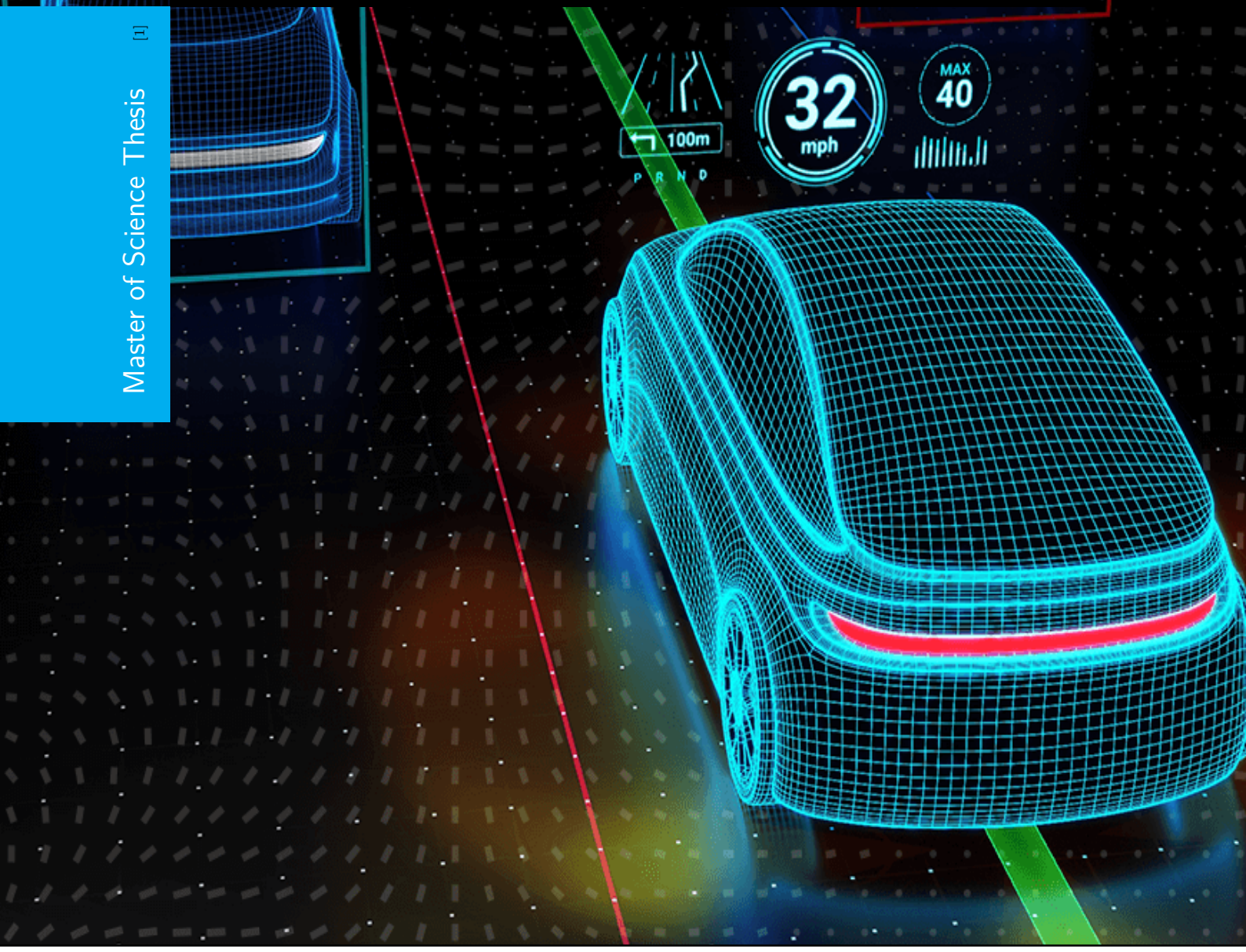
Efficient Data-driven Reference Governor Design for Safe Evasive Manoeuvring

Vehicle Motion Control

Petar Antonov Velchev

Master of Science Thesis [1]



TU Delft Delft University of Technology

Delft Center for Systems and Control

Efficient Data-driven Reference Governor Design for Safe Evasive Manoeuvring

Vehicle Motion Control

Master of Science Thesis

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

Petar Antonov Velchev

June 5, 2025

Faculty of Mechanical Engineering (ME) · Delft University of Technology

TOYOTA MOTOR EUROPE NV/SA

The work in this thesis was supported by Toyota Motor Europe. Their cooperation is hereby gratefully acknowledged.

Abstract

Automated Vehicles (AV) control at the limits of handling poses significant challenges due to the highly non-linear dynamics, tyre saturation effects, and the need for rapid decision-making. Traditional motion planning approaches often rely on global trajectory planners that do not fully account for real-time variations in vehicle dynamics or un-modelled environmental factors. Consequently, ensuring high tracking performance while simultaneously maintaining safety in the presence of uncertainties and obstacles remains an open problem. This thesis proposes a Bayesian Optimisation-based Reference Governor (BORG) that enhances the controller's ability to navigate complex scenarios by shaping a control-friendly reference trajectory. Unlike conventional replanning strategies that modify paths based on overly simplistic vehicle models, the BORG approach leverages the controller's knowledge of the vehicle's dynamics and control constraints to generate a refined reference signal that ensures both improved tracking performance and obstacle avoidance.

The BORG method operates in a model-free manner, optimising the reference trajectory without requiring an explicit system model, making it inherently robust to model mismatches and un-modelled disturbances. By integrating this optimised reference into a Model Predictive Control (MPC) framework, the proposed approach ensures that the generated trajectories are feasible and adhere to the vehicle's dynamic limitations. Furthermore, the reference governor not only enhances nominal MPC tracking performance but also guarantees safety by reshaping the trajectory to avoid obstacles, potentially even those not explicitly considered by a global planner. Through systematic optimisation, the reference signal accounts for the controlled system's behaviour, enabling smoother and dynamically feasible manoeuvres.

The effectiveness of this approach is evaluated through extensive simulation and experimental validation, demonstrating its capability to improve tracking performance while maintaining safe operation in uncertain and dynamically changing environments. Results show that the proposed method successfully balances aggressive performance optimisation with safety guarantees, making it a viable solution for high-performance autonomous driving applications where traditional model-based planners may fall short.

Table of Contents

1 Introduction **1**

1-1 Automated Vehicles . . . 1

1-2 Vehicle Motion Control . . . 3

1-2-1 Literature Study Review: Reference Governor . . . 4

1-2-2 Key Takeaways and Research Gaps . . . 5

1-2-3 Summary: Literature Review Conclusion . . . 5

1-3 Problem Formulation . . . 5

2 Model Predictive Control Design **7**

2-1 Model Predictive Control . . . 7

2-2 Predictive Models . . . 9

2-2-1 Vehicle Predictive Model . . . 9

2-2-2 Optimal Control Problem . . . 11

2-3 Non-linear Framework . . . 12

2-3-1 Overview of ACADO Toolkit . . . 12

2-3-2 Sequential Quadratic Programming . . . 12

2-3-3 Active Set Method (ASM) . . . 14

2-3-4 Handling Infeasibility . . . 14

2-4 Summary . . . 14

3 Bayesian Optimisation **17**

3-1 Black-box Global Optimisation . . . 17

3-2 Bayesian Optimisation Problem Formulation . . . 19

3-2-1 Gaussian Process Surrogate Model . . . 19

3-2-2 Acquisition Functions . . . 20

3-2-3 Maximisation of Acquisition Functions . . . 23

3-2-4 Bayesian Optimisation Algorithm . . . 25

3-3 Gaussian Processes . . . 25
3-3-1 Gaussian Process Regression . . . 25
3-3-2 Kernel Selection . . . 26
3-3-3 Hyperparameter optimisation . . . 27
3-3-4 Mean & Likelihood functions, Inference methods - probabilistic derivation 27
3-4 Summary . . . 28

4 Reference Governor for Nonlinear MPC 29
4-1 Optimisation of Sigmoid Parameters via Bayesian Optimisation . . . 29
4-2 Online Reference Governed MPC . . . 32
4-3 Summary . . . 33

5 Toward Efficient Surrogate Training 35
5-1 Method 1: Uniform Random Contextual Sampling Across Full Envelope . . . 36
5-2 Method 2: Critical Region Sampling with Structured Sweeps . . . 37
5-3 Method 3: Critical Zone Random Sampling with Representative Velocities . . . 37
5-4 Summary and Practical Considerations . . . 38

6 Simulation Environment 39
6-1 Overview Simulink system . . . 39
6-2 Controller Tuning . . . 40
6-3 Performance Evaluation . . . 42
6-3-1 Safety Indicator: Distance to Obstacle . . . 42
6-3-2 Trajectory Tracking: Root mean square errors . . . 42
6-3-3 Dynamic Response: Step-Inspired Metrics . . . 43
6-3-4 Summary of KPIs . . . 43
6-4 Robustness & Sensitivity Analysis Methodology . . . 44
6-4-1 Model Mismatch Simulation . . . 44
6-4-2 Perception and localisation errors . . . 45
6-4-3 Velocity-Dependent Evaluation . . . 45
6-5 Summary . . . 46

7 Performance and Results 47
7-1 Lane Change Manoeuvre Analysis . . . 47
7-1-1 Offline Bayesian Optimisation . . . 48
7-1-2 Lane Change Performed at 80 km/h with Obstacle at (420, -3) m . . . 49
7-1-3 Lane Change Performed at 70 km/h with Obstacle at (415, -4) m . . . 50
7-1-4 Lane Change Performed at 60 km/h with Obstacle at (420, -4) m . . . 52
7-2 Sensitivity and Robustness Analysis . . . 53
7-2-1 Tyre Model Mismatch . . . 53
7-2-2 Localization Mismatch . . . 54

8 Discussion & Conclusion **57**
8-1 Discussion 57
8-2 Future Work 58
8-3 Conclusions 58

A Vehicle parameters and controller settings **59**
A-1 Prediction horizon and sampling time 59
A-2 Vehicle parameters 59

B IEEE Robotics and Automation Letters Paper **61**

Glossary **77**
List of Acronyms 77

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisors, Alberto Bertipaglia, Mohammad Khosravi, and Barys Shyrokau, for their continued support and guidance throughout this thesis. Their regular feedback, valuable insights, and genuine interest in the project were instrumental in shaping both the research direction and the final outcome.

I am deeply thankful to Felipe Santafe, Giovanni Berardo, and Soichiro Suzuki for their immense support during my internship at Toyota Motor Europe. Their mentorship, technical expertise, and thoughtful suggestions had a profound impact on the development of this work from start to finish.

I would also like to thank my colleagues Alessandro, Hamid, and Francesco for their enjoyable company and the much-needed breaks from the screen during lunch and coffee hours. Your conversations and humor provided the perfect balance to long workdays.

To my fellow students Floris, Nathan, Abdulkayum, Bram, Niels, Sam, Leon, Hovsep, Daniel Mallika, and David—thank you for your companionship throughout the Master's journey. From academic support to memorable trips and countless foosball games, your friendship made this experience both productive and unforgettable.

Lastly, I am incredibly grateful to my family, my parents and my brother, for their unwavering support, encouragement, and patience. Your belief in me has always been a source of strength, and I would not have reached this point without you.

Delft, University of Technology
June 5, 2025

Petar Antonov Velchev

Chapter 1

Introduction

1-1 Automated Vehicles

Automated Vehicles (AV) hold significant promise in enhancing road safety by reducing human error, a factor implicated in approximately 94% of traffic fatalities . Studies suggest that AV could potentially decrease crashes by up to 90%, translating to an estimated annual saving of $190 billion in the United States [2] [3]. Advanced Driver-Assistance Systems (ADAS), integral components of AV technology, have been projected to prevent about 62% of total traffic deaths, equating to approximately 20,841 lives saved per year [4]. Moreover, real-world data indicates that autonomous driving systems can significantly lower crash rates. For instance, Waymo's autonomous vehicles have demonstrated an 85% reduction in injury-related crashes compared to human drivers [5]. These statistics underscore the transformative potential of AVs in mitigating collisions and enhancing overall road safety.

AVs have rapidly evolved in recent years, driven by advancements in sensing technologies, computational power, and control algorithms. These vehicles rely on a combination of perception, planning, and control modules as depicted in Figure 1-1 to navigate complex environments while ensuring safety and efficiency. Among these modules, the control system plays a crucial role in executing planned trajectories with high precision, maintaining stability, and reacting dynamically to uncertainties in the environment.

One of the most challenging aspects of AV is achieving robust and high-performance path following, particularly in scenarios that involve high-speed manoeuvres, obstacle avoidance, and dynamically changing road conditions. At the core of this challenge lies the ability to generate and follow control-friendly reference trajectories that adhere to vehicle dynamics while ensuring safety constraints are met. Standard motion planning approaches typically generate trajectories in a decoupled manner, relying on simplified vehicle models or idealized assumptions about system behaviour which often need complex solutions to ensure feasible trajectories [6] [7] [8]. However, these approaches still fail to account for the intricate non-linearities of vehicle dynamics, leading to suboptimal tracking performance and a lack of adaptability when confronted with real-world uncertainties.

To bridge this gap, model-based control techniques, particularly Model Predictive Control (MPC), have gained prominence in the field of autonomous vehicle motion control. MPC offers significant advantages due to its predictive capabilities and inherent ability to enforce system constraints during optimisation. However, despite its widespread research, classical MPC performance remains highly dependent on the accuracy of the underlying vehicle model and the quality of the reference trajectory it receives. Any mismatch between the model and the real system, whether due to tyre-road interactions, external disturbances, or un-modeled dynamics, can lead to degraded tracking performance and potential safety concerns.

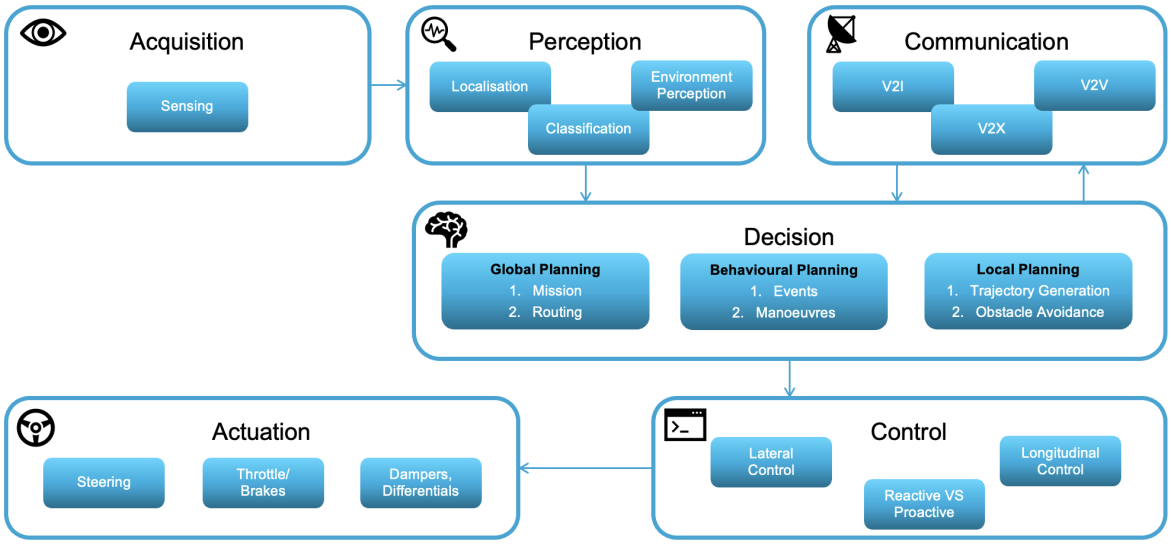


Figure 1-1: General control architecture for Autonomous Driving system adapted from [9]

To address these limitations, data-driven methodologies have emerged as powerful tools to enhance model accuracy and improve controller performance. By leveraging real-world driving data, machine learning techniques and optimisation-based strategies can refine vehicle models, providing more accurate predictions of system behaviour. In particular, data-driven approaches have shown promise in improving the accuracy of tyre models, a crucial component in autonomous vehicle dynamics, especially under varying road conditions. However, while these approaches improve model accuracy, they do not directly address the need for generating reference signals that are more suitable for the controller itself. Thus, an alternative approach is required to reshape reference trajectories in a manner that optimally balances tracking performance, dynamic feasibility, and safety.

1-2 Vehicle Motion Control

Ensuring accurate path following is a fundamental requirement for autonomous vehicles, particularly when operating at high speeds or in dynamically uncertain environments. Traditional control techniques such as Proportional-Integral-Derivative (PID) control and Linear Quadratic Regulator (LQR) have been employed in early autonomous systems, but these methods struggle to handle complex constraints and non-linear vehicle dynamics effectively [10] [11] [12]. In contrast, MPC has emerged as the preferred approach for motion control due to its predictive nature, optimisation-based formulation, and ability to explicitly incorporate constraints on states and control inputs.

MPC operates by solving an optimisation problem at each time step, predicting the system's future behaviour over a finite time horizon and determining the optimal control inputs that minimize a predefined cost function. This framework allows MPC to proactively adjust control actions based on predicted future states rather than reacting to past errors, resulting in superior tracking performance compared to traditional controllers. Additionally, the ability to integrate state and input constraints enables MPC to ensure vehicle stability while maintaining adherence to safety limits, making it particularly suitable for high-performance autonomous driving applications.

Despite its advantages, MPC still faces significant challenges when deployed in real-world autonomous driving scenarios. One of the primary concerns is the dependency on an accurate vehicle model. Any discrepancy between the modelled dynamics and the real system can introduce errors in the predicted trajectories, leading to suboptimal control actions. While data-driven modelling techniques offer potential improvements, these methods require extensive data collection and may still fail to generalize across different driving conditions. Furthermore, the reference trajectory provided to the MPC controller plays a crucial role in determining the overall system performance. A poorly chosen reference signal may result in aggressive control actions, unnecessary oscillations, or even infeasibility in terms of system constraints.

Another major challenge in motion control is obstacle avoidance, particularly when dealing with planner-unseen obstacles. Conventional approaches for trajectory generation and obstacle avoidance often rely on global motion planners that operate at a higher level, providing predefined paths that the controller is expected to follow. However, these global planners typically do not account for real-time vehicle dynamics or rapid environmental changes, leading to potential conflicts between the planned path and the controller's ability to execute it. Consequently, there is a need for a more integrated approach where the reference trajectory itself is dynamically adjusted based on both the vehicle's control capabilities and obstacle avoidance requirements.

One promising avenue for addressing these challenges is the integration of a Reference Governor (RG) within the motion control framework. RGs are designed to modify the reference trajectory dynamically, ensuring that the controlled system operates within feasible limits. Classical RGs primarily focus on constraint satisfaction, preventing violations of system limitations rather than actively improving performance. However, by extending the capabilities of RGs through data-driven optimisation methods, it becomes possible to enhance both tracking performance and obstacle avoidance capabilities.

This research proposes a novel approach that integrates Bayesian Optimisation (BO) into the

RG framework, effectively generating an optimised reference trajectory tailored to both the MPC controller's needs and the presence of previously unaccounted obstacles. By leveraging BO ability to iteratively refine control parameters based on performance feedback, the proposed method dynamically adjusts the reference signal to improve tracking accuracy, maintain safety margins, and enhance robustness against model uncertainties. This approach represents a significant advancement over traditional reference governors, offering a model-free, data-driven alternative to conventional model compensation techniques.

1-2-1 Literature Study Review: Reference Governor

RG have been widely explored as a means to enhance control system performance while ensuring constraint satisfaction. By modifying the reference signal before it is processed by the controller, RGs prevent constraint violations and improve system stability without requiring modifications to the primary control law. The literature on RGs can be categorized into three main types: *classical control-based RGs*, *constraint satisfaction and safety RGs*, and *data-driven RGs*.

Classical RGs primarily focus on ensuring that control system constraints are satisfied by shaping the reference input. Common approaches include:

- **Virtual State Governors**: Used in multi-actuator systems to ensure constraint satisfaction across multiple control inputs.
- **Parameter Governors**: Adjust the system parameters dynamically to enhance constraint fulfilment.
- **Explicit Reference Governors**: Shape reference trajectories based on predefined steady-state safety sets.

While effective in enforcing constraints, these methods typically do not optimise tracking performance and rely on predefined system models, making them susceptible to inaccuracies when system parameters change.

To address the limitations of classical RGs, optimisation-based RGs have been developed. These methods use real-time optimisation techniques to dynamically adjust the reference signal while ensuring safe operation.

A common approach is *incremental reference governance*, where reference modifications occur gradually to balance safety and performance. One formulation involves solving an online optimisation problem at each time step to minimize deviation from the original reference while maintaining constraint adherence.

When integrated with MPC, these methods provide an added layer of safety but introduce additional computational complexity. This trade-off between real-time feasibility and constraint satisfaction remains a critical challenge.

Recent advancements in machine learning have led to the emergence of data-driven Reference Governors. These methods leverage real-world data to improve adaptability and robustness beyond model-based approaches. Notable techniques include:

- **Neural Network-Based RGs**: Learn maximal output admissible sets, enabling adaptive constraint enforcement.
- **Gaussian Process (GP)-Based RGs**: Use probabilistic modelling to estimate system uncertainties and modify reference signals accordingly.
- **Deep Reinforcement Learning-based RGs**: utilise Artificial Intelligence (AI) agents to iteratively optimise reference modifications based on system feedback.

These methods offer significant advantages in handling model uncertainties and adapting to changing system dynamics. However, ensuring their stability and computational efficiency in real-time applications remains an open research problem.

1-2-2 Key Takeaways and Research Gaps

- Classical RGs are effective for constraint satisfaction but do not actively improve tracking performance.
- Optimisation-based RGs enhance safety but introduce computational overhead, limiting real-time applicability.
- Data-driven RGs provide a model-free, adaptable approach, making them robust to system uncertainties. However, ensuring stability and computational feasibility is still a challenge.
- Integrating data-driven RGs within MPC frameworks presents a promising research direction, particularly for autonomous vehicle applications where both tracking performance and safety are critical.

1-2-3 Summary: Literature Review Conclusion

The literature suggests a growing trend toward data-driven RGs, which offer improved adaptability and robustness compared to classical approaches. However, their integration into real-time safety-critical systems, such as autonomous vehicle control, requires further investigation. This research aims to leverage *Bayesian Optimisation-based RGs within an MPC framework* to improve tracking performance while ensuring constraint satisfaction, addressing key gaps identified in the existing literature.

1-3 Problem Formulation

The primary objective of this work is to develop an RG that operates in conjunction with a Model Predictive Controller, as illustrated in Figure 1-2. Inspired by [13], the RG is designed to modulate reference trajectories in real time in order to enhance both tracking performance and obstacle avoidance safety. By integrating high-level reference adaptation with low-level optimal control, the overall system aims to improve vehicle behaviour in complex and dynamic driving environments.

The design and implementation of the RG system is structured into two main phases: an offline training phase and an online deployment phase. These are carried out as follows:

- **Offline Training Phase:** In the initial stage, the Reference Governor is trained using Bayesian optimisation techniques in combination with Gaussian Process (GP). This phase involves simulating a wide range of driving scenarios and systematically exploring the space of reference parameters, specifically the sigmoid parameters that define the shape and position of the lateral manoeuvre. The Bayesian optimisation algorithm uses the GP model as a surrogate to minimize a cost function that reflects tracking accuracy, control effort, and obstacle clearance. The result is a rich dataset and a trained GP model that approximates the optimal reference parameter configurations across varying contexts. This methodology builds on recent advances in data-driven control and Bayesian optimisation, as demonstrated in works such as [14, 15, 16] where BO is mainly utilized for controller tuning.
- **Online Deployment Phase:** Once the offline training is completed, the resulting GP model is employed as a lightweight reference augmentation filter in real-time operation. During each control update, the current driving context, such as vehicle speed and the relative location of surrounding obstacles is used as input to the pre-trained GP. The model then predicts the performance of candidate reference trajectories under the current conditions. The Reference Governor selects the best set of reference parameters using an acquisition function (e.g. Expected Improvement) and generates an adapted reference that is fed to the MPC. This allows the vehicle to adjust its behaviour dynamically and efficiently in response to environmental changes, without the computational overhead of re-optimising from scratch. Safety-aware and constrained Bayesian optimisation approaches, such as those proposed in [17], further motivate the integration of constraint handling within the governor to enhance robustness.

By combining data-efficient offline learning with responsive online adaptation, the proposed Reference Governed Model Predictive Control (RG-MPC) framework ensures robust control performance in uncertain environments. It enables the system to generalize from previously encountered situations while remaining flexible to new and evolving contexts, aligning with the performance-driven and safety-aware control paradigms discussed in [13, 16].

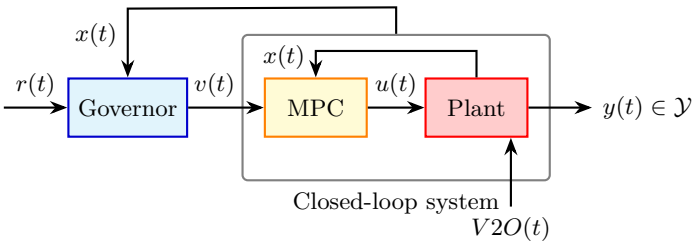


Figure 1-2: Block diagram of the proposed control system with governor, controller, and plant.

Chapter 2

Model Predictive Control Design

In this chapter, the theoretical foundations of optimal control will be explored in the context of Model Predictive Control (MPC) for vehicle motion control. The formulation of the optimal control problem is crucial for ensuring safe, efficient, and constrained trajectory planning in autonomous driving applications. By leveraging a predictive model, MPC enables real-time optimisation of control actions while adhering to system dynamics and physical limitations. The discussion will cover the mathematical representation of the control problem, the role of cost functions, constraints, and optimisation methods to provide a structured approach to motion planning in automated vehicles.

2-1 Model Predictive Control

MPC is an optimal control technique which computes a control action based on an optimisation problem via an evaluation of a predictive model. The advantage of the MPC scheme is that it is relatively simple in architecture, but can provide high performance due to control optimisation at every sampled time instance for a horizon into the future. The downside of MPC is that the tuning is heavily dependant on the efforts of the human designer as well as on the accuracy of the predictive model.

The MPC scheme consists of the following four core components, where the overarching goal is to control a plant via an optimal control action under constraints:

1. Prediction model
2. Cost function
3. Constraints
4. Optimisation

The MPC controller is formulated in discrete time for the following Linear Time Invariant (LTI) system dynamics and constraints where x and u represent the sate and input dynamics respectively. The subscript lower case 'm' denotes a minimum bound and subscript upper case 'M' denotes the upper bound.

$$\text{System dynamics: } \begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) \end{cases} \tag{2-1}$$

$$\text{Constraints: } \begin{cases} x_m \leq x(k) \leq x_M \\ u_m \leq u(k) \leqslant u_M \end{cases} \quad \forall k \in \{0, 1, \dots N\} \tag{2-2}$$

Via recursion of the time evolution of k, the system states can be represented in terms of initial states and optimised input sequence $u(0), ..., u(N_p - 1)$ where N_p is the prediction horizon:

$$\begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ x(N_p) \end{bmatrix} = \underbrace{\begin{bmatrix} A^0 \\ A^1 \\ A^2 \\ \vdots \\ A^N \end{bmatrix}}_{T \in \mathbb{R}^{n(N_p+1) \times n}} x_0 + \underbrace{\begin{bmatrix} 0 & 0 & \cdots & 0 \\ B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^N B & A^{N-2}B & \cdots & B \end{bmatrix}}_{S \in \mathbb{R}^{n(N_p+1) \times N_p}} \begin{bmatrix} u(0) \\ u(1) \\ u(2) \\ \vdots \\ u(N-1) \end{bmatrix} \tag{2-3}$$

The state constraints can be then formulated in terms of the newly defined T and S as:

$$\begin{bmatrix} x_m \\ x_m \\ x_m \\ \vdots \\ x_m \end{bmatrix} - Tx_0 \leqslant S \begin{bmatrix} u(0) \\ u(1) \\ u(2) \\ \vdots \\ u(N-1) \end{bmatrix} \leq \begin{bmatrix} x_M \\ x_M \\ x_M \\ \vdots \\ x_M \end{bmatrix} - Tx_0 \tag{2-4}$$

The input constraints follow from the original inequalities:

$$\begin{bmatrix} u_m \\ u_m \\ u_m \\ \vdots \\ u_\mu \end{bmatrix} \leq \begin{bmatrix} u(0) \\ u(1) \\ u(2) \\ \vdots \\ u(N_p - 1) \end{bmatrix} \leq \begin{bmatrix} u_M \\ u_M \\ u_m \\ \vdots \\ u_M \end{bmatrix} \iff \begin{pmatrix} I \\ -I \end{pmatrix} \bar{u} = \begin{pmatrix} u_M \\ -U_m \end{pmatrix} \tag{2-5}$$

Combining Equations 2-4 and 2-5 yields the following constraint in standard form:

$$\underbrace{\begin{bmatrix} S \\ -S \\ I \\ -I \end{bmatrix}}_{A} \bar{u} \leq \underbrace{\begin{bmatrix} x_M - Tx_0 \\ Tx_0 - x_m \\ u_M \\ -u_m \end{bmatrix}}_{b} \tag{2-6}$$

The cost function required by the MPC formulation over a prediction horizon N_p is expressed as:

$$J(u(k)) = \frac{1}{2}\sum_{k=0}^{N-1}\left[x_k^\top Q x_k + u_k^\top R \cdot u_k\right] + \frac{1}{2}x_N^\top P x_N \tag{2-7}$$

Where x_k, u_k are the predicted state and input at time step k respectively. Matrix Q is the weight matrix which influences the control strategy's emphasis on minimising deviations in the predicted system states from the desired or reference states. It determines the relative importance of the state variables in the control objective. The weighting matrix R in the cost function of an MPC controller influences the control strategy's emphasis on minimising the magnitude or effort of the control inputs. It determines the relative importance of the control inputs in the control objective. Lastly P is the term associated with the terminal cost at the end of the prediction horizon. After simplification the cost function can be written as:

$$J(u(k)) = \frac{1}{2}\bar{u}^\top H \bar{u} + h\bar{u} + c. \tag{2-8}$$

Where the cost function matrices are defined to be:

$$\begin{aligned} H &= \bar{R} + S^T \bar{Q} S \\ h &= S^T \bar{Q} T x_0 \\ c &= x_0^T T^T \bar{Q} T x_0 \\ \bar{Q} &= \begin{pmatrix} I_{N\times N} \otimes Q & 0 \\ 0 & 0 \end{pmatrix} \\ \bar{R} &= T_{N\times N} \otimes R \end{aligned} \tag{2-9}$$

2-2 Predictive Models

2-2-1 Vehicle Predictive Model

The dynamic behaviour of the vehicle is captured using the planar bicycle model, a commonly used representation for modelling lateral and longitudinal dynamics in a 2D driving environment. This model simplifies the vehicle by aggregating the effects of two front wheels into one and two rear wheels into one, reducing the complexity while maintaining essential dynamic characteristics for controller design and prediction.

The model includes seven key state variables: longitudinal velocity (v_x), lateral velocity (v_y), yaw rate (r), yaw angle (ψ), vehicle global position coordinates (x_p, y_p), and the steering angle (δ). These variables are crucial in describing the full motion of the vehicle on a flat plane, assuming small roll and pitch effects. The vehicle receives control input via steering angle rate d_δ and longitudinal forces generated through braking or propulsion.

The lateral tyre forces, which dominate during manoeuvres like turning or lane changes, are modelled using a linear tyre assumption:

$$F_{yf} = C_{\alpha_f}\alpha_f, \quad F_{yr} = C_{\alpha_r}\alpha_r \tag{2-10}$$

where α_f and α_r are the slip angles for the front and rear tyres, respectively, and C_{α_f}, C_{α_r} are the corresponding cornering stiffness coefficients.

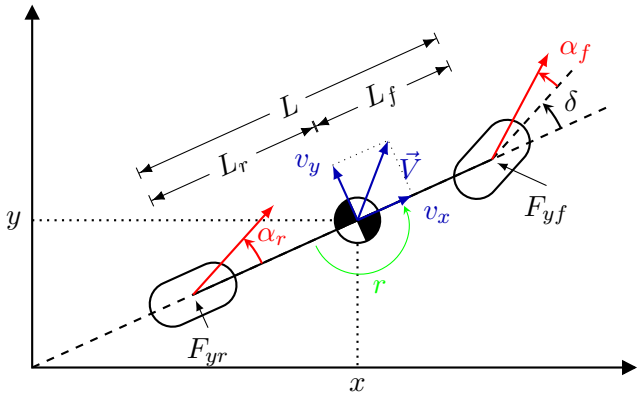


Figure 2-1: Bicycle Model Free Body Diagram.

The vehicle's motion is governed by the following set of non-linear differential equations:

$$\begin{aligned}
\dot{v}_x &= v_y r \\
\dot{v}_y &= -\left(\frac{C_{\alpha_f} + C_{\alpha_r}}{m v_x}\right) v_y + \left(\frac{L_r C_{\alpha_r} - L_f C_{\alpha_f}}{m v_x} - v_x\right) r + \frac{C_{\alpha_f}}{m}\delta \\
\dot{r} &= \left(\frac{L_r C_{\alpha_r} - L_f C_{\alpha_f}}{I_z v_x}\right) v_y - \left(\frac{L_r^2 C_{\alpha_r} + L_f^2 C_{\alpha_f}}{I_z v_x}\right) r + \frac{L_f C_{\alpha_f}}{I_z}\delta \\
\dot{\psi} &= r \\
\dot{x}_p &= v_x \cos(\psi) - v_y \sin(\psi) \\
\dot{y}_p &= v_x \sin(\psi) + v_y \cos(\psi) \\
\dot{\delta} &= d_\delta
\end{aligned} \tag{2-11}$$

Tyre Force modelling : Magic Formula vs. Dugoff Model

While linear tyre models are efficient and suitable for small-angle manoeuvres, they lack accuracy in high slip conditions or aggressive manoeuvres. Thus, more detailed models like the Magic Formula [18] and Dugoff model [19] provide non-linear representations of tyre dynamics, critical for improving prediction fidelity in these scenarios.

The Magic Formula, also known as the Pacejka model, provides an empirical fit to tyre force data:

$$F_y = D \sin\left(C \arctan\left(B\alpha - E\left(B\alpha - \arctan(B\alpha)\right)\right)\right) \tag{2-12}$$

where:

- α is the slip angle
- B, C, D, and E are fitting parameters (stiffness, shape, peak, curvature)

The cornering stiffness is approximated in the small-slip region as the slope of the Magic Formula at $\alpha = 0$:

$$C_\alpha = \left.\frac{dF_y}{d\alpha}\right|_{\alpha=0} \tag{2-13}$$

An alternative, physics-based approach is the Dugoff tyre Model, which relates lateral force to normal load, tyre stiffness, and road friction:

$$F_y = \begin{cases} \mu F_z \left(\frac{C_\alpha \alpha}{2\mu F_z}\left(2 - \frac{C_\alpha \alpha}{\mu F_z}\right)\right), & \text{if } \frac{C_\alpha |\alpha|}{\mu F_z} < 1 \\ \mu F_z \cdot \text{sgn}(\alpha), & \text{otherwise} \end{cases} \tag{2-14}$$

The Dugoff model accounts for combined slip and saturation effects, providing a smooth transition between linear and non-linear force regimes. It is particularly suitable when estimating realistic tyre behaviour near the friction limit.

The cornering stiffness values used in the bicycle model are often derived from fitting either of these models to experimental or simulated tyre data in the low-slip range, offering a trade-off between accuracy and computational efficiency.

2-2-2 Optimal Control Problem

The dynamics described above form the basis for a predictive control problem. Using an (MPC) framework, the vehicle is guided along a desired trajectory while avoiding obstacles and remaining within physical limits.

The objective is to minimise the following cost function:

$$\begin{aligned} J = \sum_{i=1}^{N} \Big(& q_{e_{Yp}} e_{Yp,i}^2 + q_{e_{Xp}} e_{Xp,i}^2 + q_{\dot{\delta}} \dot{\delta}_i^2 + q_{\dot{F}_x} \dot{F}_{x_i}^2 + ... \\ & + \sum_{j=1}^{N_{obs}} \left(q_{e_{V2O}} e_{V2O,j,i}^2 \right) \Big) \end{aligned} \tag{2-15}$$

The individual cost terms are defined to prioritize tracking accuracy, smooth actuation, and obstacle avoidance, with respective weight parameters.

The obstacle distance error is defined as:

$$D_{V2O} = \sqrt{(X - X_{obs})^2 + (Y - Y_{obs})^2} - r_{obs} - r_{veh} \tag{2-16}$$

This ensures a buffer between vehicle and obstacle is maintained throughout the planning horizon.

Constraints include:

$$\begin{aligned} 0 \leq v_x &\leq \frac{170}{3.6} \\ -\frac{5\pi}{180} \leq \frac{v_y}{v_x} &\leq \frac{5\pi}{180} \\ -\frac{25\pi}{180} \leq \frac{\dot{v}_y}{v_x} &\leq \frac{25\pi}{180} \\ -0.85\mu g \leq \dot{v}_y + v_x r &\leq 0.85\mu g \\ \frac{-2.76 \cdot 360 \cdot \pi}{180 \cdot st_{\text{ratio}}} \leq \delta &\leq \frac{2.76 \cdot 360 \cdot \pi}{180 \cdot st_{\text{ratio}}} \\ \frac{-800 \cdot \pi}{180 \cdot st_{\text{ratio}}} \leq \dot{\delta} &\leq \frac{800 \cdot \pi}{180 \cdot st_{\text{ratio}}} \end{aligned} \tag{2-17}$$

These enforce physical limitations on vehicle operation in terms of longitudinal velocity, vehicle body-slip angle and rate, lateral acceleration and steer actuation.

2-3 Non-linear Framework

2-3-1 Overview of ACADO Toolkit

To implement non-linear MPC, the ACADO Toolkit is employed as the core optimisation engine. ACADO is an open-source software suite tailored for solving Optimal Control Problem (OCP)s, with real-time capabilities through automatic code generation. It has been used in embedded control applications due to its efficient C code export, compatibility with MATLAB/Simulink, and support for non-linear systems, parameter estimation, and real-time iteration schemes [20].

ACADO adopts a two-phase approach to solving OCPs:

- **Preparation Phase**: Involves discretization, linearization, and condensing of the original non-linear OCP.
- **Feedback Phase**: Solves the resulting Gaussian Process (GP) using an embedded Quadratic Programming (QP) solver (qpOASES) based on the Active Set Method.

This methodology enables fast computation of control actions by performing only one Sequential Quadratic Programming (SQP) iteration per sample, a strategy referred to as Real-time Iteration (RTI) [21].

2-3-2 Sequential Quadratic Programming

SQP is the principal algorithm used within ACADO to handle non-linear constrained optimisation problems [22]. At each iteration, the non-linear OCP is approximated as a QP around the current estimate of the solution, which is then solved to produce a new estimate. The iteration continues until convergence or a single step is taken in the RTI context.

Formulation

Consider a general non-linear optimisation problem:

$$\begin{aligned} \underset{x\in\mathbb{R}^n}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & h(x) = 0, \quad g(x) \leq 0 \end{aligned} \tag{2-18}$$

where:

- $f(x)$ is the objective function.
- $h(x)$ and $g(x)$ are the equality and inequality constraint functions, respectively.

To apply SQP, a Lagrangian function is defined:

$$L(x, \lambda, \mu) = f(x) + \lambda^\top h(x) + \mu^\top g(x) \tag{2-19}$$

with Lagrange multipliers λ and μ for the constraints. The Karush-Kuhn-Tucker (KKT) optimality conditions are then derived as:

$$\nabla_x L(x, \lambda, \mu) = 0 \tag{2-20}$$

Quadratic Approximation

The Lagrangian is locally approximated using a second-order Taylor expansion around x_k:

$$\begin{aligned} L(x, \lambda_k, \mu_k) \approx\ & L(x_k) + \nabla_x L(x_k)^\top (x - x_k) \\ & + \frac{1}{2}(x - x_k)^\top H_L(x_k)(x - x_k) \end{aligned} \tag{2-21}$$

where H_L is the Hessian of the Lagrangian. Equality and inequality constraints are linearized as:

$$h(x) \approx h(x_k) + \nabla h(x_k)(x - x_k), \quad g(x) \approx g(x_k) + \nabla g(x_k)(x - x_k) \tag{2-22}$$

Letting $d = x - x_k$, we obtain a QP sub-problem of the form:

$$\begin{aligned} \underset{d}{\text{minimize}} \quad & \nabla_x L(x_k)^\top d + \frac{1}{2} d^\top H_L(x_k) d \\ \text{subject to} \quad & \nabla h(x_k) d + h(x_k) = 0 \\ & \nabla g(x_k) d + g(x_k) \leq 0 \end{aligned} \tag{2-23}$$

Solving this QP yields a search direction d_k, and the next iterate is given by:

$$x_{k+1} = x_k + \alpha_k d_k \tag{2-24}$$

with step size α_k chosen through a line search.

2-3-3 Active Set Method (ASM)

To solve the condensed QP efficiently, ACADO uses Active Set Point Method (ASM) [23] via the qpOASES solver [24]. ASM is well-suited for embedded applications due to its deterministic behaviour and rapid convergence for small-scale QPs.

qpOASES intergation mechanism

The ASM maintains an estimate of the set of active constraints (equalities and inequalities treated as equalities). At each iteration, the following equality-constrained QP is solved:

$$\begin{aligned} \underset{d}{\text{minimize}} \quad & \frac{1}{2} d^\top H d + g^\top d \\ \text{subject to} \quad & A_{\text{eq}} d = 0, \quad A_{\text{active}} d = 0 \end{aligned} \tag{2-25}$$

Here, A_{active} corresponds to the gradient of active inequality constraints. The set of active constraints is updated by checking feasibility and optimality conditions at each iteration.

qpOASES is integrated in ACADO for solving such QPs using ASM. It supports warm-starts and handles hot-started initial guesses efficiently, which is ideal for real-time applications like MPC.

2-3-4 Handling Infeasibility

In practice, NMPC formulations can become infeasible due to:

- Poor initial conditions violating constraints.
- Linearization or condensing errors that make the QP infeasible or unbounded.

ACADO manages infeasibility via status flags. A return code of 0 indicates success, while 1 means a maximum number of iterations was reached and the last feasible iterate is returned. Codes like -1 or -2 suggest failure, in which case the last known feasible control input is used. It is to be noted that these flags were instrumental during the implementation phase.

This fallback mechanism, along with warm-starting in subsequent steps, ensures a degree of robustness during real-time execution of MPC.

2-4 Summary

This chapter presented the formulation and design of a MPC framework tailored for automated vehicle motion planning. Theoretical foundations of optimal control were laid out, beginning with a discrete-time Linear LTI formulation incorporating system dynamics, state/input constraints, and a quadratic cost function. The standard structure of MPC was introduced, highlighting the roles of the prediction model, cost function, constraints, and optimisation strategy.

A non-linear bicycle model was adopted to capture the vehicle's dynamics, considering lateral and longitudinal behaviour along with steering dynamics. Linear and non-linear tyre models were discussed, including the Magic Formula and Dugoff model, to illustrate how tyre forces are incorporated into predictive models. These models serve to improve prediction fidelity in both low and high slip scenarios.

The chapter further introduced the formulation of the non-linear optimal control problem within the MPC framework, emphasizing tracking, smoothness of control, and obstacle avoidance. Practical constraints on vehicle states and actuators were included to ensure safety and feasibility.

To implement nonlinear MPC, the ACADO Toolkit was employed. Its two-phase structure, composed of a preparation and feedback phase, allows efficient real-time computation using a RTI scheme. The underlying optimisation algorithm, SQP, was detailed, along with its approximation via Quadratic Programs and the use of ASM through the qpOASES solver. Lastly, mechanisms to handle infeasibility were explained, ensuring robustness in real-world implementations.

This comprehensive MPC design enables predictive, constraint-aware, and real-time vehicle control, forming a core part of autonomous driving strategies.

Chapter 3

Bayesian Optimisation

3-1 Black-box Global Optimisation

Black-box optimisation commonly refers to the optimisation of objective functions where the internal structure is unknown or inaccessible. These methods are widely used in machine learning, engineering design, and scientific simulations, where evaluating the function is costly or lacks analytical gradients [25] [26]. In relation to control, BO has been developed for advanced controller tuning for cascade controlled linear axis drive systems [15], [16] and [17] as well as heat pumps [14]. Various strategies exist for optimising black-box functions, including evolutionary algorithms, Bayesian optimisation , and direct search methods [27, 28].

Evolutionary Algorithms (EA)s simulate natural selection to explore the search space iteratively. Examples include Genetic Algorithms (GA) and Covariance Matrix Adaptation Evolution Strategy (CMAES) [30]. These methods are highly parallelise-able and robust but often require a large number of function evaluations.

Direct search methods, such as Nelder-Mead and pattern search, iteratively refine solutions without requiring gradients [31]. They are simple to implement but may converge slowly or get trapped in local optima.

Each method has trade-offs:

- **Evolutionary Algorithms:** Highly explorative but computationally expensive.
- **Bayesian Optimisation :** Sample-efficient but limited in scalability.
- **Direct Search Methods:** Simple and effective for low-dimensional problems but inefficient for large-scale optimisation .

Bayesian Optimisation (BO) builds a probabilistic surrogate model, typically using Gaussian processes, to guide the search efficiently [32]. BO is effective for expensive-to-evaluate functions but struggles in high-dimensional spaces. Lastly, global optimisation methods exist for a variety of surrogate models and variable type [33],[34].

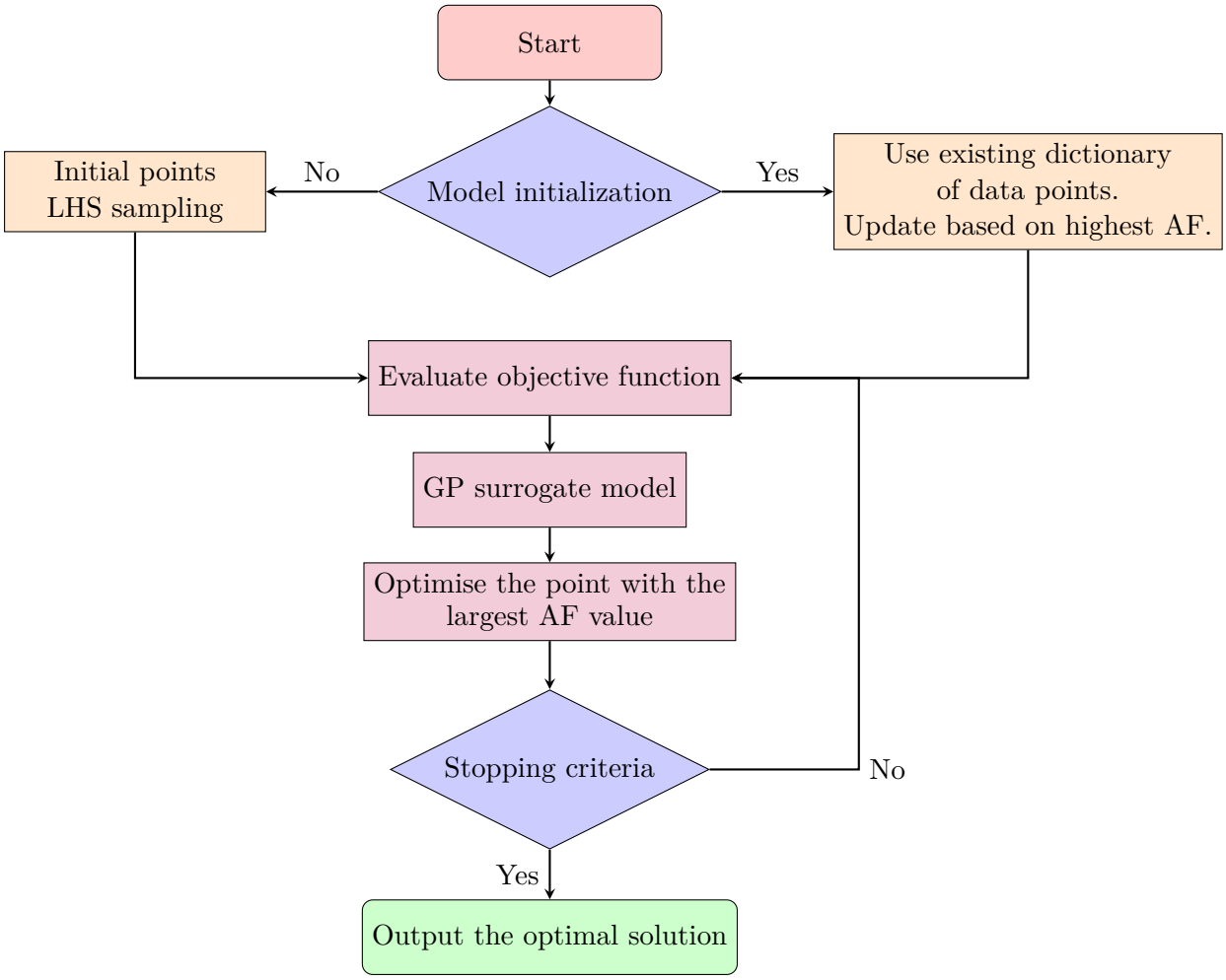


Figure 3-1: Flowchart of the Bayesian optimisation procedure using Gaussian Process (GP) surrogate model and acquisition function (AF) as adapted from [29].

Generally, selecting an appropriate optimisation method depends on factors such as dimensionality, evaluation cost, and the presence of noise. However, the proposed Reference Governor solution requires online evaluation parallel to controller computation. This system requirement means that actually performing the optimisation online is extremely costly and thus unrealisable. Therefore, the only feasible black-box optimisation candidate is BO, due to the utilisation of a Gaussian Process model of the target objective function and continuos variable type.

3-2 Bayesian Optimisation Problem Formulation

BO is an approach for global optimisation of *black-box* functions that are expensive to evaluate [35], [36]. We consider the problem of finding the optimum:

$$x^* = \arg\min_{x \in \mathcal{X}} f(x), \tag{3-1}$$

where $f(x)$ is a continuous objective function, and $\mathcal{X}$ is a feasible domain. BO builds a probabilistic *surrogate model* of f, using it to intelligently select promising evaluation points. Unlike classical methods, BO does not require gradients and is effective for problems where evaluations are costly.

3-2-1 Gaussian Process Surrogate Model

BO typically models $f(x)$ using a GP prior [37]. A GP is defined as:

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')). \tag{3-2}$$

The posterior mean and variance of $f(x)$ after observing n data points are given by:

$$\mu_n(x) = m(x) + k(x, X)[K(X, X) + \sigma^2_{\text{noise}} I]^{-1}(y - m(X)), \tag{3-3}$$
$$\sigma^2_n(x) = k(x, x) - k(x, X)[K(X, X) + \sigma^2_{\text{noise}} I]^{-1} k(X, x). \tag{3-4}$$

Where each term in these equations is defined as:

- $\mu_n(x)$: The posterior mean, which is the best estimate of the function value at x given the observations.
- $\sigma^2_n(x)$: The posterior variance, representing the uncertainty of the prediction at x.
- $m(x)$: The prior mean function, which is often set to zero for simplicity but can encode prior knowledge about the function.
- $k(x, X)$: The covariance vector between the test point x and all previously observed points X, where each entry is given by the kernel function $k(x, x_i)$.
- $K(X, X)$: The covariance matrix of the observed points, computed using the kernel function.

- σ^2_{noise}: The variance of the observation noise, accounting for measurement errors in function evaluations.
- I: The identity matrix, ensuring numerical stability when inverting the covariance matrix.
- y: The vector of observed function values corresponding to the training points X.

The term $[K(X, X) + \sigma^2_{\text{noise}} I]^{-1}$ serves as a correction factor that adjusts predictions based on observed data. This term accounts for dependencies among the observations and corrects for noise. The kernel function $k(x, x')$ is crucial, as it defines the smoothness and correlation structure of the model, significantly influencing how new points are predicted.

Further explanation regarding the derivations of the Gaussian Processes as well as kernel selection and hyper-parameters optimisation are provided below.

3-2-2 Acquisition Functions

Upper Confidence Bound

One of the simplest forms of acquisition functions is the Upper Confidence Bound (UCB), which explicitly balances exploitation and exploration through the formula:

$$\alpha_{\text{UCB}}(x) = \mu_n(x) - \kappa \sigma_n(x). \tag{3-5}$$

In this formulation, $\mu_n(x)$ represents the predicted mean of the GP, guiding exploitation, while $\sigma_n(x)$ denotes the standard deviation, encouraging exploration. The trade-off between these two objectives is directly controlled by the parameter κ. A small κ emphasizes exploitation by prioritizing regions with high predicted performance. In contrast, a large κ promotes exploration by steering the search toward areas with greater uncertainty. For instance, setting κ to a high value leads the UCB to favour regions of the search space that remain unexplored due to a lack of prior evaluations [38], [39].

To formalize the notion of achieving a better solution during optimisation , we can define the *improvement* at a candidate point x as the positive gain over the current best observation $x^\star$, that is:

$$I(x) = \max(f(x) - f(x^\star), 0).$$

This expression ensures that only beneficial updates are considered, if the new evaluation $f(x)$ fails to surpass the current best value $f(x^\star)$, the improvement is zero. Conversely, when $f(x)$ exceeds the current best, the improvement is equal to the margin of the increase.

Probability of Improvement

An extension of UCB is to consider information about how much improvement the next evaluated point will yield [39]. The Probability of Improvement (PI) acquisition strategy leverages this idea by quantifying the likelihood that a proposed input x leads to an improvement. Under the GP model, each point x is associated with a normal distribution $\mathcal{N}(\mu(x), \sigma^2(x))$,

representing the surrogate model's uncertainty. Exploiting this, we can express the random variable $f(x)$ using a standard normal variable $z \sim \mathcal{N}(0,1)$ via the transformation:

$$f(x) = \mu(x) + \sigma(x)z.$$

Rewriting the improvement using this transformation yields:

$$I(x) = \max(\mu(x) + \sigma(x)z - f(x^\star), 0).$$

The PI acquisition function computes the probability that this improvement is strictly positive, i.e.,

$$\text{PI}(x) = \Pr(f(x) > f(x^\star)).$$

This corresponds to the tail probability of the normal distribution exceeding the threshold $f(x^\star)$. Mathematically, it becomes:

$$\text{PI}(x) = \Phi\left(\frac{\mu(x) - f(x^\star)}{\sigma(x)}\right),$$

where $\Phi(\cdot)$ denotes the cumulative distribution function (CDF) of the standard normal distribution.

Expected Improvement

While the PI acquisition function estimates the chance of obtaining a better solution than the current best, it does not take into account how substantial that improvement might be. The Expected Improvement (EI) criterion addresses this by considering the average amount of improvement over all possible outcomes. Specifically, it evaluates the expected value of the improvement function $I(x)$:

$$\text{EI}(x) = \mathbb{E}[I(x)] = \int_{-\infty}^{\infty} I(x)\,\phi(z)\,dz,$$

where $\phi(z)$ denotes the probability density function of the standard normal distribution, i.e., $\phi(z) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right)$.

The GP surrogate model again allows for $f(x)$ to be modelled as a normal random variable with mean $\mu(x)$ and standard deviation $\sigma(x)$, such that:

$$f(x) = \mu(x) + \sigma(x)z, \quad z \sim \mathcal{N}(0,1).$$

The improvement is then expressed as:

$$I(x) = \max(\mu(x) + \sigma(x)z - f(x^\star), 0).$$

To evaluate the expectation, we split the integral at the threshold $z_0 = \frac{f(x^\star)-\mu(x)}{\sigma(x)}$, where the argument of the max function transitions from zero to positive:

$$\text{EI}(x) = \int_{z_0}^{\infty} (\mu(x) + \sigma(x)z - f(x^\star))\,\phi(z)\,dz.$$

This can be simplified into a closed-form expression:

$$\mathrm{EI}(x) = (\mu(x) - f(x^\star))\Phi\left(\frac{\mu(x) - f(x^\star)}{\sigma(x)}\right) + \sigma(x)\phi\left(\frac{\mu(x) - f(x^\star)}{\sigma(x)}\right),$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ are the Cumulative Distribution Function (CDF) and Probability Density Function (PDF) of the standard normal distribution, respectively.

The EI tends to be large in two cases: when the predicted mean $\mu(x)$ significantly exceeds the current best $f(x^\star)$ (indicating a potentially better solution), and when the uncertainty $\sigma(x)$ is high (allowing for the possibility of discovering unexplored optima).

To allow for a tunable exploration-exploitation balance, a non-negative parameter ξ can be introduced, yielding the generalised EI formulation:

$$\mathrm{EI}(x;\xi) = (\mu(x) - f(x^\star) - \xi)\Phi\left(\frac{\mu(x) - f(x^\star) - \xi}{\sigma(x)}\right) + \sigma(x)\phi\left(\frac{\mu(x) - f(x^\star) - \xi}{\sigma(x)}\right).$$

When $\xi = 0$, the original EI expression is recovered. Larger values of ξ raise the improvement threshold, encouraging the search to prioritise more uncertain or less-explored regions of the input space.

To summarize, the resulting EI form used in this study follows as:

$$\alpha_{\mathrm{EI}}(x) = \mathbb{E}[\max(0, f_{\min} - f(x))]. \tag{3-6}$$

Using properties of Gaussian distributions, it simplifies to:

$$\alpha_{\mathrm{EI}}(x) = (f_{\min} - \mu_n(x))\Phi(Z) + \sigma_n(x)\phi(Z), \tag{3-7}$$

where $Z = \frac{f_{\min} - \mu_n(x)}{\sigma_n(x)}$ and Φ, ϕ are the CDF and PDF of the standard normal distribution.

Figure 3-2 illustrates the GP surrogate model trained during BO to minimize a noisy, multi-modal objective function. The blue line represents the GP's posterior mean, while the shaded region indicates variance, as a direct result of underlying model uncertainty. Red dots denote the evaluated sample points, and the green marker highlights the best solution found. The magenta dashed line shows the Expected Improvement Plus (EI^+) acquisition function, guiding exploration toward promising regions. The objective function exhibits multiple valleys and mild noise, making it a non-trivial optimisation problem. Bayesian Optimisation successfully identifies the global minimum with relatively few evaluations.

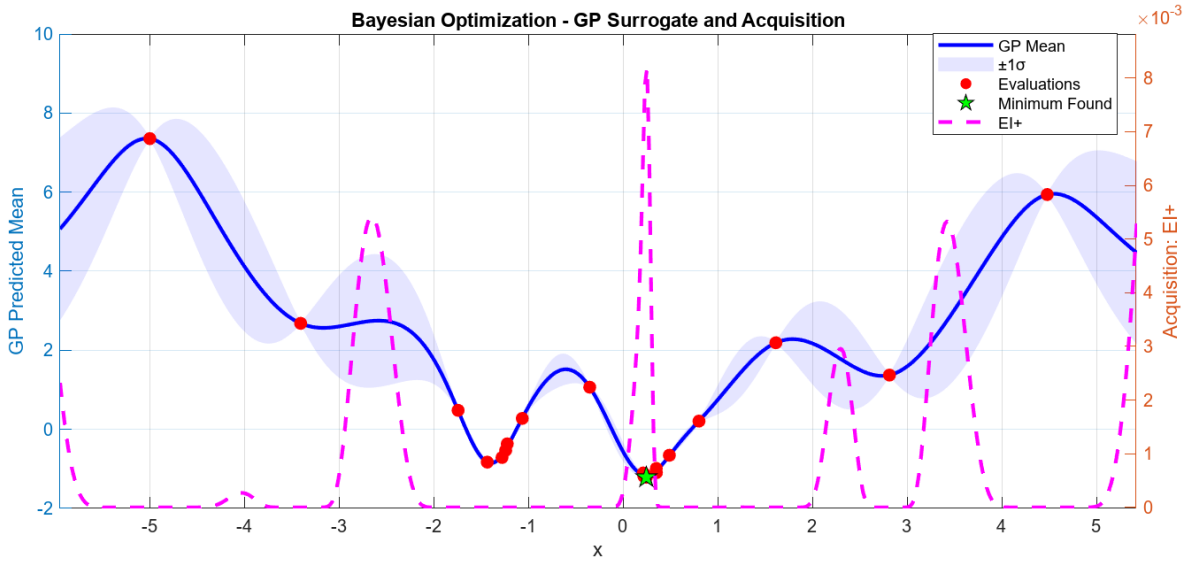


Figure 3-2: Bayesian optimisation process used to find the global minimum of an unknown noisy parabolic function.

3-2-3 Maximisation of Acquisition Functions

Since the acquisition function $\alpha(x)$ is typically multimodal and non-convex, finding its maximum $x^* = \arg\max \alpha(x)$ is a challenging optimisation problem. Several methods are commonly used for this task [40]:

- **Gradient-based methods**: These rely on differentiability of $\alpha(x)$ and employ first-order or second-order optimisation methods, such as L-BFGS or Adam.
- **Grid Search**: A coarse evaluation of $\alpha(x)$ over a discrete grid, refining around promising regions.
- **Genetic Algorithm (GA)**: A population-based evolutionary method that iteratively improves candidate solutions using selection, crossover, and mutation.
- **Particle Swarm Optimisation (PSO)**: A swarm-based meta-heuristic where particles explore the space, communicating findings to improve convergence.

Genetic Algorithm (GA)

GA is a nature-inspired optimisation method based on the principles of evolution [41]. It maintains a population of candidate solutions $\{x_1, \ldots, x_N\}$, evolving them over generations via selection, crossover, and mutation.

1. **Selection**: At each generation, individuals in the population are evaluated using a fitness function, which in the case of Bayesian optimisation is the acquisition function $\alpha(x)$. Selection is often performed using methods such as roulette wheel selection or tournament

selection. In roulette wheel selection, the probability of selecting an individual x_i is proportional to its fitness $\alpha(x_i)$, meaning that better candidates have a higher chance of being chosen. Mathematically, the probability of selecting x_i is:

$$P(x_i) = \frac{\alpha(x_i)}{\sum_{j=1}^{N} \alpha(x_j)}$$

where N is the population size.

2. **Crossover**: Once parent solutions are selected, they undergo crossover (recombination) to produce offspring solutions. The most common method is single-point crossover, where two parents x^A, x^B exchange part of their genetic material at a randomly chosen crossover point c. If each solution is represented as a vector $x = (x_1, x_2, \ldots, x_d)$, then the new offspring are:

$$x^C = (x_1^A, \ldots, x_c^A, x_{c+1}^B, \ldots, x_d^B)$$
$$x^D = (x_1^B, \ldots, x_c^B, x_{c+1}^A, \ldots, x_d^A)$$

This operation allows for the exchange of high-fitness traits between parents.

3. **Mutation**: To maintain genetic diversity and avoid premature convergence, mutation introduces small random perturbations in the offspring. A common mutation strategy is Gaussian perturbation:

$$x_i' = x_i + \mathcal{N}(0, \sigma^2)$$

where σ is a small step size controlling exploration. Mutation helps escape local optima and ensures better exploration of the search space.

After multiple iterations, the population is expected to converge towards high-quality solutions, leading to an optimal or near-optimal choice of x^* for maximising $\alpha(x)$.

Algorithm 1 Genetic Algorithm for Acquisition Maximisation

1: Initialize population P of size N.
2: **for** generation $t = 1, 2, \ldots, T$ **do**
3: Evaluate fitness of each $x_i \in P$ using $\alpha(x_i)$.
4: Select top candidates for reproduction.
5: Perform crossover and mutation to generate offspring.
6: Replace worst solutions in P with offspring.
7: **end for**
8: Return best found solution.

Particle Swarm Optimisation

PSO optimises by simulating a group (swarm) of particles moving through the search space [42]. Each particle updates its position based on personal best and global best positions.

Mathematical Formulation: Each particle i has a position x_i and velocity v_i. Updates follow:

$$v_i^{t+1} = \omega v_i^t + c_1 r_1 (p_i^t - x_i^t) + c_2 r_2 (g^t - x_i^t), \tag{3-8}$$
$$x_i^{t+1} = x_i^t + v_i^{t+1}, \tag{3-9}$$

where p_i^t is the personal best, g^t is the global best, ω is inertia weight, and c_1, c_2 are learning coefficients.

Algorithm 2 Particle Swarm Optimisation for Acquisition Maximisation

1: Initialize swarm with random positions and velocities.
2: **for** iteration $t = 1, 2, \ldots, T$ **do**
3: Evaluate $\alpha(x_i)$ for each particle.
4: Update personal and global bests.
5: Update velocities and positions using velocity equations.
6: **end for**
7: Return global best solution.

3-2-4 Bayesian Optimisation Algorithm

Through maximisation of the acquisition function with methods as described above, the Bayesian Optimisation algorithm in practice can be computed as follows:

Algorithm 3 Bayesian Optimisation Algorithm

1: **Input:** Objective function $f(x)$, GP prior, acquisition function $\alpha(x)$.
2: **for** $t = 1, 2, \ldots, T$ **do**
3: Fit GP to current data $\mathcal{D}_t$.
4: Select next point $x_{t+1} = \arg\max \alpha(x)$.
5: Evaluate $y_{t+1} = f(x_{t+1})$.
6: Update GP with $\mathcal{D}_{t+1} = \mathcal{D}_t \cup \{(x_{t+1}, y_{t+1})\}$.
7: **end for**
8: Return best solution $x^* = \arg\min y_t$.

Lastly, can be seen that BO faces challenges including computational scalability (GP scales as $\mathcal{O}(n^3)$), optimisation of acquisition functions, and high-dimensional inefficiencies [35].

3-3 Gaussian Processes

GP [43] are a powerful and flexible statistical tool used in machine learning applications and can be applied to control theory. GPs are particularly useful for modelling complex, non-linear relationships and are employed in various applications such as regression, optimisation , and uncertainty quantification.

At its core, a GP is a collection of random variables, where any finite subset follows a multivariate Gaussian distribution, i.e. it is a distribution over functions. GPs are defined by a mean function and a covariance function (or kernel), which encodes the assumptions about the smoothness and correlations in the underlying process.

3-3-1 Gaussian Process Regression

The GP regression can be used to identify an unknown function $\mathbf{d}_{\text{true}} : \mathbb{R}^{n_z} \to \mathbb{R}^{n_d}$ via a collection of inputs $\mathbf{z}_k \in \mathbb{R}^{n_z}$ and outputs $\mathbf{y}_k \in \mathbb{R}^{n_d}$. The inputs and outputs are thus related

in accordance to the unknown function through:

$$\mathbf{y}_k = \mathbf{d}_{\text{true}}\ (\mathbf{z}_k) + \mathbf{w}_k \tag{3-10}$$

The additive term $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}^{\mathbf{w}})$ is independently, identically distributed Gaussian noise with variance $\mathbf{\Sigma}^{\mathbf{w}} = \text{diag}\left(\left[\sigma_1^2, \ldots, \sigma_{n_d}^2\right]\right)$. Formally, the input and output data sets can be stored in dictionaries following from the input and output data pairs $(\mathbf{z}_i, \mathbf{y}_i)$:

$$\mathcal{D} = \{\mathbf{Y} = \left[\mathbf{y}_0^T; ...; \mathbf{y}_m^T\right] \in \mathbb{R}^{m \times n_d}, \mathbf{Z} = \left[\mathbf{z}_0^T; ...; \mathbf{z}_m^T\right] \in \mathbb{R}^{m \times n_z}\} \tag{3-11}$$

The data collection of inputs and outputs in the dictionary can used for Sparse Gaussian Process Regression in order to reduce computational cost via inducing inputs. In essence, sparse GPs [44] make use of the fact that many quantities can be precomputed and the effective size of the kernel matrix can be drastically reduced. This was not utilised in this research, however it is discussed as a recommendation for future work (see respective section).

The GP is evaluated at a test point z to produce posterior distribution in dimension a comprised of a mean value $\mu^a(\mathbf{z})$ and variance $\Sigma^a(\mathbf{z})$. It is inherent that each output dimension $a \in \{1, ..., n_d\}$ is treated independently. (Thus there is no need for individual normalization of the input and output data for each dimension.)

$$\begin{aligned} \mu^a(\mathbf{z}) =& \mathbf{k}_{\mathbf{zZ}}^a \left(\mathbf{K}_{\mathbf{ZZ}}^a + \mathbf{I}\sigma_a^2\right)^{-1} [\mathbf{Y}]_{\cdot,a}, \\ \Sigma^a(\mathbf{z}) =& k_{\mathbf{ZZ}}^a - \mathbf{k}_{\mathbf{zZ}}^a \left(\mathbf{K}_{\mathbf{zZ}}^a + \mathbf{I}\sigma_a^2\right)^{-1} \mathbf{k}_{\mathbf{Zz}}^a \end{aligned} \tag{3-12}$$

Where $[\mathbf{K}_{\mathbf{ZZ}}^a]_{ij} = k^a(\mathbf{z}_i, \mathbf{z}_j)$ is known as the Gramm matrix and the following covariance are described as $[\mathbf{k}_{\mathbf{zZ}}^a]_j = k^a(\mathbf{z}_j, \mathbf{z}) \in \mathbb{R}, \mathbf{k}_{\mathbf{zZ}}^a = (\mathbf{k}_{\mathbf{zZ}}^a)^T \in \mathbb{R}^m$ and $k^a(\mathbf{z}, \mathbf{z}) \in \mathbb{R}$.

3-3-2 Kernel Selection

A multitude kernel families exist, which are essentially described by covariance functions. The shape of the kernel functions is governed by additional hyperparameters. The kernels and respective hyperparameters are to be chosen and tuned respectively, often based on the practical application. The documentation of the GPML (Gaussian Process for Machine Learning) toolbox for Matlab [45] provides a concise overview of standard kernel families and a method of building (concatenating) more complex kernels.

Squared Exponential (SE) kernel [46], which has become the default kernel in the GP community, was used. The SE kernel integrates well against most physical functions and is defined by only two parameters; the length scale l which affects the rate of change of the model, and the output variance σ^2 which determines the average distance of the function from its mean. In the multivariate scenario the length scale is $\mathbf{L}^a \in \mathbb{R}^{n_z \times n_z}$,the SE kernel is given by:

$$k^a(\mathbf{z}, \bar{\mathbf{z}}) = \sigma_{f,a}^2 \exp\left(-(\mathbf{z} - \bar{\mathbf{z}})^T \mathbf{L}^a (\mathbf{z} - \bar{\mathbf{z}})\right) \tag{3-13}$$

3-3-3 Hyperparameter optimisation

In order to model using a GP the hyperparameters must be chosen, which are in fact not known a priori. It is common practice to initialise these parameters conservatively often even without justification and run an optimisation over a (log) marginal likelihood. The idea being, that the marginal likelihood has a clear maximum around the true hyperparameters which correspond to the true underlying process which is being modelled. Optimising the (log) marginal likelihood thus yields a better model fit, in turn as more data is provided to the optimisation the more informative the likelihood becomes about the underlying process.

The marginal log likelihood is given by, where the covariance $K_y = K_f + \sigma_n^2 I = K(Z,Z) + \sigma_n^2 I$:

$$\begin{aligned} \log p(\mathbf{y}|\mathbf{X},\theta) =& -\frac{1}{2}\mathbf{y}^T K_y^{-1}\mathbf{y} - \frac{1}{2}\log|K_y| - \frac{n}{2}\log 2\pi \\ \theta =& \underset{\theta}{\operatorname{argmax}} \log p(Y|Z,\theta) \end{aligned} \tag{3-14}$$

The marginal likelihood shown above is maximised by utilising gradients in the form of the partial derivatives w.r.t. the hyperparameters as follows below, where $\boldsymbol{\alpha} = K^{-1}\mathbf{y}$. This optimisation is of the non-convex type, thus local optimisation is possible.

$$\begin{aligned} f\theta_j \log p(\mathbf{y}|\mathbf{X},\theta) =& \frac{1}{2}\mathbf{y}^T K^{-1} K\theta_j K^{-1}\mathbf{y} - \frac{1}{2}\mathrm{tr}\left(K^{-1}K\theta_j\right) \\ =& \frac{1}{2}\mathrm{tr}\left(\left(\boldsymbol{\alpha}\boldsymbol{\alpha}^T K^{-1}\right)K\theta_j\right) \end{aligned} \tag{3-15}$$

3-3-4 Mean & Likelihood functions, Inference methods - probabilistic derivation

It is common practice to assume a zero mean of the data and thus employ a zero mean function for modelling of the GP. If necessary, the data can be normalized to be centred around a mean of zero, and the original mean offset can be added post GP evaluation.

Formally the mean function is described as $m_\phi : \mathcal{X} \to \mathbb{R}$, with hyperparameters ϕ which is a scalar function defined over the entire domain $\mathcal{X}$ that evaluates the expected values $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{X})]$ of f for values of x.

Likelihood functions are used for approximate inference during a 'prediction' of the GP model. The predictive mean and variance of a GP prediction are essentially evaluated from the latent marginal moments, namely from the Gaussian marginal approximation $\mathcal{N}\left(f|\mu_f, \sigma_f^2\right)$ via:

$$p(y|\mathcal{D},\mathbf{x}) = \int p(y|f)\, p(f|\mathcal{D},\mathbf{x})\, df \approx \int p(y|f)\, \mathcal{N}\left(f|\mu_f,\sigma_f^2\right) df \tag{3-16}$$

The moments, namely predictive mean and variance respectively follow as:

$$\mu_y = \int y p(y|\mathcal{D},\mathbf{x})\, dy \tag{3-17}$$

$$\sigma_y^2 = \int (y-\mu_y)^2\, p(y|\mathcal{D},\mathbf{x})\, dy \tag{3-18}$$

In the above prediction, a likelihood function p_ρ with hyperparameters ρ is the Bayesian conditional density in the form $\int p_\rho(y|f)\,dy = 1$ defined as a s scalar function values f and outputs y. In turn the likelihood function specified the probability of the observation given the GP model and hyperparameters, which can then be used in the prediction for the mean and variance. Exact Gaussian likelihood regression $y_i \in \mathbb{R}$ is defined as:

$$\mathcal{N}\left(y|f_i,\sigma^2\right) = p_\rho(y_i|f_i) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y_i-f_i)^2}{2\sigma^2}\right) \tag{3-19}$$

3-4 Summary

This chapter introduced BO as a sample-efficient black-box optimisation technique, particularly suited for scenarios where evaluating the objective function is expensive or gradient information is unavailable. BO was compared with other global optimisation strategies such as Evolutionary Algorithms and Direct Search Methods, highlighting the advantages of BO in low-dimensional, costly evaluation settings.

The mathematical formulation of BO was presented, emphasizing its use of Gaussian Processes (GPs) as surrogate models. The predictive mean and variance of the GP allow for the construction of acquisition functions, which balance exploration and exploitation. Two widely used acquisition functions, Expected Improvement and Upper Confidence Bound, were described in detail, including their mathematical derivations and trade-offs.

Maximising the acquisition function is a non-trivial task, and several approaches were discussed: gradient-based optimisation , grid search, Genetic Algorithms, and Particle Swarm Optimisation. GA and PSO were described algorithmically and mathematically, showcasing their ability to navigate multimodal and non-convex spaces for acquisition maximisation.

The Bayesian Optimisation loop was outlined in pseudocode, summarizing the sequential process of surrogate model fitting, acquisition maximisation, and function evaluation. Practical challenges such as acquisition optimisation difficulty, GP scalability ($\mathcal{O}(n^3)$), and limitations in high-dimensional spaces were also acknowledged.

In the second part of the chapter, Gaussian Processes were discussed in greater depth. GP regression was formulated as a probabilistic approach for modelling unknown non-linear functions. The formulation included noise modelling, dictionary-based data handling, and independent modelling of multi-output dimensions. The SEkernel was used to encode assumptions of smoothness and continuity in the modelled function.

Further sections detailed kernel selection, hyperparameters optimisation via marginal likelihood maximisation, and inference mechanisms involving likelihood functions and predictive distributions. This included discussion on mean function assumptions (typically zero mean), and the role of Gaussian likelihoods in generating predictive uncertainty estimates.

Together, these foundations form the basis of using BO with GPs for reference trajectory optimisation in constrained control tasks, as explored in later chapters.

Constrained BO methods include strategies such as using merit functions [47], information-based search frameworks [48], and extensions to noisy experiments [49]. More recent works have also proposed corrected acquisition functions under noise [50].

Chapter 4

Reference Governor for Nonlinear MPC

This chapter outlines the role of the RG in reshaping the reference trajectory which is in-turn provided to the MPC controller for vehicle motion control and path following. The research limits itself to a class sigmoid-shaped reference signals, as those commonly used for evasive manoeuvres for single and double lane changes.

4-1 Optimisation of Sigmoid Parameters via Bayesian Optimisation

The sigmoid reference trajectory described in Equation 4-1 is governed by a parameter vector $\theta = [\theta_1, \theta_2, \theta_3]$. These parameters respectively control the final lateral offset of the trajectory, the spatial location of the transition centre, and the steepness (or sharpness) of the trajectory's lateral displacement as depicted in Figure 4-1. Due to their direct influence on the reference path, selecting appropriate values for these parameters is crucial to ensure the overall performance of the controller, particularly for accurate path tracking and robust obstacle avoidance.

$$y(x) = \frac{\theta_1}{1 + e^{-\theta_2 * (x - \theta_3)}} \tag{4-1}$$

In Model Predictive Control (MPC)-based planning and control schemes, the reference trajectory plays a central role in determining the feasibility and quality of the resulting vehicle behaviour. Therefore, an unoptimised or poorly parametrised reference can result in excessive control effort, suboptimal obstacle clearance, or degraded tracking performance. To address this challenge, a data-efficient optimisation strategy is employed to fine-tune the sigmoid parameters in a principled manner.

To systematically search for optimal parameters, a global optimisation problem over the parameter space Θ is formulated using Bayesian Optimisation (BO). The goal is to minimise

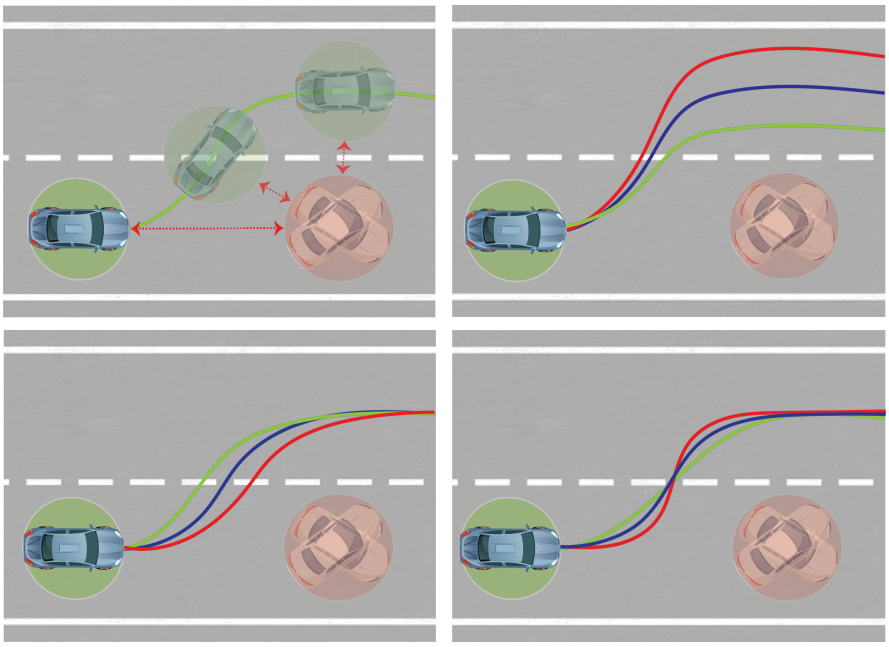


Figure 4-1: Obstacle avoidance lane change scenario. The top left diagram depicts vehicle to obstacle distance (V2O) using red arrows for a target manoeuvre shown in green. The top right, bottom left and bottom right diagrams show a variation in end location, starting point and steepness of manoeuvre.

a scalar performance metric $\Upsilon(\theta)$, which evaluates the closed-loop behaviour of the system with a given set of parameters θ:

$$\theta^* = \arg\min_{\theta \in \Theta} \Upsilon(\theta), \tag{4-2}$$

The objective function $\Upsilon(\theta)$ encapsulates multiple performance criteria, including the lateral and longitudinal tracking errors, the total control effort, and the safety margin to nearby obstacles. Specifically, the function is defined as:

$$\Upsilon(\theta) = w_J J_T(\theta) + w_{Yp}\bar{e}_{Yp}^2 + w_{Xp}\bar{e}_{Xp}^2, \tag{4-3}$$

where $J_T(\theta)$ denotes the cumulative MPC cost over the manoeuvre, and $\bar{e}_{Yp}$ and $\bar{e}_{Xp}$ represent the root mean square lateral and longitudinal tracking errors, respectively. The weights w_J, w_{Yp}, and w_{Xp} can be adjusted to prioritize different aspects of performance, such as smoothness or safety.

This cost function encourages solutions that not only track the desired path accurately but also maintain safe distances from obstacles while minimising unnecessary deviations from the nominal reference. The non-linear and potentially expensive-to-evaluate nature of $\Upsilon(\theta)$ makes BO a suitable framework for this task.

BO constructs a surrogate probabilistic model of the unknown cost function using a Gaussian Process (GP). The GP is trained on previously evaluated samples and provides a posterior distribution over the function's value at unseen locations. Given the GP's predictive mean

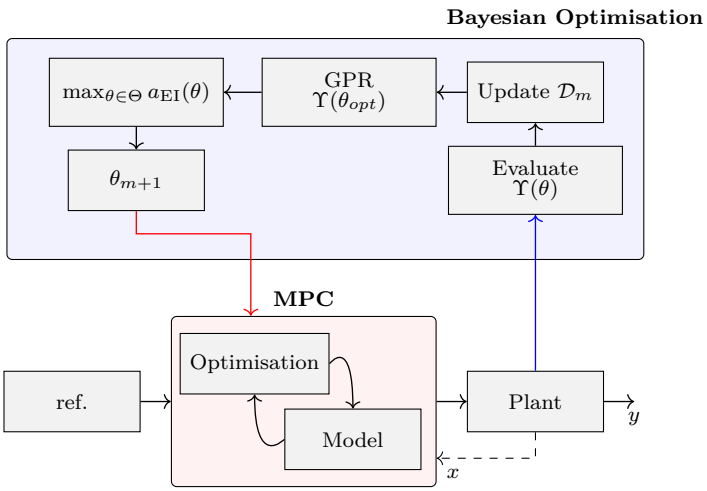


Figure 4-2: Schematic of Bayesian Optimisation loop interacting with MPC for path following and obstacle avoidance.

$\mu_m(\theta)$ and variance $\sigma_m^2(\theta)$ at iteration m, an acquisition function is employed to guide the search for the next candidate parameter:

$$\alpha_{\mathrm{EI},m}(\theta) = (\Upsilon_{\min} - \mu_m(\theta))\Phi(Z) + \sigma_m(\theta)\phi(Z), \tag{4-4}$$

where $Z = \frac{\Upsilon_{\min} - \mu_m(\theta)}{\sigma_m(\theta)}$, and $\Phi(\cdot)$, $\phi(\cdot)$ are the cumulative distribution function and probability density function of the standard normal distribution, respectively. The term $\Upsilon_{\min}$ denotes the best cost observed so far. The Expected Improvement (EI) criterion balances exploration (searching uncertain areas) and exploitation (refining near known optima), and is maximised to select the next evaluation point:

$$\theta_{m+1} = \arg\max_{\theta\in\Theta} \alpha_{\mathrm{EI},m}(\theta). \tag{4-5}$$

The full optimisation loop proceeds iteratively as follows:

1. Generate an initial design of experiments using Latin Hypercube Sampling (LHS) or uniform random sampling.
2. Evaluate the performance metric $\Upsilon(\theta)$ for each sampled θ via closed-loop simulation.
3. Fit a GP model to the collected data and compute the posterior distribution.
4. Select new candidate parameters by maximising the acquisition function.
5. Update the GP model with the newly acquired data and repeat until convergence or maximum budget is reached.

This sample-efficient approach enables convergence to high-performing sigmoid parameters without exhaustively searching the space, which would be infeasible given the computational cost of running full closed-loop simulations for every candidate.

Algorithm 4 Bayesian Optimisation for Offline Sigmoid Parameter Tuning

INPUT: Parameter domain Θ, objective $\Upsilon(\theta)$, budget $m_{\max}$

Initialize: Select initial design $\{\theta_0, \ldots, \theta_{m_0}\}$, evaluate $\Upsilon(\theta_i)$
$S_0 \leftarrow \{(\theta_i, \Upsilon(\theta_i))\}_{i=1}^{m_0}$, fit GP model
for $m = m_0 + 1$ to $m_{\max}$ **do**
  Select $\theta_m = \arg\max_{\theta \in \Theta} \alpha_m(\theta)$
  Evaluate $y_m = \Upsilon(\theta_m)$ via simulation
  Update dataset: $S_m \leftarrow S_{m-1} \cup \{(\theta_m, y_m)\}$
  Refit GP model
end for

OUTPUT: Optimal parameters θ^* minimising Υ

Figure 4-2 summarizes the interaction between the BO loop and the MPC controller.

4-2 Online Reference Governed MPC

While offline optimisation ensures optimality over a diverse range of scenarios, it is often desirable to adapt reference trajectories in real time to cope with dynamic environments. To this end, we introduce an online Bayesian Optimisation-based Reference Governor (BORG), which leverages a pre-trained Gaussian Process model to dynamically adjust the sigmoid parameters θ at runtime.

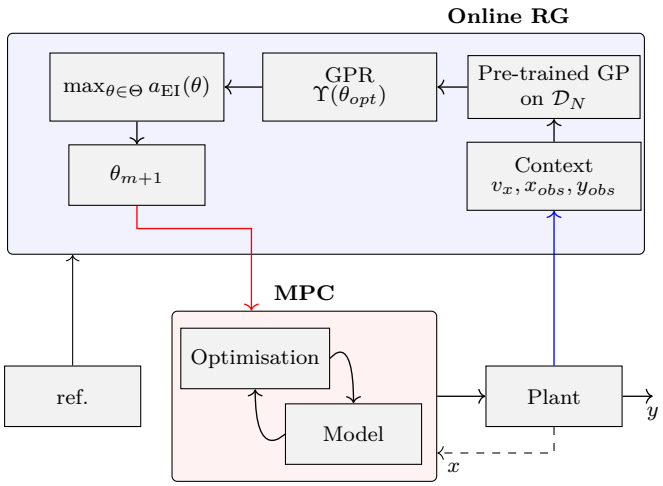


Figure 4-3: Schematic of Bayesian Optimisation loop interacting with MPC for path following and obstacle avoidance.

The core idea is to separate the computationally demanding BO loop into two phases: (i) an offline training phase where the GP model is learned over a wide range of driving contexts and parameters; and (ii) an online execution phase where the GP is used to rapidly infer high-performing parameters based on current contextual observations.

During the offline phase, the MPC controller is simulated under varying contexts, characterised by vehicle velocity v_x and relative obstacle positions (x_{obs}, y_{obs})—and the associated optimal sigmoid parameters and cost values are recorded in a dataset $\mathcal{D}_N$. The GP is then trained to model the mapping:

$$(v_x, x_{obs}, y_{obs}) \rightarrow \Upsilon(\theta)$$

In the online phase, this model is used to infer the expected performance of candidate sigmoid parametrisations. By optimising the same acquisition function as in the offline case (e.g., Expected Improvement), the best candidate θ^* is selected with minimal latency.

This allows the BORG to adapt the reference trajectory in real-time in response to changing environmental conditions, such as new obstacles or velocity changes, while retaining the computational efficiency needed for deployment in embedded systems.

Algorithm 5 Online Reference Governor using Pre-trained GP

INPUT: Pre-trained GP, runtime context (v_x, x_{obs}, y_{obs})
for each control update **do**
  Estimate posterior $\mu(\theta), \sigma^2(\theta)$ conditioned on context
  Compute acquisition $\alpha(\theta)$ and select θ^*
  Generate updated sigmoid reference using θ^*
  Pass reference to MPC for optimal control input computation
end for

This hybrid architecture enables a powerful combination of offline learning and online adaptability. It ensures that the vehicle operates safely and efficiently across a wide range of scenarios without requiring real-time retraining or expensive optimisation during deployment.

4-3 Summary

This chapter introduced a Reference Governor (RG) approach for non-linear MPC using BO to automatically tune the parameters of a sigmoid reference trajectory for path following and obstacle avoidance.

In the first section, an offline BO framework was proposed to optimise the sigmoid parameter vector $\theta = [\theta_1, \theta_2, \theta_3]$, which governs the final offset, centre, and steepness of the trajectory. The optimisation objective $\Upsilon(\theta)$ was formulated to encapsulate tracking accuracy, control smoothness, and obstacle clearance, aggregating metrics such as cumulative MPC cost and root mean square tracking errors. A GP surrogate model was employed to model $\Upsilon(\theta)$, and the EI acquisition function was used to iteratively explore the parameter space. This sample-efficient method significantly reduces the computational cost of exploring feasible parametrizations while converging to high-performing solutions.

In the second section, a strategy for online reference adaptation was presented. This method uses a pre-trained GP model to adjust the reference trajectory in real time based on the current driving context, such as vehicle speed and obstacle position. During the offline training

phase, a dataset $\mathcal{D}_N$ was constructed through closed-loop simulations under diverse scenarios, capturing optimal parameters and associated performance metrics. The GP was trained to model the mapping from environmental context to performance prediction.

At runtime, this model is queried to rapidly infer the expected performance of candidate parameters without re-running full simulations. An acquisition function (EI) is used to select the best parameters in each control update cycle. This results in an adaptive, context-aware reference that improves responsiveness to dynamic environments while maintaining computational tractability.

The overall architecture enables a hybrid control strategy, leveraging data-driven offline learning and efficient online deployment and offering robust performance across varying conditions without real-time retraining or high computational overhead.

Chapter 5

Toward Efficient Surrogate Training

In order to rigorously evaluate and train the Reference Governed Model Predictive Control (RG-MPC) for robust real-time operation, the design of an efficient and representative training dataset is critical. The goal is to balance the quality and diversity of training samples with the constraints of computational cost and practical feasibility, particularly when considering physical testing on real vehicles. This section presents and analyses three alternative experiment design strategies for generating contextual training data to support the offline Bayesian Optimisation loop (Algorithm 4) and the construction of the Gaussian Process (GP) model, which is later queried online by the Reference Governor during closed-loop execution.

The RG-MPC framework is sensitive to contextual factors such as vehicle velocity and obstacle placement, which directly influence the shape and aggressiveness of the evasive reference trajectory. Therefore, a well-structured experiment design should ensure adequate coverage of this contextual parameter space, namely, the longitudinal velocity of the vehicle v_x and the obstacle location (x_{obs}, y_{obs}) to facilitate reliable generalisation of the GP model across relevant and safety-critical scenarios.

Figure 5-1 provides a visual overview of the sigmoid reference trajectory and obstacle zones considered during the training phase. Each training sample represents the outcome of a simulation or physical test run with a fixed context, where the RG-MPC is applied and its performance is evaluated based on a defined cost function $\Upsilon(\theta)$. This cost captures the efficacy of a candidate sigmoid reference parameter set θ in achieving obstacle avoidance with desirable dynamic properties.

To efficiently generate these training samples, three candidate experiment design strategies are proposed and discussed below.

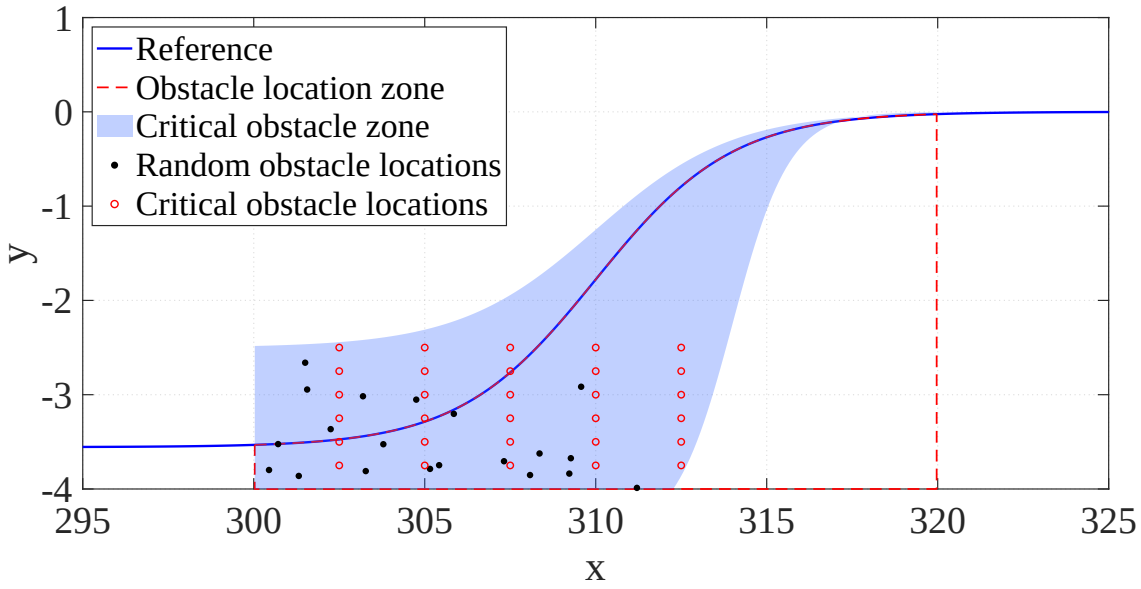


Figure 5-1: Sigmoid reference trajectory overlaid with obstacle zones and randomised obstacle locations used for training sample generation.

5-1 Method 1: Uniform Random Contextual Sampling Across Full Envelope

This method involves generating training samples through uniform random sampling of the contextual parameters v_x, x_{obs}, and y_{obs} within a wide feasible domain that represents the operational envelope of the automated vehicle. In practice, this could range from low-speed urban driving (e.g., 20–30 km/h) to highway speeds exceeding 90 km/h, as well as a range of obstacle positions representing various lateral offsets and distances from the ego-vehicle.

- **Advantage:** This approach maximizes diversity in the training dataset, allowing the GP model to experience a wide range of driving conditions, including edge cases that may not arise in more structured experiments. As such, it enhances the model's generalization ability and robustness to unseen conditions during deployment.

- **Disadvantage:** Despite its comprehensiveness, this method is highly inefficient in terms of data usage. A large number of samples are required to sufficiently cover the full parameter space, especially given the curse of dimensionality. Many of these samples may correspond to non-critical or redundant driving scenarios that contribute little to the refinement of the GP model. Furthermore, the computational and logistical burden of collecting such a large dataset, particularly in physical vehicle tests is often prohibitive.

5-2 Method 2: Critical Region Sampling with Structured Sweeps

To improve efficiency, this method restricts the sampling domain to a more focused region of the contextual space, emphasizing critical conditions where obstacle avoidance becomes challenging or safety-critical. Specifically, obstacle positions are sampled near the ego-lane centreline, which are more likely to trigger evasive actions. Longitudinal speeds are selected near the vehicle's dynamic handling limits, typically in the range of 60 to 90km/h.

A structured sweep approach is used to systematically vary the parameters. Fine-grained discretization is applied to the obstacle lateral location, capturing the sensitivity of lateral positioning, while longitudinal position may be varied more coarsely due to its lower impact on trajectory design.

- **Advantage:** By focusing on high-risk scenarios, this method yields a more informative training dataset with fewer samples, accelerating the convergence of the GP model and Bayesian Optimisation. Additionally, it aligns well with physical testing, where high-speed and close-call obstacle encounters are both critical to test and more likely to elicit nuanced vehicle responses.
- **Disadvantage:** A reduced sampling envelope may lead to limited extrapolation capability of the GP model outside the critical region. In scenarios with low-speed or low-risk conditions, where fewer training examples exist, the model may revert to overly conservative or suboptimal reference choices.

5-3 Method 3: Critical Zone Random Sampling with Representative Velocities

This hybrid strategy seeks to further reduce training effort while preserving relevance by introducing two simplifying assumptions. First, obstacle locations are randomly sampled within the critical region defined in Method 2, ensuring the training data remains focused on meaningful and informative cases. Second, rather than sampling velocity continuously or across a wide grid, only two representative vehicle speeds are used:

- A moderate speed (e.g., 55 km/h), representative of comfort-oriented driving or urban environments.
- A high-speed value (e.g., 90 km/h), representative of aggressive or emergency conditions requiring fast evasive manoeuvres.

The idea is that by covering the extremities of the velocity spectrum, the GP model can interpolate between these conditions and remain effective at intermediate velocities encountered during deployment.

- **Advantage:** This method drastically reduces the number of required simulations while still covering key behavioural regimes of the vehicle. It is particularly well-suited for

iterative model development where rapid testing cycles are necessary. It also supports efficient tuning of the offline Bayesian Optimisation loop with a tractable number of design points.

- **Disadvantage:** The main limitation of this approach is the assumption that the optimal sigmoid reference trajectory changes smoothly with respect to velocity. If this assumption fails, such as in highly non-linear or threshold-dependent behaviour, the model may fail to generalize accurately across intermediate velocities not represented in training.

5-4 Summary and Practical Considerations

Each proposed training strategy represents a different trade-off between diversity, efficiency, and practical feasibility. While Method 1 provides the broadest data coverage, it is rarely feasible for real-world experimentation due to its scale. Method 2 offers a more targeted and realistic path by focusing on high-impact scenarios but may require complementary strategies for full envelope generalization. Method 3 offers a pragmatic starting point, especially during early development phases or when computational resources are limited.

In practical applications, a staged or hybrid approach is often most effective. Initial training may leverage Method 3 to quickly bootstrap the GP model with informative data. Subsequently, Method 2 can be used to improve the model's knowledge in critical areas, improving robustness and reducing uncertainty in high-risk contexts. This layered strategy allows for efficient scaling of the Bayesian Optimisation-based Reference Governor (BORG) framework from simulation to physical deployment.

Chapter 6

Simulation Environment

This chapter presents an overview of the Simulink and IPG Automotive CarMaker integration used in the simulation environment, where the CarMaker plant can be interchanged with any simpler vehicle model. It begins with a general workflow description, followed by a brief explanation of the controller tuning method. Subsequently, the Key Performance Indicator (KPI)s used for objective evaluation are outlined and lastly, the method for testing robustness and sensitivity is outlined.

6-1 Overview Simulink system

An overall representation of the simulation framework is depicted in Figure 6-1. Within this setup, the Model Predictive Control (MPC) is responsible for computing the control inputs required to follow the desired trajectory. These control signals are then forwarded to the vehicle model, being either a simple single track or high fidelity CarMaker model, which is integrated within the Simulink environment. The plant, specifically CarMaker model, subsequently simulates the physical behaviour of the vehicle in response to the applied controls, including effects such as suspension, steering system and power train dynamics, and tire-road interactions. The result of this simulation is a set of updated vehicle states, which represent the current condition and motion of the vehicle, such as its position, velocity, and yaw rate.

These vehicle states serve as essential inputs to several components within the control architecture. Specifically, they are utilised by the Reference Generator to compute the desired future trajectory, by the reference governor block responsible for augmenting the original reference, and by the online data module. While the Reference Generator determines the reference signals for the MPC to track, the reference governor augments this reference to better meet tracking and obstacle safety requirements. Lastly, the online data block is used to compute and manage additional vehicle-related quantities that are not directly available as part of the state vector but are nevertheless important for accurate control. One key example of such quantities are the cornering stiffness coefficients, which are derived using the

Dugoff tyre model and reflect the lateral grip characteristics of the tires under current driving conditions.

The overall co-simulation approach between Simulink and CarMaker follows a structure that is broadly aligned with that presented in [51] and [52]. However, the Bayesian optimisation reference governor is a key distinction which differentiates the implementation described in this work. The proposed Bayesian Optimisation-based Reference Governor (BORG) design extends the scope of obstacle avoidance beyond what was considered in [51] and [52] by incorporating ego vehicle to obstacle safety distance into the reference governor re-planner as seen in 6-1.

Furthermore the strategy employed for reference trajectory in [51] relies on a reactive scheme where the reference manoeuvre is dynamically computed based on the relative distance and velocity to a preceding object. Whilst [52] adopts a more structured and pre-planned approach where the reference is constructed from a predefined manoeuvre that depends solely on the longitudinal position of the ego vehicle. Thereby decoupling the reference path from surrounding traffic and enabling more consistent benchmarking of controller performance. The latter approach of a pre-determined lane change manoeuvre is adopted for this study however, with the re-introduction of obstacle information and avoidance through the reference governor.

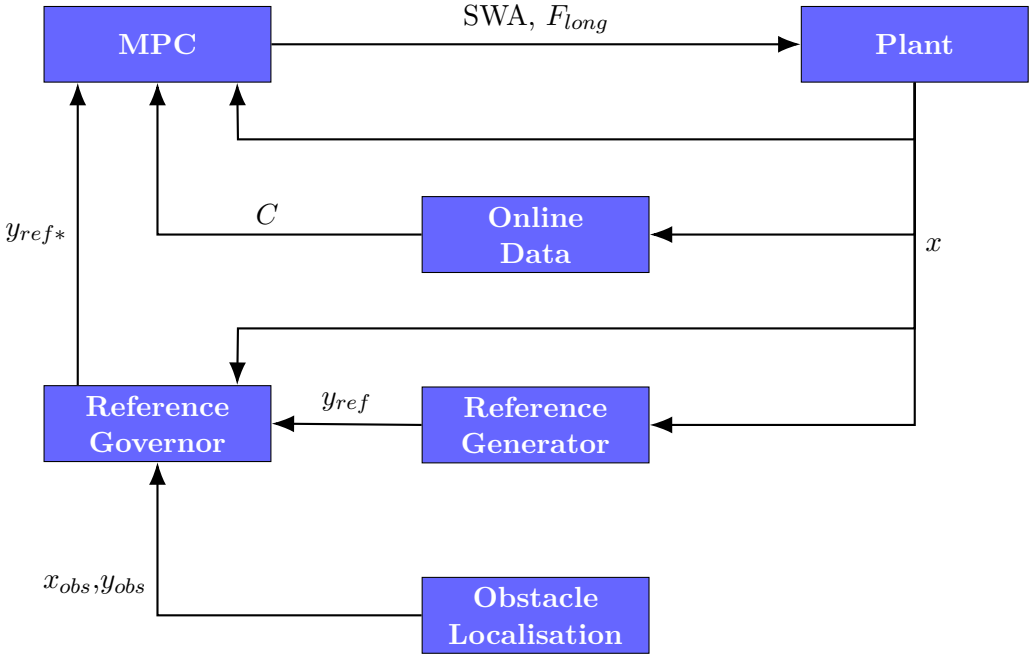


Figure 6-1: Simulink model workflow overview.

6-2 Controller Tuning

The design of MPC inherently relies on an effective tuning process to ensure satisfactory closed-loop performance. Similar to conventional control strategies such as Proportional-Integral-Derivative (PID) or Linear Quadratic Regulator (LQR), MPC requires the careful

selection of tuning parameters, most notably the weights in the cost function. However, MPC tuning is often perceived as more challenging due to the higher number of parameters involved, which are not only more numerous but also interdependent in the context of constrained optimisation.

In [51], the authors provide a comprehensive discussion on the complexity of MPC tuning, using the example of a planar vehicle model that requires tuning of more than 35 parameters in addition to other controller settings. The process outlined in their work involves detailed consideration of which states to include in the cost function, how to allocate appropriate weighting values, and how to balance state-tracking accuracy against control effort. A tuning rule based on a desired state error threshold is proposed, wherein the tuning weight is set inversely proportional to the square of the acceptable tracking error. For example, an error tolerance of 1 cm translates to a tuning weight of 100. Additionally, their work highlights the importance of penalizing control effort to protect actuator health and manage control energy, especially in long-term deployment scenarios.

Despite the insight and granularity of such approaches, a major drawback lies in the heavy dependence on manual tuning, simulation-based trial-and-error iterations, and expert domain knowledge. This tuning burden is further amplified when the controller must perform across a range of operating conditions, such as varying vehicle speeds or road surface friction values (μ).

In contrast to the methodology in [51], the tuning strategy adopted in the present work is intentionally simplified. The objective is to reduce the manual design effort, eliminate subjectivity in tuning, and improve the generalizability of the controller through systematic yet lightweight design. To that end, we define a minimal set of tuning weights focused only on the key performance objectives relevant to the application.

Specifically, three main objectives are considered in the tuning phase:

- **Lateral position tracking**: Ensuring accurate trajectory following in the lateral domain is essential for safe and stable vehicle motion.
- **Longitudinal velocity tracking**: A constant or desired forward velocity is important for maintaining the intended manoeuvre timing and execution.
- **Obstacle avoidance performance**: Penalizing deviations from a safe trajectory during training enhances the learning-based reference generation.
- **Actuation command**: Penalizing large control action commands to preserve actuation health and energy demands.

These weights are applied uniformly in the cost function of the MPC across all manoeuvres during training, without separate fine-tuning for each condition or driving scenario. By adopting this approach, we significantly reduce the number of tuning decisions and focus on achieving baseline performance rather than scenario-specific optimisation.

In the online implementation of the BORG framework, the Reference Governor plays a critical role in adapting the planned trajectory. It compensates for any potential sub-optimality or degradation in performance that may arise from the use of generic, non-specialized tuning weights. Since the reference is updated at runtime based on the current state and environment,

the controller is effectively provided with a feasible and safer reference, reducing the demand for highly precise cost function tuning.

This strategy trades off scenario-specific optimality for robustness and ease of deployment. Instead of fine-tuning weights for each manoeuvre and velocity setting, as done in [51], our approach relies on a single training scenario to perform and validate the tuning. While this may lead to some performance loss in edge cases, the benefit lies in a highly streamlined design pipeline, minimal human intervention, and greater automation in controller configuration.

Ultimately, this simplified tuning paradigm aligns with the goal of scalable, general-purpose autonomous vehicle control design, where tuning complexity and manual effort must be minimised to enable efficient deployment across varied operational contexts.

6-3 Performance Evaluation

The evaluation of controller performance is a critical step in validating the effectiveness of the proposed BORG framework. In line with the methodology adopted in [51], KPIs are employed to assess the system's ability to safely and accurately execute evasive manoeuvres. However, given the nature of the current study, which focuses on a Reference Governor-based online replanning strategy, the selected KPIs are chosen to better capture both safety and tracking fidelity across different phases of the manoeuvre. A study of performance metrics of controlled systems related to vehicle motion control [53], [54], [55], [56] revealed that the indicators used for evaluation should indeed reflect performance characteristic related to the manoeuvre type, in this case a lane change similar to a traditional controlled system step response [57], [58], [59].

This study emphasises safety-driven evaluation metrics alongside traditional trajectory tracking criteria. In this context, the performance indicators are grouped into three main categories: safety-focused metrics, trajectory tracking errors, and dynamic response characteristics inspired by step response analysis.

6-3-1 Safety Indicator: Distance to Obstacle

The primary KPI in terms of safety is the Distance to Obstacle (D2O), which measures the minimum distance between the ego vehicle and the obstacle (e.g., a leading vehicle) throughout the manoeuvre. This is analogous to the Distance to Collision (DTC) introduced in [51], but generalized to capture lateral clearance in any avoidance scenario. A higher value of D2O indicates safer behaviour, as it reflects a more conservative lateral deviation from the obstacle during the evasive action.

6-3-2 Trajectory Tracking: Root mean square errors

To evaluate tracking accuracy, Root Mean Square Error (RMSE) metrics are computed for key vehicle states. These quantify the average deviation of the ego vehicle's actual trajectory from the desired reference. In this study, we focus primarily on the lateral position error y_{RMS}, which is particularly relevant for lane change and obstacle avoidance tasks.

Given that the BORG framework dynamically modifies the reference in real-time, we further divide the lateral RMSE into two distinct components:

- **RMSE ($y_{\mathbf{RMS}}^{\mathbf{total}}$)**: The tracking error for the complete lane change manoeuvre.
- **Pre-Obstacle RMSE ($y_{\mathbf{RMS}}^{\mathbf{pre}}$)**: The tracking error in the portion of the trajectory leading up to the obstacle.
- **Post-Obstacle RMSE ($y_{\mathbf{RMS}}^{\mathbf{post}}$)**: The tracking error in the recovery phase after passing the obstacle.

This segmentation allows assessment of the trade-off introduced by the Reference Governor (RG) mechanism, whether the controller is prioritizing obstacle clearance (increased safety distance) at the cost of deviating from the original manoeuvre, and how effectively it recovers tracking performance after the avoidance phase.

6-3-3 Dynamic Response: Step-Inspired Metrics

In addition to RMSE-based tracking evaluation, inspiration is also drawn from step response analysis to examine the temporal dynamics of the system. As in [51], the evasive manoeuvre exhibits characteristics similar to a step response, particularly in lateral displacement. Accordingly, the following classical KPIs are considered but with but with longitudinal distance as independent variable instead of time:

- **Overshoot (M_p)**: The maximum deviation of the lateral position beyond the steady-state reference, expressed as a percentage.
- **Rise Distance (X_r)**: The distance taken for the lateral position to transition from 10% to 90% of its final value.
- **Settling Distance (X_s)**: The distance required for the lateral position to remain within a 10% band of the steady-state lane change lateral value.

These metrics provide insight into how aggressively and quickly the vehicle responds to the planned manoeuvre and how efficiently it stabilizes post-execution.

6-3-4 Summary of KPIs

In total, the following KPIs are used to quantify controller performance in the BORG study:

- **Minimal Distance to Obstacle (D2O)** – primary safety metric
- **Lateral Position RMSE ($y_{\mathbf{RMS}}$)** – overall tracking error
- **Pre-Obstacle RMSE ($y_{\mathbf{RMS}}^{\mathbf{pre}}$)** – tracking error before avoidance
- **Post-Obstacle RMSE ($y_{\mathbf{RMS}}^{\mathbf{post}}$)** – tracking error after avoidance

- **Overshoot** (M_p) – maximum overshoot in lateral trajectory
- **Rise distance** (x_r) – responsiveness of the lateral transition
- **Settling distance** (x_s) – post-manoeuvre stabilization rate

The ideal controller behaviour, therefore, corresponds to a maximized safety distance (high D2O) and minimised values for all other KPIs, indicating accurate and efficient tracking with stable and smooth transitions.

By evaluating controller performance using this comprehensive yet tailored set of indicators, we ensure that both tracking fidelity and obstacle avoidance are adequately captured, while also identifying the specific contributions of the RG module in different phases of the manoeuvre.

6-4 Robustness & Sensitivity Analysis Methodology

To evaluate the resilience and adaptability of the BORG framework under realistic conditions, a robustness and sensitivity analysis was designed and implemented. The goal of this analysis is to investigate how well the controller generalizes to unseen or imperfect scenarios that deviate from the nominal design conditions, particularly in the presence of model mismatches and perception uncertainties.

This methodology focuses not on the quantitative results of performance degradation but rather on the experimental structure and rationale, with emphasis on how the reference governor's learned policy inherently compensates for un-modelled discrepancies. At the core of the BORG is a Gaussian Process (GP) model, trained offline via Bayesian Optimisation using a purely data-driven, model-free framework. This training strategy allows the reference governor to implicitly account for imperfections in the plant model or measurement pipeline, as it optimises the reference trajectory not with respect to an idealized model, but based on observed performance metrics.

6-4-1 Model Mismatch Simulation

To study the effects of structural modelling errors, the system was subjected to artificially introduced discrepancies between the prediction model used in the MPC and the actual vehicle dynamics in simulation. Specifically, this was achieved by altering tyre model parameters during simulation to represent tyre stiffness degradation or mismatch conditions which are common in real-world deployment due to wear or even environmental variability in road surface conditions. This is done by generating errors in the Magic Formula factors B_y and D_y [18] which govern the stiffness factor of how quickly the tyre can generate grip based on slip and the peak force respectively, as depicted in Figure 6-2.

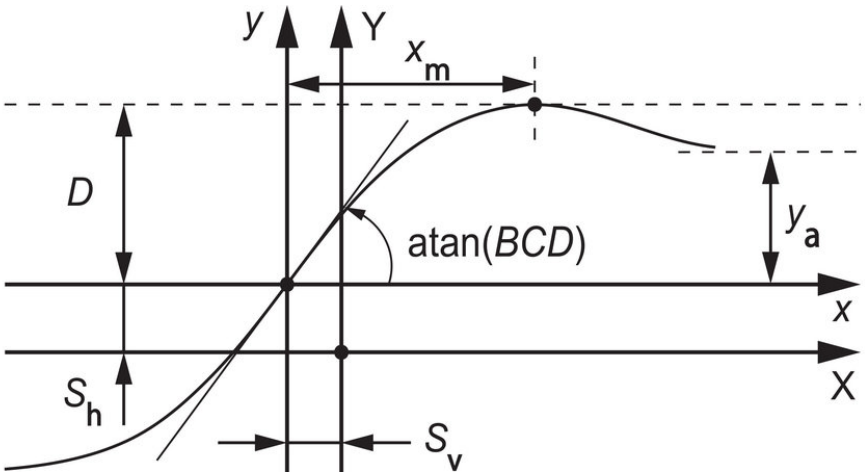


Figure 6-2: Graph depicting Pacejka's Magic Formula tyre model as per [60]. X-axis shows slip vs. normalized force on Y-axis. B, C, D, E are the stiffness, shape, peak and curvature factor parameters, respectively. S_h is the horizontal shift and S_v is the vertical shift.

In the nominal training phase, the BORG was tuned using a known, fixed set of tyre parameters. During robustness testing, however, the actual vehicle dynamics were altered by scaling the cornering stiffness values. This results in a mismatch between expected and actual lateral dynamics, which cannot be fully compensated by the baseline MPC alone. In contrast, the BORG utilises the reference governor's GP model trained in a model-free manner, to adapt the reference trajectory such that feasible and safer trajectories are selected even when the prediction model is no longer fully accurate.

6-4-2 Perception and localisation errors

In parallel to model mismatch, the robustness of the controller against perceptual uncertainty was assessed through the injection of synthetic Gaussian noise into the obstacle position estimates. The estimated obstacle locations used in both the reference generator and the constraint formulation of the MPC were perturbed with zero-mean Gaussian noise of varying standard deviation to simulate perception inaccuracies stemming from sensor noise, latency, or poor detection confidence.

Since the MPC relies on these estimates to generate constraints and assess feasibility, inaccurate obstacle localization can severely impact planning and control. However, in the case of BORG, the GP model learns to encode trajectory patterns that were optimal not only for specific obstacle configurations but also for varied realizations of noise encountered during the Bayesian optimisation process. As a result, the learned policy inherently biases the generated references toward solutions that generalize across a distribution of perception errors.

6-4-3 Velocity-Dependent Evaluation

Both the model mismatch and perception noise experiments were conducted across a range of longitudinal velocities to simulate different driving conditions and levels of manoeuvre criticality. At higher speeds, vehicle dynamics become more sensitive to modelling errors

and perception inaccuracies have more significant implications due to reduced reaction times. This velocity sweep allows for an evaluation of the BORG's robustness across multiple levels of operational stress, ensuring that conclusions drawn from the analysis are not confined to a single operating point.

To isolate the contribution of the reference governor, the BORG is compared to a baseline MPC implementation that uses a static reference trajectory. The only distinction between the two controllers is the handling of obstacle information. The baselines handles obstacle avoidance through a maximization of obstacle safety distance in the MPC cost function, whilst the proposed system as discussed re-plans the reference trajectory. This ensures that any observed improvements in robustness can be attributed directly to the learned model's ability to adapt reference trajectories.

Through the systematic injection of both model and perception disturbances and the use of a velocity-dependent evaluation framework, the robustness analysis aims to demonstrate that the BORG controller inherits resilience properties from the underlying GP-based reference generation strategy. Trained via model-free Bayesian Optimisation, the reference governor learns policies that are implicitly tolerant to imperfections and uncertainty, and thus offers an elegant and scalable solution to one of the primary challenges in predictive control robustness.

6-5 Summary

This chapter introduced the simulation framework used to develop and evaluate the proposed BORG architecture. The system integrates a high-fidelity CarMaker vehicle model with Simulink-based control components, allowing for realistic closed-loop simulations and modular plant model substitution.

Furthermore, the chapter introduces the workflow regarding the key innovation presented in this study, the use of a Gaussian Process-based Reference Governor, trained via model-free Bayesian Optimisation. This module enables dynamic reference adaptation that enhances obstacle avoidance and improves controller robustness to environmental and modelling uncertainties.

The chapter also detailed the controller tuning methodology, which adopts a simplified, objective-driven weighting scheme to reduce manual design effort. This is complemented by the use of a curated set of KPIs, designed to assess safety, tracking accuracy, and transient response across different phases of the manoeuvre.

Lastly, a robustness and sensitivity analysis methodology was outlined, demonstrating how the BORG can handle model mismatches and perception disturbances through its learned reference adaptation policy. These experimental procedures are fundamental for verifying the controller's generalizability to real-world uncertainties and contribute to the overarching goal of scalable, safe autonomous driving.

In the following chapters, the results of the simulation studies are presented and discussed in light of the framework and methodology established here.

Chapter 7

Performance and Results

This chapter presents a detailed analysis of the case study simulations, as introduced and described in Chapter 6. The objective is to evaluate the effectiveness and robustness of the proposed Bayesian Optimisation-based Reference Governor (BORG) system in comparison with the baseline Model Predictive Control (MPC) framework. Firstly, a standard lane change manoeuvre is simulated for the nominal MPC system, which operates using a combined objective function for path-following and obstacle avoidance. The performance of this nominal baseline is then systematically compared against the proposed BORG system, which integrates a reference governor scheme for online trajectory reshaping. Secondly, a robustness analysis is conducted by introducing realistic perception errors and tyre model mismatches to examine how each controller performs under non-ideal, uncertain conditions that are representative of real-world deployment.

7-1 Lane Change Manoeuvre Analysis

To evaluate the effectiveness of the BORG system, a Gaussian Process (GP) model was pre-trained according to the methodology laid out in Chapters 4 and 5. To summarise, the training envelope was constructed to include a range of representative driving scenarios, covering variations in velocity and obstacle position. These scenarios were chosen to ensure that the model would generalize well and be robust in deployment. Specifically, the training set encompassed the following conditions:

- **Velocity scenarios** – A representative spread of vehicle speeds was considered, including 80 km/h and 55 km/h.
- **Obstacle location (longitudinal range)** – Obstacle positions were randomly sampled between 400 m and 420 m along the longitudinal axis.
- **Obstacle location (lateral range)** – Lateral placements of obstacles were constrained to lie between -4 m and -2 m.

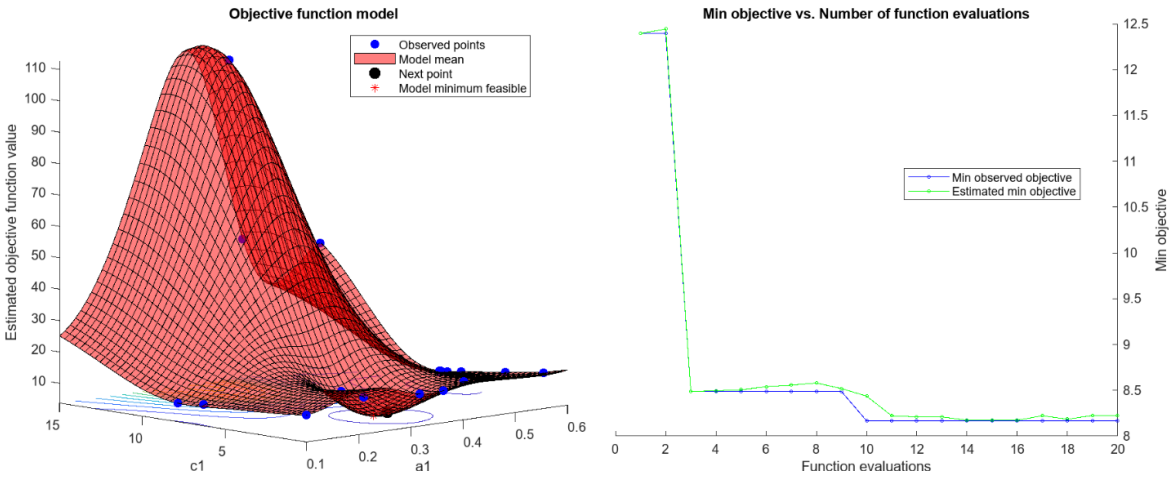


Figure 7-1: Bayesian optimisation results during training. Left: Objective function landscape over sigmoid parameters. Right: Minimum objective value as a function of BO iterations (warm-started with 20 LHS points).

7-1-1 Offline Bayesian Optimisation

Figure 7-1 illustrates an example of the Bayesian Optimisation (BO) process carried out during the offline training of the GP surrogate model. Notably, when the optimisation process is warm-started with 20 randomly sampled data points using Latin Hypercube Sampling (LHS), convergence to the optimal solution is achieved in approximately 10–15 iterations, as shown in the right-hand plot of Figure 7-1.

The shape of the objective function being minimised—designed to balance positional tracking accuracy and safe obstacle clearance—is visualised in the left-hand plot. The optimisation is conducted over the parameters of the sigmoid reference trajectory. In particular, parameters a_1 and c_1 represent the longitudinal midpoint of the lane change and the sigmoid gradient (i.e., aggressiveness), respectively. The third parameter, controlling the final lateral position, is excluded from the plot because, in this specific evaluation, the lane change width is fixed. Therefore, optimisation naturally prioritises the remaining parameters to improve the cost without violating this constraint.

The plots indicate that the objective function possesses a local maximum near the initial reference trajectory parameters. The obstacle's position contributes to this high cost, prompting the BO process to adjust the sigmoid parameters in order to identify a safer and more efficient reference trajectory. The optimal configuration is reached after roughly 10 iterations following the warm start. The resulting parameters correspond to a manoeuvre that initiates earlier and is less aggressive, thereby enhancing safety margins.

The subsequent sections present a performance analysis of the BORG and nominal MPC systems under three test cases, each defined by different vehicle velocities and obstacle locations. The first case represents an in-distribution scenario (80 km/h), whereas the remaining two test cases (70 km/h and 60 km/h) assess the generalisation capability of the controller to conditions outside the training set.

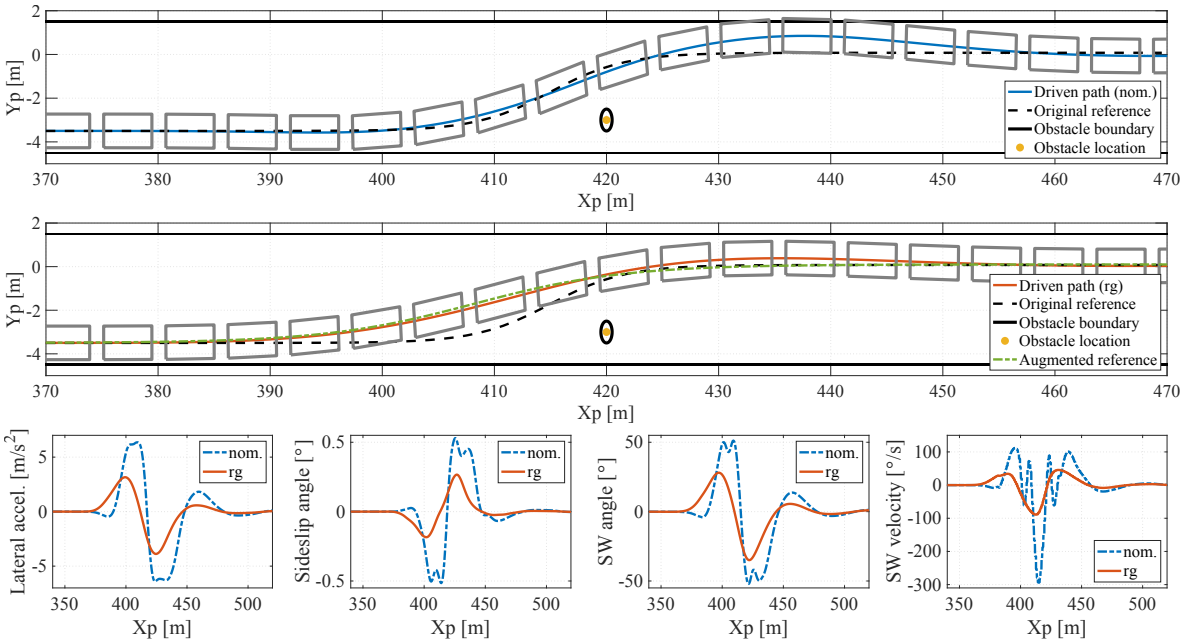


Figure 7-2: Trajectory performance comparison for MPC and BORG at 80 km/h.

	x_r [m]	M_p [%]	x_s [m]	y_{RMSE} [m]	$y^{\text{pre}}_{\text{RMSE}}$ [m]	$y^{\text{post}}_{\text{RMSE}}$ [m]	D2O_{min} [m]
MPC	17.45	21.61	102.62	0.228	0.097	0.277	0.516
BORG	26.47	8.61	95.23	0.3226	0.519	0.119	0.976

Table 7-1: Key performance indicators at 80 km/h.

7-1-2 Lane Change Performed at 80 km/h with Obstacle at (420, -3) m

In the case at 80 km/h, where the velocity of test scenario is included within the training data, the BORG system demonstrates clear performance advantages. Leveraging the pre-trained GP model, the reference governor adapts the trajectory online with a high level of confidence. As a result, the BORG achieves substantially greater clearance from the obstacle ($\text{D2O}_{\text{min}} = 0.976$ m) compared to the baseline MPC (0.516 m), which enhances overall safety.

Moreover, the nominal MPC system exhibits a lateral acceleration of 6.3 m/s^2, indicating operation close to the vehicle's handling limits. In contrast, the BORG system yields a lower lateral acceleration, as depicted in Figure 7-2, thereby reducing the stress on the vehicle's dynamics and providing a less aggressive manoeuvre. The corresponding control actions—namely the steering rate and steering wheel angle—are smoother and of smaller magnitude under BORG. This leads to reduced actuator wear, lower energy consumption, and improved passenger comfort.

These improvements stem from the governor's tendency to initiate the lane change earlier, producing a smoother rise in trajectory and reducing overshoot. While this results in a slight increase in pre-obstacle lateral deviation (reflected in higher $y^{\text{pre}}_{\text{RMSE}}$), it simultaneously lowers the post-obstacle tracking error, indicating a smooth reentry into the desired path.

Additionally, due to the smoother nature of the generated trajectory, the BORG imposes a

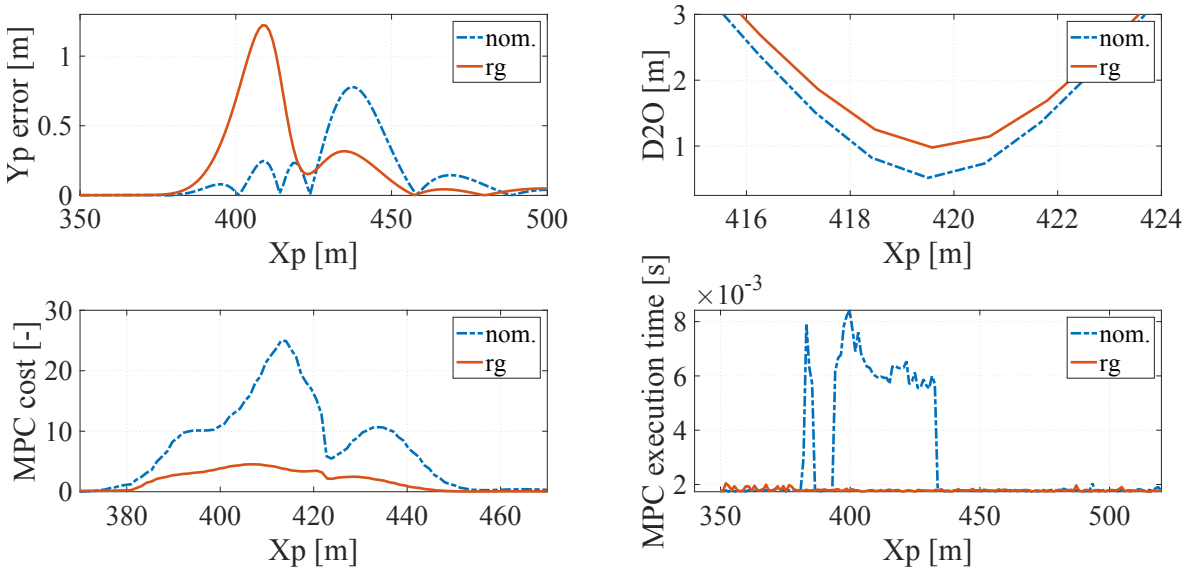


Figure 7-3: Error metrics and control cost comparison for MPC and BORG at 80 km/h.

lower computational burden. The reduced control effort also results in a lower total cost, confirming the efficiency of the proposed strategy.

7-1-3 Lane Change Performed at 70 km/h with Obstacle at (415, -4) m

The second test case evaluates controller performance at a reduced velocity of 70 km/h, a condition that lies outside the distribution of the training dataset. Both the velocity and obstacle location differ from those seen during the offline optimisation phase, allowing for an assessment of the generalisation capability of the BORG controller.

Despite this being an extrapolated scenario, the BORG continues to demonstrate improved performance relative to the baseline MPC controller. Most notably, the minimum distance to the obstacle is significantly higher for BORG ($D2O_{min} = 0.887$ m) compared to the baseline (0.543 m). This confirms the BORG's ability to maintain safety margins through adaptive trajectory adjustments, even under previously unseen scenarios.

As shown in Figure 7-4, both controllers benefit from the lower vehicle speed, which reduces the severity of non-linear vehicle dynamics and makes the manoeuvre inherently more manageable. However, the BORG still demonstrates a more conservative response by initiating the lane change slightly earlier, as reflected in the longer rise distance and smaller overshoot.

The lateral position Root Mean Square Error (RMSE) is lower for MPC before the obstacle, due to its more direct approach, but post-obstacle RMSE is improved under BORG. This behaviour is consistent with a strategy that prioritises safe and smooth rejoining of the lane. The total RMSE is slightly higher for BORG, attributed to its pre-emptive deviation from the original path, yet this is an intentional trade-off favouring safety and comfort.

In terms of controller effort, the lateral acceleration and sideslip angle under BORG remain lower than the nominal case, highlighting a continued focus on vehicle stability. While the control signals are naturally smaller due to the lower speed, BORG still yields smoother

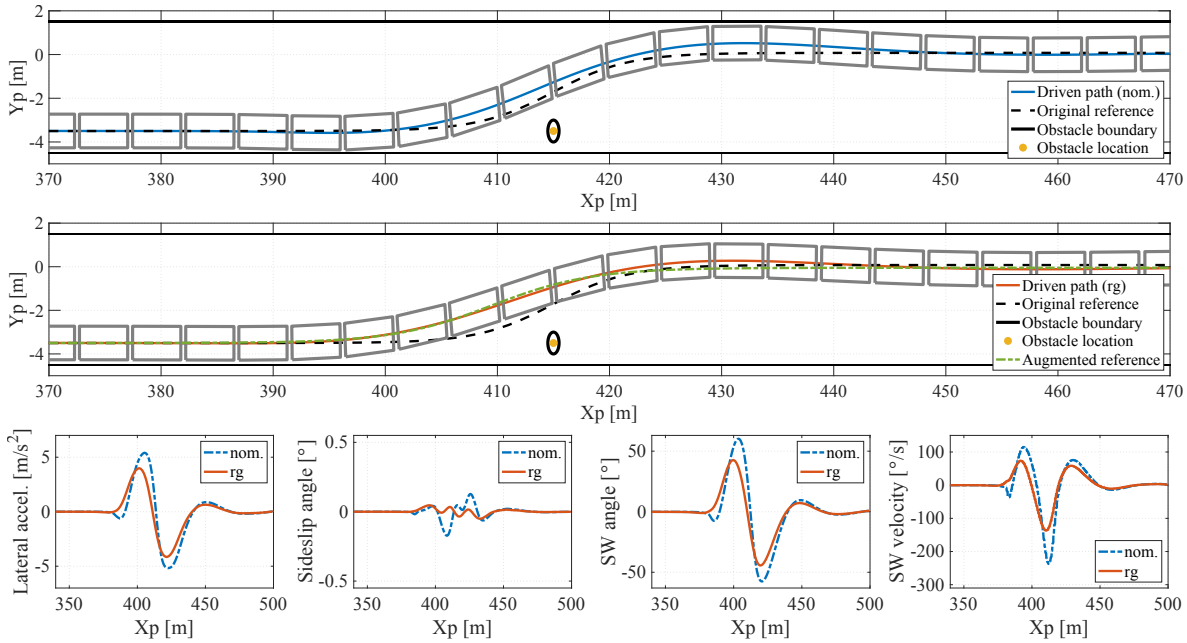


Figure 7-4: Trajectory performance comparison for MPC and BORG at 70 km/h.

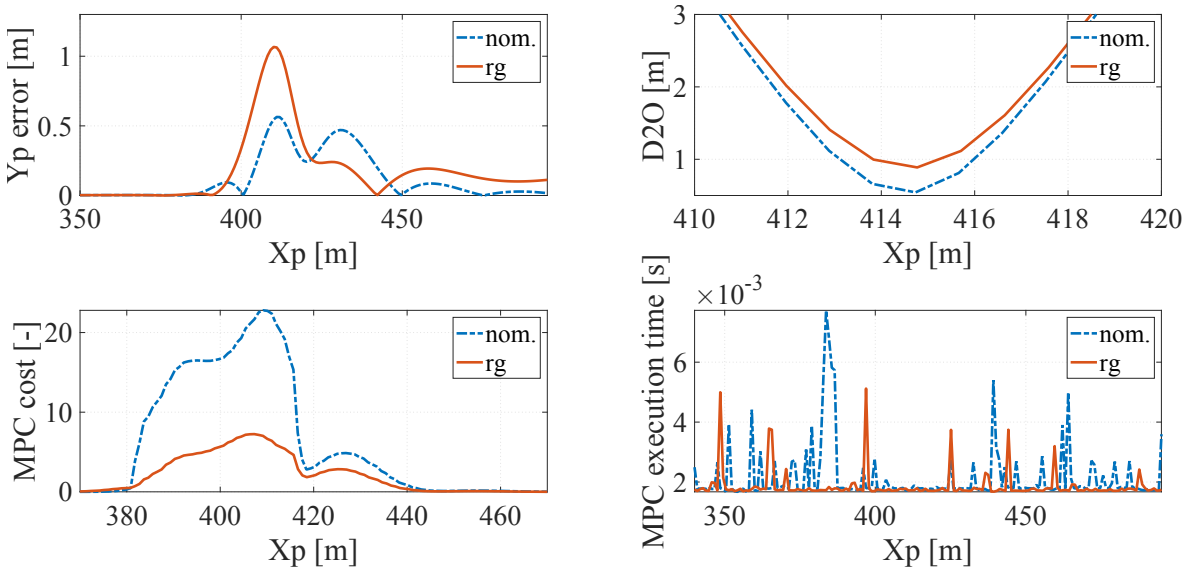


Figure 7-5: Tracking error, obstacle distance, controller cost and execution time at 70 km/h.

	x_r [m]	M_p [%]	x_s [m]	y_{RMSE} [m]	$y_{\text{RMSE}}^{\text{pre}}$ [m]	$y_{\text{RMSE}}^{\text{post}}$ [m]	D2O_{min} [m]
MPC	16.75	12.43	114.60	0.193	0.174	0.212	0.543
BORG	19.70	5.54	135.94	0.294	0.365	0.187	0.887

Table 7-2: Key performance indicators at 70 km/h.

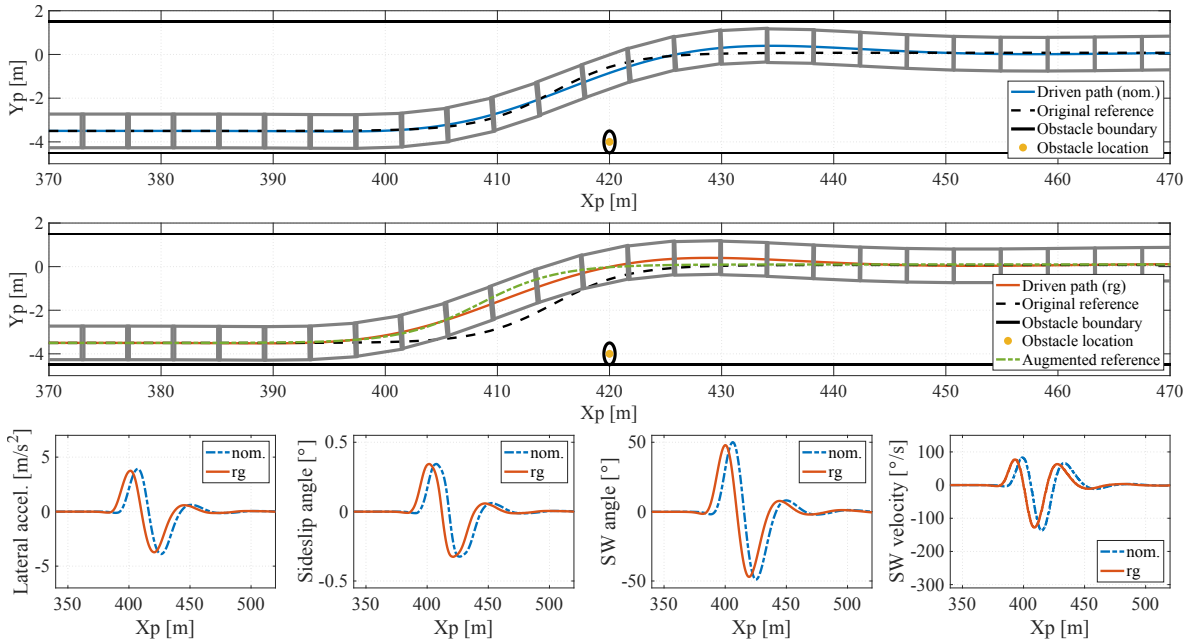


Figure 7-6: Trajectory performance comparison for MPC and BORG at 60 km/h.

	x_r [m]	M_p [%]	x_s [m]	y_{RMSE} [m]	$y^{\text{pre}}_{\text{RMSE}}$ [m]	$y^{\text{post}}_{\text{RMSE}}$ [m]	D2O_{min} [m]
MPC	17.00	8.91	144.21	0.066	0.060	0.072	1.468
BORG	17.77	9.09	139.22	0.2246	0.374	0.094	2.327

Table 7-3: Key performance indicators at 60 km/h.

and less aggressive actuation. Execution times and controller cost values remain favourable, confirming the BORG's computational efficiency.

7-1-4 Lane Change Performed at 60 km/h with Obstacle at (420, -4) m

The final case study involves a further reduction in speed to 60 km/h, with the obstacle placed at a lateral distance of -4 m. At this velocity, the vehicle dynamics enter a relatively linear operating regime, and both controllers benefit from increased temporal and spatial flexibility. As a result, the nominal MPC already performs well, exhibiting reduced overshoot and adequate obstacle avoidance.

Nonetheless, the BORG continues to deliver meaningful improvements. Although the manoeuvre aggressiveness is similar, the reference trajectory produced by the governor diverges earlier from the nominal path. This early initiation maximises the available clearance to the obstacle, resulting in a significantly increased minimum distance of $\text{D2O}_{\text{min}} = 2.327$ m compared to 1.468 m for the MPC. The dominant term in the objective function at low speeds is the obstacle clearance, which explains the more cautious behaviour of the BORG.

Although the total RMSE is higher for BORG due to earlier deviation from the nominal path, the post-obstacle tracking is again better. Control signals remain smooth, and computational times remain within acceptable bounds. The consistency of these trends across all three

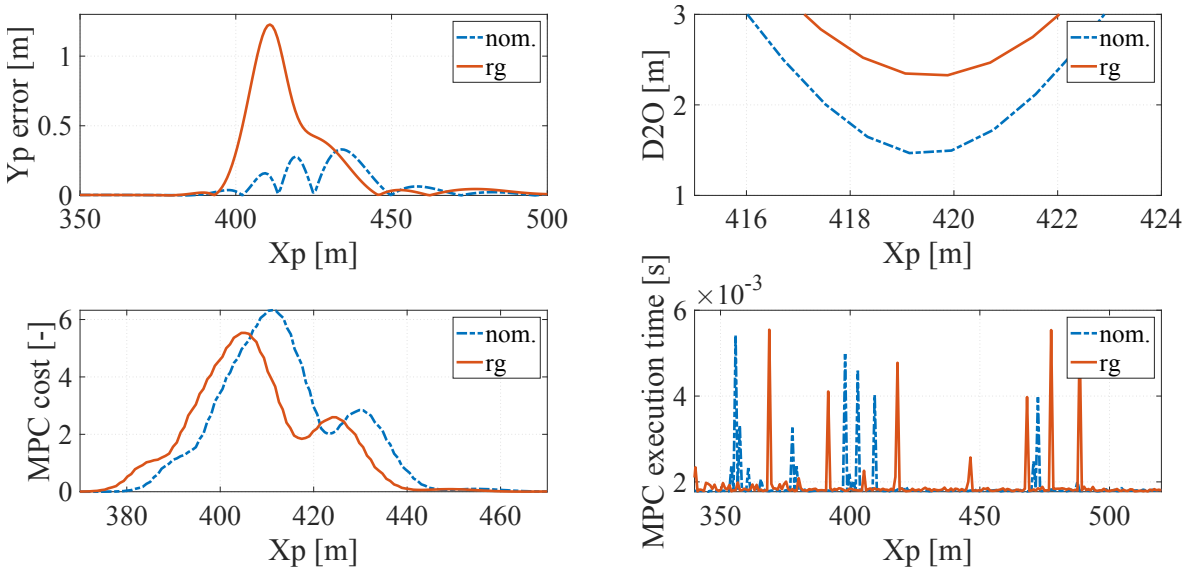


Figure 7-7: Tracking error, obstacle distance, controller cost and execution time at 60 km/h.

velocity regimes further strengthens the claim that BORG enhances safety and comfort with minimal cost in control effort or computational complexity.

7-2 Sensitivity and Robustness Analysis

This section investigates how each controller performs in the presence of real-world uncertainties, such as mismatches in the vehicle model and perception-related errors. These tests are critical for assessing the viability of the proposed approach in deployment scenarios where perfect conditions cannot be guaranteed.

7-2-1 Tyre Model Mismatch

Model-based controllers such as MPC rely on internal models that represent the physical behaviour of the vehicle. However, in practice, this model is never perfect. Variations in tyre characteristics due to wear, temperature, inflation levels, or surface conditions can result in discrepancies that compromise controller performance. To study this, a set of 100 randomized tyre parameter configurations were generated using Weibull and Gaussian distributions, as visualized in Figure 7-8. These variations reflect realistic deviations that may occur during everyday operation.

Each sample was used to evaluate both controllers over a range of longitudinal velocities. As shown in Figure 7-9, the BORG consistently maintains a larger average minimum distance to the obstacle, across all velocity conditions. This highlights its robustness in handling model mismatch.

This benefit arises from the online adaptation capability of the BORG, which shapes the reference trajectory based on performance, rather than relying solely on an accurate dynamic

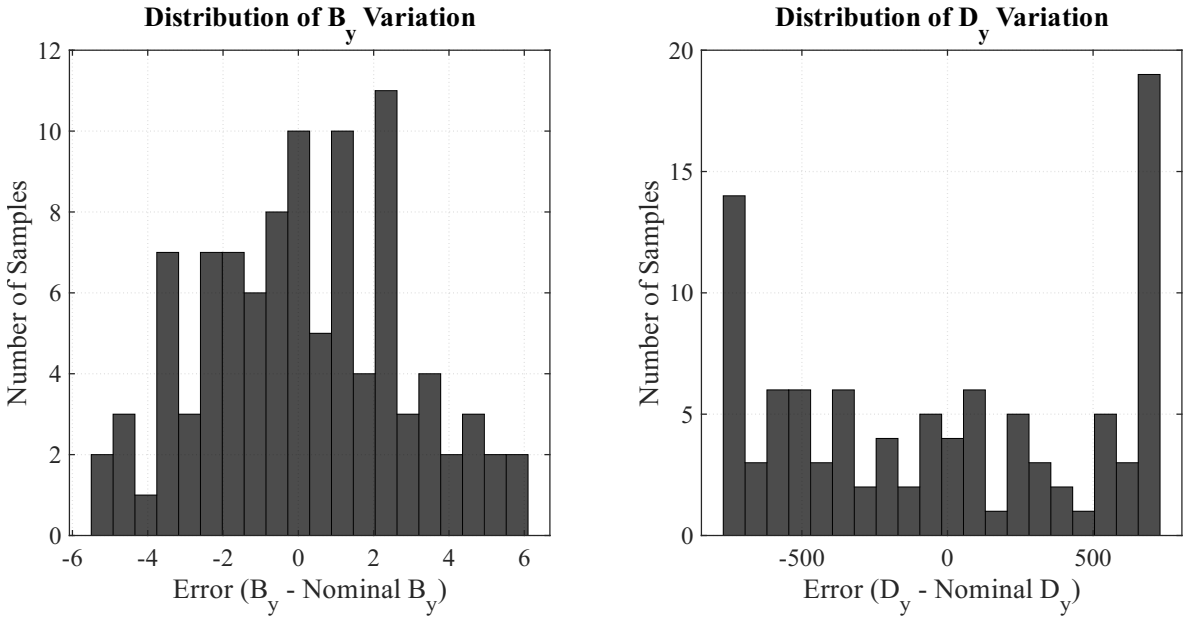


Figure 7-8: Distribution of 100 samples of tyre parameter mismatches.

model. By leveraging the GP model's predictions, the controller implicitly compensates for un-modelled dynamics without requiring direct tuning or redesign.

Importantly, none of the tested mismatches resulted in collisions or near-misses, hence these metrics are not reported for this subsection. However, the improvement in clearance margins directly translates to better safety and increased robustness in uncertain environments.

7-2-2 Localization Mismatch

Perception errors represent another major challenge for autonomous systems. Imperfect ego vehicle or obstacle localization due to sensor noise, latency, or GPS drift can lead to degraded control performance or even unsafe behaviour.

To simulate this, zero-mean Gaussian noise was injected into the perception signals of both the ego vehicle and detected obstacles. Figure 7-10 illustrates the distribution of 100 such noise samples used in the evaluation, selected to be representative of real-world GPS and sensor fusion systems [61].

Under these noisy conditions, performance is assessed using two critical safety indicators: the Near Miss Rate (NMR) and Collision Rate (CLR). The results, summarised in Figure 7-11, show a clear benefit of using BORG. Across all scenarios, it reduces both near misses and actual collisions compared to the nominal MPC.

As vehicle velocity increases, both NMR and CLR worsen for both controllers, due to the reduced reaction time and tighter constraints. However, BORG retains a significant safety buffer by planning trajectories more cautiously and earlier. This behaviour arises from the surrogate model's learned sensitivity to trajectory safety, resulting in wider safety margins in uncertain contexts.

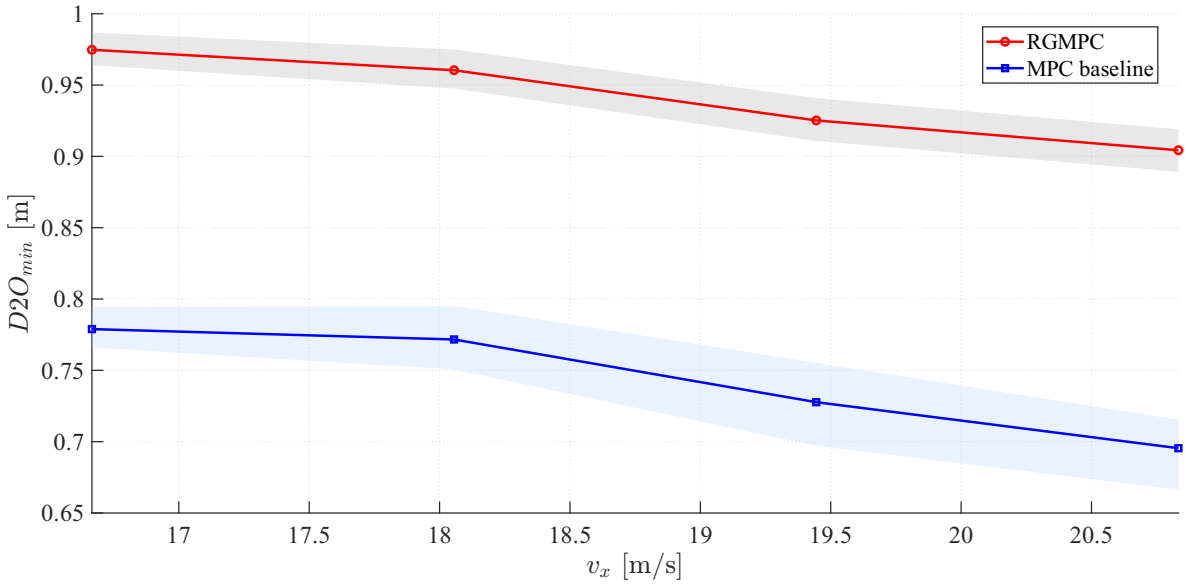


Figure 7-9: Average minimum distance to obstacle under tyre parameter mismatch for nominal MPC and BORG.

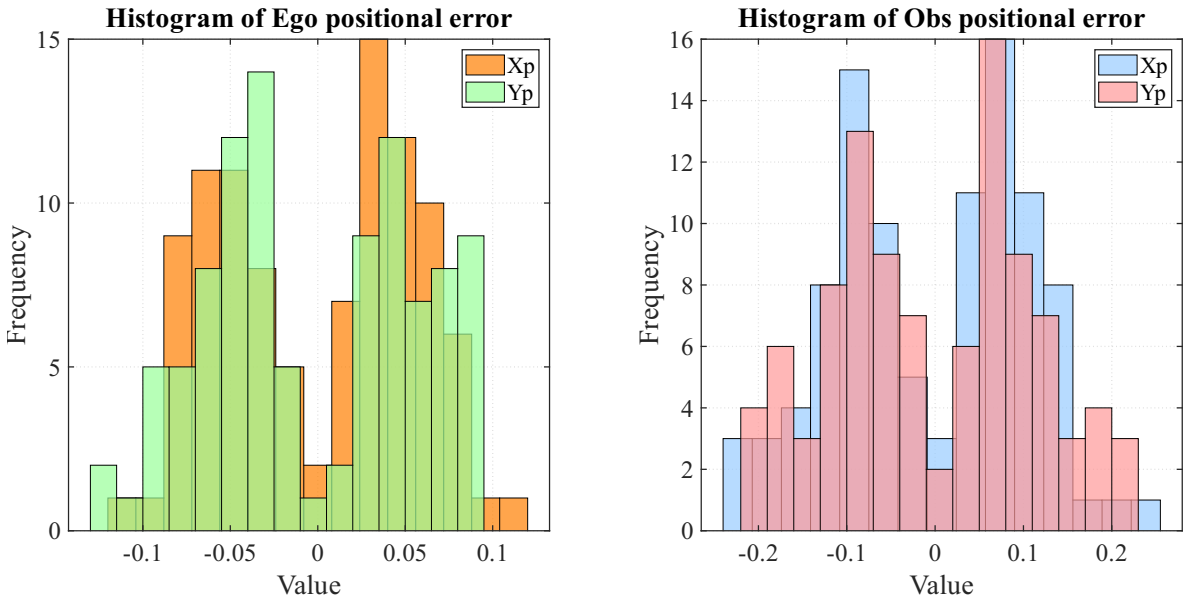


Figure 7-10: Distribution of 100 samples of ego vehicle and obstacle location mismatches.

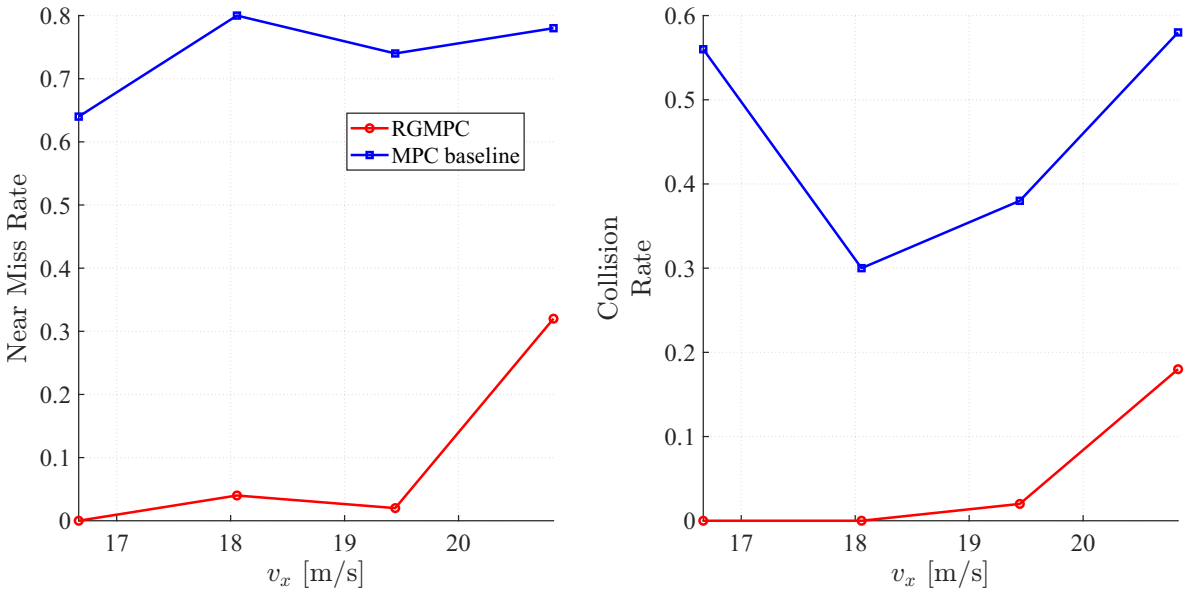


Figure 7-11: Near miss and collision rates under perception noise for MPC and BORG.

In summary, the robustness analysis confirms the advantage of combining model-based predictive control with a model-free reference adaptation mechanism. The BORG architecture offers improved safety and reliability, even when the system is subjected to significant modelling or sensing inaccuracies. These capabilities make the approach highly promising for real-world autonomous driving tasks, where uncertainty is unavoidable.

Chapter 8

Discussion & Conclusion

8-1 Discussion

The results presented in this thesis demonstrate the potential of the Bayesian Optimisation-based Reference Governor framework in enhancing the performance of non-linear Model Predictive Control (MPC) for autonomous vehicle motion planning. In particular, the BORG successfully improves both trajectory tracking and obstacle avoidance through context-aware reference adaptation. This is achieved via the optimisation of sigmoid trajectory parameters in a data-efficient, model-free manner using Gaussian Process (GP) models.

One of the key findings is that the reference governor significantly reduces overshoot and lateral tracking errors during evasive manoeuvres. By optimising trajectory parameters offline, building a surrogate model and evaluating said model in real-time based on the current context, the system effectively modulates manoeuvre aggressiveness. The reference signals produced by the Reference Governor (RG) are thus filtered to be controller friendly, resulting in smoother vehicle responses that are dynamically feasible and easier for the controller to track, enhancing robustness in scenarios with perception errors or model mismatch.

However, this replanning effect also introduces a trade-off: the reference governor acts as a form of comfort filter, reducing the aggressiveness of the manoeuvre. While this can be desirable in terms of passenger comfort and control feasibility, it may not fully exploit the dynamic capabilities of the vehicle in emergency situations.

Additionally, a key limitation of the current implementation is that the reference is generated in a one-shot fashion at the beginning of the manoeuvre. Once selected, the reference trajectory is not further adapted during the course of execution. This could limit performance in highly dynamic or uncertain environments where real-time replanning may be essential. Furthermore, the current method optimises fixed sigmoid parameters rather than performing continuous reparametrisation or using more expressive trajectory representations like splines.

8-2 Future Work

Several directions can be explored to further enhance the effectiveness of the BORG framework:

- **Online Replanning:** Integrate continuous reference adjustment throughout the manoeuvre execution to better respond to dynamically evolving environments.
- **Trajectory Representation:** Replace the fixed sigmoid reference formulation with more flexible parametrisations, such as piecewise splines or Bézier curves, which can capture a broader class of manoeuvres.
- **Sparse Gaussian Processes:** To address the computational scalability of GPs in higher-dimensional contexts or larger datasets, sparse GP approximations can be considered.
- **Multi-Objective Optimisation:** Extend the cost function to include additional objectives such as energy consumption or ride comfort, balancing safety and efficiency.
- **Constrained acquisition functions:** Integrate constraints in the acquisition function, such as Constrained Expected Improvement, to guarantee that the reference augmentation produced by the governor is safe, or to additionally enforce comfort or manoeuvre aggression constraints.

8-3 Conclusions

This thesis proposed and validated a novel control architecture for automated vehicle manoeuvring based on a Bayesian Optimisation-based Reference Governor integrated within a non-linear MPC framework. The main contributions and findings are summarised as follows:

- A data-driven reference governor capable of optimising sigmoid trajectory parameters was developed, enabling dynamically feasible and control-friendly reference signals.
- The BORG framework was shown to significantly improve tracking performance, particularly by reducing overshoot and trajectory error during obstacle avoidance scenarios.
- Through the use of Gaussian Processes and Bayesian Optimisation, the proposed method operates in a model-free manner, ensuring robustness to system uncertainties and to a certain extent, compensation of modelling errors.
- Experimental results demonstrated that the BORG provides a computationally tractable and effective alternative to conventional replanning techniques, though at the cost of reduced manoeuvre aggressiveness and one-shot planning limitations.

Overall, the Bayesian Optimisation Reference Governor framework bridges the gap between optimal control and motion planning, offering a promising direction for enhancing safety and performance in high-speed, real-time automated driving applications.

Appendix A

Vehicle parameters and controller settings

A-1 Prediction horizon and sampling time

The selection of MPC parameters such as the sampling time t_s and prediction horizon N_p plays a crucial role in balancing computational efficiency and control performance. A longer prediction horizon generally improves tracking performance but increases computational complexity, which is critical for real-time implementation. Therefore, tuning t_s and N_p requires careful consideration of both system dynamics and processing limitations.

In this work, the sampling time t_s was selected based on the update rates of signals available on the vehicle's CAN network. By analysing the ECU signal update frequencies—including acceleration, yaw rate, brake pressure, and reference trajectory it was determined that the slowest relevant signal updated every 0.032 seconds. To ensure the MPC had access to all required data at each step, t_s was set to 0.035 seconds, following the design considerations discussed in [51].

A-2 Vehicle parameters

A high-fidelity vehicle model of a small SUV, developed by Toyota Motor Europe, was used for simulation studies, as described in [51]. This model was implemented in the IPG CarMaker simulation environment and includes detailed representations of various vehicle subsystems, including suspension, steering, powertrain, and braking systems. The tire behaviour is modelled using a physics-based approach that accounts for non-linearities and transient dynamics, enabling the reproduction of realistic driving responses under different road and manoeuvring conditions. This comprehensive system-level model serves as a virtual test bench for evaluating the real-time performance and robustness of advanced vehicle control algorithms.

In addition to the full vehicle model, a simplified version with adjusted parameters was also utilized during the early stages of controller design and sensitivity analysis. This reduced-complexity model, based on the bicycle model formulation, enables faster simulation cycles and analytical insights while maintaining the essential dynamic behaviour necessary for developing and tuning the control strategies. By switching between the detailed and simplified models, a balance between simulation speed and fidelity was achieved, supporting both conceptual development and validation phases.

Table A-1: IPG CarMaker model vehicle parameters

Parameter	**Explanation**	**Value**
$C_{r,f}$	Front roll stiffness	42307 [Nm/rad]
$C_{r,r}$	Rear roll stiffness	36039 [Nm/rad]
h_{cg}	Height of CoG	0.673 [m]
h_f	Front roll height	0.27 [m]
h_r	Rear roll height	0.28 [m]
I_z	Moment of Inertia around z-axis	3386 [$\mathrm{kg} \cdot \mathrm{m}^2$]
K_h	Stability factor	0.004 [$\mathrm{s}^2/\mathrm{m}^2$]
l_f	Distance between front axle and CoG	1.093 [m]
l_r	Distance between rear axle and CoG	1.570 [m]
m	Vehicle mass	1712 [kg]
$R_{\mathrm{eff},f}$	Front effective tire radius	0.359 [m]
$R_{\mathrm{eff},r}$	Rear effective tire radius	0.353 [m]
S_{rat}	Steering ratio	15.8 [−]
t_f	Front track width	1.628 [m]
t_r	Rear track width	1.635 [m]

Appendix B

IEEE Robotics and Automation Letters Paper

Efficient Data-driven Reference Governor Design for Safe Evasive Manoeuvring

Petar Velchev

***Abstract*—This paper presents a novel data-driven Reference Governor with Model Predictive Control integrating local motion replanning and path following for collision avoidance. Employing a model-free Reference Governor, the proposed solution utilises system knowledge through Bayesian Optimisation to augment predetermined evasive trajectories, minimising path-following errors and simultaneously ensuring obstacle safety margins. A single-track vehicle model in combination with non-linear tyre models is used to capture the vehicle's dynamics. The optimised control action is the vehicle steering angle, whilst the Reference Governor optimises parameters of a sigmoid reference signal to minimise the tracking error and guarantee safety with respect to obstacles in emergency manoeuvres. The proposed approach is evaluated on a single lane change using a high-fidelity simulation environment and its performance is compared to a baseline controller integrating path following and obstacle avoidance. The results show a 14% reduction of safety critical overshoot, maximising obstacle safety distance and a four times lower controller cycle time compared to the baseline. Furthermore, through a robustness analysis, it is demonstrated that the proposed approach is more robust towards model mismatches and perception-based errors, as seen by average 30% and 40% reductions in near-miss and collision rates.**

***Index Terms*—Obstacle avoidance, model predictive control, reference governor, path following, Bayesian optimisation.**

I. Introduction

AUTOMATED vehicles operating near the handling limits must be capable of executing evasive manoeuvres with high precision and robustness to avoid collisions in dynamic and uncertain environments. In such scenarios, due to complex non-linearities, deviations from a reference path can result in safety-critical situations, making the design of accurate and adaptable path following systems essential [1], [2].

A key challenge is the ability to locally and rapidly re-plan trajectories while maintaining safe distances from obstacles. This is critical for advanced driver assistance and autonomous driving systems, particularly in emergencies requiring fast, reactive decisions near the vehicle's handling limits. Applications include urban scenarios with unpredictable pedestrians, highway lane incursions, and last-moment evasive manoeuvrers. However, re-planning and path following at the limit of handling is inherently challenging due to the highly non-linear vehicle dynamics and the need to balance tracking performance with safety constraints in real-time [3], [4]. Classical motion planning and trajectory tracking methods assume linear dynamics through the use of basic point mass models, which fail to capture the complex interactions between control inputs, tyre forces, and vehicle responses under high slip or load transfer conditions [5], [6]. Furthermore, incorporating real-time re-planning into a control pipeline with minimal computational burden remains a persistent difficulty

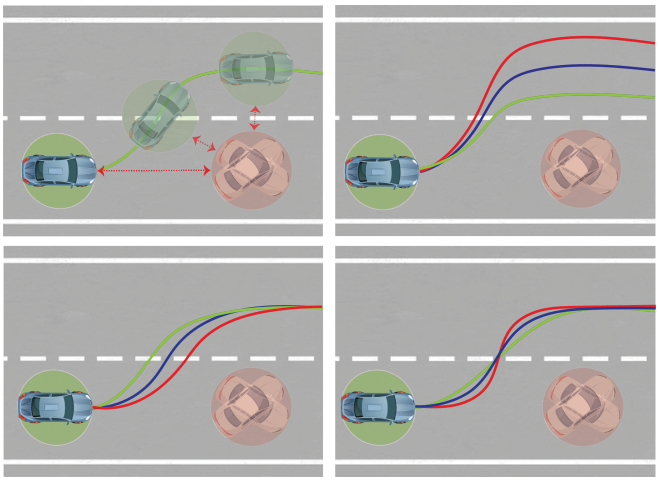


Fig. 1: Obstacle avoidance lane change scenario. The top left diagram depicts vehicle to obstacle distance (V2O) using red arrows for a target manoeuvrer shown in green. The top right, bottom left and bottom right diagrams show a variation in end location, starting point and steepness of manoeuvre.

[7], requiring dedicated hardware instead of implementing on existing less powerful in-vehicle control units.

To address these challenges, this paper proposes a novel control architecture combining a model predictive controller (MPC) with a model-free Bayesian Optimisation-based Reference Governor (BO-RG) as inspired by [8]. The MPC is responsible for tracking a reference trajectory by computing the optimal steering input, using a bicycle vehicle model with non-linear tyre dynamics to balance model complexity and computational efficiency. The RG optimizes the parameters of a sigmoid-shaped evasive reference trajectory, visualised in Fig. 1 using BO. This decoupled architecture enables the system to adaptively adjust the planned manoeuvre in real-time, reducing tracking error by compensating for model mismatch and improving vehicle to obstacle distance while maintaining computational efficiency.

II. Related Work

Efficient obstacle avoidance in automated vehicles often relies on simplified models such as kinematic bicycles or point-mass approximations. These models are favoured for their low computational cost and suitability for real-time use in structured or low-speed scenarios [5], [6]. While effective in linear regime of motion, these models fail to capture essential dynamics, e.g. tyre slip, load transfer, and actuator limits, critically under aggressive or high-speed manoeuvres. This

oversimplification [9] leads to unsafe or infeasible trajectories due to poor planning-control consistency. Similarly, neglecting high-fidelity dynamics [10] often results in overly conservative or suboptimal paths, particularly in cluttered or high-curvature environments. Recent work [11] has shifted toward hybrid approaches, combining accurate dynamic models or learning-based surrogates with simplified planners to improve robustness without sacrificing real-time feasibility. To achieve similar performance improvements, data-driven methods have also been applied for predictive model correction [12], [13] as well as data-driven controller tuning [14], [15]. Reference Governors enforce system constraints by modifying reference inputs, enabling safe control without re-optimising at every step. Initially proposed for linear unconstrained systems [16], RGs have since been applied to vehicle steering to maintain safety limits like lateral acceleration and yaw rate [17]. RGs can also function as pre-filters for MPC, reducing constraint violations and computational load by adjusting references before they reach the controller [18]. Recent work explores learning-based RGs that derive safe sets from data, reducing reliance on explicit models and improving adaptability in uncertain, real-time environments [19]. In summary, simplified re-planners lack fidelity near handling limits, while traditional RGs are generally adopted for constraint enforcement in unconstrained systems, may require accurate models or lack adaptability. Hybrid or data-driven methods [20], [21] offer promising solutions for robust, real-time control for use in the domain of automated vehicles.

Motivated by the above analysis, the proposed BO-RG architecture addresses path following and obstacle avoidance challenges by integrating MPC with a model-free Bayesian Optimisation-based Reference Governor which adapts evasive manoeuvres in real-time. The BO-RG provides a single-shot reference signal augmentation to the online MPC, by optimising sigmoid defining parameters of the target lane change using an offline pre-trained surrogate Gaussian Process (GP) model. The RG depicted in Fig. 2 takes as inputs the original lane change reference signal as well as system states and obstacle localisation information to augment the reference. The resulting lane change target is fed an input to the MPC, which in turn computes the optimal control actions to drive the vehicle to the target lateral position. The contributions of this paper are threefold.

1) Development of efficient data-driven RG based on Bayesian optimisation, integrated with MPC framework, with four times lower cycle time of a baseline which integrates path following and obstacle avoidance in the same cost function.
2) The BO-RG can be trained to consider the handling limit and make the trajectory smoother, reducing the overshoot of a baseline by 14%.
3) Improve robustness against model mismatch and uncertainty through the use of model-free re-planning by an average 30% reduction on collision rate and 40% reduction in near miss rate across a range of velocities.

III. Reference Governor Design Problem

In the system depicted in Figure 2, the controller computes control inputs to track a given reference while satisfying system constraints. However, if the reference is too aggressive, it may lead to infeasible or unsafe behaviour. To address this, an RG is introduced to modify the reference in real-time based on the vehicle state and constraints. As shown in Figure 2, the RG outputs a feasible reference $r^*(t)$, ensuring safe and constraint-compliant tracking by the controller. Consider the generic closed-loop system in Figure 2 designed for controlling a non-linear plant, let $\theta \in \Theta$ be the vector of parameters describing the reference target signal, where $\Theta \subseteq \mathbb{R}^{n_\theta}$ is the set of reference parameters. The parameters should be configured to optimize the overall closed-loop performance while ensuring the system's safety and desired tracking behaviour. By analysing the tracking error, which represents the difference between the reference and output signals, various metrics can be introduced to assess tracking quality, such as overshoot magnitude, settling time, steady-state error, rise time and other application specific indicators. Given a fixed control structure, the tracking quality and errors depend mainly on the vector of parameters θ which reshape the reference trajectory.

Accordingly, each of the above performance metrics is a function of θ, denoted by $c_i : \Theta \to \mathbb{R}$, for $i = 1, \ldots, n_c$. The overall performance metric, $f : \Theta \to \mathbb{R}$, is defined as a weighted sum of the individual performance metrics:

$$f(\theta) := w^T c(\theta) = \sum_{i=1}^{n_c} w_i c_i(\theta), \quad \theta \in \Theta,$$

where $c(\theta) := [c_i(\theta)]_{i=1}^{n_c}$ and $w := [w_i]_{i=1}^{n_c}$. The weights $w_1, \ldots, w_{n_c}$ are determined based on the relevance and scale of the associated performance metrics. Given the defined performance metrics, we can formalize the reference augmentation problem. Specifically, to find the optimal reference parameters θ^* while maintaining safety, the following optimisation problem is solved:

$$\min_{\theta \in \Theta} f(\theta) = w^T c(\theta)$$

The objective function f is not analytically computable, i.e., it lacks tractable closed-form expressions, even if the system dynamics are well understood. For each reference parameter θ, an experiment must be performed on the system to measure the tracking error signal and subsequently evaluate $f(\theta)$. Therefore, these functions are accessible only through a black-box oracle. To augment the reference parameters according to the optimisation problem above, we employ BO, a data-driven method for solving optimisation problems where the objective function and constraints are represented as expensive-to-evaluate black-box functions.

The design of the RG through use of Bayesian optimisation is described in the following sections and its application is demonstrated by integration with an MPC vehicle motion control scheme.

IV. Governed Vehicle Motion Control System Design

The proposed Reference Governor for a Model Predictive controller as depicted in Fig. 2 to meet tracking and obstacle

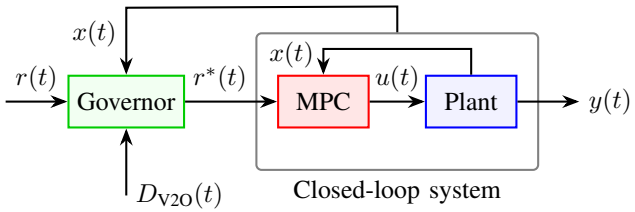


Fig. 2: Block diagram of the proposed control system with governor, controller, and plant.

safety performance goals. This section first describes the predictive model, the setup of the optimal control problem, the Bayesian optimisation formulation and lastly the utilisation of the RG. The latter is split into two phases:

- Offline: RG is trained using BO through the use of GP surrogate modelling to augment reference parameters.
- Online: the trained GP surrogate model is used as a reference augmentation filter in an online fashion.

A. Vehicle Predictive Model

The vehicle dynamics are modelled using the non-linear single track model, which captures the lateral, yaw, and longitudinal behaviour of the vehicle in a planar environment. The model assumes a single front and rear wheel to simplify the analysis while preserving key dynamic characteristics. The system states include the longitudinal and lateral velocities (v_x, v_y), yaw rate (r), yaw angle (ψ), global position coordinates (x_p, y_p), and steering angle (δ). Tyre lateral forces are represented using a linear tyre model with cornering stiffness coefficients C_{α_f} and C_{α_r} [22].

The resulting equations of motion describe the time evolution of the vehicle's dynamics under the influence of steering inputs and inertial properties, as illustrated in Fig. 3.

$$\begin{aligned}
\dot{v}_x &= v_y r \\
\dot{v}_y &= -\frac{C_{\alpha_f}+C_{\alpha_r}}{m v_x} v_y + \frac{L_r C_{\alpha_r} r - L_f C_{\alpha_f}}{m v_x} - v_x r + \frac{C_{\alpha_f}}{m}\delta \\
\dot{r} &= \frac{L_r C_{\alpha_r} - L_f C_{\alpha_f}}{I_z v_x} v_y - \frac{L_r^2 C_{\alpha_r} + L_f^2 C_{\alpha_f}}{I_z v_x} r + \frac{L_f C_{\alpha_f}}{I_z}\delta \\
\dot{\psi} &= r \\
\dot{x}_p &= v_x \cos(\psi) - v_y \sin(\psi) \\
\dot{y}_p &= v_x \sin(\psi) + v_y \cos(\psi) \\
\dot{\delta} &= d_\delta
\end{aligned} \tag{1}$$

The cornering stiffnesses C_{α_f} and C_{α_r} are approximations of the tyre lateral force response at small slip angles, derived from the Magic Formula tyre model. The Magic Formula describes the non-linear relationship between tyre slip angle and lateral force, typically expressed as $F_y = D\sin(C\arctan(B\alpha - E(B\alpha - \arctan(B\alpha))))$, where α is the slip angle, and B, C, D, and E are empirical fitting factors [23]. The cornering stiffness is obtained through this relationship, yielding an approximation of the tyre behaviour in the small-slip regime, allowing the non-linear single track model-based controller to remain computationally efficient while retaining physical accuracy. Additionally, the Dugoff tyre model is also utilised in the predictive model in combination with a high-fidelity vehicle model [24]. This system is used to evaluate the performance of the proposed BO-RG with a still computationally efficient MPC for real-time implementation but in combination with a very accurate vehicle simulation as a plant.

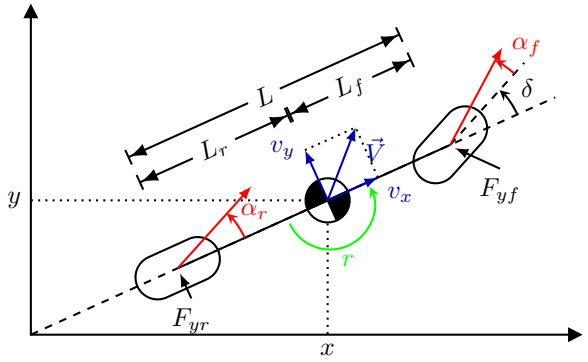


Fig. 3: Bicycle Model Free Body Diagram.

B. Optimal Control Problem

The presented optimal control problem is formulated for use within a Model Predictive Control (MPC) framework to enable safe and smooth trajectory tracking with obstacle avoidance.

Accordingly, given suitable sampling time t_s and prediction horizon N, the MPC cost function J is defined as

$$\begin{aligned}
J = \sum_{i=1}^{N} \Big(& q_{e_{Yp}} e_{Yp,i}^2 + q_{e_{Xp}} e_{Xp,i}^2 + q_{\dot{\delta}} \dot{\delta}_i^2 + q_{\dot{F}_x} \dot{F}_{x_i}^2 + \\
& + \sum_{j=1}^{N_{\text{obs}}} \left(q_{e_{\text{V2O}}} e_{\text{V2O},j,i}^2 \right) \Big)
\end{aligned} \tag{2}$$

where $e_{Yp,i}$ and $e_{Xp,i}$ represent the lateral and longitudinal position tracking errors at time step i, weighted by $q_{e_{Yp}}$ and $q_{e_{Xp}}$, respectively, $\dot{\delta}_i$ is the steering rate, penalized through $q_{\dot{\delta}}$ to ensure smooth steering actions [25], $\dot{F}_{x_i}$ denotes the longitudinal force rate, penalized by $q_{\dot{F}_x}$ to prevent aggressive acceleration or braking, and, $e_{\text{V2O},j,i}$ is the distance-based error term representing proximity to the j-th obstacle at time step i, weighted by $q_{e_{\text{V2O}}}$, to enforce obstacle avoidance.

The obstacle distance metric is defined by:

$$D_{\text{V2O}} = \left((X - X_{\text{obs}})^2 + (Y - Y_{\text{obs}})^2 \right)^{\frac{1}{2}} - r_{\text{obs}} - r_{\text{veh}} \tag{3}$$

This represents the Euclidean distance between the vehicle and the obstacle centres, reduced by their respective safety radii (r_{obs} and r_{veh}). This term encourages the controller to maintain a safe buffer from all detected obstacles [12].

To ensure vehicle stability, constraints are imposed on the side slip angle and its rate (bounded within $\pm 5°$ and $\pm 25°/\text{s}$, respectively), steering wheel angle and velocity (limited to ± 2.76 turns and $\pm 800°/\text{s}$), and lateral acceleration (bounded within $\pm 0.85\mu g$). This constraint reflects the fact that the required yaw rate to achieve a lateral manoeuvre cannot always

be achieved due to the road friction coefficient not being able to provide the required tyre forces. Therefore, the yaw rate is bounded by a function based on the friction coefficient through a safety factor [26]. Furthermore, the limits reflect physical actuator capabilities and define a stable operating envelope during aggressive manoeuvrers [22].

C. Optimisation of Sigmoid Parameters via Bayesian Optimisation

Bayesian Optimisation (BO) is used in this work to efficiently tune the parameters $\theta = [\theta_1, \theta_2, \theta_3]$ of the sigmoid reference trajectory defined in Eq. (11), which governs the final lateral offset, spatial centre, and transition aggressiveness of a lane change manoeuvre. The goal is to find the parameter vector θ^* that minimizes a closed-loop performance cost, enhancing both path following and obstacle avoidance performance.

Given an objective function $f(\theta)$ that evaluates the performance of a closed-loop manoeuvre with a given sigmoid parametrization, the optimisation problem is formulated as:

$$\theta^* = \arg\min_{\theta \in \Theta} f(\theta), \tag{4}$$

where Θ is the feasible parameter domain.

To solve this, a Gaussian Process (GP) surrogate model is trained to approximate $f(\theta)$ based on sampled evaluations [27]. BO leverages this surrogate model to select new candidate parameters by optimising an acquisition function that balances exploration of uncertain regions and exploitation of promising areas. Specifically, the Expected Improvement (EI) acquisition function [28] is used, defined as:

$$\alpha_{\text{EI}}(\theta) = \mathbb{E}[\max(0, f_{\min} - f(\theta))], \tag{5}$$

where $f_{\min}$ is the best observed value of the objective function.

The surrogate GP model provides a posterior mean $\mu_n(\theta)$ and variance $\sigma_n^2(\theta)$ for any θ, allowing the EI to be computed in closed form under the Gaussian assumption:

$$\alpha_{\text{EI}}(\theta) = (f_{\min} - \mu_n(\theta))\Phi(Z) + \sigma_n(\theta)\varphi(Z), \tag{6}$$

where $Z = \frac{f_{\min} - \mu_n(\theta)}{\sigma_n(\theta)}$, and $\Phi(\cdot)$ and $\varphi(\cdot)$ denote the cumulative distribution and probability density functions of the standard normal distribution, respectively.

The next query point θ_{m+1} is selected by maximizing the acquisition function:

$$\theta_{m+1} = \arg\max_{\theta \in \Theta} \alpha_{\text{EI}}(\theta), \tag{7}$$

which is performed using a genetic algorithm [29] to efficiently search the parameter space.

The GP surrogate model assumes:

$$f(\theta) \sim \mathcal{GP}(m(\theta), k(\theta, \theta')), \tag{8}$$

where $m(\theta)$ is the mean function, typically assumed zero, and $k(\theta, \theta')$ is the covariance function or kernel. The Squared Exponential kernel is used, given by:

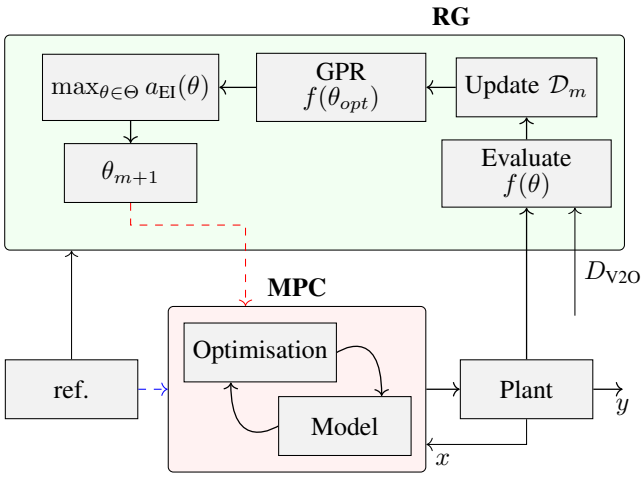


Fig. 4: Schematic of Bayesian Optimisation loop interacting with MPC for path following and obstacle avoidance. In the offline training phase only the RG reference is fed to the MPC depicted by a red dashed line. During online operation, the original reference in blue is fed until obstacle detection, prompting the RG augmentation.

$$k(\theta, \theta') = \sigma_f^2 \exp\left(-\frac{(\theta - \theta')^2}{2l^2}\right), \tag{9}$$

where σ_f^2 controls the function's variance and l determines the length scale of smoothness.

GP hyperparameters, including length scale l, signal variance σ_f^2, and noise variance σ_n^2, are optimised by maximising the log marginal likelihood [30]:

$$\log p(y|\Theta, \vartheta) = -\frac{1}{2} y^T K^{-1} y - \frac{1}{2}\log|K| - \frac{n}{2}\log 2\pi, \tag{10}$$

where $\vartheta = \{l, \sigma_f^2, \sigma_n^2\}$, K is the covariance matrix over observed samples, and y are the corresponding function evaluations.

This BO framework is applied to generate an optimised sigmoid trajectory that serves as a controller-friendly reference signal, balancing tracking precision, control effort, and obstacle clearance, without resorting to computationally expensive exhaustive search.

$$y(x) = \frac{\theta_1}{1 + e^{-\theta_2(x - \theta_3)}} \tag{11}$$

The sigmoid reference trajectory defined in Eq. (11) depends on the parameter vector $\theta = [\theta_1, \theta_2, \theta_3]$, which governs the final lateral offset, the spatial centre of the manoeuvrer, and the steepness of the transition, respectively. These parameters directly influence the ability of the controller to achieve accurate and smooth path following while maintaining obstacle avoidance.

To augment θ for optimal control performance, we formulate a BO problem aimed at minimising a closed-loop cost function $J(\theta)$ over a set of candidate parameters as per Eq. 4 where Θ denotes the admissible domain for the

parameters, and $J(\theta)$ is evaluated by simulating the closed-loop behaviour of the MPC with the sigmoid-based reference. The cost function aggregates performance criteria such as tracking accuracy, control effort, and additional obstacle safety through:

$$f(\theta) = w_J J_T(\theta) + w_{Yp}\bar{e}_{Yp}^2 + w_{Xp}\bar{e}_{Xp}^2, \tag{12}$$

where $\bar{e}_{Yp}$ and $\bar{e}_{Xp}$ are the root mean square of the longitudinal and lateral tracking errors, respectively, and $J_T(\theta)$ is the sum of the MPC cost as defined in Eq. 2 for the total manoeuvre. This objective function formulation ensures that the Bayesian Optimisation minimum at θ produces a reference which minimizes tracking error, maximizes ego vehicle to obstacle distance and additionally enforces that the augmented reference deviates as little as possible from the nominal reference.

In turn, BO is used to efficiently explore the parameter space Θ, relying on a GPR to model the unknown objective function $f(\theta)$. At each iteration m, the GP provides a predictive posterior distribution characterized by a mean $\mu_m(\theta)$ and variance $\sigma_m^2(\theta)$, which are used to compute the Expected Improvement acquisition function as per Eq. 6 and the next candidate parameter vector is selected by maximising this acquisition.

The overall parameter tuning process consists of the following steps:

1) Sample an initial set of parameter vectors $\{\theta_i\}_{i=1}^{m_0}$ using Latin Hypercube Sampling or random sampling.
2) Evaluate $f(\theta_i)$ for each θ_i by running closed-loop simulations.
3) Fit a GP surrogate model to the collected data.
4) Iteratively select new candidates by maximising the acquisition function and updating the dataset.

This iterative optimisation efficiently converges to optimal sigmoid parameters θ^* that enhance controller performance while avoiding exhaustive search over the parameter space. The process is summarized in Algorithm 1.

The interaction between the Bayesian optimiser and the MPC system is illustrated in Fig. 4.

Algorithm 1 GP surrogate pre-training via BO

INPUT: Decision variable space Θ, objective function $f(\theta)$, max evaluations $m_{\max}$
Select initial configuration $\theta_0 \in \Theta$
Evaluate initial objective $y_0 = f(\theta_0)$ via system simulation
$\theta^* \leftarrow \theta_0$, $y^* \leftarrow y_0$, $S_0 \leftarrow \{\theta_0, y_0\}$
for $n = 1$ to $m_{\max}$ **do**
  Select new configuration $\theta_m \in \Theta$ by optimising acquisition function α_m
  $\theta_m \leftarrow \arg\max_{\theta\in\Theta} \alpha_n(\theta; S_t)$
  Evaluate $f(\theta_m)$ to obtain y_m
  $S_m \leftarrow S_{m-1} \cup \{\theta_m, y_m\}$
  Update surrogate model and $\mathcal{D}_m$
  if $y_m < y^*$ **then**
    $\theta^* \leftarrow \theta_m$, $y^* \leftarrow y_m$
  end if
end for
OUTPUT: θ^* and y^*

Algorithm 2 Online BO-RG for optimizing unknown function f

INPUT: Target function $f(\theta)$, training dataset D_N, number of iterations T
Pre-train a Gaussian Process:
$\mathcal{GP}(\hat{\mu}, \hat{k} \circ \hat{\sigma}^2) \leftarrow$ PRE-TRAIN(D_N)
Initialize: $D_f \leftarrow \emptyset$
for $t = 1$ to T **do**
  Select next query point θ_t by optimizing acquisition function:
    $\theta_t \leftarrow \arg\max_{\theta\in\Theta} \alpha\left(\theta; \mathcal{GP}(\hat{\mu}, \hat{k} \circ \hat{\sigma}^2 \mid D_f)\right)$
  Evaluate function output: $y_t \leftarrow f(\theta_t)$
  $D_f \leftarrow D_f \cup \{(\theta_t, y_t)\}$
end for
OUTPUT: D_f,θ_t

D. Online Reference Governed MPC

The online implementation of the RG-MPC system leverages a pre-trained [31] GP model to perform real-time augmentation of the reference trajectory based on contextual information. This scheme enables efficient and adaptive reference adjustment without incurring the computational overhead of full Bayesian Optimisation during runtime.

To facilitate this, an offline training phase is conducted using the Bayesian Optimisation framework described in the previous section (Algorithm 1), wherein the closed-loop system is simulated across a diverse range of manoeuvre contexts and sigmoid reference parameters $\theta = [\theta_1, \theta_2, \theta_3]$. The resulting dataset $\mathcal{D}_N = \{(\theta_i, f(\theta_i))\}_{i=1}^N$ captures the system's response in terms of tracking error, control effort, and obstacle clearance, and is subsequently used to fit a GP model.

In the online phase, illustrated in Fig. 4, the pre-trained GP serves as a surrogate model for the objective function $f(\theta)$, enabling rapid evaluation of reference trajectory quality based on current driving context. The context is defined by the vehicle's longitudinal velocity v_x and the relative position of nearby obstacles (x_{obs}, y_{obs}). These variables form the input to the GP, which estimates the expected performance of different sigmoid parametrizations under the present scenario.

The RG selects the optimal parameter vector θ^* by maximising the acquisition function using the posterior GP distribution conditioned on the current context.

The online operation is formalised in Algorithm 2, where the acquisition function is evaluated using the pre-trained GP and updated with runtime evaluations of $f(\theta)$. The expected improvement inherently balances exploration and exploration of the surrogate through use of the mean and variance, but it can further be tailored to favour exploration. For online use this can be done by reducing how much improvement is needed to consider a point valuable. This hybrid offline-online approach enables fast and context-aware reference adaptation with minimal computational burden and improving tracking.

V. Efficient Data-driven RG Training

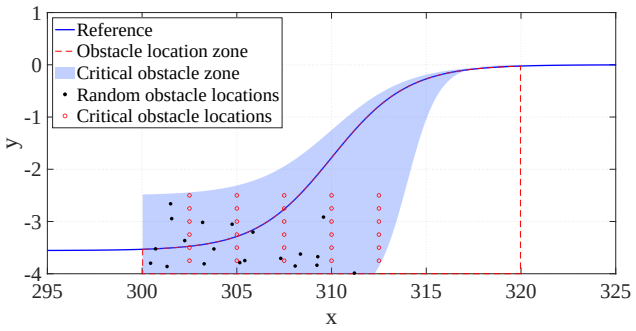


Fig. 5: Sigmoid with obstacle zones and random obstacle locations.

To evaluate the BO-RG system's performance and robustness, an efficient experiment design is needed to balance training data quality and practical feasibility. This section outlines three strategies for generating training data to support the offline Bayesian Optimization loop (Algorithm 1) and the GP model used online by the RG. Each strategy balances data volume, training efficiency, and feasibility. A hybrid approach starting with Method 3 for rapid development and moving to Method 2 for robustness refinement may offer the most practical and effective GPR model. Figure 5 shows an example of the sigmoid reference trajectory and obstacle zones used in the experiments.

The goal is to populate the contextual parameter space—vehicle velocity v_x and obstacle position (x_{obs}, y_{obs})—so the GP model can generalise across relevant driving scenarios. Each training sample is a simulation of a specific manoeuvrer with defined conditions, from which the performance cost $\Upsilon(\theta)$ is evaluated to refine the GP.

The first strategy involves uniform random sampling across a broad domain of vehicle velocity and obstacle locations. While it promotes generalization, it requires many samples, making it computationally expensive and impractical for physical data collection. Additionally, many scenarios may be low-risk and add little value.

The second strategy focuses on high-risk conditions, such as obstacles near the ego-vehicle lane and higher speeds near handling limits. The sampling is finer in y_{obs}, which has a larger impact on evasive actions, and coarser in x_{obs}. This reduces sample size while capturing the most relevant dynamics but limits the model's ability to extrapolate to less critical scenarios.

The third strategy refines the second by introducing random obstacle positions in the critical zone and limiting velocity sampling to two speeds: one for comfort driving and one near the handling limit. This reduces training effort while enabling interpolation between key velocities, but assumes the optimal reference is weakly dependent on intermediate velocities, which may not hold in rare edge cases. The training envelope covered scenarios with velocities of 80 km/h and 55 km/h, obstacle locations in the longitudinal range between 400 m and 420 m, and lateral positions between -4 m and -2 m.

VI. Experiment & Results

This section presents simulation results for the proposed BO-RG and compares it against a nominal MPC with obstacle avoidance [12]. The performance is assessed in both high-fidelity and uncertain environments, focusing on trajectory safety, control effort, and robustness to model and perception inaccuracies.

A. Simulation using high-fidelity vehicle model

Validation was conducted using IPG CarMaker with a high-fidelity SUV model featuring detailed suspension, steering and power-train actuation systems. The baseline controller is a bicycle model MPC with Dugoff tyre forces [12], [22].

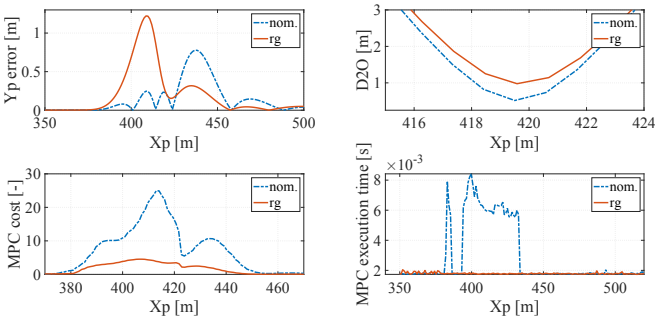


Fig. 7: Tracking error, minimum obstacle distance, control cost, and solver time for baseline vs. BO-RG.

The BO-RG was trained using 20 samples across critical obstacle positions (380m $< x_p <$ 425m, -4.5m $< y_p < -2$m) at 50 and 90 km/h. Bayesian Optimization (BO) was used to train a Gaussian Process (GP) model for online reference adjustment.

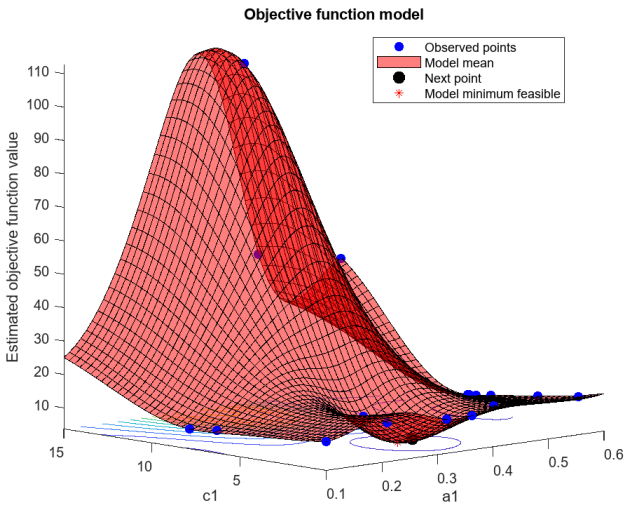


Fig. 8: BO objective landscape over sigmoid parameters a_1 (slope) and c_1 (longitudinal midpoint).

As seen in Fig. 6, the baseline MPC exhibits overshoot and violated clearance to road edges during obstacle avoidance. BO-RG increases safety margins by initially deviating more to shift the trajectory and ensuring better post-obstacle

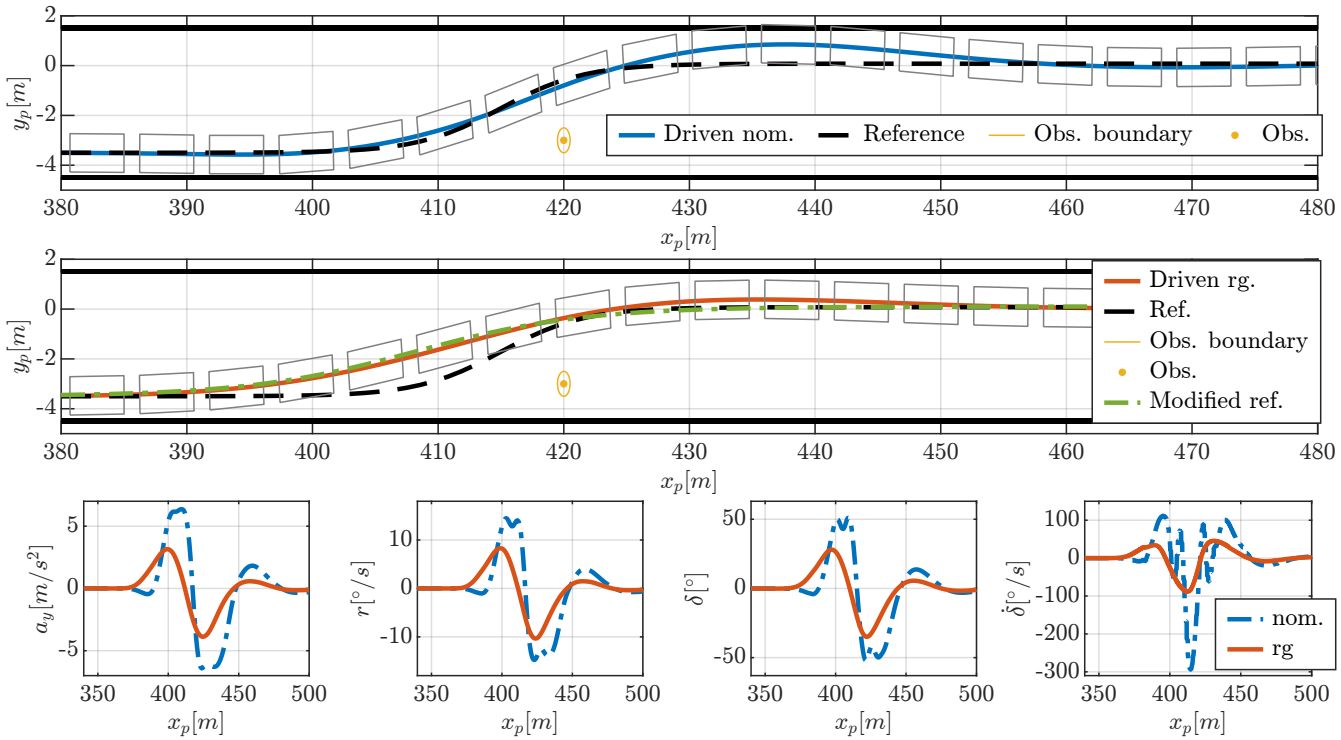


Fig. 6: Top: baseline MPC with obstacle avoidance. Middle: BO-RG with online reference adaptation. Bottom: from left to right—lateral acceleration, yaw rate, steering angle, and steering rate.

	x_r [m]	M_p [%]	x_s [m]	y_{RMSE} [m]	$y^{\text{pre}}_{\text{RMSE}}$ [m]	$y^{\text{post}}_{\text{RMSE}}$ [m]	D2O_{min} [m]
MPC	17.45	21.61	102.62	0.228	0.097	0.277	0.516
BO-RG	26.47	8.61	95.23	0.3226	0.519	0.119	0.976

TABLE I: Performance indicators at $80\,\text{km/h}$: rise/settling distance, overshoot, RMSE (overall, pre-/post-obstacle), and minimum distance to obstacle.

convergence. This yields reduced lateral acceleration and yaw rate with lower peak rates as well as smoother transients. Table I shows a +89% increase in minimum obstacle distance and a 13% reduction in overshoot. A trade-off is observed in pre-obstacle RMSE before $420m$ longitudinally, which increases due to earlier and more conservative deviation from the original reference. After $420\,\text{m}$, obstacle avoidance is completed. The overshoot is reduced, and the vehicle stays farther from the road edges. This is shown in Table I by the shorter settling distance, lower RMSE after avoidance, and smaller overshoot percentage. Furthermore, the steering rate and corresponding steering angle show that the magnitude and jitter have been reduced. This preserves actuator health and potentially improves occupant comfort.

In Fig. 7, the BO-RG reduces both controller cost and solver time during the lane change. This indicates that by reducing manoeuvrer aggressiveness, the vehicle avoids limit-handling conditions and remains in a more linear operating regime, supporting the use of simplified predictive models in conjunction with data-driven reference adaptation.

Fig. 8 illustrates the BO cost surface for two key sigmoid parameters. Cost peaks at aggressive, nominal manoeuvrers due to low obstacle clearance. The optimal solution lowers the slope and shifts initiation earlier to improve safety and tracking. Most samples concentrate near the optimum, showing successful convergence and efficient exploration around the optimum. However, the broad, flat minimum region implies a less ideal objective function surface, suggesting potential benefits from refined objective shaping or acquisition strategies.

B. Robustness & Sensitivity Analysis

To assess robustness, a second experiment introduced perception noise and model mismatch using a simplified model. Ego-localization errors followed a Gaussian distribution centred at $0.1\,\text{m}$, tailing to $0.3\,\text{m}$; obstacle localization errors were centred at $0.05\,\text{m}$ with $0.1\,\text{m}$ tail. Tyre dynamics were perturbed via $\pm 20\%$ variation in Magic Formula parameters B_y and D_y [23], altering stiffness and peak grip.

Fig. 9 shows that BO-RG consistently maintains greater obstacle clearance across all uncertainty sources. Fewer near-miss and collision events are recorded, particularly under perception noise. At higher speeds, both controllers experience degraded margins, but BO-RG sustains safer distances. A clear trend is observed where increasing longitudinal velocity leads to decreasing minimum obstacle distances and higher criticality rates. Nonetheless, the BO-RG outperforms the baseline across all tested velocities. This result can be attributed to the

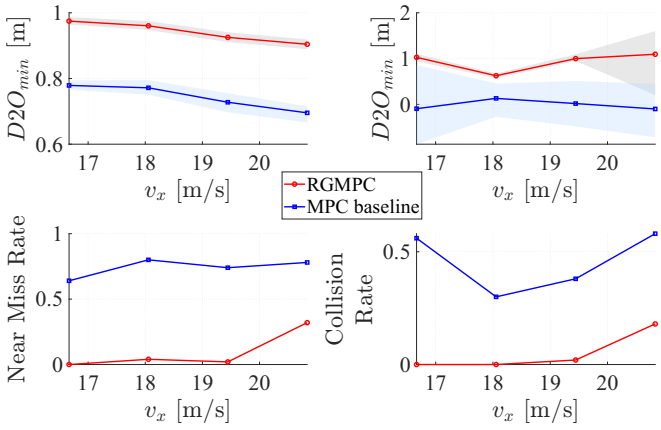


Fig. 9: Top: average obstacle clearance under tyre mismatch. Middle: clearance under perception errors. Bottom: near-miss and collision rates.

fact that the BO-RG system reduces manoeuvre aggressiveness and thus consistently increases vehicle to obstacle distance across the variation of velocities compared to the baseline.

These results demonstrate that BO-RG improves robustness by indirectly compensating for un-modelled dynamics and state uncertainty through data-driven reference shaping. Nonetheless, its reduced aggressiveness may underutilise vehicle capabilities in time-critical scenarios. Additionally, the current one-shot reference generation limits adaptability during execution. Future work should explore online re-optimisation, spline-based references, sparse GPs for scalability, and constrained/multi-objective acquisition functions to better balance safety, comfort, and efficiency.

VII. Conclusion

This work presents a novel control architecture that enhances obstacle avoidance and path following for automated vehicles by integrating a data-driven-model-free, Reference Governor within a nonlinear MPC framework. By optimising sigmoid trajectory parameters offline using Bayesian optimisation and adapting references in real-time using Gaussian Processes surrogate model, the proposed approach improves tracking performance and increases obstacle clearance. High-fidelity simulations demonstrated that the RG-MPC reduces overshoot, improves safety margins, and maintains lower control costs compared to baseline MPC, while remaining computationally efficient. The method's model-free nature enables robustness to perception errors and model mismatch, addressing a key limitation of conventional planning-control pipelines. Although limited by its one-shot trajectory design and reduced aggressiveness, the architecture effectively bridges the gap between high-fidelity planning and tractable control, offering a promising direction for robust, real-time motion replanning.

References

[1] Y. K. Al-Nadawi, H. Al-Qassab, D. Kent, S. Pang, V. Srivastava, and H. Radha, "Design of robust path-following control system for self-driving vehicles using extended high-gain observer," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 1435–1440.

[2] F. Altché, P. Polack, and A. de La Fortelle, "High-speed trajectory planning for autonomous vehicles using a simple dynamic model," in *2017 IEEE 20th international conference on intelligent transportation systems (ITSC)*. IEEE, 2017, pp. 1–7.

[3] A. Bertipaglia, D. Tavernini, U. Montanaro, M. Alirezaei, R. Happee, A. Sorniotti, and B. Shyrokau, "On the benefits of torque vectoring for automated collision avoidance at the limits of handling," *IEEE Transactions on Vehicular Technology*, pp. 1–16, 2025.

[4] D. Lenssen, A. Bertipaglia, F. Santafe, and B. Shyrokau, "Combined path following and vehicle stability control using model predictive control," SAE Technical Paper, Tech. Rep., 2023.

[5] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a frenet frame," in *Proc. IEEE Intell. Vehicles Symp.*, 2010, pp. 987–993.

[6] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1135–1145, Apr. 2016.

[7] J. K. Subosits and J. C. Gerdes, "From the racetrack to the road: Real-time trajectory replanning for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 2, pp. 309–320, 2019.

[8] A. Rupenyan, M. Khosravi, and J. Lygeros, "Performance-based trajectory optimization for path following control using bayesian optimization," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 2116–2121.

[9] J. Ziegler, P. Bender, T. Dang, and C. Stiller, "Trajectory planning for bertha – a local, continuous method," in *Proc. IEEE Intell. Vehicles Symp.*, 2014, pp. 450–457.

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

[11] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Trans. Intell. Vehicles*, vol. 1, no. 1, pp. 33–55, Mar. 2016.

[12] A. Bertipaglia, M. Alirezaei, R. Happee, and B. Shyrokau, "A learning-based model predictive contouring control for vehicle evasive manoeuvres," in *Advanced Vehicle Control Symposium*. Springer, 2024, pp. 632–638.

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

[14] M. Khosravi, A. Eichler, N. Schmid, R. S. Smith, and P. Heer, "Controller tuning by bayesian optimization an application to a heat pump," in *2019 18th European Control Conference*. IEEE, 2019, pp. 1467–1472.

[15] M. Khosravi, C. Konig, M. Maier, R. S. Smith, J. Lygeros, and A. Rupenyan, "Safety-aware cascade controller tuning using constrained bayesian optimization," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 2, pp. 2128–2138, 2023.

[16] I. Kolmanovsky and E. Gilbert, "Theory and computation of disturbance invariant sets for discrete-time linear systems," *Math. Problems Eng.*, vol. 4, no. 4, pp. 317–367, 1998.

[17] I. Kolmanovsky, A. Bemporad, and E. Garone, "Vehicle systems with reference and command governors: Opportunities for semi-autonomous driving," in *IEEE Conf. Decis. Control*, Osaka, Japan, 2015, pp. 4012–4017.

[18] E. Garone, S. D. Cairano, and I. Kolmanovsky, "Reference and command governors for systems with constraints: A survey on theory and applications," *Automatica*, vol. 75, pp. 306–328, Jan. 2017.

[19] A. Carvalho, S. Lefévre, H. Gao, and F. Borrelli, "Stochastic predictive control of autonomous vehicles in uncertain environments," in *Proc. 12th Int. Symp. Adv. Vehicle Control*, Tokyo, Japan, 2014, pp. 712–717.

[20] M. Khosravi, V. Behrunani, R. S. Smith, A. Rupenyan, and J. Lygeros, "Cascade control: Data-driven tuning approach based on bayesian optimization," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 382–387, 2020.

[21] M. Khosravi, V. N. Behrunani, P. Myszkorowski, R. S. Smith, A. Rupenyan, and J. Lygeros, "Performance-driven cascade controller tuning with bayesian optimization," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 1, pp. 1032–1042, 2021.

[22] N. Chowdhri, L. Ferranti, F. S. Iribarren, and B. Shyrokau, "Integrated nonlinear model predictive control for automated driving," *Control Engineering Practice*, vol. 106, p. 104654, 2021.

[23] H. B. Pacejka and E. Bakker, "The magic formula tyre model," *Vehicle system dynamics*, vol. 21, no. S1, pp. 1–18, 1992.

[24] A. Bhoraskar and P. Sakthivel, "A review and a comparison of dugoff and modified dugoff formula with magic formula," in *2017 International Conference on Nascent Technologies in Engineering*, 2017, pp. 1–4.

[25] I. Bae, J. H. Kim, and S. Kim, "Steering rate controller based on curvature of trajectory for autonomous driving vehicles," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 1381–1386.
[26] R. Rajamani, "Vehicle dynamics and control," *Mechanical Engineering Series*, 2012.
[27] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.
[28] P. I. Frazier, "A tutorial on bayesian optimization," *arXiv preprint arXiv:1807.02811*, 2018.
[29] A. Lambora, K. Gupta, and K. Chopra, "Genetic algorithm-a literature review," in *2019 international conference on machine learning, big data, cloud and parallel computing*. IEEE, 2019, pp. 380–384.
[30] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," *Advances in neural information processing systems*, vol. 24, 2011.
[31] Z. Wang, G. E. Dahl, K. Swersky, C. Lee, Z. Nado, J. Gilmer, J. Snoek, and Z. Ghahramani, "Pre-trained gaussian processes for bayesian optimization," *Journal of Machine Learning Research*, vol. 25, no. 212, pp. 1–83, 2024.

Bibliography

[1] Murata Manufacturing Co., Ltd., *K6EU-3 Application Guide V2X*, 2023, accessed: 2024-09-10. [Online]. Available: https://go.murata.com/rs/382-MEZ-125/images/K6EU-3%20Application%20Guide_V2X.pdf

[2] U. o. M. Center for Sustainable Systems, "Autonomous vehicles factsheet," 2024, accessed: 2025-03-04. [Online]. Available: https://css.umich.edu/publications/factsheets/mobility/autonomous-vehicles-factsheet

[3] Brookings, "The evolving safety and policy challenges of self-driving cars," 2024, accessed: 2025-03-04. [Online]. Available: https://www.brookings.edu/articles/the-evolving-safety-and-policy-challenges-of-self-driving-cars/

[4] NSC, "Advanced driver assistance systems-data details," 2024, accessed: 2025-03-04. [Online]. Available: https://injuryfacts.nsc.org/motor-vehicle/occupant-protection/advanced-driver-assistance-systems/data-details/

[5] AVI, "Waymo reduces crash rates compared to human drivers over 7+ million miles," 2024, accessed: 2025-03-04. [Online]. Available: https://theavindustry.org/resources/blog/waymo-reduces-crash-rates-compared-to-human-drivers

[6] F. Tao, Z. Ding, Z. Fu, M. Li, and B. Ji, "Efficient path planning for autonomous vehicles based on rrt* with variable probability strategy and artificial potential field approach," *Scientific Reports*, vol. 14, no. 1, p. 24698, 2024.

[7] W. Huang, N. Wu, Z. Song, X. Wu, Q. Zhang, and S. Yao, "Hybrid strategy of motion planning with kinematic optimization for autonomous driving," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, 2015, pp. 1666–1671.

[8] M. Diachuk and S. M. Easa, "Motion planning for autonomous vehicles based on sequential optimization," *Vehicles*, vol. 4, no. 2, pp. 344–374, 2022.

[9] D. Gonzalez Bautista, J. Pérez, V. Milanes, and F. Nashashibi, "A Review of Motion Planning Techniques for Automated Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, pp. 1–11, Nov. 2015.

[10] Z. Wang, K. Sun, S. Ma, L. Sun, W. Gao, and Z. Dong, "Improved linear quadratic regulator lateral path tracking approach based on a real-time updated algorithm with fuzzy control and cosine similarity for autonomous vehicles," *Electronics*, vol. 11, no. 22, p. 3703, 2022.

[11] S. Abdallaoui, A. Kribèche, and E.-H. Aglzim, "Comparative study of mpc and pid controllers in autonomous vehicle application," in *International Symposium on Automation, Mechanical and Design Engineering*. Springer, 2021, pp. 133–144.

[12] T. Yang, Z. Bai, Z. Li, N. Feng, and L. Chen, "Intelligent vehicle lateral control method based on feedforward+ predictive lqr algorithm," in *Actuators*, vol. 10, no. 9. MDPI, 2021, p. 228.

[13] A. Rupenyan, M. Khosravi, and J. Lygeros, "Performance-based trajectory optimization for path following control using bayesian optimization," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 2116–2121.

[14] M. Khosravi, A. Eichler, N. Schmid, R. S. Smith, and P. Heer, "Controller tuning by bayesian optimization an application to a heat pump," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 1467–1472.

[15] M. Khosravi, V. Behrunani, R. S. Smith, A. Rupenyan, and J. Lygeros, "Cascade control: Data-driven tuning approach based on bayesian optimization," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 382–387, 2020.

[16] M. Khosravi, V. N. Behrunani, P. Myszkorowski, R. S. Smith, A. Rupenyan, and J. Lygeros, "Performance-driven cascade controller tuning with bayesian optimization," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 1, pp. 1032–1042, 2021.

[17] M. Khosravi, C. Konig, M. Maier, R. S. Smith, J. Lygeros, and A. Rupenyan, "Safety-aware cascade controller tuning using constrained bayesian optimization," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 2, pp. 2128–2138, 2023.

[18] H. B. Pacejka and E. Bakker, "The magic formula tyre model," *Vehicle system dynamics*, vol. 21, no. S1, pp. 1–18, 1992.

[19] A. Bhoraskar and P. Sakthivel, "A review and a comparison of dugoff and modified dugoff formula with magic formula," in *2017 International Conference on Nascent Technologies in Engineering (ICNTE)*, 2017, pp. 1–4.

[20] F. Pretagostini, B. Shyrokau, and G. Berardo, "Anti-lock braking control design using a nonlinear model predictive approach and wheel information," in *2019 IEEE international conference on mechatronics (ICM)*, vol. 1. IEEE, 2019, pp. 525–530.

[21] M. Diehl, "Real-time optimization for large scale nonlinear processes," Ph.D. dissertation, 2001.

[22] P. T. Boggs and J. W. Tolle, "Sequential quadratic programming," *Acta numerica*, vol. 4, pp. 1–51, 1995.

[23] R. Bartlett, A. Wachter, and L. T. Biegler, "Active set vs. interior point strategies for model predictive control," in *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No. 00CH36334)*, vol. 6. IEEE, 2000, pp. 4229–4233.

[24] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpoases: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, pp. 327–363, 2014.

[25] C. Audet and W. Hare, "Derivative-free and blackbox optimization," *Springer Optimization and Its Applications*, vol. 54, 2017.

[26] S. Shan and G. G. Wang, "Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions," *Structural and Multidisciplinary Optimization*, vol. 41, no. 2, pp. 219–241, 2010.

[27] A. Ekamperi, "Acquisition functions in bayesian optimization," June 2021, accessed: 2025-04-07. [Online]. Available: https://ekamperi.github.io/machine%20learning/2021/06/11/acquisition-functions.html

[28] R. Garnett, "Cse 515t: Bayesian methods in machine learning – spring 2015," 2015, accessed: 2025-04-07. [Online]. Available: https://www.cse.wustl.edu/~garnett/cse515t/spring_2015/

[29] F. Liu, L. Zheng, M. Li, and J. Tang, "Analysis and prediction of the interval duration between the first and second accidents considering the spatiotemporal threshold," *Journal of Advanced Transportation*, vol. 2022, pp. 1–14, 02 2022.

[30] N. Hansen, "The cma evolution strategy: A tutorial," *arXiv preprint arXiv:1604.00772*, 2016.

[31] L. M. Rios and N. V. Sahinidis, "Derivative-free optimization: A review of algorithms and comparison of software implementations," *Journal of Global Optimization*, vol. 56, no. 3, pp. 1247–1293, 2013.

[32] P. I. Frazier, "A tutorial on bayesian optimization," *arXiv preprint arXiv:1807.02811*, 2018.

[33] L. Bliek, A. Guijt, R. Karlsson, S. Verwer, and M. de Weerdt, "Benchmarking surrogate-based optimisation algorithms on expensive black-box functions," *Applied Soft Computing*, vol. 147, p. 110744, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494623007627

[34] P. S. Palar, R. P. Liem, L. R. Zuhal, and K. Shimoyama, "On the use of surrogate models in engineering design optimization and exploration: The key issues," in *Proceedings of the genetic and evolutionary computation conference companion*, 2019, pp. 1592–1602.

[35] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2016.

[36] P. I. Frazier, "A tutorial on bayesian optimization," *arXiv preprint arXiv:1807.02811*, 2018.

[37] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[38] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," in *Proceedings of the 27th International Conference on Machine Learning (ICML-2010)*, 2010, pp. 1015–1022.

[39] "Acquisition functions in bayesian optimisation," https://ekamperi.github.io/machine%20learning/2021/06/11/acquisition-functions.html, accessed: 2025-03-25.

[40] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[41] J. H. Holland, *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.

[42] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948.

[43] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006, vol. 2.

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

[45] C. E. Rasmussen and H. Nickisch, "The gpml toolbox version 4.2," 2018.

[46] D. Duvenaud, "Automatic model construction with gaussian processes," Ph.D. dissertation, University of Cambridge, 2014.

[47] J. Wang, C. G. Petra, and J. L. Peterson, "Constrained bayesian optimization with merit functions," *arXiv preprint arXiv:2403.13140*, 2024.

[48] J. M. Hern, M. A. Gelbart, R. P. Adams, M. W. Hoffman, Z. Ghahramani *et al.*, "A general framework for constrained bayesian optimization using information-based search," *Journal of Machine Learning Research*, vol. 17, no. 160, pp. 1–53, 2016.

[49] B. Letham, B. Karrer, G. Ottoni, and E. Bakshy, "Constrained bayesian optimization with noisy experiments," 2019.

[50] H. Zhou, X. Ma, and M. B. Blaschko, "A corrected expected improvement acquisition function under noisy observations," in *Asian Conference on Machine Learning*. PMLR, 2024, pp. 1747–1762.

[51] N. Chowdhri, L. Ferranti, F. S. Iribarren, and B. Shyrokau, "Integrated nonlinear model predictive control for automated driving," *Control Engineering Practice*, vol. 106, p. 104654, 2021.

[52] D. Lenssen, A. Bertipaglia, F. Santafe, and B. Shyrokau, "Combined path following and vehicle stability control using model predictive control," 04 2023.

[53] B. Zhou, C. Hu, J. Zeng, Z. Li, J. Betz, L. Xie, and H. Su, "Adaptive learning-based model predictive control strategy for drift vehicles," *Robotics and Autonomous Systems*, p. 104941, 2025.

[54] M. Sekadakis, C. Katrakazas, E. Santuccio, P. Mörtl, and G. Yannis, "Key performance indicators for safe fluid interactions within automated vehicles," in *10th International Congress on Transportation Research*, 2021.

[55] N. Sawarkar, "Ai driven predictive functional safety leveraging key performance indicators in autonomous vehicles," in *2024 International Conference on Artificial Intelligence and Quantum Computation-Based Sensor Application (ICAIQSA)*. IEEE, 2024, pp. 1–9.

[56] T. Boscher, A. Günther, and K. Scheck, "Development of an objective evaluation method for manual and automated parking maneuvers," in *12th International Munich Chassis Symposium 2021: chassis. tech plus*. Springer, 2022, pp. 124–143.

[57] A. P. Swanda and D. E. Seborg, "Controller performance assessment based on setpoint response data," in *Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251)*, vol. 6. IEEE, 1999, pp. 3863–3867.

[58] S. Dominic, Y. A. Shardt, S. X. Ding, and H. Luo, "An adaptive, advanced control strategy for kpi-based optimization of industrial processes," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 5, pp. 3252–3260, 2015.

[59] B. Wohlers, S. Dziwok, F. Pasic, A. Lipsmeier, and M. Becker, "Monitoring and control of production processes based on key performance indicators for mechatronic systems," *International journal of production economics*, vol. 220, p. 107452, 2020.

[60] B. Goldfain, P. Drews, C. You, M. Barulic, O. Velev, P. Tsiotras, and J. Rehg, "Autorally an open platform for aggressive autonomous driving," 06 2018.

[61] A. Bertipaglia, D. Tavernini, U. Montanaro, M. Alirezaei, R. Happee, A. Sorniotti, and B. Shyrokau, "On the benefits of torque vectoring for automated collision avoidance at the limits of handling," *IEEE Transactions on Vehicular Technology*, 2025.

Glossary

List of Acronyms

ADAS	Advanced Driver-Assistance Systems
AI	Artificial Intelligence
ASM	Active Set Point Method
AV	Automated Vehicles
BO	Bayesian Optimisation
BORG	Bayesian Optimisation-based Reference Governor
CMAES	Covariance Matrix Adaptation Evolution Strategy
CDF	Cumulative Distribution Function
CLR	Collision Rate
D2O	Distance to Obstacle
DTC	Distance to Collision
EA	Evolutionary Algorithms
EI	Expected Improvement
GA	Genetic Algorithms
GP	Gaussian Process
KKT	Karush-Kuhn-Tucker
KPI	Key Performance Indicator
LHS	Latin Hypercube Sampling
LQR	Linear Quadratic Regulator
LTI	Linear Time Invariant
MPC	Model Predictive Control
NMR	Near Miss Rate
OCP	Optimal Control Problem

PDF	Probability Density Function
PID	Proportional-Integral-Derivative
PI	Probability of Improvement
PSO	Particle Swarm Optimisation
QP	Quadratic Programming
RG	Reference Governor
RG-MPC	Reference Governed Model Predictive Control
RMSE	Root Mean Square Error
RTI	Real-time Iteration
SE	Squared Exponential
SQP	Sequential Quadratic Programming
UCB	Upper Confidence Bound