(x)} \\ \mathbf{g(x)} + \mathbf{S} \end{bmatrix} = \begin{bmatrix} r_p \\ r_s \\ r_\lambda \\ r_\mu \end{bmatrix}$$

In a similar way as the equation (2.23) , the search direction d_k can be computed by using the following equation for a fixed r_k.

$$J_{F_{r_k}}(x_k, s_k, \lambda_k, \mu_k) d + F(x_k, s_k, \lambda_k, \mu_k) = 0 \tag{2.29}$$

Where $d = [\triangle x_k, \triangle \lambda_k]^T$. In this case, the algorithm is based on solving the equality constrained minimization problem decreasing the barrier parameter μ at each iteration. As r_k approaches zero, the solution of (2.27) converges to the solution. The primal barrier method is summarized in Algorithm 5

Algorithm 5: Primal Barrier Methods [75]

Input : Strictly feasible choose the starting point x_0, λ_0 and $\mu_0 \geq 0$
Output : Solution point close x^*

```
Set k= 0
repeat
    Compute x_{k+1} and λ_{k+1} for a fixed r_k by solving (2.29).
    if r_k ≤ ϵ then
        STOP
    end
    Choose r_{k+1} ∈ (0, r_k)
    Set k = k + 1
until stopping criterion;
```

Chapter 3

Chance Constrained Stochastic Model Predictive Control

DMPC has been widely developed and studied due to exceptional results in an optimization problem with constraint. Moreover, DMPC can assure a certain degree of robustness to system uncertainties due to its receding horizon. However, its deterministic framework is inadequate for systems in presence of uncertainties. Therefore, it is required to consider the possible a-priori knowledge of the statical properties of the random variables such as mean, covariance or probability distribution function. The uncertainties can be considered as constant or time dependent, as shown in the Figure 3.1 [51]. Their stochastic properties can be obtained base on analysis of historical data or even experiences of experts.

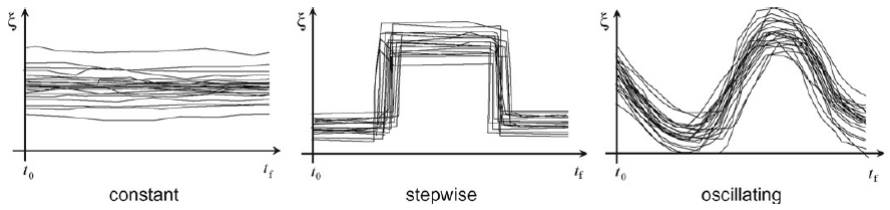

Figure 3.1 – Different forms of uncertain variables [51]

3.1 SMPC for Autonomous Vehicle

In the last decades, chance constrained optimization methods has been studied in AV. In [13, 14], a novel method for optimal control of nonlinear systems in presence of uncertainties is presented, in which Sample Average Approximation (SAA) is employed for the simulation of stochastic variables.

Linearization at each iteration is proposed in [50], where the error characteristics of the motion model are considered and the obstacles is formulated as a set of linear equations that allows the optimization problem to be solved as QP, so that it reduces the computation time. Other work using LTV is presented in [55], in which dynamic obstacles are considered.

In order to guarantee the safety, the driver's behavior are modeled in [38, 53, 64] in which the probability constraint are computed by using Chebyshev's inequality [31]. Other interesting formulation in AV, in which the controller requires an environment model, where the AV is operating. For instance, in [18] an integration of an environment model with a stochastic predictive control is formulated (Figure 3.2). Moreover, double lane change with uncertainties was tested in [41], in which the stochastic variables arises from noise affected motion. Robust MPC has also been used in AV. In [32], the uncertainties are considered in the equation model based on experiments. Two scenarios are proposed, in which the vehicle is tested with obstacles and the friction coefficient is considered as random variable.

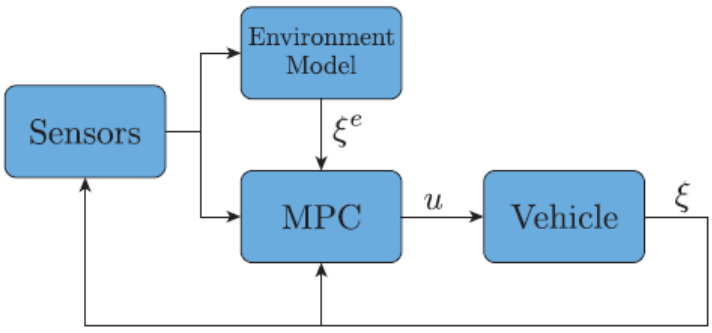

Figure 3.2 – Block diagram of overall system proposed in [18]

3.2 Stochastic Linear Model Predictive Control

The main idea of optimization under uncertainty is to integrate the available stochastic information into optimization problem formulation. The majority of studies in the literature on SMPC deals with stochastic linear systems [29]. In this approach, many SMPC algorithms have been developed considering linear systems with additive or

multiplicative uncertainties model [29, 30]. In the case of additive uncertainty takes the following form.

$$x_{k+1} = Ax_k + Bu_k + W\xi_k, \quad \text{with} \quad \xi \sim N(\mu, \sum) \tag{3.1}$$

Where A, B and W are the state-space matrices, $x_k \in R^n$, $u_k \in R^m$ and $\xi_k \in R^l$ are state variables, control variables and stochastic variables, respectively. In addition, μ and $\sum$ are the vector of expected value and the covariance matrix of stochastic variables ξ, respectively. In the case of systems with multiplicative uncertainty the model are described as follow.

$$x_{k+1} = Ax_k + Bu_k + \sum_{i=0}^{q}(C_i x_k + D_i u_k)\xi_k^i \quad \text{with} \quad \xi \sim N(\mu, \sum) \tag{3.2}$$

3.2.1 LMPC problem formulation

In this approach, likewise to deterministic case is considered a quadratic convex objective function for a linear discrete time system with probabilistic constraint. The objective function is formulated for solving regulator control problem as in the case of deterministic MPC [29, 30, 40, 77]. However, when the stochastic variables are taken into account it is necessary to relax the objective function . The majority of cases the expected values is merely considered in the objective function. The framework of the problem formulation is the following.

$$\min_{\mathbf{x_k}, \mathbf{u_k}} \quad E[x_N^T \mathbf{P} x_N] + \sum_{k=0}^{N-1} E[x_k^T \mathbf{Q} x_k + u_k^T \mathbf{R} u_k] \tag{3.3a}$$

$$\text{subject to} \quad x_{(0)} = x_0 \tag{3.3b}$$

$$x_{k+1} = Ax_k + Bu_k + W\xi_k \quad k = 0, 1, \ldots, N-1 \tag{3.3c}$$

$$u_{min} \leq u_k \leq u_{max} \quad k = 0, 1, \ldots, N-1 \tag{3.3d}$$

$$Pr[x_{kmin} \leq x_k \leq x_{kmax}] \leq \alpha_k \quad k = 1, 2, \ldots, N \tag{3.3e}$$

Where $\mathbf{P}$, $\mathbf{Q}$ and $\mathbf{R}$ are positive definite constant matrices. $\phi := E[x_N^T \mathbf{P} x_N]$ is the terminal cost function and $L := \sum_{k=0}^{N-1} E[x_k^T \mathbf{Q} x_k + u_k^T \mathbf{R} u_k]$ is the stage cost function over the prediction horizon. Consider the equation (3.3c) in the formulation of stochastic variable with a prediction horizon N and the probability constraint are formulated in the equation (3.3e), where x_{kmin} and x_{kmax} are the bound of the state profiles over the prediction horizon.

3.2.1.1 Control Strategies

According to the [29] , the randomized methods have widely been employed for the linear case. The main features of these methods will be presented in this section.

Open loop control

In this approach, the control strategy is an open-loop control. The lineal system (3.3c) with additive uncertainties is formulated. Now, Assuming that at the sampling instant t_0 the state variable vector is available through measurement or estimate, the state x_0 provides the current plant information. The future control trajectory, the state variable, and stochastic variable are denoted by $\mathbf{U}, \mathbf{X}$ and ξ, respectively.

$$\mathbf{U} = [u_0, u_1, \ldots, u_{N-1}]^T \tag{3.4a}$$

$$\mathbf{X} = [x_1, x_2, \ldots, x_N]^T \tag{3.4b}$$

$$\xi = [\xi_0, \xi_1, \ldots, \xi_{N-1}]^T \tag{3.4c}$$

Hence, the system (3.3c) over prediction horizon can be rewritten as follow.

$$\mathbf{X} = \mathbf{A}x_0 + \mathbf{B}\mathbf{U} + \mathbf{W}\xi \tag{3.5}$$

Where the matrices $\mathbf{A}, \mathbf{B}$ and $\mathbf{W}$ are given by.

$$\mathbf{A} = \begin{bmatrix} A \\ A^2 \\ A^3 \\ \vdots \\ A^N \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} B & 0 & 0 & \cdots & 0 \\ AB & B & 0 & \cdots & 0 \\ A^2B & AB & B & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & A^{N-3}B & \cdots & B \end{bmatrix}$$

$$\mathbf{W} = \begin{bmatrix} W & 0 & 0 & \cdots & 0 \\ AW & W & 0 & \cdots & 0 \\ A^2W & AW & W & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A^{N-1}W & A^{N-2}W & A^{N-3}W & \cdots & W \end{bmatrix}$$

For notational convenience $\mathbf{A} \in R^{nN \times n}$, $\mathbf{B} \in R^{nN \times nN}$ and $\mathbf{W} \in R^{lN}$. Substituting (3.5) into (3.3a) and neglecting the terms that do not contain $\mathbf{U}$. the objective function is rewritten as follow.

$$\min_{\mathbf{U}} \quad J = \mathbf{U}^T(\mathbf{B}^T\mathbf{Q}\mathbf{B} + \mathbf{R})\mathbf{U} + 2(\mathbf{A}x_0 + \mathbf{W}E[\xi])^T\mathbf{Q}\mathbf{B}\mathbf{U} \tag{3.6}$$

Note that the minimization of J has been reduced to the quadratic programming with respect to $\mathbf{U}$. In this approach, the variance of ξ evolves in an uncontrolled fashion, when the system is unstable, this strategy has significant drawbacks, since it may induce serious feasibility problems [29, 37].

Disturbance feedback control

In this approach, the lineal system (3.3c) with additive uncertainties is widely formulated. The input sequence u_k is defined as a function of w_k. As a result, the control input can be directly parameterized as an affine function of the disturbance. The common representation is formulated as follow [37, 63].

$$u_k = v_k + \sum_{j=0}^{k-1} M_{k,j}\xi_j \quad k = 0, 1, \ldots, N-1 \tag{3.7}$$

Which $M_{k,j} \in R^{mxl}$ and $v_k \in R^m$. Note that the past disturbance sequence can be calculated through the equation (3.3c) as the difference between the predicted and actual state at each step.

$$\xi_k = x_{k+1} - Ax_k - Bu_k \tag{3.8}$$

Hence, the control strategy is equivalent to an open-loop control system with feedforward disturbance compensator. The equations (3.7) and (3.8) can be rewritten using the prediction horizon as follow.

$$\mathbf{X} = \mathbf{F}x_0 + \mathbf{G}\mathbf{U} + \mathbf{H}\xi \tag{3.9a}$$

$$\mathbf{U} = \mathbf{M}\xi + \mathbf{V} \tag{3.9b}$$

Where the matrices $\mathbf{F}, \mathbf{G}$ and $\mathbf{H}$ are given by.

$$\mathbf{F} = \begin{bmatrix} A \\ A^2 \\ A^3 \\ \vdots \\ A^N \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} B & 0 & 0 & \cdots & 0 \\ AB & B & 0 & \cdots & 0 \\ A^2B & AB & B & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & A^{N-3}B & \cdots & B \end{bmatrix}$$

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ A & 1 & 0 & \cdots & 0 \\ A^2 & A & & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A^{N-1} & A^{N-2} & A^{N-3} & \cdots & 1 \end{bmatrix}$$

For notational convenience $\mathbf{F} \in R^{nNxn}$, $\mathbf{G} \in R^{nNxnN}$ and $\mathbf{H} \in R^{lxN}$. Whereas matrices $\mathbf{M}$ and $\mathbf{V}$ that contains the control law parameter are given by.

$$\mathbf{V} = \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ \vdots \\ v_{N-1} \end{bmatrix}$$

$$\mathbf{M} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ M_{1,0} & 0 & 0 & \cdots & 0 \\ M_{2,0} & M_{2,1} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ M_{N-1,0} & M_{N-1,1} & M_{N-1,2} & \cdots & M_{N-1,N-1} \end{bmatrix}$$

Where, we consider $\mathbf{V} \in R^{lN}$ and $\mathbf{M} \in R^{mN \times lN}$. Substituting (3.9a) and (3.9b) into (3.3a) and neglecting the terms that do not contain U. the objective function is rewritten as follow.

$$\min_{\mathbf{V},\mathbf{M}} \alpha^T \mathbf{Q}\alpha + 2\alpha^T \mathbf{Q}\tau.\gamma 2 + tr[\tau^T \mathbf{Q}\tau.\gamma_1] + \mathbf{V}^T \mathbf{R}\mathbf{V} + 2\mathbf{V}^T \mathbf{R}\mathbf{M}\gamma_2 + tr[\mathbf{M}^T \mathbf{R}\mathbf{M}.\gamma_1] \tag{3.10a}$$

$$\alpha_1 = \mathbf{F}x_0 + \mathbf{G}\mathbf{V} \tag{3.10b}$$

$$\tau = \mathbf{H} + \mathbf{G}\mathbf{M} \tag{3.10c}$$

$$\gamma_1 = E[\xi\xi^T] \tag{3.10d}$$

$$\gamma_2 = E[\xi] \tag{3.10e}$$

Where $tr[]$ is the operator which indicates the trace of a matrix. The objective function is reduced to (3.10), note that the results is a convex function of the control policy parametrization of $\mathbf{V}$ and $\mathbf{M}$ [63].

State feedback control

In this approach the control variable is defined in terms of state variable. Assume the pair (A, B) of the system (3.3c) can be stabilized, there exist a linear control law Kx. Now, we define a nominal system and the linear system.

$$\bar{x}_{k+1} = A\bar{x}_k + B\bar{u}_k \tag{3.11a}$$

$$x_{k+1} = Ax_k + Bu_k + W\xi_k \tag{3.11b}$$

The most common formulation of control law is represented as follow [29, 30].

$$u_k = \bar{u}_k + K(x_k - \bar{x}_k) \tag{3.12}$$

Define a new state variable $e_k = x_k - \bar{x}_k$ as the deviation between the linear system and the nominal system . The error system under the linear control law is denoted by.

$$e_{k+1} = \overbrace{(A + BK)}^{A_{cl}} e_k + W\xi_k \tag{3.13}$$

Now, the equation (3.13) can be rewritten using prediction horizon assuming that at the sampling instant $k_i > 0$ the state variable vector is available through measurement or estimate. The error variable trajectory and the stochastic variable are denoted by $\mathbf{E}$, ξ, respectively.

$$\mathbf{E} = [e_{k_i}, e_{k_i+1}, \ldots, e_{k_i+N-1}]^T \tag{3.14a}$$

$$\xi = [\xi_{k_i}, \xi_{k_i+1}, \ldots, \xi_{k_i+N-1}]^T \tag{3.14b}$$

$$\mathbf{E} = \mathbf{A_{cl}} e_{k_i} + \mathbf{J}\xi \tag{3.15}$$

Where the matrices **Acl** and **J** are given by.

$$\mathbf{A_{cl}} = \begin{bmatrix} A_{cl} \\ A_{cl}^2 \\ A_{cl}^3 \\ \vdots \\ A_{cl}^N \end{bmatrix}$$

$$\mathbf{J} = \begin{bmatrix} W & 0 & 0 & \cdots & 0 \\ A_{cl}W & W & 0 & \cdots & 0 \\ A_{cl}^2W & A_{cl}W & W & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{cl}^{N-1}W & A_{cl}^{N-2}B & A_{cl}^{N-3}W & \cdots & W \end{bmatrix}$$

In this approach a proper choice of the control gain , which is an optimization variable or a design parameter, can reduce the effect the stochastic variable sequence ξ. This results in a larger feasibility region with respect to the approach of open-loop-control, although that the system is unstable [29, 30].

3.3 Stochastic nonlinear Model Predictive Control

3.3.1 Formulation of a stochastic optimization problem

The model predictive formulation of a chance constrained dynamic optimization problem takes the form.

$$\min_{x_k,u_k} \quad J(x_{k+1}, u_k, \xi_k) \tag{3.16a}$$

$$\text{subject to} \quad x(0) = x_0 \tag{3.16b}$$

$$x_{k+1} = f_k(x_k, u_k, \xi_k) \qquad k = 0, 1, \ldots, N-1 \tag{3.16c}$$

$$u_{min} \leq u_k \leq u_{max} \qquad k = 0, 1, \ldots, N-1 \tag{3.16d}$$

$$Pr[g(x_{k+1}, u_k, \xi_k) \leq 0] \geq \alpha \qquad k = 0, 1, \ldots, N-1 \tag{3.16e}$$

In which x_k are the state variables, u_k are the control variables, ξ_k are the uncertain variables and x_0 are the initial values of the state variables. J is the objective function, the vectors g and h represent the equality (model equation) and inequality constraint, respectively.
Due to the presence of the uncertainties ξ, the problem cannot be solved directly with the available deterministic optimization methods. The problem has to be relaxed into equivalent deterministic problem. It is well-known in the literature as relaxation. To treat the new objective function J, minimizing the expected values and the variances of the objective function f has usually been adopted. Sometimes, the cost function in a stochastic framework is the following.

$$\min_{x_k,u_k} \{J = E[f(x_{k+1}, u_k, \xi_k)] + wD[f(x_{k+1}, u_k, \xi_k)]\} \tag{3.17}$$

Where E and D are the operators of expectation and variance, respectively. The value of w is a weighting factor between the two terms. In the case of the relaxation of the inequality constraint will be formulated as chance or probability constraints. It means holding the inequality constraint with a predefined probability level. There exist two different forms of representing the chance constraints [51]. In the equation (3.18) and (3.19) depict single chance constraint and joint chance constraint, respectively. Pr is probability operator and α is the reliability level probability.

$$Pr[g(x_{k+1}, u_k, \xi_k) \leq 0] \geq \alpha_k, \quad k = 0, 1, \ldots, N-1 \tag{3.18}$$

$$Pr[g(x_{k+1}, u_k, \xi_k) \leq 0, \quad k = 0, 1, \ldots, N-1] \geq \alpha \tag{3.19}$$

In the case of single chance constraint, individual probabilities have to ensure that each inequality will be hold, whereas in the joint chance constraint the inequalities should be satisfied simultaneously with a given probability.

3.4 Approximation Methods

The major difficulty for solving chance constrained optimization problems is in to evaluate the probability constraint. In general, the state variables depend on the control variables and stochastic variables while the control variables are considered deterministic. Therefore, the state variables can be expressed in terms of the control and stochastic variables using the non-linear dynamic equation (3.16b) and initial conditions (3.16c). As a result, the formulation (3.16) can be rewritten as follow.

$$\min_{u_k} \quad E\left[f(x_k(u_k, \xi_k), u_k)\right] \tag{3.20a}$$

$$u_{min} \leq u_k \leq u_{max} \qquad k = 0, 1, \ldots, N-1 \tag{3.20b}$$

$$p(u) = Pr[g(x_{k+1}(u_k, \xi_k), u_k) \leq 0] \geq \alpha \qquad k = 0, 1, \ldots, N-1 \tag{3.20c}$$

The main challenge for solving CCOPT is to compute the probability $p(u)$ of (3.20c). This computation is more complicated task if the gradient $\bigtriangledown p(u)$ is required for the numerical solver. As a result, the approximation methods have been proposed for a better tractability of CCOPT. For instance, **Back mapping method**, **Robust optimization**, **Analytic Approximation** and **SAA** have been widely developed in [13, 34, 52]. Therefore, in this work the new analytic approximation strategy formulated in [33] using a parametric function will be employed.

The idea of this method is based on SAA approach that replaces the probability constraint (3.20c) by a relative-frequency count of satisfaction of the constraint $g(x_k(u_k, \xi_k), u_k) > 0$ for generated samples QSM $[\xi_1, \xi_2, \ldots, \xi_{Ns}]$ with low-discrepancy properties.

The approximation of the probability distribution of random variables is considered as a viable method for better tractability of CCOPT [13, 33, 34, 49]. The expected value and the probability can be calculated as follow.

$$E[g(\xi)] = \int g(\xi)\rho(\xi)d\xi \tag{3.21}$$

$$P_F = \int_{g(\xi \in F)} \rho(\xi) d\xi \tag{3.22}$$

Where $\rho(\xi)$ is the probability distribution function of the random variables $\xi_1, \xi_2, \ldots, \xi_{N_s}$ and P_F is the probability of the certain event $\xi \in F$. This expressions can be approximated by.

$$E[g(\xi)] \simeq \frac{1}{N_s} \sum_{i=1}^{N_s} g(\xi)(\xi_i) \tag{3.23}$$

$$P_F \simeq E[\Theta(\xi)] \tag{3.24}$$

Hence, the probability constraint can be expressed using the expected value ($Pr[g(u_k, \xi_k) > 0] = E[\Theta(g(u_k, \xi_k))]$.

$$\Theta(g(u_k, \xi_k)) = \begin{cases} 0 & g(u_k, \xi_k) \leq 0 \\ 1 & g(u_k, \xi_k) > 0 \end{cases}$$

Where $g(u_k, \xi_k) := g(x_{k+1}(u_k, \xi_k), u_k)$. Nevertheless, the drawback is that the function Θ is discontinuous and is not amenable for the numerical computation. Moreover, the feasibility of the obtained solution is guaranteed only if the number of samples N_s is very large. For this reason, the analytic approximations approach replaces this discontinuous function Θ by a parametric function Ψ that could be smooth and assure a-priori the feasibility. In order to formulate a new parametric function, it is required that the function Ψ satisfies the following properties [33].

- $P1$: $E[\Theta(g(u_k, \xi_k))] \leq \Psi(\tau, u_k)$ for each $0 < \tau < 1$.
- $P2$: $inf_{\tau > 0} \Psi(\tau, u_k) = E[\Theta(g(u_k, \xi_k))]$ for each u_k.
- $P3$: $\Psi(\tau, u_k)$ is non-decreasing with respect to τ.

In the literature, there exist suggestion above the parametric function that accomplishes some of the properties $P1 - P3$ in [56, 58, 62, 68]. Under this assumption it is possible to provide a good approximation to the solution of (3.16) for a decreasing sequence of $\tau \to 0^+$.
The next section two approaches of the new parametric function Ψ proposed in [33] will presented.

3.4.1 Analytical Inner Approximation (IA)

The Analytical IA approximates the above stochastic inequality constraint by a deterministic one, which is defined as follows.

$$\Psi(\tau, g(u_k, \xi)) = \frac{1 + m_1\tau}{1 + m_2\tau \exp\left(\frac{-1}{\tau} g(u_k, \xi)\right)} \tag{3.25a}$$

$$Pr[g(u_k, \xi_k) > 0] = E[\Psi(\tau, g(u_k, \xi))] < 1 - \alpha \tag{3.25b}$$

Where $0 < \tau < 1$ is the approximation parameter, and $m1$, $m2$ are positive values such that $m1 > m2$. The expected value can be obtained using quasi-Monte-Carlo sampling. In this way, using the quasi-sequential approach, the chance-constrained optimization problem (3.16) is transformed into a deterministic OCP which can be solved using standard NLP. Therefore, on each prediction horizon the deterministic OCP must be solved.

Other important consideration is the feasibility, where the feasible set of CCOPT in the formulation (3.16) is $\mathcal{P} = \{u \in \mathcal{U} \mid Pr\{u\} \geq \alpha\}$ and the feasible set of CCOPT using the parametric function is $\mathbf{M}(\tau) = \{u \in \mathcal{U} \mid \Psi(\tau, u) < 1 - \alpha\}$. As we can see in the Figure 3.3 $\mathbf{M}(\tau)$ is a subset of the feasible set of $\mathcal{P}$ and **the sense of the convergence** of the sets when $\tau \to 0^+$.

Therefore, the new formulations for IA is presented as follow.

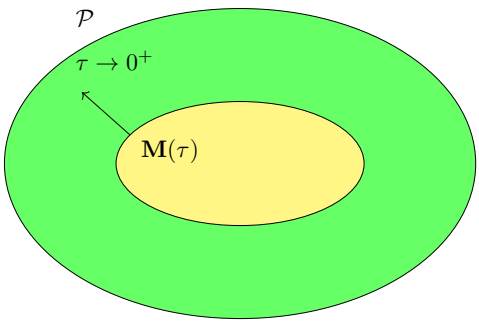

Figure 3.3 – Inner Approach-Convergence $\lim_{\tau \to 0^+} \mathbf{M}(\tau) = \mathcal{P}$ [33]

$$\min_{u_k} \quad \frac{1}{N_s}\sum_{j=1}^{N_s}(f(x_{k,j}(u_k,\xi_{k,j}),u_k)) \tag{3.26a}$$

$$u_{min} \leq u_k \leq u_{max} \qquad k=0,1,\ldots,N-1 \tag{3.26b}$$

$$\frac{1}{N_s}\sum_{j=1}^{N_s}\Psi(\tau, g(x_{k+1,j}(u_k,\xi_{k,j}),u_k)) \leq 1-\alpha \qquad k=0,1,\ldots,N-1 \tag{3.26c}$$

3.4.2 Analytical outer approximation (OA)

Similar to the IA, the OA approach approximates the chance constraint by the following deterministic inequality.

$$\Psi(\tau,-g(u_k,\xi)) = \frac{1+m_1\tau}{1+m_2\tau\exp\left(\frac{1}{\tau}g(u_k,\xi)\right)} \tag{3.27a}$$

$$Pr[g(u_k,\xi_k)\leq 0] = E[\Psi(\tau,-g(u_k,\xi))] \geq \alpha \tag{3.27b}$$

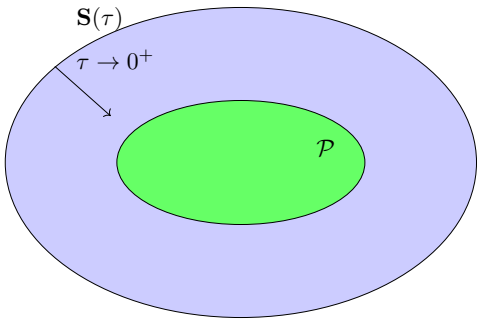

Figure 3.4 – Outer Approach-Convergence $\lim_{\tau\to 0^+}\mathbf{S}(\tau)=\mathcal{P}$ [33]

In this approach,the feasible set of CCOPT in the formulation (3.16) is $\mathcal{P}=\{u\in\mathcal{U}\mid Pr\{u\}\geq\alpha\}$ and the feasible set of CCOPT using the parametric function is $\mathbf{S}(\tau)=\{u\in\mathcal{U}\mid\Psi(\tau,u)\geq\alpha\}$. As we can see in the Figure 3.4 $\mathcal{P}$ is a subset of the feasible set of $\mathbf{S}(\tau)$ and **the sense of the convergence** of the sets when $\tau\to 0^+$. Therefore, the new formulations for OA is presented as follow.

$$\min_{u_k} \quad \frac{1}{N_s} \sum_{j=1}^{N_s} (f(x_{k,j}(u_k, \xi_{k,j}), u_k)) \tag{3.28a}$$

$$u_{min} \leq u_k \leq u_{max} \qquad k = 0, 1, \ldots, N-1 \tag{3.28b}$$

$$\frac{1}{N_s} \sum_{j=1}^{N_s} \Psi(\tau, -g(x_{k+1,j}(u_k, \xi_{k,j}), u_k)) \geq \alpha \qquad k = 0, 1, \ldots, N-1 \tag{3.28c}$$

Chapter 4

Implementation Framework

The implementation of the SMPC using IA and OA is based on C++ programming language to obtain high performance in run-time. The optimization task was carried using IpOpt and 10 000 samples were generated for stochastic variables using QSM with Sobol sequence. All the computations were carried out on a Intel Core I7-4960X running at 3.4 GHz and with 16 GB of RAM.
As mentioned above, to solve a chance constrained MPC problem in each iteration a CCOPT problem must be solve as defined 3.16 in the chapter 3. These types of problems are characterized by having probability constraints. Therefore, the objective of the methods of approximation is to transform these probability constraints. IA and OA transform the probability constraint into inequalities that are formulated according to the expected value 3.25 and 3.27. Finally, to transform into a deterministic problem, QSM samples are used. Hence, the expected value and the probability are approximated and the optimization problem as formulated as 3.26 and 3.28. Therefore, this problem can be solved by available solver. As mentioned above,in this work IpOpt solver based on interior point methods will used for the case studies proposed.

4.1 IpOpt Solver

IpOpt [73] is an open source software package that can be used to solve general NLP problems of the following form.

$$\min_{\mathbf{x}} \quad f(\mathbf{x}) \tag{4.1a}$$

$$\text{subject to} \quad g_l \leq g_j(x) \leq g_u, \quad j = 1, \ldots, m \tag{4.1b}$$

$$x_l \leq x_i \leq x_u, \quad i = 1, \ldots, n \tag{4.1c}$$

Where $\mathbf{x} \in \Re^n$ are the optimization variables, x_l and x_u are the lower and upper bounds, respectively. The function g is the set of constraints. The IpOpt implements an interior-point line-search filter method, the algorithms and their implementation can be found in [57, 69, 72, 73] . To implement the CCOPT with SMPC in IpOpt with program code. It is required to formulate the CCOPT into a sequence NLP problem on moving horizon. Then, according to the [74], the user must provide to IpOpt the following information:

- Number of optimization variables, number of constraint and bounds.
- Starting Point.
- Function values $f(\mathbf{x_k})$ and $g(\mathbf{x_k})$.
- First derivatives $\nabla f(\mathbf{x_k})$ and $\nabla g(\mathbf{x_k})$.
- Second derivatives $H(\mathbf{x_k}, \lambda_\mathbf{k})$ (the Hessian matrix of the Lagrangian).
- Number of non-zeros and sparsity structure of the $\nabla g(\mathbf{x_k})$ and $H(\mathbf{x_k}, \lambda_\mathbf{k})$.

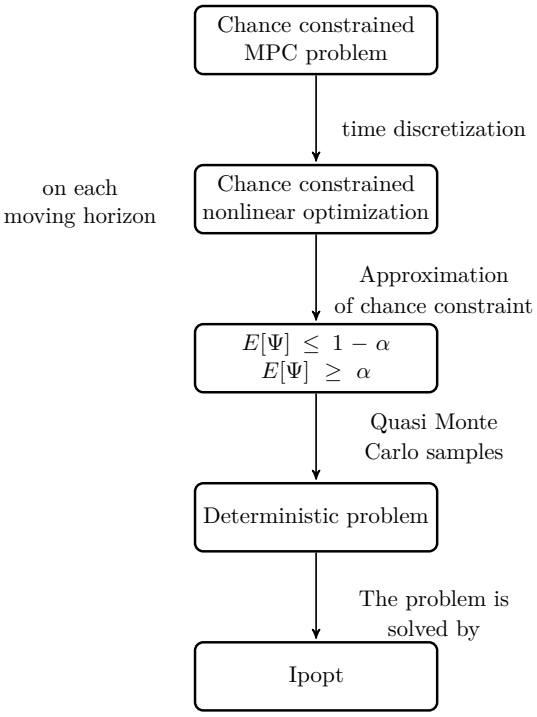

Figure 4.1 – Diagram of the solution procedure of chance constrained MPC problem

From the formulation obtained in the previous chapter (3.26) and (3.28), it is possible to represent the NLP problem as follow.

$$\min_{\mathbf{w}} \quad \mathbf{F}(\mathbf{x}(\mathbf{w}), \mathbf{w}) \tag{4.2a}$$

$$\text{subject to} \quad \mathbf{w_{min}} \leq \mathbf{w} \leq \mathbf{w_{max}}, \tag{4.2b}$$

$$G(\mathbf{x}(\mathbf{w}), \mathbf{w}) \leq \mathbf{0}, \tag{4.2c}$$

Where $\mathbf{w} = [u_0, u_1, \ldots, u_{N-1}]^T$ is the set of optimization variable, while the set of state variables is defined as.

$$\mathbf{x} = \begin{bmatrix} x_{(1,1)} \\ x_{(1,2)} \\ \vdots \\ x_{(1,N_s)} \\ x_{(2,1)} \\ \vdots \\ x_{(2,N_s)} \\ \vdots \\ \vdots \\ x_{(N,1)} \\ \vdots \\ x_{(N,N_s)} \end{bmatrix}$$

Where N is the prediction horizon and N_s is the number of samples generated by QSM method. As as result, there are $N \times N_s$ states variables. On the other hand, F represents the objective function and the set of constraint is defined as.

$$\mathbf{F}(\mathbf{x}(\mathbf{w}), \mathbf{w}) = \frac{1}{N_s} \sum_{j=1}^{N_s} f(\mathbf{x_j}(\mathbf{w}), \mathbf{w}) \tag{4.3}$$

$$\mathbf{G}(\mathbf{x}(\mathbf{w}), \mathbf{w}) = \begin{bmatrix} \frac{1}{N_s} \sum_{j=1}^{N_s} \Psi(\pm\tau, g(x_{(1,j)}, \mathbf{w})) \\ \frac{1}{N_s} \sum_{j=1}^{N_s} \Psi(\pm\tau, g(x_{(2,j)}, \mathbf{w})) \\ \frac{1}{N_s} \sum_{j=1}^{N_s} \Psi(\pm\tau, g(x_{(3,j)}, \mathbf{w})) \\ \vdots \\ \frac{1}{N_s} \sum_{j=1}^{N_s} \Psi(\pm\tau, g(x_{(N,j)}, \mathbf{w})) \end{bmatrix} \tag{4.4}$$

Where Ψ is the parametric smooth function presented in the previous chapter. As can be seen, N constraints are formulated that represents the probability constraint transformed as deterministic inequalities constraint.

4.1.1 Computation of the first derivatives

As have mentioned, only the control variables are considered as the optimization variables. Therefore, the gradient ∇F of the objective function and the Jacobian ∇G in the problem (4.2) can be formulated as follow.

$$\nabla F = \frac{\partial \mathbf{F}}{\partial \mathbf{w}} + \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \mathbf{w}} \tag{4.5a}$$

$$\nabla \mathbf{G} = \frac{\partial \mathbf{G}}{\partial \mathbf{w}} + \frac{\partial \mathbf{G}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \mathbf{w}} \tag{4.5b}$$

Where the sensitivities $\frac{\partial \mathbf{F}}{\partial \mathbf{w}}$, $\frac{\partial \mathbf{F}}{\partial \mathbf{x}}$, $\frac{\partial \mathbf{G}}{\partial \mathbf{w}}$, $\frac{\partial \mathbf{G}}{\partial \mathbf{x}}$ are computed using CASADI [6] that is an Automatic Differentiation software with interface C++, while $\frac{\partial \mathbf{x}}{\partial \mathbf{w}}$ is obtained by discretized dynamic equation. The model equations of the studies cases are discretized by using Runge Kutta 4th order.

$$\begin{aligned} \mathbf{H}(x_{(k+1,j)}, u_k) &= 0 \\ \mathbf{H}(x_{(k+1,j)}, u_k) = x_{(k+1,j)} - f_{k,j}(x_{(k,j)}, u_k, \xi_{(k,j)}) \quad & k = 0, 1, \ldots, N-1\,. \\ & j = 1, 2, \ldots, N_s \end{aligned} \tag{4.6}$$

$$\nabla \mathbf{H} = \frac{\partial \mathbf{H}}{\partial \mathbf{w}} + \frac{\partial \mathbf{H}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \mathbf{w}} = 0\,. \tag{4.7}$$

In a similar way as the Gradient and Jacobian $\frac{\partial \mathbf{H}}{\partial \mathbf{w}}$, $\frac{\partial \mathbf{H}}{\partial \mathbf{x}}$ are computed by CASADI. Therefore, the equation (4.7) is solved by using a algebra lineal method to obtain $\frac{\partial \mathbf{x}}{\partial \mathbf{w}}$.

Algorithm 6: Computation of gradient $\nabla \mathbf{F}$ and Jacobian $\nabla \mathbf{G}$

Input : Starting point $\mathbf{w_0}$, initial states $x(0)$, sensitivities $\frac{\partial \mathbf{F}}{\partial \mathbf{w}}$, $\frac{\partial \mathbf{F}}{\partial \mathbf{x}}$, $\frac{\partial \mathbf{G}}{\partial \mathbf{w}}$, $\frac{\partial \mathbf{G}}{\partial \mathbf{x}}$ and random variables ξ

Output : $\mathbf{F}$, $\nabla \mathbf{F}$, $\mathbf{G}$ and $\nabla \mathbf{G}$

1 Compute $\mathbf{x}$ by solving (4.6)
2 Set $\mathbf{F}$ (4.3)
3 Set $\mathbf{G}$ (4.4)
4 Compute $\frac{\partial \mathbf{x}}{\partial \mathbf{w}}$ by solving (4.7)
5 Set $\nabla \mathbf{F}$ (4.5a)
6 Set $\nabla \mathbf{G}$ (4.5b)

Nevertheless, the code generation of the sensitivities can be computationally expensive due to the state variables $\mathbf{x}$ depend mainly on the number of random variables and the prediction horizon. Therefore, to reduce the problem of the code generation, the following method is used to obtain the sensitivities required. The objective function and the inequality constraint are expressed as follow.

$$\mathbf{F}(\mathbf{x}(\mathbf{w}),\mathbf{w}) = \frac{1}{N_s}[\overbrace{f(\mathbf{x_1}(\mathbf{w}),\mathbf{w})}^{\mathbf{F_1}} + \overbrace{f(\mathbf{x_2}(\mathbf{w}),\mathbf{w})}^{\mathbf{F_2}} + \ldots + \overbrace{f(\mathbf{x_{N_s}}(\mathbf{w}),\mathbf{w})}^{\mathbf{F_{N_s}}}] \tag{4.8}$$

$$\mathbf{G_1} = \begin{bmatrix} \Psi(\pm\tau, g(x_{(1,1)},\mathbf{w})) \\ \Psi(\pm\tau, g(x_{(2,1)},\mathbf{w})) \\ \vdots \\ \Psi(\pm\tau, g(x_{(N,1)},\mathbf{w})) \end{bmatrix}$$

$$\mathbf{G_2} = \begin{bmatrix} \Psi(\pm\tau, g(x_{(1,2)},\mathbf{w})) \\ \Psi(\pm\tau, g(x_{(2,2)},\mathbf{w})) \\ \vdots \\ \Psi(\pm\tau, g(x_{(N,2)},\mathbf{w})) \end{bmatrix}$$

$$\vdots$$

$$\mathbf{G_{N_s}} = \begin{bmatrix} \Psi(\pm\tau, g(x_{(1,N_s)},\mathbf{w})) \\ \Psi(\pm\tau, g(x_{(2,N_s)},\mathbf{w})) \\ \vdots \\ \Psi(\pm\tau, g(x_{(N,N_s)},\mathbf{w})) \end{bmatrix}$$

$$\mathbf{G}(\mathbf{x}(\mathbf{w}),\mathbf{w}) = \frac{1}{N_s}(\mathbf{G_1} + \mathbf{G_2} + \ldots + \mathbf{G_{N_s}}) \tag{4.9}$$

Moreover, the gradient ∇F of the objective function and the Jacobian ∇G can be computed as follow.

$$\nabla\mathbf{F} = \frac{1}{N_s}(\nabla\mathbf{F_1} + \nabla\mathbf{F_2} + \ldots + \nabla\mathbf{F_{N_s}}) \tag{4.10}$$

$$\nabla\mathbf{G} = \frac{1}{N_s}(\nabla\mathbf{G_1} + \nabla\mathbf{G_2} + \ldots + \nabla\mathbf{G_{N_s}}) \tag{4.11}$$

The elements F_i and F_{i+1} of the objective function have the same symbolic representation, so that ∇F_i and ∇F_{i+1} keep the same structure. In a similar way, the elements of the set constraints G_{i+1} and G_i accomplish the same properties. Therefore, the sparse

structure ∇G_i is equal to ∇G_{i+1}. The function generated by CASADI is summarized in Algorithm 8.

Algorithm 7: Computation of gradient $\nabla \mathbf{F_k}$ and Jacobian $\nabla \mathbf{G_k}$

Input : Initial states $x(0)$, random variables ξ_i and control variables $\mathbf{w}$
Output : $\mathbf{F_k}$, $\nabla \mathbf{F_k}$, $\mathbf{G_k}$ and $\nabla \mathbf{G_k}$

Compute $\mathbf{x}$ by solving (4.6)
Set $\mathbf{F_k}$
Set $\mathbf{G_k}$
Compute $\nabla \mathbf{F_k}$ by using functions CASADI
Compute $\nabla \mathbf{G_k}$ by using functions CASADI

Algorithm 8: Computation of gradient $\nabla \mathbf{F}$ and Jacobian $\nabla \mathbf{G}$

Input : Starting point $\mathbf{w_0}$ and initial states $x(0)$
Output : $\mathbf{F}$, $\nabla \mathbf{F}$, $\mathbf{G}$ and $\nabla \mathbf{G}$

Set $k = 1$
Set $\mathbf{F} = 0$
Set $\nabla \mathbf{F} = 0$
Set $\mathbf{G} = 0$
Set $\nabla \mathbf{G} = 0$
repeat
    Compute $\mathbf{F_k}$, $\nabla \mathbf{F_k}$, $\mathbf{G_k}$ and $\nabla \mathbf{G_k}$ using the algorithm 7.
    Set $\mathbf{F} = \mathbf{F} + \mathbf{F_k}$.
    Set $\nabla \mathbf{F} = \nabla \mathbf{F} + \nabla \mathbf{F_k}$.
    Set $\mathbf{G} = \mathbf{G} + \mathbf{G_k}$.
    Set $\nabla \mathbf{G} = \nabla \mathbf{G} + \nabla \mathbf{G_k}$.
    Set $k = k + 1$
until $k > N_s$;
Set $\mathbf{F} = \frac{\mathbf{F}}{N_s}$
Set $\mathbf{G} = \frac{\mathbf{G}}{N_s}$
Set $\nabla \mathbf{F} = \frac{\nabla \mathbf{F}}{N_s}$
Set $\nabla \mathbf{G} = \frac{\nabla \mathbf{G}}{N_s}$

4.1.2 Computation of the second derivatives

To compute the Hessian matrix at each iteration, it is possible to use the option IpOpt's approximation by BFGS. However, if the Hessian can be computed, the algorithm is usually more robust and converges faster. In this work from the 3 cases studied are proposed, where the L-BFGS Hessian approximation is used for the case studies 2 and 3, while the Hessian matrix is computed for the case studio 1 due to the fact that the computational time is considerably smaller than using the approximation.

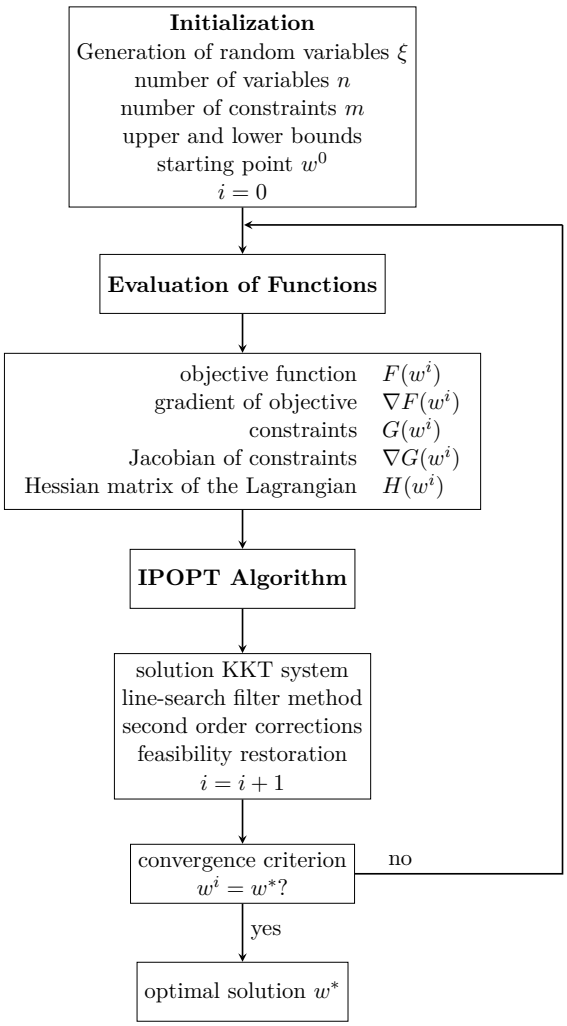

Figure 4.2 – Diagram of the solution procedure of IpOpt [74].

The Hessian of the Lagrangian function depends on the function of second order $\nabla^2 F$(Objective Function) and $\nabla^2 G(k)$ (N constraints).

$$\nabla^2 L = \overbrace{\sigma_f \ \nabla^2 F}^{A} + \overbrace{\sum_{i=1}^{N} \lambda_i \nabla^2 G(i)}^{B}. \tag{4.12}$$

Where σ_f is the factor that IpOpt use to identify the Hessian of the objective A or the constraints B independently, λ_i are the multipliers for the constraints.

In a similar way as the functions computed of the first order, the Hessian matrix are formulated as follow.

$$\nabla^2\mathbf{L} = \frac{1}{N_s}(\nabla^2\mathbf{L_1} + \nabla^2\mathbf{L_2} + \ldots + \nabla^2\mathbf{L_{N_s}}) \tag{4.13}$$

Algorithm 9: Computation of the Hessian matrix $\nabla^2\mathbf{L}$

Input : Starting point $\mathbf{w_0}$, initial states $x(0)$ and random variables ξ_i
Output : $\nabla^2\mathbf{L}$

Set $k = 1$
Set $\nabla^2\mathbf{L} = 0$
repeat
  Compute $\nabla^2\mathbf{L_k}$ by using functions CASADI
  Set $\nabla^2\mathbf{L} = \nabla^2\mathbf{L} + \nabla^2\mathbf{L_k}$.
  Set $k = k + 1$
until $k > N_s$;
Set $\nabla^2\mathbf{L} = \frac{\nabla^2\mathbf{L}}{N_s}$

The sparse structure of the functions of the first and second order and the number of non zeros are obtained by the CASADI functions. Finally, the option of the derivative checker in IpOpt is activated for the verification of the computation for the first and second derivative. When this option is activated the IpOpt solver calculate internally only for the starting point w_0 to compare with the one provided by the user through a error tolerance. In this work , the tolerance of the check derivative is set to 10^{-6}. The diagram of the solution procedure is shown in Figure 4.2.

4.1.3 Generation of random variables

To implement the quasi-random variables, the library $gsl - qrng$ is used that is based on the algorithms described in [16]. The random variables ξ are generated using QSM with Sobol sequence (with low discrepancy proporties). This library allows the generation of 10 000 samples until a maximum of 40 sequences [36].

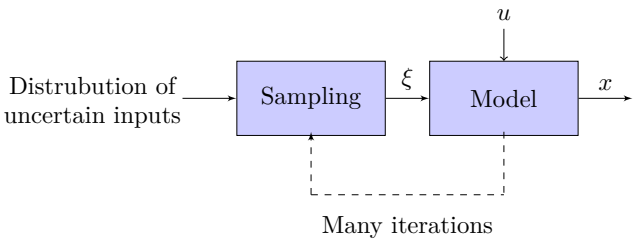

Figure 4.3 – Framework of Monte-Carlo simulation

Chapter 5

Case-Studies and Computational Results

In this section, three case studies are presented related with the obstacle avoidance problem and trajectory tracking for autonomous vehicle considering uncertainties. The kinematic and bicycle model are employed for the our case studies.

5.1 Case 1: Car-like Vehicle Model

The car-like model is based only on kinematic relationships. The vehicle sketch is represented in the Figure 5.1.The states of the model are $\mathbf{X} = [x \quad y \quad \theta \quad v]^T$, where x and y are the center point coordinate of the rear axle, θ is the heading angle of the car body with respect to the x-axis and v is the line velocity. The distance between the front and the rear axles is represented by l. The control variable $\mathbf{U} = [\delta \quad a]^T$, where ψ and v are the steering angle and linear acceleration , respectively. The stochastic variables are $\xi = [\xi_1 \quad \xi_2]$ where both represent the error of the motion model.The numerical value of the parameters are obtained from [50] and are shown in Table 5.1. The following differential equations describes the kinematic relationship.

Table 5.1 – Parameters of the random variables in the Car-like vehicle model obtained from [50]

Parameter	Value	Description
$\sum_{\xi_1,\xi_2}$	diag(0.5, 0.02)	Covariance between ξ_1 and ξ_2

$$\begin{aligned} \dot{x} &= v\cos(\theta)\,, \\ \dot{y} &= v\sin(\theta)\,, \\ \dot{\theta} &= v\left(\frac{\tan(\delta)}{l}\right) + \xi_1\,, \\ \dot{v} &= a + \xi_2\,, \end{aligned} \tag{5.1}$$

The aim of the optimization problem is to arrive at a desired final position by following (x_d, y_d) while avoiding the obstacle that exists in the trajectory. To test the avoidance strategy, Figure 5.2 is considered for the first case. The desired position is set to $(x_d, y_d) = (108m, 0m)$ and the obstacle is located at $(x_{obs}, y_{obs}) = (60m, 0m)$.

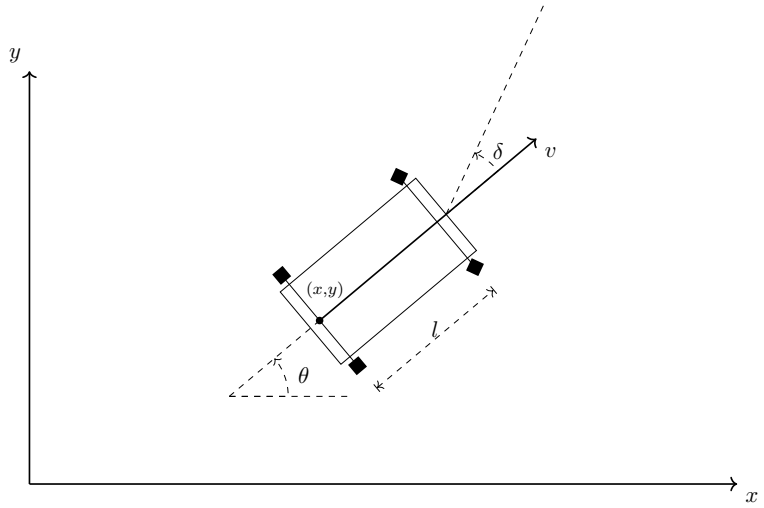

Figure 5.1 – Kinematic vehicle model [66]

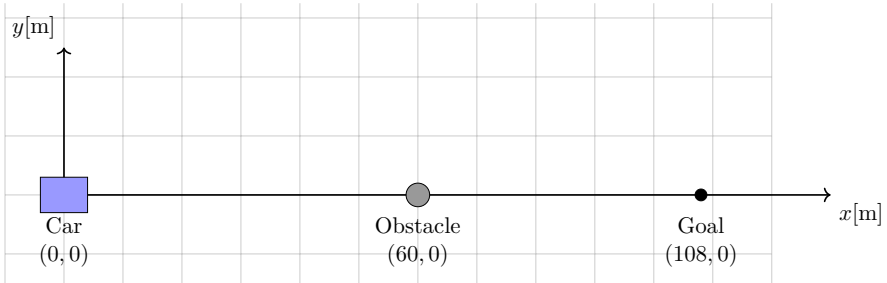

Figure 5.2 – Mobile working area and obstacle to be avoided for the Car-like vehicle model

Thus, the objective function is defined as follow:

$$J_{track}(x,u,\xi) = E[\sum_{k=1}^{N-1} q_x(x_{k,\xi} - x_{ref,k})^2 + q_y(y_{k,\xi} - x_{ref,k})^2] + r_a a_k^2 + r_\delta \delta_k^2 \,. \quad (5.2)$$

Where $E[\ \ ]$ is the expected value operator. The state vector of initial conditions is given by

$$x(0) = [0,\ 0,\ 0,\ 0]^T \,. \quad (5.3)$$

The final time is fixed to $t_f = 12$ s and the sampling time is set to $\Delta T = 0.1$s. Apply Runge Kutta Methods to equation 5.1 to obtain the discrete-time model. The control variables are constrained as follows,

$$\begin{aligned} -8^\circ \leq \quad \delta \quad &\leq 8^\circ \,, \\ -0.5^\circ \leq \quad \Delta\delta \quad &\leq 0.5^\circ \,, \\ -10m/s^2 \leq \quad a \quad &\leq 10m/s^2 \,, \end{aligned} \quad (5.4)$$

The chance constraint is formulated for obstacle avoidance using the path constraint approach.

$$Pr[1 - \frac{(x_{k,\xi} - x_{obs})^2}{r_a^2} - \frac{(y_{k,\xi} - y_{obs})^2}{r_b^2} \leq 0] \geq \alpha, \quad k = 1,2,\ldots,N \quad (5.5)$$

the resulting Stochastic NMPC problem is given by:

$$\begin{aligned} \min_u \quad & J_{track}(x,u,\xi) \\ \text{subject to:} \quad & x_0 = x(0) \,, \\ & \text{Model equation}(5.1) \,, \\ & \text{Path constraint } (5.4) \,, \\ & \text{Chance constraint } (5.5) \,. \end{aligned}$$

5.1.1 Deterministic MPC for Car-like vehicle model

In deterministic optimization, the stochastic variables are set constant with their expected values. The prediction horizon considered is of $N = 8$, the gain factors in the objective function are set $q_x = 1$, $q_y = 1$, $r_a = 0.01$, $r_\delta = 0.1$ and the parameter of the ellipse are set $r_a = 4m$ and $r_b = 0.4m$. The results of the obstacle avoidance problem are shown

in Figure 5.3, the optimal controls in Figure 5.4 and the optimal states 5.5. As can be seen, the path constraint approach accomplish the task of the obstacles avoidance. In deterministic optimization, the trajectory of the vehicle moves slightly close to the border of the ellipse.

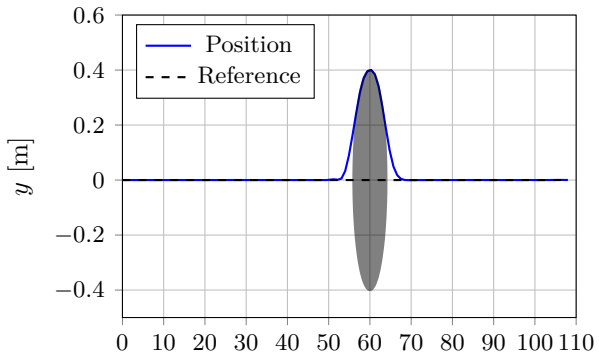

Figure 5.3 – Simulation results for the case 1: Trajectory in the X-Y plane for obstacle avoidance using deterministic MPC.

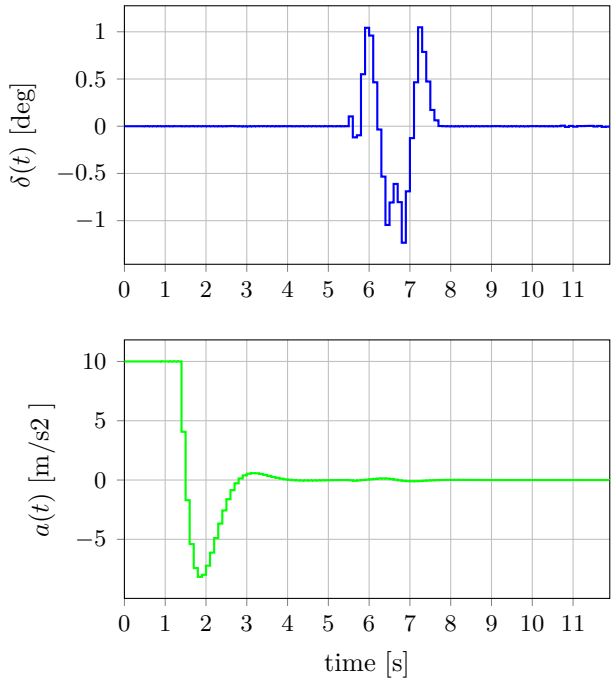

Figure 5.4 – Simulation results for the case 1: Optimal control inputs for obstacle avoidance using deterministic MPC.

The convergence of the objective function is shown in Figure 5.6 and CPU time is reported in Table 5.2 in deterministic optimization.

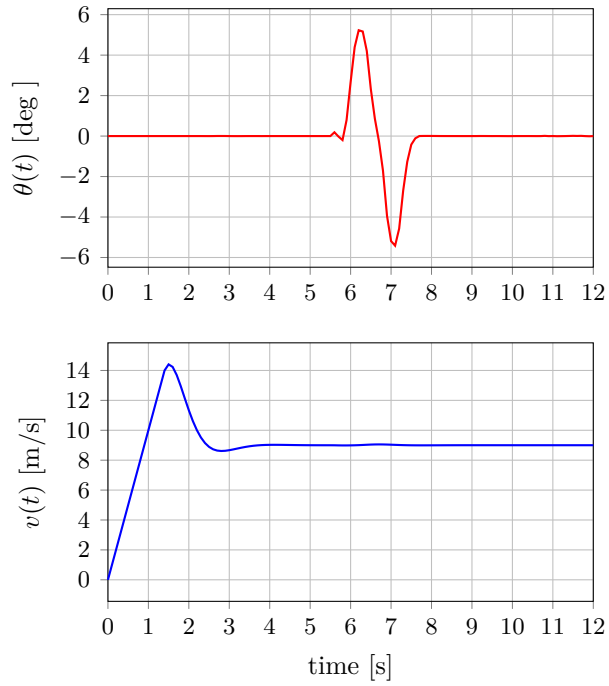

Figure 5.5 – Simulation results for the case 1: Optimal states for obstacle avoidance using deterministic MPC.

However, the results from the deterministic optimization cannot be applied due to fact that the constraint of the obstacle avoidance are not satisfied when the stochastic variables are taken into account as can be seen in Figure 5.7. Considering 10 000 samples generated by the Monte Carlo method, the deterministic optimization lead to about 41.8% of violations of the constraint.

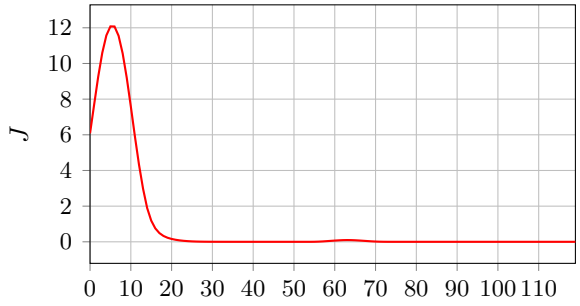

Figure 5.6 – Simulation results for the case 1: Convergence of the objective function using deterministic MPC

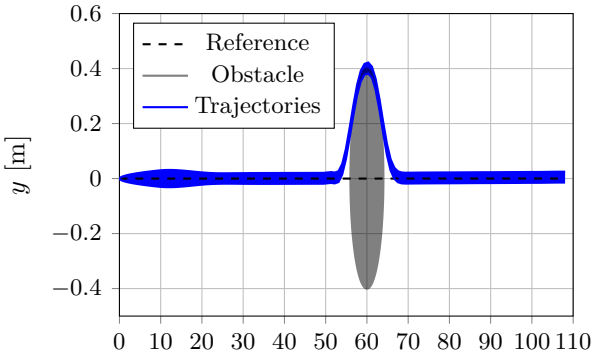

Figure 5.7 – Simulation result for the case 1: Trajectory in the X-Y plane for obstacle avoidance under uncertainties using deterministic MPC.

Table 5.2 – Average computation time per iteration (ms) required for the solution of case 1 using deterministic MPC.

	Case 1
CPU time in IpOpt	1.233
CPU time in NLP function evaluations	9.066

5.1.2 Stochastic MPC for car-like vehicle model

In this section, IA and OA approach was tested with a prediction horizon $N = 8$. For the case 1, the values of $m1 = 1$ and $m2 = 0.333$ of the parametric function were considered. In the case of IA, the results were obtained for $1 \leq \tau \leq 0.1$ and OA for $1 \leq \tau \leq 0.001$. Moreover, different levels of probability were considered for $\alpha = 0.80,\ 0.9,\ 0.95$.

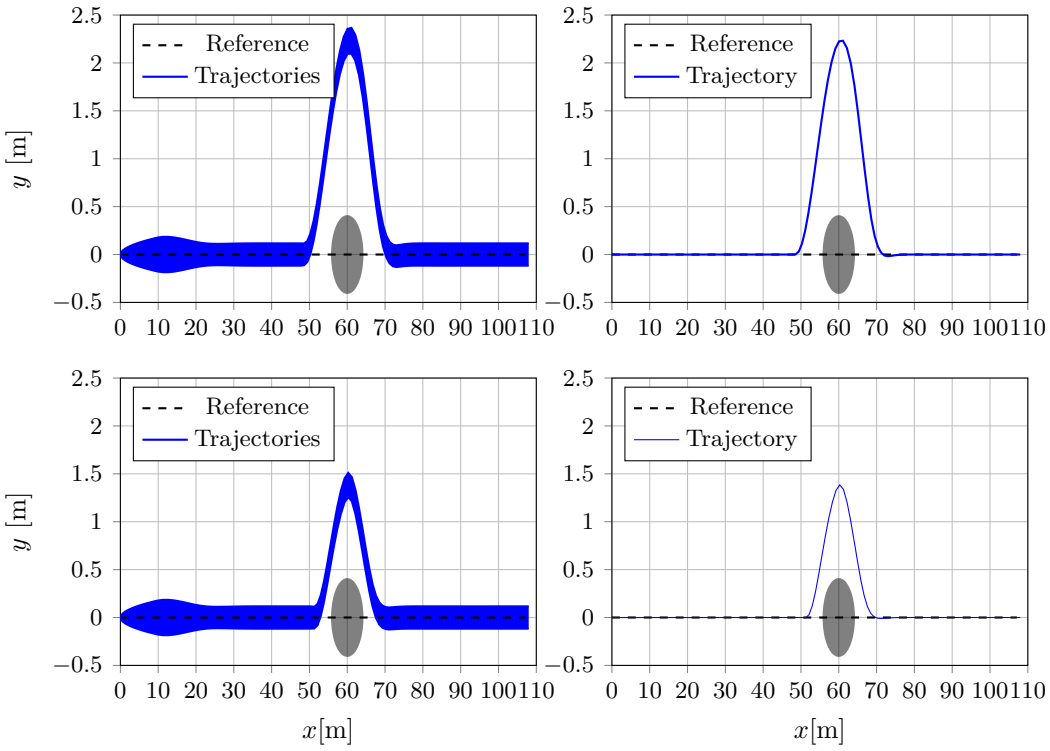

Figure 5.8 – Simulation results for the case 1: Trajectory in the X-Y plane for obstacle avoidance (up: IA $\tau = 0.1$)and (down:OA $\tau = 0.001$) with $\alpha = 0.95$.(left:N_s Trajectories of the simulation QSM) and (right: Expected Value of the trajectories)

The results of the stochastic optimization are shown in Figure 5.8, 5.9 and 5.10 that correspond the feasible trajectory for the obstacle avoidance for different values of probability. As can be seen, the feasibility of the OA is bigger that the IA. As a result, the trajectory of the vehicle in OA is closer to obstacle avoidance than IA. Moreover, if the probability level is increased, the feasible trajectory of the vehicle decreases.
On the other, to shown the effects of the uncertainties Figure 5.12 and 5.13 depict the state variables and Figure 5.11 show the deterministic control variables. All the solutions only correspond considering a level probability $\alpha = 0.95$ for IA and OA with the value of $\tau = 0.1$ and $\tau = 0.001$, respectively. As can be seen in Figure 5.12 and 5.13,

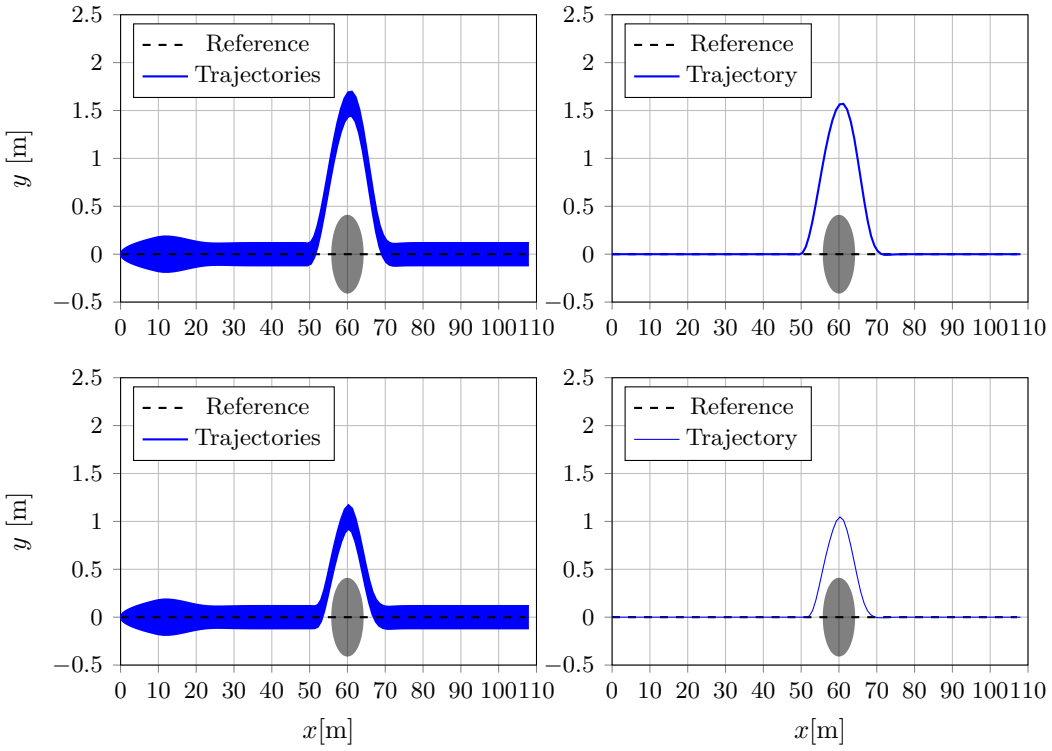

Figure 5.9 – Simulation results for the case 1: Trajectory in the X-Y plane for obstacle avoidance (up: IA $\tau = 0.1$)and (down:OA $\tau = 0.001$) with $\alpha = 0.90$.(left:N_s Trajectories of the simulation QSM) and (right: Expected Value of the trajectories)

The influence of the stochastic variables at the heading angle θ are greater than the velocity v.

Table 5.3 – Average computation time in IpOpt per iteration (seconds) required for the solution of case 1.

α	IA	CPU time	OA	CPU time
0.8	$\tau = 1$	0.121	$\tau = 0.1$	0.119
	$\tau = 0.5$	0.118	$\tau = 0.01$	0.123
	$\tau = 0.1$	0.117	$\tau = 0.001$	0.127
0.9	$\tau = 1$	0.132	$\tau = 0.1$	0.117
	$\tau = 0.5$	0.128	$\tau = 0.01$	0.123
	$\tau = 0.1$	0.120	$\tau = 0.001$	0.128
0.95	$\tau = 1$	0.135	$\tau = 0.1$	0.117
	$\tau = 0.5$	0.129	$\tau = 0.01$	0.124
	$\tau = 0.1$	0.125	$\tau = 0.001$	0.128

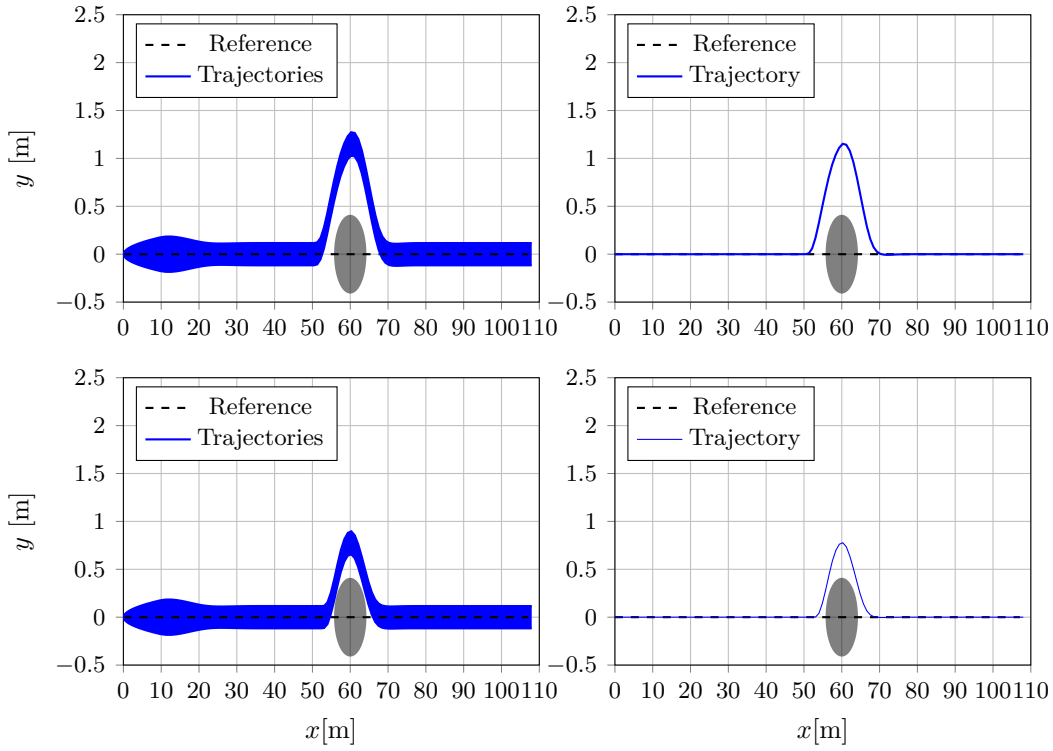

Figure 5.10 – Simulation results for the case 1: Trajectory in the X-Y plane for obstacle avoidance (up: IA $\tau = 0.1$)and (down:OA $\tau = 0.001$) with $\alpha = 0.80$.(left:N_s Trajectories of the simulation QSM) and (right: Expected Value of the trajectories)

Table 5.4 – Average computation time in NLP function evaluations per iteration (seconds) required for the solution of case 1.

α	IA	CPU time	OA	CPU time
0.8	$\tau = 1$	4.264	$\tau = 0.1$	3.242
	$\tau = 0.5$	3.856	$\tau = 0.01$	4.318
	$\tau = 0.1$	4.264	$\tau = 0.001$	5.059
0.9	$\tau = 1$	4.623	$\tau = 0.1$	3.153
	$\tau = 0.5$	4.352	$\tau = 0.01$	4.819
	$\tau = 0.1$	3.870	$\tau = 0.001$	6.017
0.95	$\tau = 1$	5.206	$\tau = 0.1$	3.247
	$\tau = 0.5$	4.670	$\tau = 0.01$	5.061
	$\tau = 0.1$	3.999	$\tau = 0.001$	6.245

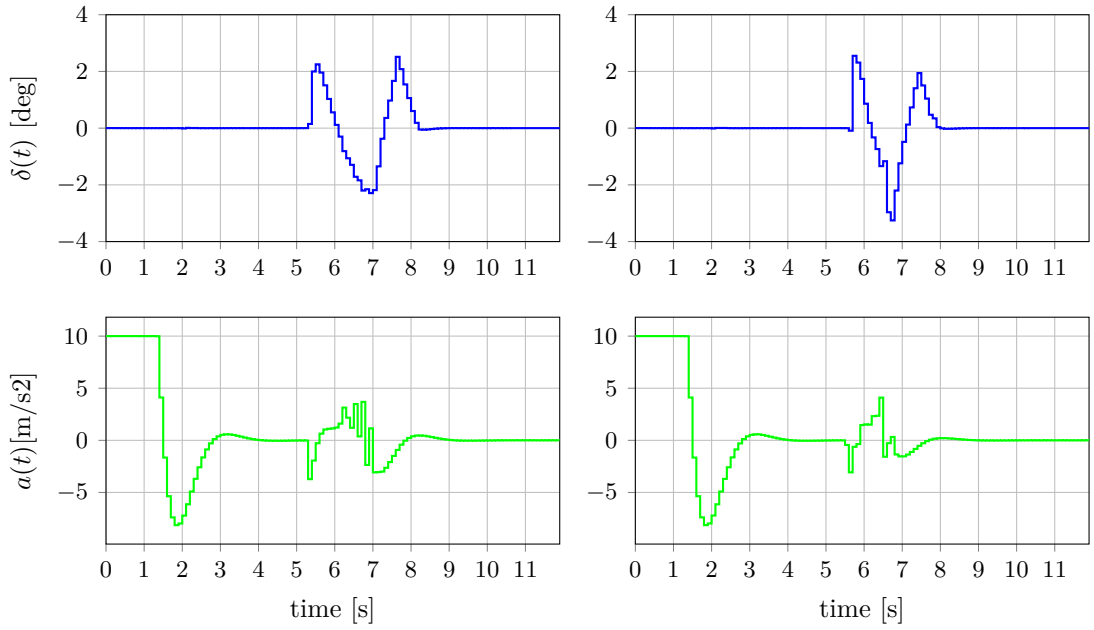

Figure 5.11 – Simulation results for the case 1: Optimal control inputs for the obstacle avoidance (left: Inner Approximation $\tau = 0.1$)and (right: Outer Approximation $\tau = 0.001$) with $\alpha = 0.95$

Moreover, as can be seen (Figure 5.14) , as τ decreases, the optimal value of the objective function tends to be the same for both approaches. Table 5.3 and 5.4 ilustrate the CPU times in both approach where the average time reported in IA and OA are significant respect to the CPU time obtained in deterministic optimization Table 5.2.

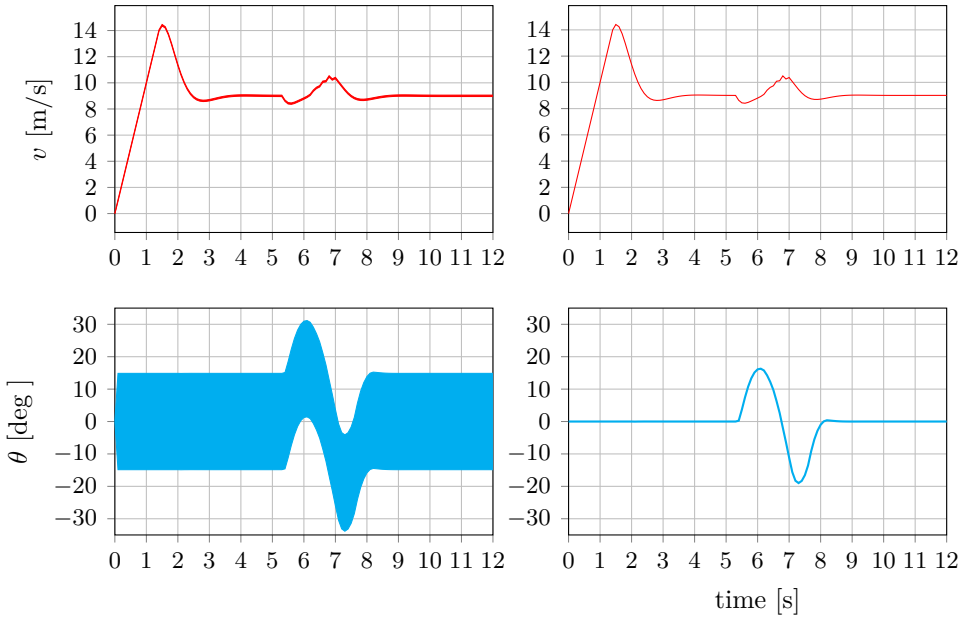

Figure 5.12 – Simulation results for the case 1: Optimal states for the obstacle avoidance using IA ($\tau = 0.1$) with $\alpha = 0.95$. (left: N_s trajectories of the simulation QSM) and (right: expected value of the trajectories)

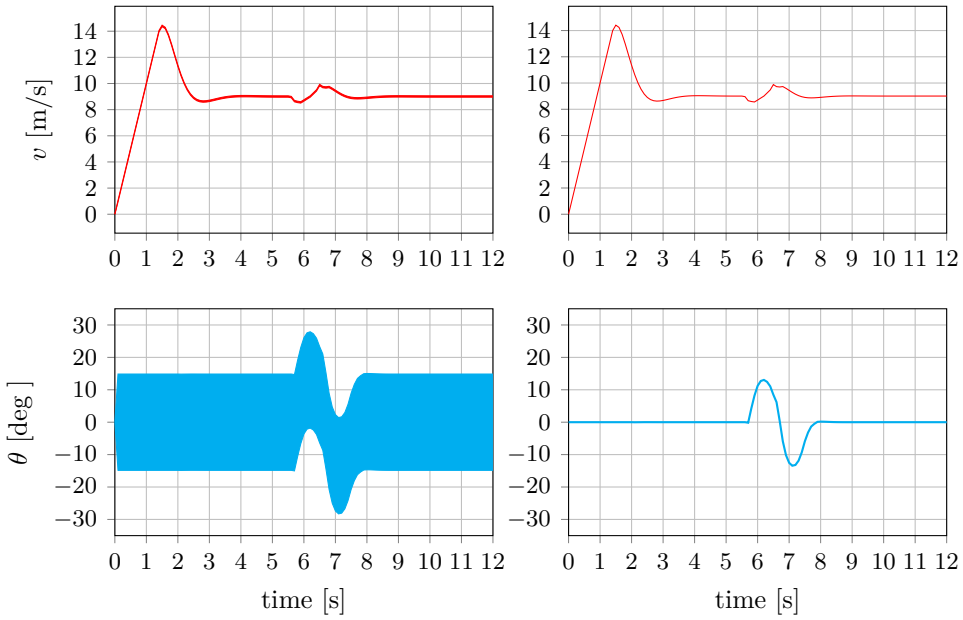

Figure 5.13 – Simulation results for the case 1: Optimal states for the obstacle avoidance using OA ($\tau = 0.001$) with $\alpha = 0.95$. (left: N_s trajectories of the simulation QSM) and (right: expected value of the trajectories)

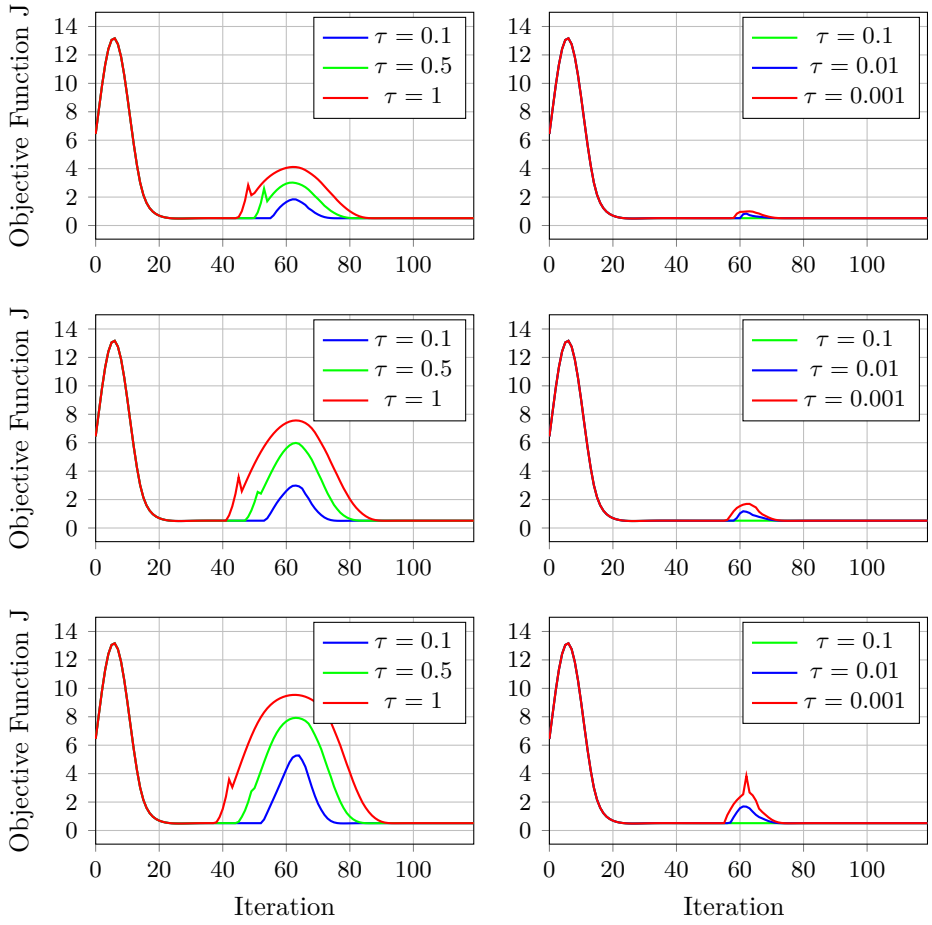

Figure 5.14 – Simulation results for the case 1: Optimal value of the objective function for different values of α (left: Inner Approximation) and (right: Outer Approximation). From top to bottom: $\alpha = 0.8$, $\alpha = 0.9$, $\alpha = 0.95$

5.2 Case 2: Vehicle single track model

The single-track model, also called bicycle model, is a more general representation of a four-wheel vehicle. This model lumps the two front wheels into one wheel and makes the same with the rear wheels. Apart from the Cartesian position ($X \quad Y$) and the yaw angle ψ, the description of the vehicles motion considers the yaw rate $\dot{\psi}$. The state vector is represented as $\mathbf{X} = [\dot{y} \quad \psi \quad \dot{\psi} \quad X \quad Y]$, while the control variable is represented by $\mathbf{U} = [\delta \quad v_x]$, where δ is the front steering angle v_x is the line velocity. In addition, the stochastic variables are introduced in the model equations and represented by $\xi = [\xi_1 \quad \xi_2 \quad \xi_3 \quad \xi_4 \quad \xi_5]$ where ξ_1 and ξ_2 are additive error caused by unknown forces on the vehicle produced by crosswind, lane grooves or a sloping , meanwhile ξ_3, ξ_4 and ξ_5 represent the error estimation and measurement. The numerical value of the parameters are obtained from [41] and are shown in Table 5.6.The bicycle model is represented as follow.

$$\begin{aligned}
\ddot{y} &= -(\frac{C_f + C_r}{m\ v_x})\ \dot{y} + (\frac{C_r l_r - C_f l_f}{m\ v_x} - v_x)\ \dot{\psi} + \frac{C_f}{m}\ \delta + \xi_1\,, \\
\ddot{\psi} &= (\frac{C_r l_r - C_f l_f}{I_z\ v_x})\ \dot{y} - (\frac{C_r\ l_r^2 + C_f\ l_f^2}{I_z\ v_x})\dot{\psi} + \frac{C_f\ l_f}{I_z}\ \delta + \xi_2\,, \\
\dot{\psi} &= \dot{\Psi} + \xi_3\,, \\
\dot{X} &= v_x \cos(\Psi) - \dot{y}\sin(\Psi) + \xi_4\,, \\
\dot{Y} &= v_x \sin(\Psi) + \dot{y}\cos(\Psi) + \xi_5\,,
\end{aligned} \tag{5.7}$$

The trajectory tracking problem is a well-known application for autonomous vehicles. Due to the problem consist of obtaining the optimal control inputs that make the AV follow a given desired trajectory. As a result, for second case the desired path is

Table 5.5 – Physical parameters of the single-track model [41]

Parameter	Value	Description
m	1550 kg	Mass of the vehicle
I_z	2800 kg.m^2	Inertia moment
l_f	1.344 m	Distance from COG to front axle
l_r	1.456 m	Distance from COG to rear axle
C_f	75000 N/rad	Front cornering stiffness
C_r	150000 N/rad	Rear cornering stiffness

described in terms of the lateral position Y_{ref} and the yaw angle ψ_{ref} as function of the longitudinal position X_{ref}. The equations 5.8 describe a double lane change that have been employed in different tests for different scenarios i.e.[24, 25, 48] as follow :

$$\begin{aligned} y_{ref}(x_{ref}) = & \quad \frac{d_{y1}}{2}(1+tanh(z_1)) - \frac{d_{y2}}{2}(1+tanh(z_2))\,, \\ \psi_{ref}(x_{ref}) = & \quad atan\left(\frac{1.2}{d_{x1}}(d_{y1})(\frac{1}{cosh(z1)})^2 - \frac{1.2}{d_{x2}}(d_{y2})(\frac{1}{cosh(z2)})^2\right)\,. \end{aligned} \tag{5.8}$$

Where the numeric parameter of double lane change are the following:

Table 5.6 – Parameters of the random variables in the Single-track Model based on [41]

Parameter	Value	Description
σ_{ξ_1}	0.5	Standard deviation of perturbation
σ_{ξ_2}	0.17	Standard deviation of perturbation
σ_{ξ_3}	0.1	Standard deviation of noise
σ_{ξ_4}	0.05	Standard deviation of noise
σ_{ξ_5}	0.05	Standard deviation of noise
r_{ξ_3,ξ_4,ξ_5}	0.6	Correlation between ξ_3, ξ_4 and ξ_5

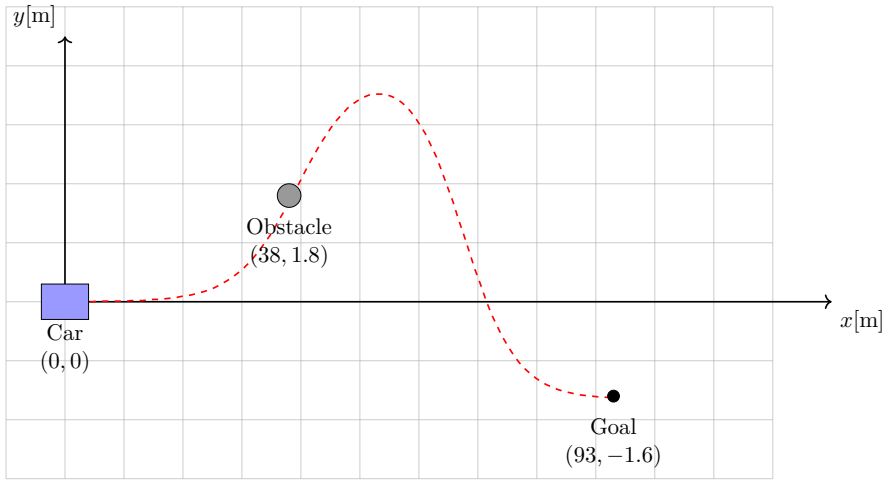

Figure 5.15 – Mobile working area and obstacle to be avoided for the Single-track model. The dashed red line is the double lane change reference

$$
\begin{aligned}
z_1 &= \frac{24}{2.5}(X_{ref} - 27.19) - 1.2\,, \\
z_2 &= \frac{24}{21.95}(X_{ref} - 56.46) - 1.2\,, \\
d_{x1} &= 25\,, \\
d_{x2} &= 21.95\,, \\
d_{y1} &= 4.05\,, \\
d_{y2} &= 5.7\,,
\end{aligned} \tag{5.9}
$$

The aim of the optimization problem is to arrive at a desired final position while avoiding the obstacle that exists in the double lane change trajectory. The scenario for the second case is shown in Figure 5.15 where the desired position is $(x_d, y_d) = (93m, -1.6m)$ and the obstacle is located at $(x_{obs}, y_{obs}) = (38m, 1.8m)$.
Thus, the objective function is defined as follow:

$$
\begin{aligned}
J_{track}(x, u, \xi) =& E[\sum_{k=1}^{N-1} q_X(X_{k,\xi} - x_{ref,k})^2 + q_Y(Y_{k,\xi} - y_{ref,k})^2 + q_\psi(\psi_{k,\xi} - \psi_{ref,k})^2] \\
&+ r_{v_x} v_{x,k}^2 + r_\delta \delta_k^2
\end{aligned} \tag{5.10}
$$

The state vector of initial conditions is given by.

$$
x(0) = [0,\ 0,\ 0,\ 0,\ 0]^T\,. \tag{5.11}
$$

the final time is fixed to $t_f = 10$ s and the sampling time is set to $\Delta T = 0.1$s. Apply Runge Kutta Methods to equation 5.7 to obtain the discrete-time model. The control variables are constrained as follows:

$$
\begin{aligned}
-8^\circ \leq &\quad \delta &&\leq 8^\circ\,, \\
-0.5^\circ \leq &\quad \Delta\delta &&\leq 0.5^\circ\,, \\
3.2m/s \leq &\quad v_x &&\leq 6m/s\,,
\end{aligned} \tag{5.12}
$$

The chance constraint is formulated for obstacle avoidance.

$$Pr[1 - \frac{(x_{k,\xi} - x_{obs})^2}{r_a^2} - \frac{(y_{k,\xi} - y_{obs})^2}{r_b^2} \leq 0] \geq \alpha, \quad k = 1, 2, \ldots, N \tag{5.13}$$

the resulting Stochastic NMPC problem is given by:

$$\begin{aligned} \min_{u} \quad & J_{track}(x, u, \xi) \\ \text{subject to:} \quad & x_0 = x(0)\,, \\ & \text{Model equation (5.7)}\,, \\ & \text{Path constraint (5.12)}\,, \\ & \text{Chance constraint (5.13)}\,. \end{aligned}$$

5.2.1 Deterministic MPC for vehicle single track model

The prediction horizon considered is of $N = 8$, the gain factors in the objective function are set $q_x = 1$, $q_y = 1$, $q_\psi = 1$, $r_{v_x} = 0.1$, $r_\delta = 0.1$ and the parameter of the ellipse are set to $r_a = 4m$ and $r_b = 0.2m$. The results of double lane change with the obstacle avoidance are shown in Figure 5.16, the optimal controls in Figure 5.17 and the optimal states 5.18.
The CPU-time is shown in Figure 5.7 in deterministic optimization.
Similar to previous case, the results from the deterministic optimization cannot be applied since the constraint of the obstacle avoidance are not satisfied when the stochastic variables are taken into account (Figure 5.20). Considering 10 000 samples generated by the Monte Carlo method, the deterministic optimization lead to about 41% of violations of the constraint.

Table 5.7 – Average computation time per iteration (ms) required for the solution of case 2 using deterministic MPC.

	Case 2
CPU time in IpOpt	2.28
CPU time in NLP function evaluations	5.72

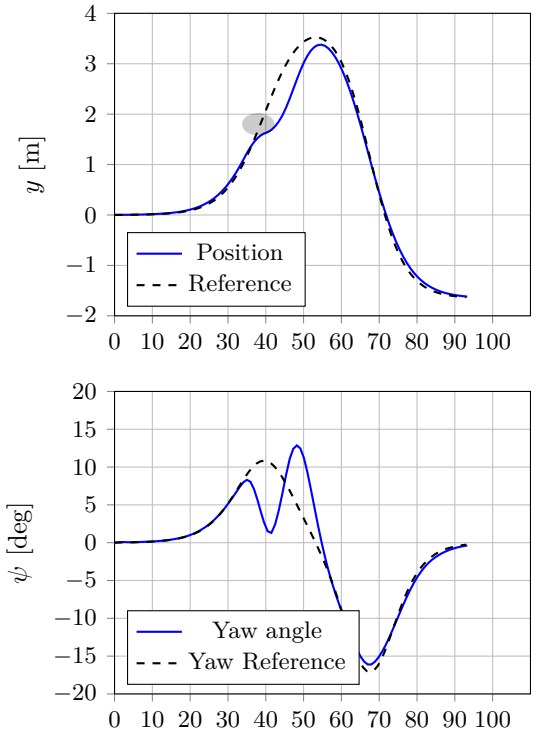

Figure 5.16 – Simulation result for the case 2: Trajectory in the X-Y plane and yaw angle for obstacle avoidance using deterministic MPC.

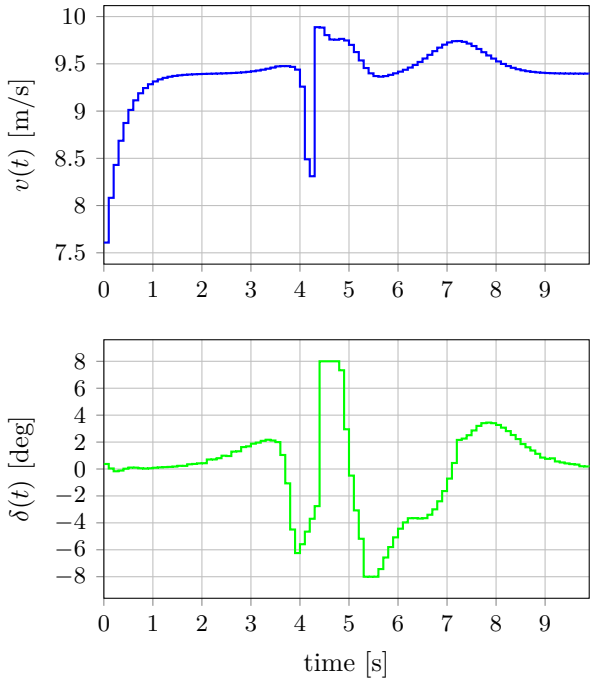

Figure 5.17 – Simulation results for the case 2: Optimal control inputs for the tracking problem considering the kinematic model using deterministic MPC.

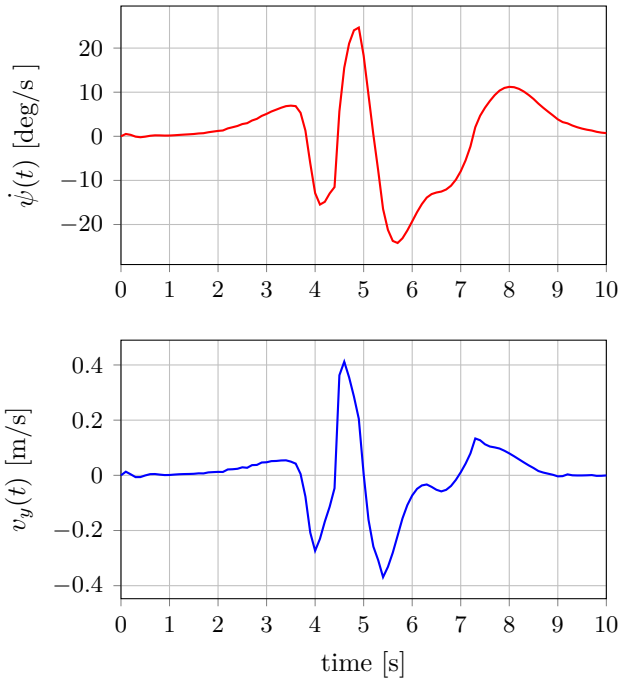

Figure 5.18 – Simulation results for the case 2: Optimal states for the tracking problem considering the kinematic model using Deterministic MPC

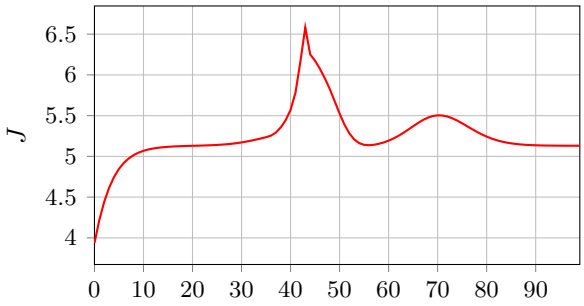

Figure 5.19 – Simulation results for the case 2: Convergence of the objective function and computation time using Deterministic MPC

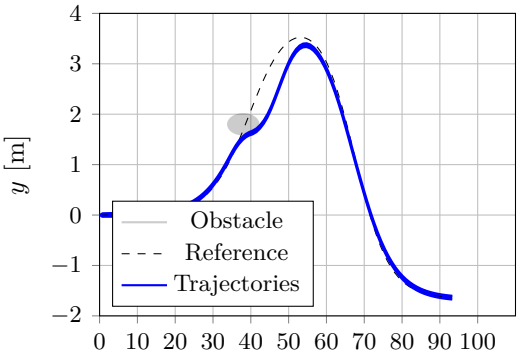

Figure 5.20 – Simulation result for the Case 2: Trajectory in the X-Y plane for obstacle avoidance under uncertainties using Deterministic MPC.

5.2.2 Stochastic MPC for vehicle single track model

IA and OA approach was tested with a prediction horizon $N = 8$. For the case 2, the values of $m1 = 1$ and $m2 = 0.333$ of the parametric function were considered. In the case of IA, the results were obtained for $1 \leq \tau \leq 0.1$ and OA for $0.1 \leq \tau \leq 0.0002$. Moreover, different levels of probability were considered for $\alpha = 0.80,\ 0.9,\ 0.95$.

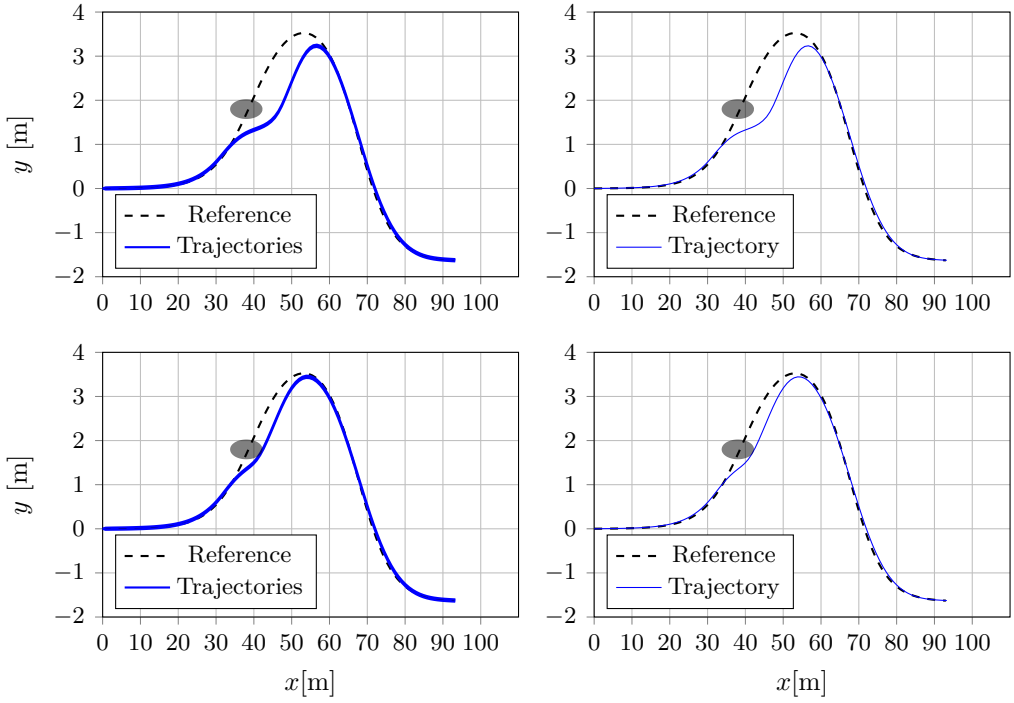

Figure 5.21 – Simulation results for the case 2: Trajectory in the X-Y plane for obstacle avoidance (up: Inner Approximation $\tau = 0.1$)and (down: Outer Approximation $\tau = 0.0002$) with $\alpha = 0.95$.(left:N_s Trajectories of the simulation QSM) and (right: Expected Value of the trajectories)

The results of the stochastic optimization are shown in Figure 5.21, 5.22 and 5.23 that correspond the feasible trajectory for the obstacle avoidance. On the other, to shown the effects of the uncertainties Figure 5.24 depicts the deterministic control variables and 5.25 and 5.26 represent the state variables. All the solutions only correspond considering a level probability $\alpha = 0.95$ for IA and OA with the value of $\tau = 0.1$ and $\tau = 0.0002$, respectively. As can be seen, the influence of the stochastic variables is greater in the velocity $\dot{y}$, yaw angle ψ and yaw rate $\dot{\psi}$.

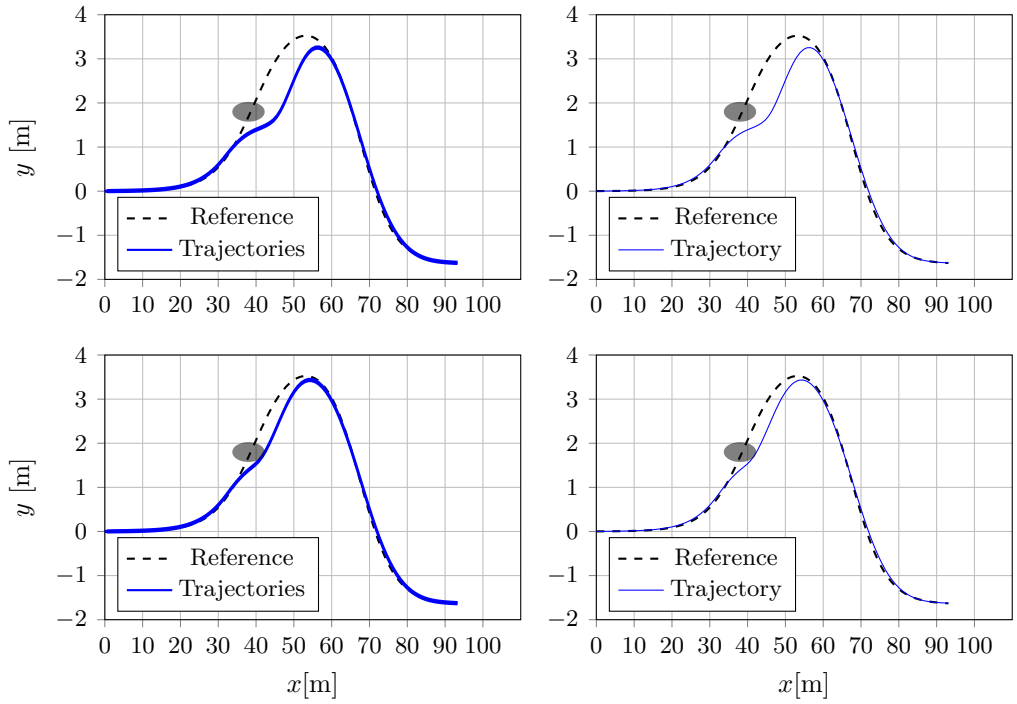

Figure 5.22 – Simulation results for the case 2: Trajectory in the X-Y plane for obstacle avoidance (up: Inner Approximation $\tau = 0.1$)and (down: Outer Approximation $\tau = 0.0002$) with $\alpha = 0.90$.(left:N_s Trajectories of the simulation QSM) and (right: Expected Value of the trajectories)

Table 5.8 – Average computation time in IpOpt per iteration (seconds) required for the solution of case 2.

α	Inner	CPU time	Outer	CPU time
0.80	$\tau = 1$	0.277	$\tau = 0.1$	0.276
	$\tau = 0.5$	0.284	$\tau = 0.01$	0.271
	$\tau = 0.1$	0.276	$\tau = 0.001$	0.289
			$\tau = 0.0002$	0.284
0.9	$\tau = 1$	0.281	$\tau = 0.1$	0.277
	$\tau = 0.5$	0.294	$\tau = 0.01$	0.283
	$\tau = 0.1$	0.310	$\tau = 0.001$	0.286
			$\tau = 0.0002$	0.294
0.95	$\tau = 1$	0.282	$\tau = 0.1$	0.273
	$\tau = 0.5$	0.281	$\tau = 0.01$	0.278
	$\tau = 0.1$	0.270	$\tau = 0.001$	0.284
			$\tau = 0.0002$	0.275

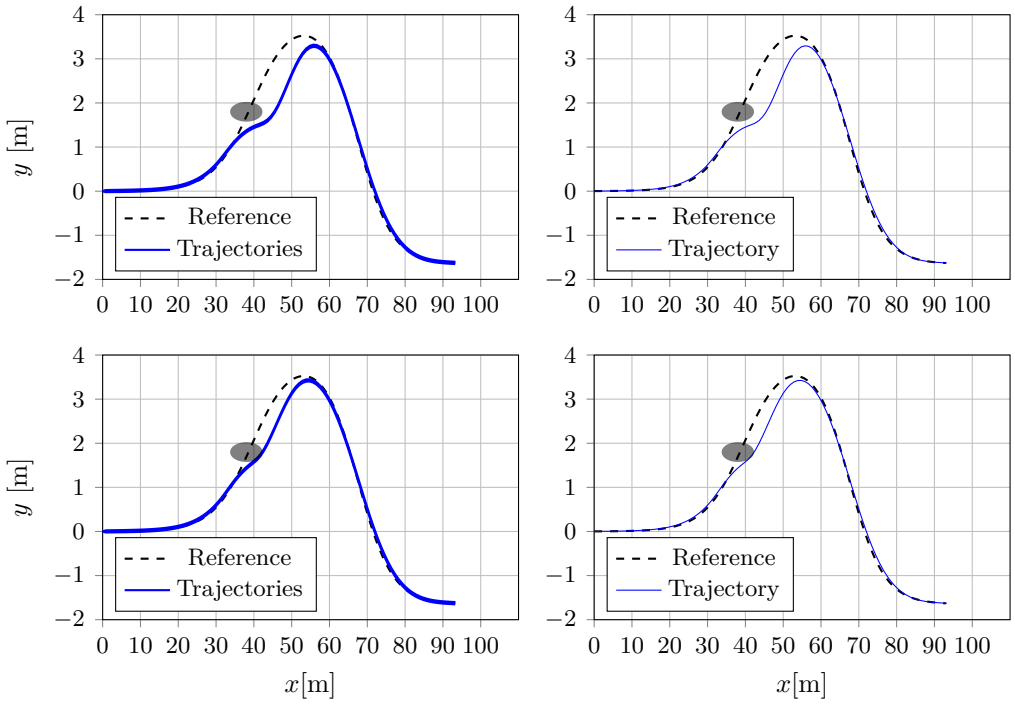

Figure 5.23 – Simulation results for the case 2: Trajectory in the X-Y plane for obstacle avoidance (up: Inner Approximation $\tau = 0.1$)and (down: Outer Approximation $\tau = 0.0002$) with $\alpha = 0.80$.(left:N_s Trajectories of the simulation QSM) and (right: Expected Value of the trajectories)

Table 5.9 – Average computation time in NLP function evaluations per iteration (seconds) required for the solution of case 2.

α	Inner	CPU time	Outer	CPU time
0.80	$\tau = 1$	5.524	$\tau = 0.1$	5.859
	$\tau = 0.5$	5.540	$\tau = 0.01$	5.800
	$\tau = 0.1$	5.615	$\tau = 0.001$	6.069
			$\tau = 0.0002$	6.017
0.9	$\tau = 1$	5.552	$\tau = 0.1$	6.048
	$\tau = 0.5$	5.637	$\tau = 0.01$	5.810
	$\tau = 0.1$	6.078	$\tau = 0.001$	5.898
			$\tau = 0.0002$	6.016
0.95	$\tau = 1$	5.652	$\tau = 0.1$	5.916
	$\tau = 0.5$	5.638	$\tau = 0.01$	5.958
	$\tau = 0.1$	5.868	$\tau = 0.001$	6.245
			$\tau = 0.0002$	5.870

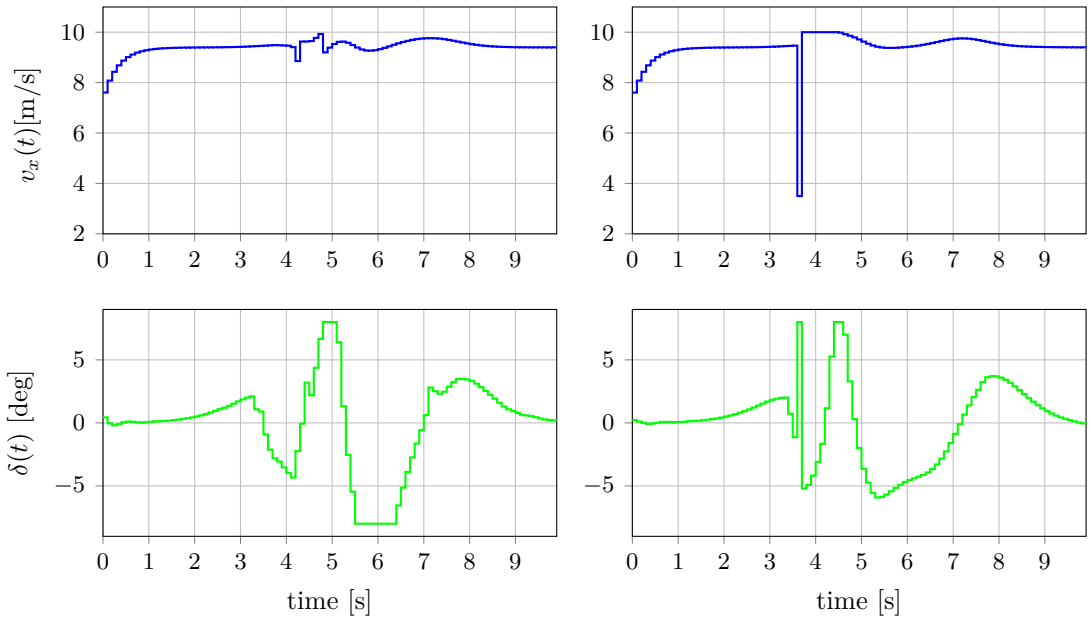

Figure 5.24 – Simulation results for the Case 2: Optimal control inputs for the obstacle avoidance (left: Inner Approximation $\tau = 0.1$)and (right: Outer Approximation $\tau = 0.0002$) with $\alpha = 0.95$

As can be seen in Figure 5.21, 5.22 and 5.23 the feasibility of the OA is bigger that the IA. As a result, the trajectory of the vehicle in OA is closer to obstacle avoidance than IA. Moreover, if the probability level is increased, the feasible trajectory of the vehicle decreases.

Moreover, as can be seen in Figure 5.27 , as τ decreases, the optimal value of the objective function tends to be the same for both approaches. Table 5.8 and 5.9 ilustrate the CPU times in both approach where the average time reported in IA and OA are significant respect to the deterministic optimization 5.7.

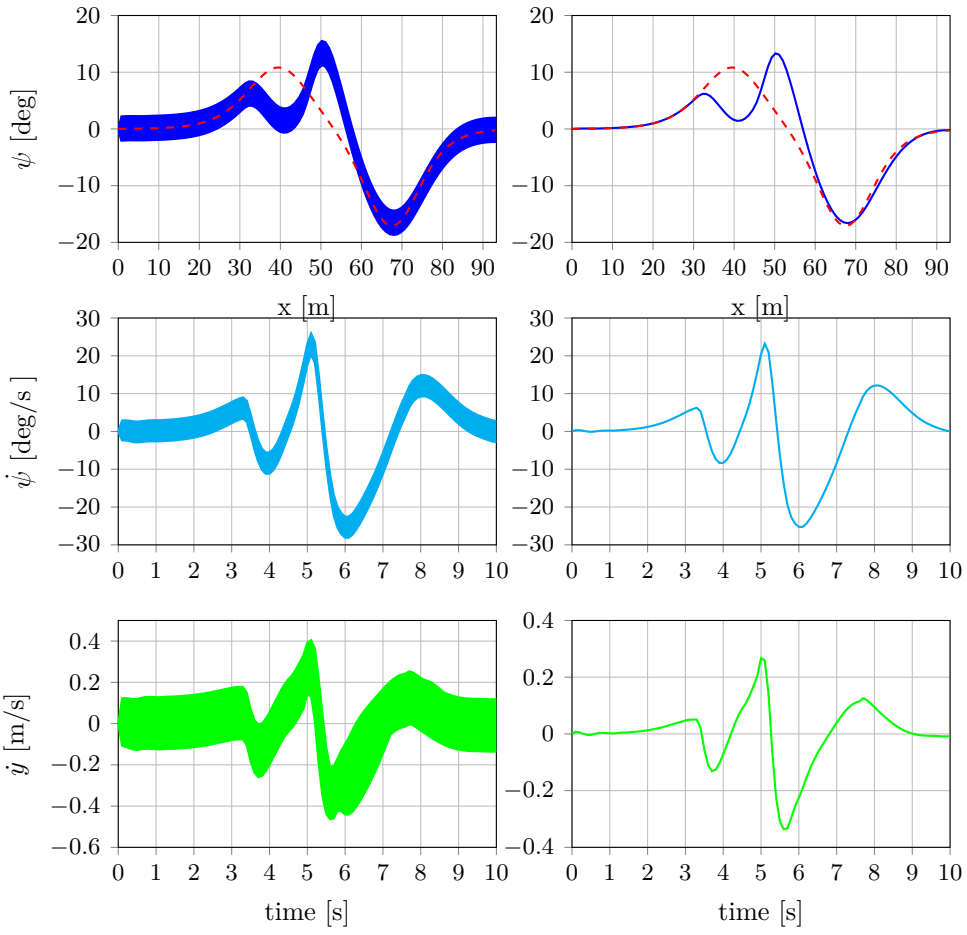

Figure 5.25 – Simulation results for the case 2: Optimal states for the obstacle avoidance using IA ($\tau = 0.1$) with $\alpha = 0.95$. (left: N_s trajectories of the simulation QSM) and (right: expected value of the trajectories)

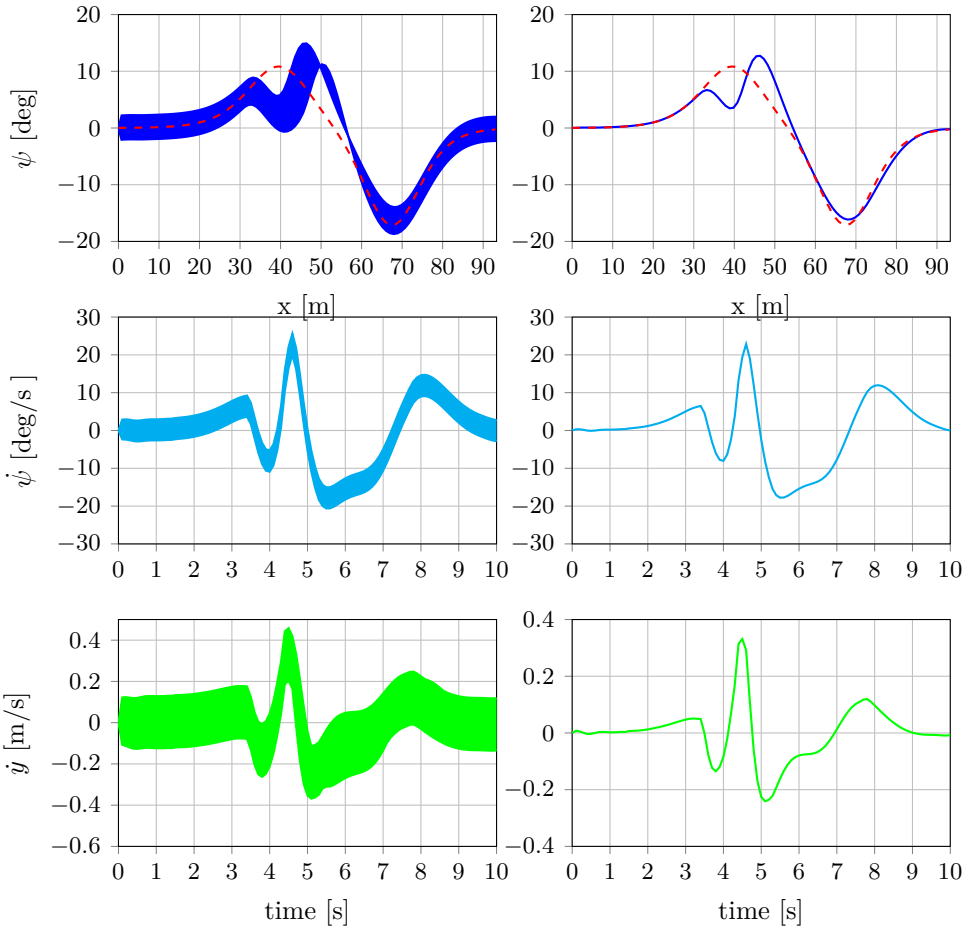

Figure 5.26 – Simulation results for the case 2: Optimal states for the obstacle avoidance using OA ($\tau = 0.0002$) with $\alpha = 0.95$. (left: N_s trajectories of the simulation QSM) and (right: expected value of the trajectories)

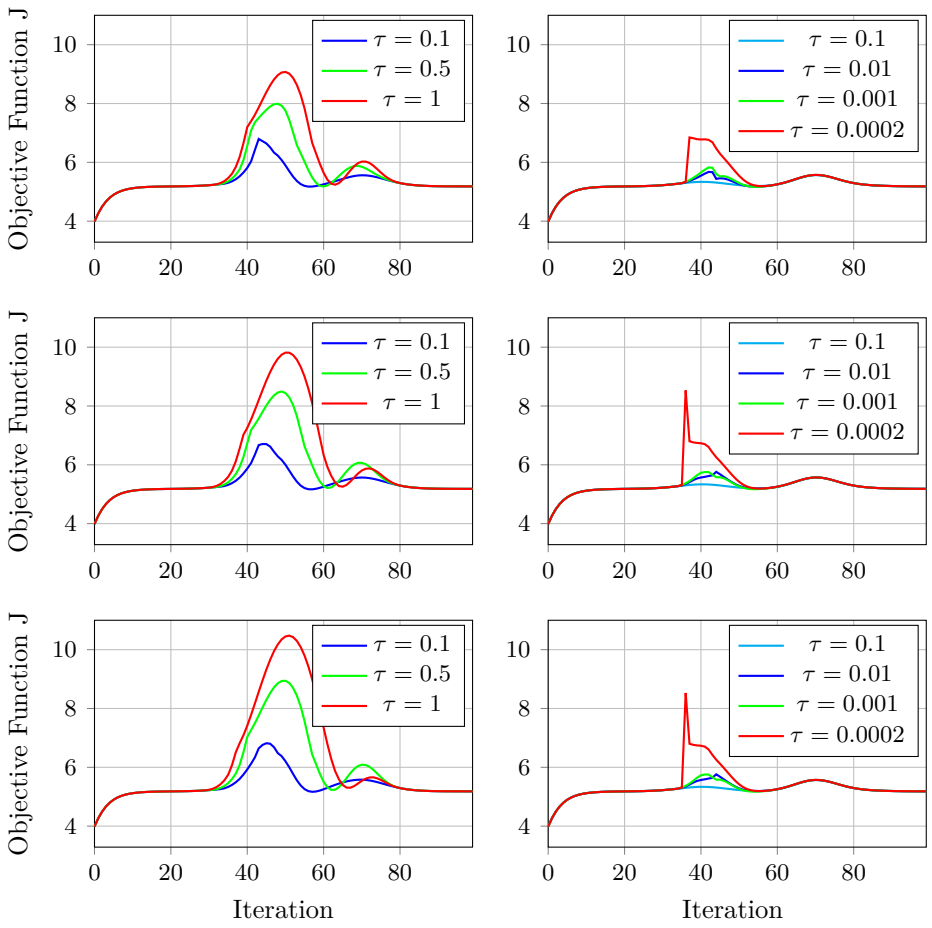

Figure 5.27 – Simulation results for the case 2: Optimal value of the objective function for different values of α (left: Inner Approximation) and (right: Outer Approximation). From top to bottom: $\alpha = 0.8$, $\alpha = 0.9$, $\alpha = 0.95$

5.3 Case 3: Bicycle vehicle model with nonlinear tire lateral model

This model equation is based on Bicycle Model with non-linear equation Tire forces.Where F_{yf} and F_{yr} are front and rear tires force acting on the vehicle lateral axis, F_{xf} and F_{xr} tire forces acting on the vehicle longitudinal axis. The vehicle sketch is represented in Figure 5.29. The state vector is represented by $\mathbf{X} = [\dot{x} \quad \dot{y} \quad \dot{\psi} \quad \psi \quad X \quad Y]$. Likewise to previous model, the yaw rate is considered $\dot{\psi}$, while the control variable is represented by $\mathbf{U} = [\delta \quad \beta]$, where δ is the front steering angle and β is referred as braking ratio, where $\beta = -1$ corresponds to full braking and $\beta = 1$ to full throttle. In addition, the stochastic variables are considered in the model $\xi = [\xi_1 \quad \xi_2 \quad \xi_3 \quad \xi_4 \quad \xi_5 \quad \xi_6]$ that represents the error of the model motion. The following set of differential equations to describe the vehicle motion.

$$\begin{aligned}
\ddot{x} &= \dot{y}\,\dot{\psi} + 2\,\frac{F_{xf}}{m} + 2\,\frac{F_{xr}}{m} + \xi_1\,, \\
\ddot{y} &= -\dot{x}\,\dot{\psi} + 2\frac{F_{yf}}{m} + 2\,\frac{F_{yr}}{m} + \xi_2\,, \\
\ddot{\psi} &= 2\,(\frac{l_f\,F_{yf} - l_r\,F_{yr}}{I_z}) + \xi_3\,, \\
\dot{\psi} &= \dot{\Psi} + \xi_4\,, \\
\dot{X} &= \dot{x}\cos(\,\Psi) - \dot{y}\sin(\,\Psi) + \xi_5\,, \\
\dot{Y} &= \dot{x}\sin(\,\Psi) + \dot{y}\cos(\,\Psi) + \xi_6\,,
\end{aligned} \tag{5.15}$$

The front and rear tires are expressed as follow:

$$\begin{aligned}
F_{xf} &= F_{lf}\cos(\,\delta) - F_{cf}\sin(\,\delta) \\
F_{yf} &= F_{lf}\sin(\,\delta) + F_{cf}\cos(\,\delta) \\
F_{xr} &= F_{lr} \\
F_{yr} &= F_{cr}
\end{aligned}$$

Where F_{lf}, F_{lr} are the longitudinal tire forces and F_{cf}, F_{cr} are the lateral tire forces for front and rear wheel. In this model, the longitudinal tire forces are considered linear due to the assumption that the slip ratio is small, as a result the input control β is constrained to $[-0.5 \quad 0.5]$ meanwhile the lateral tire forces is represented using Fiala model [43]. The tire forces are formulated as:

$$\begin{aligned} F_{lr} &= \beta\ \mu\ F_{zr} \\ F_{lf} &= \beta\ \mu\ F_{zf} \end{aligned}$$

$$F_{c,f/r} = \begin{cases} -C_\alpha \tan(\alpha_{f/r}) + \frac{C_\alpha^2\ |\tan(\alpha_{f/r})|\tan(\alpha_{f/r})}{2\ n\ \mu\ F_{z,f/r}} - \frac{C_\alpha^3\ \tan(\alpha_{f/r})^3}{27\ n^2\ \mu^2\ F_{z,f/r}} & |\ \alpha_{f/r}\ | < \alpha_{s,f/r} \\ -n\mu F_{z,f/r}\ sgn(\alpha_{f/r}) & |\ \alpha_{f/r}\ | \geq \alpha_{s,f/r} \end{cases}$$

Where $\alpha_{f/r}$ and $\alpha_{s,f/r} = \frac{3\ n\ \mu\ F_{z,f/r}}{C_\alpha}$ represent the tire slip angle and the saturation point of the slip angle, respectively. Moreover, u is the friction coefficient that is assumed constant on each wheel, C_α is the cornering stiffness and $n = \sqrt{1-\beta^2}$ can be represented in terms of β. The vertical forces $F_{z,f/r}$ are assumed constant and are calculated by steady weight distribution of the vehicle at the center of the gravity.

$$\begin{aligned} F_{zf} &= \frac{l_r\ m\ g}{2(l_r+l_f)} \\ F_{zr} &= \frac{l_f\ m\ g}{2(l_r+l_f)} \end{aligned}$$

The slip angle is formulated based on the longitudinal and lateral velocity.

$$\begin{aligned} \alpha_f &= atan\left(\frac{(\dot{y}\ + l_f\ \dot{\psi})\ \cos(\delta) - \dot{x}\ \sin(\delta)}{\dot{x}\ \cos(\delta) + (\dot{y}\ + l_f\ \dot{\psi})\sin(\delta)}\right) \\ \alpha_r &= atan\left(\frac{\dot{y} - l_f\ \dot{\psi}}{\dot{x}}\right) \end{aligned}$$

The aim of the optimization problem is to arrive at a desired final position while avoiding the obstacle that exists in the trajectory. The scenario for the third case is shown in Figure 5.28 where the desired position is $(x_d, y_d) = (108m, 0m)$ and the obstacle is located at $(x_{obs}, y_{obs}) = (60m, 0)$.

Thus, the objective function is formulated as follow.

$$J_{track}(x,u,\xi) = E[\sum_{k=1}^{N-1} q_X(X_{k,\xi} - x_{ref,k})^2 + q_Y(Y_{k,\xi} - y_{ref,k})^2 + q_\psi(\psi_{k,\xi} - \psi_{ref,k})^2] + r_\beta \beta_k^2 + r_\delta \delta_k^2 \tag{5.16}$$

Table 5.10 – Physical parameters of the Non-Linear Bicycle model [32]

Parameter	Value	Description
m	2050 kg	Mass of the vehicle
I_z	3344 kg.m^2	Inertia moment
l_f	1.4 m	Distance from COG to front axle
l_r	1.0 m	Distance from COG to rear axle
C_α	65000 N/rad	Cornering stiffness
μ	0.5	Friction coefficient

Table 5.11 – Parameters for the uncertain inputs obtained from [32]

Parameter	Value	Description
σ_{ξ_1}	0.2	Standard deviation
σ_{ξ_2}	0.2	Standard deviation
σ_{ξ_3}	0.2	Standard deviation
σ_{ξ_4}	0.005	Standard deviation
σ_{ξ_5}	0.05	Standard deviation
σ_{ξ_6}	0.05	Standard deviation

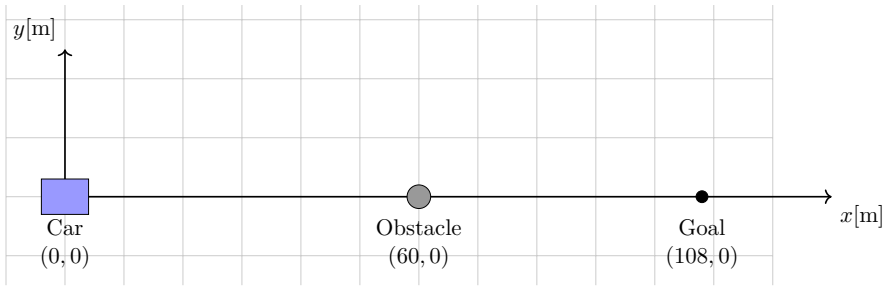

Figure 5.28 – Mobile working area and obstacle to be avoided for the Non-linear Bicycle Model

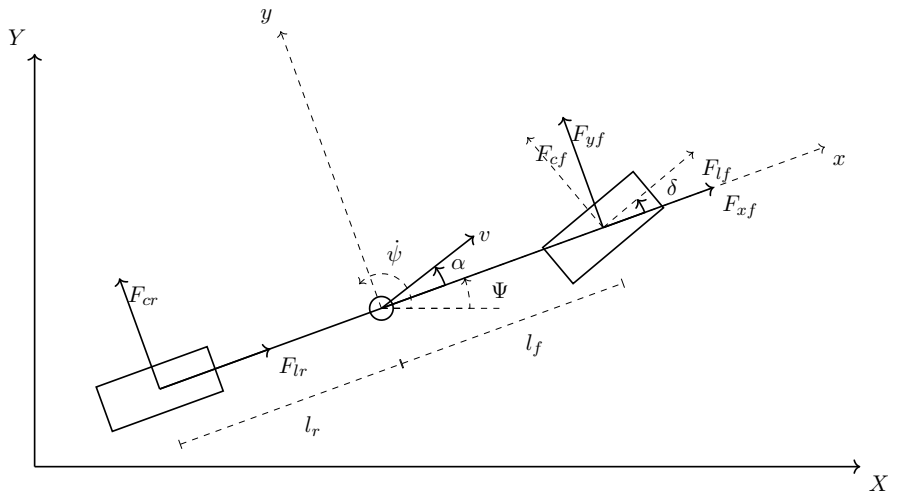

Figure 5.29 – Bicycle vehicle model of the vehicle [66]

The state vector of initial conditions is given by.

$$x(0) = [3,\ 0,\ 0,\ 0,\ 0,\ 0]^T\,. \tag{5.17}$$

the final time is fixed to $t_f = 12$ s and the sampling time is set to $\Delta T = 0.1$s. Apply Runge Kutta Methods to equation 5.15 to obtain the discrete-time model. The control variables are constrained as follows:

$$\begin{aligned} -5^\circ \leq\ & \delta & \leq 5^\circ\,, \\ -0.5^\circ \leq\ & \Delta\delta & \leq 0.5^\circ\,, \\ -0.5 \leq\ & \beta & \leq 0.5\,, \\ -0.01 \leq\ & \Delta\beta & \leq 0.01\,, \end{aligned} \tag{5.18}$$

The chance constraint is formulated for obstacle avoidance.

$$Pr[1 - \frac{(x_{k,\xi} - x_{obs})^2}{r_a^2} - \frac{(y_{k,\xi} - y_{obs})^2}{r_b^2} \leq 0] \geq \alpha, \quad k = 1, 2, \ldots, N \tag{5.19}$$

the resulting Stochastic NMPC problem is given by:

$$\begin{aligned} \min_{u} \quad & J_{track}(x, u, \xi) \\ \text{subject to:} \quad & x_0 = x(0)\,, \\ & \text{Model equation (5.15)}\,, \\ & \text{Path constraint (5.18)}\,, \\ & \text{Chance constraint (5.19)}\,. \end{aligned}$$

5.3.1 Deterministic MPC for bicycle vehicle model with nonlinear tire lateral model

The prediction horizon considered is of $N = 8$, the gain factors in the objective function are set $q_x = 1$, $q_y = 1$, $q_\psi = 1$, $r_\beta = 0.01$ and $r_\delta = 0.01$ and the parameter of the ellipse are set to $r_a = 4m$ and $r_b = 0.4m$. The results of the obstacle avoidance problem are shown in Figure 5.30, the optimal controls in Figure 5.31 and the optimal states 5.32.

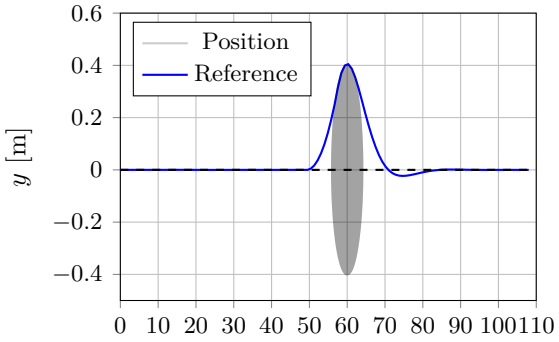

Figure 5.30 – Simulation result for the case 3: Trajectory in the X-Y plane for obstacle avoidance using Deterministic MPC

The CPU Time in deterministic optimization is reported in 5.12.
Similar to previous case, the results from the deterministic optimization cannot be applied since the constraint of the obstacle avoidance are not satisfied when the stochastic

Table 5.12 – Average computation time per iteration (ms) required for the solution of case 3 using deterministic MPC.

	Case 3
CPU time in IpOpt	1.433
CPU time in NLP function evaluations	20.466

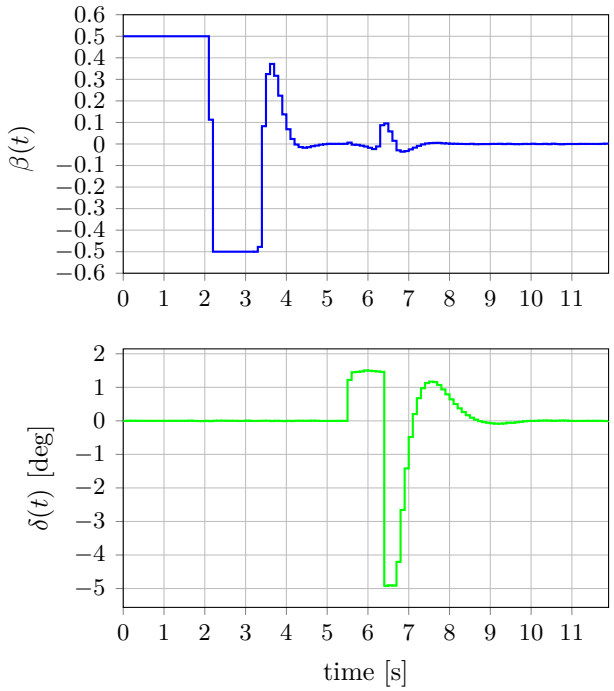

Figure 5.31 – Simulation results for the case 3: Optimal control inputs for obstacle avoidance using Deterministic MPC

variables are taken into account (Figure 5.34). Considering 10 000 samples generated by the Monte Carlo method, the deterministic optimization lead to about 71% of violations of the constraint.

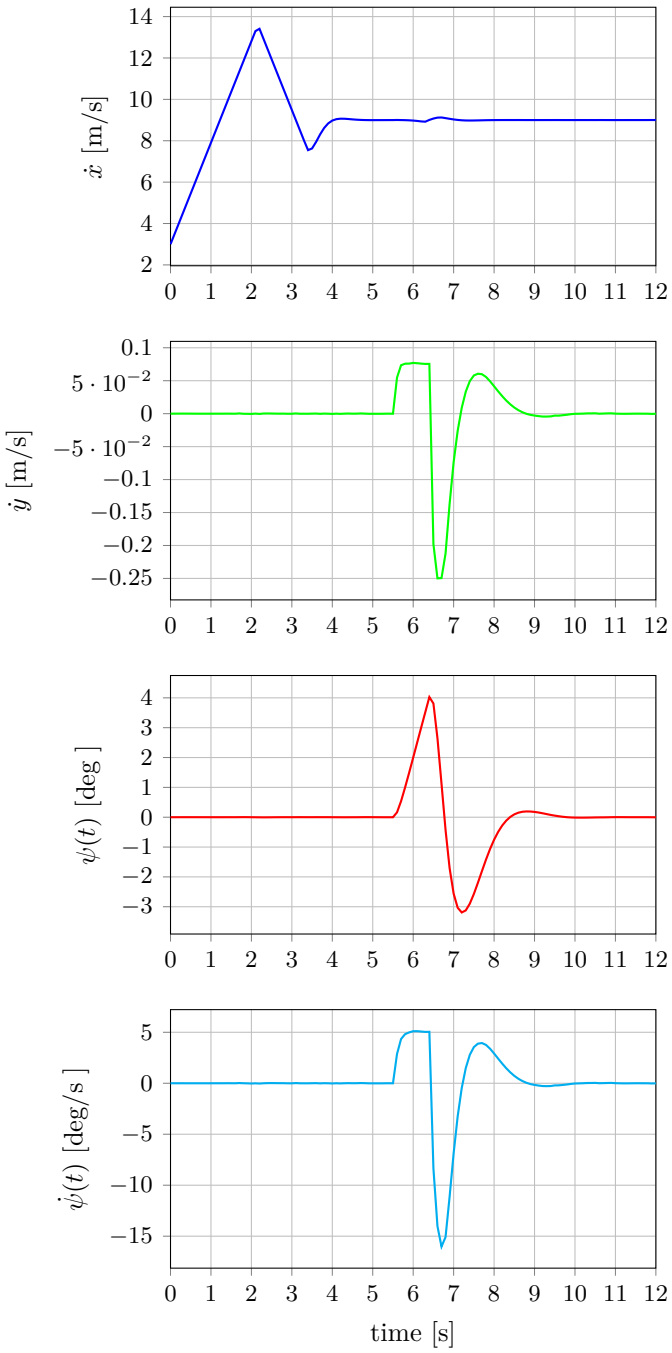

Figure 5.32 – Simulation results for the case 3: Optimal states for obstacle avoidance using Deterministic MPC

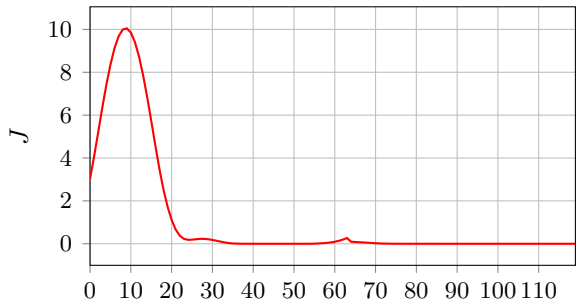

Figure 5.33 – Simulation results for the case 3: Convergence of the objective function using Deterministic MPC

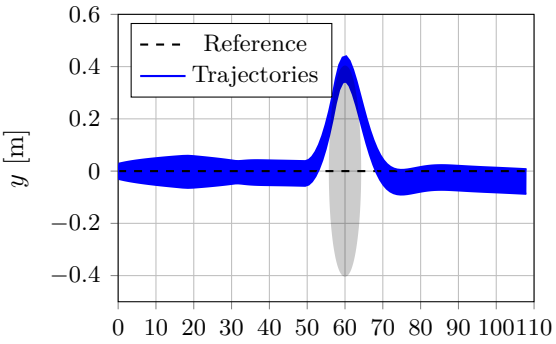

Figure 5.34 – Simulation result for the case 3: Trajectory in the X-Y plane for obstacle avoidance under uncertainties using Deterministic MPC.

5.3.2 Stochastic MPC for bicycle vehicle model with nonlinear tire lateral model

IA and OA approach was tested with a prediction horizon $N = 8$. For the case 3, the values of $m1 = 2$ and $m2 = 1$ of the parametric function were considered. In the case the inner, the results were obtained for $1 \leq \tau \leq 0.1$, and outer for $0.1 \leq \tau \leq 0.001$. Moreover, different level of probability were considered for $\alpha = 0.80, \quad 0.9, \quad 0.95$.

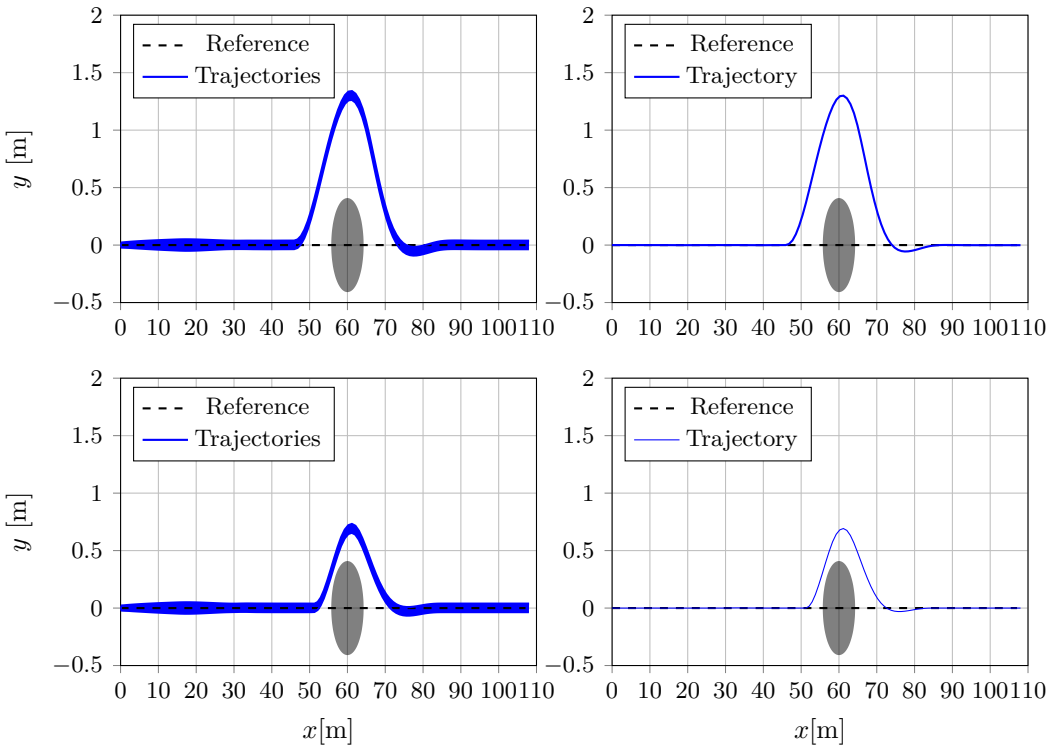

Figure 5.35 – Simulation results for the case 3: Trajectory in the X-Y plane for obstacle avoidance (up: IA $\tau = 0.1$)and (down:OA $\tau = 0.001$) with $\alpha = 0.95$.(left:N_s Trajectories of the simulation QSM) and (right: Expected Value of the trajectories)

The results of the stochastic optimization are shown in Figure 5.35, 5.36 and 5.37 that corresponds the feasible trajectory for the obstacle avoidance. As can be seen, the trajectory of the vehicle in OA is closer to obstacle avoidance than IA. Moreover, if the probability level is increased, the feasible trajectory of the vehicle decreases.
On the other, to shown the effects of the uncertainties Figure 5.38 depicts the deterministic control variables and Figure 5.39, 5.40 represent the state variables. All the solutions only correspond considering a level probability $\alpha = 0.95$ for IA and OA with

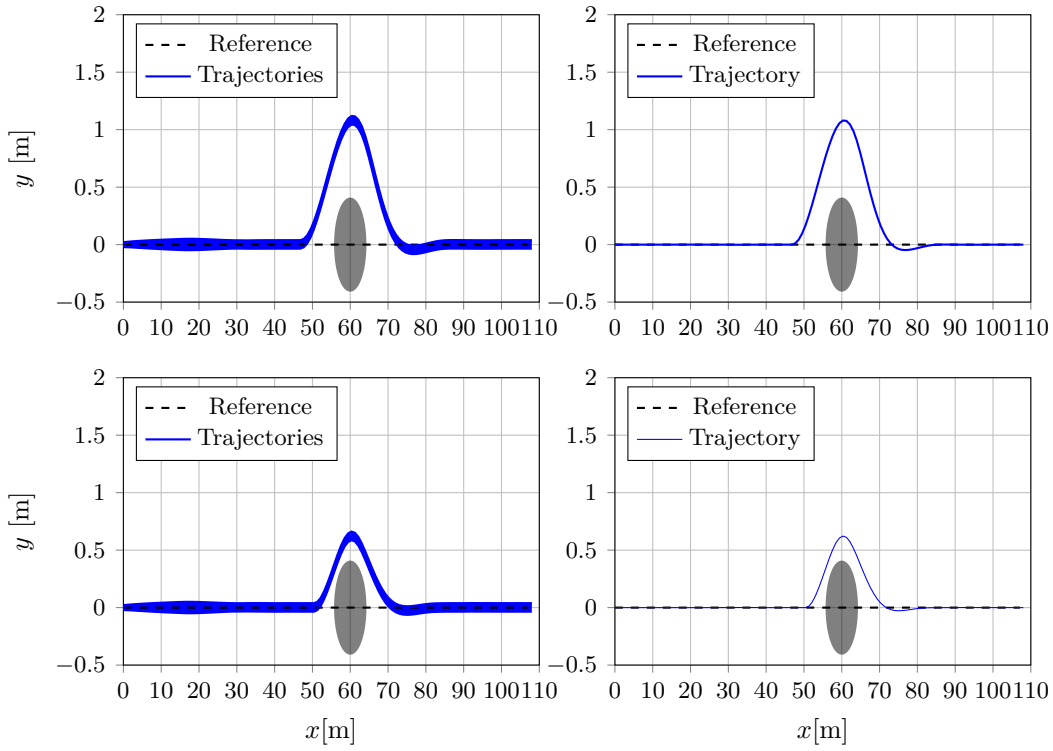

Figure 5.36 – Simulation results for the case 3: Trajectory in the X-Y plane for obstacle avoidance (up: IA $\tau = 0.1$)and (down:OA $\tau = 0.001$) with $\alpha = 0.90$.(left:N_s Trajectories of the simulation QSM) and (right: Expected Value of the trajectories)

the value of $\tau = 0.1$ and $\tau = 0.001$, respectively. As can be seen, the influence of the stochastic variables is greater in the velocity $\dot{y}$ and yaw angle ψ.

Table 5.13 – Average computation time in IpOpt per iteration (seconds) required for the solution of case 3.

α	IA	CPU time	OA	CPU time
0.8	$\tau = 1$	0.489	$\tau = 0.1$	0.506
	$\tau = 0.5$	0.499	$\tau = 0.01$	0.514
	$\tau = 0.1$	0.480	$\tau = 0.001$	0.511
0.9	$\tau = 1$	0.492	$\tau = 0.1$	0.469
	$\tau = 0.5$	0.497	$\tau = 0.01$	0.496
	$\tau = 0.1$	0.494	$\tau = 0.001$	0.492
0.95	$\tau = 1$	0.473	$\tau = 0.1$	0.493
	$\tau = 0.5$	0.499	$\tau = 0.01$	0.503
	$\tau = 0.1$	0.520	$\tau = 0.001$	0.508

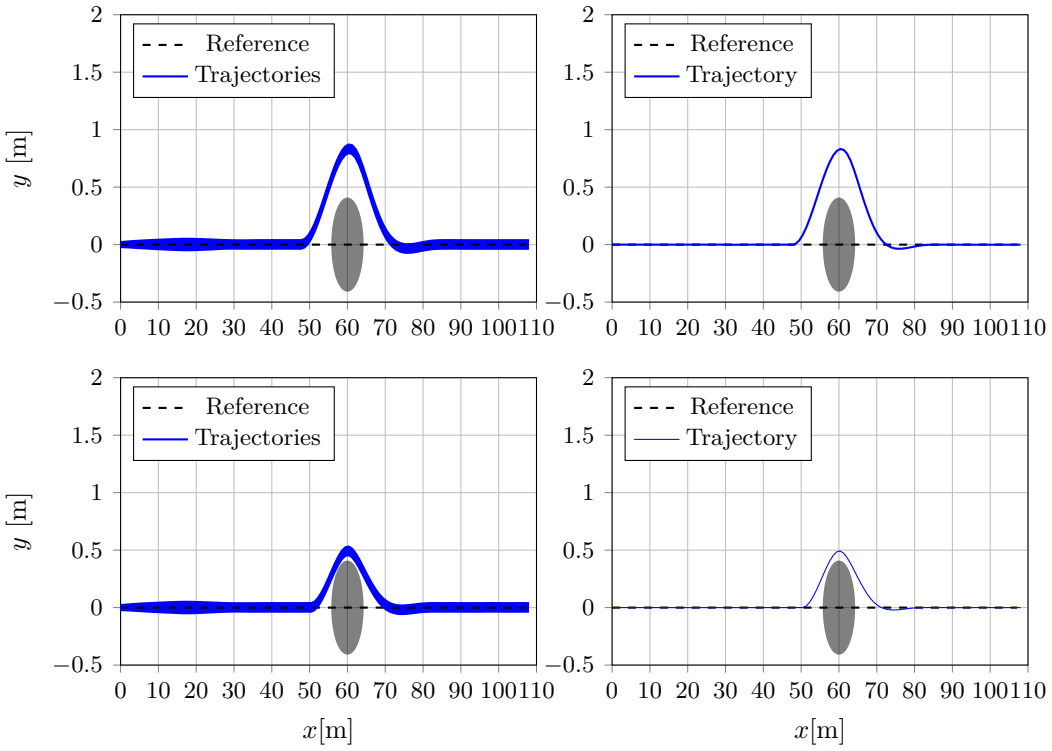

Figure 5.37 – Simulation results for the case 3: Trajectory in the X-Y plane for obstacle avoidance (up: IA $\tau = 0.1$)and (down:OA $\tau = 0.001$) with $\alpha = 0.80$.(left:N_s Trajectories of the simulation QSM) and (right: Expected Value of the trajectories)

Table 5.14 – Average computation time in NLP function evaluations per iteration (seconds) required for the solution of case 3.

α	IA	CPU time	OA	CPU time
0.8	$\tau = 1$	6.967	$\tau = 0.1$	7.156
	$\tau = 0.5$	6.869	$\tau = 0.01$	7.053
	$\tau = 0.1$	6.489	$\tau = 0.001$	7.129
0.9	$\tau = 1$	7.142	$\tau = 0.1$	6.909
	$\tau = 0.5$	7.149	$\tau = 0.01$	7.063
	$\tau = 0.1$	6.874	$\tau = 0.001$	7.215
0.95	$\tau = 1$	7.103	$\tau = 0.1$	7.154
	$\tau = 0.5$	7.104	$\tau = 0.01$	7.460
	$\tau = 0.1$	7.020	$\tau = 0.001$	7.813

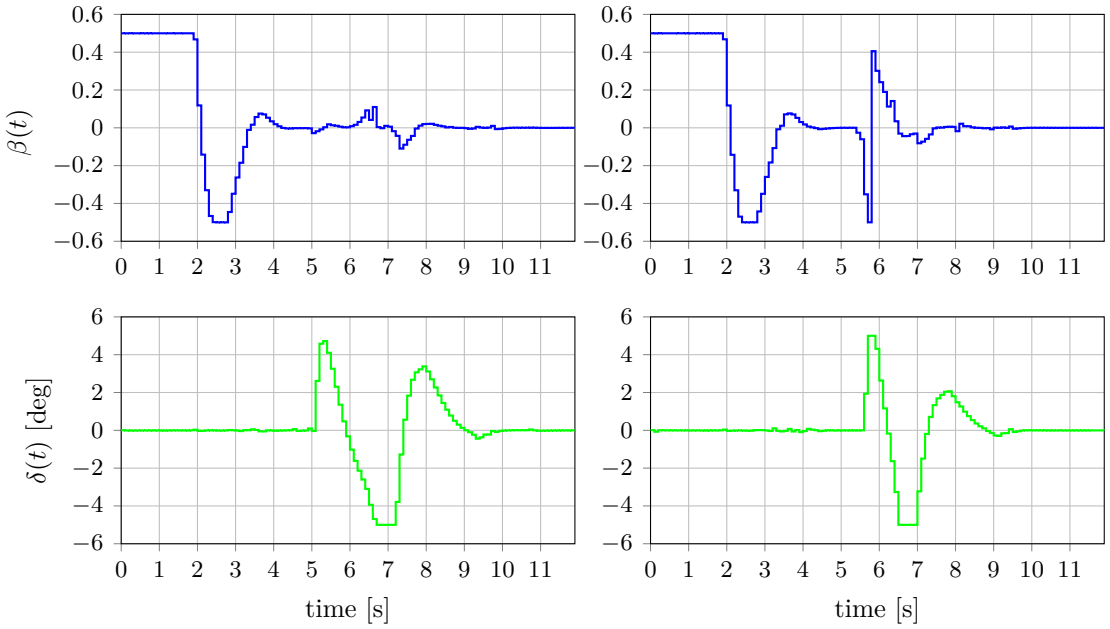

Figure 5.38 – Simulation results for the case 3: Optimal control inputs for the obstacle avoidance (left: Inner Approximation $\tau = 0.1$)and (right: Outer Approximation $\tau = 0.001$) with $\alpha = 0.95$

Moreover, as can be seen (Figure 5.41) , as τ decreases, the optimal value of the objective function tends to be the same for both approaches. Table 5.13 and 5.14 ilustrate the CPU times in both approach where the average time reported in IA and OA are significant respect to the deterministic optimization 5.12.

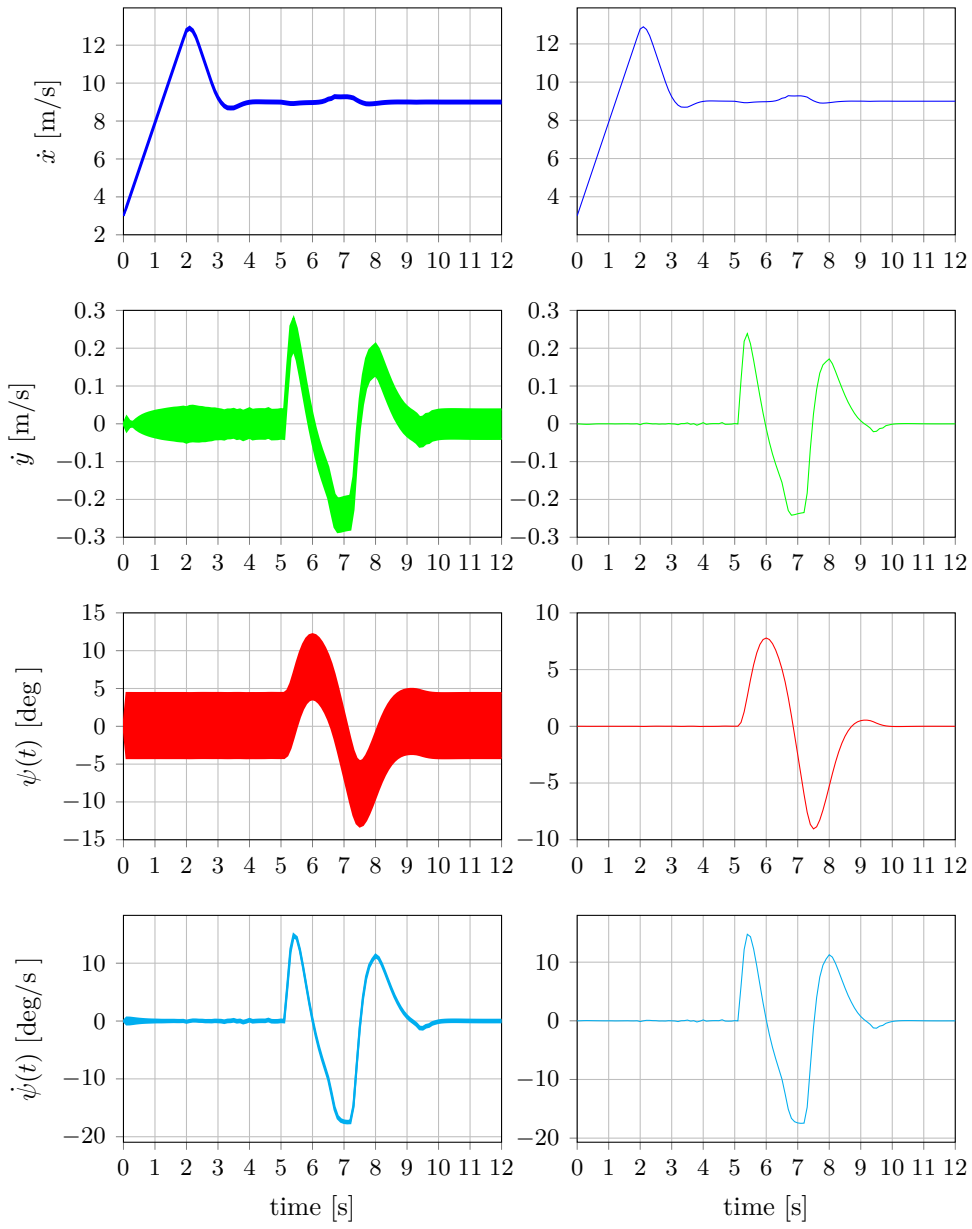

Figure 5.39 – Simulation results for the case 3: Optimal states for the obstacle avoidance using IA ($\tau = 0.1$) with $\alpha = 0.95$. (left: N_s trajectories of the simulation QSM) and (right: expected value of the trajectories)

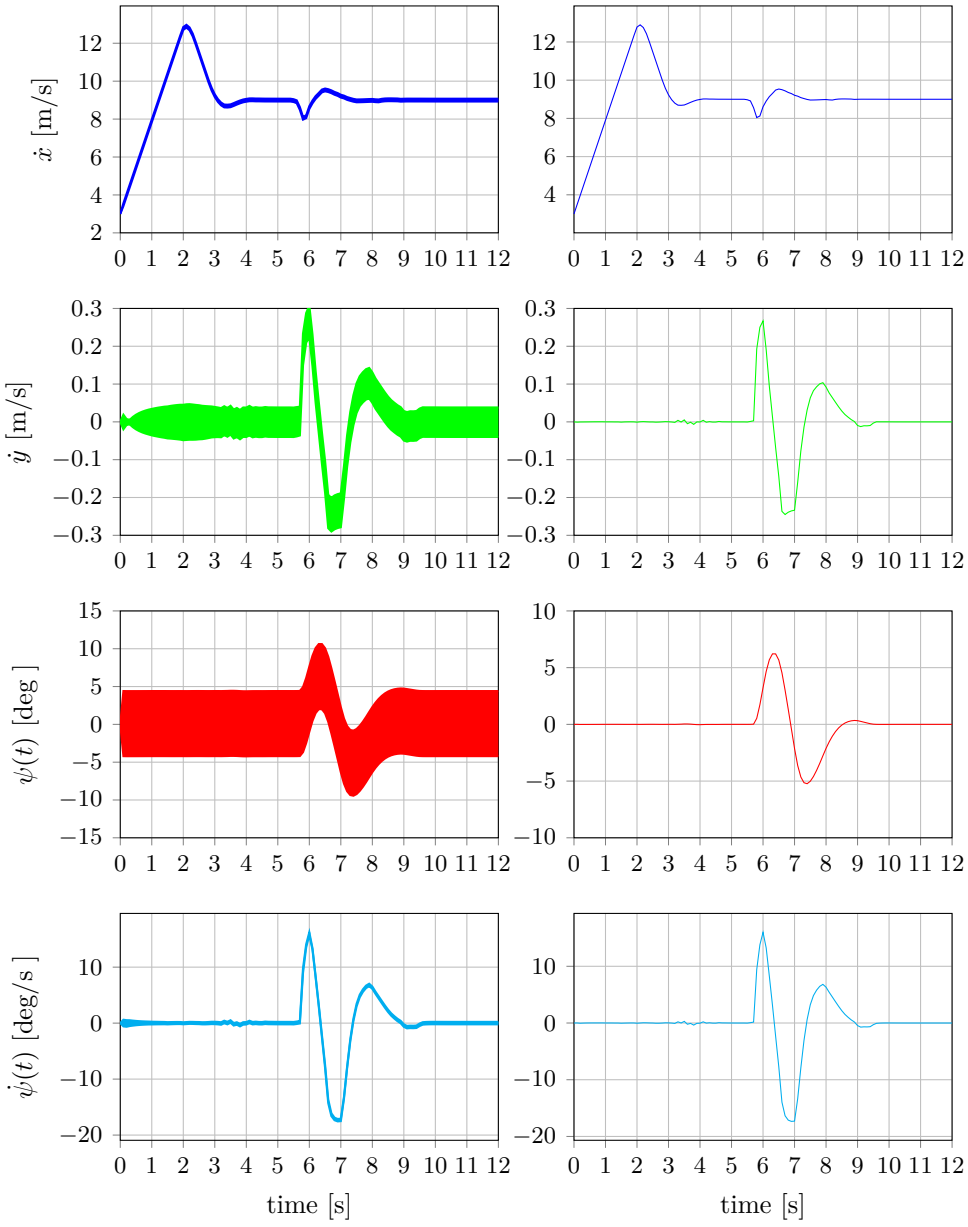

Figure 5.40 – Simulation results for the case 3: Optimal states for the obstacle avoidance using OA ($\tau = 0.001$) with $\alpha = 0.95$. (left: N_s trajectories of the simulation QSM) and (right: expected value of the trajectories)

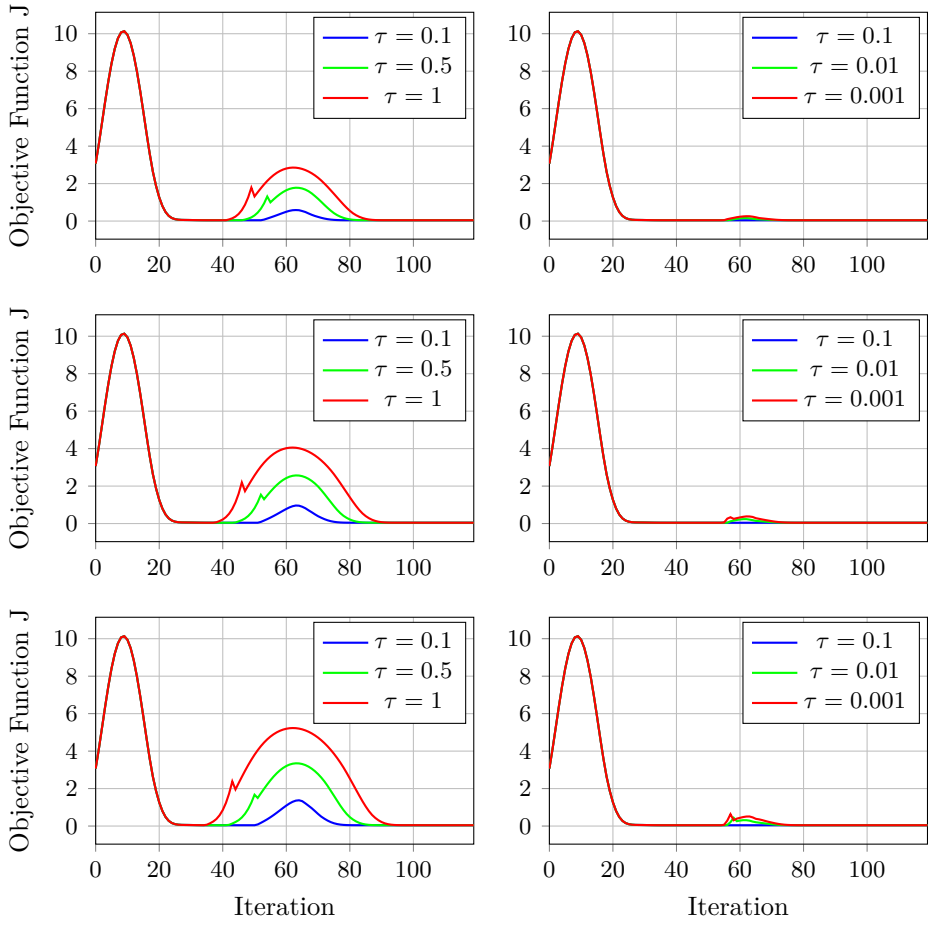

Figure 5.41 – Simulation results for the case 3: Optimal value of the objective function for different values of α (left: Inner Approximation) and (right: Outer Approximation). From top to bottom: $\alpha = 0.8$, $\alpha = 0.9$, $\alpha = 0.95$

Chapter 6

Conclusions and Future Work

6.1 Conclusions and Future Work

The work presented in this thesis is focused on the implementations of algorithms for AV considering uncertainties. The uncertainties arise due to the error model equation, measurement error or external disturbance. To solve a CCOPT two approaches (IA and OA) proposed in [33] were studied and formulated. This approximation method uses a parametric function that allows relaxing the CCOPT into NLP. As a result, it can be solved by available solver based on IPM or ASM.

The implementations are based on IpOpt with interface C++ and the computation of the Gradient and Jacobian are achieved by using CASADI. Three case studies with stochastic vehicle model are presented to test the performance of the algorithm proposed with 10 000 samples generated by QSM. Using DMPC the deterministic optimization lead to about 41.8%, 41% and 71% of violations of the constraint for the case 1, 2 and 3, respectively. Therefore, the deterministic optimization cannot be used due to high level of failure.

The simulation using SMPC are tested for IA and OA, in which the set of feasibility for OA is greater than IA. As a result, the objective function (IA) is bigger that OA.As τ decreases, the optimal value of the objective function tends to be the same for both approaches. Moreover, different levels of probability are considered. As can be seen in the results, when the probability is increased the objective function and computation time required also increase.

For the cases of OA,it has been possible to simulate with a τ closer to zero. Therefore, it is closer to the feasibility of the real problem. However, for the case of IA it has been possible to simulate up to a value of $\tau = 0.1$. As a result, the feasibility of the real problem is not very close and this is reflected in the feasible trajectories obtained.

Finally, the obtained computational time in stochastic optimization is significant in comparison to the time obtained in deterministic optimization.

6.2 Future Work

As mentioned above, the results in IA can be improved if it is simulated with a tau closer to zero, so a solver with more precision than IpOpt is required. Respect to the computation time obtained, parallel solver for real time application should be considered to reduce the burden computation. On other hand, longitudinal control should be considered with nonlinear tire longitudinal models for a better yaw stabilization, this requires the power train model.

Bibliography

[1] Automated driving . https://www.sae.org/misc/pdfs/automated_driving.pdf. Accessed: 2017-04-04.

[2] Google's Driverless Project Car. https://9to5google.files.wordpress.com/2015/06/google-driverless-car.png. Accessed: 2017-04-04.

[3] SUMMIT Mobile Robot. http://de.manu-systems.com/RBK-SUMMIT.shtml. Accessed: 2016-06-12.

[4] Abbas, M. A., Milman, R., and Eklund, J. M. (2014). Obstacle avoidance in real time with nonlinear model predictive control of autonomous vehicles. In *Electrical and Computer Engineering (CCECE), 2014 IEEE 27th Canadian Conference on*, pages 1–6. IEEE.

[5] Aburajabaltamimi, J. (2011). *Development of efficient algorithms for model predictive control of fast systems*. PhD thesis, Ilmenau, Technische Universität Ilmenau, Diss., 2011.

[6] Andersson, J. (2013). *A General-Purpose Software Framework for Dynamic Optimization*. PhD thesis, Arenberg Doctoral School, KU Leuven, Department of Electrical Engineering (ESAT/SCD) and Optimization in Engineering Center, Kasteelpark Arenberg 10, 3001-Heverlee, Belgium.

[7] Arora, R. K. (2015). *Optimization: algorithms and applications*. CRC Press.

[8] Astfalk, G., Lustig, I., Marsten, R., and Shanno, D. (1992). The interior-point method for linear programming. *IEEE software*, 9(4):61–68.

[9] Attia, R., Orjuela, R., and Basset, M. (2012). Coupled longitudinal and lateral control strategy improving lateral stability for autonomous vehicle. In *2012 American Control Conference (ACC)*, pages 6509–6514.

[10] Attia, R., Orjuela, R., and Basset, M. (2014). Combined longitudinal and lateral control for automated vehicle guidance. *Vehicle System Dynamics*, 52(2):261–279.

[11] Behrendt, M. (2009). A basic working principle of model predictive control.

[12] Bhatti, M. A. (2000). *Practical optimization methods with Mathematica applications*. Springer, 1 edition.

[13] Blackmore, L. (2006). A probabilistic particle control approach to optimal, robust predictive control. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, number 10.

[14] Blackmore, L. and Williams, B. C. (2007). Optimal, robust predictive control of nonlinear systems under probabilistic uncertainty using particles. In *Proceedings of the American Control Conference*, volume 17. Citeseer.

[15] Bock, H. G. (1981). Numerical treatment of inverse problems in chemical reaction kinetics. In *Modelling of chemical reaction systems*, pages 102–125. Springer.

[16] Bratley, P., Fox, B. L., and Niederreiter, H. (1994). Programs to generate niederreiter's low-discrepancy sequences. *ACM Transactions on Mathematical Software (TOMS)*, 20(4):494–495.

[17] 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. In *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*, pages 2335–2340. IEEE.

[18] Carvalho, A., Gao, Y., Lefevre, S., and Borrelli, F. (2014). Stochastic predictive control of autonomous vehicles in uncertain environments. In *12th International Symposium on Advanced Vehicle Control*.

[19] Castronuovo, E. D., Campagnolo, J. M., and Salgado, R. (2001). New versions of interior point methods applied to the optimal power flow problem. *IEEE Transactions on Power Systems*, 16:3.

[20] Correa Cordova, M. L. (2016). High performance implementation of mpc schemes for fast systems. Masterarbeit, Technische Universität Ilmenau.

[21] Dariani, R., Schmidt, S., and Kasper, R. (2014). Optimization based obstacle avoidance. *Optimization*, 1:9999297.

[22] Drozdova, E., Hopfgarten, S., Lazutkin, E., and Li, P. (2016). Autonomous driving of a mobile robot using a combined multiple-shooting and collocation method. *IFAC-PapersOnLine*, 49(15):193–198.

[23] Falcone, P. (2007). Nonlinear model predictive control for autonomous vehicles.

[24] Falcone, P., Borrelli, F., Asgari, J., Tseng, H. E., and Hrovat, D. (2007a). A model predictive control approach for combined braking and steering in autonomous vehicles. In *Control & Automation, 2007. MED'07. Mediterranean Conference on*, pages 1–6. IEEE.

[25] Falcone, P., Borrelli, F., Asgari, J., Tseng, H. E., and Hrovat, D. (2007b). Predictive active steering control for autonomous vehicle systems. *IEEE Transactions on control systems technology*, 15(3):566–580.

[26] Falcone, P., Borrelli, F., Tseng, H. E., Asgari, J., and Hrovat, D. (2008a). Linear time-varying model predictive control and its application to active steering systems: Stability analysis and experimental validation. *International journal of robust and nonlinear control*, 18(8):862–875.

[27] Falcone, P., Eric Tseng, H., Borrelli, F., Asgari, J., and Hrovat, D. (2008b). Mpc-based yaw and lateral stabilisation via active front steering and braking. *Vehicle System Dynamics*, 46(S1):611–628.

[28] Falcone, P., Tseng, H. E., Asgari, J., Borrelli, F., and Hrovat, D. (2007c). Integrated braking and steering model predictive control approach in autonomous vehicles. *IFAC Proceedings Volumes*, 40(10):273–278.

[29] Farina, M., Giulioni, L., and Scattolini, R. (2016). Stochastic linear model predictive control with chance constraints–a review. *Journal of Process Control*, 44:53–67.

[30] Farina, M. and Scattolini, R. (2016). Model predictive control of linear systems with multiplicative unbounded uncertainty and chance constraints. *Automatica*, 70:258–265.

[31] Feller, W. (2008). *An introduction to probability theory and its applications*, volume 2. John Wiley & Sons.

[32] Gao, Y. (2014). *Model Predictive Control for Autonomous and Semiautonomous Vehicles*. PhD thesis, University of California, Berkeley.

[33] Geletu, A., Klöppel, M., Hoffmann, A., and Li, P. (2015). A tractable approximation of non-convex chance constrained optimization with non-gaussian uncertainties. *Engineering Optimization*, 47(4):495–520.

[34] Geletu, A., Klöppel, M., Zhang, H., and Li, P. (2013). Advances and applications of chance-constrained approaches to systems optimisation under uncertainty. *International Journal of Systems Science*, 44(7):1209–1232.

[35] Gifei, S. and Salceanu, A. (2017). Integrated management system for quality, safety and security in developing autonomous vehicles. In *2017 10th International Symposium on Advanced Topics in Electrical Engineering (ATEE)*, pages 673–676.

[36] Gough, B. (2009). *GNU scientific library reference manual*. Network Theory Ltd.

[37] Goulart, P. J., Kerrigan, E. C., and Maciejowski, J. M. (2006). Optimization over state feedback policies for robust control with constraints. *Automatica*, 42(4):523–533.

[38] Gray, A., Gao, Y., Hedrick, J. K., and Borrelli, F. (2013). Robust predictive control for semi-autonomous vehicles with an uncertain driver model. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 208–213. IEEE.

[39] Grüne, L. and Pannek, J. (2017). Nonlinear model predictive control: Theory and algorithms.

[40] Hashimoto, T. (2013). Probabilistic constrained model predictive control for linear discrete-time systems with additive stochastic disturbances. In *52nd IEEE Conference on Decision and Control*, pages 6434–6439. IEEE.

[41] Hess, D. and Sattel, T. (2011). Double-lane change optimization for a stochastic vehicle model subject to collision probability constraints. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 206–211. IEEE.

[42] Himmelblau, D. M. (1972). *Applied nonlinear programming*. McGraw-Hill Companies.

[43] Hindiyeh, R. Y. and Gerdes, J. C. (2009). Equilibrium analysis of drifting vehicles for control design. In *ASME 2009 Dynamic Systems and Control Conference*, pages 181–188. American Society of Mechanical Engineers.

[44] Isaksson Palmqvist, M. (2016). Model predictive control for autonomous driving of a truck.

[45] Kanade, T., Thorpe, C., and Whittaker, W. (1986). Autonomous land vehicle project at cmu. In *Proceedings of the 1986 ACM fourteenth annual conference on Computer science*, pages 71–80. ACM.

[46] Katrakazas, C., Quddus, M., Chen, W.-H., and Deka, L. (2015). Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. *Transportation Research Part C: Emerging Technologies*, 60:416–442.

[47] Keviczky, T., Falcone, P., Borrelli, F., Asgari, J., and Hrovat, D. (2006a). Predictive control approach to autonomous vehicle steering. In *2006 American Control Conference*, pages 6–pp. IEEE.

[48] Keviczky, T., Falcone, P., Borrelli, F., Asgari, J., and Hrovat, D. (2006b). Predictive control approach to autonomous vehicle steering. In *American Control Conference, 2006*, pages 6–pp. IEEE.

[49] Klöppel, M. (2014). *Efficient numerical solution of chance constrained optimization problems with engineering applications*. PhD thesis.

[50] Lenz, D., Kessler, T., and Knoll, A. (2015). Stochastic model predictive controller with chance constraints for comfortable and safe driving behavior of autonomous vehicles. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 292–297. IEEE.

[51] Li, P., Arellano-Garcia, H., and Wozny, G. (2008). Chance constrained programming approach to process optimization under uncertainty. *Computers & Chemical Engineering*, 32(1):25–45.

[52] Li, P., Wendt, M., and Wozny, G. (2002). A probabilistically constrained model predictive controller. *Automatica*, 38(7):1171–1176.

[53] Liu, C., Gray, A., Lee, C., Hedrick, J. K., and Pan, J. (2014). Nonlinear stochastic predictive control with unscented transformation for semi-autonomous vehicles. In *2014 American Control Conference*, pages 5574–5579.

[54] Messac, A. (2015). *Optimization in Practice with MATLAB®: For Engineering Students and Professionals*. Cambridge University Press.

[55] Mousavi, M. A., Heshmati, Z., and Moshiri, B. (2013). Ltv-mpc based path planning of an autonomous vehicle via convex optimization. In *2013 21st Iranian Conference on Electrical Engineering (ICEE)*, pages 1–7. IEEE.

[56] Nemirovski, A. and Shapiro, A. (2006). Convex approximations of chance constrained programs. *SIAM Journal on Optimization*, 17(4):969–996.

[57] Nocedal, J., Wächter, A., and Waltz, R. A. (2009). Adaptive barrier update strategies for nonlinear interior methods. *SIAM Journal on Optimization*, 19(4):1674–1693.

[58] Norkin, V. I. (1993). The analysis and optimization of probability functions. *International Institute for Applied Systems Analysis technical report, Tech. Rep.*

[59] Oyelere, S. S. (2014). The application of model predictive control (mpc) to fast systems such as autonomous ground vehicles (agv). *IOSR J. Comput. Eng.(IOSR-JCE)*, 16(3):27–37.

[60] Paden, B., Cap, M., Yong, S. Z., Yershov, D., and Frazzoli, E. (2016). A survey of motion planning and control techniques for self-driving urban vehicles. *arXiv preprint arXiv:1604.07446*.

[61] Peden, M., Scurfield, R., Sleet, D., Mohan, D., Hyder, A. A., Jarawan, E., Mathers, C. D., et al. (2004). World report on road traffic injury prevention.

[62] Pintér, J. (1989). Deterministic approximations of probability inequalities. *Zeitschrift für Operations Research*, 33(4):219–239.

[63] Prandini, M., Garatti, S., and Lygeros, J. (2012). A randomized approach to stochastic model predictive control. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 7315–7320. IEEE.

[64] Qu, T., Chen, H., Ji, Y., Guo, H., and Xu, Y. (2016). A stochastic model predictive control approach for modelling human driver steering control. *International journal of vehicle design*, 70(3):249–277.

[65] Rafaila, R. C. and Livint, G. (2015). Nonlinear model predictive control of autonomous vehicle steering. In *System Theory, Control and Computing (ICSTCC), 2015 19th International Conference on*, pages 466–471. IEEE.

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

[67] Rawlings, J. B. and Mayne, D. Q. (2009). *Model predictive control: Theory and design*. Nob Hill Pub.

[68] Rockafellar, R. T. and Uryasev, S. (2000). Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42.

[69] Schenk, O., Wächter, A., and Hagemann, M. (2005). Combinatorial approaches to the solution of saddle-point problems in large-scale parallel interior-point optimization. Technical report, Technical Report RC 23824, IBM TJ Watson Research Center, Yorktown, USA.

[70] Teatro, T. A., Eklund, J. M., and Milman, R. (2014). Nonlinear model predictive control for omnidirectional robot motion planning and tracking with avoidance of moving obstacles. *Canadian Journal of Electrical and Computer Engineering*, 37(3):151–156.

[71] Thorpe, C., Hebert, M. H., Kanade, T., and Shafer, S. A. (1988). Vision and navigation for the carnegie-mellon navlab. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(3):362–373.

[72] Wächter, A. (2002). *An interior point algorithm for large-scale nonlinear optimization with applications in process engineering*. PhD thesis, PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA.

[73] Wächter, A. and Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1):25–57.

[74] Waechter, A., Laird, C., Margot, F., and Kawajir, Y. (2009). Introduction to ipopt: A tutorial for downloading, installing, and using ipopt. *Revision*.

[75] Wright, S. J. (1997). *Primal-dual interior-point methods*. SIAM.

[76] Xu, B., Stilwell, D. J., and Kurdila, A. (2010). A receding horizon controller for motion planning in the presence of moving obstacles. In *ICRA*, pages 974–980.

[77] Yu, S., Qu, T., Xu, F., and Chen, H. (2015). Model predictive control of linear stochastic systems with constraints. In *2015 American Control Conference (ACC)*, pages 950–955. IEEE.