Stochastic Model Predictive Control via Quantisation with Safety Certification

Veronika Tajgler

Master of Science Thesis

Image generated by ChatGPT (OpenAI).

Delft University of Technology

Delft Center for Systems and Control

Stochastic Model Predictive Control via Quantisation with Safety Certification

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

Veronika Tajgler

July 15, 2025

Faculty of Mechanical Engineering · Delft University of Technology

Delft University of Technology
Department of
Delft Center for Systems and Control (dcsc)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical Engineering for acceptance a thesis entitled

Stochastic Model Predictive Control via Quantisation with Safety Certification

by

Veronika Tajgler

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: July 15, 2025

Supervisor(s): ____________________
Dr. L. Laurenti

____
Eduardo Figueiredo Mota Diniz Costa

Reader(s): ____________________
Dr. Meichen Guo

Abstract

The demand for autonomous systems in safety-critical domains has increased in recent years. As real-world systems grow in complexity, a key challenge is ensuring robust performance under uncertainty, which requires the synthesis of controllers that not only operate reliably in stochastic environments but also support post-hoc validation and certification of safety. To this end, Stochastic Model Predictive Control (SMPC) is a control framework suitable for systems subjected to stochastic disturbances and model uncertainties. However, its practical application is limited by the intractability of exact uncertainty propagation of the state distribution (i.e. the prediction step), particularly for nonlinear dynamics, incentivising research to focus on approximation methods such as linearisation or Monte Carlo sampling. In this literature, however, the approximations do not provide guarantees of correctness.

This work addresses this particular limitation by leveraging quantisation-based uncertainty propagation, where both the state and disturbance distributions are discretised with formal guarantees in the Wasserstein distance. We formulate a quantised SMPC algorithm for discrete-time nonlinear systems with Gaussian additive noise, subject to individual chance constraints. For safety certification of the obtained controller, we introduce a validation scheme based on Wasserstein ambiguity sets that estimate worst-case constraint violation probabilities. The proposed approach is evaluated in simulation on benchmark tasks under both open- and closed-loop policies.

Table of Contents

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my first supervisor, Dr. Luca Laurenti, for his expertise and support throughout this project. Thank you for continuously reminding me to focus on one thing at a time and for guiding me through tough decisions with clarity and patience.

I am also deeply grateful to my second supervisor, Eduardo Figueiredo Mota Diniz Costa, for always being available to answer my questions and offer thoughtful advice. Our brainstorming sessions and insightful conversations greatly motivated me to explore further and dive deeper into the research. Your consistent encouragement, especially during challenging periods, made a lasting impact.

My thanks extend to the entire HERALD Lab. The bi-weekly meetings offered not only valuable technical knowledge but also a broader perspective on the potential of our field and cutting-edge research developments. It was an inspiring environment to grow in.

To my dear TU Delft companions, Elize and Valerio — I truly could not have made it through this journey without you. Full stop. From lunch breaks and coffee chats to our unforgettable trip to Italy, you made these past two years incredibly meaningful. Elize, thank you for being my teammate from the very beginning to the very end.

Lastly, I want to thank my family and close friends for their endless support. To my sister, who despite the distance, was always just a night call away when I needed a boost. To my family, whose love means the world to me. And to my amazing friends — Nastya (best roommate ever), Daria, Yehor, Francisco, and Claudia — thank you for making my life in the Randstad truly unforgettable.

Delft, University of Technology
July 15, 2025

Veronika Tajgler

“Any measurement that you make without the knowledge of its uncertainty is completely meaningless.”

— *Walter Lewin, MIT*

“As far as the laws of mathematics refer to reality, they are not certain; and as far as they are certain, they do not refer to reality.”

— *Albert Einstein*

Chapter 1

Introduction

1-1 Motivation

Modern systems are inherently nonlinear and are often subject to uncertainty [55]. Yet, the interest in automating systems in safety-critical domains such as smart manufacturing [52], medical devices/robotics [22], air-traffic management [32], and aerospace [39] is increasing. As a result, safety-critical control and planning for systems affected by uncertainty becomes one of the fundamental problems in automation.

In particular, as one of the most challenging control technologies, autonomous driving has advanced from controlled research settings to real-world deployment. From self-driving taxis to highway assist systems, these technologies promise to reduce road accidents—94% of which are caused by human error [63]. However, the integration of autonomous systems into the real world depends on verifiable safety. This is especially evident in traffic scenarios, where autonomous vehicles must avoid collisions under uncertain and/or dynamic conditions. For example, Tesla's Autopilot system — despite incorporating state-of-the-art perception — was involved in 736 crashes and 17 fatalities between 2019 and 2023 due to failures in obstacle detection [49]. These incidents underscore a need for safety guarantees. Yet, as of today, there are no commercially available SAE Level 4 autonomous vehicles [77].

The gap between academic innovation and real-world implementation for safety-critical applications persists unless uncertainty is explicitly accounted for in the controller design and a posteriori verification of correctness is provided. Hence, in this thesis, we focus on building a controller methodology for a stochastic, possibly nonlinear system and providing safety certification.

1-2 Challenges

Real-world systems often suffer from imperfect models and are affected by both process noise and sensor noise [59]. As such, control design must account for two types of uncertainty:

aleatoric (inherent measurement noise) and epistemic (model or environment). Furthermore, despite progress in advanced control techniques, safety certification remains an open problem [10]. The lack of formal safety guarantees and a systematic way to quantify the risk is a limiting factor between the research and real-world deployment [18].

To this end, Stochastic Model Predictive Control (SMPC) is a powerful framework that, unlike its deterministic counterpart, integrates the uncertainty model directly into the optimisation process. This enables the controller to predict future distributions over the system states and adapt its policy accordingly. However, when dealing with nonlinear stochastic systems, two major challenges arise:

- First, accurately propagating uncertainty over the prediction horizon becomes nontrivial [48]. Nonlinear dynamics distort even simple initial distributions like Gaussian, making exact inference generally intractable. As a result, we must rely on approximations — such as sampling-based particle methods [83], unscented transform [42], or polynomial chaos expansions [25] - all of which introduce trade-offs between computational complexity and approximation error.
- Second, formulating and enforcing probabilistic (chance) constraints in a tractable way is difficult [16]. These constraints require evaluating whether the system violates safety conditions, often over high-dimensional uncertainty distributions. Ensuring that such constraints are computationally feasible and statistically meaningful remains an open problem, especially when formal guarantees are needed.

1-3 Contribution and Outline

This work proposes a quantisation-based SMPC framework for discrete-time nonlinear systems with Gaussian additive noise, incorporating individual chance constraints. To manage uncertainty propagation over the prediction horizon, we discretise both state and disturbance distributions with formal guarantees in the Wasserstein distance. Furthermore, we propose a compression method at each step for computational traceability during the multi-step prediction. Obstacle avoidance is handled via a cost penalty. For safety certification, we introduce a validation scheme based on Wasserstein ambiguity sets that estimate worst-case constraint violation probabilities. The proposed approach is evaluated in simulation on nonlinear benchmark tasks under both open- and closed-loop policies.

The structure of this thesis is as follows. Chapter 2 reviews related work, providing background on SMPC, with a particular focus on existing methods for uncertainty propagation and the formulation of chance constraints. In Chapter 3, we present a formal mathematical statement of the control problem under consideration. Chapter 4 introduces the proposed control framework and formal safety certification. Chapter 5 presents empirical results demonstrating the effectiveness of the methodology on nonlinear benchmark systems. Finally, Chapter 6 concludes the thesis and provides a discussion of the findings and directions for future work.

Chapter 2

Background

This Chapter introduces the necessary preliminaries for the remainder of the thesis. Specifically, we introduce the notion of Model Predictive Control [70, 36, 44], giving emphasis to the case where the model dynamics is not linear, while also discussing the problem of uncertainty propagation [26] and chance-constrained optimisation [37].

One assumption that will be used throughout this chapter is that all states can be measured directly. This is often not the case, but as this can be accounted for using state estimation, the general theory in this chapter will apply.

2-1 Model Predictive Control

In this section, we introduce the notion of Stochastic Model Predictive Control by building up from the deterministic case. The general motivation behind Model Predictive Control (MPC) is the systematic way constraints are handled. Using knowledge of the system dynamics with MPC, we can handle long dead time in feedback measurements, or even the lack of measurements using an open-loop MPC.

2-1-1 Nominal Model Predictive Control

Model Predictive Control (MPC) is a model-based control methodology that systematically uses predictions to compute optimal control decisions in real time [16, 59, 70, 72]. This subsection investigates the MPC design for the control of constrained nonlinear time-invariant systems, described by the following difference equation:

$$x_{k+1} = f(x_k, u_k), \tag{2-1}$$

where $x_k \in \mathcal{X}$ and $u_k \in \mathcal{U}$ are the system state and control input respectively. The state space $\mathcal{X}$ and the control input space $\mathcal{U}$ are arbitrary metric spaces, that is, sets in which the distances between two elements $x, y \in \mathcal{X}$ or $u, v \in \mathcal{U}$ are measured by metrics $d_{\mathcal{X}}(x, y)$

or $d_{\mathcal{U}}(u,v)$, respectively [36]. Within the scope of this research, sets $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{U} \subseteq \mathbb{R}^m$ are assumed to be closed, while the $\mathbb{R}^n$ set denotes the n-dimensional Euclidean space. The variable x_{k+1} is the successor state. The function $f(\cdot)$ is a possibly nonlinear Borel-measurable function that characterises the system dynamics and maps $f : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$. The set of finite control sequences $u(0), \dots, u(N-1)$ for $i \in N$, will be denoted as $\mathcal{U}^N$.

The cost function $J(\cdot)$ to be used in optimisation should penalise the distance of an arbitrary state $x \in \mathcal{X}$ to the desired reference $x_\star \in \mathcal{X}$ [70]. As the state approaches the equilibrium point $x_\star$, the cost must decrease, so intuitively the only condition for the cost definition is that it is zero at the equilibrium and positive elsewhere. In practice, the control input $u \in \mathcal{U}$ is often also penalised, either for computational or due to modelling reasons [36].

Furthermore, as we compute the prediction over the finite horizon N, the cost accumulates and is thus a function of the control sequence $\mathbf{u}_k := u_{0|k}, \dots, u_{N-1|k}$, and state sequence $\mathbf{x}_k := x_{0|k}, \dots, x_{N|k}$. The final formulation can be found in Definition 1.

Definition 1 (Cumulative cost function)**.** *The cumulative cost over the prediction horizon N is defined as a function of control sequence $\mathbf{u}_k$ and state sequences $\mathbf{x}_k$ as:*

$$J(\mathbf{x}_k, \mathbf{u}_k) = \sum_{i=0}^{N-1} \ell(x_{i|k}, u_{i|k}) + V_f(x_{N|k}), \tag{2-2}$$

where $x_{i|k}$ and $u_{i|k}$ are the predicted state and the predicted control input at time $k+i$ based on information available at time k. The function $\ell(x,u)$ is the stage cost that penalises deviations from the desired state and control effort, while $V_f(x_{N|k})$ is a terminal cost that represents the cost-to-go beyond the prediction horizon.

Note that in most cases, the state sequence can be expressed in terms of the initial state value $x_{0|k}$ by the difference equation in Eq. (2-1). Furthermore, the role of $V_f(x_{N|k})$ is critical in MPC design and is investigated in further chapters.

The main underlying idea of MPC is to compute a sequence of admissible control inputs by solving a finite-horizon Optimal Control Problem (OCP) at each time step [36]. OCP minimises the objective function with respect to the control sequence $\mathbf{u}_k$, subject to system dynamics and constraints, as defined in Eq. (2-3). While the system dynamics and cost are optimised over the prediction horizon $i = 0, \dots, N-1$, only the first input of the resulting optimal sequence is applied to the system. Naturally, this process is repeated at each sampling instant, making it a receding horizon control strategy [59].

$$\begin{aligned}
\min_{u_{0|k},\dots,u_{N-1|k}} \quad & \sum_{i=0}^{N-1} \ell(x_{i|k}, u_{i|k}) + V_f(x_{N|k}) && && \text{(2-3a)}\\
\text{subject to} \quad & x_{i+1|k} = f(x_{i|k}, u_{i|k}), && \forall i = 0, \dots, N-1, && \text{(2-3b)}\\
& x_{i|k} \in \mathcal{X}, \quad u_{i|k} \in \mathcal{U}, && \forall i = 0, \dots, N-1, && \text{(2-3c)}\\
& x_N \in \mathcal{X}_f && && \text{(2-3d)}\\
& x_{0|k} = x_k, && && \text{(2-3e)}
\end{aligned}$$

2-1-2 Stochastic MPC

In the previous section, we assumed full knowledge of the state of the system x. However, in practical scenarios, the state is often subject to uncertainty arising from process noise (e.g., wind), measurement noise, or model mismatch during state estimation or learning [59]. Although MPC offers a certain degree of robustness due to its receding-horizon nature, its deterministic formulation is generally inadequate for systematically addressing such uncertainties [59].

To this end, various extensions of MPC have been developed to handle uncertainty explicitly [70]. In the stochastic setting, the uncertainty is typically modelled as a random variable $\omega_k \in \mathcal{W} \subseteq \mathbb{R}^q$, representing process disturbances or model uncertainty, with known probability distribution $\omega_k \sim \mathbb{P}_\omega \in \mathcal{P}(\mathcal{W})$, where $\mathcal{P}$ denote the probability space. It can also be assumed that the initial state is uncertain and is modelled as a random variable $x_0 \sim \mathbb{P}_{x_0} \in \mathcal{P}(\mathcal{X})$. Accordingly, the discrete-time system model in Eq. (2-1) is extended to:

$$x_{k+1} = f(x_k, u_k, \omega_k), \tag{2-4}$$

where $f(\cdot)$ defines the dynamics of the (nonlinear) stochastic system.

While deterministic MPC typically employs an open-loop formulation and optimises the control sequence $\mathbf{u_k}$ directly, stochastic formulations often require closed-loop policies, as they attempt to reduce the effect of disturbances [4, 59, 72]. As the uncertainty accumulates over the prediction horizon, the control policy must account for that, so the open-loop schemes are seen as conservative [4]. Hence, when uncertainty is present and the state is known or observations of the state are available, feedback control is superior to open-loop control [71].

Unfortunately, optimising over the policy function space is generally intractable. As a result, the policy is often parametrised as:

$$u_k = \pi_k(x_k, \theta_k), \tag{2-5}$$

with θ being the parameter vector. Parametrised feedback policies allow the controller to anticipate and react to uncertainties while keeping the resulting optimisation problem computationally tractable [59].

Different choices of the parameter vector θ lead to various Stochastic Model Predictive Control (SMPC) formulations [16]. For example, choosing $\theta_k := [b_k, vec(K)]$ corresponds to the affine state-feedback policy in (2-6), which is widely used due to its balance between performance and computational efficiency [14, 33, 59, 72]. Another common way to parametrise the feedback policy is affine disturbance-feedback from Eq. (2-7) [5, 89].

$$\pi_k(x_k, \theta_k) = b_k + K x_k, \qquad \text{affine state-feedback policy,} \tag{2-6}$$

$$\pi_k(x_k, \theta_k) = b_k + \sum_{j=0}^{k-1} \Omega_j w_j, \qquad \text{affine disturbance-feedback policy,} \tag{2-7}$$

where $b_k \in \mathbb{R}^m$, potentially time-dependent [59] $K \in \mathbb{R}^{m \times d}$, and $\Omega_j \in \mathbb{R}^{m \times q}$ are policy parameters [25].

Furthermore, unlike in the nominal MPC, in the stochastic case, the cost function is commonly formulated as the expected value cost. In this case, under the sequence of control laws

$\boldsymbol{\pi} : \pi_{0|k}(\cdot), \dots, \pi_{N-1|k}(\cdot)$, the cost can be defined as the expectation of the cumulative stage cost and the terminal cost:

$$J_N^{\boldsymbol{\pi}}(x) = \mathbb{E}_{|x}\left(c_N(x, \boldsymbol{\pi}, \omega)\right) = \mathbb{E}_{|x}\left[\sum_{i=0}^{N-1} l(x_{i|k}, \mathbf{u}_{i|k}, \mathbf{w}_{i|k}) + V_f(x_{N|k})\right], \tag{2-8}$$

where $\mathbb{E}_{|x}(\cdot) := \mathbb{E}(\cdot \mid x_0 = x)$ denotes the conditional expectation given the initial state with respect to the probability measure $\mathbb{P}$ defined on the underlying probability space [1].

Lastly, the minimisation of the cost function in Eq. (2-8) is performed subject to constraints on system states and control inputs. In the presence of uncertainty, the constraints must be reformulated probabilistically:

$$\mathbb{P}_{|x}\left[g_j(x_{i|k}) \leq 0, \quad \forall j = 1, \dots, s\right] \geq 1 - \varepsilon, \qquad \forall i = 1, \dots, N \tag{2-9}$$

where $g_j : \mathbb{R}^d \to \mathbb{R}$ is (nonlinear) Borel-measurable function, s is a total number of inequality constraints, and $\varepsilon \in (0, 1)$ specifies the allowable probability of constraint violation [59]. Conditional probability $\mathbb{P}_{|x}$ denotes the probability of a future event i conditioned on the current state at the time step k, that is, $x_{0|k} = x_k$.

An alternative formulation is the joint chance constraint, imposed through the entire prediction horizon N, as originally proposed by [75]. Ono [66] states that in the presence of unbounded stochastic disturbances, defining chance constraints jointly over the horizon is critical to ensure resolvability, also known as recursive feasibility, which is essential for maintaining stability in robust and stochastic model predictive control. The joint chance constraint takes the form:

$$\mathbb{P}_{|x}\left[g_j(x_{i|k}) \leq 0, \quad \forall j = 1, \dots, s \quad \forall i = 1, \dots, N\right] \geq 1 - \varepsilon, \tag{2-10}$$

Input constraints are commonly treated as hard constraints, as they represent the electrical/-physical limits of the system [70].

$$u_{i|k} \in \mathcal{U} \tag{2-11}$$

Nevertheless, the chance-constrained formulation in Eq. (2-9) is non-convex and computationally intractable [6, 54]. There are two primary reasons for this intractability. First, the feasible region defined by a probabilistic constraint $\mathcal{W}(x_k, u_k) : \{\omega_k \in \mathbb{R}^q | g(x_k(\omega_k), u_k) \leq 0\}$ is generally not convex [54]. Second, evaluating the probability $\mathbb{P}\left[g(x, \omega) \leq 0\right]$ for a given $x \in \mathcal{X}$ requires computing a possibly multidimensional integral:

$$\int_{\mathcal{W}(x_k, u_k)} p(\omega)\, d\omega, \tag{2-12}$$

where ω denotes the random disturbance and $p(\omega)$ is its probability density function. When $g(\cdot)$ is nonlinear or the distribution of ω is non-Gaussian, the integration domain becomes complex and does not admit a closed-form solution. As demonstrated in [54, 64], even in the case of linear systems with Gaussian noise, computing this integral is nontrivial, which complicates both constraint evaluation and optimisation and, as a result, motivates the use of tractable approximations further elaborated in Section 2-3.

[1] Technical discussions on choice of cost function can be found in Appendix A-1

Hence, the objective of the SMPC problem is to compute the sequence policy $\pi(\cdot)$ that minimises the expected cumulative cost over a finite prediction horizon N, as formulated in the stochastic optimal control problem below:

$$\begin{aligned}
\min_{\pi_{0|k}(\cdot),\ldots,\pi_{N-1|k}(\cdot)} \quad & \mathbb{E}_{x_k \sim \mathbb{P}_{x_k}} \sum_{k=0}^{N} \Big[\ell(x_{i|k}, u_{i|k}) + V_f(x_{N|k})\Big] && && \text{(2-13a)}\\
\text{subject to} \quad & x_{i+1|k} = f(x_{i|k}, u_{i|k}, \omega_{i|k}), && \forall i = 0, \ldots, N-1, && \text{(2-13b)}\\
& \mathbb{P}\Big[g(x_{i|k}, u_{i|k}) \leq 0\Big] \geq 1 - \varepsilon, && \forall i = 0, \ldots, N-1, && \text{(2-13c)}\\
& u_{i|k} = \pi_{i|k}(x_{i|k}, \theta), && \forall i = 0, \ldots, N-1, && \text{(2-13d)}\\
& x_{i|k} \in \mathcal{X}, \quad u_{i|k} \in \mathcal{U} && \forall i = 0, \ldots, N-1, && \text{(2-13e)}\\
& \omega_{i|k} \sim \mathbb{P}_{\omega}, && \forall i = 0, \ldots, N-1, && \text{(2-13f)}\\
& x_{0|k} \sim \mathbb{P}_{x_k}, && && \text{(2-13g)}
\end{aligned}$$

where $x_{i|k}$ and $u_{i|k}$ denote the predicted state and control input at time step i given information at time k, and $\mathbb{P}_{x_{i|k}}$ represents the corresponding state distribution given the initial state $x_{0|k}$. The probabilistic constraint in Eq. (2-13c) ensures that the constraint function $g(x_{i|k}, u_{i|k})$ is satisfied with high confidence, where $\varepsilon \in (0, 1)$ defines the acceptable violation probability.

2-1-3 Challenges in Nonlinear Stochastic MPC

Although the SMPC formulation provides a natural framework for uncertainty-aware control, it introduces several design and computational challenges. These include:

- **Policy parametrisation.** The control law $\pi(\cdot)$ can be parametrised in various ways, leading to different SMPC formulations with varying tractability and performance [9, 16, 72].

- **Uncertainty propagation through dynamics.** Propagating uncertainty through nonlinear system dynamics is especially complex, and often computationally inefficient. This is a central bottleneck in nonlinear SMPC (NSMPC) design [16]. Many approaches attempt to construct surrogate models or use moment matching, but these approximations may introduce bias or inaccuracy.

- **Probabilistic (chance) constraint formulation.** Incorporating chance constraints into the OCP often makes the problem intractable [16]. Reformulating these constraints into tractable approximations—such as convex relaxations or sample-based methods—remains a central challenge. In particular, random sampling techniques, commonly used in stochastic programming, offer a practical way to handle the soft constraints in SMPC by approximating probability distributions of states and parameters with finite discrete samples [16].

- **Computational complexity due to uncertainty modelling.** Related to the previous one - SMPC typically requires more computation than robust MPC, primarily because it must compute and propagate probability distributions of future states [45].

Random sampling techniques (e.g., Monte Carlo) are often used to approximate these distributions using finite sets of discrete samples, which introduces additional computational overhead as we need a lot of samples to get a good approximation

- **Nonlinearity and nonconvexity.** The presence of nonlinear dynamics typically results in a nonconvex OCP, where global optimality cannot be guaranteed. As a result, only sub-optimal solutions are generally available, which may limit guarantees related to closed-loop stability.
- **Theoretical limitations for nonlinear SMPC.** Establishing closed-loop theoretical properties such as recursive feasibility, constraint satisfaction, and stability for nonlinear SMPC schemes remains an open and challenging area of research [16].

2-2 Uncertainty Propagation for Model Predictive Control

Accurate modelling is essential for designing effective model predictive controllers, particularly for complex systems that require safety guarantees. To account for uncertainty, the evolution of stochastic dynamics is often described by the Chapman–Kolmogorov equation [48]. While probabilistic predictions are straightforward for linear systems with Gaussian distributions, nonlinear dynamics and non-Gaussian probabilistic density functions (PDF) significantly increase the complexity, making the Chapman–Kolmogorov equation intractable. As shown in Figure 2-1, after one step nonlinear transformation, the distribution is no longer Gaussian [48]. As a result, research has focused on approximation methods for uncertainty propagation that aim to balance efficiency and accuracy.

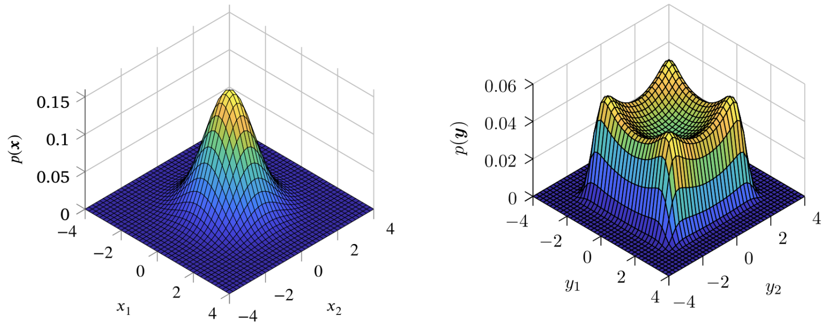


Figure 2-1: The PDF of a Gaussian random variable x and the PDF of $y = f(x) = \begin{bmatrix} 2\tanh(x_1)+0.2x_2 \\ 2\tanh(x_2)+0.2x_1 \end{bmatrix}$ computed by Monte Carlo simulation [48].

Consider a discrete-time stochastic process governed by the following difference equation [27]:

$$x_{k+1} = f(x_k, \omega_k), \quad x_0 \sim \mathbb{P}_{x_0}, \quad \omega_k \sim \mathbb{P}_\omega, \tag{2-14}$$

where $f : \mathbb{R}^d \times \mathbb{R}^q \to \mathbb{R}^d$ represents the one-step system dynamics, x_0 is a random initial state with distribution $\mathbb{P}_{x_0} \in \mathcal{P}(\mathbb{R}^d)$, and ω_k is a stochastic disturbance sampled from $\mathbb{P}_\omega \in \mathcal{P}(\mathbb{R}^q)$.

The probability distribution of the state at the next time step, denoted $\mathbb{P}_{x_{k+1}}$, can be described by its density $p_{x_{k+1}}$, which evolves according to the Chapman–Kolmogorov equation [76]:

$$p_{x_{k+1}}(x_{k+1}) = \int_{\mathbb{R}^d} p(x_{k+1} \mid x_k)\, p_{x_k}(x_k)\, dx_k. \tag{2-15}$$

This equation expresses the density $p_{x_{k+1}}$ as the marginal of the one-step transition density $p(x_{k+1} \mid x_k)$ with respect to the current state distribution p_{x_k} [27].

However, solving Eq. (2-15) is generally intractable for nonlinear systems, as the transition kernel is not available in closed form. Therefore, practical implementations rely on approximate uncertainty propagation methods to evolve the state distribution over time in a computationally feasible way. One straightforward method is to linearise nonlinear systems and take advantage of the extensive preliminary work on SMPC for linear systems [48]. However, the linearisation error increases with the degree of nonlinearity, and the control performance is expected to significantly degrade for strongly nonlinear systems [48]. Hence, in this research, we focus on methods that do not require linearisation of the system dynamics.

2-2-1 Sample-based approximations

While deterministic or moment-based uncertainty propagation techniques often rely on simplifying assumptions (e.g., Gaussianity or linearity), Monte Carlo (MC)-based sample approximation methods provide a flexible and expressive alternative. These approaches approximate the evolution of the state distribution through direct simulation of trajectories under randomly sampled disturbances [83].

Monte Carlo Simulation

MC simulation is a widely used method for propagating uncertainty in stochastic MPC, especially for nonlinear and non-Gaussian systems. Instead of analytically computing the propagated distribution $\mathbb{P}_{x_k}$ for stochastic dynamics be governed by Eq. (2-14), we draw M independent samples $\{\omega_k^{(m)}\}_{m=1}^M$ and simulate the state evolution as:

$$x_{k+1}^{(m)} = f(x_k^{(m)}, \omega_k^{(m)}), \quad m = 1, \ldots, M. \tag{2-16}$$

The expected cost is approximated using:

$$\min_{u_0,\ldots,u_{N-1}} \frac{1}{M} \sum_{m=1}^{M} \left(\sum_{k=0}^{N-1} \ell(x_k^{(m)}, u_k) + V_f(x_N^{(m)}) \right). \tag{2-17}$$

Chance constraints such as

$$\mathbb{P}(h(x_k) \leq 0) \geq 1 - \alpha \tag{2-18}$$

are enforced by requiring at least $(1-\alpha)M$ out of M simulated trajectories to satisfy $h(x_k^{(m)}) \leq 0$. While flexible and distribution-free, MC simulation can be computationally expensive, particularly in high dimensions. Efficiency can be improved using techniques such as variance reduction or scenario reduction [43, 83].

Unscented Transform Approximation

The Unscented Transform (UT) approximates the transformation of the random variable by deterministically selecting a set of sigma points that capture the true mean and covariance of the distribution [41, 42, 82]. These sigma points are then propagated through the nonlinear function directly, and their transformed statistics are used to reconstruct the output distribution [41, 42, 82].

Let $x \in \mathbb{R}^d$ be a random variable with mean $\mu_x \in \mathbb{R}^d$ and covariance $\Sigma_x \in \mathbb{R}^{d\times d}$. The UT generates $2d+1$ sigma points $x^{(i)}$ as:

$$x^{(0)} = \mu_x, \tag{2-19}$$

$$x^{(i)} = \mu_x + \left(\sqrt{\alpha^2(d+\kappa)\Sigma_x}\right)_i, \quad i = 1, \ldots, d, \tag{2-20}$$

$$x^{(d+i)} = \mu_x - \left(\sqrt{\alpha^2(d+\kappa)\Sigma_x}\right)_i, \quad i = 1, \ldots, d, \tag{2-21}$$

where $(\cdot)_i$ denotes the i-th column of the matrix square root (typically Cholesky decomposition), and α, κ are tuning parameters that determine the spread of the sigma points [48, 65].

Each sigma point is propagated through the nonlinear function $f(\cdot)$:

$$\hat{x}^{(i)} = f(x^{(i)}), \quad i = 0, \dots, 2d. \tag{2-22}$$

The weights for computing the mean and covariance are given by:

$$w_\mu^{(0)} = 1 - \frac{d}{\alpha^2(d+\kappa)}, \qquad w_\Sigma^{(0)} = w_\mu^{(0)} + (1 - \alpha^2 + \beta), \tag{2-23}$$

$$w_\mu^{(i)} = w_\Sigma^{(i)} = \frac{1}{2\alpha^2(d+\kappa)}, \quad i = 1, \dots, 2d, \tag{2-24}$$

where β is an additional parameter that incorporates prior knowledge of the distribution (e.g., $\beta = 2$ is optimal for Gaussians [48, 82]).

Finally, the predicted mean and covariance of the transformed variable are estimated as:

$$\mu_{\hat{x}} \approx \sum_{i=0}^{2d} w_\mu^{(i)} \hat{x}^{(i)}, \tag{2-25}$$

$$\Sigma_{\hat{x}} \approx \sum_{i=0}^{2d} w_\Sigma^{(i)} \left(\hat{x}^{(i)} - \mu_{\hat{x}}\right) \left(\hat{x}^{(i)} - \mu_{\hat{x}}\right)^\top. \tag{2-26}$$

The UT is an attractive alternative to sampling-based approaches like Monte Carlo, offering a favorable trade-off between computational efficiency and accuracy in uncertainty propagation, as shown by [12]. Liu et al. [50] applied UT MPC for the lane keeping assistance control of a semi-autonomous vehicle.

2-2-2 Polynomial Chaos Expansion

Polynomial Chaos Expansion (PCE) is a powerful modelling technique for uncertainty propagation. It approximates the evolution of random variables by expanding them into orthogonal polynomial basis functions that are tailored to the distribution of the uncertainties [25, 48, 60, 61, 78, 90].

Let the predicted next state $x_{k+1} = f(x_k, \omega_k)$ be approximated by a polynomial expansion:

$$\hat{x}_{k+1} = \sum_{i=0}^{P} a_i \phi_i(x_k, \omega_k), \tag{2-27}$$

where $\phi_i(\cdot)$ are multivariate orthogonal polynomials and $a_i \in \mathbb{R}$ are the associated coefficients. These basis functions satisfy the orthogonality condition with respect to the joint distribution of (x_k, ω_k):

$$\mathbb{E}[\phi_i(x_k, \omega_k)\phi_j(x_k, \omega_k)] = \langle \phi_i, \phi_j \rangle = \langle \phi_i^2 \rangle \delta_{ij}. \tag{2-28}$$

The coefficients a_i can be determined using various computational techniques:

- regularized regression [25],

- probabilistic collocation [61],
- spectral projection via Gauss–Hermite quadrature [13].

PCE enables efficient computation of the statistical moments (e.g., mean and variance) of the predicted state from the expansion coefficients. It offers a sample-free alternative to Monte Carlo methods and is especially effective when uncertainties can be captured by a small number of polynomial terms. However, the number of expansion terms grows combinatorially with the input dimension and polynomial order, which may limit scalability. The strength of PCE lies in its ability to capture higher-order stochastic moments of the probability density function (PDF), thereby improving the accuracy of uncertainty propagation [48].

2-2-3 Gaussian Mixture Model Approximation

Gaussian Mixture Models (GMMs) offer an approximation of arbitrary probability distributions by representing them as a weighted sum of Gaussian components [16, 48]. A probability density function $p(x_k)$ can be approximated by a Gaussian mixture as:

$$p(x_k) \approx \sum_{i=1}^{M} w^{(i)} \mathcal{N}\left(\mu_{x_k}^{(i)}, \Sigma_{x_k}^{(i)}\right), \tag{2-29}$$

where conditions on weight are $w^{(i)} \geq 0$ and $\sum_{i=1}^{M} w^{(i)} = 1$, with each component defined using $\mathcal{N}(\cdot, \cdot)$ as notation for the Gaussian probability density function with mean $\mu_{x_k}^{(i)}$ and covariance $\Sigma_{x_k}^{(i)}$ [48].

After applying a nonlinear transformation $x_{k+1} = f(x_k, \omega_k)$, the output distribution $p(x_{k+1})$ can also be approximated using another mixture of Gaussians:

$$p(x_{k+1}) \approx \sum_{i=1}^{M} \tilde{w}^{(i)} \mathcal{N}\left(\mu_{x_{k+1}}^{(i)}, \Sigma_{x_{k+1}}^{(i)}\right), \tag{2-30}$$

where the updated weights $\tilde{w}^{(i)}$, means $\mu_{x_{k+1}}^{(i)}$, and covariances $\Sigma_{x_{k+1}}^{(i)}$ are computed by pushing each component through the dynamics. Among common technique for that are sampling [27], moment matching with UT [79].

Despite their expressive power, GMM-based approaches face computational challenges, particularly in component explosion and numerical stability during propagation. To address this, [85] propose the use of hybrid mixtures combining Dirac and Gaussian components, allowing efficient offline computation of the value function via dynamic programming. Applications of GMM-based MPC have been demonstrated in areas requiring safety-critical planning under uncertainty, such as robotic obstacle avoidance and motion planning for autonomous systems [84, 85].

2-3 Dealing with Chance constraints

2-3-1 Second-order Cone Constraints

In the special case where the uncertainty ω_k is Gaussian and the constraint function $g(x, u, \omega_k)$ is affine in ω_k, the chance constraint can be exactly reformulated as a second-order cone constraint [54, 64, 72]. This formulation is convex and computationally tractable using conic solvers. However, it is limited to single linear constraints under Gaussian noise.

2-3-2 Chebyshev inequality

A generalisation of this approach is to release the assumption of ω_k being Gaussian and consider only its first two moments. In this case, if single linear chance constraints are present, Chebyshev's inequality can be invoked, and the constraints can be reformulated again as second-order cone constraints. When only the first and second moments of the uncertainty distribution are known, Chebyshev-type inequalities can be employed to derive conservative approximations. For example, if the mean and variance of $g(x, u, \omega_k)$ are known, the one-sided Chebyshev inequality yields:

$$\mathbb{P}\left[g(x, u, \omega_k) \leq \mathbb{E}[g] + \lambda\sqrt{\mathrm{Var}[g]}\right] \geq 1 - \frac{1}{\lambda^2}. \tag{2-31}$$

This allows reformulating the chance constraint into a deterministic bound involving the mean and standard deviation. While this method is highly general—it applies to any distribution with finite moments—it is also conservative and only suitable for single constraints. Another downside of this method is that it relies only on the first two moments of the distribution [15].

2-3-3 Enforce constraint for all disturbance realisations

Robust MPC enforces constraints for all admissible realisations of uncertainty within a bounded disturbance set $\mathcal{D}$ [57, 34]. In this setting, the constraint violation probability ε is zero, and the constraints can be reformulated as:

$$g(x_{i|k}, u_{i|k}) \leq 0 \quad \forall \omega_{0|k}, \ldots, \omega_{N-1|k} \in \mathcal{D}. \tag{2-32}$$

An advantage of this approach is that, by relying solely on a bounded disturbance set $\mathcal{D}$ instead of requiring full distributional knowledge, the formulation becomes simpler and more tractable. However, this comes at the cost of increased conservatism [72], as the controller must satisfy constraints under all possible disturbance realisations within $\mathcal{D}$, including potentially unlikely worst-case scenarios. On the other hand, scenario-based methods that sample the uncertainty and enforce constraint satisfaction over a finite set of realisations [74].

2-3-4 Distributionally Robust Chance Constraints

In many practical control problems, the uncertainty in initial conditions or system disturbances is only accessible through empirical data or learned models. Moreover, classical chance constraints in Equation 2-13c are not directly tractable under nonlinear dynamics or unbounded distributions. A robust alternative is to impose a Distributionally Robust Chance Constraint (DRCC) [68], which guarantees constraint satisfaction across all distributions in an ambiguity set $\mathcal{B}$. Then, the distributionally robust chance-constrained programme takes the following form [17, 47]:

$$\begin{aligned} \min_{x \in \mathcal{X}} \quad & c^\top x && \text{(2-33)} \\ \text{s.t.} \quad & \inf_{\mathbb{P} \in \mathcal{P}} \mathbb{P}\left[\xi : g(\xi, x) \leq 0\right] && \text{(2-34)} \\ & x \in \mathcal{X} && \text{(2-35)} \end{aligned}$$

where x is a (vector of) decision variable(s) from a compact polyhedron $\mathcal{X} \subset \mathbb{R}^n$ that minimizes a linear cost function and ensures that the exogenous random vector ξ falls within a decision-dependent safety set $\mathcal{S} \subset \mathbb{R}^d$ with high probability $1 - \varepsilon$ under every distribution $\mathbb{P}$ in the ambiguity set $\mathcal{B}$ [17].

Due to its desirable statistical properties, an appealing approach is to define $\mathcal{B}$ as a Wasserstein ball centered at an empirical distribution $\hat{\mathbb{P}}$ defined in Definition 3 with the Wasserstein distance defined in 2. This formulation captures model uncertainty and is particularly suitable for sample-based MPC [17].

Definition 2 (2-Wasserstein Distance)**.** *Let $\mathbb{P}$ and $\hat{\mathbb{P}}$ be two probability measures over the space $\mathcal{X}$, and let $\mathcal{P}(\mathbb{P}, \hat{\mathbb{P}})$ denote the set of all couplings (i.e., joint distributions) with marginals $\mathbb{P}$ and $\hat{\mathbb{P}}$, respectively. Then, for a chosen norm $\|\cdot\|$, the 2-Wasserstein distance between $\mathbb{P}$ and $\hat{\mathbb{P}}$ is defined as:*

$$\mathbb{W}_2(\mathbb{P}, \hat{\mathbb{P}}) := \left(\inf_{\pi \in \mathcal{P}(\mathbb{P}, \hat{\mathbb{P}})} \mathbb{E}_{(\xi, \xi') \sim \pi} \left[\|\xi - \xi'\|^2 \right] \right)^{1/2}. \tag{2-36}$$

Definition 3 (Wasserstein Ambiguity Set)**.** *Given an empirical distribution $\hat{\mathbb{P}}$, the* Wasserstein ambiguity set *of radius $\zeta > 0$ is defined as:*

$$\mathcal{B}(\zeta) := \left\{ \mathbb{P} \in \mathcal{P}(\mathcal{X}) \;\middle|\; \mathbb{W}_2(\mathbb{P}, \hat{\mathbb{P}}) \leq \zeta \right\}. \tag{2-37}$$

This set represents the family of all distributions lying within a Wasserstein distance ζ from the empirical distribution $\hat{\mathbb{P}}$ and captures plausible deviations from the observed data.

Chen et al. [17, Theorem 3] and Xie [87, Proposition 1] show that the Problem 2-33 can be reformulated as:

$$\begin{aligned} \min_{s,\, t,\, x} \quad & c^\top x && \text{(2-38)} \\ \text{s.t.} \quad & \varepsilon N t - e^\top s \geq \zeta N && \text{(2-39)} \\ & \operatorname{dist}(\hat{\xi}_i, \bar{\mathcal{S}}(x)) \geq t - s_i \quad \forall i \in [N] && \text{(2-40)} \\ & s \geq 0, \quad x \in \mathcal{X} && \text{(2-41)} \end{aligned}$$

This reformulation follows the dual representation of the worst-case probability over a Wasserstein ambiguity set of radius ζ. The variable t serves as a threshold representing the minimal safe distance to the unsafe set $\bar{\mathcal{S}}(x)$, while the slack variables $s_i \geq 0$ capture individual violations for each empirical sample $\hat{\xi}_i$.

Ji and Lejeune [40] showed that if the stochastic function $g(x, \xi)$ is quasi-convex in ξ and satisfies a Lipschitz continuity condition, the DRCC problem can be rewritten in a form that is tractable. Furthermore, under the worst-case distribution in the Wasserstein ball, DRCC constraint is equivalent to a constraint on a Conditional Value-at-Risk (CVaR), namely:

$$\inf_{\mathbb{P}\in\mathcal{B}} \mathbb{P}(g(x,\xi) \leq 0) \geq 1-\varepsilon \quad \Longleftrightarrow \quad \sup_{\mathbb{P}\in\mathcal{B}} \mathrm{CVaR}_{\varepsilon}^{\mathbb{P}}(g(x,\xi)) \leq 0 \tag{2-42}$$

Note: So instead of requiring the probability of violating constraints to be small, we are bounding the expected magnitude of violation.

Definition 4. *The* Conditional Value-at-Risk *(CVaR) of a random variable ξ at confidence level $1-\varepsilon$ is defined as*

$$CVaR_{\varepsilon}(\xi) := \inf_{t\in\mathbb{R}} \left\{ t + \frac{1}{\varepsilon}\mathbb{E}\left[(\xi - t)_{+}\right] \right\},$$

where $(\xi - t)_{+} := \max\{\xi - t, 0\}$ denotes the excess loss above threshold t.

- also Xie and Ahmed show that it is possible to reformulate as a mixed integer convex program under certain conditions [88]. Furthermore, Xie et al. [87] show that a Distributionally Robust (DR) chance constraint with Wasserstein ambiguity is equivalent to a Conditional Value-at-Risk (CVaR) constraint. This is especially useful in risk-sensitive applications, where one prefers soft violations to hard constraints:
- An alternative practical approach is to incorporate soft penalties directly into the cost function. For example:

  $$J = \sum_{i=0}^{N-1} \mathbb{E}_{x_i\sim\hat{\mathbb{P}}_N}[\ell(x_i,u_i)] + \lambda \cdot \mathbb{E}_{x_i}\left[\max\{g(x_i,u_i),0\}^2\right], \tag{2-43}$$

  where λ is a penalty weight. This avoids hard constraints but encourages safety probabilistically.

These reformulations enable safe control design in the presence of distributional uncertainty, particularly useful in learning-based control and autonomous systems with limited or noisy data. They ensure that safety constraints remain valid even when the underlying data is imprecise or the real-world behaviour deviates slightly from model assumptions [2].

2-3-5 Theoretical Convergence Guarantee

As shown in [48, 83], under mild regularity assumptions, the sample-based approximation converges to the true stochastic control problem as $M \to \infty$, $\varepsilon, \zeta \to 0$:

$$\lim_{M\to\infty,\varepsilon,\zeta\to 0} \tilde{V}_{\tilde{\alpha},\tilde{\varepsilon}}^{\varepsilon,\zeta}(x_0, \mathcal{D}_M) = V_{\tilde{\alpha}}(x_0). \tag{2-44}$$

This guarantees that the smooth sample-based formulation remains a valid approximation of the original stochastic optimal control problem. Sample-based MPC thus provides a tractable and highly expressive framework for nonlinear, non-Gaussian systems, enabling direct optimisation over trajectories while maintaining rigorous safety guarantees [74].

Chapter 3

Problem Formulation

Consider the following discrete-time stochastic process described by:

$$x_{k+1} = f(x_k, u_k, \omega_k), \quad x_0 \sim \mathbb{P}_{x_0}, \omega_k \sim \mathbb{P}_\omega, \tag{3-1}$$

where $x_k \in \mathcal{X} \subseteq \mathbb{R}^d$ represents the state of the system, $u_k \in \mathcal{U} \subseteq \mathbb{R}^m$ a control input, $\omega_k \in \mathcal{W} \subseteq \mathbb{R}^q$ a process noise, and $f : \mathcal{X} \times \mathcal{U} \times \mathcal{W} \to \mathcal{X}$ is a possibly non-linear measurable function representing the (time-invariant) one-step dynamics of the System (3-1). Intuitively, (3-1) represents a dynamical system with (possible) non-linearities acting on both the state, control input, and process noise, thus encompassing a large number of systems in the literature [50, 78, 83]. Further, we assume a random initial state with distribution $x_0 \sim \mathbb{P}_{x_0} \in \mathcal{P}(\mathcal{X})$, where $\mathcal{P}(\mathcal{X})$ is the probability space over the set $\mathcal{X}$, and a known process noise distribution given by $\omega_k \sim \mathbb{P}_\omega \in \mathcal{P}(\mathcal{W})$. This research aims to solve the optimal control problem 1.

Problem 1.

$$\begin{aligned}
\min_{\theta \in \Theta} \quad & J^\pi(\theta) = \sum_{k=0}^{N} \mathbb{E}_{x_{i|k} \sim \mathbb{P}_{x_{i|k}}} \left[\ell(x_{i|k}, u_{i|k}) + V_f(x_{N|k}) \right] && && (3\text{-}2) \\
s.t \quad & x_{i+1|k} = f(x_{i|k}, u_{i|k}, \omega_{i|k}), && \forall i = 0, \dots, N-1, && (3\text{-}3) \\
& u_{i|k} = \pi_{i|k}(x_{i|k}, \theta), && \forall i = 0, \dots, N-1, && (3\text{-}4) \\
& x_{i|k} \in \mathcal{S}, \quad u_{i|k} \in \mathcal{U}, \quad \omega_{i|k} \sim \mathbb{P}_\omega, && \forall i = 0, \dots, N-1, && (3\text{-}5) \\
& x_{0|k} \sim \mathbb{P}_{x_k}, && && (3\text{-}6)
\end{aligned}$$

Specifically, we design a deterministic policy $\pi : \mathcal{X} \times \Theta \to \mathcal{U}$ given by $\pi(x, \theta)$, where $\theta \in \Theta \subseteq \mathbb{R}^p$ are the parameters of the policy that minimise the expected trajectory cost in Equation 3-2 where $\ell : \mathcal{X} \to \mathbb{R}_+$ and $V_f : \mathcal{X} \to \mathbb{R}_+$ represent a stage and terminal cost functions respectively, and $N \in \mathbb{N}$ the trajectory time horizon. Note that, although not made explicit in Eq. (3-2), the random vectors x_k depend on θ via the effect of the controller. Further, following [67], to impose safety guarantees, we define a safe set $\mathcal{S} \subseteq \mathcal{X}$ (i.e. states where the system operates safely, such as lane boundaries for self-driving vehicles). The task is then to learn a set of parameters $\theta \in \Theta$ to minimise $J^{\pi}(\theta)$ while keeping the system within the safe set $\mathcal{S}$. The set of admissible control inputs is defined in $\mathcal{U}$.

The main issue with Problem 1 is that the propagation of probability distributions through non-linear dynamics is commonly intractable [48], so that closed-form expressions for the true distributions $\mathbb{P}_{x_k}$ are generally unknown. In this case, approximation schemes are required. Furthermore, since x_0 and ω follow unbounded Gaussian distributions, strictly enforcing $x_{i|k} \in \mathcal{S}$ at all time steps is infeasible. Hence, additional approximations and relaxed constraints are required. For safety-critical applications, though, it is important to guarantee that constraints are satisfied not only for the state approximations, but also for the true state distributions. Therefore, it is fundamental to track the errors introduced when approximating probability distributions.

Approach: To tackle Problem 1, Chapter 4 outlines the steps taken to obtain a tractable problem formulation. Specifically, we exploit the quantisation of probability distributions to approximate the state distribution $\hat{\mathbb{P}}_{x_0}$ and the noise distribution $\hat{\mathbb{P}}_{\omega}$ according to Theorem [26, Proposition 3.1]. This quantisation yields a finite number of samples that represent the underlying distributions, allowing us to reformulate the state constraints in a robust form. This method facilitates the computation of the approximation error using the Wasserstein distance between the true and quantised distributions. This enables a validation step to verify that the probability of the state remaining within the safe set constraints is adequately captured [17, Theorem 1].

Chapter 4

Stochastic MPC via Quantisation

In this Chapter, we propose a receding horizon control scheme exploiting the discretisation of the state probability distributions to solve the Stochastic Model Predictive Control (SMPC) problem 1. Critically, we show how the quantisation of distributions facilitates the uncertainty propagation in the system, while controlling the error introduced by the discretisation.

The chapter is structured as follows: Section 4-1 presents the quantisation-based uncertainty propagation method used in this work, as well as how to quantify approximation errors. Then, in Section 4-1-2, we explain the compression needed to enable optimisation with multi-step prediction. Section 4-1-3, covers the safety guarantees in our Model Predictive Control (MPC) program that comprise constraints as well as a posterior verification. Lastly, in Section 4-1-4, we present the reconstructed Optimal Control Problem (OCP) as a part of this research.

4-1 Uncertainty propagation with quantisation

4-1-1 Quantisation

As mentioned in Section 2-2, propagating uncertainty through nonlinear functions is a generally intractable problem [27, 48, 72], which makes the prediction step in the MPC problem infeasible. Following [1, 26], we propose to quantise the probability distributions as shown in Definition 5 to make forward propagation feasible.

Definition 5 (Quantisation of a Probability Distribution [1]). *The quantisation of a probability distribution $p \in \mathcal{P}(\mathbb{R}^d)$ with respect to points $\mathcal{C} = \{c_i\}_{i=1}^N \subset \mathbb{R}^d$ is the discrete distribution $\Delta_{\mathcal{C}} \# p = \sum_{i=1}^N \pi^{(i)} \delta_{c_i} \in \mathcal{D}_N(\mathbb{R}^d)$, where $\#$ is a pushforward operator, $\pi^{(i)} = \mathbb{P}_{x \sim p}[x \in R_i]$ with*

$$R_i = \left\{ x \in \mathbb{R}^d : \|x - c_i\| \leq \|x - c_j\|, \forall j \in \mathbb{N}_N, j \neq i \right\}. \tag{4-1}$$

More specifically, let $\{\mathcal{R}_i\}_{i=1}^{N_x}$ as defined in Equation 4-1 be the Voronoi partition of $\mathcal{X}$ with respect to the Euclidean distance and a set of locations $\mathcal{C} \subset \mathcal{X}$, define the approximation as the following quantisation:

$$\hat{\mathbb{P}}_x := \Delta_{\mathcal{C}} \# \mathbb{P}_x = \sum_{i=1}^{N_x} \rho_i \delta_{c_i} \tag{4-2}$$

With this quantisation construction, [1, 26] show that the error between the original distribution and its quantisation can be controlled in 2−Wasserstein from Definition 2 as shown in Proposition 1. This upper bound is particularly useful in practice, as it can be computed in closed form when $\mathbb{P}$ is a Gaussian distribution [1, 26].

Proposition 1 (Quantisation error [26])**.** *Let* $\mathbb{P} \in \mathcal{P}_\rho(\mathcal{X})$ *and assume a given* $\mathcal{X}$*-partition* $\mathcal{R} = \{\mathcal{R}_i\}_{i=1}^N$ *and set of locations* $\mathcal{C} = \{c_i\}_{i=1}^N$*. Then, for any* $\rho \geq 1$*,*

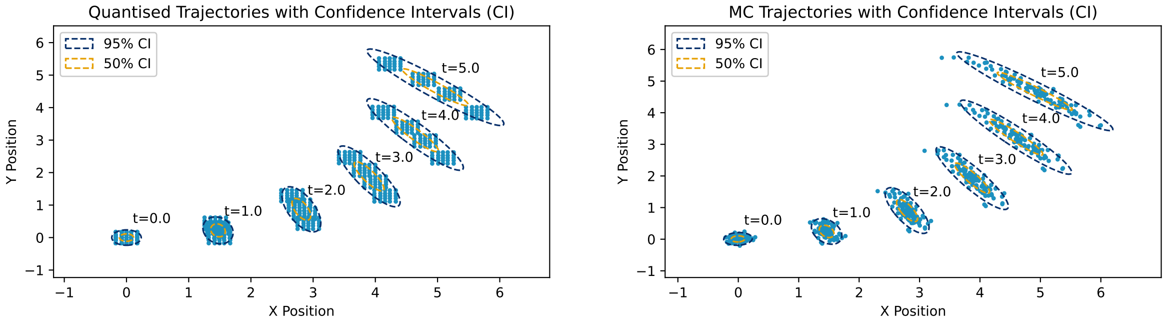
$$\mathbb{W}_\rho(\mathbb{P}, \Delta_{\mathcal{R},\mathcal{C}} \# \mathbb{P}) \leq \left(\sum_{k=1}^{N} \int_{\mathcal{R}_k} \|x - c_k\|^\rho \, d\mathbb{P}(x) \right)^{\frac{1}{\rho}} \tag{4-3}$$

Furthermore, if $\mathcal{R}$ *is chosen to be the Voronoi partition w.r.t.* $\mathcal{C}$*, then 4-3 holds with equality.*

The main advantage of handling discrete approximations is that they can be analytically push-forwarded by non-linear functions $f(x,\cdot)$, resulting in the transformed measure $f(x,\cdot)\#\hat{\mathbb{P}}_x$, where # denotes the push-forward operator. More generally, given a quantised distribution $z \sim \mathbb{P} = \sum_{i=1}^N \pi^{(i)} \delta_{c_i}$ and a (possibly) non-linear map f, it holds that $f(z)$ follows the pushforward distribution $f\#\mathbb{P}$:

$$f\#\mathbb{P} = \sum_{i=1}^{N} \pi^{(i)} \delta_{f(c_i)}$$

This is illustrated in Figure 4-1; the temporal evolution of the Dubins dynamical model [24] under a Gaussian initial condition $x_0 \sim \mathcal{N}(0, 0.01I)$ is propagated over time using Quantisation (Figure 4-1a) and Monte Carlo (Figure 4-1b).



(a) Quantised distribution propagation.

(b) Monte Carlo sampling propagation.

Figure 4-1: Comparison between the quantised method and Monte Carlo sampling for temporal evolution of the Dubins dynamics with uncertainty propagation. Speed $v = 1.5$ [m/s], constant turning angle 0.3 [rad/sec], and sampling time 0.1 s. Ellipses indicate 50% and 95% confidence regions; times are annotated in seconds.

This distance measures the minimal expected squared cost of transporting mass from $\mathbb{P}$ to $\hat{\mathbb{P}}$, where the cost of moving a unit mass from ξ to ξ' is given by $\|\xi - \xi'\|^2$.

Furthermore, using Lipschitz continuity [81], we track the error propagation over time as:

$$\mathbb{W}_2(\mathbb{P}_{k+1}\hat{\mathbb{P}}_{k+1}) = \mathbb{W}_2(f\#\mathbb{P}_k, f\#\hat{\mathbb{P}}_k) \le L_f \mathbb{W}_2(\mathbb{P}_k, \hat{\mathbb{P}}_k), \tag{4-4}$$

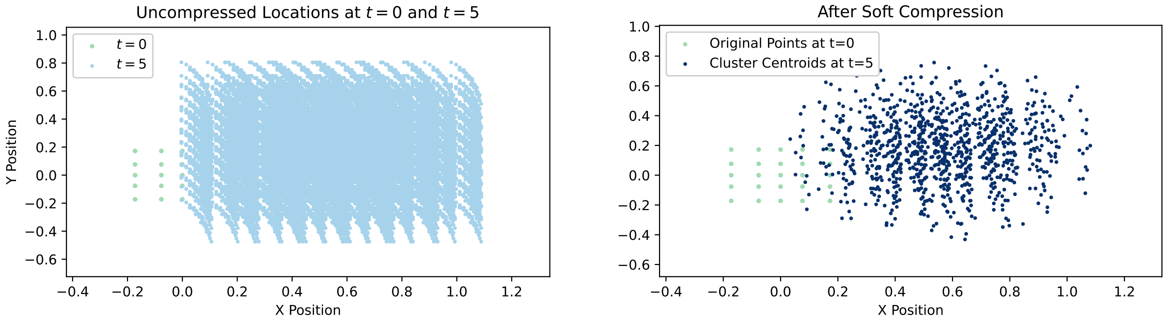
with k denoting a discrete time step. This allows us to track the probability measure $\hat{\mathbb{P}}(\mathcal{S})$ over multiple prediction steps.

4-1-2 Compression

When dealing with a system in Equation 3-1, the successor's distribution can be approximated by computing the discrete measure supported on the image of f applied to the grid points $\{(c_i, d_j)\}$ as follows:

$$\hat{\mathbb{P}}_{x_{k+1}} := f\#(\hat{\mathbb{P}}_x \times \hat{\mathbb{P}}_\omega) = \sum_{i=1}^{N_x}\sum_{j=1}^{N_\omega} \rho^{(i)}\gamma^{(j)}\delta_{f(c_i,\pi(\theta),d_j)}, \tag{4-5}$$

where $\hat{\mathbb{P}}_x$ and $\hat{\mathbb{P}}_\omega$ are the discrete measures of state and noise, respectively, with ρ and c corresponding to probability and location for the state, while γ and d are the probability and the location for the noise. The policy $\pi(\cdot, \theta)$ is either open-loop or a feedback policy parametrised by θ.



(a) Location prediction after 5 propagation steps with noise. $\hat{\mathbb{P}}_{x_{k+1}} := f\#(\hat{\mathbb{P}}_x \times \hat{\mathbb{P}}_\omega)$ as in Equation 4-5

(b) Location prediction after 5 propagation steps with noise and compression. $\hat{\mathbb{P}}_{x_{k+1}} := f_c\#(\hat{\mathbb{P}}_x \times \hat{\mathbb{P}}_\omega)$ with f_c denoting the integrated compression step.

Figure 4-2: Comparison of the naive state propagation and the on with compression on the previous Dubins dynamics example. Number of maximum partitions n_{max} is equal to the initial state discretisation, namely 100. Eight (8) partition used for noise discretisation. The propagation is done five (5) steps ahead as computational traceability is a limiting factor.

In this context, it represents the transformation of the joint distribution of discrete states and disturbances through the system dynamics f, resulting in a combinatorial explosion in the support points. For example, with 100 state samples and eight (8) noise samples, five steps yield over 3.28 million points. As shown in Figure 4-2a, the naive pushforward of discrete uncertainty leads to an exponential growth in particles, which quickly becomes both computationally expensive and representationally redundant.

Algorithm 1 Soft Compression of Weighted Samples

```
Require: Samples $c \in \mathbb{R}^{N_x \times d}$, weights $\rho \in \mathbb{R}^{N_x}$, number of clusters $n_{\text{max}}$, sharpness $\beta$
Ensure: Compressed centers $\{\tilde{c}_n\}_{n=1}^{n_{\text{max}}}$, weights $\{\tilde{\rho}_n\}_{n=1}^{n_{\text{max}}}$ forming $\tilde{\mathbb{P}} = \sum_{n=1}^{n_{\text{max}}} \tilde{\rho}_n \delta_{\tilde{c}_n}$
Initialize $\{\tilde{c}_n\}_{n=1}^{n_{\text{max}}}$ randomly
repeat
    Compute distances: $d_{in} = \|c_i - \tilde{c}_n\|_2$
    Compute soft assignments: $w_{in} = \frac{\exp(-\beta d_{in})}{\sum_{m=1}^{n_{\text{max}}} \exp(-\beta d_{im})}$
    Apply weights: $\tilde{w}_{in} = w_{in} \cdot \rho_i$
    Update cluster weights: $\tilde{\rho}_n = \sum_{i=1}^{N_x} \tilde{w}_{in}$
    Update centers: $\tilde{c}_n = \frac{1}{\tilde{\rho}_n} \sum_{i=1}^{N_x} \tilde{w}_{in} \cdot c_i$
until convergence or max iterations
    return $\{\tilde{c}_n\}_{n=1}^{n_{\text{max}}}, \{\tilde{\rho}_n\}_{n=1}^{n_{\text{max}}}$
```

Consequently, a compression is required to find a trade-off between the number of particles and the constructive information, allowing to preserve the overall geometry/structure of the distribution while reducing the number of support points. We apply a soft compression technique described in Algorithm 1, which clusters discrete measures into the fixed number of samples n_{max} as depicted in Figure 4-2b. Using this compression layer, we are able to manage complexity while retaining differentiability.

Now, the Wasserstein error between the true probability $\mathbb{P}$ and the compressed approximation $\tilde{\mathbb{P}}$ can be bounded using the triangular inequality:

$$\mathbb{W}_2(\mathbb{P}, \tilde{\mathbb{P}}) \leq \mathbb{W}_2(\mathbb{P}, \hat{\mathbb{P}}) + \mathbb{W}_2(\hat{\mathbb{P}}, \tilde{\mathbb{P}}) \tag{4-6}$$

with $\mathbb{W}_2(\mathbb{P}, \hat{\mathbb{P}})$ known, and $\mathbb{W}_2(\hat{\mathbb{P}}, \tilde{\mathbb{P}})$ computed with the Algorithm proposed by [19].

4-1-3 Safety Guarantees

Assumption 1. *In Problem 1, we assume the nonlinear programme with polytopic constraints, where each obstacle is defined as $\mathcal{O} \subset \mathbb{R}^d$. The safe set $\mathcal{S}$ as the allowable state space excluding the obstacles:*

$$\mathcal{S} = \left\{ x \in \mathbb{R}^d \,\middle|\, x_i \in [a_i, b_i] \forall i = 1, \ldots, d,\ (x_1, x_2, ..., x_d) \notin \mathcal{O} \right\}. \tag{4-7}$$

Assumption 2. *In Problem 1, the constraints on states are assumed to consist of merely individual constraints, each of which is a nonlinear function of one state variable.*

Similarly to [30, 64, 78], the chance constraints are approximated by enforcing the constraint deterministically over a finite set of sampled disturbance realisations. In our case, we robustify the chance constraints by imposing constraints on all locations $\mathcal{C}$, resulting in the following constraint:

$$c^{(i)} \in \mathcal{S} \quad \forall i = 1, \ldots, N_x. \tag{4-8}$$

Meanwhile, for the disjunctive obstacles, we exploit penalty-based relaxation in control optimisation. For this, let each obstacle be defined by a bounded region aligned by dimensions with lower and upper bounds $o_j^{\text{lo}}, o_j^{\text{hi}} \in \mathbb{R}^d$ for obstacle $j \in \{1, \dots, N_{\text{obs}}\}$, where d is the state dimension. For a given particle i, let $c^{(i)}$ be its location and $\rho^{(i)}$ its probability mass. For instance, we can have a square region $[2, 4] \times [2, 4]$ in the x_1-x_2 plane, the constraint can be expressed logically as:

$$(x_1 \leq 2 \vee x_1 \geq 4) \wedge (x_2 \leq 2 \vee x_2 \geq 4)\,, \tag{4-9}$$

which excludes the rectangular obstacle by requiring the state to lie outside it and introduces nonconvexity into the feasible region.

Hence, we define the violation of box constraints along each dimension via:

$$v^{(i,j)} = \max\left\{o_j^{\text{lo}} - c^{(i)},\, 0\right\} + \max\left\{c^{(i)} - o_j^{\text{hi}},\, 0\right\}, \tag{4-10}$$

where $v^{(i,j)} \in \mathbb{R}^d$ quantifies how far particle i lies outside obstacle j's box along each dimension. The total violation is then given as a sum of violations over the dimensions $d^{(i,j)} = \sum_{k=1}^{d} v_k^{(i,j)}$. Next, we define a soft indicator function that smoothly approximates whether the particle lies inside the obstacle:

$$\phi^{(i,j)} = \max\left\{0,\ 1 - \min\left\{d^{(i,j)}, 1\right\}\right\}, \tag{4-11}$$

which satisfies $\phi^{(i,j)} = 1$ if the particle is fully inside the obstacle, and is between 0 and 1 if only certain dimensions are inside the unsafe set. The maximum across all obstacles $\phi^{(i)} = \max_{j=1,\dots,N_{\text{obs}}} \phi^{(i,j)}$ approximates whether a particle is inside any obstacle. The expected fraction of the probability mass that lies within obstacles is then:

$$\Phi = \sum_{i=1}^{N_p} w^{(i)} \cdot \phi^{(i)}. \tag{4-12}$$

Finally, by tuning the obstacle penalty coefficient $\lambda_{\text{obs}} > 0$, we get the penalty cost:

$$J_{\text{obs}} = \lambda_{\text{obs}} \cdot \Phi, \tag{4-13}$$

Algorithm 2 Obstacle Penalty Computation

```
Require: Particles $\{c^{(i)}, \rho^{(i)}\}_{i=1}^{N_p}$, obstacles $\{o_j^{\text{lo}}, o_j^{\text{hi}}\}_{j=1}^{N_{\text{obs}}}$, penalty weight $\lambda_{\text{obs}}$
$\Phi \leftarrow 0$
for $i = 1$ to $N_p$ do
    $\phi^{(i)} \leftarrow 0$
    for $j = 1$ to $N_{\text{obs}}$ do
        $v^{(i,j)} \leftarrow \max\left\{o_j^{\text{lo}} - c^{(i)},\, 0\right\} + \max\left\{c^{(i)} - o_j^{\text{hi}},\, 0\right\}$
        $d^{(i,j)} \leftarrow \sum_{k=1}^{d} v_k^{(i,j)}$
        $\phi^{(i,j)} \leftarrow \max\left\{0,\ 1 - \min\left\{d^{(i,j)},\ 1\right\}\right\}$
        $\phi^{(i)} \leftarrow \max\left\{\phi^{(i)},\ \phi^{(i,j)}\right\}$
    end for
    $\Phi \leftarrow \Phi + \rho^{(i)} \cdot \phi^{(i)}$
end for
return $J_{\text{obs}} = \lambda_{\text{obs}} \cdot \Phi$
```

We then verify a posteriori an upper bound of the unsafe probability using the Wasserstein information. Specifically, we adopt a Wasserstein ambiguity set from Definition 3 [1] to assess the true probability $\mathbb{P}(x \in \mathcal{S})$ for the random variable x_k at each time step $k \in \{0, \dots, N\}$ to verify the level of risk associated with constraint violation:

$$\sup_{\mathbb{P} \in \mathcal{B}(\zeta)} \mathbb{P}(x \in \bar{\mathcal{S}}). \tag{4-14}$$

This represents the worst-case probability mass that could enter the unsafe set $\bar{\mathcal{S}}$ within transport budget ζ. We sort support points by distance to the unsafe set as shown in Algorithm 3, reassign mass greedily, and verify the constraint using duality (Theorem 2 in [17]):

$$\sup_{\mathbb{P} \in \mathcal{B}(\zeta)} \mathbb{P}(x_k \in \bar{\mathcal{S}}) \leq \varepsilon \iff \min \left(\sum_i \pi_i d_{[i]} ;|; \sum_i \pi_i = \varepsilon, ; 0 \leq \pi_i \leq p_{[i]} \right) \geq \zeta \tag{4-15}$$

The Algorithm 4 shows computation of the probability of the stochastic state entering the unsafe set $\bar{\mathcal{S}}$.

Algorithm 3 Compute Distance to Unsafe Region

```
Require: Point $x \in \mathbb{R}^D$, constraints list $C = [c_1, c_2, \dots, c_D]$, norm order $p$
Initialize empty list $\mathcal{D} \leftarrow []$
for $i = 1$ to $D$ do
    Extract lower and upper bounds: $[l_i, u_i] \leftarrow c_i$
    if $x_i < l_i$ or $x_i > u_i$ then
        return 0.0                                    ▷ Point is outside safe bounds
    end if
    $d_i \leftarrow \min(x_i - l_i, u_i - x_i)$
    Append $d_i$ to $\mathcal{D}$
end for
return $L^p$ norm: $(\sum_{d \in \mathcal{D}} d^p)^{1/p}$
```

4-1-4 Optimal Control Problem Redefinition

Now, using the uncertainty propagation framework described in previous sections, we formulate the stochastic optimal control problem by evaluating the expected cost of following a given policy. Recall that in SMPC the expectation of stage cost $\ell(x_{i|k}, u_{i|k})$ penalises deviations from the desired state and control effort, while the terminal cost $V_f(x_{N|k})$ represents the cost-to-go beyond the prediction horizon.

Within the context of this researcher, we consider both open- and closed-loop strategies:

$$u_k^{ol} = \pi(\theta) = \theta_k \qquad \text{open-loop policy} \tag{4-16}$$

$$u_k^{cl} = \pi(x_k, \theta) = K(x_k - \bar{x}_k) + \theta_k \qquad \text{closed-loop policy} \tag{4-17}$$

with $\bar{x}_k$ being the nominal trajectory of the system. Then the discrete measure at time step k is defined as $\tilde{\mathbb{P}}_{x_k} = \sum_i \tilde{\rho}_i \delta_{\tilde{c}_i}$ with $\tilde{\rho}$ and $\tilde{c}_i$ representing the compressed support points of

Algorithm 4 Compute Supremum Violation Probability

```
Require: Distances $\{d_i\}_{i=1}^N$, probabilities $\{\pi_i\}_{i=1}^N$, budget $\zeta$
Sort $(d_i, \pi_i)$ by increasing $d_i$
Initialize $T \leftarrow 0$, $V \leftarrow 0$                ▷ Transport used, Violation mass
for each $(d, \pi)$ in sorted list do
    $c \leftarrow d \cdot \pi$                                 ▷ Transport cost
    if $T + c \leq \zeta$ then
        $T \leftarrow T + c$
        $V \leftarrow V + \pi$
    else
        $r \leftarrow \zeta - T$                               ▷ Remaining budget
        $V \leftarrow V + \frac{r}{d}$                         ▷ Partial mass
        break
    end if
end for
return $V$                                   ▷ Upper bound on violation probability
```

the discrete measure through the (nonlinear) dynamics with disturbance. Given the target state x_{ref}, the stage cost component at prediction step $i|k$ is defined as:

$$\begin{aligned}\ell(x_{i|k}, u_{i|k}, \omega_{i|k}) &= \frac{1}{2}\Big((\mu_x - x_{ref})^\top Q\,(\mu_x - x_{ref}) + tr\,(Q\Sigma_x)\Big) + u^\top Ru = \\ &= \frac{1}{2}\sum_{i,j} \tilde{\pi}_i\tilde{\pi}_j\,(\tilde{c}_i(\theta) - x_{ref})^\top Q\,(\tilde{c}_j(\theta) - x_{ref}) + u^\top Ru.\end{aligned} \tag{4-18}$$

The terminal cost is given by:

$$V_f(x_{N|k}) = ||\mu_x - x_{ref}||^2_{Q_{term}} + tr\,(Q_{term}\Sigma_x) \tag{4-19}$$

with μ_x and Σ_x being the mean and the covariance of the approximated probability distribution of the state at $N|k$, respectively, and Q_{term} the terminal cost weight matrix.

Under the sequence of the control laws $\boldsymbol{\pi}(\theta) : \pi_{0|k}(\cdot), \ldots, \pi_{N-1|k}(\cdot)$, the cumulative cost is defined a:

$$J_N^{\boldsymbol{\pi}}(x, \theta) = \mathbb{E}_{|x}\left[\sum_{i=0}^{N-1} \ell(x_{i|k}, u_{i|k}, \omega_{i|k}) + V_f(x_{N|k})\right] + J_{obs}, \tag{4-20}$$

$$J_N^{\boldsymbol{\pi}}(x, \theta) = \sum_{i=0}^{N-1} \mathbb{E}_{|x}\left[\ell(x_{i|k}, u_{i|k}, \omega_{i|k})\right] + \mathbb{E}_{|x}V_f(x_{N|k}) + J_{obs}, \tag{4-21}$$

where $\mathbb{E}_{|x}(\cdot) := \mathbb{E}(\cdot \mid x_0 = x)$ denotes the conditional expectation given the initial state with respect to the probability measure $\hat{\mathbb{P}}$ defined on the underlying probability space. J_{obs} denotes the penalty cost computed according to Equation 4-13. Ultimately, the OCP is reformulated as Problem 2.

Problem 2 (Stochastic Optimal Control Problem via Quantisation)**.**

$$
\begin{aligned}
\min_{\theta\in\Theta} \quad & J_N^{\pi}(x,\theta) \\
s.\ t. \quad & c^{n}_{i+1|k} = (f_c(c^{(a)}_{i|k}, u_{i|k}) + \omega^{(b)}_{i|k}), && \forall i \in \{0, ..., N-1\}, \quad \forall n \in \{0, \dots, n_{max}\} \\
& && \forall a \in \{0, \dots, N_x\} \quad \forall b \in \{0, \dots, N_\omega\} \\
& c_{i|k} \in \mathcal{S} && \forall i \in \{0, ..., N\} \\
& u_{i|k} = \pi_{i|k}(c_{i|k}, \theta), && \forall i = 0, \dots, N-1, \\
& c_{i|k} \in \mathcal{X}, \quad u_{i|k} \in \mathcal{U} && \forall i = 0, \dots, N-1, \\
& x_{0|k} \sim \hat{\mathbb{P}}_{x_k},
\end{aligned}
\tag{4-22}
$$

Here, f_c denotes the dynamics applied to the compressed representation. The sets $\mathcal{X}$ and $\mathcal{U}$ represent admissible state and input sets, respectively, and $\mathcal{S}$ denotes the safety set (see Section 4-1-3). The discrete system is evolved forward using the control policy and noise realisations, and compression is applied after each propagation step. This optimisation problem is solved with Sequential Quadratic Programming (SQP) [1] and Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) [51]. As the implementation is tailored to each case study, the details follow in Section 5.

[1] see Numerical Methods for Stochastic MPC in Appendix C

Chapter 5

Experimental Results

This chapter evaluates the effectiveness of the proposed framework through empirical studies across benchmark scenarios. Specifically, we assess controller performance in terms of computational efficiency, predicted bounds, and the ability to reach the target under uncertainty. To emphasise the benefits of feedback in stochastic environments, we compare open-loop and closed-loop control strategies.

The chapter is structured as follows: Section 5-1 outlines the simulation setup and evaluation procedure. Sections 5-2 and 5-3 present two case studies involving Dubins Dynamics (DD) and Dynamic Bicycle Model (DBM), respectively, where the performance of the proposed controller is thoroughly analysed.

5-1 Simulation Procedure

Assuming that the states can be measured directly, the simulation is performed according to Algorithm 5. To account for model uncertainties, we consider two cases: one where the initial state is modelled as a Gaussian distribution, and another where both the initial state and additive noise are applied at each time step.

All parameters for the system simulation and MPC setup are tailored/tuned according to the case study. Given the stochastic nature of the systems, the simulation is terminated once the vehicle approaches the terminal region defined as $x_\star \pm 0.5$ [m].

There is a possibility for a user to customise the studied case by investigating open- or closed-loop, not noisy and noisy case, and an environment with obstacles or without. The repository [1] contains the source code for all the experiments.

Solving Optimal Control Problem (OCP) is done with constrained (SLSQP) and unconstrained (L-BFGS) nonlinear gradient-based algorithms. All simulations were run on a 2019 MacBook equipped with an i5 processor and 8GB of RAM.

[1] https://github.com/RoniTa287/stochastic-mpc

Algorithm 5 Simulation Procedure

```
Require: Initial conditions $x \sim \mathbb{P}_{x_0} = \mathcal{N}(\mu_x, \Sigma_x)$; noise $\omega \sim \mathbb{P}_\omega = \mathcal{N}(\mu_\omega, \Sigma_\omega)$; MPC configurations $(N, Q, R, Q_{\text{term}})$
Sample $x^s \sim \mathbb{P}_{x_0}$
for $k = 0$ to $N_{\text{sim}} - 1$ do
    Discretize $\mathbb{P}_x$ to obtain $\hat{\mathbb{P}}_{x_k}$
    find optimal policy $\pi^*(x, \theta)$ with $x \sim \hat{\mathbb{P}}_{x_k}$    ▷ Solve OCP with SQP or L-BFGS
    $x^s_{k+1} \leftarrow f(x^s, u^0)$    ▷ Apply first optimal control input
    Compute violation probability $\mathbb{P}(x \in \bar{\mathcal{S}})$    ▷ Algorithm 4
    $\mathbb{P}_{x_{k+1}} := \mathcal{N}(x^s_{k+1}, \Sigma_\omega)$
    if $x^s_{k+1}$ is within the terminal region then
        Terminate the simulation
    end if
    $x^s \sim \mathbb{P}_{x_{k+1}}$
end for
```

5-2 Dubin's dynamics (DD)

This subsection evaluates the proposed framework on a nonlinear system using the Dubins model [24], a canonical example of a simple nonholonomic vehicle often used in control and motion planning. It serves as a simplified dynamics of an automated car [27, 56] or a UAV [3].

The vehicle moves at a constant velocity v and is steered by controlling its angular velocity. Let the state of the system be represented by $\mathbf{x}_k = [x, y, \phi]^\top$, where x and y denote the position coordinates, and ϕ represents its heading angle in *rad*. The discrete-time dynamics of the Dubins car with time step T [*sec*] are given by:

$$\mathbf{x}_{k+1} = f_{DD}(\mathbf{x}_k, u_k) + \omega_{DD} \quad \text{with} \quad f_{DD}(\mathbf{x}_k, u_k) = \begin{bmatrix} x_k + v\cos(\phi_k)T, \\ y_k + v\sin(\phi_k)T, \\ \phi_k + u_k T. \end{bmatrix} \tag{5-1}$$

where the control input u_k [*rad/sec*] corresponds to the angular velocity rate of change. Since the Dubins dynamics are originally continuous, we discretise them using a sampling time of 0.1 seconds, with integration carried out via a fourth-order Runge-Kutta (RK4) method. Moreover, it is assumed that we have full state observability, so $f_{DD} : \mathbb{R}^3 \to \mathbb{R}^3$. ω_{DD} is i.i.d. Gaussian additive noise with $\sim \mathcal{N}(0^{(3,1)}, diag([0.01, 0.01, 0.001]))$ unless otherwise specified. Full list of parameters can be found in Table B-1.

5-2-1 Safety Constraints

We define a safe region $\mathcal{S} \subset \mathbb{R}^2$ that constrains the vehicle's position to remain within a rectangular set. Additionally, we consider environments with static obstacles. These obstacles are modelled as hyper-rectangular regions in the x_1-x_2 plane that the system must avoid. Let obstacle(s) be defined as $\mathcal{O} \subset \mathbb{R}^2$, then the safe set $\mathcal{S}$ becomes the allowable state space excluding the obstacles:

$$\mathcal{S} = \left\{ \mathbf{x} \in \mathbb{R}^3 \,\middle|\, x_1 \in [-1, 12],\ x_2 \in [-1, 12], (x_1, x_2) \notin \mathcal{O} \right\}. \tag{5-2}$$

where $[-1, 12]$ defines the allowable bounds on the horizontal and vertical position coordinates, respectively. A penalty-based approach, as described in Section 4-1-3, is used to enforce obstacle avoidance.

5-2-2 Optimisation procedure

To solve this problem, we investigate two numerical strategies: Sequential Least Squares Programming (SLSQP) [46] and the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) method. The SLSQP solver allows direct inclusion of box constraints on position and input bounds as inequality constraints. Obstacle avoidance is implemented using a penalty-based formulation as described in Section 4-1-3 to avoid disjunctive constraints.

In contrast, L-BFGS does not handle constraints and, thus, requires reformulating the control policy to inherently satisfy input limits. This is achieved by reparametrising the control input using a smooth, bounded function. Specifically, we introduce mapping via sigmoid function , so the control variable is defined as

$$\pi_k = 2u_{\max}\sigma(\theta) - u_{\max} \quad \text{with} \quad \sigma(\theta) = \frac{1}{1 + e^{-\theta}} \tag{5-3}$$

where θ is the unconstrained optimisation variable, $u_{\max}$ represents the maximum allowable input magnitude. This transformation ensures that the control inputs remain within admissible bounds while enabling the use of unconstrained gradient-based solvers such as L-BFGS.

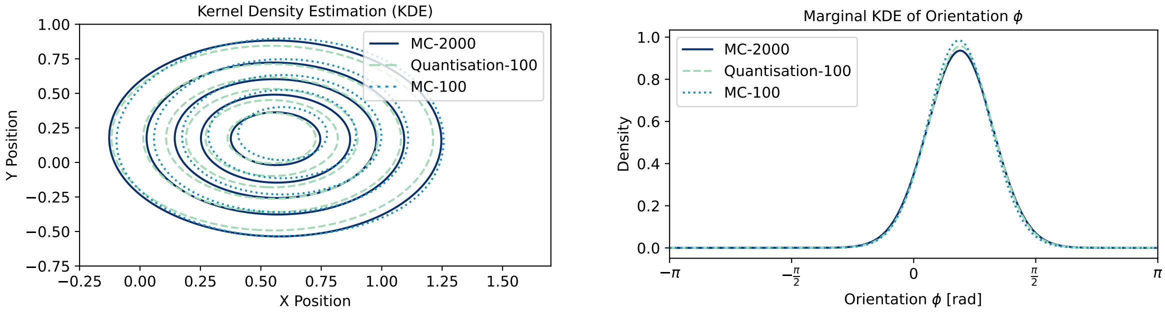
To accelerate convergence and maintain consistency across time steps, a warm-starting strategy is applied. The first iteration is initialised randomly, whereas subsequent optimisations use the solution from the previous step as the initial guess. Gradients required by both solvers are computed via automatic differentiation.

Depending on the solver, the prediction horizon is chosen (for SLSQP, we need a higher prediction horizon to let the KKT system see when it approaches tricky parts unlike for Quasi Newton method, we can make it shorter, as it sees the growth in penalty cost faster) 'maxiter': 100 used for the solvers 100.

5-2-3 Uncertainty prediction analysis

Figure 5-1 illustrates the Kernel Density Estimation (KDE) of the state distribution in the Dubins dynamic model after five time steps, comparing three scenarios: a high-fidelity Monte Carlo baseline with 2000 particles (MC-2000), a quantised and compressed representation with 100 particles, and a low-fidelity Monte Carlo approximation with 100 particles (MC-100). The left subplot 5-1a visualises the spatial distribution on the $x - y$ plane, while the right subplot 5-1b shows the marginal KDE of the orientation angle ϕ. The MC-2000 serves as the reference distribution due to its high sample resolution.

Visually, both the compressed quantisation approach and MC-100 produce KDE contours that closely align with the MC-2000 reference. However, the visual observations are quantitatively supported in Table 5-1, which reports the multivariate Wasserstein-2 distance between the approximated and the baseline distributions over ten prediction steps. Notably, the quantisation method consistently achieves lower distances than MC-100. This highlights the effectiveness of the compression scheme in preserving the essential structure of the probability distribution, even with significantly fewer representative particles.



(a) Location on x-y plane

(b) Angle

Figure 5-1: Kernel Density Estimation for the probability density function of the Dubins Dynamics after five (5) time steps. Comparison of the Discretised and MC with 100 samples with MC 2000 samples ("true dist")

Table 5-1: Wasserstein distance $\mathbb{W}_2$ to MC-2000 over 10 steps

Prediction Step	MC-100 vs MC-2000	Quantisation vs MC-2000
0	0.00688	0.00233
1	0.01027	0.00619
2	0.02308	0.01031
3	0.02394	0.01621
4	0.04091	0.02429
5	0.04543	0.03149
6	0.06080	0.03824
7	0.06177	0.04543
8	0.07279	0.05733
9	0.08266	0.06838

5-2-4 Open-loop vs closed-loop with full noise propagation

We evaluate the effect of the control policies introduced in Equation 4-16. The experimental setup uses discretisation with 100 locations for both the state and noise distributions, with a maximum of 100 particles allowed after compression. The control optimisation is performed using the L-BFGS solver. The system noise is $\omega \sim \mathcal{N}(0^{(3,1)}, \mathrm{diag}(0.01, 0.01, 0.001))$.

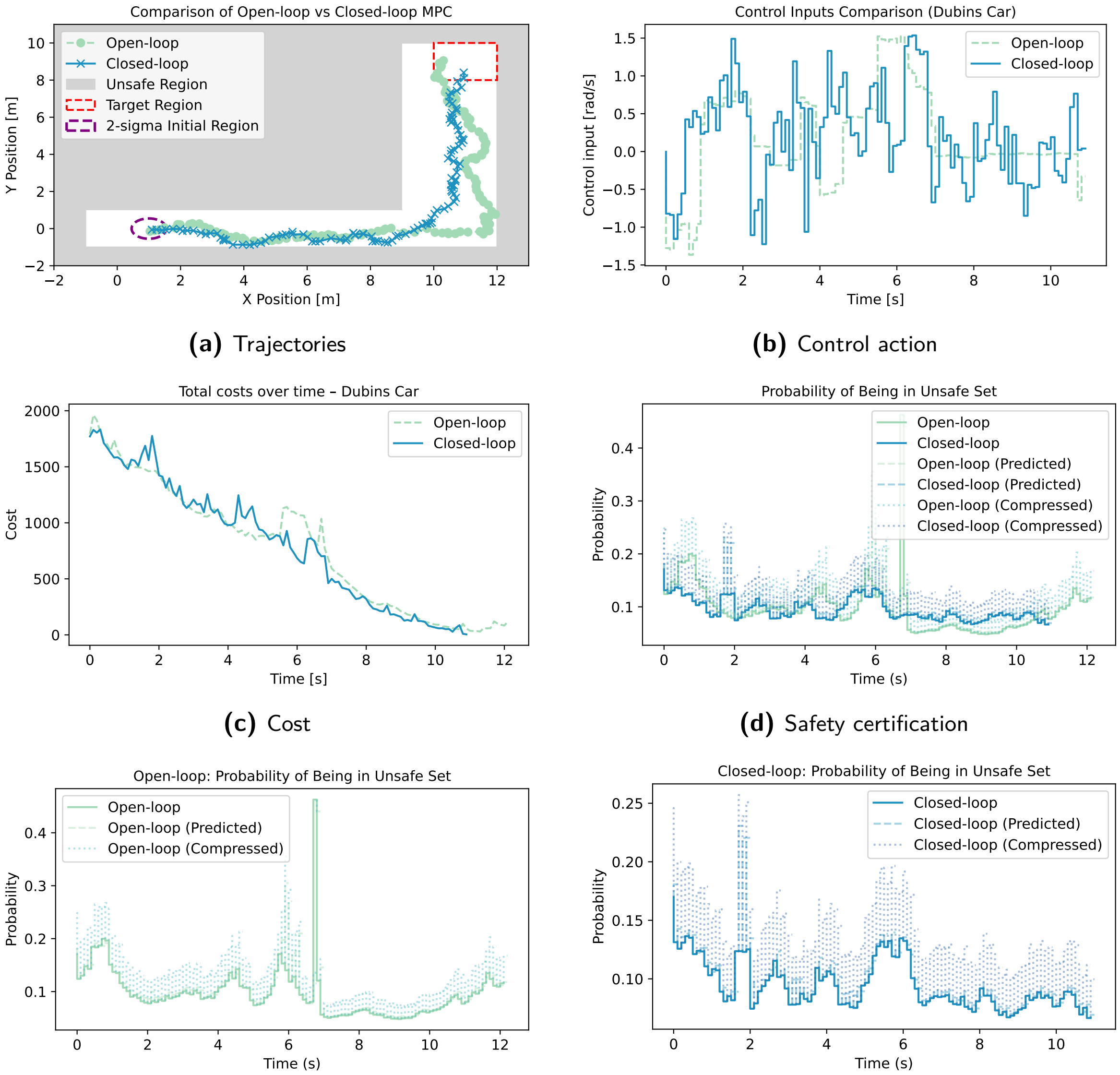
Figure 5-2 illustrates the resulting trajectories and control signals under open- and closed-loop control. As shown in Figures 5-2a–5-2b, the closed-loop policy yields smoother and more robust trajectories, albeit at the cost of more aggressive control actions.

Safety guarantees are certified at each time step, as discussed in Section 4-1-3, including the predicted probability of the state entering the unsafe set (Equation 4-4) and its compressed approximation (Equation 4-6). The results are presented in Figures 5-2e–5-2f.

Additionally, table 5-2 summarises the quantitative performance of the two approaches. Although both policies successfully reach the target state, the computational cost of solving the associated optimal control problem remains prohibitively high for real-time applications, with average solve times ranging from 230 to 325 seconds per OCP instance.

Table 5-2: Controller performance when the car is performing a turning manoeuvre in a narrow corridor

Metric	Open-loop	Closed-loop
Target reached in [s]	12.10	10.90
Optimisation time [s]	2767.23	3548.85
Average optimisation time per time step [s]	230	325
Max $\mathbb{P}(x \in \bar{\mathcal{S}})$	0.476	0.172



(a) Trajectories

(b) Control action

(c) Cost

(d) Safety certification

(e) Safety certification open loop

(f) Safety certification closed-loop

Figure 5-2: Comparison of simulation results using open- and closed-loop policies for a Dubins car performing a turning manoeuvre in a constrained environment. N=8, noise ([0.01, 0.01, 0.001]), init cond $\mathcal{N}(0, 0.01I)$. The rows show: (a-b) state trajectories and control action, (c-d) cost over time and probability of state getting into unsafe set $\mathbb{P}(\bar{\mathcal{S}})$, (e-f) $\mathbb{P}(\bar{\mathcal{S}})$ under open- and closed-loop strategies.

5-2-5 Robustness to unseen noise

To assess the robustness of the closed-loop controller under stochastic disturbances, we analyse a scenario in which process noise is not accounted for during the optimisation, but is instead injected at each simulation time step.

Figure 5-3 represents the results of this experiment where the noise propagation and compression layer are omitted during the optimisation. Consequenctly, only the initial distribution $x_0 \sim \mathbb{P}_{x_0}$ is propagated over the prediction horizon. The closed-loop system is still able to keep the states within the safe set $\mathcal{S}$. However, as shown in Figure 5-3b, the control action is very aggressive compared to the previous result, where noise propagation was included during optimisation.

Finally, Table 5-3 confirms that the optimisation is a bottleneck in the proposed framework, even though is it comparably faster than in the case with noise propagation and compression.

Table 5-3: Runtime analysis of key computational components for open- and closed-loop control

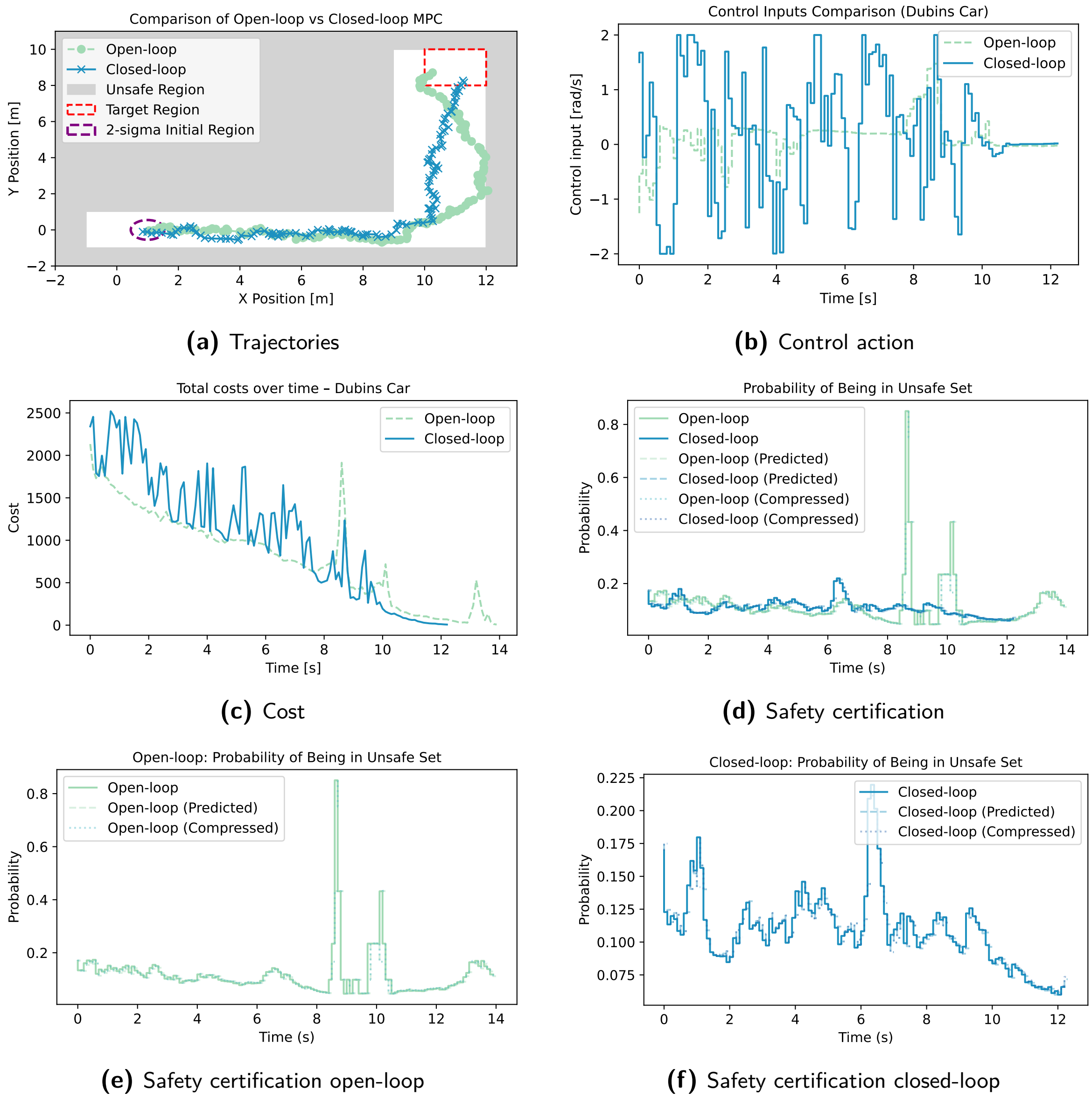
Component	**Open-loop**	**Closed-loop**
Propagation [ms]	0.43 ± 0.09	0.50 ± 0.27
Formal bound computation [ms]	7.15 ± 1.11	8.16 ± 1.71
Compression [s]	-	-
Optimisation time [s]	0.68 ± 0.35	1.91 ± 0.47

5-2-6 Scalability and Quantisation Analysis

To evaluate the scalability of the designed controller, we assess its performance in a more complex environment, illustrated in Figure 5-4a. In this setting, a longer prediction horizon is required to prevent the car from becoming trapped near the point $(11, 3)$ on the x-y plane. Accordingly, the prediction horizon was increased to 15 and 18 for the SL-Sequential Quadratic Programming (SQP) and Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) solvers, respectively.

We also investigate how the number of locations used in the quantisation process influences controller performance. Table 5-4 summarises the key metrics — specifically, the optimisation time and the maximum value of $\mathbb{P}(x \in \hat{\mathcal{S}})$ — across three levels of discretisation: 100, 490, and 1000 components. As expected, increasing the number of discrete signatures improves the representation of the underlying probability distribution, resulting in tighter bounds and more accurate safety guarantees.

Additionally, Figures 5-4g and 5-4h visualise the predicted probability of entering the unsafe set at the next step if the same policy is applied. It is evident that SL-SQP produces a more conservative (i.e., safer) policy and consistently respects the imposed constraints, as these are directly embedded within its optimisation framework. This safety, however, comes at the expense of higher computation times, as shown in Table 5-4. In contrast, L-BFGS is generally faster, likely due to its treatment of constraints via soft penalties in the cost function. While this approach improves runtime, mapping the control inputs through sigmoid transformations to enforce control limits introduces additional complexity and may degrade optimisation efficiency.



(a) Trajectories

(b) Control action

(c) Cost

(d) Safety certification

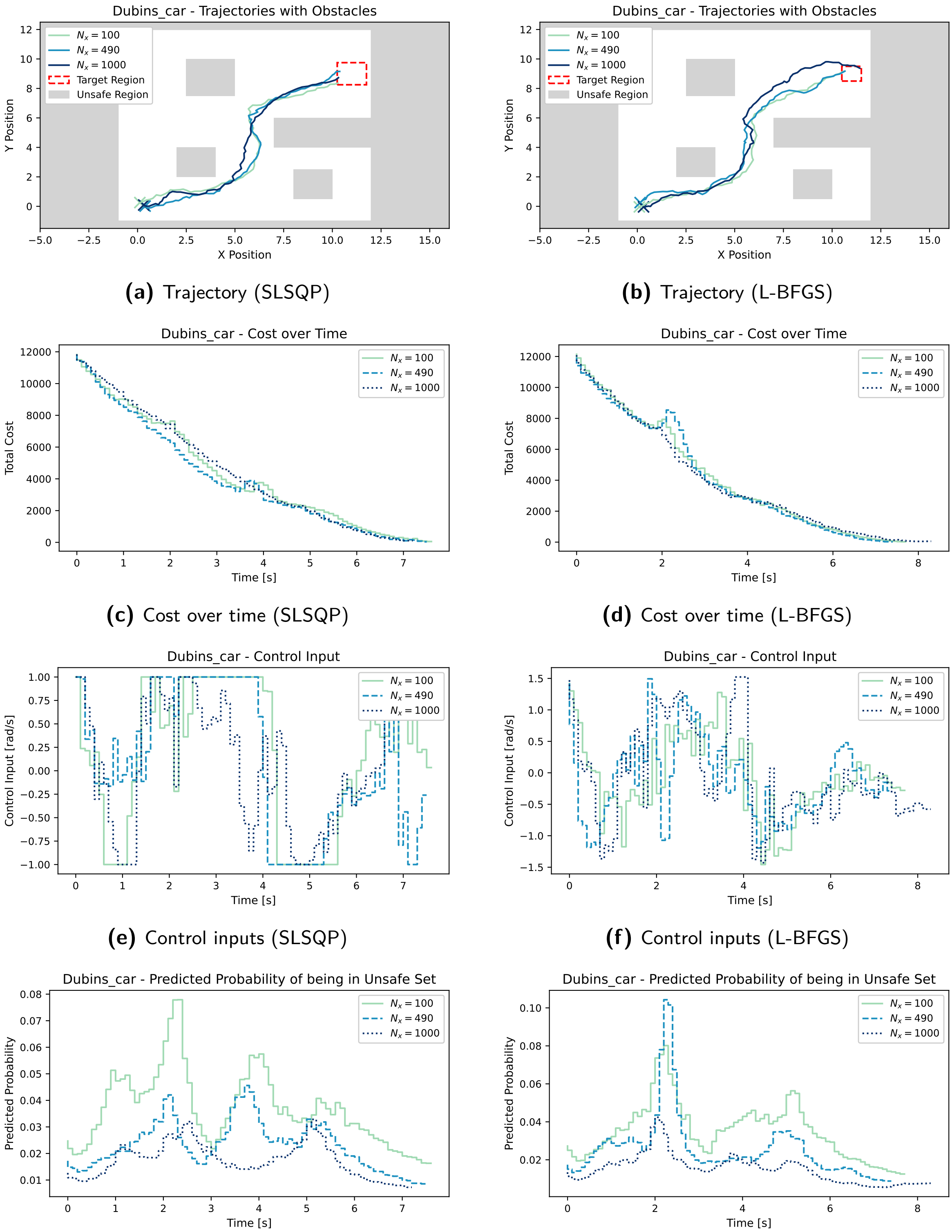
(e) Safety certification open-loop

(f) Safety certification closed-loop

Figure 5-3: Comparison of simulation results using open- and closed-loop policies for a Dubins car performing a turning manoeuvre in a constrained environment. N=8, noise ([0.01, 0.01, 0.001])), init cond $\mathcal{N}(0, 0.01I)$. The rows show: (a-b) state trajectories and control action, (c-d) cost over time and probability of state getting into unsafe set $\mathbb{P}(\bar{\mathcal{S}})$, (e-f) $\mathbb{P}(\bar{\mathcal{S}})$ under open- and closed-loop strategies.

Table 5-4: Controller performance in environment with static obstacles depicted in Figure 5-4b

	SLSQP			**L-BFGS**		
Number of Signatures	100	500	1000	100	500	1000
Target reached in [s]	7.5	7.4	7.1	7.3	7.7	8.2
Optimisation time [s]	509.28	730.02	1015.58	149.60	185.83	231.72
Max $\mathbb{P}(x \in \bar{\mathcal{S}})$	0.078	0.044	0.032	0.11	0.08	0.04



(a) Trajectory (SLSQP)

(b) Trajectory (L-BFGS)

(c) Cost over time (SLSQP)

(d) Cost over time (L-BFGS)

(e) Control inputs (SLSQP)

(f) Control inputs (L-BFGS)

(g) Probability of unsafe states (SLSQP)

(h) Probability of unsafe states (L-BFGS)

Figure 5-4: Comparison of simulation results using SLSQP and L-BFGS for a Dubins car in a constrained environment. The rows show: (a-b) state trajectories, (c-d) cost over time, (e-f) control inputs, and (g-h) probabilities of entering unsafe states.

5-3 Dynamic Bicycle Model

Furthermore, we apply the proposed framework to a vehicle dynamics-based model [29]. The derivatives of such a model are used for the lane keeping [53] and obstacle avoidance [28] control of a semi-autonomous vehicle. Gao et al. [28] used a simplified bicycle model and implemented robust NModel Predictive Control (MPC) with tightened input and state constraints to ensure constraint satisfaction in the presence of unknown bounded disturbances.

The state vector of the vehicle is defined as $\mathbf{x} = \begin{bmatrix} x & y & \phi & v_x & v_y & \omega \end{bmatrix}^\top$, where x, y are the global position of the vehicle, ϕ is the yaw angle (heading), v_x is the longitudinal velocity in the vehicle frame, v_y is the lateral velocity in the vehicle frame, ω is the yaw rate. The control input u is: $\begin{bmatrix} a & \delta \end{bmatrix}^\top$ with longitudinal acceleration a and a steering angle of the front wheel δ.

Under the assumption of small steering angles and a linear tyre model, the discrete-time dynamics are given by $f_{DBM} : \mathbb{R}^6 \to \mathbb{R}^6$ [29]:

$$\mathbf{x}_{k+1} = f_{DBM}(\mathbf{x}, u) + \omega_{DBM} \quad \text{with}$$
$$f_{DBM} = \begin{bmatrix} x_k + T(v_{x,k} \cos\phi_k - v_{y,k} \sin\phi_k) \\ y_k + T(v_{y,k} \cos\phi_k + v_{x,k} \sin\phi_k) \\ \phi_k + T\omega_k \\ v_{x,k} + Ta_k \\ \frac{m v_{x,k} v_{y,k} + T(l_f k_f - l_r k_r)\omega_k - T k_f \delta_k v_{x,k} - T m v_{x,k}^2 \omega_k}{m v_{x,k} - T(k_f + k_r)} \\ \frac{I_z v_{x,k} \omega_k + T(l_f k_f - l_r k_r) v_{y,k} - T l_f k_f \delta_k v_{x,k}}{I_z v_{x,k} - T(l_f^2 k_f + l_r^2 k_r)} \end{bmatrix} \tag{5-4}$$

Here, T is the discretization step, m is the vehicle mass, I_z is the yaw inertia, l_f and l_r are distances from the center of gravity to the front and rear axles, and k_f, k_r are the cornering stiffness coefficients of the front and rear tyres, respectively.

5-3-1 Safety constraints and optimisation specifications

- input constraints: steering angle $\delta \in [-\pi/4; \pi/4]$ and longitudinal acceleration $a \in [-5; 2]$ m/s^2
- state constraint: longitudinal velocity $v_x > 0.5m/s$; lateral velocity v_y and yaw rate ω often are no bounded, but can $v_y \in [-4, 4]$ m/s; $\omega \in [-3, 3]$ rad/s.

To reflect the mechanical limitations of the vehicle we must implement the constraints on the control input. The steering angle and acceleration are expressed as

$$\delta = \frac{\pi}{4} \cdot \tanh(z_\delta), \quad a = \frac{7}{2} \cdot \tanh(z_a) - \frac{3}{2}, \tag{5-5}$$

where $z_\delta, z_a \in \mathbb{R}$ are unconstrained auxiliary variables. This transformation ensures that the resulting control values always satisfy the imposed bounds, allowing the original constrained problem to be reformulated as an unconstrained optimization, thereby simplifying solver implementation while maintaining constraint feasibility.

One downside of this implementation is that is slows down the solver, hence we preferred to add a penalty to the cost function, such as:

$$J_{penalty} = ReLU(a - 2.0)^2 + ReLU(-5.0 - a)^2 \tag{5-6}$$

The main idea is that we add a penalty when the control constraints are outside the desired bounds. Worth noting that this implementation is not as strict as the direct constraint elimination, so it must be tuned properly. Yet, this implementation still lets L-BFGS or Adam converge while learning to avoid invalid accelerations.

5-3-2 The effect of discretisation points

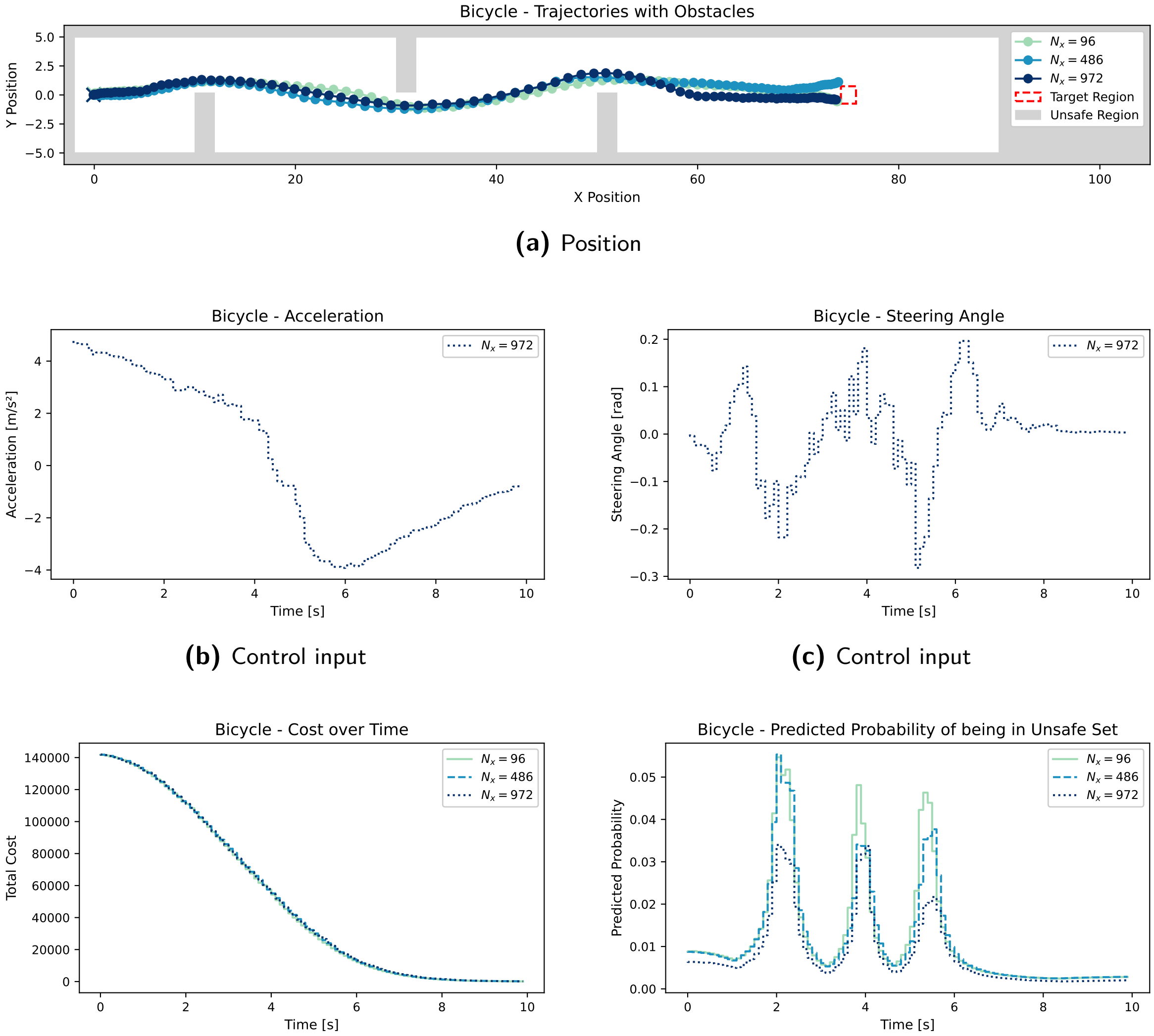
Figure 5-5 illustrates the effect of varying the number of discretisation points $N_x \in \{96, 486, 972\}$ on the control performance of the bicycle model tasked with obstacle avoidance. As shown in Figure 5-5a, increasing N_x leads to safer trajectory planning, particularly in the presence of obstacles. Similarly to the results on the previously studied system, the finer discretisation captures the uncertainty propagation more accurately, resulting in trajectories that consistently reach the target region while respecting safety margins. This is further supported by the predicted probabilities in Figure 5-5e, where higher N_x yields tighter and more conservative estimates of entering the unsafe set.

Subplots 5-5b and 5-5c show the corresponding control inputs (acceleration and steering angle). These results collectively confirm that increasing the number of discretisation points improves the controller's robustness to noise, though at the expense of increased computational complexity (see Table 5-5).

To further assess the scalability of the framework, we analyse the performance when full noise propagation and compression are included. The results, shown in Figure 5-6, indicate that safety guarantees are preserved (as seen in Figure 5-6e). However, the computational cost rises substantially, with the solver requiring up to 1.3 hours to complete ten 10-second simulations.

Table 5-5: Performance of L-BFGS under varying number of signatures

Number of Signatures	100	500	1000
Simulation time [s]	9.9	9.9	9.9
Optimisation time [s]	213.57	246.96	311.43
Max $\mathbb{P}(x \in \bar{\mathcal{S}})$	0.053	0.046	0.034



(a) Position

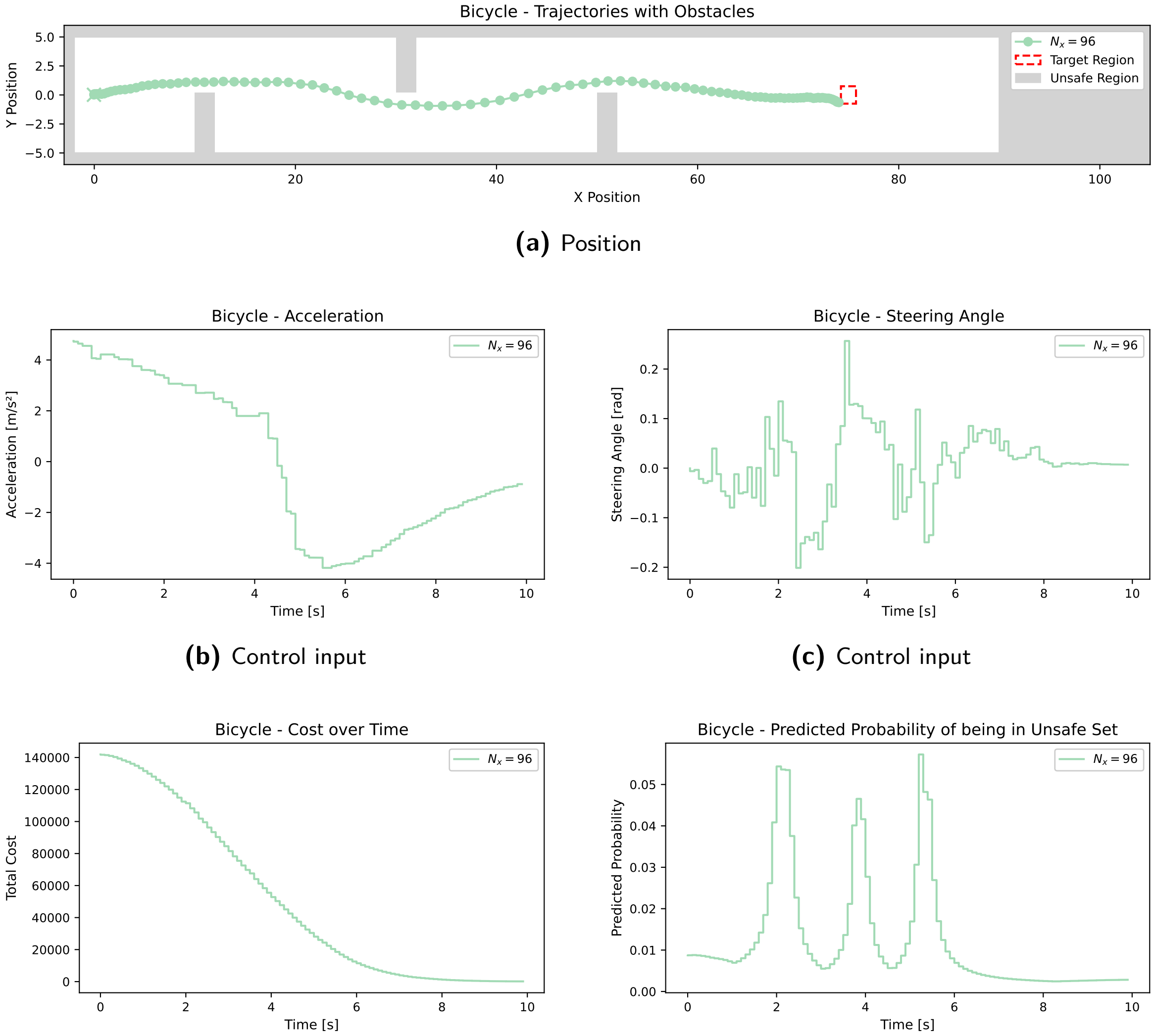
(b) Control input

(c) Control input

(d) Cost

(e) Probabilities

Figure 5-5: Bicycle model with initial $x_0 \sim \mathcal{N}(0, 0.1I)$ performing obstacle avoidance. Prediction horizon N=18; solver L-BFGS.



(a) Position

(b) Control input

(c) Control input

(d) Cost

(e) Probabilities

Figure 5-6: Bicycle noisy model with compression. Prediction horizon N=18. Optimisation algorithm L-BFGS; optimisation time 1.3 h.

Chapter 6

Discussions

6-1 Summary

This thesis introduced a novel quantisation-based Stochastic Model Predictive Control (SMPC) framework for uncertainty propagation and safety verification in nonlinear systems. By approximating both the state and noise distributions with discrete measures and applying a compression mechanism, the proposed method enables tractable multi-step uncertainty propagation and constraint handling, even when the system is affected by unbounded noise.

A key contribution of this work is the reformulation of the optimal control problem to incorporate robustified chance constraints, as well as non-convex obstacle avoidance implemented via a penalty-based approach. This formulation ensures compatibility with both constrained and unconstrained gradient-based optimisation algorithms. Furthermore, the framework integrates formal safety certification using Wasserstein ambiguity sets. Specifically, by evaluating the worst-case probability mass through the Wasserstein radius, the true probability of constraint violation under unbounded noise can be estimated multiple steps ahead.

Experimental results on nonlinear benchmarks, including the Dubins Dynamics (DD) and Dynamic Bicycle Model (DBM) models, demonstrate the effectiveness of the proposed framework. In particular, the compressed quantised approximation consistently outperforms low-fidelity Monte Carlo methods in Wasserstein-2 distance. Unlike sample-based MPC methods, which require a sufficiently large number of samples N_r to ensure statistical validity [64], the proposed approach achieves high accuracy with fewer particles due to the availability of formal bounds on the approximation error. This structural estimation of the underlying probability distribution enables efficient uncertainty propagation and improved computational scalability.

Furthermore, the use of affine feedback significantly reduces the probability of constraint violations compared to open-loop policies, even when noise propagation is omitted during optimisation.

These findings confirm that the proposed quantisation-based SMPC approach offers a theoretically sound and practically feasible solution for control under uncertainty. In particular, the results demonstrate that it is possible to control stochastic nonlinear systems affected by Gaussian noise while formally certifying the probability of the system's state entering an unsafe set.

6-2 Implications

This framework paves the way for deploying SMPC in safety-critical settings where real-time tractability and formal safety verification are crucial—e.g., autonomous driving, UAV path planning, or robotic motion control. The ability to approximate probability distributions compactly and propagate uncertainty through nonlinear dynamics with controlled error is especially powerful in high-dimensional or constrained environments. Furthermore, the modular nature of the implementation allows future integration of learning-based dynamics or real-time adaptation without compromising the safety layer.

6-3 Limitations and Future work

Despite its strengths, the proposed framework has several limitations that highlight avenues for practical improvement and further research.

1. **Computational burden and real-time applicability.** The most significant limitation is the long optimisation time. With average solve times of several minutes per control step, the framework is not currently suitable for real-time applications. While Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) provides faster convergence than SL-Sequential Quadratic Programming (SQP), both solvers remain sensitive to initialisation and hyperparameter tuning. Additionally, applying nonlinear transformations to enforce control input constraints—such as sigmoid mappings—further slows down the optimisation. Sequential Quadratic Programming (SQP) requires solving a QP at each iteration, while nonlinear interior-point methods (e.g., IPOPT [86]) are computationally cheaper per iteration. Future work should consider solver improvements and alternative methods such as structured MPC acceleration techniques [31].

2. **Discretisation fidelity and safety guarantees.** The quality of the safety guarantees depends on the accuracy of the quantised and compressed distributions. Poorly selected support points or insufficient particles can lead to under-approximation of risk, especially near constraint boundaries. Although the use of formal Wasserstein bounds improves robustness, discretisation remains a delicate balance between computational cost and precision.

3. **Limited generalizability and complexity.** Although the framework performs well on Dubins and dynamic bicycle models, its deployment on other systems is non-trivial. Designing the controller for a new system requires system-specific tuning of compression parameters, constraint representations, optimisation and solver parameters.

4. **A posteriori safety certification.** Currently, safety verification is performed post hoc using Wasserstein-based bounds. While this provides guarantees after optimisation, it

does not ensure that the selected control input respects the true chance constraint before application. A promising extension would be to embed Distributionally Robust Chance Constraint (DRCC) directly into the optimisation problem. Several approaches exist, such as CVaR-based DR-MPC using Wasserstein ambiguity sets [62], total-variation sets [23], and moment-based ambiguity sets with real-time guarantees [73]. However, their integration in nonlinear systems remains largely unexplored.

5. **Lack of smoothness in the quantisation process.** Quantisation is done by minimising the Wasserstein distance, but the resulting distribution may not be smooth. This can introduce discontinuities during the optimisation. Recent work on smoothing sample-based MPC [83] suggests a potential extension to regularise quantised approximations, improving numerical stability.

6. **Lack of stability analysis from the MPC perspective.** Stability of the closed-loop system is not analysed in this work. A rigorous investigation of Lyapunov-based stability or recursive feasibility under stochastic disturbances is necessary to ensure robustness over long horizons [58].

7. **Absence of noise-dependent control parametrisation.** The current formulation assumes a policy that depends only on the state. However, feedback laws parametrised directly by past noise values have shown improved performance under unbounded disturbances, especially using saturated noise inputs [38]. This line of work could further enhance control robustness.

8. **Full-state observability assumption.** The method assumes full access to the system state, which is often unrealistic in practical scenarios. An extension toward output-based SMPC is needed. Additionally, model uncertainty could be addressed via learning-based system identification techniques, such as Gaussian Processes [69], which provide uncertainty-aware models that integrate naturally with SMPC frameworks. This would also enable efficient reinforcement learning of control policies [21, 20].

These limitations underscore the trade-offs between theoretical rigour, computational tractability, and practical applicability. While the proposed approach provides a solid foundation for safe control under uncertainty, future research should address these open challenges to enable real-time, scalable deployment in complex, partially observed, and high-dimensional systems.

Appendix A

Nominal Nonlinear MPC

A-1 Cost function

Under appropriate terminal cost V_f and terminal region $\mathcal{X}_f$, this formulation leads to mean-square stability of the closed-loop system [45]. Specifically, the cost function acts as a stochastic Lyapunov function satisfying [45]:

$$\mathbb{E}_k \left[V_f(f(x, \pi(x), \omega)) \right] \leq V_f(x) - ||x||_Q^2 - ||\pi(x)||_R^2, \quad \forall x \in \mathcal{X}_f. \tag{A-1}$$

In the nonlinear case, provided the system dynamics and cost functions satisfy appropriate regularity and boundedness conditions, these formulations remain valid [45]. A quadratic function in Eq. (A-2) is commonly used to determine the cost of the system's predicted performance. This definition facilitates maintaining the convexity of the optimisation problem as long as Q and Q_{term} are positive semi-definite.

$$c_N(x, \boldsymbol{\pi}, \omega) = \sum_{i=0}^{N-1} \left(\|x_{i|k}\|_Q^2 + \|\mathbf{u}_{i|k}\|_R^2 \right) + \|x_{N|k}\|_{Q_{\text{term}}}^2. \tag{A-2}$$

Appendix B

Simulation and Optimisation Parameters for Studied Systems

Table B-1: List of Optimal Control Problem (OCP) Parameters for Dubin's Dynamics

Parameter	Value	Units	Explanation
Q	diag([60.0, 60.0, 1.0])	-	State penalty matrix
R	2diag([1.0, 1.0])	-	Control input penalty matrix
Q_{term}	diag([60.0, 60.0, 5.0])	-	Terminal penalty
T	0.1	s	Sampling time
N	varies	-	Prediction horizon
v	1.5	m/s	Velocity
x_0	$\mathcal{N}(0, 0.1I)$	m	Initial state
x_{ref}	[11.0, 9.0, 0.0]	m	Reference/Desired state
x_{min}	[-1.0, -1.0, None]	m	State constraints lower bound
x_{max}	[12.0, 12.0, None]	m	State constraints upper bound
u_{min}	-2	rad/s	Control input lower bound
u_{max}	2	rad/s	Control input upper bound
learning rate	0.02	-	for L-BFGS
maximum iterations	100/100	-	for L-BFGS/SLSQP
Penalty on box violation	450.0	-	Constraint violation cost
Penalty on obstacle violation	400.0	-	Obstacle cost (if any)

Table B-2: Simulation Parameters [29]

Parameter	Description	Value
I_z	yaw inertia of vehicle body	1536.7 kg·m^2
k_f	front axle equivalent sideslip stiffness	-128916 N/rad
k_r	rear axle equivalent sideslip stiffness	-85944 N/rad
l_f	distance between C.G. and front axle	1.06 m
l_r	distance between C.G. and rear axle	1.85 m
m	mass of the vehicle	1412 kg

Table B-3: OCP Parameters for Dynamic Bicycle Model. An explanation of each variable is provided in the previous section.

Parameter	Value	Units
Q	diag([50., 180., 10.5, 92.5, 0.5, 0.5])	-
R	diag([100., 100.*250.])	-
Q_{term}	1.5diag([61.0, 60.0, 10.5, 87.5, 3.5, 0.5])	-
T	0.1	s
N	varies	-
x_0	$\mathcal{N}(0, 0.1I)$	m
x_{ref}	[11.0, 9.0, 0.0]	m
x_{min}	[-1.0, -1.0, None]	m
x_{max}	[12.0, 12.0, None]	m
u_{min}	-2	rad/s
u_{max}	2	rad/s
learning rate	0.02	-
max iterations	100/100	-
Penalty on box violation	550.0	-
Penalty on obstacle violation	550.0	-

Table B-4: List of OCP Parameters for the Double Integrator

Parameter	Value	Units	Explanation
Q	diag([30.0, 5.0])	-	State penalty matrix
R	20	-	Control input penalty
Q_{term}	diag([40.0, 10.0])	-	Terminal state cost
T	0.1	s	Sampling time
N	varies	-	Prediction horizon
x_0	$\mathcal{N}(0, 0.05I)$	-	Initial state distribution
x_{ref}	[2.0, 0.0]	-	Target state
x_{min}	[-1.0, -2.0]	-	State lower bound
x_{max}	[3.0, 2.0]	-	State upper bound
Σ_ω	diag([0.001, 0.001])	-	Process noise covariance
Penalty on box violation	300.0	-	Constraint violation cost
Penalty on obstacle violation	300.0	-	Obstacle cost (if any)

Table B-5: List of OCP Parameters for the Inverted Pendulum on a Cart

Parameter	Value	Units	Explanation
Q	diag([25.0, 1.0, 120.0, 1.0])	-	State penalty matrix
R	15	-	Control input penalty
Q_{term}	$1.5 \times Q$	-	Terminal state cost
T	0.05	s	Sampling time
N	varies	-	Prediction horizon
x_0	$\mathcal{N}(0, 0.05I)$	-	Initial state distribution
x_{ref}	[0.0, 0.0, π, 0.0]	-	Upright target state
x_{min}	[-2.5, -10.0, $-\pi - 0.5$, -5.0]	-	State lower bound
x_{max}	[2.5, 10.0, $\pi + 0.5$, 5.0]	-	State upper bound
Σ_ω	$0.0001 \times I$	-	Process noise covariance
Penalty on box violation	700.0	-	Constraint violation cost
Penalty on obstacle violation	700.0	-	Obstacle cost (if any)

Appendix C

Numerical Methods for Nonlinear and Stochastic MPC

C-1 Numerical Methods for Nonlinear and Stochastic MPC

Within the Model Predictive Control (MPC) context, the term nonlinear MPC is often associated with solving an unconvex optimisation problem, as nonlinear dynamics, convexity is usually lost even if there is an implicit convexity assumption on the objective function and constraint sets [70]. If the solution to the nonconvex problem is found, one can certify that it is feasible or locally optimal.

Definition 6. *A point $z^\star \in R^{n_z}$ is a local minimiser of the problem (NLP) if there exists a neighborhood $\mathcal{N}$ of $z^\star$ such that $F(z^\star) \leq F(z)$ holds for all $z \in \mathcal{N} \cap \Omega$*

The first-order *necessary* conditions for optimality in a nonlinear constrained optimisation problem are Karush-Kuhn-Tucker (KKT) conditions.

Definition 7. *[11]: Let $z^\star$ and $(\lambda^\star, \nu^\star)$ be any primal and dual optimal points with zero duality gap. Since $z^\star$ minimizes the Lagrangian $L(z, \lambda^\star, \nu^\star)$ over z, its gradient must vanish at $z^\star$, i.e.,*

$$\nabla f_0(z^\star) + \sum_{i=1}^{m} \lambda_i^\star \nabla g_i(z^\star) + \sum_{i=1}^{p} \nu_i^\star \nabla h_i(z^\star) = 0 \tag{C-1}$$

Thus, we have the KKT conditions:

$$g_i(z^\star) \leq 0, \quad i = 1, \ldots, m \tag{C-2}$$

$$h_i(z^\star) = 0, \quad i = 1, \ldots, p \tag{C-3}$$

$$\lambda_i^\star \geq 0, \quad i = 1, \ldots, m \tag{C-4}$$

$$\lambda_i^\star g_i(z^\star) = 0, \quad i = 1, \ldots, m \tag{C-5}$$

$$\nabla f_0(z^\star) + \sum_{i=1}^{m} \lambda_i^\star \nabla g_i(z^\star) + \sum_{i=1}^{p} \nu_i^\star \nabla h_i(z^\star) = 0 \tag{C-6}$$

First step in solving a nonconvex problem with inequality constraint is to simplify it by reformulating the problem and eliminating the constraints [80]. Constraint elimination can be done if there exists mapping $\Phi : \bar{z} \longrightarrow z$ such that

$$\{z \mid z = \Phi(\bar{z}),\ \bar{z} \in \mathbb{R}^m\} = \{z \in \mathbb{R}^n \mid g(z) \leq 0\} \tag{C-7}$$

The constrained optimization of $f(z)$ over $z \in \mathbb{R}^n$ for $g(z) \leq 0$ now becomes an unconstrained optimization of $f(\Phi(\bar{z}))$ over $\bar{z} \in \mathbb{R}^m$.

If the inequality constraints are nonlinear, it complicates the problem, as the constraint set is not a hyperplane, so the gradient projection method cannot be used. A common way to deal with them is to incorporate the constraint function into the objective function through a penalty or a barrier function [80]. If the constraints are eliminated, first order and quasi-Newton optimisers (like BFGS or L-BFGS) can be used to solve the resulting unconstrained optimisation problem, but like any gradient-based method, it's local unless convex.

When the inequality constraints cannot be eliminated form the problem definition, the optimisation algorithms for constraint problems are used. Sequential quadratic programming (SQP) methods [7] solve in each iteration an inequality-constrained quadratic program (QP) that is obtained by linearising all problem functions, and it handles the constraints [35]. SLSQP solves Least Squares QPs (special case of SQP).

Algorithm 6 Sequential Quadratic Programming (SQP) Algorithm

```
Require: Initial guess $x_0$, Lagrange multipliers $\boldsymbol{\lambda}_0$, tolerance $\epsilon$
Set $k \leftarrow 0$
while not converged do
    Compute gradient $\nabla f(\boldsymbol{x}_k)$ and constraint gradients $\nabla g_i(\boldsymbol{x}_k)$
    Approximate the Hessian of the Lagrangian: $\boldsymbol{H}_k \approx \nabla^2_{xx}\mathcal{L}(\boldsymbol{x}_k, \boldsymbol{\lambda}_k)$
    Solve the Quadratic Programming (QP) subproblem:
        $\min_{\boldsymbol{d}} \quad \frac{1}{2}\boldsymbol{d}^\top \boldsymbol{H}_k \boldsymbol{d} + \nabla f(\boldsymbol{x}_k)^\top \boldsymbol{d}$
        subject to $g_i(\boldsymbol{x}_k) + \nabla g_i(\boldsymbol{x}_k)^\top \boldsymbol{d} \leq 0, \quad \forall i$
    Let $\boldsymbol{d}_k$ be the optimal direction and $\boldsymbol{\lambda}_k^*$ the optimal multipliers from the QP
    Compute dual update: $\boldsymbol{\Delta}_k \leftarrow \boldsymbol{\lambda}_k^* - \boldsymbol{\lambda}_k$
    Perform line search to find step size $s_k$ minimizing the merit function $\psi(\boldsymbol{x}_k + s_k\boldsymbol{d}_k)$
    Update primal and dual variables:
        $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k + s_k\boldsymbol{d}_k, \quad \boldsymbol{\lambda}_{k+1} \leftarrow \boldsymbol{\lambda}_k + s_k\boldsymbol{\Delta}_k$
    $k \leftarrow k + 1$
end while
return Optimal solution $\boldsymbol{x}^*, \boldsymbol{\lambda}^*$
```

Even though there are variants of SQP suitable for large-scale problems[8], when the size of the optimisation problem increases, it is generally preferable to use sparse solvers such as Interior Point [86].

Appendix D

Other systems: Double integrator and Pendulum on cart

The double integrator is a canonical/conventional second-order system, frequently used as a benchmark in control theory. Its state vector consists of position and velocity, and it evolves according to Newtonian mechanics:

$$\dot{x}_1 = x_2, \tag{D-1}$$

$$\dot{x}_2 = u, \tag{D-2}$$

where x_1 is position, x_2 is velocity, and u is the control force. We discretise these dynamics using the classical Runge-Kutta (RK4) integrator.

For the double integrator, box constraints are imposed on both position and velocity: $x_1 \in [-2, 2]$, $x_2 \in [-3, 3]$.

The inverted pendulum is a standard benchmark for underactuated control. Its state consists of cart position and velocity, and pendulum angle and angular velocity:

$$\mathbf{x} = [x, \dot{x}, \theta, \dot{\theta}]^\top. \tag{D-3}$$

The nonlinear dynamics are derived from Lagrangian mechanics and include gravitational and coupling terms:

$$\ddot{\theta} = \frac{g \sin\theta - \cos\theta \left(\frac{u + mL\dot{\theta}^2 \sin\theta}{M+m}\right)}{L\left(\frac{4}{3} - \frac{m\cos^2\theta}{M+m}\right)}, \tag{D-4}$$

$$\ddot{x} = \frac{u + mL\dot{\theta}^2 \sin\theta - mL\ddot{\theta}\cos\theta}{M+m}. \tag{D-5}$$

RK4 is used for discretisation to ensure numerical stability. The inverted pendulum uses constraints such as $x \in [-2.5, 2.5]$ and $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$ to enforce safety and stabilize around the upright position.

Bibliography

[1] Steven Adams, Patanè, Morteza Lahijanian, and Luca Laurenti. Finite neural networks as mixtures of gaussian processes: From provable error bounds to prior selection, 2024.

[2] Adriano Arrigo, Christos Ordoudis, Jalal Kazempour, Zacharie De Grève, Jean-François Toubeau, and François Vallée. Wasserstein distributionally robust chance-constrained optimization for energy and reserve dispatch: An exact and physically-bounded formulation. *European Journal of Operational Research*, 296(1):304–322, 2022.

[3] Andrew J. Barry, Anirudha Majumdar, and Russ Tedrake. Safety verification of reactive controllers for uav flight in cluttered environments using barrier certificates. In *2012 IEEE International Conference on Robotics and Automation*, pages 484–490, 2012.

[4] A. Bemporad. Reducing conservativeness in predictive control of constrained systems with disturbances. In *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No.98CH36171)*, volume 2, pages 1384–1389 vol.2, 1998.

[5] Alberto Bemporad, Tommaso Gabbriellini, Laura Puglia, and Leonardo Bellucci. Scenario-based stochastic model predictive control for dynamic option hedging. In *49th IEEE Conference on Decision and Control (CDC)*, pages 6089–6094, 2010.

[6] Dimitris Bertsimas, David B. Brown, and Constantine Caramanis. Theory and applications of robust optimization, 2010.

[7] Paul T. Boggs and Jon W. Tolle. Sequential quadratic programming. *Acta Numerica*, 4:1–51, 1995.

[8] Paul T. Boggs and Jon W. Tolle. Sequential quadratic programming for large-scale nonlinear optimization. *Journal of Computational and Applied Mathematics*, 124(1):123–137, 2000. Numerical Analysis 2000. Vol. IV: Optimization and Nonlinear Equations.

[9] Francesco Borrelli, Alberto Bemporad, and Manfred Morari. *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, 2017.

[10] Dimitris Boskos, Jorge Cortés, and Sonia Martínez. High-confidence data-driven ambiguity sets for time-varying linear systems. *IEEE Transactions on Automatic Control*, 69(2):797–812, 2024.

[11] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[12] Eric Bradford and Lars Imsland. Economic stochastic model predictive control using the unscented kalman filter. *IFAC-PapersOnLine*, 51(18):417–422, 2018. 10th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2018; This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 675215.

[13] Eric Bradford and Lars Imsland. Output feedback stochastic nonlinear model predictive control for batch processes. *Computers & Chemical Engineering*, 126:434–450, 2019.

[14] Eric Bradford, Lars Imsland, Dongda Zhang, and Ehecatl Antonio del Rio Chanona. Stochastic data-driven model predictive control using gaussian processes. *Computers & Chemical Engineering*, 139:106844, 2020.

[15] Giuseppe C. Calafiore and Marco C. Campi. The scenario approach to robust control design1. *IFAC Proceedings Volumes*, 39(9):602–607, 2006. 5th IFAC Symposium on Robust Control Design.

[16] Jicheng Chen and Yang Shi. Stochastic model predictive control framework for resilient cyber-physical systems: review and perspectives. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, 379(2207):pp. 1–13, 2021.

[17] Zhi Chen, Daniel Kuhn, and Wolfram Wiesemann. Data-driven chance constrained programs over wasserstein balls, 2022.

[18] Richard Cheng, Richard M. Murray, and Joel W. Burdick. Limits of probabilistic safety guarantees when considering human uncertainty, 2021.

[19] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[20] Marc Peter Deisenroth, Dieter Fox, and Carl Edward Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):408–423, 2015.

[21] Marc Peter Deisenroth and Carl Edward Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 465–472, Bellevue, WA, USA, 2011.

[22] Rijhi Dey, Naiwrita Dey, Rudra Sankar Dhar, Ujjwal Mondal, Sudhakar Babu Thanikanti, and Nnamdi Nwulu. Advances in controller design of pacemakers for pacing control: A comprehensive review. *Annual Reviews in Control*, 57:100930, 2024.

[23] Anushri Dixit, Mohamadreza Ahmadi, and Joel W. Burdick. Distributionally robust model predictive control with total variation distance, 2022.

[24] L. E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79(3):497–516, 1957.

[25] Lorenzo Fagiano and Mustafa Khammash. Nonlinear stochastic model predictive control via regularized polynomial chaos expansions. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 142–147, 2012.

[26] Eduardo Figueiredo, Steven Adams, Peyman Mohajerin Esfahani, and Luca Laurenti. Efficient uncertainty propagation with guarantees in wasserstein distance, 2025.

[27] Eduardo Figueiredo, Andrea Patane, Morteza Lahijanian, and Luca Laurenti. Uncertainty propagation in stochastic systems via mixture models with error quantification. *arXiv preprint arXiv:2403.15626*, 2024.

[28] Yiqi Gao, Andrew Gray, H. Eric Tseng, and Francesco Borrelli and. A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles. *Vehicle System Dynamics*, 52(6):802–823, 2014.

[29] Qiang Ge, Shengbo Eben Li, Qi Sun, and Sifa Zheng. Numerically stable dynamic bicycle model for discrete-time control, 2020.

[30] Xinbo Geng and Le Xie. Data-driven decision making in power systems with probabilistic guarantees: Theory and applications of chance-constrained optimization. *Annual Reviews in Control*, 47:341–363, 2019.

[31] L. Gharavi, C. Liu, B. De Schutter, and S. Baldi. Sensitivity analysis for piecewise-affine approximations of nonlinear programs with polytopic constraints. *IEEE Control Systems Letters*, 8:1271–1276, 2024.

[32] William Glover and John Lygeros. A stochastic hybrid model for air traffic control simulation. In Rajeev Alur and George J. Pappas, editors, *Hybrid Systems: Computation and Control*, pages 372–386, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[33] Edwin González, Javier Sanchis, Sergio García-Nieto, and José Salcedo. A comparative study of stochastic model predictive controllers. *Electronics*, 9(12), 2020.

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

[35] Sébastien Gros, Mario Zanon, Rien Quirynen, Alberto Bemporad, and Moritz Diehl and. From linear to nonlinear mpc: bridging the gap via the real-time iteration. *International Journal of Control*, 93(1):62–80, 2020.

[36] Lars Grüne and Jürgen Pannek. *Nonlinear Model Predictive Control: Theory and Algorithms.* Springer International Publishing, Cham, 2017.

[37] Lukas Hewing, Kim Peter Wabersich, Marcel Menner, and Melanie Nicole Zeilinger. Learning-based model predictive control: Toward safe learning in control. *Annu. Rev. Control. Robotics Auton. Syst.*, 3:269–296, 2020.

[38] Peter Hokayem, Debasish Chatterjee, and John Lygeros. On stochastic receding horizon control with bounded control inputs. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, page 6359–6364. IEEE, December 2009.

[39] Naira Hovakimyan, Chengyu Cao, Evgeny Kharisov, Enric Xargay, and Irene M. Gregory. L1 adaptive control for safety-critical systems: Guaranteed robustness with fast adaptation. *IEEE Control Systems*, 31(5):54–104, October 2011.

[40] Ran Ji and Miguel A. Lejeune. Data-driven distributionally robust chance-constrained optimization with wasserstein metric. *J. of Global Optimization*, 79(4):779–811, April 2021.

[41] Simon J Julier and Jeffrey K Uhlmann. A new extension of the kalman filter to nonlinear systems. In *Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, volume 3, pages 182–193. SPIE, 1997.

[42] Simon J Julier and Jeffrey K Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[43] Nikolas Kantas, Jan M Maciejowski, and Andrea Lecchini-Visintini. Sequential monte carlo methods for stochastic optimal control. *Handbook of Uncertainty Quantification*, 2020.

[44] B. Kouvaritakis and M. Cannon. *Model Predictive Control: Classical, Robust and Stochastic*. Advanced Textbooks in Control and Signal Processing. Springer International Publishing, 2015.

[45] Basil Kouvaritakis and Mark Cannon. Stochastic model predictive control. In *Encyclopedia of Systems and Control*. Springer, 2014. Available at: https://users.ox.ac.uk/~engs0169/pdf/kouvaritakis_encyclopedia14.pdf.

[46] Dirk Kraft. A software package for sequential quadratic programming. Technical Report DFVLR-FB 88-28, DLR German Aerospace Center – Institute for Flight Mechanics, Köln, Germany, 1988.

[47] Simge Küçükyavuz and Ruiwei Jiang. Chance-constrained optimization under limited distributional information: A review of reformulations based on sampling and distributional robustness, 2022.

[48] Daniel Landgraf, Andreas Völz, Felix Berkel, Kevin Schmidt, Thomas Specker, and Knut Graichen. Probabilistic prediction methods for nonlinear systems with application to stochastic model predictive control. *Annual Reviews in Control*, 56:100905, 2023.

[49] Rachel Lerman and Greg Bensinger. 17 fatalities, 736 crashes: The shocking toll of tesla's autopilot. *The Washington Post*, June 2023. Accessed: [24 Jun 2025].

[50] Changchun Liu, Andrew Gray, Chankyu Lee, J. Karl Hedrick, and Jiluan Pan. Nonlinear stochastic predictive control with unscented transformation for semi-autonomous vehicles. In *2014 American Control Conference*, pages 5574–5579, 2014.

[51] Dong C. Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45(1-3):503–528, 1989.

[52] Quan Liu, Zhihao Liu, Bo Xiong, Wenjun Xu, and Yang Liu. Deep reinforcement learning-based safe interaction for industrial human-robot collaboration using intrinsic reward function. *Advanced Engineering Informatics*, 49:101360, 2021.

[53] Wei Liu, Min Hua, Zhiyun Deng, Zonglin Meng, Yanjun Huang, Chuan Hu, Shunhui Song, Letian Gao, Changsheng Liu, Bin Shuai, Amir Khajepour, Lu Xiong, and Xin Xia. A systematic survey of control techniques and applications in connected and automated vehicles, 2023.

[54] James Luedtke and Shabbir Ahmed. A sample approximation approach for optimization with probabilistic constraints. *SIAM Journal on Optimization*, 19(2):674–699, 2008.

[55] Frederik Baymler Mathiesen, Simeon Calvert, and Luca Laurenti. Safety certification for stochastic systems via neural barrier functions, 2022.

[56] Frederik Baymler Mathiesen, Simeon C. Calvert, and Luca Laurenti. Safety certification for stochastic systems via neural barrier functions. *IEEE Control Systems Letters*, 7:973–978, 2023.

[57] D.Q. Mayne, S.V. Raković, R. Findeisen, and F. Allgöwer. Robust output feedback model predictive control of constrained linear systems. *Automatica*, 42(7):1217–1222, 2006.

[58] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[59] Ali Mesbah. Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems Magazine*, 36(6):30–44, 2016.

[60] Ali Mesbah, Stefano Di Cairano, and Ilya V. Kolmanovsky. Stochastic model predictive control. Technical Report TR2017-217, Mitsubishi Electric Research Laboratories (MERL), February 2017.

[61] Ali Mesbah, Stefan Streif, Rolf Findeisen, and Richard D. Braatz. Stochastic nonlinear model predictive control with probabilistic constraints. In *2014 American Control Conference*, pages 2413–2419, 2014.

[62] Francesco Micheli, Tyler Summers, and John Lygeros. Data-driven distributionally robust mpc for systems with uncertain dynamics, 2022.

[63] National Center for Statistics and Analysis, NHTSA. 2016 fatal motor vehicle crashes: Overview. Research Note DOT HS 812 456, U.S. Department of Transportation, National Highway Traffic Safety Administration, October 2017. Traffic Safety Facts.

[64] Arkadi Nemirovski and Alexander Shapiro. Convex approximations of chance constrained programs. *SIAM Journal on Optimization*, 17(4):969–996, 2007.

[65] Kristin Nielsen, Caroline Svahn, Hector Rodriguez-Deniz, and Gustaf Hendeby. Ukf parameter tuning for local variation smoothing. In *2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 1–8, 2021.

[66] Masahiro Ono. Joint chance-constrained model predictive control with probabilistic resolvability. In *2012 American Control Conference (ACC)*, pages 435–441, 2012.

[67] Kyriakos Polymenakos, Alessandro Abate, and Stephen Roberts. Safe policy search with gaussian process models, 2019.

[68] Hamed Rahimian and Sanjay Mehrotra. Frameworks and results in distributionally robust optimization. *Open Journal of Mathematical Optimization*, 3:1–85, July 2022.

[69] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006. c© 2006 Massachusetts Institute of Technology.

[70] James B. Rawlings, David Q. Mayne, and Moritz Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, Madison, WI, 2nd edition, 2017.

[71] J.B. Rawlings, E.S. Meadows, and K.R. Muske. Nonlinear model predictive control: A tutorial and survey. *IFAC Proceedings Volumes*, 27(2):185–197, 1994. IFAC Symposium on Advanced Control of Chemical Processes, Kyoto, Japan, 25-27 May 1994.

[72] Ugo Rosolia, Xiaojing Zhang, and Francesco Borrelli. Data-driven predictive control for autonomous systems. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:259–286, 2018.

[73] Kanghyun Ryu and Negar Mehr. Integrating predictive motion uncertainties with distributionally robust risk-aware control for safe robot navigation in crowds, 2024.

[74] Georg Schildbach, Lorenzo Fagiano, Christoph Frei, and Manfred Morari. The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations. *Automatica*, 50(12):3009–3018, 2014.

[75] Alexander T. Schwarm and Michael Nikolaou. Chance-constrained model predictive control. *AIChE Journal*, 45(8):1743–1752, 1999.

[76] Fred C. Schweppe. *Uncertain Dynamic Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1973.

[77] Society of Automotive Engineers. Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. Technical Report J3016–202104, SAE International, 2021. Accessed: [24 Jun 2025].

[78] Stefan Streif, Matthias Karl, and Ali Mesbah. Stochastic nonlinear model predictive control with efficient sample approximation of chance constraints, 2014.

[79] Gabriel Terejanu, Puneet Singla, Tarunraj Singh, and Peter D. Scott. Uncertainty propagation for nonlinear dynamic systems using gaussian mixture models. *Journal of Guidance, Control, and Dynamics*, 31(6):1623 – 1633, 2008.

[80] Ton van den Boom and Bart De Schutter. Optimization for systems and control: Lecture notes for the course sc42056, 2022. Lecture notes, September 2022 edition.

[81] Cédric Villani. *Optimal transport – Old and new*, volume 338. Springer-Verlag Berlin Heidelberg, 01 2008.

[82] Eric A Wan and Rudolph Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, pages 153–158. IEEE, 2000.

[83] Ye Wang, Xun Shen, and Hongyu Qian. Stochastic model predictive control with probabilistic control barrier functions and smooth sample-based approximation. In *2024 IEEE 63rd Conference on Decision and Control (CDC)*, pages 4798–4803, 2024.

[84] Florian Weissel, Marco F. Huber, and Uwe D. Hanebeck. A nonlinear model predictive control framework approximating noise corrupted systems with hybrid transition densities. In *Proceedings of the 46th IEEE Conference on Decision and Control (CDC)*, pages 3661–3666, New Orleans, LA, USA, December 2007. IEEE.

[85] Florian Weissel, Marco F. Huber, and Uwe D. Hanebeck. *Stochastic Nonlinear Model Predictive Control based on Gaussian Mixture Approximations*, pages 239–252. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

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

[87] Weijun Xie. On distributionally robust chance constrained programs with wasserstein distance, 2020.

[88] Weijun Xie and Shabbir Ahmed. On deterministic reformulations of distributionally robust joint chance constrained optimization problems. *SIAM Journal on Optimization*, 28(2):1151–1182, 2018.

[89] Xiaojing Zhang, Kostas Margellos, Paul Goulart, and John Lygeros. Stochastic model predictive control using a combination of randomized and robust optimization. In *52nd IEEE Conference on Decision and Control*, pages 7740–7745, 2013.

[90] Ya zhong Luo and Zhen Yang. A review of uncertainty propagation in orbital mechanics. *Progress in Aerospace Sciences*, 89:23–39, 2017.

Glossary

List of Acronyms

CVaR	Conditional Value-at-Risk
DR	Distributionally Robust
DRCC	Distributionally Robust Chance Constraint
DD	Dubins Dynamics
DBM	Dynamic Bicycle Model
GMM	Gaussian Mixture Model
KDE	Kernel Density Estimation
L-BFGS	Limited-memory Broyden–Fletcher–Goldfarb–Shanno
MC	Monte Carlo
MPC	Model Predictive Control
OCP	Optimal Control Problem
PDF	probabilistic density functions
PCE	Polynomial Chaos Expansion
SQP	Sequential Quadratic Programming
SMPC	Stochastic Model Predictive Control
UT	Unscented Transform

List of Symbols

$\mathbb{R}^n$	n-dimensional Euclidean space
$\mathcal{U}$	Closed set of control inputs
$\mathcal{X}$	Closed set of states
$f(\cdot)$	$f : \mathcal{X} \times \mathcal{U} \rightarrow \mathcal{X}$ nonlinear continuous /Borel-measurable function that characterises the system dynamics