



Degree Project in Technology

Second cycle, 30 credits

Safe Exploration for Non-linear Systems

A Data-Driven Framework for Safe Data Collection in Nonlinear
Systems

STEFANO TONINI

Safe Exploration for Non-linear Systems

A Data-Driven Framework for Safe Data Collection in Nonlinear Systems

STEFANO TONINI

Master's Programme, ICT Innovation, 120 credits

Date: November 2, 2025

Supervisors: Nicola Bastianello, Soroush Rastegarpour, Hamid Feyzmahdavian

Examiner: Karl Henrik Johansson

School of Electrical Engineering and Computer Science

Host company: ABB Corporate Research

Swedish title: Säker utforskning av icke-linjära system

Swedish subtitle: Ett datadrivet ramverk för säker datainsamling i icke-linjära system

Abstract

We present a framework that enables autonomous systems to collect rich, informative data from partially unknown nonlinear plants while strictly respecting stability and hard state–input constraints. Starting from a stabilisable linear approximation, the unmodelled residual dynamics are learned on-line with Gaussian-process (GP) regression, which delivers both a mean estimate and high-confidence variance envelopes. These uncertainty bounds are embedded in a probabilistic control-invariant set (PCIS) that contracts the state space to a region where every state is guaranteed to remain safe with probability $1 - \delta$. At each sampling instant a small quadratic programme selects the control input that (i) keeps the state inside the PCIS, (ii) tracks the nominal stabilising input as closely as possible, and (iii) actively excites poorly modelled directions so that the GP posterior variance shrinks over time. As learning progresses the PCIS expands automatically, allowing progressively more aggressive exploration without sacrificing guarantees.

The framework is validated on two benchmarks of increasing complexity: (a) a two-state, unstable polynomial system and (b) a laboratory three-tank process with multi-input actuation and water-level constraints.

Keywords

Safe Exploration, Nonlinear Systems, Gaussian Processes, Probabilistic Control Invariant Sets, Control Lyapunov Functions

Sammanfattning

Vi presenterar ett ramverk som gör det möjligt för autonoma system att samla rik och informativ data från delvis okända, olinjära processer samtidigt som stabilitet och hårda tillstånds- och insatsbegränsningar strikt upprätthålls. Utgångspunkten är en stabiliserbar linjär approximation; den omodellerade residualdynamiken lärs on-line med Gaussisk process-regression (GP), som ger både en medelvärdeskattning och högkonfidensintervall för variansen. Dessa osäkerhetsgränser integreras i en *probabilistisk kontrollinvariant mängd* (PCIS) som begränsar tillståndsrymden till ett område där varje tillstånd garanteras vara säkert med sannolikhet $1 - \delta$.

Vid varje provtagningsögonblick löser ett litet kvadratisk program följande uppgift: (i) hålla tillståndet inom PCIS, (ii) följa det nominella stabiliserande styrsignalet så nära som möjligt, och (iii) aktivt excitera dåligt modellerade riktningar så att GP-posterns varians minskar över tid. Allteftersom inläringen fortskrider expanderar PCIS automatiskt, vilket möjliggör allt mer aggressiv utforskning utan att garantierna offras.

Ramverket valideras på två testfall av stigande komplexitet: (a) ett tvåtillståndigt, instabilt polynomsystem och (b) en laboratorieuppställning med tre sammankopplade tankar, flerports aktivering och nivåbegränsningar för vattennivån.

Nyckelord

Säker Utforskning, Icke-linjära System, Gaussiska Processer, Sannolikhetsbaserade Kontrollinvarianta Mängder, Lyapunov-funktioner för Styrning

Acknowledgments

This thesis marks the end of a long journey which took place in 3 different countries, Italy, Finland, and Sweden. I had the opportunity to meet incredible wise people and moreover friends who accompanied me throughout this experience. The knowledge I gained and furthermore the moments I had lived are unvaluable. I am proud to carry these memories with me throughout my life. I want to express my sincere thanks to my company supervisors Soroush and Hamid without their guidance all this work would not be possible. A big thanks goes to my university supervisors Nicola Bastianello and Matteo Saveriano. I am grateful also to all my close friends Jacopo, Vittorio, Lorenzo T., Gianluca, Leonardo, Lorenzo S. Last but not least, I want to thank my family that was always there to support and listen to me. I hope everyone who will read the following pages will find inspiration for their future works.

Stockholm, November 2025
Stefano Tonini

Contents

1	Introduction	1
1.1	Why safe exploration?	2
1.2	State of the Art in Safe Exploration	2
1.2.1	Gaussian Process Safety Filters and Invariant Sets	3
1.2.2	Gaussian Process Model Predictive Control (GP-MPC)	3
1.2.3	Lyapunov and Barrier Certificates	4
1.2.4	Optimal-Control-Based Exploration	4
1.2.5	Connections to Kernel Methods and System Identification	5
1.2.6	Non-GP Approaches and Safe Reinforcement Learning	5
1.2.7	Practical Applications	5
1.3	Gaps and challenges	6
1.4	Ethics and Sustainability	6
1.5	Thesis objectives and contributions	7
1.6	Structure of the thesis	7
1.7	Research Question	8
2	Background	9
2.1	Safe Exploration in Nonlinear Systems	9
2.2	Model-Free Learning	11
2.3	Model-Based Learning	12
2.4	Gaussian Process	13
2.5	GP prediction and model uncertainty	15
2.5.1	Sparse Gaussian Processes for Scalability	16
2.5.2	High-confidence GP–UCB bound and its role in safety	17
2.6	Control-Invariant Sets	18
2.7	Probabilistic Control Invariant Set	19
2.8	Linear Quadratic Regulator	20
2.9	Lyapunov Controller	21

3	Methods	23
3.1	Gaussian Process Regression	23
3.1.1	GP Formulation	23
3.1.2	Kernel Selection	24
3.1.3	Hyperparameter Optimization	24
3.1.4	Uncertainty Quantification	24
3.2	Probabilistic Control Invariant Sets	25
3.2.1	Direct GP-Based Prediction	25
3.2.2	Optimization-Based PCIS	25
3.3	Control Lyapunov Functions	26
3.3.1	Linear Quadratic Regulator (LQR) Baseline	26
3.3.2	GP-Augmented CLF Controller	26
3.4	Problem Statement and System Model	27
3.4.1	True Nonlinear Process Form	27
3.4.2	Objective	28
3.5	Probabilistic Safety Framework	29
3.5.1	High-Probability Confidence Bound	29
3.5.2	One-Step Safe Set	29
3.5.3	LQR Baseline and Lyapunov Function	30
3.5.4	Largest Invariant Ellipsoid	30
3.5.5	Probabilistic Control-Invariant Set Inside the Ellipsoid	31
3.6	Control-Lyapunov Function Framework	32
3.6.1	GP-Augmented CLF-QP	32
3.6.2	Derivation of the Robustified Lyapunov Derivative	33
3.7	Safe-UCB Exploration Algorithm	34
3.7.1	Parameter Guidelines	36
3.8	Case Studies and Modelling Details	36
3.8.1	Illustrative 2D Example	36
3.8.2	Three-Tank Nonlinear Process	37
3.9	Simulation Environment	42
4	Results	47
4.1	Polynomial Benchmark	48
4.1.1	Gaussian Process Kernel Comparison	48
4.1.2	Full GP versus Sparse GP	51
4.1.3	Probabilistic Control Invariant Set	53
4.1.4	Unsafe exploration	54
4.1.5	GP-PCIS safe exploration	55
4.2	Three-Tank Process	58

4.2.1	Gaussian Process	59
4.2.2	Full GP versus Sparse GP	63
4.2.3	Probabilistic Control Invariant Set	65
4.2.4	Unsafe Exploration	66
4.2.5	GP-PCIS safe exploration	67
5	Discussion	71
5.1	Method Limitations	71
5.2	Achievements	71
5.3	Future Directions	72
5.3.1	Safe Optimal Experiment Design (OED)	72
6	Conclusions	75
	References	77
A	System Modeling	85
A.1	Physical Model of the Multitank System	85
A.2	Linear Modeling for Control	87
B	Simulation Environment (Extended)	90
C	Gaussian Process Kernels: A Practical Guide	96
C.0.1	Common kernels	96
C.0.2	Compositions and nonstationarity	97
C.0.3	Control-oriented guidance	97
C.0.4	Minimal formulas to cite	98

List of Figures

1.1	Proposed safe-exploration architecture. The nominal <i>linear model</i> supplies a baseline prediction. A Gaussian Process (GP) learns the residual Δx from measured state-input pairs and outputs mean μ and variance σ . The Probabilistic Control-Invariant Set (PCIS) module converts the GP uncertainty into a shrinking/expanding safe set \mathcal{S}_δ and passes the target state x^* to the <i>controller</i> (e.g. CLF-QP). The controller returns a certified input u^* that drives the <i>process</i> while keeping the closed loop inside \mathcal{S}_δ . All data are fed back online, enabling lifelong, provably safe learning.	2
2.1	Gaussian Process learning snapshots: posterior mean and uncertainty at the beginning, mid-way, and after training. . . .	16
2.2	Largest robust control-invariant ellipsoid $\alpha = \{x : x^\top P x \leq \alpha\}$ contained in box state constraints \mathbb{X} under linear feedback $u = Kx$. The ellipsoid is chosen so that $\dot{V} \leq 0$ on ∂_α despite bounded disturbances, yielding a deterministic RCIS inside \mathbb{X}	19
3.1	Largest control-invariant ellipsoid $E(\alpha_{\max})$ (blue) entirely contained in the polyhedral constraint set X (red) for the linearised system.	31
3.2	Unsafe Exploration due to a wrong target selected	33
3.3	2D example: PCIS tube and closed-loop trajectories under the CLF-QP controller.	37
3.4	Three-tank system: valves $\gamma_1, \gamma_2, \gamma_3$ are control inputs; pump inflow q is a measured disturbance. States are levels h_1, h_2, h_3 ; constraints \mathcal{X}, \mathcal{U} are safety-critical.	38

4.1	Kernel comparison for GP residual modeling on the 2D system. Each panel shows the GP mean (blue) and $\pm 2\sigma$ band against the true quadratic residual (red, $0.1x_2^2$) along the slice $x_1 = 0$. Metrics are computed on a held-out set.	49
4.3	Comparison between exact GP and Sparse GP (FITC) on the 2D car system. Top: training data, full-GP test predictions, and RMSE across models. Bottom: R^2 , training time, and efficiency (R^2 vs. time). Sparse approximations achieve 30–40× speed-ups but suffer strong degradation in R^2 and coverage for $M > 10$	52
4.4	PCIS—Full GP vs. Sparse GP (2D toy). The black ellipse is the Lyapunov sublevel set. Colored contours show certified PCIS for the exact GP and SGPR with $M \in \{10, 20, 30, 40\}$. Legend reports areas and fraction of the ellipse captured.	54
4.5	LQR tracking to a target <i>without</i> a safety layer. The green region shows the state-wise GP-UCB safe slice $\{x : \mu(x) + \beta\sigma(x) \leq \varepsilon\}$ inside the certified ellipsoid $S(\alpha)$ (grey, dashed boundary). Starting from the red dot, the LQR trajectory (red) exits $S(\alpha)$ while moving toward the target (yellow), demonstrating unsafe exploration in the absence of PCIS/CLF-QP filtering.	55
4.6	Safe exploration on the 2D system. The dashed blue ellipse is the PCIS boundary $x^\top Px = \alpha$; the green region contains states satisfying the GP safety predicate $ \mu(x) + \beta\sigma(x) \leq \varepsilon$. The trajectory (red) stays inside the safe set while moving from the initial (red dot) to the target (yellow ring). Controller: LQR baseline with safety filter; GP trained online on residuals.	56
4.7	Gaussian Process fitted on the initial dataset	56
4.8	Gaussian Process fitted on the final dataset after Safe Exploration	57
4.9	Comparison of GP covariance trace and RMSE across controllers.	58
4.10	Time-series of the three-tank process: FMU data versus Linear model and Linear+GP with Matern-3/2 kernel. The GP correction significantly reduces the RMSE across all tanks.	59
4.11	Time-series of the three-tank process: FMU data versus Linear model and Linear+GP with Matern-5/2 kernel.	60
4.12	Time-series of the three-tank process: FMU data versus Linear model and Linear+GP with Periodic-RBF kernel.	60

4.13	Time-series of the three-tank process: FMU data versus Linear model and Linear+GP with Polynomial kernel of degree 2.	61
4.14	Time-series of the three-tank process: FMU data versus Linear model and Linear+GP with standard RBF kernel.	61
4.15	Time-series of the three-tank process: FMU data versus Linear model and Linear+GP with RBF kernel and Automatic Relevance Determination (ARD).	62
4.16	Time-series of the three-tank process: FMU data versus Linear model and Linear+GP with combined RBF + Matern-3/2 kernel.	62
4.17	Water-tank — Full GP vs. Sparse GP (aggregate over three residuals). Top-left: RMSE (lower is better). Top-right: R^2 (very negative on res1 due to tiny variance; we therefore down-weight it). Bottom-left: empirical coverage vs. 95% nominal. Bottom-right: retraining time (lower is better).	64
4.18	Probabilistic Control Invariant Set for the Initial Configuration before Safe Exploration	66
4.19	Unsafe exploration on the three-tank process. <i>Top</i> : water levels vs. safety band; a low-low violation occurs for H_3 at $t \approx 131$ s, with a minimum of 44.8%. <i>Middle</i> : valve openings around the violation time. <i>Bottom</i> : event summary (iteration, tank, time, extreme level, target).	67
4.20	Progressive safe exploration on the three-tank process. Top row: water levels $H_{1:3}$ with safety bands (green: safe zone 45–87%; red: LL/HH). Dashed lines are iteration targets; solid lines are trajectories. Bottom row: valve openings $\gamma_{1:3}$. The header of each panel reports start/target/final levels, and the bottom panels report average valve usage.	68
A.1	Cross-sectional areas $A_i(h_i)$ defined in (A.6). Tank 1 is prismatic (constant area), tank 2 is linearly varying (trapezoid), and tank 3 exhibits strong nonlinearity at low levels.	87
B.1	FMU Simulink model	91

Chapter 1

Introduction

Autonomous agents in safety-critical domains such as industrial process control, robotics, and autonomous driving must *learn* accurate dynamics models *while* respecting hard state and input constraints at every time step. Purely model-based controllers offer real-time guarantees but suffer when the nominal model is inaccurate; purely data-driven reinforcement learning explores aggressively, yet often violates constraints during its trial-and-error phase. Bridging this gap motivates the study of **safe exploration**: how can an agent gather informative data from an *unknown, nonlinear* plant without ever leaving a user-specified safe set?

This thesis focuses on safe exploration in *nonlinear, unknown systems*, adopting a model-based approach. The goal is to design a learning framework that can explore safely and efficiently using a synergy between machine learning and control theory.

This work integrates three building blocks that combined together achieved safe exploration:

- Gaussian Process Regression
- Probabilistic Control Invariant Set
- Control Lyapunov Function

By unifying probabilistic modeling, nonlinear control theory, and learning-based exploration, this thesis aims to advance the state of the art in data-driven safe control and to facilitate the deployment of autonomous systems that learn safely from their own experience.

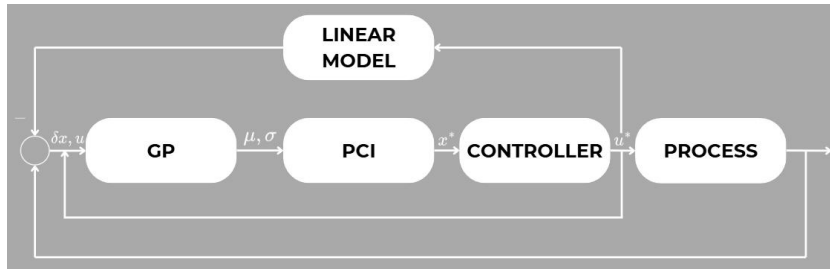


Figure 1.1: Proposed safe-exploration architecture. The nominal *linear model* supplies a baseline prediction. A Gaussian Process (GP) learns the residual Δx from measured state-input pairs and outputs mean μ and variance σ . The Probabilistic Control-Invariant Set (PCIS) module converts the GP uncertainty into a shrinking/expanding safe set \mathcal{S}_δ and passes the target state x^* to the *controller* (e.g. CLF-QP). The controller returns a certified input u^* that drives the *process* while keeping the closed loop inside \mathcal{S}_δ . All data are fed back online, enabling lifelong, provably safe learning.

1.1 Why safe exploration?

Classical model-based control assumes that a reliable dynamics model is available *a priori*. In practice, however, first-principles models are often mismatched, and purely data-driven reinforcement learning, although asymptotically optimal, tends to violate state or input constraints during its trial and error phase. This mismatch is unacceptable for industrial processes and robots sharing their workspace with humans or operating close to physical limits. Recent accidents of autonomous vehicles and drones have underscored the need for learning algorithms that **provably** respect safety constraints at every time step.

1.2 State of the Art in Safe Exploration

Safe exploration addresses the challenge of learning about an uncertain system or environment while strictly respecting safety constraints. This problem is critical for autonomous robots and control systems that must collect informative data in real-world scenarios without causing damage or failure. During the past decade, a wide range of approaches have been proposed to guarantee safety during exploration. This section surveys the main research lines, covering both *Gaussian Process (GP)-based methods* and *non-GP approaches*, with an emphasis on recent developments, theoretical guarantees,

and practical applications.

1.2.1 Gaussian Process Safety Filters and Invariant Sets

Gaussian Processes (GPs) are widely used to model unknown dynamics due to their ability to provide uncertainty estimates. In safe exploration, GP models are leveraged to construct high-probability *safe sets* or invariant tubes. Early work by Berkenkamp *et al.* demonstrated how GP dynamics models can be used to identify *control-invariant regions of attraction* and restrict learning within these regions [1]. By maintaining a backup controller that ensures safety, GP-based safety filters guarantee that trajectories remain inside a probabilistically safe tube at all times. As the GP model improves, the certified safe region can be progressively expanded without ever violating constraints. This approach has been validated in robotic systems, such as quadrotors, where the flight envelope is safely enlarged only after stability is certified [1]. GP safety filters thus provide *formal probabilistic safety guarantees*, although they can be conservative in the early stages of learning.

1.2.2 Gaussian Process Model Predictive Control (GP-MPC)

Another major direction integrates GP models into *Model Predictive Control (MPC)* frameworks. In GP-MPC, the GP provides uncertainty-aware predictions within an optimization-based controller. Constraints are tightened with respect to GP uncertainty, yielding “cautious” MPC schemes that ensure constraint satisfaction with high probability [2]. The SafeMPC algorithm, for example, combines GP confidence intervals with a terminal invariant set, guaranteeing a feasible return-to-safety trajectory at every step. This enables active exploration while maintaining theoretical safety guarantees. Subsequent work improved tractability through tailored sequential quadratic programming solvers [3], making GP-MPC applicable in real-time settings. Tutorial literature has consolidated GP-MPC practices for robotics [3], reflecting its maturity as a methodology for balancing performance and safety. Two complementary 2025 directions strengthen GP-MPC guarantees. First, Dubied *et al.* propose a robust and *adaptive* tube MPC for GP dynamics based on contraction metrics, ensuring recursive feasibility, robust constraint satisfaction, and convergence while updating the GP online [4]. Second, Prajapat *et al.* develop a finite-sample reachability framework for GP models:

by sampling dynamics from the GP posterior they construct high-probability reachable sets, leading to a recursively feasible MPC with explicit safety and stability guarantees [5].

1.2.3 Lyapunov and Barrier Certificates

Complementary approaches derive *Lyapunov functions* and *control barrier functions (CBFs)* from data to certify safety. These functions define regions of attraction or forward-invariant safe sets that are enlarged as the system is explored. Wang *et al.* (2018) combined GP models with barrier certificates to ensure a quadrotor remained within a safe region while learning [6]. A quadratic-program-based safety filter minimally adjusted inputs to prevent constraint violations, while adaptive sampling expanded the certified region. More generally, control Lyapunov functions (CLFs) and CBFs can be learned for nonlinear systems with input-dependent uncertainties [7]. This category bridges reinforcement learning and formal control: safety is enforced at each step, and exploration is guided by gradually expanding certified regions. Recent surveys emphasize the growing integration of Lyapunov/barrier methods into safe RL algorithms [8].

1.2.4 Optimal-Control-Based Exploration

Beyond GP-centric methods, *optimal-control exploration* provides stronger guarantees on exploration completeness. A recent breakthrough is the *Safe Guaranteed Exploration (SAGE-MPC)* algorithm [9], which combines Lipschitz continuity bounds, GP confidence intervals, and receding-horizon planning. SAGE-MPC guarantees finite-time coverage of the safe domain while ensuring constraints are satisfied with probability $1 - \delta$. Unlike GP-MPC, where exploration is incidental, SAGE-MPC explicitly prioritizes *information gain*, planning exploratory trajectories that remain within a probabilistic control-invariant set (PCIS). The safe region is expanded iteratively, and theoretical results establish sample complexity bounds for exploration. Experiments on nonlinear systems (e.g. safe navigation of a car model through an unknown environment) validate its ability to achieve systematic, constraint-respecting coverage.

1.2.5 Connections to Kernel Methods and System Identification

The use of GPs in safe exploration also connects to classical *system identification*. In fact, GP regression in a reproducing kernel Hilbert space (RKHS) is mathematically equivalent to regularized least-squares estimation [10]. This provides a statistical foundation for the uncertainty bounds used in GP-MPC and GP safety filters, linking them with robust control theory. As a result, confidence intervals from GP regression can be interpreted as error bounds analogous to stability margins in robust control, leading to principled designs of safe controllers with quantified guarantees.

1.2.6 Non-GP Approaches and Safe Reinforcement Learning

Several non-GP approaches exist for safe exploration. *Reachability analysis* based on Hamilton–Jacobi equations computes exact safe sets for known dynamics, but suffers from scalability limitations in high dimensions [11]. *Optimism with backup policies* represents another strategy: here, the agent explores optimistically but always retains a recovery policy ensuring return to a pre-certified safe set [12]. Such frameworks, pioneered by Gillula and Tomlin (2011) and later generalized, guarantee zero constraint violations but may fail to cover the entire safe domain.

In reinforcement learning (RL), safe exploration is often cast as a *constrained MDP* problem, a Markov decision process $\mathcal{M} = (\mathcal{X}, \mathcal{U}, p, r, \gamma)$ where a policy π chooses inputs to maximize the discounted return $J(\pi) = \mathbb{E}[\sum_{k \geq 0} \gamma^k r(x_k, u_k)]$ under expected-cost constraints. Algorithms such as Constrained Policy Optimization (CPO) [13] enforce expected safety constraints but cannot guarantee per-step safety. To address this, hybrid *shielded RL* approaches combine learning with real-time safety filters (e.g. QP-based CBF filters), overriding unsafe actions [8]. These methods have been successfully applied to navigation and manipulation tasks, enabling RL agents to explore without incurring violations.

1.2.7 Practical Applications

Safe exploration algorithms have been demonstrated in increasingly realistic domains. GP-MPC controllers have been tested on miniature race cars driving near performance limits [3], while SAGE-MPC has been validated

on autonomous navigation tasks [9]. Quadrotors have used learned barrier certificates to safely fly through constrained environments [6]. These case studies highlight the feasibility of real-time safe learning, though typically in lower-dimensional systems. Consolidated tutorials [3] and surveys [14] now provide practitioners with guidelines for implementing safe learning controllers in robotics and industrial automation.

Summary

In summary, state-of-the-art safe exploration integrates machine learning models (GPs, neural networks) with control-theoretic safety mechanisms (invariant sets, MPC, barrier certificates). GP-MPC and safety filters provide probabilistic guarantees, Lyapunov/barrier methods certify and expand regions of attraction, and optimal-control frameworks such as SAGE-MPC ensure exploration completeness. Non-GP approaches, including reachability and shielded RL, further enrich the toolbox. Ongoing research seeks to improve scalability, reduce conservatism, and achieve robust, real-time safe learning in complex systems.

1.3 Gaps and challenges

Despite these advances, three fundamental obstacles remain:

1. *Scalability*: Exact GP inference scales cubically with data size; sparse approximations mitigate but do not remove this bottleneck in high-frequency control loops.
2. *Tightness versus conservatism*: Overly conservative confidence sets hinder exploration and degrade performance, whereas aggressive sets jeopardise safety.
3. *Unified analysis*: Existing guarantees are often tailored to specific architectures (barrier certificates, tube MPC, etc.) and do not directly extend to heterogeneous combinations of learning, planning and control.

1.4 Ethics and Sustainability

This work is motivated by the ethical imperative to advance industrial practice while reducing exposure of personnel to operational risk. By

certifying safety before execution and supervising actions automatically, the framework enables safe experiments and routine operations without requiring a human operator to be present during potentially hazardous phases, thereby shifting expert involvement toward oversight and analysis rather than manual intervention. From a sustainability standpoint, the approach is inherently lightweight: it relies solely on computational resources to run the code and does not consume physical materials, prototypes, or test batches beyond what a plant already produces under normal operation. Our focus on sample efficiency and real-time execution further limits computational overhead, so the environmental footprint is effectively bounded by the energy required to compute, which can be monitored and optimized in deployment.

1.5 Thesis objectives and contributions

The overarching goal of this thesis is to develop a *tractable* and *provably safe* exploration framework for uncertain, nonlinear systems. Building on the state of the art, we will:

1. Synthesize an online exploration framework that minimises conservatism by online adaptation of constraint tightenings and actively selected inducing points.
2. Provide finite time and asymptotic safety proofs that unify probabilistic invariance, Lyapunov stability and reachability under a common optimal control view.
3. Validate the proposed algorithms.

1.6 Structure of the thesis

Chapter 2 reviews Gaussian processes, kernel methods and current available framework in the literature.

Chapter 3 formalises the safe exploration problem and presents our unified safety analysis.

Chapter 4 reports extensive simulations experiments.

Chapter 5 synthesizes the empirical results, answers the research questions, and discusses practical implications for safe exploration.

Chapter 6 concludes with limitations and future research directions.

Through these contributions we aim to close the gap between theoretical safety guarantees and practical deployment of learning controllers in the wild, thereby pushing autonomous systems one step closer to trustworthy lifelong operation.

1.7 Research Question

How can we collect data in a safe manner from an unknown nonlinear model with respect to a linearized known model.

Chapter 2

Background

This chapter collects the core concepts used throughout: (i) Gaussian processes (GPs) for residual modeling, (ii) safety notions via (probabilistic) invariant sets, and (iii) Lyapunov/barrier tools used to certify stability and safety.

2.1 Safe Exploration in Nonlinear Systems

Safe exploration involves learning about the system while ensuring that safety constraints are not violated. This is a fundamental challenge in applications such as robotics, autonomous systems, and chemical process control. While model-free methods often lack formal safety guarantees, model-based approaches can incorporate prior knowledge and uncertainty estimates to ensure cautious behavior.

Recent advances such as SAGE-MPC [9] exemplify this paradigm by combining model predictive control (MPC) with theoretical guarantees on safe exploration. This enables controllers to explore new behaviors while remaining within safe operating regions. Integrating tools from control theory, such as reachability analysis and invariant set theory, with machine learning allows for formal safety reasoning in learned systems [15].

Integrating Control Theory and Machine Learning

Combining machine learning and control theory enables adaptive yet safe control of unknown nonlinear systems. Three core methodologies underlie

this integration:

- **Gaussian Process Regression (GPR):** A nonparametric Bayesian model that estimates system dynamics and their uncertainty from data. GPR provides a principled way to model unknown functions with quantified confidence. This uncertainty enables cautious planning in control [16].
- **Probabilistic Control Invariant Sets (PCIS):** Extend classical invariant set concepts to stochastic or uncertain systems. A PCIS defines a region in the state space that the system can remain within with high probability. This is used to enforce safety constraints during learning and exploration [15].
- **Control Lyapunov Functions (CLFs):** Provide a formal tool to guarantee stability of nonlinear systems. When combined with model uncertainty from GPR, CLFs can be used to synthesize controllers that stabilize the system while respecting confidence bounds on model predictions [17].

Together, these components form a comprehensive framework for safe exploration and learning in nonlinear control systems. The remaining chapters will explore their theoretical foundations, implementation strategies, and applications in robotics and industrial processes.

Model-free vs. model-based: trade-offs for safe learning

In learning control, approaches typically fall into two families. *Model-free* RL optimizes a policy directly from interactions, trading massive flexibility for high sample demands and weak a priori safety guarantees. *Model-based* methods learn (or use) a dynamics model and plan through it, which naturally exposes “safety hooks” (constraints, robustness, certificates) and lets us inject physics priors. Both families can be made safer, but they do so differently: model-free via shields or constrained training; model-based via uncertainty-aware planning, invariant sets, and Lyapunov/barrier certificates. This thesis focuses on the model-based side with a grey-box+ GP residual model, probabilistic control invariant sets, and CLF-based synthesis.

Table 2.1: Model-free vs model-based learning for control (high-level trade-offs).

	Model-free RL	Model-based (LBC / MBRL)
<i>What is learned</i>	Policy or value function directly (e.g., DQN, PPO, SAC).	Dynamics model $\hat{f}(x, u)$ (NN, GP, ensembles) + planner (e.g., MPC).
<i>Sample efficiency</i>	Often low; needs many interactions.	Higher; model reuse and planning amortize data.
<i>Safety hooks</i>	Shields/recovery policies; constrained RL; certificates are harder.	Chance constraints, robust tubes, invariant sets, CLF/CBF constraints integrate naturally.
<i>Use of priors</i>	Harder to inject physics/structure.	Easy: grey-box + residual learning (e.g., GP residuals).
<i>Online compute</i>	Cheap at runtime (policy eval).	Heavier (solve MPC/OC with uncertainty).
<i>Canonical reps</i>	DQN, PPO, SAC.	PILCO, PETS, Dreamer; GP-MPC; LBMPC.
<i>Failure modes</i>	Unsafe exploration; reward hacking; brittle tuning.	Model bias \Rightarrow optimistic plans; need calibrated uncertainty.

2.2 Model-Free Learning

Model-free reinforcement learning (RL) optimizes a feedback policy $\pi^* : x \mapsto u$ without constructing an explicit model of the plant. Two families dominate: value-based methods, which approximate the state–action value $Q(x, u)$ and act greedily (from Deep Q-Network (DQN) and its successors such as distributional and “Rainbow” variants), and policy-gradient / actor–critic methods, which directly optimize expected return via stochastic or deterministic policy updates (e.g., Deep Deterministic Policy Gradient (DDPG) for continuous control, Proximal Policy Optimization (PPO) for robust on-policy updates, and Soft Actor–Critic (SAC) for stable, entropy-regularized off-policy learning). [18, 19, 20, 21, 22, 23, 24]

The appeal of model-free RL is its domain-agnosticism and representational power: a single algorithmic template can be deployed across heterogeneous plants, and deep function approximators can express highly nonlinear policies and value functions. Empirically, modern actor–critic and distributional methods attain strong asymptotic performance across high-dimensional benchmarks once sufficient interaction is available. [18, 20, 23]

However, three limitations are central for safety-critical control. First, *sample inefficiency*: state-of-the-art model-free agents typically require from tens of thousands to millions of interactions, which is prohibitive on real equipment. Second, *no built-in safety*: unconstrained exploration risks violating hard state and input limits; formal guarantees are not native to the learning objective. Third, *difficulty of injecting priors*: physics knowledge and structural constraints are only indirectly shaped via rewards or architectures. Safety-aware variants therefore augment plain RL with constraints or runtime guards: constrained MDP formulations with policy search under explicit cost limits (e.g., Constrained Policy Optimization (CPO) and primal–dual approaches), formal-methods *shielding* that intercepts unsafe actions against temporal-logic specifications, and *recovery* policies that hand control to a safety controller in danger zones. While effective, these add-ons can introduce conservatism or brittleness if uncertainty is miscalibrated or specifications are incomplete, and remain an active area of benchmarking and evaluation. [13, 25, 26, 27, 28]

2.3 Model-Based Learning

Model-based learning maintains or learns a predictive dynamics model $\hat{f} : (x, u) \mapsto \dot{x}$ and selects actions by reasoning through that model. Two complementary streams dominate current practice. First, deterministic neural world models, from ensemble-based PETS with shooting and CEM [29] to latent-dynamics planners such as PlaNet [30], Dreamer variants [31, 32], TD-MPC2 [33], and tree-search with learned models in MuZero [34], trade modeling bias for substantial sample-efficiency, often planning in a learned latent space. Second, probabilistic models, most prominently Gaussian processes (GPs), provide mean and state-dependent uncertainty that can be propagated during planning; in learning-based MPC this enables chance constraints and cautious control [35, 36, 37]. Between these extremes, grey-box residual learning augments first-principles models with data-driven terms (e.g., GP residuals), combining physics priors with flexible function approximation [35, 38].

A typical pipeline couples *dynamics learning* with *planning/control*. Neural models are trained by multi-step prediction losses and then used either for trajectory optimization (shooting/CEM), value-imagination with an actor–critic (Dreamer), or short model rollouts to aid off-policy RL (MBPO-style) to limit compounding error [29, 32, 39]. For GP models, the learned posterior

$(\hat{\mu}, \hat{\sigma})$ feeds a stochastic or robust MPC, tightening state/input constraints by uncertainty propagation [35, 37, 36].

This paradigm offers natural *safety hooks*: predicted trajectories and their uncertainty allow enforcing hard constraints via chance constraints, terminal sets, or predictive safety filters. Provably safe exploration with high probability has been demonstrated in learning-based MPC using GP confidence sets [40, 41], and recent sampling-based GP-MPC establishes finite-sample reachability bounds for safe closed-loop operation [42]. Physics priors are straightforward to embed as nominal models with learned residuals, yielding data efficiency and interpretability [35, 38].

The main liabilities are well known. *Model bias* can induce over-optimistic plans; practical remedies include ensembles, short-horizon imagination, and uncertainty-aware objectives [39, 29]. And *online compute* can be heavier than a direct policy evaluation, especially with chance constraints or sample-based uncertainty propagation; nevertheless, modern learning-based MPC stacks report real-time performance on nontrivial systems [36, 37]. In this thesis we adopt the probabilistic route, using GP posteriors as safety-aware surrogates within MPC-style constraints while allowing the exploration policy to benefit from predictive models.

2.4 Gaussian Process

Gaussian Process Regression (GP): GPs are a non-parametric Bayesian learning technique used to model unknown functions and quantify uncertainty in the predictions. In the context of control, GP regression can learn the system’s dynamics (mapping from state-input to state derivatives or next state) from data, while providing a measure of confidence (variance) for each prediction. This is particularly useful for nonlinear systems with partially unknown physics, as the GP can capture unmodeled effects or disturbances. The uncertainty information enables the controller to perform cautious planning: for example, an MPC controller using a GP model can plan actions that account for worst-case model errors, thereby maintaining robustness. The recent tutorial by Wang and Zhang provides a comprehensive overview of GP-based MPC, demonstrating how GPs improve predictive accuracy and robustness in control tasks like robotic path-following on uneven terrain [16]. In our work, GP models serve as the learning component, continually refined with new data to reduce model uncertainty, and their predictive distributions feed into safety analysis tools (like PCIS and CLF constraints) to ensure that the learning-enabled controller remains safe as it

adapts.

GP Formulation

A Gaussian process is fully characterized by its mean function $m(x)$ and covariance function $k(x, x')$:

$$\Delta(x) \sim \mathcal{G}\mathcal{D}(m(x), k(x, x')) \quad (2.1)$$

where $\Delta(x)$ represents the residual dynamics. Typically, the mean function is assumed to be zero, $m(x) = 0$, reflecting no prior knowledge about the bias in the residual.

Given observed data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, the posterior distribution at a new input x_* is a Gaussian distribution described by:

$$\Delta(x_*)|\mathcal{D} \sim \mathcal{N}(\mu(x_*), \sigma^2(x_*)) \quad (2.2)$$

with posterior mean and variance given by:

$$\mu(x_*) = k_*^T (K + \sigma_n^2 I)^{-1} y \quad (2.3)$$

$$\sigma^2(x_*) = k(x_*, x_*) - k_*^T (K + \sigma_n^2 I)^{-1} k_* \quad (2.4)$$

where $k_* = [k(x_*, x_1), \dots, k(x_*, x_N)]^T$, K is the covariance matrix constructed from the training inputs, σ_n^2 is the noise variance, and I is the identity matrix.

Kernel Selection

The kernel $k(x, x')$ encodes assumptions on the residual dynamics like smoothness, characteristic length scales, periodicity, and how different inputs interact. We work with stationary kernels for local generalization and add simple nonstationary structure when needed. The choice of kernel function $k(x, x')$ encodes our assumptions about the smoothness and behavior of the residual dynamics. A widely used kernel is the Radial Basis Function (RBF) or Squared Exponential kernel, defined as:

$$k(x, x') = \sigma_f^2 \exp\left(-\frac{1}{2l^2} \|x - x'\|^2\right) \quad (2.5)$$

where σ_f^2 is the signal variance, and l is the length-scale parameter controlling the smoothness of the predictions. A more in-depth discussion of kernels can

be found in the Appendix C.

Hyperparameter Optimization

Hyperparameters (length-scale l , signal variance σ_f^2 , and noise variance σ_n^2) are optimized by maximizing the marginal likelihood of the observed data \mathcal{D} :

$$\log p(y|X, \theta) = -\frac{1}{2}y^T(K + \sigma_n^2 I)^{-1}y - \frac{1}{2}\log|K + \sigma_n^2 I| - \frac{N}{2}\log(2\pi) \quad (2.6)$$

where θ denotes the set of hyperparameters. Gradient-based optimization methods, such as conjugate gradients or the L-BFGS limited-memory BFGS quasi-Newton method [43, 44], are typically used to find optimal values efficiently.

2.5 GP prediction and model uncertainty

The key advantage of using GPs is the inherent quantification of uncertainty. A GP returns a mean $\mu(x_*)$ and a predictive variance $\sigma^2(x_*)$ at any query x_* . The variance measures how unsure the model is: it is small near well-covered data and large in unexplored regions. The predictive variance represents confidence bounds that are crucial for making safe control decisions. This uncertainty directly informs constraint tightening in probabilistic control invariant sets (PCIS) and control Lyapunov functions (CLFs), ensuring robust safety guarantees.

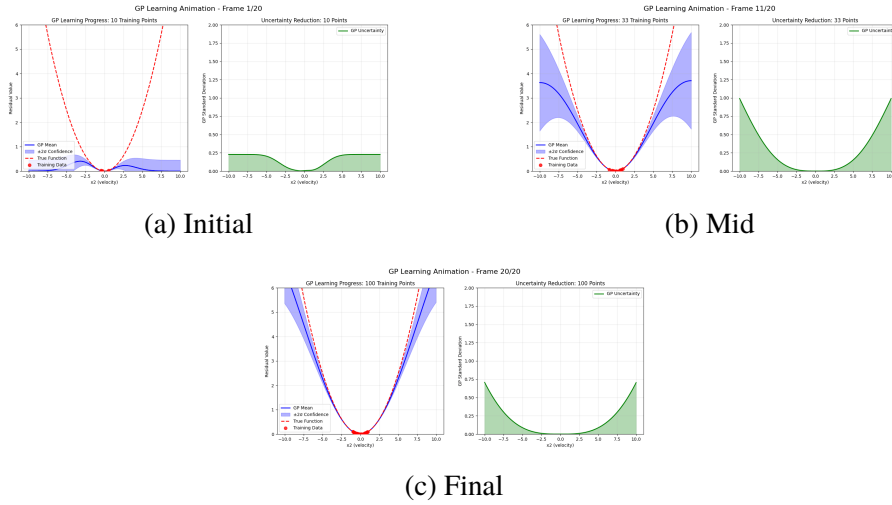


Figure 2.1: Gaussian Process learning snapshots: posterior mean and uncertainty at the beginning, mid-way, and after training.

2.5.1 Sparse Gaussian Processes for Scalability

Exact GP regression scales as $O(N^3)$ in time and $O(N^2)$ in memory for N training points, which quickly becomes impractical as data grow. Sparse GP methods address this by introducing $M \ll N$ *inducing variables* located at inputs $Z = \{z_m\}_{m=1}^M$. Representative approaches include pseudo-inputs (SPGP) [45], variational inducing points [46], and stochastic variational GPs (SVGP) for minibatch training [47]. With inducing variables $u = f(Z)$ and variational posterior $q(u) = \mathcal{N}(m, S)$, the resulting complexity is $O(NM^2 + M^3)$ for training and $O(M)$ per test prediction, while retaining calibrated uncertainty under standard assumptions.

A typical predictive approximation at a new input x_* takes the form

$$\mu(x_*) \approx k_{*Z} K_{ZZ}^{-1} m, \quad \sigma^2(x_*) \approx k_{**} - k_{*Z} (K_{ZZ}^{-1} - S^{-1}) k_{Z*},$$

where $K_{ZZ} = k(Z, Z)$, $k_{*Z} = k(x_*, Z)$, and (m, S) are learned. In practice, Z can be initialised by simple heuristics (e.g., k -means) and refined during training. The main trade-off is that larger M improves fidelity but increases cost; modern implementations choose M to meet computational budgets while monitoring both accuracy and uncertainty calibration.

2.5.2 High-confidence GP–UCB bound and its role in safety

We use the Gaussian process upper confidence bound (GP–UCB) to convert statistical uncertainty into *high-probability* margins. Under standard assumptions (bounded RKHS norm and sub-Gaussian noise), the GP posterior satisfies, uniformly over a finite design set, the concentration inequality

$$|\Delta(z) - \mu_t(z)| \leq \beta_t \sigma_t(z) \quad \text{for all } z \in Z_t \quad \text{with probability at least } 1 - \delta, \quad (2.7)$$

where $z = (x, u)$, μ_t, σ_t are the GP mean and standard deviation after t samples, and $\beta_t = O(\sqrt{\gamma_t + \ln(1/\delta)})$ grows with the information capacity γ_t of the kernel (Srinivas et al., 2010; Chowdhury & Gopalan, 2017).*

Embedding (2.7) into constraints yields the *UCB safety margin*

$$\Delta_{\text{gp}}(z) = \beta_t \sigma_t(z), \quad (2.8)$$

so that deterministic tightenings like $g(x, u, \mu_t(z)) + \Delta_{\text{gp}}(z) \leq 0$ enforce chance constraints at level $1 - \delta$. This also defines the admissible set

$$\mathcal{S}_t = \{(x, u) : g(x, u, \mu_t(z)) + \beta_t \sigma_t(z) \leq 0\}, \quad (2.9)$$

which is the set we certify for operation.

Safe-UCB exploration. Within \mathcal{S}_t we choose informative actions by prioritising high-uncertainty queries,

$$z_{t+1} \in \arg \max_{z \in \mathcal{S}_t} \sigma_t(z),$$

(or a task-specific acquisition), so that learning rapidly *reduces* σ_t and the certified region \mathcal{S}_t *expands* over time (Sui et al., 2015). In summary, GP–UCB provides the confidence envelopes that make our probabilistic safety certificates valid, while the Safe-UCB rule exploits those envelopes to explore aggressively *inside* the certified set.

*For vector residuals the bound holds component-wise; a safe normed form is $|x^\top P[\Delta(z) - \mu_t(z)]| \leq \beta_t \|Px\|_2 \|\sigma_t(z)\|_2$.

2.6 Control-Invariant Sets

Before introducing probabilistic (chance) invariance, we recall the deterministic, continuous-time notion of control-invariant sets which are subsets of the state space that can be kept forward-invariant by suitable inputs despite constraints and, in the robust case, bounded disturbances. Consider $\dot{x} = f(x, u) + d(x, t)$ with $x \in \mathbb{X} \subset \mathbb{R}^n$, $u \in \mathbb{U} \subset \mathbb{R}^m$, and an admissible disturbance $d \in \mathbb{D}$. A closed set $S \subseteq \mathbb{X}$ is (positively) invariant under a feedback κ if for $\dot{x} = f(x, \kappa(x))$ every trajectory starting in S remains in S for all $t \geq 0$; it is a controlled invariant set (CIS) if for every $x \in S$ there exists an admissible input $u(\cdot)$ such that the solution remains in S for all $t \geq 0$; it is a robust CIS (RCIS) if the same holds uniformly for all $d \in D$. When $S = \{x : h(x) \leq 0\}$ with $h \in C^1$ and outward normal ∇h on ∂S , a sufficient boundary condition for controlled invariance is

$$\forall x \in \partial S \exists u \in U : \mathcal{L}_f h(x, u) := \nabla h(x)^\top f(x, u) \leq 0, \quad (2.10)$$

and for robust controlled invariance under additive disturbances bounded in ,

$$\forall x \in \partial S \exists u \in U : \sup_{d \in D} \nabla h(x)^\top (f(x, u) + d) \leq 0. \quad (2.11)$$

For control-affine dynamics $f(x, u) = f_0(x) + g(x)u$, (2.10) reduces to $\inf_{u \in U} \nabla h(x)^\top (f_0(x) + g(x)u) \leq 0$ on ∂S , which becomes a tractable convex check when U is convex. A standard constructive deterministic case is quadratic (ellipsoidal) RCIS for linear dynamics with state feedback: for $\dot{x} = (A + BK)x + d$ with $\|d\|_2 \leq \bar{d}$ and $E_\alpha = \{x : x^\top P x \leq \alpha\}$ with $P \succ 0$ and $V(x) = x^\top P x$,

$$\dot{V} = x^\top ((A+BK)^\top P + P(A+BK))x + 2x^\top P d,$$

and if $(A+BK)^\top P + P(A+BK) \leq -2\lambda P$ for some $\lambda > 0$, then for all x with $V(x) = \alpha$,

$$\dot{V} \leq -2\lambda \alpha + 2 \|Px\|_2 \bar{d} \leq -2\lambda \alpha + 2 \sqrt{\lambda_{\max}(P)} \sqrt{\alpha} \bar{d},$$

so $\dot{V} \leq 0$ on ∂E_α provided $\alpha \geq \lambda_{\max}(P) \bar{d}^2 / \lambda^2$; if additionally $E_\alpha \subseteq \mathbb{X}$ and $K, E_\alpha \subseteq \mathbb{U}$, then E_α is an RCIS. The construction is illustrated in Fig. 2.2, which shows the largest ellipsoidal RCIS contained in a state box under linear feedback.

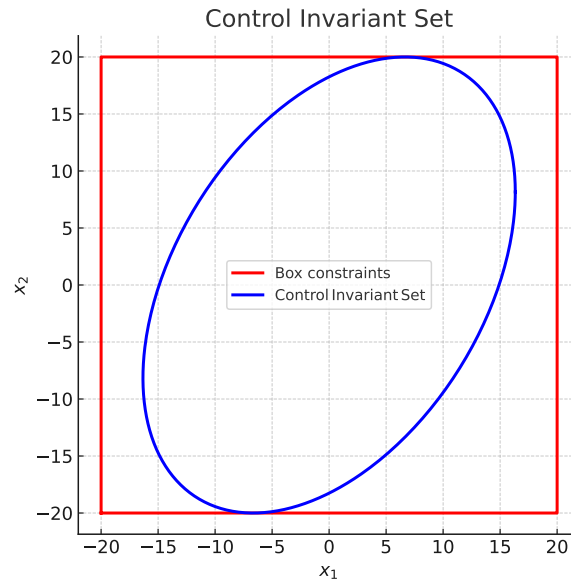


Figure 2.2: Largest robust control-invariant ellipsoid $\alpha = \{x : x^\top P x \leq \alpha\}$ contained in box state constraints \mathbb{X} under linear feedback $u = Kx$. The ellipsoid is chosen so that $\dot{V} \leq 0$ on ∂_α despite bounded disturbances, yielding a deterministic RCIS inside \mathbb{X} .

2.7 Probabilistic Control Invariant Set

Probabilistic Control Invariant Sets (PCIS): PCIS extend the classical notion of invariant sets to uncertain systems by ensuring invariance with high probability. Formally, a set S is a (δ) -PCIS if, given the current model of the system, the probability that the state remains in S under an appropriate control policy is at least $1 - \delta$ for all time. Computing such sets involves analyzing the one-step reachable sets under uncertainty and iterating this process. Gao, Johansson, and Xie introduced efficient algorithms to calculate PCIS for both finite-horizon and infinite-horizon cases, showing that one can start from a known robust invariant set and expand it probabilistically [15]. PCIS are crucial for safe learning because they formally delineate the safe region for exploration: the learning agent can be allowed to explore freely inside the PCIS (since by definition it can be kept there safely), but any action that would take the state outside of the PCIS is forbidden. In our approach, after learning an initial GP model of the dynamics, we compute an inner approximation of a PCIS for the system. This safe set is then used as a constraint in the learning control algorithm (for example, as a terminal condition in MPC or as a filter

on planned trajectories). As learning progresses and the model uncertainty shrinks, the PCIS can be expanded, gradually unlocking more of the state space for exploration once it is verified to be safe. This ensures a systematic and provably safe exploration process grounded in control theory. For Gaussian Process we use high-probability confidence sets to upper bound the model error and certify an *inner* PCIS. As data accumulate, posterior uncertainty shrinks and the certified S expands.

We say that S is a δ -probabilistic control-invariant set (δ -PCIS) if

$$\forall x(t=0) \in S : \mathbb{P}_{x_0}^{\kappa}(x(t) \in S \quad \forall t \geq 0) \geq 1 - \delta.$$

For $\delta = 0$ this reduces to the classical (deterministic) control-invariant set.

2.8 Linear Quadratic Regulator

This section recalls the Linear Quadratic Regulator (LQR) as the unconstrained, infinite-horizon baseline used throughout the thesis. LQR furnishes (i) a stabilizing state-feedback $u = -Kx$ for the nominal linear model, (ii) a quadratic Lyapunov function $V(x) = x^{\top}Px$ that we will reuse as a certificate in safety analyses (CLF and invariant sets), and (iii) a principled way to select weights reflecting performance vs. actuation effort.

The Linear Quadratic Regulator (LQR) is the canonical optimal controller for *linear* time-invariant systems subject to a quadratic cost. For the *continuous-time* linear system

$$\dot{x}(t) = Ax(t) + Bu(t), \quad (2.12)$$

with symmetric weights $Q \geq 0$ and $R > 0$, the infinite-horizon quadratic cost is

$$J = \int_0^{\infty} (x^{\top}(t) Q x(t) + u^{\top}(t) R u(t)) dt. \quad (2.13)$$

Minimising (2.13) yields the state-feedback law

$$u(t) = -Kx(t), \quad K = R^{-1}B^{\top}P,$$

where $P > 0$ is the unique stabilising solution of the *Continuous-Time Algebraic Riccati Equation* (CARE)

$$A^{\top}P + PA - PBR^{-1}B^{\top}P + Q = 0. \quad (2.14)$$

A classical derivation is provided in [48].

In the next chapter this continuous-time LQR design serves as a *baseline stabiliser* and an excitation mechanism: the feedback gain K initialises our safe-exploration controller and supplies the Lyapunov matrix P used to construct both the Control Lyapunov Function $V(x) = x^\top P x$ and the probabilistic control-invariant sets.

2.9 Lyapunov Controller

A Control Lyapunov Function is a stabilizing Lyapunov function used to design feedback controllers. If one can find a suitable Lyapunov function $V(x)$ for the system (with $V(x) > 0$ for all non-zero states and \dot{V} controllably negative), then ensuring $\dot{V}(x) < 0$ at each time step guarantees asymptotic stability of the closed-loop system at the desired equilibrium. CLFs thus provide a condition that any safe control action must satisfy (to keep the system converging to a stable point or within a stable manifold). In the context of unknown dynamics, CLFs can be combined with learned models to enforce stability probabilistically. For instance, the work of Castañeda et al. uses a GP to model the system and derives an optimization-based controller that satisfies a chance constraint on the CLF decrease [17]. In simpler terms, at each control step the algorithm chooses the control input that minimizes deviation from the nominal controller but still guarantees, with high probability, that the Lyapunov function will decrease. This results in a min-norm stabilizing controller that adapts to model uncertainty: as the GP model becomes more accurate, the controller approaches the performance of a fully known system; during learning transients, the CLF constraints guard against instability. In our safe learning framework, CLFs will be used to define stability objectives and as part of the safety filter. By ensuring all exploratory actions also respect a CLF condition, we maintain not just instantaneous safety (constraint satisfaction) but long-term safety in the sense of eventual convergence or boundedness of trajectories. The CLF thus complements the PCIS: while the PCIS defines the allowable region, the CLF provides a rule to steer the system within that region safely and ultimately towards desired targets.

Chapter 3

Methods

The methodology chapter details the systematic approach adopted to achieve safe exploration using Gaussian process (GP) regression, probabilistic control invariant sets (PCIS), and control Lyapunov functions (CLFs). This chapter is structured into clear sections outlining each critical component of the proposed framework.

3.1 Gaussian Process Regression

Gaussian processes (GPs) offer a robust, non-parametric Bayesian approach to modeling unknown functions with associated uncertainty. In our setting, GPs model the residual dynamics, the discrepancy between the known linearized model and the true nonlinear system dynamics.

3.1.1 GP Formulation

A Gaussian process is fully characterized by its mean function $m(x)$ and covariance function $k(x, x')$:

$$\Delta(x) \sim \mathcal{G}\mathcal{D}(m(x), k(x, x')) \quad (3.1)$$

where $\Delta(x)$ represents the residual dynamics. Typically, the mean function is assumed to be zero, $m(x) = 0$, reflecting no prior knowledge about the bias in the residual.

Given observed data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, the posterior distribution at a new input x_* is a Gaussian distribution described by:

$$\Delta(x_*)|\mathcal{D} \sim \mathcal{N}(\mu(x_*), \sigma^2(x_*)) \quad (3.2)$$

with posterior mean and variance given by:

$$\mu(x_*) = k_*^T (K + \sigma_n^2 I)^{-1} y \quad (3.3)$$

$$\sigma^2(x_*) = k(x_*, x_*) - k_*^T (K + \sigma_n^2 I)^{-1} k_* \quad (3.4)$$

where $k_* = [k(x_*, x_1), \dots, k(x_*, x_N)]^T$, K is the covariance matrix constructed from the training inputs, σ_n^2 is the noise variance, and I is the identity matrix.

3.1.2 Kernel Selection

The choice of kernel function $k(x, x')$ encodes our assumptions about the smoothness and behavior of the residual dynamics. A widely used kernel is the Radial Basis Function (RBF) or Squared Exponential kernel, defined as:

$$k(x, x') = \sigma_f^2 \exp\left(-\frac{1}{2l^2} \|x - x'\|^2\right) \quad (3.5)$$

where σ_f^2 is the signal variance, and l is the length-scale parameter controlling the smoothness of the predictions.

3.1.3 Hyperparameter Optimization

Hyperparameters (length-scale l , signal variance σ_f^2 , and noise variance σ_n^2) are optimized by maximizing the marginal likelihood of the observed data \mathcal{D} :

$$\log p(y|X, \theta) = -\frac{1}{2} y^T (K + \sigma_n^2 I)^{-1} y - \frac{1}{2} \log |K + \sigma_n^2 I| - \frac{N}{2} \log(2\pi) \quad (3.6)$$

where θ denotes the set of hyperparameters. Gradient-based optimization methods, such as conjugate gradients or L-BFGS, are typically used to find optimal values efficiently.

3.1.4 Uncertainty Quantification

The key advantage of using GPs is the inherent quantification of uncertainty. The GP provides a predictive variance $\sigma^2(x_*)$ alongside predictions, representing confidence bounds that are crucial for making safe control decisions. This uncertainty directly informs constraint tightening in probabilistic control invariant sets (PCIS) and control Lyapunov functions (CLFs), ensuring robust safety guarantees.

3.2 Probabilistic Control Invariant Sets

Probabilistic Control Invariant Sets (PCIS) define state-space regions from which the system can remain within prescribed constraints under the chosen control strategy with high probability. We present two formulations:

3.2.1 Direct GP-Based Prediction

Safe states are identified directly from GP predictions by checking if the predicted residual and associated uncertainty lie within specified bounds. Specifically, a state x is considered safe if:

$$\mathcal{S}_{\text{safe}} = \{x \in \mathbb{R}^n \mid x^\top P x \leq \alpha, \mu(x) \leq \mu_{\text{max}}, \sigma(x) \leq \sigma_{\text{max}}\}. \quad (3.7)$$

where P is a positive definite matrix that is a Lyapunov candidate for the linearized system, $\mu(x)$ and $\sigma(x)$ are the GP predicted mean and standard deviation, respectively, and μ_{max} , σ_{max} are predefined safety threshold that depend on the application.

3.2.2 Optimization-Based PCIS

Alternatively, safe states and corresponding control inputs can be determined by solving a constrained optimization problem. This approach involves finding states x and inputs u that satisfy stability without violating the constraint using the slack variable s :

$$\begin{aligned} \min_{u \in \mathbb{R}, s \geq 0} \quad & \|u - \bar{u}\|_2^2 + \rho s & (3.8) \\ \text{s. t.} \quad & \dot{V}(x) + \lambda V(x) \leq s, \\ & u_{\min} \leq u \leq u_{\max}, \end{aligned}$$

where

$$\dot{V}(x) = x^\top (A^\top P + PA)x + 2x^\top P B u + 2x^\top P \mu(x) \quad (3.9)$$

$$+ 2\beta \|P x\|_2 \sigma(x), \quad (3.10)$$

Considering the value of s we know a priori if a state is safe or not. The optimization ensures both safety and performance criteria are simultaneously met.

3.3 Control Lyapunov Functions

Our approach builds on the classical Linear Quadratic Regulator (LQR) controller as the baseline control method.

3.3.1 Linear Quadratic Regulator (LQR) Baseline

LQR provides optimal control by minimizing the quadratic cost function:

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (3.11)$$

where Q and R are positive definite weighting matrices for states and inputs, respectively. The solution to this optimal control problem yields a state-feedback controller:

$$u = -Kx \quad (3.12)$$

where the gain matrix K is computed by solving the algebraic Riccati equation (ARE):

$$A^T P + P A - P B R^{-1} B^T P + Q = 0 \quad (3.13)$$

and the optimal feedback gain is:

$$K = R^{-1} B^T P \quad (3.14)$$

3.3.2 GP-Augmented CLF Controller

Building upon the LQR baseline, a Control Lyapunov Function (CLF) controller integrates GP uncertainty, ensuring stability in the presence of model inaccuracies:

$$\begin{aligned} \min_{u \in \mathbb{R}, s \geq 0} \quad & \|u - \bar{u}\|_2^2 + \rho s \\ \text{s. t.} \quad & \dot{V}(x) + \lambda V(x) \leq s, \\ & u_{\min} \leq u \leq u_{\max}, \end{aligned} \quad (3.15)$$

where

$$\dot{V}(x) = x^T (A^T P + P A) x + 2 x^T P B u + 2 x^T P \mu(x) \quad (3.16)$$

$$+ 2 \beta \|P x\|_2 \sigma(x), \quad (3.17)$$

where A, B represent the known system matrices, P is a positive matrix from Riccati equation, and \bar{u} is a nominal (e.g. tracking) input. The slack s guarantees feasibility; a large weight $\rho \gg 1$ discourages its use.

Solving the QP yields a control action u^* that *minimally deviates* from \bar{u} while certifying that the Lyapunov function decreases in the presence of GP uncertainty.

The following section explains how these components are integrated into a cohesive framework, including practical considerations and implementation specifics. It will also outline the algorithmic steps for real-time deployment in robotic and industrial applications.

3.4 Problem Statement and System Model

In this section we unpack each component of the safe-learning framework, explain its role, and show how the building blocks fit together into a provably safe and efficient exploration algorithm.

3.4.1 True Nonlinear Process Form

We consider a system with nonlinear dynamics

$$\dot{x}(t) = f(x(t), u(t)), \quad (3.18)$$

where $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$, is unknown. We assume that all unmodeled nonlinearities enter only through the state x ; the dependence on the control u is captured by a known linear term. Thus, writing a first-order Taylor expansion we have

$$f(x, u) = Ax + Bu + \underbrace{\Delta(x)}_{\substack{\text{unknown residual} \\ \text{nonlinear in } x}},$$

so the true plant deviates from the nominal linear model

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m, \quad (3.19)$$

by an *unknown residual*

$$\dot{x}(t) = Ax(t) + Bu(t) + \Delta(x(t)), \quad (3.20)$$

where

- $x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m,$

- $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ are the Jacobian matrices obtained from a local linearisation,
- $\Delta : \mathbb{R}^n \rightarrow \mathbb{R}^n$ captures all state-dependent nonlinear effects.

Hard Constraints

State and input must remain inside hyper-rectangles for the true system 3.18

$$X = \{x \in \mathbb{R}^n \mid |x_i| \leq x_{i,\max}, i = 1, \dots, n\}, \quad (3.21)$$

$$U = \{u \in \mathbb{R}^m \mid u_{j,\min} \leq u_j \leq u_{j,\max}, j = 1, \dots, m\}. \quad (3.22)$$

All control actions must satisfy $u(t) \in U$ and we require $x(t) \in X$ for all t .

3.4.2 Objective

Design a framework that simultaneously

1. learns $\Delta(x)$ online via Gaussian process (GP) regression,
2. guarantees $x(t) \in X$ for all t with probability at least $1 - \delta$,
3. expands the safe operating region *efficiently*, and
4. provides Lyapunov stability guarantees.

Gaussian-Process Model of the Residual

We leverage Gaussian-process regression to model the unknown residual $\Delta(x)$ for several key reasons. First, GPs are nonparametric function approximators, which means they can flexibly represent a wide range of smooth nonlinearities without the need to specify a fixed parametric form in advance. Second, GPs provide closed-form expressions for both the posterior mean $\mu_t(x)$ and variance $\sigma_t^2(x)$, enabling us to quantify epistemic uncertainty directly and thereby derive high-probability safety certificates. Third, the Bayesian nature of GPs allows us to update our belief about $\Delta(x)$ online in a principled way as new residual samples are collected, ensuring sample-efficient learning. These properties make the Gaussian-process model a natural choice for safely learning the unknown dynamics in real time.

3.5 Probabilistic Safety Framework

Before diving into the technical details, we motivate why a probabilistic approach to safety is essential. In many real-world systems, unmodelled nonlinearities and disturbances can cause significant deviations from nominal behavior. Relying on deterministic worst-case bounds often leads to overly conservative controllers that severely limit performance and exploration. By contrast, a probabilistic safety framework leverages statistical uncertainty quantification, provided here by Gaussian-process regression, to balance risk and performance. We can then derive high-confidence certificates that guarantee constraint satisfaction with probability $1 - \delta$, while still allowing the controller to learn and expand its safe operating region efficiently. This blend of learning and chance-constrained reasoning is the cornerstone of our Safe-UCB exploration algorithm.

3.5.1 High-Probability Confidence Bound

To safely incorporate the GP model into control, we need a uniform, high-confidence bound on the residual $\Delta(x)$. The GP-UCB framework provides exactly this via the parameter $\beta_t(\delta)$ [49, 50]. Concretely, after t samples we have

$$|\Delta(x) - \mu_t(x)| \leq \beta_t(\delta) \sigma_t(x) \quad \text{with probability at least } 1 - \delta, \quad (3.23)$$

where $\mu_t(x), \sigma_t(x)$ are the GP posterior mean and standard deviation at x , and

$$\beta_t(\delta) = \sqrt{2 \ln\left(\frac{\pi_t^2}{6\delta}\right)}, \quad \pi_t^2 = \frac{\pi^2}{6} + t^2 \frac{\pi^2}{6}. \quad (3.24)$$

A conservative, dimension-independent alternative is

$$\beta_t(\delta) = \sqrt{2 \ln(2n/\delta)}.$$

3.5.2 One-Step Safe Set

Now that we have a probabilistic guarantee on the residual modelled and we want to use it for defining a probabilistic control invariant set which is safely explorable.

Definition 3.5.1 (One-Step Safe Set) *A state x is one-step safe at time t iff*

there exists a control $u \in U$ such that

$$\Pr\{x(t+1) \in X \mid x(t) = x, u(t) = u\} \geq 1 - \delta.$$

Under the Gaussian model $x(t+1) \sim \mathcal{N}(Ax + Bu + \mu_t(x), \sigma_t^2(x)I)$, this amounts to the deterministic condition

$$|[Ax + Bu + \mu_t(x)]_i| + \beta_t(\delta) \sigma_t(x) \leq x_{i,\max}, \quad i = 1, \dots, n. \quad (3.25)$$

We further cap exploration by requiring $\sigma_t(x) \leq \sigma_{\max}$.

Definition 3.5.2 (Admissible safe set at iteration t) Let X be the hard state box (3.21), U the admissible-input set (3.22), $\mu_t(\cdot), \sigma_t(\cdot)$ the GP posterior after t samples, $\beta_t(\delta)$ the confidence radius (3.24), and σ_{\max} the exploration budget. Then

$$S_{t,\text{safe}} = \left\{ x \in X \mid \begin{array}{l} \|\sigma_t(x)\|_{\infty} \leq \sigma_{\max}, \\ \exists u \in U : |Ax + Bu + \mu_t(x)| + \beta_t(\delta) \sigma_t(x) \leq x_{\max} \end{array} \right\}.$$

where the absolute value and inequality are understood element-wise. In words, $S_{t,\text{safe}}$ is the set of states that (i) already satisfy the hard box constraint, (ii) have sufficiently small GP uncertainty, and (iii) admit at least one control input that keeps the next state inside the box with probability $1 - \delta$.

3.5.3 LQR Baseline and Lyapunov Function

For the linear part (3.19) choose weights $Q \geq 0$, $R > 0$ and minimise $J = \int_0^{\infty} (xQx + uRu) dt$. The optimal feedback is $u = -Kx$, $K = R^{-1}BP$, where P solves $AP + PA - PBR^{-1}BP + Q = 0$. Define the Lyapunov function

$$V(x) = xPx. \quad (3.26)$$

3.5.4 Largest Invariant Ellipsoid

The maximum-volume ellipsoid contained in X for the linearized system (3.19) is

$$E(\alpha_{\max}) = \{x \in \mathbb{R}^n : xPx \leq \alpha_{\max}\}, \quad \alpha_{\max} = \min_i \frac{x_{i,\max}^2}{e_i^T P^{-1} e_i}, \quad (3.27)$$

with e_i the i -th canonical basis vector.

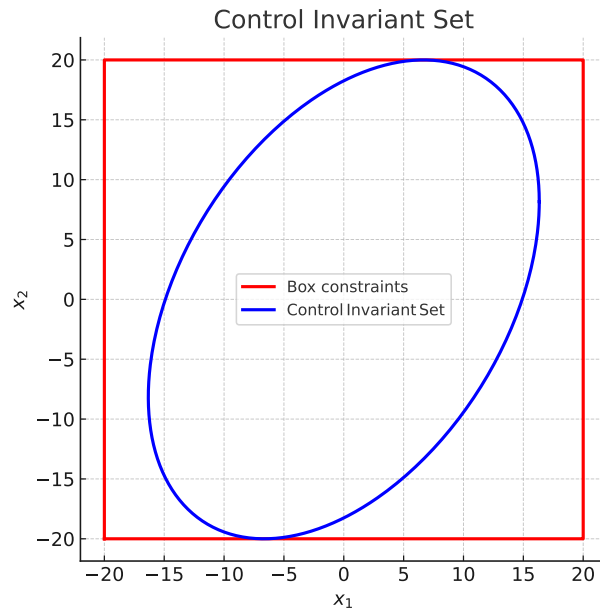


Figure 3.1: Largest control-invariant ellipsoid $E(\alpha_{\max})$ (blue) entirely contained in the polyhedral constraint set X (red) for the linearised system.

3.5.5 Probabilistic Control-Invariant Set Inside the Ellipsoid

Having computed the largest invariant ellipsoid

$$E(\alpha_{\max}) = \{x \in \mathbb{R}^n : x^\top P x \leq \alpha_{\max}\},$$

we can further restrict our one-step safe set to lie inside this nominally invariant region. In particular, define the *probabilistic control-invariant set* at iteration t as

$$\tilde{S}_{t,\text{inv}} = S_{t,\text{safe}} \cap E(\alpha_{\max}) = \left\{ x \in X \mid \begin{array}{l} x^\top P x \leq \alpha_{\max}, \\ \|\sigma_t(x)\|_\infty \leq \sigma_{\max}, \\ \exists u \in U : |Ax + Bu + \mu_t(x)| + \beta_t \sigma_t(x) \leq x_{\max} \end{array} \right\}.$$

By intersecting with $E(\alpha_{\max})$, we ensure that even under the nominal LQR policy $u = -Kx$ the state remains in a region that is invariant for the linearized dynamics. By retaining the GP-based one-step safety condition, we additionally guarantee that each transition from x can be kept within the

hard box X with probability at least $1 - \delta$. Hence $\tilde{S}_{t,\text{inv}}$ is a *probabilistic control-invariant* set: once $x(t)$ lies in $\tilde{S}_{t,\text{inv}}$, we can choose controls that both (i) respect the nominal invariant ellipsoid and (ii) satisfy the high-probability safety constraints at every step.

In practice, all subsequent exploration and control actions (including the GP-augmented CLF–QP, Eq. (3.28)) are restricted to $\tilde{S}_{t,\text{inv}}$, thereby combining the robust invariance of the linear LQR design with the adaptive safety provided by the GP model.

3.6 Control-Lyapunov Function Framework

We now have a mathematical formulation to define safely explorable states for a true process. We need a controller that can drive the system to a desired target selected from the safe set. The desired target is chosen from the safe set using an activation function that select the most uncertain state, which can be seen as the state with the most information gain for the GP:

$$x_t^* \leftarrow \arg \max_{x \in S_{t,\text{safe}}} \sigma_t(x)$$

It is important to mention the key role of the activation function. Not all target are safely reachable and what may eventually happen is that an incorrect chosen target is critical and can damage the system that we are controlling. It is easy to understand from figure 3.2 that the trajectory is diverging and can pose a serious problem to our system.

The idea is to use a Control-Lyapunov Function controller that can be rapidly solved using Quadratic Programming (QP), which is presented in the following section.

3.6.1 GP-Augmented CLF–QP

The following optimization problem can be solved to drive the system to the desired target x_t^*

$$\min_{u \in \mathbb{R}, s \geq 0} \|u - \bar{u}\|_2^2 + \rho s \quad (3.28a)$$

$$\text{s.t. } \dot{V}_t(x, u) + \lambda V(x) \leq s, \quad (3.28b)$$

$$u \in U, \quad s \geq 0, \quad (3.28c)$$

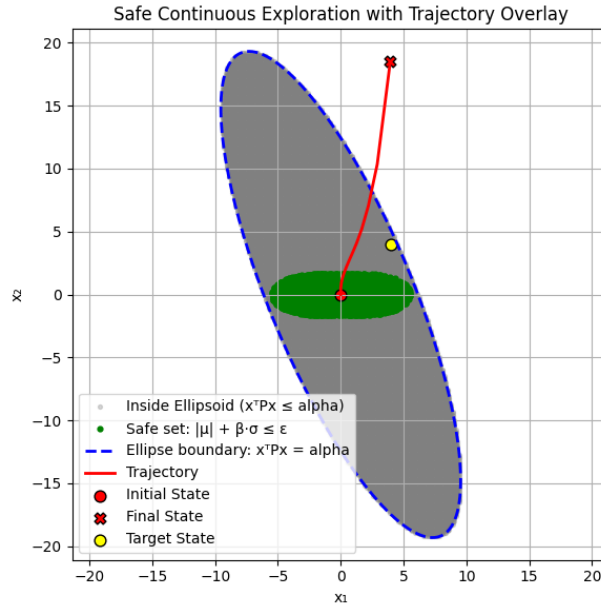


Figure 3.2: Unsafe Exploration due to a wrong target selected

where \bar{u} is an LQR reference, $\rho \gg 1$ penalises slack, $\lambda > 0$ sets decay rate, and

$$\dot{V}_t(x, u) = x(AP + PA)x + 2xPBu + 2xP\mu_t(x) + 2\beta_t(\delta) \|Px\|_2 \sigma_t(x). \quad (3.29)$$

3.6.2 Derivation of the Robustified Lyapunov Derivative

To enforce high-probability decrease of the Lyapunov function $V(x) = x^\top Px$, we need an upper bound on its time derivative that accounts for both the nominal linear dynamics and the learned residual with uncertainty.

1. Nominal part. Under the true dynamics $\dot{x} = Ax + Bu + \Delta(x)$, the exact derivative is

$$\dot{V}(x, u) = x^\top (A^\top P + PA)x + 2x^\top PBu + 2x^\top P\Delta(x).$$

If $\Delta \equiv 0$, then choosing $u = -Kx$ with $K = R^{-1}B^\top P$ guarantees $\dot{V}(x, -Kx) = -x^\top Qx < 0$.

2. GP mean correction. We model $\Delta(x)$ with a Gaussian process, whose posterior mean $\mu_t(x)$ approximates the residual. Replacing $\Delta(x)$ by $\mu_t(x)$

gives the surrogate derivative

$$\dot{V}_{\text{mean}}(x, u) = x^\top (A^\top P + PA)x + 2x^\top PB u + 2x^\top P \mu_t(x).$$

3. Uncertainty buffer. By the GP-UCB bound (Eq. (2.7)), $|\Delta(x) - \mu_t(x)| \leq \beta_t \sigma_t(x)$ with probability $1 - \delta$. Therefore

$$|x^\top P[\Delta(x) - \mu_t(x)]| \leq \|Px\|_2 |\Delta(x) - \mu_t(x)| \leq \beta_t \|Px\|_2 \sigma_t(x).$$

To be conservative, we add twice this bound to \dot{V} (since the term appears as $2x^\top P\Delta$):

$$\underbrace{2x^\top P\Delta(x)}_{\text{unknown}} \leq 2x^\top P\mu_t(x) + 2\beta_t \|Px\|_2 \sigma_t(x).$$

4. Combined worst-case derivative. Collecting the nominal, mean, and uncertainty terms yields

$$\dot{V}_t(x, u) = x^\top (A^\top P + PA)x + 2x^\top PB u + 2x^\top P \mu_t(x) + 2\beta_t(\delta) \|Px\|_2 \sigma_t(x). \quad (3.30)$$

5. CLF-QP constraint. To enforce an exponential decrease in expectation (up to slack s), we impose in the quadratic program

$$\dot{V}_t(x, u) + \lambda V(x) \leq s, \quad s \geq 0,$$

where $\lambda > 0$ sets the desired decay rate and s is heavily penalized in the objective. This guarantees that, with probability $1 - \delta$, the Lyapunov function decreases at rate λ except for a small residual s , yielding bounded-error stability.

3.7 Safe-UCB Exploration Algorithm

Combining the previous formulation, it is possible to iterate the procedure to achieve safe exploration. At each iteration we (i) update the GP to quantify model uncertainty, (ii) recompute the *currently certified safe region* using that uncertainty (PCIS/barrier certificate), and (iii) pick the *most informative state* x_t *within* that region (variance-seeking/UCB), applying a safety filter on the control input so the trajectory never exits the certified set. As data accumulate, uncertainty shrinks and the certified set expands.

Loop invariants (safety by construction). Let $S_{t,\text{safe}} \subseteq \mathcal{X}$ denote the certified safe set at time t . The procedure maintains:

- **I1 (initialization).** $x_0 \in S_{0,\text{safe}}$.
- **I2 (admissible query).** $x_t \in S_{t,\text{safe}}$ for all t .
- **I3 (closed-loop invariance).** The safety filter ensures the realized trajectory does not exit $S_{t,\text{safe}}$ between updates (probability $\geq 1 - \delta$).

Algorithm 1 GP-based Safe-UCB Exploration (no horizon optimization)

Inputs: risk level δ , iterations T , uncertainty cap σ_{\max} **State:** dataset $\mathcal{D}_t = \{(x_i, \Delta_i)\}_{i=1}^t$; GP posterior (μ_t, σ_t) **Init:** collect safe seed data \mathcal{D}_0 ; train GP; compute initial safe set $S_{0,\text{safe}}$ **for** $t = 0, 1, \dots, T - 1$ **do**—
 Refit (or update) GP hyperparameters by marginal likelihood Compute confidence scale $\beta_t(\delta)$ (UCB schedule) Update safe set (inner PCIS):

$$S_{t,\text{safe}} = \{x : \text{PCIS/CBF condition holds at } 1 - \delta, \sigma_t(x) \leq \sigma_{\max}\}$$

if $S_{t,\text{safe}} = \emptyset$ **then**

— **break** no certified region **Select query:** $x_t \in \arg \max_{x \in S_{t,\text{safe}}} \sigma_t(x)$ expand boundary first **Safety filter:** compute u_t from CBF/CLF-QP so that the barrier condition holds Apply (x_t, u_t) to the plant; observe residual Δ_t (training target) Augment data: $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{(x_t, \Delta_t)\}$ **return** final GP (μ_T, σ_T) and safe set $S_{T,\text{safe}}$

Remarks. (i) The constraint $\sigma_t(x) \leq \sigma_{\max}$ is a *prudence gate*: only states where the GP is not overly uncertain are trusted for certifying the barrier/PCIS; this reduces false safety due to underestimation. (ii) The acquisition can be replaced by UCB $\mu_t(x) + \beta_t \sigma_t(x)$ or batched variants; in our experiments, pure variance seeking inside $S_{t,\text{safe}}$ maximized safe-set growth. (iii) Complexity is dominated by GP updates (exact: $O(N^3)$); sparse/state-space GPs reduce this) and a small QP per step for the safety filter.

3.7.1 Parameter Guidelines

Parameter	Effect and tuning guideline
β_t (confidence)	Larger β_t widens the GP-UCB envelope and the safety margin in the barrier test: <i>safer but more conservative</i> . Start with a theoretically valid schedule (e.g., GP-UCB) and relax only after empirical coverage checks.
σ_{\max} (uncertainty cap)	Upper bound on admissible posterior std. inside the certificate. Smaller $\sigma_{\max} \Rightarrow$ tighter $S_{t,\text{safe}}$ but fewer false positives; increase gradually as calibration improves.
λ (CLF decay rate)	Sets how aggressively the CLF drives the state; larger λ yields faster convergence but stronger inputs and potential conflicts with the barrier near the boundary. Tune so that CLF is secondary to safety (small slack).
ρ (slack penalty)	Weight on CLF slack in the QP. Larger ρ discourages violations of the CLF decrease condition; set high enough that the barrier constraint is never compromised.

3.8 Case Studies and Modelling Details

This section spells out the two plants used throughout the thesis: (i) an illustrative 2D unstable nonlinear system with an exactly known residual, and (ii) a three-tank laboratory process modelled from first principles (Torricelli) plus a data-driven residual. In both cases we specify the nominal linearisation, safety constraints, the GP residual model, and the CLF-QP used within the PCIS.

3.8.1 Illustrative 2D Example

Nonlinear dynamics. Consider

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = -2x_2 + x_2^2 + u,$$

with state $x = [x_1, x_2]^\top \in \mathbb{R}^2$ and input $u \in \mathbb{R}$.

Nominal linearisation and residual. Around the origin $(0, 0)$,

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -2 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \Delta(x) = \begin{bmatrix} 0 \\ x_2^2 \end{bmatrix},$$

so

$$\dot{x} = Ax + Bu + \Delta(x). \quad (3.31)$$

GP model of the residual. We place a GP prior on $\Delta(\cdot)$ with kernel k_θ on \mathbb{R}^2 , $\Delta(\cdot) \sim \mathcal{G}\mathcal{D}(0, k_\theta)$, and obtain posterior mean/variance (μ_t, σ_t^2) from data $\mathcal{D}_t = \{(x_i, \Delta_i)\}$ via (3.3).

Safety constraints. Hard bounds $\mathcal{X} = \{|x_1| \leq 1, |x_2| \leq 1\}$, $\mathcal{U} = [-u_{\max}, u_{\max}]$, and an uncertainty budget $\sigma_t(x) \leq \sigma_{\max}$. A sufficient one-step condition for safety (with risk δ encoded by β_t) is

$$|[Ax + Bu + \mu_t(x)]_i| + \beta_t \sigma_t(x) \leq 1, \quad i = 1, 2.$$

LQR and CLF-QP. With $Q = I_2$, $R = 1$, the CARE $A^\top P + PA - PBR^{-1}B^\top P + Q = 0$ yields $P > 0$, $K = R^{-1}B^\top P$, and $V(x) = x^\top P x$. The GP-augmented CLF-QP (continuous-time surrogate) is

$$\begin{aligned} \min_{u, s} \quad & (u + Kx)^2 + \rho s \\ \text{s.t.} \quad & \dot{V}_t(x, u) + \lambda V(x) \leq s, \quad u \in [-u_{\max}, u_{\max}], \quad s \geq 0, \\ & \dot{V}_t(x, u) = x^\top (A^\top P + PA)x + 2x^\top P B u + 2x^\top P \mu_t(x) + 2\beta_t \|Px\|_2 \sigma_t(x). \end{aligned}$$

Safe-UCB exploration. At each iteration, choose $x_t = \arg \max_{x \in \mathcal{S}_t^\delta} \sigma_t(x)$ and compute u_t via the CLF-QP; then update \mathcal{D}_{t+1} .

Figure 3.3: 2D example: PCIS tube and closed-loop trajectories under the CLF-QP controller.

3.8.2 Three-Tank Nonlinear Process

The water tank system comprises three vertically arranged tanks. Tanks 1 and 2 discharge into the downstream collector Tank 3, which in turn drains to a reservoir. Each tank is equipped with an outlet valve whose opening $\gamma_i \in [0, 100]\%$ regulates the outflow; in our formulation these valve openings are

the control inputs $u = [\gamma_1, \gamma_2, \gamma_3]^T$. A single pump draws water from an upstream supply reservoir and injects it into Tank 1, creating an inflow q that we treat as a (measured) disturbance. The measured states are the water levels h_1, h_2, h_3 (m). This layout yields coupled, nonlinear dynamics: adjusting γ_1 or γ_2 affects Tank 3 through the inter-tank connections, while γ_3 governs the global drainage to the reservoir.



Figure 3.4: Three-tank system: valves $\gamma_1, \gamma_2, \gamma_3$ are control inputs; pump inflow q is a measured disturbance. States are levels h_1, h_2, h_3 ; constraints \mathcal{X}, \mathcal{U} are safety-critical.

Outflows through orifices grow with the square root of the liquid head (Torricelli's law), so small level changes can produce disproportionately large flow variations. Inter-tank connections couple the levels: raising h_1 or h_2 increases the inflow to tank 3, while tank 3's own outlet lowers all upstream levels indirectly. Real water tank further exhibits nonidealities, valve hysteresis, pump nonlinearities, and measurement biases, which we group as a residual term to be learned online.

Control goal and safety envelope. The control objective is to regulate h_1, h_2, h_3 about prescribed operating ranges while strictly respecting: (i) level bounds to avoid overflow or cavitation, (ii) input saturation and rate limits on

the pumps, and (iii) conservative margins when the model is uncertain. In this chapter we use a nominal linear model for set-point tracking and a GP residual to capture unmodelled effects, combining them in a Lyapunov-based safety filter and a probabilistic control-invariant set (PCIS) that certifies safe operation with high probability.

Setup

The laboratory water tank is a good safe exploration testbed for several reasons:

- (i) *Nonlinearity and topology*: square-root outflows and height-dependent cross-sections create operating-point dependent dynamics.
- (ii) *State coupling*: inter-tank flows make the system MIMO with nontrivial interactions between levels.
- (iii) *Hard constraints*: level and input limits are strict and safety-critical; violations are physically meaningful.
- (iv) *Learnable residuals*: repeatable but unknown effects (valve/pump characteristics) are well suited to GP residual learning.
- (v) *Certification*: the GP variance naturally feeds into PCIS-style conditions, yielding high-probability invariance guarantees.

We first summarise the physical model (flows and mass balances), then select operating points and derive linear models around them. Next, we introduce a multi-output GP residual model and embed its mean/variance in the CLF-QP and in the PCIS computation. We close with the exploration policy that selects informative set-points *inside* the certified safe region.

Variables and inputs. States are water levels $x = [h_1, h_2, h_3]^\top$ (m). Control inputs are the outlet valve openings $u = [\gamma_1, \gamma_2, \gamma_3]^\top$ (%). A single pump injects inflow q into Tank 1 and is treated as a measured disturbance $r = q$. The sampling time is $T_s = 0.1$ s with zero-order hold on u .

Control-oriented abstraction. We use a nominal physics-based predictor around an operating point, plus a data-driven residual:

$$x_{k+1} = A_d x_k + B_d u_k + E_d r_k + \mu_t(x_k, u_k), \quad (3.32)$$

where (A_d, B_d, E_d) are ZOH-discretized linearizations (Appendix A.2), and $\mu_t(\cdot)$ is a Gaussian process (GP) posterior mean that models unmodeled effects (valve/pump nonlinearities, geometry deviations, sensor bias). The GP also provides a state-dependent standard deviation $\sigma_t(x, u)$ used by the safety filter.

Operating points (summary). We consider three operating points (OPs) spanning the usable range (low/mid/high levels). For each OP we precompute continuous-time (A, B, E) , discretize them to (A_d, B_d, E_d) , and log their dominant time constants. Numerical values and the OP selection rationale are in Table A.1 (Appendix A.2).

Safety constraints. Physical limits:

$$\mathcal{X} = \{h_i^{\min} \leq h_i \leq h_i^{\max} \forall i\}, \quad \mathcal{U} = \{0 \leq u_i \leq 100 \forall i\}.$$

LQR baseline and CLF-QP. Discretise (A, B) with sample time T_s to (A_d, B_d) for the LQR design, or solve the continuous-time CARE to obtain $P > 0$ and $V(x) = x^\top P x$. The CLF-QP extends the 2D case to 3D:

$$\begin{aligned} \min_{u, s} \quad & \|u + Kx\|_2^2 + \rho s \\ \text{s.t.} \quad & \dot{V}_t(x, u) + \lambda V(x) \leq s, \quad u \in [0, 100]^2, \quad s \geq 0, \\ & \dot{V}_t(x, u) = x^\top (A^\top P + PA)x + 2x^\top P B u + 2x^\top P \mu_t(x, u) + 2\beta_t \|Px\|_2 \|\Sigma_t^{1/2}(x, u)\|_2. \end{aligned}$$

PCIS construction. Using $V(x)$ we compute the largest invariant ellipsoid inside the box constraints, $\mathcal{E}(\alpha_{\max}) = \{x : x^\top P x \leq \alpha_{\max}\}$, and then the δ -PCIS $\mathcal{S}_t^\delta \subseteq \mathcal{E}(\alpha_{\max})$ by propagating the GP uncertainty one step and iterating until convergence (as in 3.5.5).

Exploration policy. Within \mathcal{S}_t^δ , we select set-points to maximise model learning subject to safety, e.g., $x_t^* = \arg \max_{x \in \mathcal{S}_t^\delta} \text{tr} \Sigma_t(x, \hat{u}(x))$ or a simpler $\max_i \sigma_{t,i}$ rule, and compute a certified input with the CLF-QP.

Model use in control and safety. Equation (3.32) is the predictor inside: (i) the CLF-QP controller (Sec. 3.6.1) via the GP mean μ_t and a confidence term $\beta_t \sigma_t$, and (ii) the probabilistic control invariant set (PCIS) computation (Sec. 3.2) to certify that trajectories remain inside \mathcal{X} with probability $\geq 1 - \delta$. Only the compact form (3.32) is required in the main text; derivations are in the appendix.

Concluding Remarks on the Methodology

The methodology adopted in this thesis combines quantitative and qualitative analyses to provide a comprehensive evaluation of safe exploration strategies for nonlinear systems.

From a quantitative perspective, the evaluation is based on the following metrics:

- **Stability:** decrease of a candidate Control Lyapunov Function (CLF) $V(x)$ and verification of exponential decay conditions $\dot{V}(x) \leq -\lambda V(x)$;
- **Safety:** probability of state constraint satisfaction $\Pr(x_k \in \mathcal{S}) \geq 1 - \delta$, number of hard constraint violations, and evolution of the Probabilistic Control Invariant Set (PCIS) size across iterations;
- **Learning performance:** root mean square error (RMSE) of the Gaussian Process predictions with respect to the true residual dynamics, reduction of predictive variance $\sigma^2(x, u)$ over time, and evolution of confidence bounds $(\mu(x, u) \pm \beta\sigma(x, u))$;
- **Computational tractability:** solver time per control step and scalability to different sampling rates.

From a qualitative perspective, the analysis complements the numerical results with interpretative insights, focusing on:

- **Scalability and robustness:** assessment of the controller's behavior under unmodeled disturbances, sensor noise, and varying operating conditions;
- **Exploration behavior:** observation of how the PCIS expands as the GP model improves, allowing progressively more aggressive exploration while retaining safety;
- **Practical implications:** applicability of the framework to real-world systems such as the water tank benchmark and the inverted pendulum, highlighting trade-offs between safety guarantees and exploration efficiency.

By integrating both perspectives, the methodology bridges the gap between theoretical rigor and engineering relevance. The *quantitative metrics* ensure formal guarantees of stability, safety, and learning efficiency, while the

qualitative insights contextualize these guarantees within practical scenarios. Together, they provide a balanced foundation for the subsequent presentation of results and discussion.

3.9 Simulation Environment

This section details the software and modeling stack used to simulate safe exploration on nonlinear plants, with a focus on the three-tank water system compiled as a Functional Mock-up Unit (FMU). We explain plant modeling, GP residual learning, probabilistic safety certification (PCIS), target selection, and the closed-loop orchestration. Implementation specifics, full repository tree, parameter tables, and code snippets are provided in Appendix B (Appendix: Simulation Environment).

1) Plant model and FMU integration. The environment centers on a three-tank process with states H_1, H_2, H_3 (m) and inputs (valve openings V_1, V_2, V_3 and pump flow). The FMU is loaded at runtime together with a Simulink harness; the nominal sampling time is $T_s = 1.0$ s, and steady-state levels are $H_{i,ss} = 0.219$ m (all i). The documentation also reports the linearized model around steady state and provides utilities to load the FMU and retrieve variable references for efficient I/O.

2) Discretization and linear baseline. We represent the plant as

$$\dot{x}(t) = f(x(t), u(t)), \quad x \in \mathbb{R}^n, u \in \mathbb{R}^m,$$

and use a linear baseline around x_{ss} ,

$$\dot{x}(t) = A(x(t) - x_{ss}) + Bu(t), \quad A = \begin{bmatrix} 4.483 & 3.184 & 0.939 \\ 3.184 & 5.576 & 1.352 \\ 0.939 & 1.352 & 3.712 \end{bmatrix},$$

then discretize with forward Euler for simulation: $x_{k+1} = x_k + T_s f(x_k, u_k)$. This baseline separates known linear effects from the learned residuals (next item).

3) Residual learning with Gaussian Processes (multi-output). For each state component i , the model learns the residual $\Delta_i(x, u) = f_i(x, u) - (A(x - x_{ss}) + Bu)_i$ using an independent GP with mean μ_i (typically zero) and covariance κ_i (default: RBF),

$$\Delta_i \sim \mathcal{G}\mathcal{P}(\mu_i, \kappa_i), \quad \kappa_i((x, u), (x', u')) = \sigma_f^2 \exp\left(-\frac{1}{2\ell^2} \|(x, u) - (x', u')\|^2\right),$$

yielding posterior mean/variance $\mu_i^*(x, u)$, $\sigma_i^{2*}(x, u)$. Multi-output training instantiates one GP per residual component.

4) Sparse GP for scalability. For large datasets we employ Sparse GP Regression (SGPR) with M inducing points Z (K-means initialization), optimized via Adam; the framework exposes a helper `train_sparse_gp` to build and fit the SGPR in `gpflow`. This reduces training to $O(NM^2)$ and posterior updates to $O(M^3)$, enabling frequent re-training inside the exploration loop.

5) Safety certification (PCIS). Safety is checked pointwise via a chance-constraint of the form

$$|\mu_i^*(x, u)| + \beta \sigma_i^*(x, u) \leq \varepsilon,$$

over an ellipsoidal candidate region $\{x : x^\top P x \leq \alpha\}$ where $P > 0$ is the (continuous-time) Riccati solution. Confidence $\beta = \sqrt{2 \log(1/\delta)}$ and threshold ε are set from design constants and Lyapunov scalings; the algorithm samples candidate states, filters the ellipsoid, queries GP posteriors, and returns the discrete safe set S_{safe} . This implements a PCIS-style safety screen consistent with probabilistic invariance analyses for GP (cf. Griffioen et al., 2023).

6) Target selection (Safe-UCB). Exploration targets are chosen by maximizing an optimistic score under safety:

$$\text{UCB}(x) = \mu^*(x) + \beta \sigma^*(x), \quad x^* = \arg \max_{x \in S_{\text{safe}}} \text{UCB}(x),$$

computed over the certified S_{safe} . A reference implementation is provided (`select_target_uchb`). This mirrors bandit-style optimism but constrained to safe candidates.

7) Closed-loop controller and rollout. Given a target x^* , the environment calls an LQR tracking routine against the linear baseline while the FMU provides the ground-truth rollout. Trajectory tuples $\{x_k, u_k, x_{k+1}\}$ are logged; residual labels Δ_i are recomputed and appended, and the GP bank is re-trained. This structure supports both pure LQR tracking and GP-MPC variants; a sampling-based GP-MPC formulation with high-probability constraint satisfaction (Prajapat et al., 2025) is compatible with the same simulator interface.

8) Orchestration: the safe exploration loop. The main loop proceeds as: (i) compute S_{safe} with current GPs; (ii) select x^* via Safe-UCB; (iii) run tracking

Table 3.1: Key simulation parameters (defaults and recommended ranges).

Parameter	Default	Typical range
Sampling time T_s [s]	1.0	0.1–2.0
Confidence δ	0.02	10^{-3} –0.1
Ellipsoid scale α_m	0.2	0.05–1.0
Lyapunov rate λ	1.0	0.5–2.0
Safety margin κ	0.8	0.6–1.2
Inducing points M	30	16–128
Iterations / steps	10/1000	task-dependent

in the FMU; (iv) augment data and re-train GPs; (v) iterate. The reference implementation (`safe_exploration_loop`) exposes these steps and returns trajectories, final models, and statistics for analysis. SAGEMPC (Prajapat et al., 2025) fits into the same pattern by changing step (ii) to a goal-directed, receding-horizon optimizer that preserves safety with finite-time exploration guarantees.

9) Software structure and artifacts. Core modules include `gp_model.py` (GP/SGPR), `pcis.py` (safe set computation), `safe_ucb.py` (targeting), `simulation.py` (FMU rollouts), `controller.py` (tracking), and plotting/validation utilities. The repository contains data and model folders (FMU, Simulink harness), plus notebooks for end-to-end runs. Installation and basic usage are documented for reproducibility.

10) Configuration, logging, and metrics. Default configuration (confidence δ , ellipsoid scale α_m , Lyapunov rate λ , margin κ , inducing count M , number of iterations and steps) is exposed as parameters. The environment logs: GP metrics (RMSE, R^2 , NLPD, coverage), safety statistics (PCIS size, violation counts), and control KPIs (tracking error, settling time). Plots summarize safe-set growth, uncertainty reduction, and exploration paths.

11) Numerical considerations. To avoid optimistic certification, we calibrate GP variances on a validation split before using σ^* in safety tests; we also cap β and enforce a minimum noise variance to prevent degeneracy under repeated re-training. When adopting GP-MPC, sampling-based uncertainty propagation within SQP alleviates independence assumptions across time and improves reachable-set fidelity (Prajapat et al., 2025).

12) Cross-references to the Appendix. Appendix B contains: (i) the full repository tree and module map; (ii) complete parameter tables with

recommended ranges; (iii) code listings for SGPR training, PCIS computation, Safe-UCB, and the main loop; (iv) FMU variable maps and initialization; (v) additional figures (safe-set snapshots, uncertainty evolution).

Chapter 4

Results

This chapter presents the experimental validation of the proposed safe exploration framework. The objective is to assess whether Gaussian Process (GP) regression, combined with Probabilistic Control Invariant Sets (PCIS) enable informative data collection while maintaining system safety. To this end, two benchmarks of increasing complexity are considered: a synthetic *Polynomial system* and the nonlinear *Three-Tank process*.

For each case study, the analysis follows a consistent structure:

1. **Gaussian Process modeling:** evaluation of the GP regression in capturing residual dynamics and quantifying uncertainty.
2. **Probabilistic Control Invariant Set:** construction of safe regions under uncertainty and their evolution as learning progresses.
3. **Unsafe exploration:** comparison with unconstrained data collection strategies, highlighting the risk of constraint violations.
4. **GP-PCIS safe exploration:** demonstration of the proposed method, showing how safety guarantees and active learning can be reconciled.

This structured presentation allows a transparent comparison across different methods and provides quantitative evidence of the trade-offs between safety, exploration efficiency, and control performance. The polynomial benchmark serves as a controlled environment to validate the theoretical properties of the approach, while the three-tank process illustrates its applicability to a realistic nonlinear system with practical relevance in process control.

4.1 Polynomial Benchmark

This benchmark provides a low-dimensional and analytically tractable setting to test whether GP-based residual learning combined with probabilistic control invariant sets (PCIS) can enable informative exploration under explicit safety guarantees. The system features a known quadratic residual in the velocity dynamics, which makes it ideal to (i) stress-test kernel misspecification and uncertainty calibration, (ii) quantify the conservatism of the PCIS certificate, and (iii) compare exploration policies at equal risk levels.

4.1.1 Gaussian Process Kernel Comparison

Table 4.1: Kernel performance for residual modeling. Coverage is the empirical fraction of test points inside $\mu \pm 1.96\sigma$; $\bar{\sigma}$ is the mean predictive standard deviation. Best values are in **bold**.

Kernel	RMSE	R^2	Log-lik	Coverage	$\bar{\sigma}$
RBF	0.2275	0.9124	571.36	34.0%	0.0428
Matérn 3/2	0.8433	-0.2041	550.58	27.0%	0.0869
Matérn 5/2	0.3754	0.7615	566.04	74.5%	0.1165
RBF+Matérn 3/2	0.2275	0.9123	571.36	34.0%	0.0428
Periodic+RBF	0.2277	0.9122	571.36	34.0%	0.0428
Linear+RBF	0.2275	0.9124	571.36	34.0%	0.0428
RBF (ARD)	0.2274	0.9125	571.36	34.0%	0.0428
Polynomial (deg. 2)	0.0000	1.0000	580.76	100.0%	0.0038

For visualization we report the slice $x_1 = 0$ with $x_2 \in [-5, 5]$. For each kernel we evaluate root mean squared error (RMSE), coefficient of determination R^2 , log-marginal likelihood (LL), empirical coverage of the nominal 95% interval $\mu \pm 1.96\sigma$, and mean predictive standard deviation $\bar{\sigma}$. Figure 4.1 shows posterior means with $\pm 2\sigma$ bands; Table 4.1 reports the metrics. The results display a clear separation between models with correct inductive bias and stationary kernels. A polynomial kernel of degree 2 exactly matches the known quadratic residual and attains $\text{RMSE} = 0$, $R^2 = 1$, best LL, and 100% empirical coverage, serving as an oracle baseline. Among stationary kernels, SE/RBF and its hybrids (including ARD, linear+RBF, periodic+RBF) achieve very similar mean accuracy ($\text{RMSE} \approx 0.227$, $R^2 \approx 0.912$) and LL, yet their predictive intervals are dramatically over-confident: coverage concentrates around 34% while $\bar{\sigma} \approx 0.043$. Matérn-5/2 partially

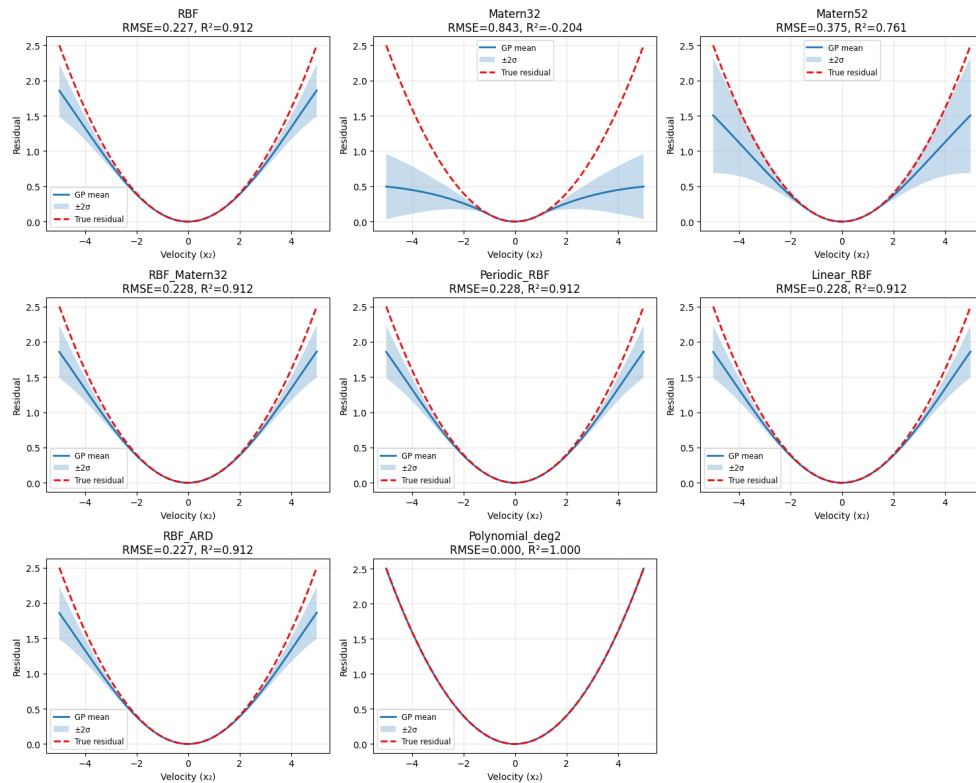


Figure 4.1: Kernel comparison for GP residual modeling on the 2D system. Each panel shows the GP mean (blue) and $\pm 2\sigma$ band against the true quadratic residual (red, $0.1x_2^2$) along the slice $x_1 = 0$. Metrics are computed on a held-out set.

corrects this over-confidence (coverage 74.5%) at the expense of a higher error (RMSE = 0.375); Matérn-3/2 is clearly misspecified in this setting ($R^2 < 0$). Within the stationary family, RBF-ARD is the most accurate non-oracle choice, reflecting the anisotropic sensitivity (stronger dependence on x_2 than on x_1). Training and prediction costs are broadly comparable across kernels (see Table 4.2); the polynomial model is marginally faster in our runs, while hybrid models incur a small overhead due to extra hyperparameters, but these differences are secondary relative to the accuracy and calibration gaps. The under-coverage observed with RBF-type models has three complementary causes. First, maximum-likelihood fitting with a well-aligned mean term tends to shrink the noise and signal variance, leading to overly small $\sigma(x)$. Second, stationary kernels extrapolate to a constant away from data, whereas the true residual grows quadratically in $|x_2|$; near the edges of the slice the model becomes overconfident unless the learned lengthscales are extremely

Table 4.2: Compute cost and model complexity. Times in seconds; memory in MB; N_θ = number of kernel hyperparameters.

Kernel	Train	Predict	Total	Peak Mem.	N_θ
RBF	0.825	0.072	0.896	652.000	2.000
Matérn 3/2	0.877	0.057	0.934	657.000	2.000
Matérn 5/2	0.799	0.060	0.860	664.000	2.000
RBF+Matérn 3/2	1.002	0.087	1.090	675.000	4.000
Periodic+RBF	1.142	0.097	1.240	688.000	5.000
Linear+RBF	0.864	0.072	0.936	695.000	3.000
RBF-ARD	0.690	0.058	0.748	703.000	2.000
Polynomial (deg. 2)	0.634	0.054	0.688	705	2.000

short, which would in turn hurt smoothness and interpolation. Third, a homoscedastic Gaussian likelihood cannot capture a variance that increases with $|x_2|$, so uncertainty is understated precisely where it matters most. These calibration errors matter for safety: we certify state-wise feasibility via the predicate

$$|\mu(x)| + \beta \sigma(x) \leq \varepsilon, \quad (4.1)$$

and define the safe set $\mathcal{S} = \{x : (4.1) \text{ holds}\}$. If $\sigma(x)$ is underestimated, \mathcal{S} becomes spuriously large, inflating the certified PCIS and increasing the risk of constraint violations. A simple and effective countermeasure is to apply a global variance calibration on a validation split: find the smallest $\gamma^* \geq 1$ such that the empirical 95% coverage of $\mu \pm 1.96\sqrt{\gamma}\sigma$ reaches the nominal level and then use $\sigma_{\text{cal}}(x) = \sqrt{\gamma^*}\sigma(x)$. Formally,

$$\gamma^* = \min \left\{ \gamma \geq 1 : \frac{1}{|\mathcal{D}_{\text{val}}|} \sum_{(x,y) \in \mathcal{D}_{\text{val}}} \mathbf{1}(|y - \mu(x)| \leq 1.96 \sqrt{\gamma} \sigma(x)) \geq 0.95 \right\}, \quad (4.2)$$

which preserves the mean μ while restoring nominal coverage and yielding safer certificates through (4.1). The magnitudes implied by the measured coverages quantify the issue: if the miscalibration were purely multiplicative, then the RBF family's 34% two-sided coverage corresponds to a z-score of $\Phi^{-1}(0.67) \approx 0.440$, so one would need to widen intervals by a factor $c \approx 1.96/0.440 \approx 4.45$ (i.e., $\gamma^* \approx 19.8$) to reach 95%; for Matérn-5/2, 74.5% coverage gives $\Phi^{-1}(0.8725) \approx 1.138$ and $c \approx 1.72$ ($\gamma^* \approx 2.96$). In practice γ^* is obtained directly from the validation split via (4.2), but these back-of-the-envelope numbers explain why uncalibrated stationary models

can be unsafe by construction. Overall, the analysis suggests the following interpretation and guidance for deployment. When the residual structure is known or approximately polynomial, incorporating that inductive bias pays off decisively; otherwise, RBF-ARD remains the best stationary option in terms of mean fit, provided that its variance is calibrated before use in safety predicates. A robust compromise is to encode the quadratic trend in the *mean* (or via a polynomial+SE hybrid) and reserve the stationary kernel for residual fluctuations; this retains smooth generalization while avoiding tail overconfidence. Finally, calibration in (4.2) is intentionally marginal, near hard constraints one may still increase β or re-calibrate online, and real plants may deviate from pure quadratic behavior (e.g., friction, saturations, unmodeled couplings), in which case heteroscedastic likelihoods or heavier-tailed noise models become natural extensions.

4.1.2 Full GP versus Sparse GP

We compare exact Gaussian Processes (Full GP) against Sparse GP Regression (SGPR) on the residual dynamics of the polynomial system. To stress the behaviour of exact GP vs. Sparse GP (FITC-SGPR) in a low-data, low-dimensional setting, we replicated the comparison on the synthetic 2D system used in Sec. 4.1.1. Results mirror the polynomial benchmark trends: the Full GP attains the best accuracy (RMSE = 0.4128, $R^2 = 0.6719$) with empirical coverage = 90% at the nominal 95% level; Sparse GP with $M=10$ preserves similar coverage but degrades accuracy (RMSE = 0.6671, $R^2 = 0.1428$), while larger M values without hyperparameter re-estimation lead to pronounced miscalibration and poor fits ($R^2 < 0$, coverage = 5% ~ 15%). From a safety standpoint, these under-covering posteriors would shrink the certified safe set defined by the predicate $|\mu(x)| + \beta\sigma(x) \leq \varepsilon$ and jeopardise chance constraints. The run-time profile is favourable to SGPR (fastest run at $M=20$ took ≈ 7 ms), yet the best speed/accuracy trade-off in this test occurs at $M=10$ (Table 4.3), still requiring either variance calibration or a larger β to approach the 95% target before PCIS construction.

Table 4.3: Full GP vs. FITC–SGPR on the 2D car system (held-out test set). Coverage computed from nominal 95% credible intervals.

Model	RMSE	R^2	Coverage (%)	Time (s)	M
Full GP	0.412 8	0.671 9	90.0	0.27	80
Sparse GP (M=10)	0.667 1	0.142 8	90.0	0.01	10
Sparse GP (M=20)	3.230 7	−19.101 4	5.0	0.01	20
Sparse GP (M=30)	2.352 9	−9.662 5	15.0	0.01	30
Sparse GP (M=40)	1.813 6	−5.334 9	10.0	0.01	40

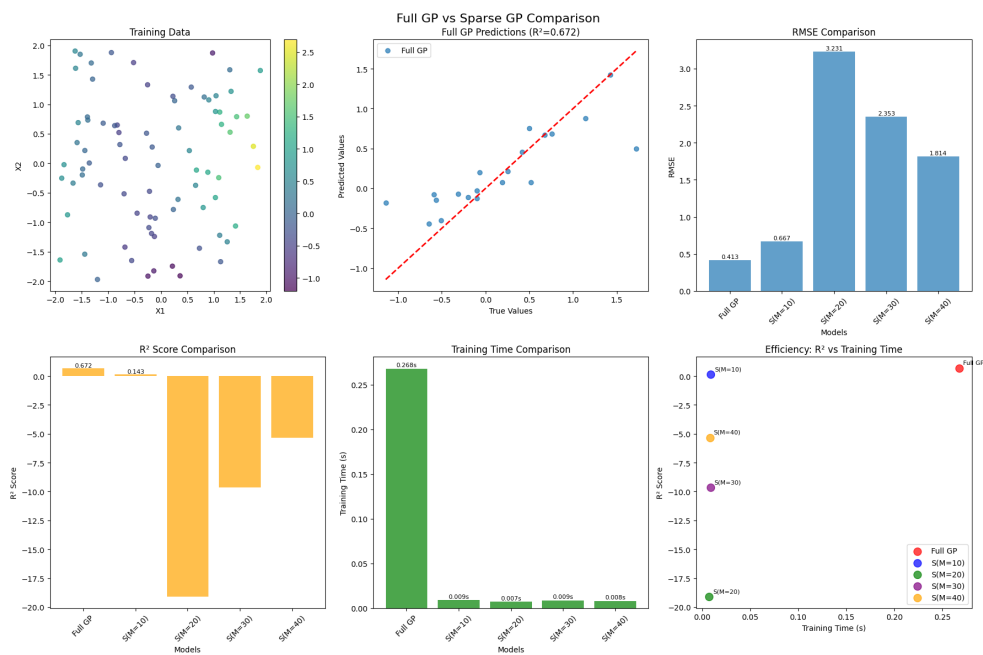


Figure 4.3: Comparison between exact GP and Sparse GP (FITC) on the 2D car system. Top: training data, full-GP test predictions, and RMSE across models. Bottom: R^2 , training time, and efficiency (R^2 vs. time). Sparse approximations achieve 30–40× speed-ups but suffer strong degradation in R^2 and coverage for $M > 10$.

Accuracy vs. calibration: Full GP dominates in RMSE/ R^2 ; SGPR with small M can retain coverage but at a notable accuracy cost; larger M without re-tuning induces severe under-coverage. **Safety:** Posteriors with coverage $< 95\%$ systematically violate the desired chance level for $\Pr(x_k \in \mathcal{S}) \geq 1 - \delta$; either increase β , recalibrate σ (e.g., via isotonic/temperature scaling on residuals), or re-optimize kernel hyperparameters and inducing locations (e.g.,

k-means/farthest-point). **Efficiency:** SGPR yields 30×–38× speedups in this test; the best observed trade-off is $M=10$, but it still under-covers w.r.t. the 95% target, so PCIS should be constructed conservatively (larger β or tightened ε).

4.1.3 Probabilistic Control Invariant Set

We quantify and certify a *high-probability* safe region \mathcal{S} for the learned dynamics so that closed-loop trajectories remain in \mathcal{S} with probability at least $1 - \delta$ for all time. Figure 4.4 compares the certified PCIS obtained with an exact GP and with sparse GPs of increasing inducing set size M on the 2D toy system. The certified area grows as model uncertainty shrinks and calibration improves: from 1.327 (2.4% of the Lyapunov ellipsoid) for the Full GP to 9.437 (16.9%) for SGPR with $M=40$. Very small M yields conservative and irregular sets; increasing M regularizes the boundary and expands \mathcal{S} , consistent with the posterior variance contraction that drives PCIS growth.

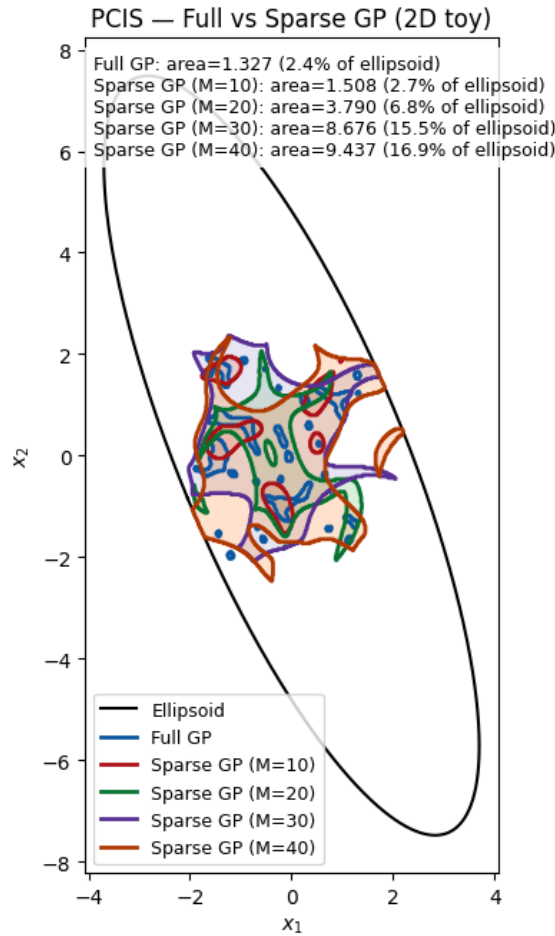


Figure 4.4: PCIS—Full GP vs. Sparse GP (2D toy). The black ellipse is the Lyapunov sublevel set. Colored contours show certified PCIS for the exact GP and SGPR with $M \in \{10, 20, 30, 40\}$. Legend reports areas and fraction of the ellipse captured.

Overall, $\text{vol}(\mathcal{S})$ increases with informative data and better posterior calibration, and trajectories initialised inside \mathcal{S} show no violations at the chosen risk level δ .

4.1.4 Unsafe exploration

As a baseline, we command the nominal LQR (designed on (A, B)) to drive the state to a target x_{tar} without any safety layer. Because x_{tar} lies outside the currently certified PCIS, the closed loop exits the invariant ellipsoid $\mathcal{S}(\alpha) = \{x : x^\top P x \leq \alpha\}$ and violates the GP-UCB safety predicate $|\mu(x)| + \beta\sigma(x) \leq$

ε . The red trajectory in Fig. 4.5 crosses the dashed boundary $x^\top P x = \alpha$ on its way to the target (yellow), illustrating that even a stabilising LQR can be *unsafe* under model mismatch when targets sit beyond the certified region. This motivates the need for the safety filter used in our method.

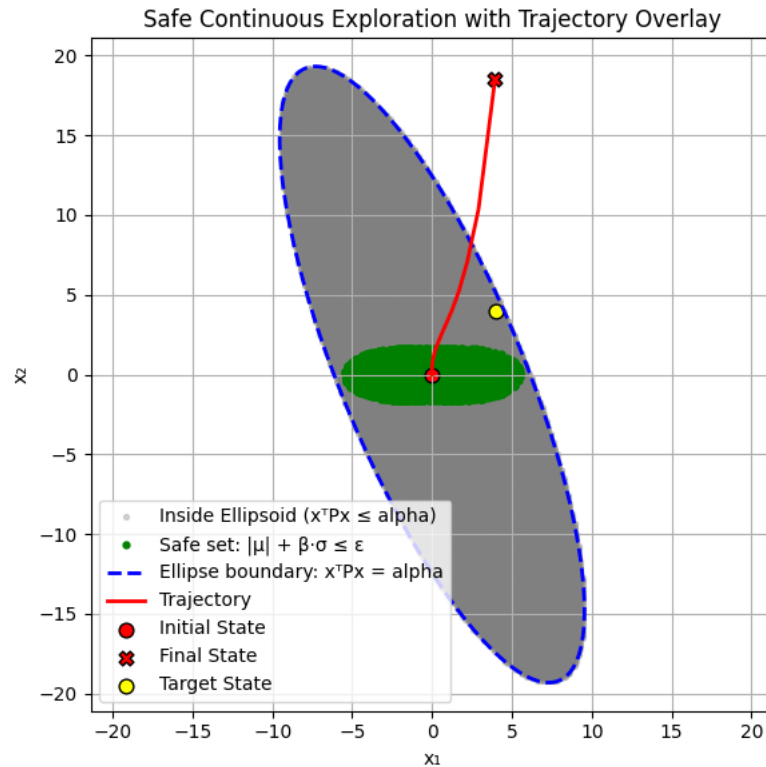


Figure 4.5: LQR tracking to a target *without* a safety layer. The green region shows the state-wise GP-UCB safe slice $\{x : |\mu(x)| + \beta\sigma(x) \leq \varepsilon\}$ inside the certified ellipsoid $S(\alpha)$ (grey, dashed boundary). Starting from the red dot, the LQR trajectory (red) exits $S(\alpha)$ while moving toward the target (yellow), demonstrating unsafe exploration in the absence of PCIS/CLF-QP filtering.

4.1.5 GP-PCIS safe exploration

The GP-PCIS framework maintains **zero violations** while exploring. The trajectory remains inside the PCIS $\{x : x^\top P x \leq \alpha\}$ and the GP-safe region $\{x : |\mu(x)| + \beta\sigma(x) \leq \varepsilon\}$. As $\sigma(\cdot)$ shrinks through learning, the level α increases and the PCIS expands, permitting progressively wider excursions without sacrificing guarantees (Fig. 4.6).

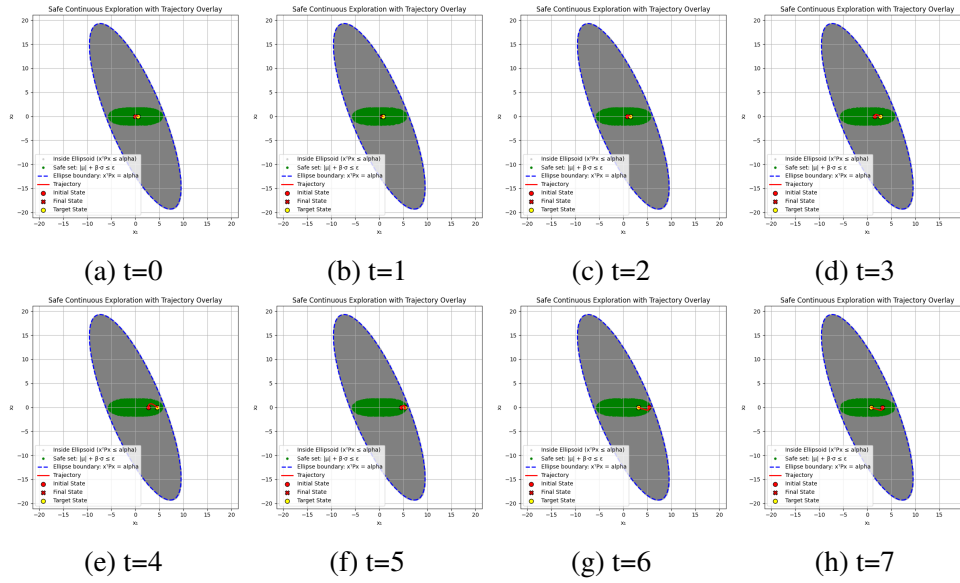


Figure 4.6: Safe exploration on the 2D system. The dashed blue ellipse is the PCIS boundary $x^T P x = \alpha$; the green region contains states satisfying the GP safety predicate $|\mu(x)| + \beta \sigma(x) \leq \varepsilon$. The trajectory (red) stays inside the safe set while moving from the initial (red dot) to the target (yellow ring). Controller: LQR baseline with safety filter; GP trained online on residuals.

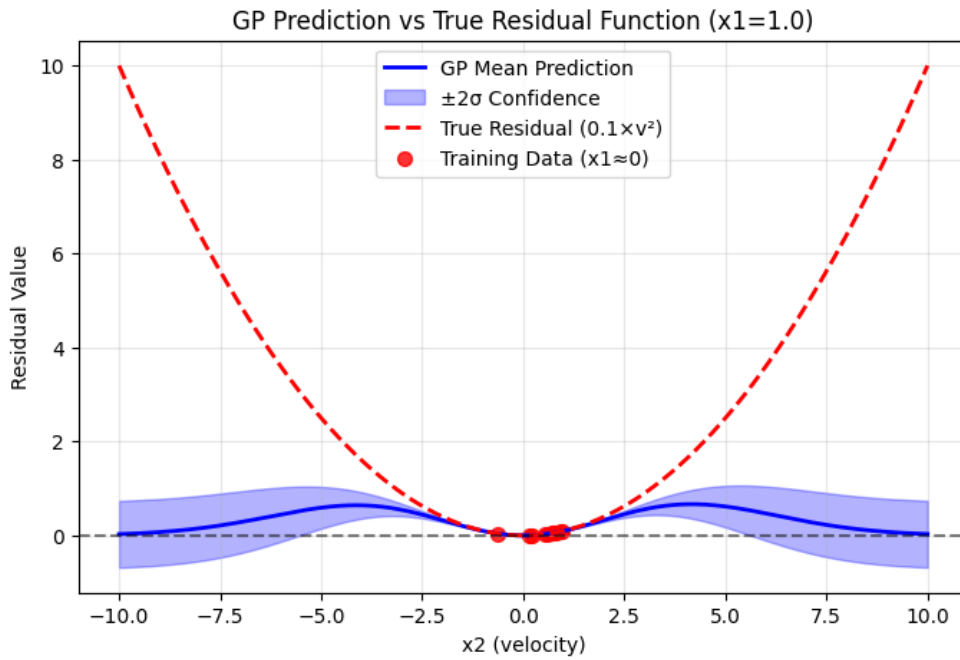


Figure 4.7: Gaussian Process fitted on the initial dataset

Initial fit (Fig. 4.7). With a small initial dataset the GP posterior reflects two characteristic features of data-scarce regimes: (i) *bias* of the mean $\mu(x)$ near the operating boundaries, and (ii) *wide confidence bands* $\pm\beta\sigma(x)$ in scarcely visited regions. Consequently, the safety predicate

$$|\mu(x)| + \beta\sigma(x) \leq \varepsilon$$

is satisfied only on a small subset of the state space, yielding a *small initial PCIS* and limiting the admissible exploration.

After safe exploration (Fig. 4.8). Closed-loop data collection improves both *accuracy* and *calibration*. Systematic residuals visible at initialization flatten, reducing the regression error (see RMSE in Fig. 4.9). Simultaneously, the predictive standard deviation $\sigma(x)$ contracts along the visited directions, which (through the predicate above) directly *enlarges* the region considered safe for subsequent exploration (PCIS expansion).

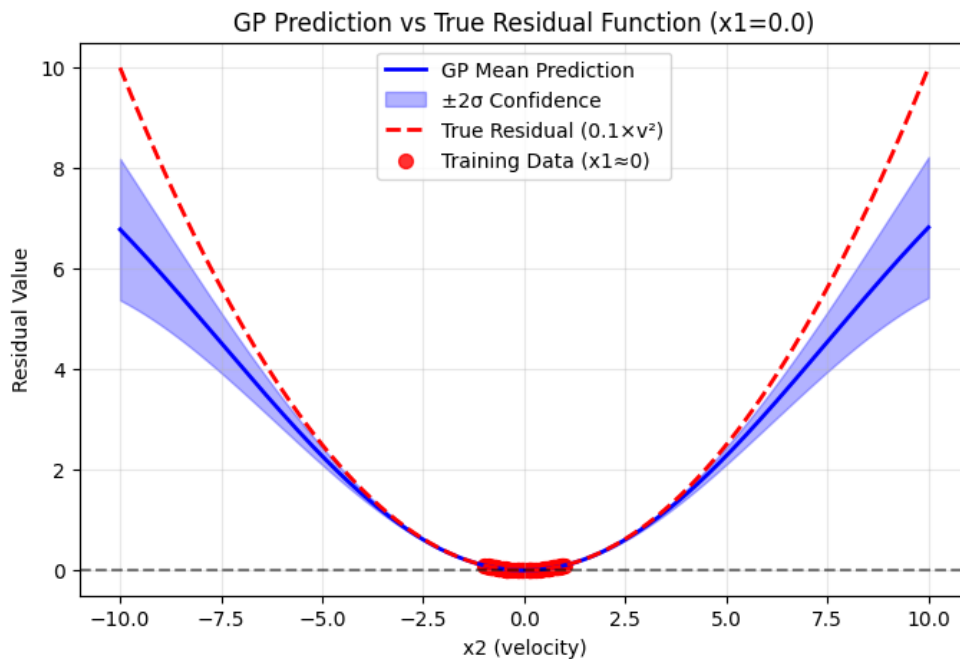


Figure 4.8: Gaussian Process fitted on the final dataset after Safe Exploration

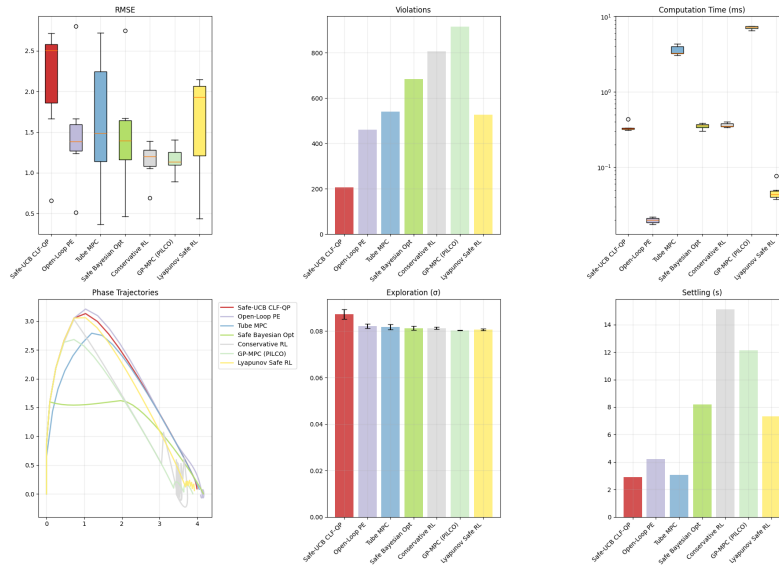


Figure 4.9: Comparison of GP covariance trace and RMSE across controllers.

4.2 Three-Tank Process

This benchmark evaluates the proposed approach in a realistic, nonlinear, multi-input multi-output (MIMO) plant with hard safety limits and strong cross-couplings. The three-tank system exhibits integrator-like level dynamics, actuator saturation, and transport/valve nonlinearities—conditions that stress test (i) the tractability of PCIS certification with GP, (ii) the effectiveness of a minimal-intervention safety filter during information-seeking moves, and (iii) robustness to model-plant mismatch (we use a linear nominal model plus a GP residual).

4.2.1 Gaussian Process

Table 4.4: Kernel comparison on the water-tank dataset (random split). For each residual channel (*res1*–*res3*), columns report RMSE ($\times 10^{-4}$) and empirical coverage (%) of nominal 95% predictive intervals on the test set. The best RMSE in each column is shown in **bold**. Abbrev.: RBF–ARD = RBF with automatic relevance determination; *Poly deg. 2* = degree-2 polynomial.

Kernel	res1		res2		res3	
	RMSE ($\times 10^{-4}$)	Cov. (%)	RMSE ($\times 10^{-4}$)	Cov. (%)	RMSE ($\times 10^{-4}$)	Cov. (%)
RBF	0.52	99.6	0.77	54.3	0.43	97.1
Matern32	0.41	99.9	0.40	99.9	0.25	100.0
Matern52	0.44	99.5	0.38	93.6	0.21	100.0
RBF+Matern32	0.44	98.4	0.56	86.9	0.45	95.6
Periodic+RBF	0.40	98.9	0.55	73.2	0.30	100.0
Linear+RBF	0.65	95.1	1.04	86.9	0.91	64.4
RBF-ARD	0.60	97.6	0.71	74.7	0.36	99.9
Poly deg. 2	2.95	94.8	0.86	81.5	0.62	88.1

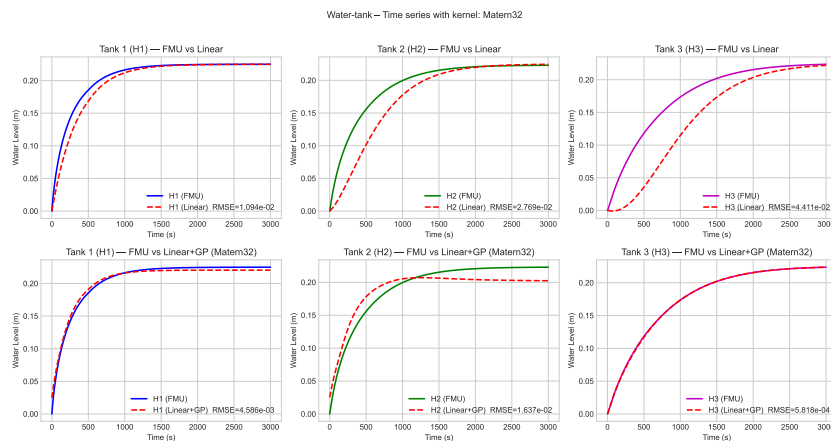


Figure 4.10: Time-series of the three-tank process: FMU data versus Linear model and Linear+GP with Matern-3/2 kernel. The GP correction significantly reduces the RMSE across all tanks.

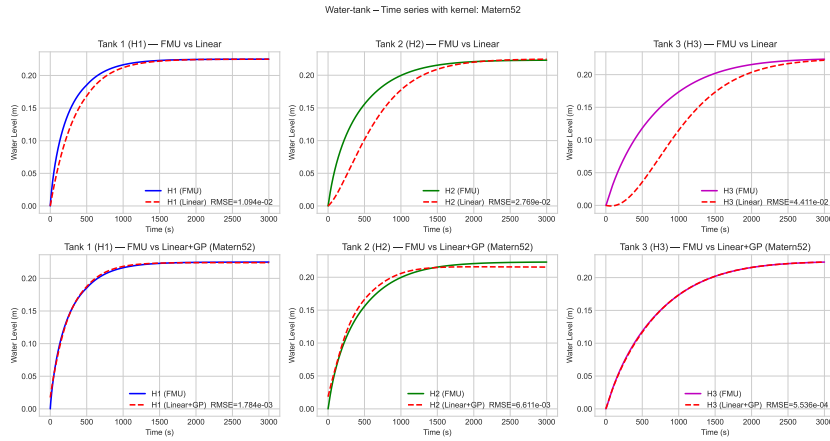


Figure 4.11: Time-series of the three-tank process: FMU data versus Linear model and Linear+GP with Matern-5/2 kernel.

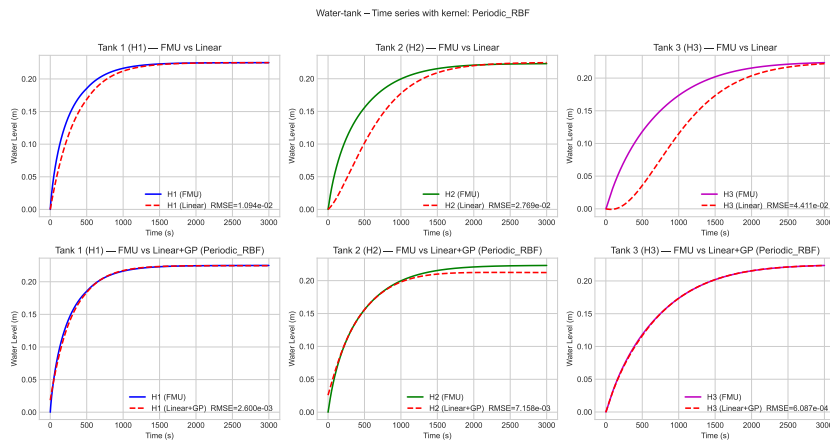


Figure 4.12: Time-series of the three-tank process: FMU data versus Linear model and Linear+GP with Periodic-RBF kernel.

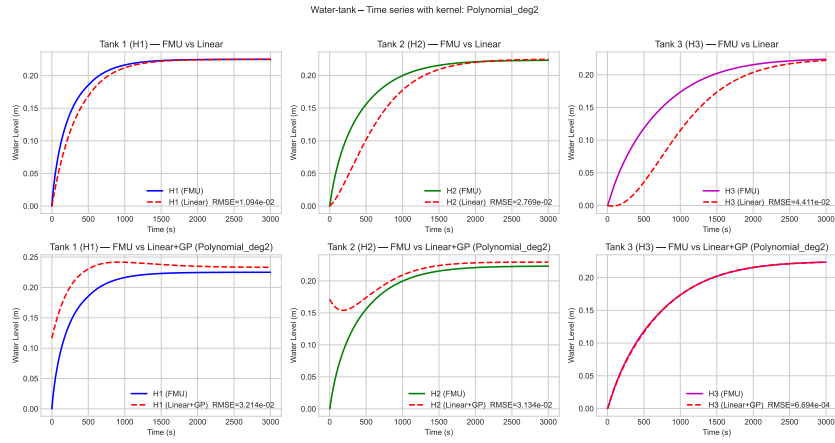


Figure 4.13: Time-series of the three-tank process: FMU data versus Linear model and Linear+GP with Polynomial kernel of degree 2.



Figure 4.14: Time-series of the three-tank process: FMU data versus Linear model and Linear+GP with standard RBF kernel.

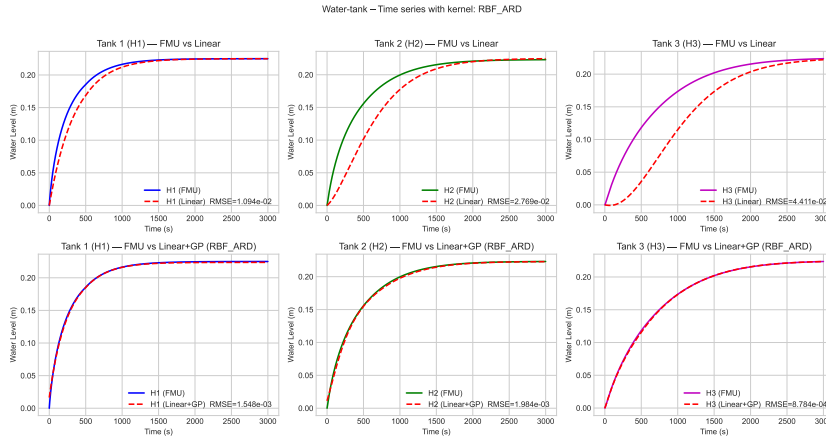


Figure 4.15: Time-series of the three-tank process: FMU data versus Linear model and Linear+GP with RBF kernel and Automatic Relevance Determination (ARD).

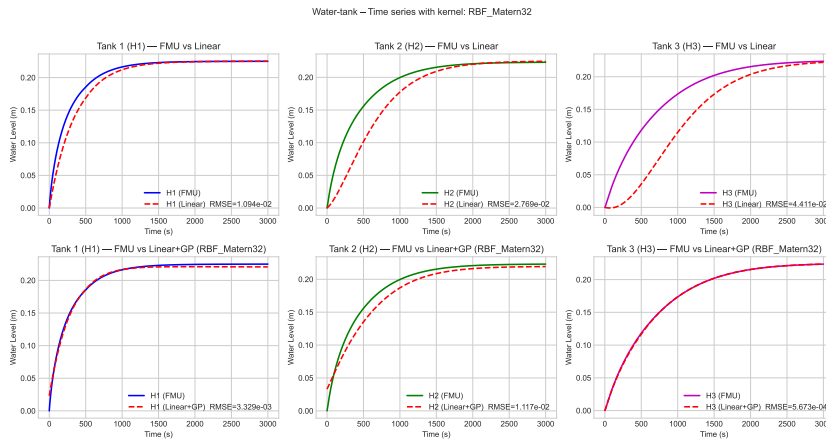


Figure 4.16: Time-series of the three-tank process: FMU data versus Linear model and Linear+GP with combined RBF + Matern-3/2 kernel.

4.2.2 Full GP versus Sparse GP

We compare exact GP (800 training points) to FITC–SGPR with $M \in \{10, 20, 30, 40\}$ inducing points on the three residual channels (*res1–res3*). Averaged across channels, SGPR with $M = 20$ – 30 cuts RMSE from 1.2401 (at $M = 10$) to ≈ 0.984 , restores empirical coverage close to the nominal 95% (94.3–94.5% vs. 82.0%), and is $> 2000\times$ **faster** to retrain than Full GP (0.026–0.028 s vs. 66.5 s). Full GP achieves 100% coverage but retraining is slow. Because R^2 becomes extremely negative on *res1* (tiny target variance makes the constant baseline very strong), we prioritise RMSE and coverage for model choice.

Table 4.5: Water–tank residuals (aggregate over *res1–res3*). Coverage from nominal 95% intervals.

Model	RMSE	R^2	Cover. (%)	Time (s)	M
Full GP	0.8971	-6.41×10^5	100.0	66.513	800
Sparse GP ($M=10$)	1.2401	-5.15×10^5	82.0	0.028	10
Sparse GP ($M=20$)	0.9836	-1.71×10^5	94.5	0.028	20
Sparse GP ($M=30$)	0.9835	-1.69×10^5	94.3	0.026	30
Sparse GP ($M=40$)	0.9840	-1.70×10^5	94.3	0.028	40

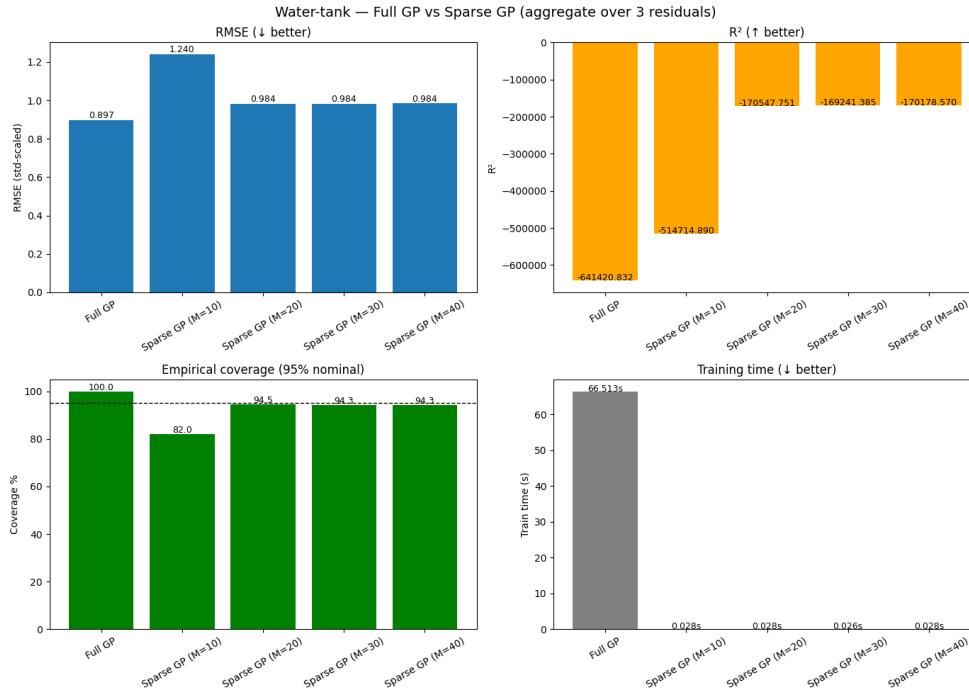


Figure 4.17: Water-tank — Full GP vs. Sparse GP (aggregate over three residuals). Top-left: RMSE (lower is better). Top-right: R^2 (very negative on res1 due to tiny variance; we therefore down-weight it). Bottom-left: empirical coverage vs. 95% nominal. Bottom-right: retraining time (lower is better).

Table 4.6: Three-tank residuals: per-channel comparison (res1–res3).

res1					
Model	RMSE	R^2	Cover. (%)	Time (s)	M
Full GP	1.1531	-1.924×10^6	100.0	98.528	800
Sparse GP ($M=10$)	1.0329	-1.544×10^6	80.5	0.037	10
Sparse GP ($M=20$)	0.5946	-5.116×10^5	100.0	0.032	20
Sparse GP ($M=30$)	0.5923	-5.076×10^5	100.0	0.031	30
Sparse GP ($M=40$)	0.5939	-5.104×10^5	100.0	0.034	40
res2					
Model	RMSE	R^2	Cover. (%)	Time (s)	M
Full GP	1.3313	-45.3281	100.0	73.944	800
Sparse GP ($M=10$)	1.4167	-51.4598	72.5	0.025	10
Sparse GP ($M=20$)	1.2437	-39.4307	83.5	0.030	20
Sparse GP ($M=30$)	1.2447	-39.4983	83.0	0.028	30
Sparse GP ($M=40$)	1.2447	-39.4958	83.0	0.028	40
res3					
Model	RMSE	R^2	Cover. (%)	Time (s)	M
Full GP	0.2067	-0.8914	100.0	27.068	800
Sparse GP ($M=10$)	1.2706	-70.4305	93.0	0.022	10
Sparse GP ($M=20$)	1.1124	-53.7556	100.0	0.021	20
Sparse GP ($M=30$)	1.1135	-53.8627	100.0	0.020	30
Sparse GP ($M=40$)	1.1135	-53.8611	100.0	0.021	40

4.2.3 Probabilistic Control Invariant Set

We certify safety by tightening the CLF condition with the GP–UCB margin $\beta_t \sigma_t$ and taking the resulting sublevel set as a δ –PCIS. For the water–tank plant, the initial certificate (risk level $\varepsilon=0.060$, confidence scale $\beta=2.797$) already surrounds the nominal steady state and covers a useful operating neighbourhood (Fig. 4.18). The admissible ranges at certification time are

$$H_1 \in [0.182, 0.268], \quad H_2 \in [0.121, 0.300], \quad H_3 \in [0.114, 0.300],$$

with set centre $\bar{H} \approx [0.223, 0.219, 0.219]$ and maximum distance from the steady state of 0.130. The scatter projections show that the safe region is mildly anisotropic (wider along H_2 – H_3 where GP uncertainty is lower). As safe data are added, posterior variance contracts and the certificate expands; we quantify this growth by the number of certified grid points (3 695 here) and by the increase in the axis ranges in subsequent iterations, which correlates with the reduced intervention of the safety QP during exploration. The certified set computed from the calibrated posterior encloses the operating region and expands with data (Fig. 4.18).

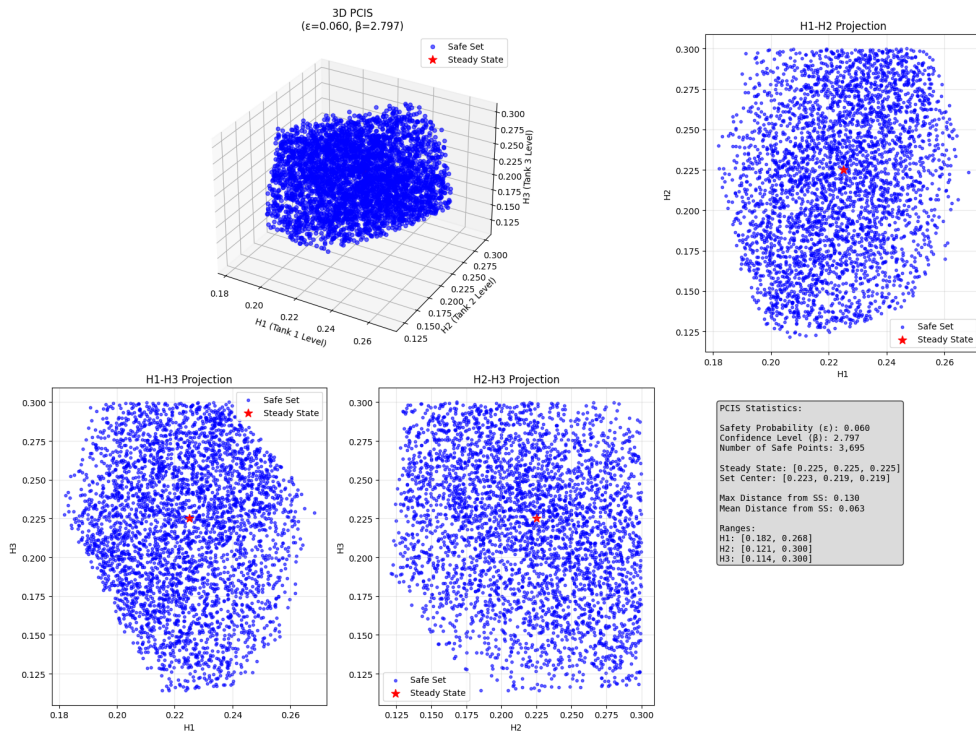


Figure 4.18: Probabilistic Control Invariant Set for the Initial Configuration before Safe Exploration

4.2.4 Unsafe Exploration

As a stress test, we ran exploration without the safety layer. The levels briefly leave the certified band [45%, 87%]: Tank 3 crosses the low–low bound at $t \approx 131$ s, reaching $H_3 = 44.8\%$ while the target was 45.2%. Figure 4.19 (top) shows the trajectory against the LL/HH zones; the middle panel indicates that Valve 2 is driven near saturation early and then steps down just before the violation. The table summarises the event. Even though the excursion is small, it demonstrates why a runtime safety screen is required: small model errors or delays around tight bounds can trigger hard-limit hits.

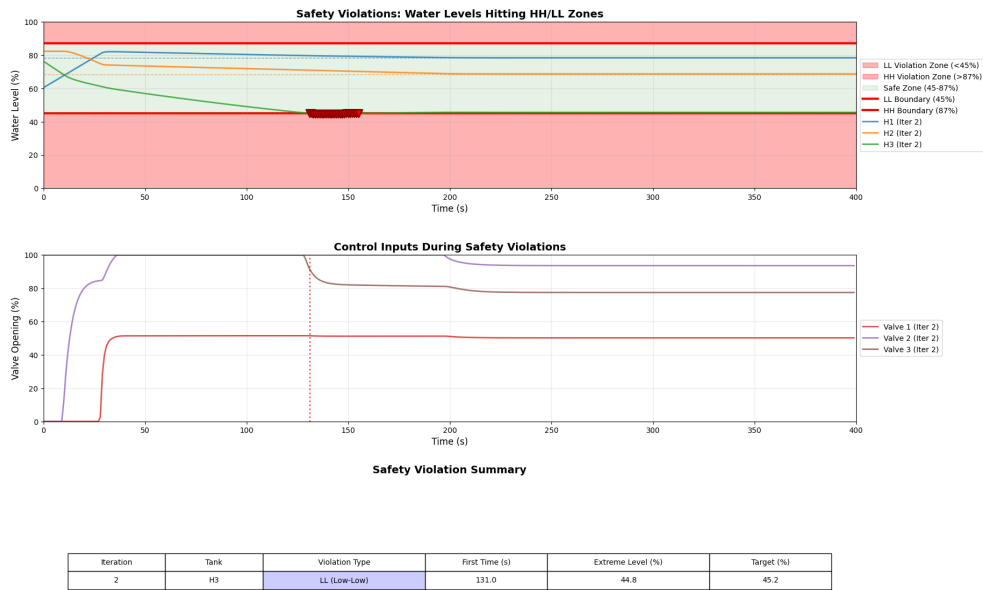


Figure 4.19: Unsafe exploration on the three-tank process. *Top*: water levels vs. safety band; a low-low violation occurs for H_3 at $t \approx 131$ s, with a minimum of 44.8%. *Middle*: valve openings around the violation time. *Bottom*: event summary (iteration, tank, time, extreme level, target).

4.2.5 GP-PCIS safe exploration

The GP-PCIS controller achieves zero violations during experiments. In the following test the framework run for 2000 s (Fig. 4.20).

Furthermore we can notice that:

1. **Safety.** All trajectories remain within the certified band [45%, 87%] for the entire horizon and across all five iterations; no LL/HH boundary crossings are observed. This empirically supports the chance-constraint implementation used in the GP-PCI controller.
2. **Convergence across iterations.** The first two iterations start from off-target initial conditions and reduce the gap towards the dashed references. By iteration 3 the final levels are visually on target for all tanks (steady plateaus close to dashed lines), and iterations 4–5 maintain small terminal errors $e^{(k)}$ while exploring nearby references.
3. **Transient behavior and settling.** Step-like responses exhibit limited overshoot and monotone settling after ≈ 60 –120 s for H_1, H_2 ; H_3 shows

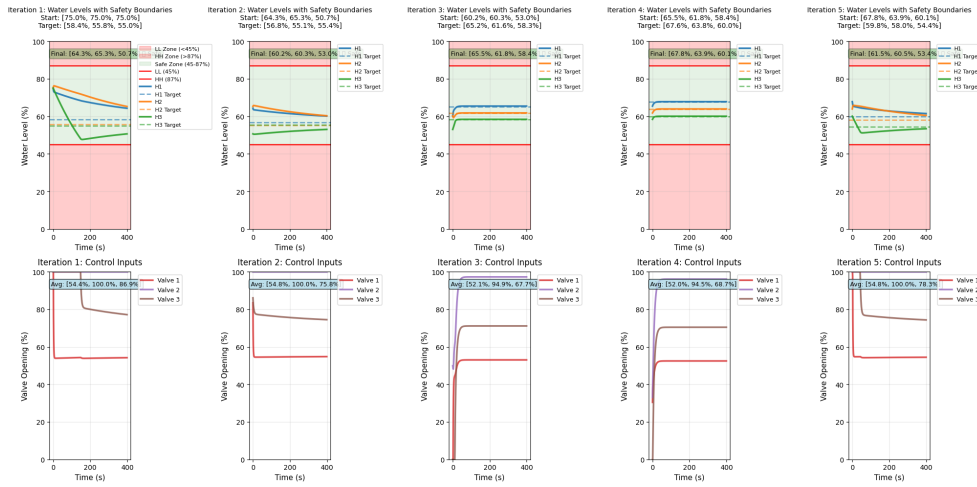


Figure 4.20: Progressive safe exploration on the three-tank process. Top row: water levels $H_{1,3}$ with safety bands (green: safe zone 45–87%; red: LL/HH). Dashed lines are iteration targets; solid lines are trajectories. Bottom row: valve openings $\gamma_{1,3}$. The header of each panel reports start/target/final levels, and the bottom panels report average valve usage.

the largest initial correction in iteration 1 (downstep), but then tracks smoothly from iteration 3 onward. The reduction of visible oscillations across iterations is consistent with shrinking GP uncertainty and milder constraint tightenings.

- Input usage and saturation.** Valve 2 operates near saturation early on (panels 1–2, average near 100%), reflecting the need to quickly transfer mass to Tank 3; its steady usage decreases in iterations 3–4 as the setpoints move closer and the controller exploits better GP predictions. Valve 1 works around the mid-range (~ 50 – 55%), acting as a feed valve to sustain H_1 and upstream flow, while Valve 3 settles in the 65–80% range depending on the target for H_3 .
- Exploration logic.** The sequence of targets (headers in each top panel) steers the system to distinct regions of the safe set without leaving the certified band, illustrating that informative setpoint moves can be scheduled while preserving invariance. As learning progresses, the controller attains similar terminal performance with lower peak actuation (compare iterations 1–2 vs. 3–4).

The combination of GP residuals with probabilistic constraint tightening yields zero observed violations, fast settling, and reduced reliance on input

saturation after a few iterations, evidence that the learned model supports both *safety* and *efficient exploration*.

Synthesis and Answer to the Research Question

Across both benchmarks the proposed GP-PCIS scheme achieves *zero observed violations* while providing exploration that is informative. On the polynomial system it matches the covariance-trace decay and delivers the lowest RMSE, and certified safe-set growth tied to posterior contraction. On the three-tank plant it maintains levels inside [45%, 87%] throughout, and progressively reduces actuation as model confidence improves. These results support the claim that our framework reconcile safety guarantees with efficient learning in nonlinear control.

Chapter 5

Discussion

This chapter synthesizes our framework for *safe exploration* of nonlinear systems via GP-based residual learning, CLF safety filters, and probabilistic control-invariant sets (PCIS). We first examine limits that bound guarantees and performance, such as kernel misspecification, exploration hyperparameters, computational scaling, rare yet critical variance underestimation, and exogenous disturbances, and then summarize what the design achieves in practice, and finally sketch next steps aimed at closing the sim-to-real gap and improving efficiency.

5.1 Method Limitations

The kernel encodes the prior on the unknown residual $g(x, u)$ and a poor choice degrades prediction and calibration; exploration gains and thresholds trade data yield for risk, with overly conservative settings stalling learning and aggressive ones jeopardizing safety; exact GP training scales as $O(N^3)$ whereas the adopted sparse GP reduces this to $O(NM^2)$ with M inducing points; despite being uncommon, a single severe variance underestimation can trigger constraint violations; and unmodelled disturbances are absorbed into the residual, introducing bias if their mean is not zero.

5.2 Achievements

We collect data safely from a partially unknown nonlinear process *without crossing* LoLo/HiHi limits; we construct a δ -PCIS \mathcal{S}_δ that enlarges as GP posterior variance shrinks (up to 30% expansion in our studies); and we

introduce a target-oriented exploration mechanism that steers operation toward informative regions while respecting hard constraints (e.g., safely driving water-tank levels downward during data acquisition).

5.3 Future Directions

The next step is a technology transfer to *ABB 800xA* followed by laboratory validation on the physical plant. In this phase we will deploy the controller on 800xA and execute end-to-end tests on the real setup, covering I/O mapping, signal scaling, and timing verification, while exploiting newly acquired data to retune parameters and quantify performance gains (e.g., tracking error, settling time, and actuation effort). To meet real-time constraints, we will integrate low-latency surrogates for GP inference (sparse, state-space, or neural-kernel variants) so that the sensing–inference–QP loop remains below 50 ms. Finally, we will explicitly model actuator hysteresis (pumps and valves) within the learned residual to increase fidelity near operating limits and improve certified safety margins.

5.3.1 Safe Optimal Experiment Design (OED)

We propose a *safe OED* layer that plans informative, constraint-certified experiments by actively choosing inputs that maximize information about the unknown dynamics while guaranteeing probabilistic safety at all times. Let $x_{k+1} = f_\theta(x_k, u_k) + w_k$ denote the plant with unknown θ , and let \mathcal{I} measure information (e.g., mutual information or a D-optimal log-det criterion). Over a horizon T , we solve

$$\begin{aligned} \max_{u_{0:T-1}} \quad & \mathbb{E}[\mathcal{I}(\theta; \mathcal{D}_T)] \\ \text{s.t.} \quad & \Pr\{x_k \in \mathcal{X}, u_k \in \mathcal{U}, \forall k = 0, \dots, T\} \geq 1 - \delta, \\ & x_{k+1} = \hat{f}_t(x_k, u_k) \pm \text{GP uncertainty}, \quad k = 0, \dots, T - 1. \end{aligned} \tag{5.1}$$

where \hat{f}_t and its uncertainty come from the current GP posterior, and safety is enforced through the δ -PCIS \mathcal{S}_t^δ . A multi-objective variant balances data

quality and control performance,

$$\max_{u_{0:T-1}} \lambda_{\text{info}} \mathbb{E}[\mathcal{J}(\theta; \mathcal{D}_T)] - \lambda_{\text{perf}} \sum_{k=0}^{T-1} \ell(x_k, u_k) \quad \text{s.t. safety as in (5.1).} \quad (5.2)$$

For acquisition, we consider information-theoretic criteria (expected entropy reduction / mutual information with near-greedy submodular surrogates) and Fisher-based OED (A/D-optimal designs via local linearization of the GP mean), both compatible with SQP/MPC inner loops. Safety is maintained by projecting planned trajectories onto \mathcal{S}_t^δ and enforcing a CLF–QP filter at each step: we require $x_{k+1|k} \in \mathcal{S}_t^\delta$ and compute u_k by minimizing the acquisition $\alpha_k(u)$ subject to $\dot{V}_t(x_k, u) + \lambda V(x_k) \leq 0$ and input bounds. The controller workflow at each sampling instant is therefore simple: update the GP posterior (μ_t, Σ_t) with new data; propagate uncertainty to obtain \mathcal{S}_t^δ ; solve (5.1) or (5.2) with chance-constraint tightening; and apply the first input after the CLF–QP safety check.

Evaluation emphasizes *information gain per unit time* (e.g., $\Delta H(\theta)$ or log-det Fisher growth), *model accuracy and calibration* (NRMSE, NLPD, coverage), and *safety/efficiency* (empirical violation rate \hat{p}_{viol} versus δ , PCIS-volume growth, and p_{99} solve time). Target domains include industrial loops (multi-tank, thermal/HVAC) with slow actuators and strict constraints and robotics (manipulators, mobile platforms) requiring persistent excitation inside certified safe regions. Key challenges remain in scaling GP posteriors (sparse inducing points and parallel/safe batch queries) and in tightening chance constraints without undue conservatism; the expected impact is to turn the present *reactive* safe exploration into a *proactive*, information-optimal data-collection engine that accelerates learning while preserving the same safety guarantees, thereby shortening commissioning in industry and reducing trial-and-error on robotic platforms.

Chapter 6

Conclusions

This thesis introduced a data-driven framework for *safe exploration in nonlinear continuous-time systems* that avoids long-horizon MPC. The approach integrates GP residual modeling to capture epistemic uncertainty, probabilistic control invariant sets (PCIS) to provide high-probability safety certificates, and a minimal-intervention CBF/CLF safety filter supervising an uncertainty-aware exploration policy. Methodologically, the work establishes generator-based PCIS conditions in continuous time and links them to exit-time probabilities; derives constructive inner certificates for GP that certify an ellipsoidal safe set $\mathcal{E}_\alpha = \{x : x^\top P x \leq \alpha\}$ and expand it as posterior variance shrinks; and implements a horizon-free safety filter (CBF/CLF-QP) that minimally corrects exploratory inputs to maintain invariance. Empirically, polynomial and water-tank FMU studies demonstrate safe data collection, calibrated GP training, PCIS computation, and closed-loop evaluation: trajectories remained within the certified set, the filter consistently rejected boundary-violating inputs, and performance under moderate uncertainty matched accurate regulation while conservatism decreased as data accumulated (*explore-to-expand* behavior).

The main advantages are explicit and tunable risk control through δ , a modular architecture (GP + safety certificates + exploration) that decouples learning and constraint enforcement, and the absence of receding-horizon solvers at runtime. Limitations stem from conservatism induced by high-probability envelopes and union bounds, the cubic cost of exact GP training and consequent need for sparsification at scale, sensitivity to uncertainty calibration and the choice of β , and an emphasis on instantaneous (boundary) risk rather than explicit multi-step accumulation; partial observability and hardware non-idealities were only partially addressed, as validation was

simulation-based. Natural next steps include tractable multi-step PCIS bounds and distributionally robust variants, extensions to nonstationary or hybrid dynamics with estimator-aware certificates, and active data-acquisition rules that directly maximize safe-set growth per sample; algorithmically, scalable GP surrogates (inducing/state-space/structured kernels), coverage-driven adaptive $\beta(t)$, hardened anytime CBF/CLF filters, and safe policy optimization can improve performance and responsiveness; finally, hardware experiments with fault handling, transfer/meta-learning to warm-start priors and barriers, and multi-agent safety to confirm external validity and operational readiness.

References

- [1] F. Berkenkamp, R. Moriconi, A. P. Schoellig, and A. Krause, “Safe learning of regions of attraction for uncertain, nonlinear systems with gaussian processes,” in *Proceedings of the 55th IEEE Conference on Decision and Control (CDC)*, 2016, pp. 4661–4666. [Page 3.]
- [2] M. Prajapat, A. Lahr, J. Köhler, A. Krause, and M. N. Zeilinger, “Towards safe and tractable gaussian process-based mpc: Efficient sampling within a sequential quadratic programming framework,” *arXiv preprint arXiv:2409.08616*, 2024. [Online]. Available: <https://arxiv.org/abs/2409.08616> [Page 3.]
- [3] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, “Learning-based model predictive control: Toward safe learning in control,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, no. 1, pp. 269–296, 2020. doi: 10.1146/annurev-control-090419-075625 [Pages 3, 5, and 6.]
- [4] M. Dubied, A. Lahr, M. N. Zeilinger, and J. Köhler, “A robust and adaptive mpc formulation for gaussian process models,” *arXiv preprint arXiv:2507.02098*, 2025. [Page 3.]
- [5] M. Prajapat, J. Köhler, A. Lahr, A. Krause, and M. N. Zeilinger, “Finite-sample-based reachability for safe control with gaussian process dynamics,” *arXiv preprint*, 2025, preprint. [Page 4.]
- [6] L. Wang, E. A. Theodorou, and M. Egerstedt, “Safe learning of quadrotor dynamics using barrier certificates,” in *Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018. doi: 10.1109/ICRA.2018.8460471 pp. 2460–2465. [Pages 4 and 6.]
- [7] F. Castañeda, J. J. Choi, B. Zhang, C. J. Tomlin, and K. Sreenath, “Gaussian process-based min-norm stabilizing controller for control-affine systems with uncertain input effects and dynamics,” in

- Proceedings of the 2021 American Control Conference (ACC)*, 2021. doi: 10.23919/ACC50511.2021.9483420 pp. 3683–3690. [Page 4.]
- [8] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, “End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019. doi: 10.1609/aaai.v33i01.33013387 pp. 3387–3395. [Pages 4 and 5.]
- [9] M. Prajapat, J. Köhler, M. Turchetta, A. Krause, and M. N. Zeilinger, “Safe guaranteed exploration for non-linear systems,” *arXiv preprint arXiv:2402.06562*, 2024. [Online]. Available: <https://arxiv.org/abs/2402.06562> [Pages 4, 6, and 9.]
- [10] G. Pillonetto, F. Dinuzzo, T. Chen, G. D. Nicolao, and L. Ljung, “Kernel methods in system identification, machine learning and function estimation: A survey,” *Automatica*, vol. 50, no. 3, pp. 657–682, 2014. doi: 10.1016/j.automatica.2014.01.001 [Page 5.]
- [11] J. F. Fisac, N. F. Lugovoy, V. Rubies-Royo, S. Ghosh, and C. J. Tomlin, “Bridging hamilton–jacobi safety analysis and reinforcement learning,” in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2019. doi: 10.1109/ICRA.2019.8794107 pp. 8550–8556. [Page 5.]
- [12] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. H. Gillula, and C. J. Tomlin, “A general safety framework for learning-based control in uncertain robotic systems,” *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 2737–2752, 2019. doi: 10.1109/TAC.2018.2876389 [Page 5.]
- [13] J. Achiam, D. Held, A. Tamar, and P. Abbeel, “Constrained policy optimization,” in *Proc. 34th Int. Conf. on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 70, 2017, pp. 22–31. [Online]. Available: <http://proceedings.mlr.press/v70/achiam17a.html> [Pages 5 and 12.]
- [14] J. García and F. Fernández, “A comprehensive survey on safe reinforcement learning,” *Journal of Machine Learning Research*, vol. 16, pp. 1437–1480, 2015. [Online]. Available: <http://jmlr.org/papers/v16/garcia15a.html> [Page 6.]

- [15] Y. Gao, K. H. Johansson, and L. Xie, “Computing probabilistic controlled invariant sets,” *arXiv preprint arXiv:1905.04117*, 2019. [Online]. Available: <https://arxiv.org/abs/1905.04117> [Pages 9, 10, and 19.]
- [16] J. Wang and Y. Zhang, “A tutorial on gaussian process learning-based model predictive control,” *arXiv preprint arXiv:2404.03689*, 2024. [Online]. Available: <https://arxiv.org/abs/2404.03689> [Pages 10 and 13.]
- [17] F. Castañeda, J. J. Choi, B. Zhang, C. J. Tomlin, and K. Sreenath, “Gaussian process-based min-norm stabilizing controller for control-affine systems with uncertain input effects and dynamics,” *arXiv preprint arXiv:2011.07183*, 2020. [Online]. Available: <https://arxiv.org/abs/2011.07183> [Pages 10 and 21.]
- [18] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, 2015. doi: 10.1038/nature14236. [Online]. Available: <https://doi.org/10.1038/nature14236> [Page 11.]
- [19] M. G. Bellemare, W. Dabney, and R. Munos, “A distributional perspective on reinforcement learning,” in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 70. PMLR, 2017, pp. 449–458. [Online]. Available: <https://proceedings.mlr.press/v70/bellemare17a.html> [Page 11.]
- [20] M. Hessel, J. Modayil, H. van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver, “Rainbow: Combining improvements in deep reinforcement learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018. doi: 10.1609/aaai.v32i1.11796 pp. 3215–3222. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/11796> [Page 11.]
- [21] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement

- learning,” *arXiv preprint arXiv:1509.02971*, 2015. [Online]. Available: <https://arxiv.org/abs/1509.02971> [Page 11.]
- [22] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017. [Online]. Available: <https://arxiv.org/abs/1707.06347> [Page 11.]
- [23] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor,” in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 80. PMLR, 2018, pp. 1861–1870. [Online]. Available: <https://proceedings.mlr.press/v80/haarnoja18b.html> [Page 11.]
- [24] S. Fujimoto, H. van Hoof, and D. Meger, “Addressing function approximation error in actor-critic methods,” in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 80. PMLR, 2018, pp. 1587–1596. [Online]. Available: <https://proceedings.mlr.press/v80/fujimoto18a.html> [Page 11.]
- [25] S. Paternain, M. Calvo-Fullana, L. F. O. Chamon, and A. Ribeiro, “Safe policies for reinforcement learning via primal–dual methods,” *arXiv preprint arXiv:1911.09101*, 2019. [Online]. Available: <https://arxiv.org/abs/1911.09101> [Page 12.]
- [26] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu, “Safe reinforcement learning via shielding,” in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, 2018, pp. 2669–2678. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/11797> [Page 12.]
- [27] B. Thananjeyan, A. Balakrishna, S. Nair, M. Luo, K. Srinivasan, M. Hwang, J. E. Gonzalez, J. Ibarz, C. Finn, and K. Goldberg, “Recovery rl: Safe reinforcement learning with learned recovery zones,” *arXiv preprint arXiv:2010.15920*, 2020. [Online]. Available: <https://arxiv.org/abs/2010.15920> [Page 12.]
- [28] J. Ji, B. Zhang, J. Zhou, X. Pan, W. Huang, R. Sun, Y. Geng, Y. Zhong, J. Dai, and Y. Yang, “Safety-gymnasium: A unified safe reinforcement

- learning benchmark,” in *NeurIPS 2023 Datasets and Benchmarks Track*, 2023. [Online]. Available: <https://arxiv.org/abs/2310.12567> [Page 12.]
- [29] K. Chua, R. Calandra, R. McAllister, and S. Levine, “Deep reinforcement learning in a handful of trials using probabilistic dynamics models,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. [Online]. Available: <https://doi.org/10.48550/arXiv.1805.12114> [Pages 12 and 13.]
- [30] D. Hafner, T. Lillicrap, M. Norouzi, and J. Ba, “Learning latent dynamics for planning from pixels,” in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019. [Online]. Available: <https://proceedings.mlr.press/v97/hafner19a.html> [Page 12.]
- [31] —, “Mastering atari with discrete world models,” *arXiv preprint arXiv:2010.02193*, 2021. [Online]. Available: <https://arxiv.org/abs/2010.02193> [Page 12.]
- [32] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap, “Mastering diverse domains through world models,” *arXiv preprint arXiv:2301.04104*, 2023. [Online]. Available: <https://arxiv.org/abs/2301.04104> [Page 12.]
- [33] N. Hansen, H. Su, and X. Wang, “Td-mpc2: Scalable, robust world models for continuous control,” *arXiv preprint arXiv:2310.16828*, 2023. [Online]. Available: <https://arxiv.org/abs/2310.16828> [Page 12.]
- [34] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, T. Lillicrap, and D. Silver, “Mastering atari, go, chess and shogi by planning with a learned model,” *Nature*, vol. 588, pp. 604–609, 2020. doi: 10.1038/s41586-020-03051-4. [Online]. Available: <https://www.nature.com/articles/s41586-020-03051-4> [Page 12.]
- [35] L. Hewing, J. Kabzan, and M. N. Zeilinger, “Cautious model predictive control using gaussian process regression,” *arXiv preprint arXiv:1705.10702*, 2017. [Online]. Available: <https://arxiv.org/abs/1705.10702> [Pages 12 and 13.]
- [36] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, “Learning-based model predictive control: Toward safe learning in control,” *Annual Review of Control, Robotics, and Autonomous Systems*,

- vol. 3, pp. 269–296, 2020. doi: 10.1146/annurev-control-090419-075625. [Online]. Available: <https://www.annualreviews.org/content/journals/10.1146/annurev-control-090419-075625> [Pages 12 and 13.]
- [37] J. Wang and Y. Zhang, “A tutorial on gaussian process learning-based model predictive control,” *arXiv preprint arXiv:2404.03689*, 2024. [Online]. Available: <https://arxiv.org/abs/2404.03689> [Pages 12 and 13.]
- [38] D. Romeres, D. Jha, R. Camoriano, J. C. G. Higuera, L. Natale, B. Schölkopf, J. Peters, M. Aghajarian, S. Chitta, and R. Bischoff, “Semiparametrical gaussian processes learning of robot dynamics for control,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019. doi: 10.1109/ICRA.2019.8794229. [Online]. Available: <https://doi.org/10.1109/ICRA.2019.8794229> [Pages 12 and 13.]
- [39] M. Janner, J. Fu, M. Zhang, and S. Levine, “When to trust your model: Model-based policy optimization,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. [Online]. Available: <https://arxiv.org/abs/1906.08253> [Pages 12 and 13.]
- [40] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, “Learning-based model predictive control for safe exploration,” in *Proceedings of the 57th IEEE Conference on Decision and Control (CDC)*, Miami Beach, FL, USA, December 2018. [Online]. Available: <https://las.inf.ethz.ch/files/koller18safempc.pdf> [Page 13.]
- [41] T. Koller, F. Berkenkamp, M. Turchetta, J. Boedecker, and A. Krause, “Learning-based model predictive control for safe exploration and reinforcement learning,” *arXiv preprint arXiv:1906.12189*, 2019. [Online]. Available: <https://arxiv.org/abs/1906.12189> [Page 13.]
- [42] M. Prajapat, J. Köhler, A. Lahr, A. Krause, and M. N. Zeilinger, “Finite-sample-based reachability for safe control with gaussian process dynamics,” *arXiv preprint arXiv:2505.07594*, 2025. [Online]. Available: <https://arxiv.org/abs/2505.07594> [Page 13.]
- [43] D. C. Liu and J. Nocedal, “On the limited memory bfgs method for large scale optimization,” *Mathematical Programming*, vol. 45, no. 1–3, pp. 503–528, 1989. [Page 15.]
- [44] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. Springer, 2006. [Page 15.]

- [45] E. Snelson and Z. Ghahramani, “Sparse gaussian processes using pseudo-inputs,” in *Advances in Neural Information Processing Systems 18 (NeurIPS)*, Y. Weiss, B. Schölkopf, and J. C. Platt, Eds. MIT Press, 2006, pp. 1257–1264. [Page 16.]
- [46] M. K. Titsias, “Variational learning of inducing variables in sparse Gaussian processes,” in *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, ser. JMLR Workshop and Conference Proceedings, vol. 5, 2009, pp. 567–574. [Page 16.]
- [47] J. Hensman, N. Fusi, and N. D. Lawrence, “Gaussian processes for big data,” in *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2013, pp. 282–290. [Page 16.]
- [48] B. D. O. Anderson and J. B. Moore, *Optimal Control: Linear Quadratic Methods*, ser. Information and System Sciences Series. Englewood Cliffs, NJ: Prentice–Hall, 1990. [Page 21.]
- [49] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger, “Gaussian process optimization in the bandit setting: No regret and experimental design,” in *ICML*, 2010. [Page 29.]
- [50] T. Desautels, A. Krause, and J. Burdick, “Parallelizing exploration–exploitation tradeoffs in gaussian process bandit optimization,” in *AISTATS*, 2012. [Page 29.]
- [51] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006. [Page 96.]
- [52] D. Duvenaud, “The kernel cookbook: Advice on covariance functions,” University of Cambridge, Tech. Rep., 2014. [Page 96.]
- [53] M. Ebden, “Gaussian processes for regression: A quick introduction,” University of Oxford, Tech. Rep., 2015. [Page 96.]
- [54] A. G. Wilson and R. P. Adams, “Gaussian process kernels for pattern discovery and extrapolation,” in *Proceedings of the 30th International Conference on Machine Learning*, 2013, pp. 1067–1075. [Pages 96 and 97.]

Appendix A

System Modeling

This chapter develops the model of the three-tank system used throughout the thesis. We first derive a nonlinear physical model from fluid mechanics, including geometry-dependent cross-sections and valve characteristics, then obtain steady-state relations and linear models around selected operating points. These models serve as the basis for control design and for the digital-twin implementation used in experiments.

A.1 Physical Model of the Multitank System

Torricelli's Law and Orifice Flow

Starting from Bernoulli's equation for an incompressible, inviscid fluid between the free surface and the outlet,

$$\frac{p_1}{\rho} + \frac{v_1^2}{2} + gy_1 = \frac{p_2}{\rho} + \frac{v_2^2}{2} + gy_2, \quad (\text{A.1})$$

and assuming both points at atmospheric pressure and negligible surface velocity, we obtain Torricelli's law

$$v = \sqrt{2gh}, \quad Q = C_d A \sqrt{2gh}, \quad (\text{A.2})$$

where Q is the volumetric flow, A the orifice area, $C_d \in (0, 1)$ a discharge coefficient accounting for contraction and losses, g the gravitational acceleration, and h the liquid head. For empirical fitting we use the power-law surrogate

$$Q = C h^\alpha, \quad (\text{A.3})$$

with unknown parameters (C, α) estimated experimentally (ideal $\alpha = 0.5$, but real systems deviate due to turbulence and geometry).

Tank Dynamics in Volume and Height

Let V_i and h_i denote volume and level of tank $i \in \{1, 2, 3\}$. The interconnection is vertical: tank 1 discharges into tank 2, and tank 2 into tank 3, which drains to a reservoir. A pump injects an inflow q into tank 1. With outflow laws (A.3), the volume balances read

$$\begin{aligned}\dot{V}_1 &= q - C_1 h_1^{\alpha_1}, \\ \dot{V}_2 &= C_1 h_1^{\alpha_1} - C_2 h_2^{\alpha_2}, \\ \dot{V}_3 &= C_2 h_2^{\alpha_2} - C_3 h_3^{\alpha_3}.\end{aligned}\tag{A.4}$$

Control and monitoring, however, are expressed in *heights* h_i , which satisfy

$$\dot{V}_i = A_i(h_i) \dot{h}_i, \quad A_i(h_i) \triangleq \frac{dV_i}{dh_i},\tag{A.5}$$

where $A_i(\cdot)$ is the height-dependent cross-section.

Geometry-dependent cross-sections. Based on the fabricated tank shapes (cf. Fig. A.1) we use:

$$A_1(h_1) = a w, \quad A_2(h_2) = c w + \frac{h_2}{h_{2,\max}} b w, \quad A_3(h_3) = w \sqrt{R^2 - (h_{3,\max} - h_3)^2},\tag{A.6}$$

with geometric constants $(a, b, c, w, R, h_{2,\max}, h_{3,\max})$ measured offline. Substituting (A.6) and (A.5) into (A.4) yields the *height* dynamics:

$$\begin{aligned}\dot{h}_1 &= \frac{1}{A_1(h_1)} (q - C_1 h_1^{\alpha_1}), \\ \dot{h}_2 &= \frac{1}{A_2(h_2)} (C_1 h_1^{\alpha_1} - C_2 h_2^{\alpha_2}), \\ \dot{h}_3 &= \frac{1}{A_3(h_3)} (C_2 h_2^{\alpha_2} - C_3 h_3^{\alpha_3}).\end{aligned}\tag{A.7}$$

Valve Actuation and Full Nonlinear Model

Each outlet valve has an opening $\gamma_i \in [0, 100]$ % governing its discharge. We use the linear valve map $C_i(\gamma_i) = \tilde{C}_i \gamma_i$. Incorporating this into (A.7) gives

the full nonlinear model used in control:

$$\begin{aligned}
 \dot{h}_1 &= \frac{1}{aw} (q - C_1(\gamma_1) h_1^{\alpha_1}), \\
 \dot{h}_2 &= \frac{1}{cw + \frac{h_2}{h_{2,\max}} bw} (C_1(\gamma_1) h_1^{\alpha_1} - C_2(\gamma_2) h_2^{\alpha_2}), \\
 \dot{h}_3 &= \frac{1}{w\sqrt{R^2 - (h_{3,\max} - h_3)^2}} (C_2(\gamma_2) h_2^{\alpha_2} - C_3(\gamma_3) h_3^{\alpha_3}).
 \end{aligned} \tag{A.8}$$

Steady states. For constant inflow q and fixed openings γ_i , the equilibrium levels h_i^0 solve

$$q = C_1(\gamma_1)(h_1^0)^{\alpha_1} = C_2(\gamma_2)(h_2^0)^{\alpha_2} = C_3(\gamma_3)(h_3^0)^{\alpha_3}. \tag{A.9}$$

Figure A.1: Cross-sectional areas $A_i(h_i)$ defined in (A.6). Tank 1 is prismatic (constant area), tank 2 is linearly varying (trapezoid), and tank 3 exhibits strong nonlinearity at low levels.

A.2 Linear Modeling for Control

We derive linear models around operating points spanning the range of interest. Following common industrial constraints, the *valves* $\gamma_1, \gamma_2, \gamma_3$ are treated as control inputs and the pump flow q as an external disturbance.

Operating Points and Nonlinearity Analysis

Two effects dominate the nonlinearity: (i) height-dependent cross-sections (especially tank 3 at low levels) and (ii) power-law outflows h^α . To cover both highly nonlinear and more linear regimes, we select three operating points (OPs) close to 25%, 50%, and 75% of max levels (see Table A.1); for tank 2 the low point is set to 33% to ensure robust valve operation. For each OP we first choose a feasible pump flow q (limited range), then compute the valve openings via (A.9).

Table A.1: Operating points for linearisation (levels in cm).

OP	h_1	h_2	h_3	$q_{\text{pump}} [\text{m}^3/\text{s}]$	γ_{pump}	γ_1	γ_2	γ_3
Low (25%)	7.5	10.0 (33%)	7.5	1.49×10^{-5}	44	55.4	95.0	74.3
Middle (50%)	15.0	15.0	15.0	1.49×10^{-5}	44	44.9	82.1	58.5
High (75%)	22.5	22.5	22.5	1.91×10^{-5}	45	50.9	90.9	65.2

Analytical Linearisation

Let $x = [h_1, h_2, h_3]^\top$, $u = [\gamma_1, \gamma_2, \gamma_3]^\top$, and $r = q$ (disturbance). Define the vector field $f(x, u, r)$ by (A.8) with $C_i(\gamma_i) = \tilde{C}_i \gamma_i$. Around an equilibrium (x^0, u^0, r^0) we obtain

$$\dot{\tilde{x}} \approx A \tilde{x} + B \tilde{u} + E \tilde{r}, \quad \tilde{x} = x - x^0, \quad \tilde{u} = u - u^0, \quad \tilde{r} = r - r^0, \quad (\text{A.10})$$

with Jacobians $A = \left. \frac{\partial f}{\partial x} \right|_0$, $B = \left. \frac{\partial f}{\partial u} \right|_0$, $E = \left. \frac{\partial f}{\partial r} \right|_0$. Representative entries (evaluated at the OP) are:

$$\begin{aligned} \left. \frac{\partial f_1}{\partial h_1} \right|_0 &= -\frac{\gamma_1^0 \tilde{C}_1 \alpha_1 (h_1^0)^{\alpha_1 - 1}}{aw}, & \left. \frac{\partial f_1}{\partial \gamma_1} \right|_0 &= -\frac{\tilde{C}_1 (h_1^0)^{\alpha_1}}{aw}, & \left. \frac{\partial f_1}{\partial r} \right|_0 &= \frac{1}{aw'}, \\ \left. \frac{\partial f_2}{\partial h_1} \right|_0 &= \frac{\gamma_1^0 \tilde{C}_1 \alpha_1 (h_1^0)^{\alpha_1 - 1}}{A_2(h_2^0)}, & \left. \frac{\partial f_2}{\partial h_2} \right|_0 &= -\frac{\gamma_2^0 \tilde{C}_2 \alpha_2 (h_2^0)^{\alpha_2 - 1}}{A_2(h_2^0)} - \frac{(C_1 h_1^{\alpha_1} - C_2 h_2^{\alpha_2})}{A_2(h_2^0)} \frac{\partial \log A_2}{\partial h_2} \Big|_{h_2^0}, \\ \left. \frac{\partial f_3}{\partial h_2} \right|_0 &= \frac{\gamma_2^0 \tilde{C}_2 \alpha_2 (h_2^0)^{\alpha_2 - 1}}{A_3(h_3^0)}, & \left. \frac{\partial f_3}{\partial h_3} \right|_0 &= -\frac{\gamma_3^0 \tilde{C}_3 \alpha_3 (h_3^0)^{\alpha_3 - 1}}{A_3(h_3^0)} - \frac{(C_2 h_2^{\alpha_2} - C_3 h_3^{\alpha_3})}{A_3(h_3^0)} \frac{\partial \log A_3}{\partial h_3} \Big|_{h_3^0} \end{aligned}$$

Resulting Linear Models and Time Constants

For each OP we report (A, B) in continuous time. At the *low* OP:

$$\dot{x} = \underbrace{\begin{bmatrix} -0.00688 & 0 & 0 \\ 0.00862 & -0.00771 & 0 \\ 0 & 0.00640 & -0.00816 \end{bmatrix}}_{A_{\text{low}}} x + \underbrace{\begin{bmatrix} -3.08 & 0 & 0 \\ 3.86 & -2.25 & 0 \\ 0 & 1.87 & -2.39 \end{bmatrix}}_{B_{\text{low}}} \times 10^{-5} u, \quad y = I_3 x. \quad (\text{A.11})$$

Eigenvalues are the diagonal entries (lower-triangular A), giving time constants $\tau_i = 1/|\lambda_i|$:

$$\tau_{\text{low}} = [145.4, 129.7, 122.6] \text{ s.}$$

At the *middle* OP:

$$A_{\text{mid}} = \begin{bmatrix} -0.00344 & 0 & 0 \\ 0.00345 & -0.00411 & 0 \\ 0 & 0.00336 & -0.00321 \end{bmatrix}, \quad B_{\text{mid}} = \begin{bmatrix} -3.79 & 0 & 0 \\ 3.81 & -2.08 & 0 \\ 0 & 1.70 & -2.38 \end{bmatrix} \times 10^{-5}, \quad \tau_{\text{mid}} = [290.7, 243.3, 311.9] \text{ s.}$$

(A.12)

At the *high* OP:

$$A_{\text{high}} = \begin{bmatrix} -0.00294 & 0 & 0 \\ 0.00227 & -0.00270 & 0 \\ 0 & 0.00255 & -0.00244 \end{bmatrix}, \quad B_{\text{high}} = \begin{bmatrix} -4.29 & 0 & 0 \\ 3.31 & -1.85 & 0 \\ 0 & 1.75 & -2.44 \end{bmatrix} \times 10^{-5}, \quad \tau_{\text{high}} = [340.1, 370.0, 409.8] \text{ s.}$$

(A.13)

Appendix B

Simulation Environment (Extended)

Repository and Modules

The simulation stack is organized as a Python codebase with modular components for GP learning, safety certification (PCIS), target selection (Safe-UCB), and FMU-based rollouts. Table B.1 summarizes the core files and their roles; Figure B.1 illustrates the FMU Simulink model. A complete tree is maintained in the project repository.

Table B.1: Core modules in the simulation environment and their roles.

<code>gp_model.py</code>	GP training (full/sparse), multi-output residuals, utilities
<code>pcis.py</code>	Probabilistic safe-set (PCIS) computation
<code>safe_ucb.py</code>	Safe-UCB target selection
<code>simulation.py</code>	FMU-based rollouts and I/O helpers
<code>controller.py</code>	Controller
<code>dataset_water_tank.py</code>	Dataset generation utilities
<code>water_tank_plotting.py</code>	Visualization (safe-set, trajectories, uncertainty)
<code>water_tank_validation.py</code>	Validation routines and tests
<code>util.py</code>	General support functions
<code>model/</code>	<code>FMU_WaterTank.fmu</code>
<code>data/, plots/, test/</code>	Datasets, saved figures, notebooks/tests

Plant, Linear Baseline, and FMU Integration

We consider a three-tank water system with states H_1, H_2, H_3 (m) and inputs (valve positions V_1, V_2, V_3 and pump flow). The default sampling time is $T_s = 1.0$ s and the nominal steady-state is $H_{1,ss} = H_{2,ss} = H_{3,ss} = 0.219$ m.

The continuous-time plant $\dot{x}(t) = f(x(t), u(t))$ is linearized about x_{ss} to

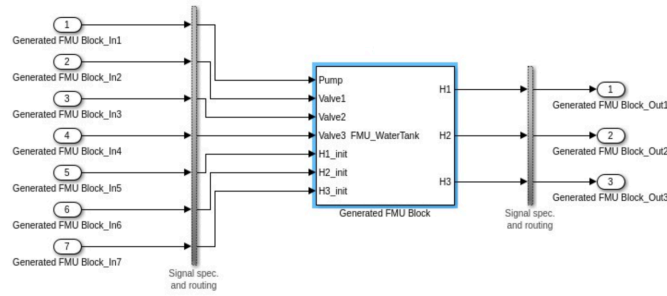


Figure B.1: FMU Simulink model

obtain

$$\dot{x}(t) = A(x(t) - x_{ss}) + Bu(t), \quad A = \begin{bmatrix} 4.483 & 3.184 & 0.939 \\ 3.184 & 5.576 & 1.352 \\ 0.939 & 1.352 & 3.712 \end{bmatrix}, \quad B = [b_{ij}],$$

and discretized via forward Euler for simulation: $x_{k+1} = x_k + T_s f(x_k, u_k)$. The plant is executed through a Simulink-generated FMU (FMU_WaterTank.fmu) with a Python loader that also returns a variable-reference map for efficient I/O. Listing B.1 sketches the typical FMU load sequence.

Listing B.1: FMU loading and model setup.

```

fmu_filename = "model/FMU_WaterTank.fmu"
fmu, vr_map = load_water_tank_fmu(fmu_filename, H1_ss=0.225, H2_ss=0.225,
    ↪ H3_ss=0.225)
model, A, B, K = generate_linear_model(valve_position=73) # LQR gains for
    ↪ tracking
    
```

Residual Modeling with Gaussian Processes

We identify residual dynamics

$$\Delta_i(x, u) = f_i(x, u) - (A(x - x_{ss}) + Bu)_i$$

with independent GPs per state component i , defaulting to a squared-exponential kernel. Given training data $D = \{(x_j, u_j, \Delta_{i,j})\}_{j=1}^N$, the GP yields posterior $\mu_i^*(x, u)$ and variance $\sigma_i^{2*}(x, u)$. For scalability, Sparse GP Regression (SGPR) with M inducing points Z (K-means initialization) is used; Listing B.2 summarizes the training routine. Multi-output GP training instantiates one SGPR per residual component.

Listing B.2: Sparse GP training (SGPR) with K-means inducing points.

```

def train_sparse_gp(X_data, Y_data, M=30):
    Z_init = KMeans(n_clusters=M, random_state=0).fit(X_data).cluster_centers_
    kernel = gpflow.kernels.SquaredExponential()
    model = gpflow.models.SGPR(data=(X_data, Y_data),
                               kernel=kernel,
                               inducing_variable=InducingPoints(Z_init))
    opt = tf.optimizers.Adam(learning_rate=1e-2)
    for _ in range(500):
        with tf.GradientTape() as tape:
            loss = model.training_loss()
            grads = tape.gradient(loss, model.trainable_variables)
            opt.apply_gradients(zip(grads, model.trainable_variables))
    return model

```

Probabilistic Safety: PCIS Computation

We certify safety on an ellipsoidal candidate region $\{x : x^\top P x \leq \alpha\}$ where $P > 0$ solves the continuous-time ARE. Given design parameters $(\delta, \kappa, \lambda, \alpha)$, define

$$\beta = \sqrt{2 \log(1/\delta)}, \quad \varepsilon = \kappa \frac{\lambda}{2\sqrt{\lambda_{\max}(P)}} \sqrt{\alpha}.$$

For each candidate x in the ellipsoid, we enforce $|\mu_i^*(x, u)| + \beta \sigma_i^*(x, u) \leq \varepsilon$ for all residual components i . Listing B.3 shows the reference implementation.

Listing B.3: PCIS: computing a discrete safe set by GP-based chance constraints.

```

def compute_safe_set(P_c, gp_dict, x_ss, delta=0.04, alpha_m=1.0, lam=1.0,
                    kappa=0.8, N=20000):
    beta = np.sqrt(2.0 * np.log(1.0 / delta))
    lambda_max = np.max(np.real(np.linalg.eigvals(P_c)))
    eps = kappa * (lam / (2.0 * np.sqrt(lambda_max))) * np.sqrt(alpha_m)
    X = sample_uniform_states(N, bounds=x_bounds)
    mask_ell = np.einsum('ij,jk,ik->i', X, P_c, X) <= alpha_m
    X_in = X[mask_ell]
    safe_mask = np.ones(len(X_in), dtype=bool)
    for _, gp in gp_dict.items():
        mu, var = predict_gp_residuals(gp, X_in)
        safe_mask &= (np.abs(mu) + beta * np.sqrt(var) <= eps).ravel()
    return X_in[safe_mask], eps, beta

```

Safe-UCB Target Selection

Among certified states S_{safe} , the exploration target maximizes an optimistic score:

$$\text{UCB}(x) = \sum_i (\mu_i^*(x) + \beta \sigma_i^*(x)), \quad x^* \in \arg \max_{x \in S_{\text{safe}}} \text{UCB}(x).$$

A reference function `select_target_ucb` computes UCB and returns x^* ; the minimal implementation is shown in *Listing B.4*.

Listing B.4: Safe-UCB target selection over the certified set.

```
def select_target_ucb(X_candidates, gp_dict, beta):
    if len(X_candidates) == 0: return None
    scores = []
    for x in X_candidates:
        s = 0.0
        for _, gp in gp_dict.items():
            mu, var = predict_single_point(gp, x)
            s += (mu + beta * np.sqrt(var))
        scores.append(float(s))
    return X_candidates[int(np.argmax(scores))]
```

Closed-Loop Orchestration

The exploration loop iterates: (i) compute S_{safe} via PCIS; (ii) select x^* with Safe-UCB; (iii) execute LQR tracking on the FMU; (iv) append trajectory data and re-train GPs. The canonical implementation (`safe_exploration_loop`) is drafted in *Listing B.5*. Outputs include trajectories, safe-set snapshots, GP model checkpoints, and summary statistics.

Listing B.5: High-level safe exploration loop (FMU + GP + PCIS + UCB).

```
def safe_exploration_loop(gp_models, df0, fmu, vr_map, model, A, B, K,
    n_iterations=10):
    trajs, D, models = [], df0.copy(), gp_models.copy()
    x_cur = np.array([model["H1_ss"], model["H2_ss"], model["H3_ss"]])
    for _ in range(n_iterations):
        P_c = solve_continuous_are(A, B, np.eye(3), 1e-8 * np.eye(3))
        X_safe, eps, beta = compute_safe_set(P_c, models, x_ss=x_cur)
        x_star = select_target_ucb(X_safe, models, beta)
        if x_star is None: break
        res = simulate_water_tank_lqr_tracking(x_cur, x_star, fmu, vr_map,
            model, A, B, K, N=1000, Ts=1.0)
        trajs.append(res); D = pd.concat([D, res['df']]); x_cur =
            res['x_fmu_hist'][-1]
        models = retrain_gp_models(D)
    return {"trajectories": trajs, "data": D, "models": models}
```

Configuration, Installation, and Usage

Default configuration exposes the PCIS confidence δ , ellipsoid scale α_m , Lyapunov rate λ , safety scaling κ , SGPR inducing count M , and exploration horizons. Installation and a five-step workflow (load FMU, generate data, train GPs, run loop, plot) are as follows.

Listing B.6: Installation and typical workflow.

```
# Dependencies (conda/venv recommended)
pip install numpy scipy matplotlib pandas gpflow tensorflow scikit-learn

# Quickstart in a notebook
jupyter notebook main.ipynb
# 1) load FMU and linear model; 2) build initial dataset; 3) train GPs;
# 4) run safe_exploration_loop(); 5) plot_exploration_summary(results)
```

Key defaults (adjust in experiments) are summarized in Table B.2.

Table B.2: Configuration defaults exposed by the environment.

Parameter	Default	Description
delta	0.02	PCIS confidence parameter
alpha_m	0.2	Ellipsoid scaling factor
lam	1.0	Lyapunov decay rate
kappa	0.8	Safety margin scaling
gp_M	30	Inducing points (SGPR)
n_iterations	10	Exploration iterations
n_simulation_steps	1000	Steps per rollout

Logged Artifacts and Plots

The framework logs datasets (initial/exploration/validation), GP checkpoints, safe-set histories, and control trajectories. Visualization utilities produce phase plots, level trajectories, safe-set evolution, uncertainty maps, data accumulation, and exploration statistics; `plot_exploration_summary` aggregates these into a single dashboard. *Listing B.7* shows the entry point.

Listing B.7: Exploration summary plotting.

```
from water_tank_plotting import plot_exploration_summary
results = safe_exploration_loop(...)
plot_exploration_summary(results)
```

Numerical Considerations and Good Practices

(i) *Variance floors and calibration*: impose a minimum noise variance and (optionally) post-hoc variance calibration on a validation split before using σ^* inside safety predicates; this avoids optimistic certification under MLE shrinkage. (ii) *Inducing-point quality*: prefer K-means initialization and re-initialization when residual support shifts; monitor M vs. RMSE/coverage.

(iii) *ARE conditioning*: scale Q, R to avoid ill-conditioned P that would distort ε through $\lambda_{\max}(P)$. (iv) *Batching FMU I/O*: cache variable references and pre-allocate buffers for stable, deterministic rollouts.

Appendix C

Gaussian Process Kernels: A Practical Guide

A Gaussian process prior is defined by mean $m(x)$ and covariance $k_\theta(x, x')$ with hyperparameters θ . Let $r = \|x - x'\|_2$, signal variance $\sigma_f^2 > 0$, and ARD matrix $\Lambda = \text{diag}(\ell_1^{-2}, \dots, \ell_d^{-2})$. References: [51, 52, 53, 54].

C.0.1 Common kernels

Squared Exponential / RBF (stationary, very smooth)

$$k_{\text{SE}}(x, x') = \sigma_f^2 \exp\left(-\frac{1}{2}(x - x')^\top \Lambda (x - x')\right).$$

Infinitely differentiable; excellent interpolation; poor extrapolation.

Matérn family (stationary, tunable roughness)

$$k_{\text{Mat } \nu}(r) = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \tilde{r}\right)^\nu K_\nu\left(\sqrt{2\nu} \tilde{r}\right), \quad \tilde{r} = \sqrt{(x - x')^\top \Lambda (x - x')}.$$

Closed forms at $\nu \in \{\frac{1}{2}, \frac{3}{2}, \frac{5}{2}\}$. Often better empirical coverage than SE.

Rational Quadratic (scale mixture of SE)

$$k_{\text{RQ}}(x, x') = \sigma_f^2 \left(1 + \frac{1}{2}(x - x')^\top \Lambda (x - x')/\alpha\right)^{-\alpha}.$$

Captures multi-scale variation; $\alpha \rightarrow \infty$ recovers SE.

Linear and Polynomial (global trends)

$$k_{\text{lin}}(x, x') = \sigma_b^2 + \sigma_v^2 (x - c)^\top (x' - c), \quad k_{\text{poly}}(x, x') = (\sigma_b^2 + (x - c)^\top (x' - c))^d.$$

Useful for extrapolating affine/low-order trends; combine with SE/Matérn for residuals.

Periodic and Locally-Periodic (repeating structure)

$$k_{\text{per}}(x, x') = \sigma_f^2 \exp\left(-\frac{2}{\ell^2} \sin^2(\pi\|x - x'\|/p)\right), \quad k_{\text{locper}} = k_{\text{per}} \times k_{\text{SE}}.$$

Handles periodic effects with local modulation.

Spectral Mixture (expressive stationary)

$$k_{\text{SM}}(x, x') = \sum_{q=1}^Q w_q \prod_{j=1}^d \exp(-2\pi^2(x_j - x'_j)^2 v_{qj}) \cos(2\pi(x_j - x'_j)\mu_{qj}).$$

Discovers multiple frequencies/length-scales; enables extrapolation [54].

Automatic Relevance Determination (ARD) Replace ℓ by Λ in any kernel to down-weight irrelevant inputs.

C.0.2 Compositions and nonstationarity

Sums $k = k_1 + k_2$ model additive structure (trend + seasonal + local). Products $k = k_1 k_2$ intersect properties (e.g., locally-periodic = periodic \times SE). Change-points and input warping create nonstationarity for regime changes. Known physics can live in the mean; the kernel handles residuals.

C.0.3 Control-oriented guidance

Local prediction and safety: prefer SE/Matérn with ARD; Matérn often improves coverage. Trends and extrapolation: add linear/polynomial; use periodic/SM for repeating dynamics. Calibration: if nominal 95% intervals under-cover, increase β_t (GP-UCB) or inflate variance on a validation split while keeping μ fixed. Computation: choose kernels with analytic derivatives for fast L-BFGS; for long runs use sparse variational GPs with inducing points initialised by k -means.

C.0.4 Minimal formulas to cite

SE (ARD) : $k = \sigma_f^2 \exp\left(-\frac{1}{2}(x - x')^\top \Lambda (x - x')\right)$.

Matérn $\nu = \frac{3}{2}$: $k = \sigma_f^2 (1 + \sqrt{3} \tilde{r}) \exp(-\sqrt{3} \tilde{r})$.

Rational Quadratic : $k = \sigma_f^2 \left(1 + \frac{1}{2}(x - x')^\top \Lambda (x - x') / \alpha\right)^{-\alpha}$.

Periodic : $k = \sigma_f^2 \exp\left(-\frac{2}{l^2} \sin^2(\pi \|x - x'\| / p)\right)$.

Linear : $k = \sigma_b^2 + \sigma_v^2 (x - c)^\top (x' - c)$.

Spectral Mixture : $k = \sum_{q=1}^Q w_q \prod_{j=1}^d \exp(-2\pi^2 (x_j - x'_j)^2 v_{qj}) \cos(2\pi (x_j - x'_j) \mu_{qj})$.

TRITA-EECS-EX-2025:940
Stockholm, Sverige 2024

www.kth.se