Error Bounds Estimate for Gaussian Processes and Kernel Methods and Application in Stability Analysis

by

Yirun Fu

A research paper
presented to the University of Waterloo
in fulfillment of the
research paper requirement for the degree of
Master
in
Applied Mathematics

Waterloo, Ontario, Canada, 2025

Author's Declaration

I hereby declare that I am the sole author of this research paper. This is a true copy of the research papaer, including any required final revisions, as accepted by my readers.

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

Abstract

Gaussian processes and kernel methods are two learning-based approaches to model unknown functions. A major advantage of GPs is the existence of simple analytic formulas for the mean and covariance of the posterior distribution, which allows easy implementations of the algorithms. The models provided by kernel methods also have the same advantage. In order to deploy such learning-based models in safety-critical applications, it is important to rigorously quantify the errors between the learned models and the real physical systems. As a result, we are more interested in obtaining uniform error bounds that the unknown function cannot go beyond or stays in with high probability than an estimation function only. This research paper can be partitioned into two parts. Error bounds estimate by Gaussian Processes and kernel methods will be introduced in the first part, and a comparison with Bayesian Neural Networks is also included here. We then demonstrate those error bounds estimate are meaningful by showing its application in stability analysis. Moreover, an unacceptable assumption has been made and there are deficiencies which result in loss of theoretical guarantees in the experiments in previous research. We adjust the assumptions to get more reasonable results and provide numerical examples which is consistent with theoretical derivation and discuss remaining issues.

Acknowledgements

I would like to thank all the people who made this research paper possible.

Dedication

This is dedicated to the one my family.

Table of Contents

Author's Declaration ii

Abstract iii

Acknowledgements iv

Dedication v

List of Figures viii

List of Tables ix

1 Introduction 1

1.1 Motivation 1

1.2 Scope of the Research Paper 2

2 Learning Unknown Function 3

2.1 Problem Definition 3

2.1.1 RKHS Norm 4

2.1.2 RKHS Norm Estimate 4

2.2 Gaussian Processes Approach 5

2.2.1 GPs Probabilistic Error Bounds 6

2.2.2 GPs Deterministic Error Bounds . . . 7
2.3 Kernel Methods Approach . . . 7
2.3.1 Noise Free Case . . . 8
2.3.2 Kernel Ridge Regression Analysis . . . 9
2.3.3 ε-Support Vector Regression Analysis . . . 10

3 Stability Analysis **11**
3.1 Lyapunov Theory . . . 11
3.2 Safe Learning . . . 13

4 Experiments **17**
4.1 Bayesian Neural Networks . . . 17
4.2 A Comparison Among Different Approaches . . . 18
4.3 Issues with Truncated Gaussian Noise . . . 18
4.4 One Dimensional Stability Analysis . . . 21
4.5 Higher Dimensional Stability Analysis . . . 23

5 Disscussion **30**

References **32**

List of Figures

1.1 (a) Example of temperature data collected by a network of 46 sensors at Intel Research Berkeley. (b,c) Two iterations of the GP-UCB algorithm. It samples points that are either uncertain (b) or have high posterior mean (c). 2

2.1 Estimating the RKHS norm using randomly sampled data (all circles). The quadratic form $\hat{\Gamma}$ for the random samples is shown on the right plot. If one sampled only the black subset of the data, the corresponding $\hat{\Gamma}$ would capture over 90% of the total complexity, i.e., $\hat{\Gamma}/\|f\|_{\mathcal{H}} > 0.9$. 5

3.1 Algorithm 1. 16

4.1 Ground-truth (- -), KRR (blue), SVR (green), and GPs (yellow). The error bounds are depicted using the same colors of their respective models, and were computed for $N = 20$ (top) and $N = 100$ (bottom) samples. The noisy data-points are shown as black circles. 19

4.2 Comparison of BNNs and GPs 20

4.3 Chi-square distribution table 22

4.4 True $\dot{V}(x)$ (blue), GPs upper bound of $\dot{V}(x)$ (orange), and $y = -L\tau$ (green). 23

4.5 True $\dot{V}(x)$ (orange), KRR upper bound of $\dot{V}(x)$ (blue), and $y = -L\tau$ and $y = 0$ (yellow). 24

4.6 Experiment logic 26

4.7 Learned Inverted Pendulum Dynamics 27

4.8 ROA learned from 100 measurements learned β 28

4.9 ROA learned from 100 measurements with $\beta = 2\Gamma^2$ 29

List of Tables

4.1 Comparison of different approaches. 19
4.2 Inverted pendulum 27

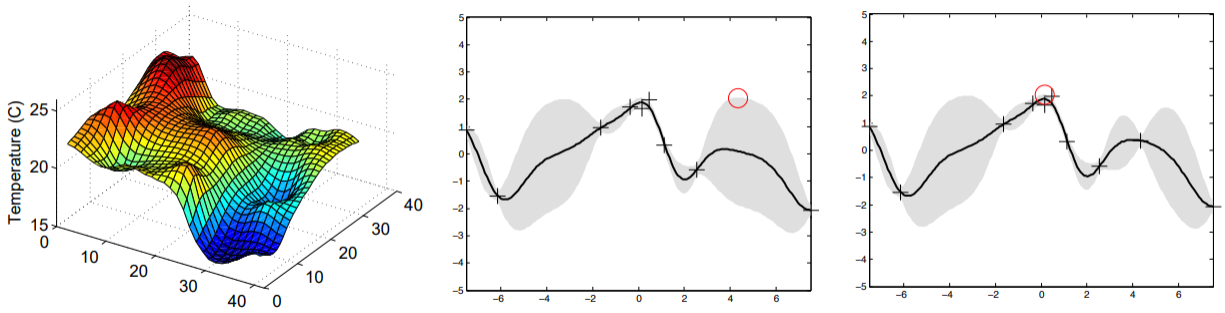
Chapter 1

Introduction

1.1 Motivation

Consider an unknown function $f : \mathcal{X} \to \mathbb{R}$. In each round t; we choose a point $x_t \in \mathcal{X}$ and get to see the function value there, but only a perturbed result can be measured: $y_t = f(x_t) + \varepsilon_t$. Our goal is to estimate the value of $f(x)$ where x is an arbitrary point in $\mathcal{X}$.

For example, we might want to find locations of highest temperature in a building by sequentially activating sensors in a spatial network and regressing on their measurements. $\mathcal{X}$ consists of all sensor locations, $f(x)$ is the temperature at x, and sensor accuracy is quantified by the noise variance. Each activation draws battery power, so we want to sample from as few sensors as possible. Figure 1.1 from [12] specifically shows how to achieve this goal by a learning-based method. Basically, Gaussian Processes give both mean estimate and error estimate. Since we want to find locations of highest temperature, the algorithm samples new data at the location with the largest mean and the largest uncertainty which quantified by posterior variance respectively and then better fits the temperature function and gives a more precise error bound estimate in next iteration. If we further imagine the building to be a cold storage and the valuable targeted drug goes bad once the temperature exceeds a certain level, then what we need is not only an estimate but also an upper bound of the temperature at a certain point. In this kind of cases, we care more about the confidence interval especially the upper bound than the estimate itself as it can be used as a criteria of making decisions.



(a) *Temperature data* (b) *Iteration* t (c) *Iteration* $t+1$

Figure 1.1: (a) Example of temperature data collected by a network of 46 sensors at Intel Research Berkeley. (b,c) Two iterations of the GP-UCB algorithm. It samples points that are either uncertain (b) or have high posterior mean (c).

1.2 Scope of the Research Paper

The remainder of this research paper is organized as follows, error bounds for Gaussian Processes and kernel methods will be introduced in chapter 2. Necessary stability analysis Content will then be presented in chapter 3. Experiments including comparison of Bayesian Neural Networks and different approaches and doing stability analysis with learned dynamics is contained in chapter 4. Lastly, some personal perspectives and potential research topics will be provided in chapter 5.

Chapter 2

Learning Unknown Function

2.1 Problem Definition

Consider an unknown function $f : \mathcal{X} \to \mathbb{R}$. In order to reconstruct f, we collect a noised sample $\boldsymbol{y_T} = [y_1 \cdots y_T]^T$ at pairwise distinct positions $A_T = \{x_1, \ldots, x_T\}$. $y_t = f(x_t) + \varepsilon_t$ is the measurement result, where ε_t is some measurement noise and has different restrictions in different approaches. Next, we introduce the definition of kernel, Reproducing Kernel Hilbert Space and RKHS norm.

Definition 2.1.1(Kernel): A continuous function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a positive-definite kernel if it is symmetric and for any set of pairwise-distinct sites $D = \{x_1, \ldots, x_N\}$, with an arbitrary natural number N, it holds that $\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j k(x_i, x_j) > 0$ for any set of weighting constants $\alpha_1, \ldots, \alpha_N \in \mathbb{R} \backslash \{0\}$.

Although limiting our scope to positive-definite functions excludes some certain kernels, this class has already contained the most powerful and widely used ones such as the squared-exponential kernel and is enough for doing error bounds estimate.

Definition 2.1.2(Reproducing Kernel Hilbert Space): A RKHS $\mathcal{H}$ is a complete subspace of L_2. Each element $g \in \mathcal{H}$ is a map from $\mathcal{X}$ to $\mathbb{R}$ assuming the form of a weighted sum of kernels $g = \sum_{i \in \Omega_g} \alpha_i k(x_i, \cdot)$, where the index set Ω_g of g can possibly be countably infinite. $\mathcal{H}$ is equipped with the inner product $\langle g, f \rangle_k = \sum_{i \in \Omega_g} \sum_{j \in \Omega_f} \alpha_i \beta_j k(x_i, x_j)$ and the induced norm is $\|g\|_k := \sqrt{\langle g, g \rangle_k}$, measures the smoothness of g, where k refers to a symmetric, positive definite kernel. This inner product obeys the reproducing property: $\langle g(\cdot), k(x, \cdot) \rangle_k = g(x)$ for all $g \in \mathcal{H}$.

Notation: Denote $K \in \mathbb{R}^{T\times T}$ to be the constant matrix that has kernel evaluations at data locations as its elements, i.e., $k(x_i, x_j)$ at its i th row and j th column for $x_i, x_j \in A_T$. Moreover, $k_{\mathcal{X}x} : \mathcal{X} \to \mathbb{R}^T$ denotes the column vector function $x \mapsto \begin{bmatrix} k(x_1, x) & \dots & k(x_T, x) \end{bmatrix}^\top$, and $k_{x\mathcal{X}}$ simply represents its transpose.

2.1.1 RKHS Norm

Suppose that g has a finite expansion in terms of N_g kernel functions. Due to the reproducing property and the linearity of inner products, it holds that

$$\begin{aligned}\|g\|_k^2 &= \left\langle \sum_{i=1}^{N_g} \alpha_i k(x_i, \cdot), \sum_{i=1}^{N_g} \alpha_i k(x_i, \cdot) \right\rangle_k \\ &= \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} \alpha_i \alpha_j k(x_i, x_j) \\ &= \alpha^\top K \alpha\end{aligned}$$

where $\alpha := \begin{bmatrix} \alpha_1 & \dots & \alpha_{N_g} \end{bmatrix}^\top$ is the weighting vector. Note that here we know the actual kernel expansion of g, so that x_i and x_j are no longer data positions in A_T. In this case, K refers to the true kernel matrix but not the constant matrix that has kernel evaluations at data locations as its elements.

It's impossible to give any estimate without any assumptions made on the unknown ground truth f so that we add some restrictions on the smoothness of f here in assumption 1.

Assumption 1. Assume that the unknown ground-truth f lies in the RKHS $\mathcal{H}$ corresponding to a kernel k. Additionally, an upper bound for its RKHS norm $\|f\|_k \leq \Gamma$ is available.

2.1.2 RKHS Norm Estimate

All the results we will show later requires Assumption 1 which is an available upper bound of the RKHS norm of an unknown function. Therefore, how to get access to Γ is worthy of concern. Unfortunately, similar to estimating Lipschitz constant of an unknown function,

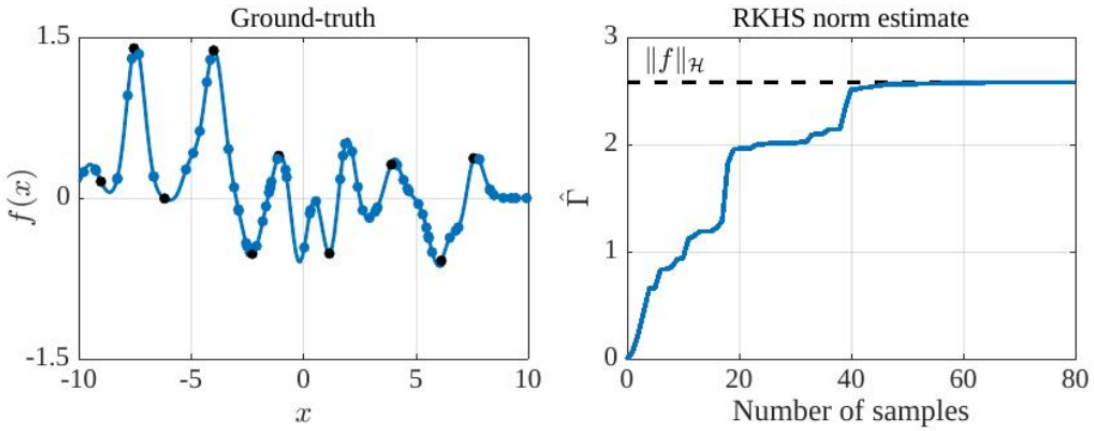


Figure 2.1: Estimating the RKHS norm using randomly sampled data (all circles). The quadratic form $\hat{\Gamma}$ for the random samples is shown on the right plot. If one sampled only the black subset of the data, the corresponding $\hat{\Gamma}$ would capture over 90% of the total complexity, i.e., $\hat{\Gamma}/\|f\|_{\mathcal{H}} > 0.9$.

it is not possible to get an upper estimate purely from measured data. However, as shown in [10], if our measurements are not disturbed by any noise, $\hat{\Gamma} := \sqrt{f_T^\top K^{-1} f_T} \leq \|f\|_k$ is an efficient lower estimate of the RKHS norm. We include Figure 2.1 from [10] here to illustrate the lower estimate is efficient.

2.2 Gaussian Processes Approach

In this approach, we generally use $\mathcal{GP}(0, k(x, x^{'}))$ as prior distribution over f, for the noisy sample described in Definition 2.1.1 with $\varepsilon_t \sim N(0, \sigma^2)$ i.i.d. Gaussian noise, the posterior over f is a GP distribution again, with mean $\mu_T(x)$, covariance $k_T(x, x')$, and variance $\sigma_T^2(x)$:

$$\mu_T(x) = k_{x\mathcal{X}} \left(K + \sigma^2 I\right)^{-1} \boldsymbol{y_T}$$
$$k_T(x, x') = k(x, x') - k_{x\mathcal{X}} \left(K + \sigma^2 I\right)^{-1} k_{\mathcal{X}x}$$
$$\sigma_T^2(x) = k_T(x, x)$$

There is a connection between the k norm and the k_T norm of f hiding behind the GP posterior covariance formula. By applying eigendecomposition, Woodbury identity and change of basis matrix, the explicit expression of this connection can be obtained:

$$\|f\|_{k_T}^2 = \|f\|_k^2 + \sigma^{-2} \sum_{t=1}^{T} f\left(x_t\right)^2 \quad \forall f \in \mathcal{H} \quad (1)$$

As a result, finite k norm implies finite k_T norm, so that $\mathcal{H}_k(\mathcal{X}) = \mathcal{H}_{k_T}(\mathcal{X})$ by definition. Then,

$$\begin{aligned} |\mu_T(x) - f(x)| &= \left|\langle \mu_T(\cdot), k_T(\cdot, x)\rangle_{k_T} - \langle f(\cdot), k_T(\cdot, x)\rangle_{k_T}\right| \\ &= |\left\langle \mu_T(\cdot) - f(\cdot), k_T(\cdot, x\right\rangle_{k_T}| \\ &\leq k_T(x,x)^{\frac{1}{2}} \|\mu_T - f\|_{k_T} \\ &= \sigma_T(x) \|\mu_T - f\|_{k_T} \qquad (2) \end{aligned}$$

The first line comes from the reproducing property of f in $\mathcal{H}_{k_T}$ as we have already shown $\mathcal{H}_k(\mathcal{X}) = \mathcal{H}_{k_T}(\mathcal{X})$. And the inequality is a direct use of the Cauchy–Schwarz inequality.

This means once we get an estimate to the k_T norm of $\mu_T(x) - f(x)$, an interval estimate of f which consists of $\sigma_T(x)$ and $\mu_T(x)$ can be derived in a few straightforward steps.

2.2.1 GPs Probabilistic Error Bounds

We first show the error bounds given in [12], other Gaussian Processes results basically rely on the proof in [12] and provide some improvements.

Theorem 2.2.1(GPs Probabilistic Error Bounds): Let $\delta \in (0,1)$. Assume that the noise variables ε_t are uniformly bounded by σ. Define

$$\beta_t = 2\Gamma^2 + 300\gamma_t \ln^3(t/\delta)$$

Then

$$\Pr\left\{\forall T, \forall x \in \mathcal{X}, |\mu_T(x) - f(x)| \leq \beta_{T+1}^{1/2} \sigma_T(x)\right\} \geq 1 - \delta$$

where $\gamma_t := \max_{A \subset \mathcal{X}:|A|=t} \frac{1}{2} \sum_{s=1}^{t} \log\left(1 + \sigma^{-2}\sigma_{s-1}^2\left(x_s\right)\right)$ is the maximum information gain, and T is the number of measurements.

In practice, the maximum information gain can be obtained directly from the choice of kernel function without actually calculating the closed form solution given in Theorem 2.1.1, so that the computational cost won't be high.

2.2.2 GPs Deterministic Error Bounds

We have already shown that the k_T norm of f can be expressed by the k norm of f plus some extra terms in (1). And the k inner product has been defined in section 2.1. The strategy of estimating the k_T norm of $\mu_T(x) - f(x)$ would be first transform it to the k norm and do some direct calculation. Specifically, denote $\boldsymbol{\alpha_T} = (K + \sigma^2 I)^{-1} \boldsymbol{y_T}$ and $\boldsymbol{f_T} = f(x_t) \in R^T$, then $\mu_T(x) = k_{x\mathcal{X}}\boldsymbol{\alpha_T}$, $\langle \mu_T, f \rangle_k = \langle k_{x\mathcal{X}}\boldsymbol{\alpha_T}, f \rangle_k = \boldsymbol{f_T^\top}\boldsymbol{\alpha_T}$ by reproducing property, $\|\mu_T\|_k^2 = \boldsymbol{\alpha_T^\top} K \boldsymbol{\alpha_T} = \boldsymbol{y_T^\top}\boldsymbol{\alpha_T} - \sigma^2 \|\boldsymbol{\alpha_T}\|^2$ which follows the same calculation shown in sec 2.1.1, and for $t \leq T, \mu_T(x_t) = \boldsymbol{\delta_t^\top} K (K + \sigma^2 I)^{-1} \boldsymbol{y_T} = y_t - \sigma^2 \alpha_t$. Finally,

$$\begin{aligned}
\|\mu_T - f\|_{k_T}^2 &= \|\mu_T - f\|_k^2 + \sigma^{-2} \sum_{t \leq T} (\mu_T(x_t) - f(x_t))^2 \\
&= \|f\|_k^2 - 2\boldsymbol{f_T^\top}\boldsymbol{\alpha_T} + \boldsymbol{y_T^\top}\boldsymbol{\alpha_T} - \sigma^2 \|\boldsymbol{\alpha_T}\|^2 + \sigma^{-2} \sum_{t=1}^{T} \left(\varepsilon_t - \sigma^2 \alpha_t\right)^2 \\
&= \|f\|_k^2 - \boldsymbol{y_T^\top} \left(K + \sigma^2 I\right)^{-1} \boldsymbol{y_T} + \sigma^{-2} \|\boldsymbol{\varepsilon_T}\|^2
\end{aligned}$$

Here comes our GPs deterministic error bounds estimate:

Theorem 2.2.2(GPs Deterministic Error Bounds): For all $T \in \mathbb{N}$,

$$f(x) \in \left[\mu_T(x) \pm \sigma_T(x) \sqrt{\Gamma^2 - \boldsymbol{y_T^\top} (K + \sigma^2 I)^{-1} \boldsymbol{y_T} + \sigma^{-2} \|\boldsymbol{\varepsilon_T}\|^2}\right]$$

if we further assume that the noise variables ε_t are uniformly bounded by σ, then the result becomes

$$f(x) \in \left[\mu_T(x) \pm \sigma_T(x) \sqrt{\Gamma^2 - \boldsymbol{y_T^\top} (K + \sigma^2 I)^{-1} \boldsymbol{y_T} + T}\right] \quad (3)$$

In fact, Theorem 2.2.2 is an intermediate product when deriving Theorem 2.1.1. In the subsequent derivation process of Theorem 1, enlargement to the estimated interval is needed so that the probabilistic error bound is potentially more conservative than the deterministic one. This is a very weird result since we sacrifice safety but the result can be even more conservative.

2.3 Kernel Methods Approach

The main idea for kernel methods approach is finding the interpolation $s : \mathcal{X} \to \mathbb{R}$, which is analogous with the mean estimate in GPs approach by solving some optimization problems.

In this approach, we no longer require the noise to follow any particular distribution but bounded only with an available bound $\bar{\delta}$. Dropping the Gaussian noise assumption won't influence any theoretical guarantees as we don't need any result from GPs in the derivation. Thanks to the nonparametric representer theorem [11], any minimizers of the optimization problems have the form of a weighted sum of kernels centered at the data locations,

$$s(x) = \sum_{t=1}^{T} \alpha_t k\left(x_t, x\right) = \alpha^\top k_{\mathcal{X}x}$$

The weighting vector α can be decided by doing optimizations over $\mathcal{H}$, and a deterministic error bound estimate can be constructed by triangle inequality and Woodbury's matrix identity. And each kind of optimization problems of determining the weighting coefficients provides a regression analysis.

In order to simplify our results, we first define the power function.:

Definition 2.3.1(Power function): The power function is the real-valued map

$$P(x) = \sqrt{k(x,x) - k_{x\mathcal{X}} K^{-1} k_{\mathcal{X}x}}$$

which is the same as the noise free GPs posterior standard derivation so that it has two main properties:

$$P(x) \geq 0, \forall x \in \mathcal{X}$$

$$P(x_t) = 0, \forall x_t \in A_T, \text{ i.e.} P(x) = 0 \text{ at the data locations.}$$

2.3.1 Noise Free Case

When the measurements are not disturbed any noise, the optimization

$$\begin{aligned} \bar{s} = & \arg\min_{s \in \mathcal{H}} \|s\|_k^2 \\ & s.t. \quad \bar{s}(x_t) = f(x_t) \\ & \qquad \forall t = 1, \ldots, T \end{aligned}$$

gives:

$$\bar{s}(x) = \boldsymbol{f_T}^\top K^{-1} k_{\mathcal{X}x}$$

and

$$\|\bar{s}\|_k{}^2 = \boldsymbol{f_T}^\top K^{-1} \boldsymbol{f_T}$$

the interpolation $\bar{s}$ admits a deterministic error bounds:

$$|\bar{s}(x) - f(x)| \leq P(x)\sqrt{\Gamma^2 - \|\bar{s}\|_k^2}$$

Proof: If we argument the original data set by a fixed query point x_{T+1} and denote by $\bar{s}_+$ the function interpolating all measurements $\boldsymbol{f_T}$ and the unknown value $f_x := f(x_{T+1})$, then

$$\begin{aligned}\|\bar{s}_+\|_k^2 =& \begin{bmatrix} \boldsymbol{f_T} \\ f_x \end{bmatrix}^\top \begin{bmatrix} K & k_{\mathcal{X}x} \\ k_{x\mathcal{X}} & k(x,x) \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{f_T} \\ f_x \end{bmatrix} \\ =& \|\bar{s}\|_k^2 + P^{-2}(x)\left(\bar{s}(x) - f_x\right)^2 \\ \leq& \Gamma^2\end{aligned}$$

which means that the RKHS norm is non-decreasing as the number of constraints increase. Hence, $\|\bar{s}_+\|_k{}^2 \leq \|f\|_k{}^2 \leq \Gamma^2$ as f is the solution to the optimization problem when an infinite number of constrains are imposed.

2.3.2 Kernel Ridge Regression Analysis

For KRR analysis, we need the solve the optimization problem:

$$s^* = \arg\min_{s\in\mathcal{H}} \frac{1}{T}\sum_{n=1}^{T}\left(y_n - s\left(x_n\right)\right)^2 + \lambda\|s\|_k{}^2$$

which aims to find a balance between fitting the data and making the interpolation smooth. Luckily, it has a closed form solution: $s^*(x) = \alpha^{*\top} k_{\mathcal{X}x}$ where $\alpha^* = (K + N\lambda I)^{-1}\boldsymbol{y_T}$ is the optimal weights.

Lemma 1. Let $\bar{s}(x) = \boldsymbol{f_T}^\top K^{-1} k_{\mathcal{X}x}$ be the model interpolating the noise-free values $\boldsymbol{f_T}$, and $\tilde{s}(x) = \boldsymbol{y_T}^\top K^{-1} k_{\mathcal{X}x}$ the model interpolating the noisy values $\boldsymbol{y_T}$. Then $\nabla \leq \|\tilde{s}\|_k{}^2 - \|\bar{s}\|_k{}^2 \leq \Delta$ where Δ denotes the maximum and ∇ the minimum of $\left(-\boldsymbol{\delta}^\top K^{-1}\boldsymbol{\delta} + 2\boldsymbol{y}^\top K^{-1}\boldsymbol{\delta}\right)$ subject to $|\delta| \leq \bar{\delta}$.

After maximizing Δ under bounded noise, a deterministic error bounds based on KRR analysis can be obtained.

Theorem 2.3.2(KRR Error Bounds): Let T be the number of data-points, $\bar{\boldsymbol{\delta}} \in \mathbb{R}^N_{>0}$

the noise bound, and λ the regularization constant. The KRR model s^* admits the error bounds

$$|s^*(x) - f(x)| \leq P(x)\sqrt{\Gamma^2 + \Delta - \|\tilde{s}\|_k^{\,2}} + \bar{\boldsymbol{\delta}}^\top \left|K^{-1}k_{\mathcal{X}x}\right| + \left|\boldsymbol{y}^\top \left(K + \frac{1}{T\lambda}KK\right)^{-1} k_{\mathcal{X}x}\right|$$

for any $x \in \mathcal{X}$, where f is the unknown ground-truth, $\Delta = \max_{|\delta|\leq\bar{\delta}}(-\boldsymbol{\delta}^\top K^{-1}\boldsymbol{\delta} + 2\boldsymbol{y}^\top K^{-1}\boldsymbol{\delta})$, and $\|\tilde{s}\|_k^{\,2} = \boldsymbol{y}_{\boldsymbol{T}}^\top K^{-1}\boldsymbol{y}_{\boldsymbol{T}}$.

2.3.3 ε-Support Vector Regression Analysis

For SVR analysis, we need the solve the optimization problem:

$$s^* = \arg\min_{s\in\mathcal{H}} \|s\|_k^{\,2} \quad \text{s.t.} \quad |s(x_t) - y_t| \leq \bar{\delta}_t \quad \forall t = 1, \ldots, T$$

which aims to find the smoothest interpolation within the available margin, and $\bar{\delta}_t$ is the tth element of the $\bar{\delta}$ vector. Unfortunately, it doesn't have a closed form solution, so that the optimizer need to be found by solving the simple quadratic program

$$\alpha^* = \arg\min_{\alpha\in\mathbb{R}^T} \alpha^\top K\alpha \quad \text{s.t.} \quad |K\alpha - \boldsymbol{y}_{\boldsymbol{T}}| \leq \bar{\boldsymbol{\delta}}.$$

By setting $s^*(x) = \alpha^{*\top} k_{\mathcal{X}x}$, where the attained values at the data sites are denoted by $d_t := s^*(x_t), t = 1, \ldots, T$, with $\boldsymbol{d} = \begin{bmatrix} d_1 & \ldots & d_T \end{bmatrix}^\top = K\alpha^*$, we are able to bound the unknown ground-truth f in terms of some known values and the newly calculated vector d.

Theorem 2.3.3(SVR Error Bounds): Let $\bar{\boldsymbol{\delta}} \in \mathbb{R}^T_{>0}$ be the noise bound vector. The SVR model s^* admits the error bound $|s^*(x) - f(x)| \leq P(x)\sqrt{\Gamma^2 - \|s^*\|_k^{\,2}} + \bar{\boldsymbol{\delta}}^\top |K^{-1}k_{\mathcal{X}x}| + \left|(\boldsymbol{d} - \boldsymbol{y}_{\boldsymbol{T}})^\top K^{-1}k_{\mathcal{X}_x}\right|$ for all $x \in \mathcal{X}$, where f is the unknown ground-truth and $\|s^*\|_k^{\,2} = \boldsymbol{d}^\top K^{-1}\boldsymbol{d}$.

Chapter 3

Stability Analysis

In the previous chapter we have shown how to obtain interval estimates of unknown functions through Gaussian Processes and Kernel Methods. However, it is still necessary to demonstrate that these interval estimates are not too conservative to be meaningful. Otherwise, $(-\infty, \infty)$ is always a safer estimate, but it is obviously meaningless.

To demonstrate our error bounds estimates are meaningful, we will show its application in stability analysis. Our strategy is to do stability analysis on both the actual dynamic and the learned dynamic, and calculate their Region of Attraction(ROA). We expect the ROA obtained through the actual dynamic contains the one obtained through the learned dynamic almost everywhere and they are similar in size.

3.1 Lyapunov Theory

In this section, we will introduce necessary Lyapunov theory content. For ease of notation, we use the definition from [3] and [15] in this research paper. For more on stability analysis and Lyapunov theory, please read the Book [7].

Definition 3.1.1 (Controlled Dynamical Systems). An n-dimensional controlled dynamical system is

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f_u(x), \quad x(0) = x_0 \qquad (3)$$

where $f_u : \mathcal{D} \to \mathbb{R}^n$ is a Lipschitz-continuous vector field, and $\mathcal{D} \subseteq \mathbb{R}^n$ is an open set with $0 \in \mathcal{D}$ that defines the state space of the system. Each $x(t) \in \mathcal{D}$ is a state vector. The

feedback control is defined by a continuous function $u : \mathbb{R}^n \to \mathbb{R}^m$, used as a component in the full dynamics f_u.

Definition 3.1.2 (Asymptotic Stability). We say that system of (3) is stable at the origin if for any $\varepsilon \in \mathbb{R}^+$, there exists $\delta(\varepsilon) \in \mathbb{R}^+$such that if $\|x(0)\| < \delta$ then $\|x(t)\| < \varepsilon$ for all $t \geq 0$. The system is asymptotically stable at the origin if it is stable and also $\lim_{t\to\infty} \|x(t)\| = 0$ for all $\|x(0)\| < \delta$.

Definition 3.1.3 (Lie Derivatives). The Lie derivative of a continuously differentiable scalar function $V : \mathcal{D} \to \mathbb{R}$ over a vector field f_u is defined as

$$\mathrm{L}_{f_u} V(x) = \sum_{i=1}^{n} \frac{\partial V}{\partial x_i} \frac{\mathrm{d}x_i}{\mathrm{d}t} = \sum_{i=1}^{n} \frac{\partial V}{\partial x_i} \left[f_u\right]_i (x)$$

It measures the rate of change of V along the direction of the system dynamics.

Proposition 3.1.4 (Lyapunov Functions for Asymptotic Stability). Consider a controlled system (3) with equilibrium at the origin, i.e., $f_u(0) = 0$. Suppose there exists a continuously differentiable function $V : \mathcal{D} \to \mathbb{R}$ that satisfies the following conditions:

$$V(0) = 0, \text{ and, } \forall x \in \mathcal{D}\backslash\{0\}, V(x) > 0 \text{ and } \mathrm{L}_{f_u} V(x) < 0.$$

Then, the system is asymptotically stable at the origin and V is called a Lyapunov function.

Definition 3.1.5 (Forward Invariance). A set $\Omega \subset \mathbb{R}^n$ is said to be forward invariant for system (3) if $x_0 \in \Omega$ implies that $x(t) \in \Omega$ for all $t \geq 0$.

Definition 3.1.6 (Region of Attraction). For a closed forward invariant set A that is uniformly asymptotically stable (UAS), the region of attraction is the set of initial conditions in $\mathcal{D}$ such that the solution for the closed-loop system (3) is defined for all $t \geq 0$ and $\|x(t)\|_A \to 0$ as $t \to \infty$.
Remark 1. Any set satisfying Definition 3.1.6 is called a region of attraction. The ROA is the largest set contained in $\mathcal{D}$ satisfying Definition 3.1.6.

Theorem 3.1.7 (Sufficient Condition for UAS property). Consider the closed-loop nonlinear system (3). Let $A \subset \mathcal{D}$ be a compact invariant set of this system. Suppose there exists a continuously differentiable function $V : \mathcal{D} \to \mathbb{R}$ that is positive definite with respect to A, i.e.,

$$V(x) = 0 \quad \forall x \in A \text{ and } V(x) > 0 \quad \forall x \in \mathcal{D} \backslash A,$$

and the lie derivative is negative definite with respect to A, i.e.

$$\nabla_f V(x) < 0 \forall x \in \mathcal{D} \backslash A$$

Then, A is UAS for the system.

Lemma 2. (Region of Attraction with Lyapunov Functions). Suppose that V satisfies the conditions in Theorem 3.1.7, denote

$$V^c := \{x \in \mathcal{D} \mid V(x) \leq c\}.$$

For every $c > 0, V^c$ is a region of attraction for the closed-loop system (3).

3.2 Safe Learning

In the previous section, we have introduced the concept of asymptotically stable and Lyapunov theory in general. Next, we will apply Lyapunov theory to a relatively specific dynamic system and see how it works with GPs in doing stability analysis. The follow setting and derivation comes from [1] and we have revised its mistake.

Consider a nonlinear, continuous-time system,

$$\dot{x}(t) = \underbrace{f(x(t), u(t))}_{\text{a prior model}} + \underbrace{g(x(t), u(t))}_{\text{unknown model}}, \qquad (4)$$

where $x(t) \in \mathcal{X} \subseteq \mathbb{R}^q$ is the state at time t within a connected set $\mathcal{X}$ and $u(t) \in \mathcal{U} \subseteq \mathbb{R}^p$ is the control input. We further assume a control policy $u = \pi(x)$ is given, which has been designed for the prior model, $f(\cdot)$. The resulting closed-loop dynamics are denoted by $f_\pi(x) := f(x, \pi(x))$ and $g_\pi(x) := g(x, \pi(x))$. Our goal is to estimate the ROA of (4) under the control policy, $\pi(x)$, based on the measured data. Without loss of generality, we can assume the origin to be an equilibrium point of (4).

Assumption 2. The origin is an equilibrium point of (4) with $f_\pi(0) = g_\pi(0) = 0$ and a initial safe set around the origin is given.

In order to do stability analysis, we require $g_\pi(x)$ satisfies Assumption 1 and $(A_T, \boldsymbol{y_T})$ to be its measurements. We also need more assumptions on the prior model and Lyapunov function.

Assumption 3. The prior model $f_\pi(x)$ is Lipschitz continuous with Lipschitz constant L_f and bounded in $\mathcal{X}$ by B_f.

Assumption 4. A fixed, two-times continuously differentiable Lyapunov function $V(x)$ is given.

Then by Lyapunov stability theory in [7], we have:

Lemma 3: The origin of the dynamics in (4) is asymptotically stable within a level set, $\mathcal{V}(c) = \{x \in \mathcal{X} \mid V(x) \leq c\}$ with $c \in \mathbb{R}_{>0}$, if, for all $x \in \mathcal{V}(c)$,

$$\dot{V}(x) = \frac{\partial V(x)}{\partial x}\left(g_\pi(x) + f_\pi(x)\right) < 0$$

and $\mathcal{V}(c_{max})$ would be our estimated ROA.

It is impossible to evaluate $\dot{V}(x)$ everywhere in $\mathcal{X}$ so that we use discretization techniques here. And the "Lipschitz" continuity and boundedness of functions in RKHS will allow us to realize stability analysis by only evaluate finite number of grid points. In fact, functions in RKHS are not Lipschitz continuous but has a property analogous to Lipschitz continuity. We use quotation marks here to refer to this similar property.

Lemma 3. $\forall f \in \mathcal{H}$,

$$|f(x_1) - f(x_2)| \leq l_f \sqrt{||x_1 - x_2||_\infty}$$

where $l_f = 2\Gamma^2 ||\frac{\partial k}{\partial x}||_\infty$.
Proof. By applying the reproducing property inversely and the Cauchy-Schwarz inequality, we have $|f(x_1) - f(x_2)|^2 \leq \|f\|_k^2 \{k(x_1, x_1) - 2k(x_1, x_2) + k(x_2, x_2)\}$(Lemma 4.28 in [13]). Then applying the Mean Value Theorem to both $(k(x_1, x_1) - k(x_1, x_2))$ and $(k(x_2, x_2) - k(x_1, x_2))$ completes the proof.

Lemma 4. $g_\pi(x)$ is "Lipschitz" continuous with Lipschitz constant L_g and bounded by $\Gamma \|k\|_\infty$.
Proof. Boundedness by Lemma 4.28 in [13], and "Lipschitz" continuity by lemma 3.

Lemma 5. The function $\dot{V}(x)$ is "Lipschitz" continuous with "Lipschitz" constants L_1 and L_2.

Proof.

$$\begin{aligned}\left|\dot{V}(x)-\dot{V}\left(x'\right)\right| &= \left|\frac{\partial V(x)}{\partial x}\left(f_\pi(x)+g_\pi(x)\right)-\frac{\partial V\left(x'\right)}{\partial x}\left(f_\pi\left(x'\right)+g_\pi\left(x'\right)\right)\right| \\ &\leq \left|f_\pi(x)+g_\pi(x)\right|\left|\frac{\partial V(x)}{\partial x}-\frac{\partial V\left(x'\right)}{\partial x}\right| \\ &\quad+\left|\frac{\partial V\left(x'\right)}{\partial x}\right|\left|f_\pi(x)+g_\pi(x)-f_\pi\left(x'\right)-g_\pi\left(x'\right)\right|, \\ &\leq \left(B_f+\Gamma\|k\|_\infty\right)L_{\partial V}\left|x-x'\right|+L_V L_f\left|x-x'\right|+L_V L_g\sqrt{\left|x-x'\right|} \\ &:= L_1\left|x-x'\right|+L_2\sqrt{\left|x-x'\right|}\end{aligned}$$

where $L_V=\left\|\frac{\partial V(x)}{\partial x}\right\|_\infty$ and $L_{\partial V}=\left\|\frac{\partial^2 V(x)}{\partial x^2}\right\|_\infty$ are the Lipschitz constants of V and its first derivative.

Gaussian Processes regression gives us the posterior mean $\mu_T(x)$and variance $\sigma_T(x)$ of the unknown dynamics $g_\pi(x)$ for all $x\in\mathcal{X}$. As $\dot{V}(x)$ is affine in $g_\pi(x)$, $\dot{V}(x)$ is also a GP with mean $\mu_T(x)$ and variance $\sigma_T(x)$, where

$$\mu_{T,\dot{V}}(x)=\frac{\partial V(x)}{\partial x}(\mu_T(x)+f_\pi(x))$$

$$\sigma_{T,\dot{V}}(x)=|\frac{\partial V(x)}{\partial x}|\sigma_T(x)$$

Lemma 6. Let $\mathcal{X}_\tau\subset\mathcal{X}$ be a discretization of $\mathcal{X}$ with $|x-[x]_\tau|\leq\tau/2$ for all $x\in\mathcal{X}$. Here, $[x]_\tau$ denotes the closest point in $\mathcal{X}_\tau$ to $x\in\mathcal{X}$. Choosing β_T according to Theorem 2.2.1, the following holds with probability at least $(1-\delta)$ for all $x\in\mathcal{X}$ and all $T\geq 1$:

$$\left|\dot{V}(x)-\mu_{\dot{V},T-1}\left([x]_\tau\right)\right|\leq\beta_T^{1/2}\sigma_{\dot{V},T-1}\left([x]_\tau\right)+L\tau.$$

Theorem 3.2.1. With a discretization of $\mathcal{X},\mathcal{X}_\tau$, according to Lemma 6 and with β_{T+1} according to Theorem 2.2.1, the origin of (4) is asymptotically stable within $\mathcal{V}(c)$ for some $c>0$ with probability at least $(1-\delta)$ if, for all $x\in\mathcal{V}(c)\cap\mathcal{X}_\tau$,

$$u_n(x):=\mu_{\dot{V},T-1}(x)+\beta_T^{1/2}\sigma_{\dot{V},T-1}(x)<-L\tau.$$

Algorithm 1: Safe ROA exploration

Inputs: Domain $\mathcal{X}$ and discretization with τ, $\mathcal{X}_\tau$
GP prior $k(x, x')$
Initial safe set $\mathcal{S}_0 \subseteq \mathcal{X}$

for $n = 1, \ldots$ **do**
  $c_n \leftarrow \underset{c>0}{\text{argmax}}\ c$, subject to
    $u_n(x) < -L\tau$, for all $x \in \mathcal{V}(c) \cap \mathcal{X}_\tau$
  $\mathcal{S}_n \leftarrow \mathcal{S}_0 \cup \mathcal{V}(c_n)$
  $x_n \leftarrow \text{argmax}_{x \in \mathcal{S}_n}\ \sigma_{n-1}(x)$
  Update GP with measurement of $\hat{g}_\pi(x_n)$

Figure 3.1: Algorithm 1.

Proof. This is obtained through synthesising Theorem 2.2.1, "Lipschitz" continuity and boundedness of $f_\pi(x)$ and $g_\pi(x)$, and discretization.

Figure 3.1 from [1] shows how to obtain ROA through GPs. Note that GPs error bounds estimate is applied to $\dot{V}(x)$ in [1]. This requires $\dot{V}(x)$ is also a GP. In fact, it is sufficient to only apply GPs error bounds estimate to $g_\pi(x)$. This is significant because it provides theoretical guarantees of doing stability analysis through kernel methods error bounds estimate.

Proposition 3.2.1 Consider $|f(x) - s(x)| \leq \beta(x)$, where $s(x)$ is a kernel methods interpolation and $\beta(x) \geq 0$ to be its error bound. Then,

$$\begin{aligned}\left|\frac{\partial V(x)}{\partial x} f(x) - \frac{\partial V(x)}{\partial x} s(x)\right| &\leq \left|\frac{\partial V(x)}{\partial x}\right| \cdot |f(x) - s(x)| \\ &\leq \left|\frac{\partial V(x)}{\partial x}\right| \cdot \beta(x)\end{aligned}$$

Therefore,

$$\frac{\partial V(x)}{\partial x} f(x) \leq \frac{\partial V(x)}{\partial x} s(x) + |\frac{\partial V(x)}{\partial x}| \beta(x)$$

The above proof shows that an upper bound of $\dot{V}(x)$ can be obtained without requiring $\dot{V}(x)$ is a GPs. As a result, Lyapunov methods can also be used together with kernel methods.

Chapter 4

Experiments

Three examples are presented here to illustrate the performance of error bounds estimate. First we compare the deterministic GPs, KRR and SVP approaches to one another, and to the Bayesian Neural Networks(BNNs) error bounds proposed in [2]. The second and third examples follow the same framework given in [1]. We expected to demonstrate our error bounds estimate is safe enough and not too conservative to be meaningful by showing the ROA obtained through the actual dynamic contains the one obtained through the learned dynamic almost everywhere and they are similar in size. In the second experiment, we successfully verified that doing stability analysis for one-dimensional systems with error bounds estimate is feasible. However, we could not get the expected experimental results to verify the feasibility of the theory in two-dimensional systems.

4.1 Bayesian Neural Networks

Neural network is a commonly used and effective method for fitting unknown functions. According to the well-known universal approximation theorem [9], a feed forward neural network with a single hidden layer containing a sufficient number of neurons (units) can approximate any continuous function to arbitrary accuracy, given appropriate activation functions and under certain conditions. However, traditional neural networks can only provide point estimates. Bayesian Neural Networks (BNNs) are a probabilistic extension of traditional artificial neural networks, incorporating Bayesian inference techniques into the learning process. Unlike standard neural networks that produce point estimates of the weight, BNNs provide a probability distribution of it. So that it can also output a confidence interval of at any location,

In a Bayesian Neural Network, each weight and bias parameter is treated as a random variable with a prior distribution, representing our initial belief about the parameter's value. As we observe data, the posterior distribution of these parameters is updated using Bayes' theorem, combining prior beliefs with the likelihood of the data given the parameters. This process results in a more refined and data-informed estimate of the parameters.

4.2 A Comparison Among Different Approaches

First, we use different approaches to fit a kernel expansion whose exact RKHS norm can be calculated by the formula given in section 2.1.1. Most of the settings in the experiment are the same as in [9]. Specifically, we use the same squared-exponential kernel

$$k\left(x, x_n\right) = \exp\left(-\frac{\left\|x - x_n\right\|_2^2}{2\ell^2}\right)$$

with lengthscale $\ell = 0.707$, to fit the same kernel expansion $f(x) = -k(x, 0) + 3.5k(x, 2) + 1.6k(x, 3) + 6k(x, 5)$, whose exact RKHS norm is 7.49 and $\Gamma = 9$ is used in the experiments. Figure 4.1 from [9] and 4.2 show the error bounds estimate from all these approaches. Clearly, KRR has a quite good performance but relies a well-chosen hyperparameter λ. In addition, BNNs do not have strong theoretical guarantees like GPs or kernel methods, but just more interpretable than ordinary NNs. It provides interval estimates that are not as conservative as GPs, but are also significantly less secure and requires most measurements. Table 4.1 depicts the pros and cons of these methods of fitting unknown functions. In general, NNs is a commonly used and effective method for fitting unknown functions, and BNNs, as a variant that can provide interval estimates, has stronger interpretability. Compared with GPs and kernel methods, it is more suitable when the training set is large or Γ is unknown and difficult to estimate, for example, when the unknown function is quite complex. In contrast, GPs and kernel methods are more suitable for safety-critical scenario and when the training set is small since the existence of matrix inversion operations in the result will kill the entire algorithm if the kernel matrix is large.

4.3 Issues with Truncated Gaussian Noise

By truncating the i.i.d. Gaussian Noise by its standard deviation, theorem 2.2.2 provides a deterministic error bounds estimate. However, this truncation can sometimes lead to

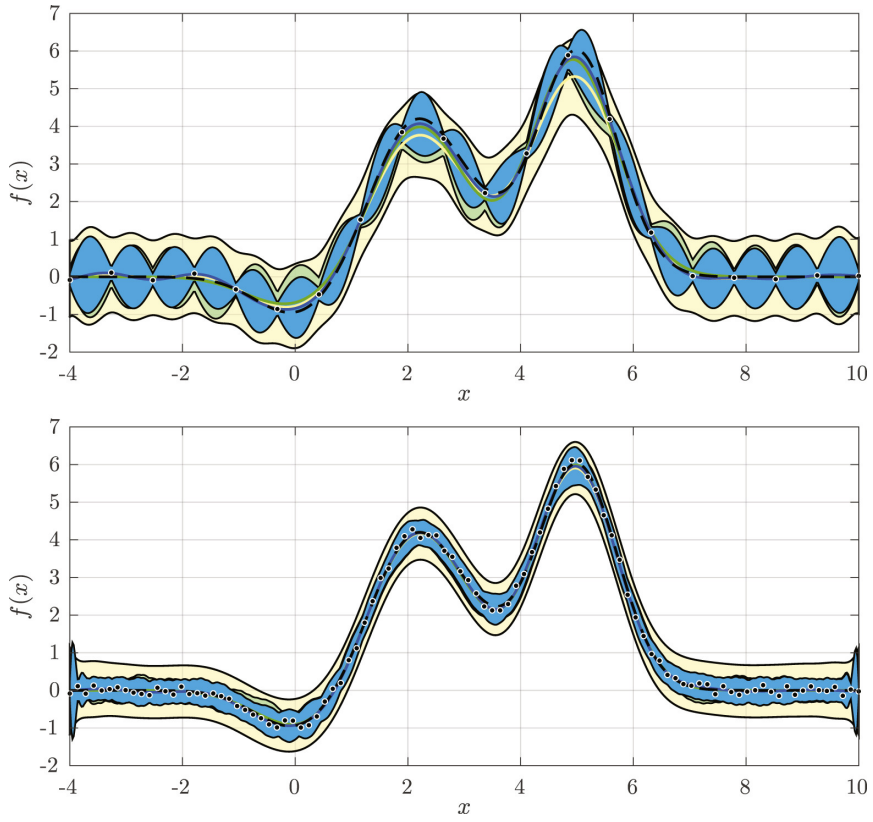


Figure 4.1: Ground-truth (- -), KRR (blue), SVR (green), and GPs (yellow). The error bounds are depicted using the same colors of their respective models, and were computed for $N = 20$ (top) and $N = 100$ (bottom) samples. The noisy data-points are shown as black circles.

	GPs	Kernel Methods	BNNs
Theoretical Guarantees	high	high	mid
Interpretability	high	high	mid
Conservatism	high	mid	low
Data Dependencies	low	low	high
Additional Needs	Γ	Γ and λ	None

Table 4.1: Comparison of different approaches.

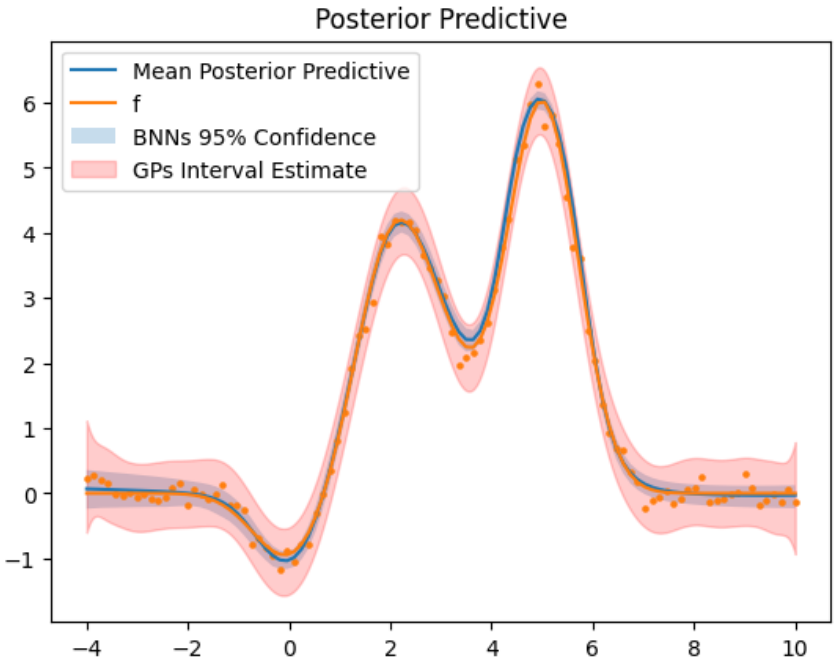


Figure 4.2: Comparison of BNNs and GPs

disastrous results. For example, in our numerical experiments, the parts under square root in (3) might be negative.

In fact, when T is greater than 20, $\Pr\left\{\sigma^{-2}\left\|\varepsilon_T\right\|^2 \leq T\right\}$ is only slightly larger than 0.5. To avoid self-contradictory assumptions and disastrous results, we propose to use chi-square distribution techniques to deal with the noise term.

Since $\varepsilon_t \sim N\left(0, \sigma^2\right)$ are i.i.d. Gaussian noise, it can be converted to the standard normal distribution, i.e: $\varepsilon_t/\sigma \sim N\left(0, 1^2\right)$. Then $\sigma^{-2}\left\|\varepsilon_T\right\|^2 \sim \mathcal{X}^2(T)$ as it is the summation of the squares of T independent standard normal random variables. Hence we can conclude that:

Theorem 4.3.1: Let $\delta \in (0,1)$. Define

$$\beta_t = \Gamma^2 - \boldsymbol{y}_{\boldsymbol{T}}^{\top}\left(K + \sigma^2 I\right)^{-1}\boldsymbol{y}_{\boldsymbol{T}} + g_T(\delta)$$

Then

$$\Pr\left\{\forall T, \forall x \in \mathcal{X}, \left|\mu_T(x) - f(x)\right| \leq \beta_T^{1/2}\sigma_T(x)\right\} \geq 1 - \delta$$

where $g_T(\delta)$ is the least horizontal coordinate of a random variable $Z \sim \mathcal{X}^2(T)$ to have $\Pr\left\{Z \leq g_T(\delta)\right\} \geq 1 - \delta$.

In fact, the safety of our proposed error bounds is significantly higher than $1 - \delta$ as we have scaled in (2) and Γ is an upper bound of $\|f\|_k$. This is also why (4) still remains a certain level of safety. By looking at the chi-square distribution table, we can see that when T is greater than 20, $g_T(0.5)$ is only slightly smaller than T, which means $\Pr\left\{\sigma^{-2}\left\|\varepsilon_T\right\|^2 \leq T\right\}$ is only slightly larger than 0.5.

4.4 One Dimensional Stability Analysis

In this experiment, we demonstrate the introduced error bounds estimate is not too conservative to be meaningful by implementing it to a one dimensional system. Specifically, we set the dynamics to be $\dot{x} = -0.25x + f(x)$ and the unknown ground truth f is chosen to be $f(x) = k(x, -0.5) - k(x, 0.5) + 6k(x, 5)$, where k refers to the same kernel in the previous experiment. The exact RKHS norm of f is 6.1045 so that Γ is chosen to be $\sqrt{40}$. We follow the strategy in [1], but use the error bounds given in Theorem 2.2.2 and 2.3.2 and no longer set a fixed fake $\beta = 2$.

k	Probability Content, p, between χ^2 and $+\infty$															
	0.995	0.99	0.975	0.95	0.9	0.75	0.5	0.25	0.1	0.05	0.025	0.01	0.005	0.002	0.001
1	3.927e-5	1.570e-4	9.820e-4	0.00393	0.0157	0.102	0.455	1.323	2.706	3.841	5.024	6.635	7.879	9.550	10.828
2	0.0100	0.0201	0.0506	0.103	0.211	0.575	1.386	2.773	4.605	5.991	7.378	9.210	10.597	12.429	13.816
3	0.0717	0.115	0.216	0.352	0.584	1.213	2.366	4.108	6.251	7.815	9.348	11.345	12.838	14.796	16.266
4	0.207	0.297	0.484	0.711	1.064	1.923	3.357	5.385	7.779	9.488	11.143	13.277	14.860	16.924	18.467
5	0.412	0.554	0.831	1.145	1.610	2.675	4.351	6.626	9.236	11.070	12.833	15.086	16.750	18.907	20.515
6	0.676	0.872	1.237	1.635	2.204	3.455	5.348	7.841	10.645	12.592	14.449	16.812	18.548	20.791	22.458
7	0.989	1.239	1.690	2.167	2.833	4.255	6.346	9.037	12.017	14.067	16.013	18.475	20.278	22.601	24.322
8	1.344	1.646	2.180	2.733	3.490	5.071	7.344	10.219	13.362	15.507	17.535	20.090	21.955	24.352	26.124
9	1.735	2.088	2.700	3.325	4.168	5.899	8.343	11.389	14.684	16.919	19.023	21.666	23.589	26.056	27.877
10	2.156	2.558	3.247	3.940	4.865	6.737	9.342	12.549	15.987	18.307	20.483	23.209	25.188	27.722	29.588
11	2.603	3.053	3.816	4.575	5.578	7.584	10.341	13.701	17.275	19.675	21.920	24.725	26.757	29.354	31.264
12	3.074	3.571	4.404	5.226	6.304	8.438	11.340	14.845	18.549	21.026	23.337	26.217	28.300	30.957	32.909
13	3.565	4.107	5.009	5.892	7.042	9.299	12.340	15.984	19.812	22.362	24.736	27.688	29.819	32.535	34.528
14	4.075	4.660	5.629	6.571	7.790	10.165	13.339	17.117	21.064	23.685	26.119	29.141	31.319	34.091	36.123
15	4.601	5.229	6.262	7.261	8.547	11.037	14.339	18.245	22.307	24.996	27.488	30.578	32.801	35.628	37.697
16	5.142	5.812	6.908	7.962	9.312	11.912	15.338	19.369	23.542	26.296	28.845	32.000	34.267	37.146	39.252
17	5.697	6.408	7.564	8.672	10.085	12.792	16.338	20.489	24.769	27.587	30.191	33.409	35.718	38.648	40.790
18	6.265	7.015	8.231	9.390	10.865	13.675	17.338	21.605	25.989	28.869	31.526	34.805	37.156	40.136	42.312
19	6.844	7.633	8.907	10.117	11.651	14.562	18.338	22.718	27.204	30.144	32.852	36.191	38.582	41.610	43.820
20	7.434	8.260	9.591	10.851	12.443	15.452	19.337	23.828	28.412	31.410	34.170	37.566	39.997	43.072	45.315
21	8.034	8.897	10.283	11.591	13.240	16.344	20.337	24.935	29.615	32.671	35.479	38.932	41.401	44.522	46.797
22	8.643	9.542	10.982	12.338	14.041	17.240	21.337	26.039	30.813	33.924	36.781	40.289	42.796	45.962	48.268
23	9.260	10.196	11.689	13.091	14.848	18.137	22.337	27.141	32.007	35.172	38.076	41.638	44.181	47.391	49.728
24	9.886	10.856	12.401	13.848	15.659	19.037	23.337	28.241	33.196	36.415	39.364	42.980	45.559	48.812	51.179
25	10.520	11.524	13.120	14.611	16.473	19.939	24.337	29.339	34.382	37.652	40.646	44.314	46.928	50.223	52.620
26	11.160	12.198	13.844	15.379	17.292	20.843	25.336	30.435	35.563	38.885	41.923	45.642	48.290	51.627	54.052
27	11.808	12.879	14.573	16.151	18.114	21.749	26.336	31.528	36.741	40.113	43.195	46.963	49.645	53.023	55.476
235	182.915	187.524	194.434	200.513	207.680	220.037	234.334	249.237	263.176	271.760	279.352	288.354	294.591	302.267	307.728
236	183.796	188.417	195.343	201.437	208.621	221.006	235.334	250.268	264.235	272.836	280.443	289.461	295.710	303.400	308.871
237	184.678	189.310	196.253	202.362	209.562	221.975	236.334	251.299	265.294	273.911	281.533	290.568	296.828	304.532	310.013
238	185.560	190.203	197.163	203.286	210.503	222.944	237.334	252.330	266.353	274.987	282.623	291.675	297.947	305.664	311.154
239	186.442	191.096	198.073	204.211	211.444	223.913	238.334	253.361	267.412	276.062	283.713	292.782	299.065	306.796	312.296
240	187.324	191.990	198.984	205.135	212.386	224.882	239.334	254.392	268.471	277.138	284.802	293.888	300.182	307.927	313.437
241	188.207	192.884	199.894	206.060	213.327	225.851	240.334	255.423	269.529	278.213	285.892	294.994	301.300	309.058	314.578
242	189.090	193.778	200.805	206.985	214.269	226.820	241.334	256.453	270.588	279.288	286.981	296.100	302.417	310.189	315.718
243	189.973	194.672	201.716	207.911	215.210	227.790	242.334	257.484	271.646	280.362	288.070	297.206	303.534	311.320	316.859
244	190.856	195.567	202.627	208.836	216.152	228.759	243.334	258.515	272.704	281.437	289.159	298.311	304.651	312.450	317.999
245	191.739	196.462	203.539	209.762	217.094	229.729	244.334	259.545	273.762	282.511	290.248	299.417	305.767	313.580	319.138
246	192.623	197.357	204.450	210.687	218.036	230.698	245.334	260.576	274.820	283.586	291.336	300.522	306.883	314.710	320.278
247	193.507	198.252	205.362	211.613	218.979	231.668	246.334	261.606	275.878	284.660	292.425	301.626	307.999	315.840	321.417
248	194.391	199.147	206.274	212.539	219.921	232.637	247.334	262.636	276.935	285.734	293.513	302.731	309.115	316.969	322.556
249	195.276	200.043	207.186	213.465	220.863	233.607	248.334	263.667	277.993	286.808	294.601	303.835	310.231	318.098	323.694
250	196.161	200.939	208.098	214.392	221.806	234.577	249.334	264.697	279.050	287.882	295.689	304.940	311.346	319.227	324.832
300	240.663	245.972	253.912	260.878	269.068	283.135	299.334	316.138	331.789	341.395	349.874	359.906	366.844	375.369	381.425
350	285.608	291.406	300.064	307.648	316.550	331.810	349.334	367.464	384.306	394.626	403.723	414.474	421.900	431.017	437.488
400	330.903	337.155	346.482	354.641	364.207	380.577	399.334	418.697	436.649	447.632	457.305	468.724	476.606	486.274	493.132
450	376.483	383.163	393.118	401.817	412.007	429.418	449.334	469.855	488.849	500.456	510.670	522.717	531.026	541.212	548.432
500	422.303	429.388	439.936	449.147	459.926	478.323	499.333	520.950	540.930	553.127	563.852	576.493	585.207	595.882	603.446
550	468.328	475.796	486.910	496.607	507.947	527.281	549.333	571.992	592.909	605.667	616.878	630.084	639.183	650.324	658.215
600	514.529	522.365	534.019	544.180	556.056	576.286	599.333	622.988	644.800	658.094	669.769	683.516	692.982	704.568	712.771
650	560.885	569.074	581.245	591.853	604.242	625.331	649.333	673.942	696.614	710.421	722.542	736.807	746.625	758.639	767.141
700	607.380	615.907	628.577	639.613	652.497	674.413	699.333	724.861	748.359	762.661	775.211	789.974	800.131	812.556	821.347
750	653.997	662.852	676.003	687.452	700.814	723.526	749.333	775.747	800.043	814.822	827.785	843.029	853.514	866.336	875.404
800	700.725	709.897	723.513	735.362	749.185	772.669	799.333	826.604	851.671	866.911	880.275	895.984	906.786	919.991	929.329
850	747.554	757.033	771.099	783.337	797.607	821.839	849.333	877.435	903.249	918.937	932.689	948.848	959.957	973.534	983.133
900	794.475	804.252	818.756	831.370	846.075	871.032	899.333	928.241	954.782	970.904	985.032	1001.630	1013.036	1026.974	1036.826
950	841.480	851.547	866.477	879.457	894.584	920.248	949.333	979.026	1006.272	1022.816	1037.311	1054.334	1066.031	1080.320	1090.418
1000	888.564	898.912	914.257	927.594	943.133	969.484	999.333	1029.790	1057.724	1074.679	1089.531	1106.969	1118.948	1133.579	1143.917

Figure 4.3: Chi-square distribution table

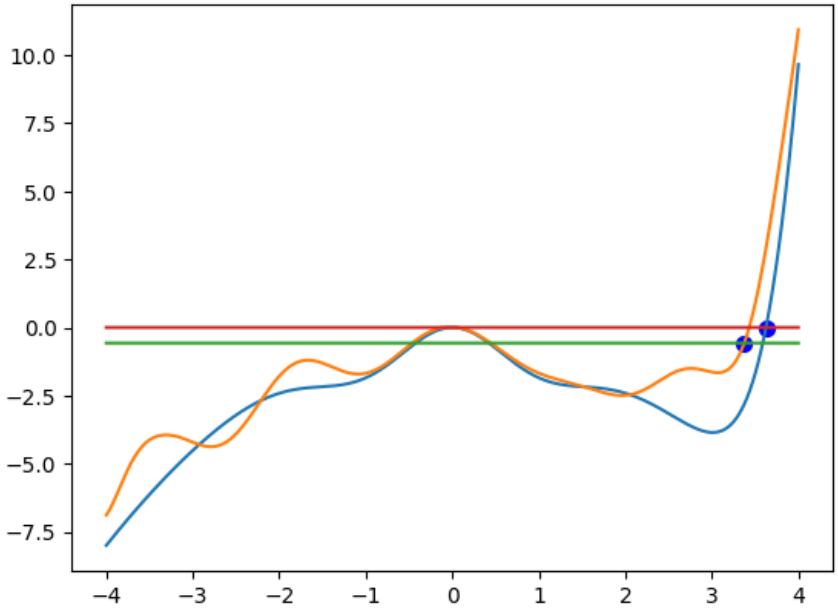


Figure 4.4: True $\dot{V}(x)$ (blue), GPs upper bound of $\dot{V}(x)$ (orange), and $y = -L\tau$ (green).

Figure 4.4 shows the result of theorem 2.2.2. The horizontal coordinates the blue nood is a reflection of the true ROA and the estimated ROA. The true safe set is $S = \{x \in \mathbb{R}] \,|\, V(x) \leq 3.625\}$ and the estimated safe set is $S = \{x \in \mathbb{R}] \,|\, V(x) \leq 3.373\}$, where $V(X) = x^2$ is the lyapunov function. The estimated ROA is within the true safe set and has 96.5% of estimated safe points. This reflection proves the feasibility of doing stability analysis with Gaussian Processes. In addition, $\dot{V}(x)$ goes above $y = -L\tau$ around the origin. It doesn't matter as states around the origin are already inside the given initial safe set. Figure 4.5 shows the result of theorem 2.3.2 using a data set containing only 25 pairs of measurements.

4.5 Higher Dimensional Stability Analysis

All error bounds estimate we have introduced so far can be written in the form of $|f(x) - s(x)| \leq e(x)$, where f is the unknown function we are trying to fit, $s(x)$ stands for interpolation in Kernel Methods and mean estimate in GPs and $e(x) \geq 0$ is the error bound. In short, $e(x)$ is the term we are trying to estimate. In particular, for GPs, $e(x) = \beta_T^{1/2}\sigma_T(x)$,

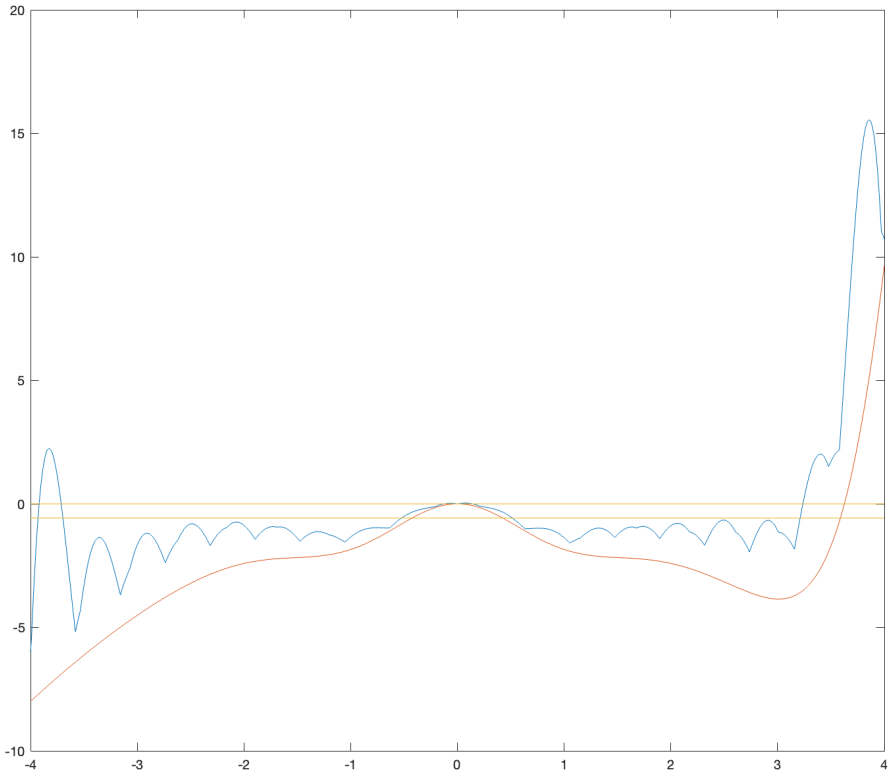


Figure 4.5: True $\dot{V}(x)$ (orange), KRR upper bound of $\dot{V}(x)$ (blue), and $y = -L\tau$ and $y = 0$ (yellow).

where $\sigma_T(x)$ is the posterior variance and T is the number of measurements, so that $\beta_T^{1/2}$ is the exact term we are trying to quantify and it should be kept updated as we get new measurements. However, in the inverted pendulum experiment in [1], a fixed $\beta = 2$ was chosen which results in loss of theoretical guarantees. We want to reproduce this experiment, with β correctly updated according to algorithm 1 and theorem 2.2.2.

We consider an inverted pendulum with angle θ, mass $m = 0.15$ kg, length $l = 0.5$ m, and friction coefficient $\mu = 0.05$Nms/rad. The dynamics are given by

$$\ddot{\theta}(t) = \frac{mgl\sin\theta(t) - \mu\dot{\theta}(t) + u(t)}{ml^2} \quad (5),$$

where $u(t)$ is the torque applied to the pendulum. The torque is limited so that the real system cannot recover from states with $|\theta| > 30$deg. The state is $x = (\theta, \dot{\theta})$. The prior dynamics are also governed by (5), but the friction is neglected and the mass is 0.05 kg lighter. We use a Linear Quadratic Regulator based on the prior model in order to design the controller, and use the corresponding quadratic Lyapunov function to determine the ROA of (5). Figure 4.6 shows the logic of this experiment. In order to achieve this goal, we uniformly sample 100, 144, and 400 states in the area of $[-4, 4] \times [-4, 4]$, output $\dot{x}$ governed by true dynamics at these states as noise free measurements, and calculate the lower estimate of the RKHS norm. As shown in Figure 4.7, we don't need to care about the hidden dynamics $d\theta/dt = \dot{\theta}$ in the sampling process as our sampling here is just to learn the RHS of (5). An i.i.d additive Gaussian noise is then imposed to the noise free measurements, so that β_T proposed in theorem 2.2.2 can be calculated. Table 4.2 shows the result of these calculations. Unfortunately, we do not get results that match theoretical expectations. In this experiment, we followed the strategy in section 2.1.2 to get the lower estimate of the right hand side of RKHs norm of (5), but it doesn't converge. In addition, the β_T we calculated was never close to 2. Figure 4.8 shows the ROA obtained through β_T we calculated with samples. Clearly, it's too conservative to be meaningful. In addition, we tried estimating Γ first and letting $\beta = 2\Gamma^2$, since $2\Gamma^2$ is the dominant term in $beta_t$ of theorem 2.2.1. Note that we are still not updating β_T in every iteration of algorithm 1, but treat β_T as a feature of (5) and try to learn this feature on a larger area. Figure 4.9 shows the ROA obtained in this way with 53.7% safe points relative to true dynamics. We boldly infer that the author of [1] picked a fixed $\beta = 2$ not based on the algorithm he proposed but on the commonly used Gaussian distribution 95% confidence interval which has a width of 1.96σ.

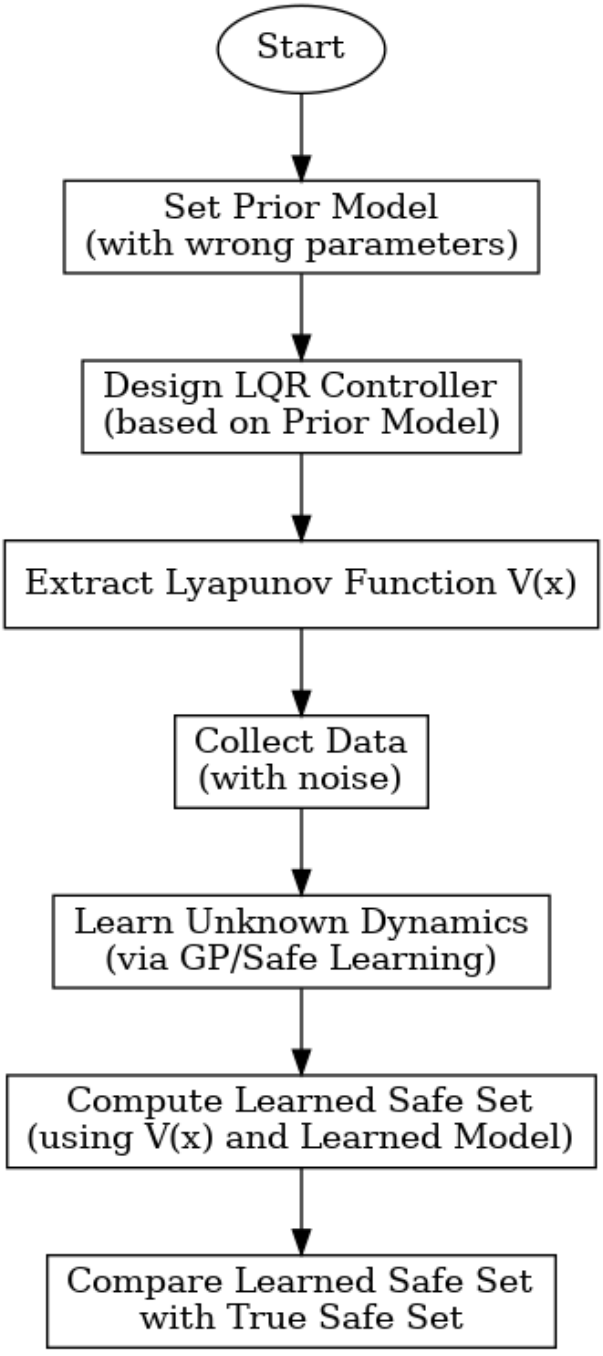


Figure 4.6: Experiment logic

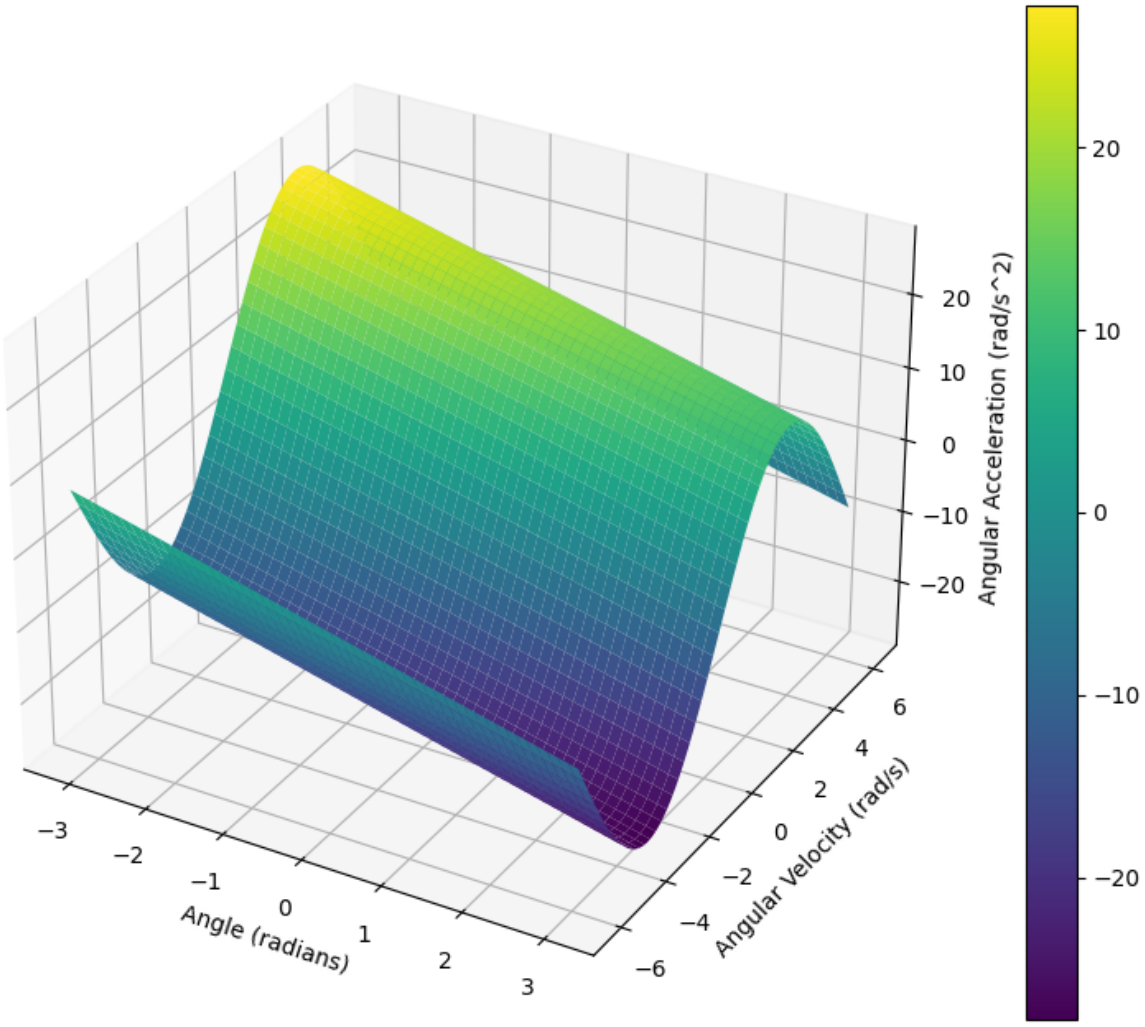


Figure 4.7: Learned Inverted Pendulum Dynamics

Number of measurements	Lower estimate ($f_T^\top K^{-1} f_T$)	β_T
100	68.47	99.65
144	96.16	144.20
400	191.83	400.19

Table 4.2: Inverted pendulum

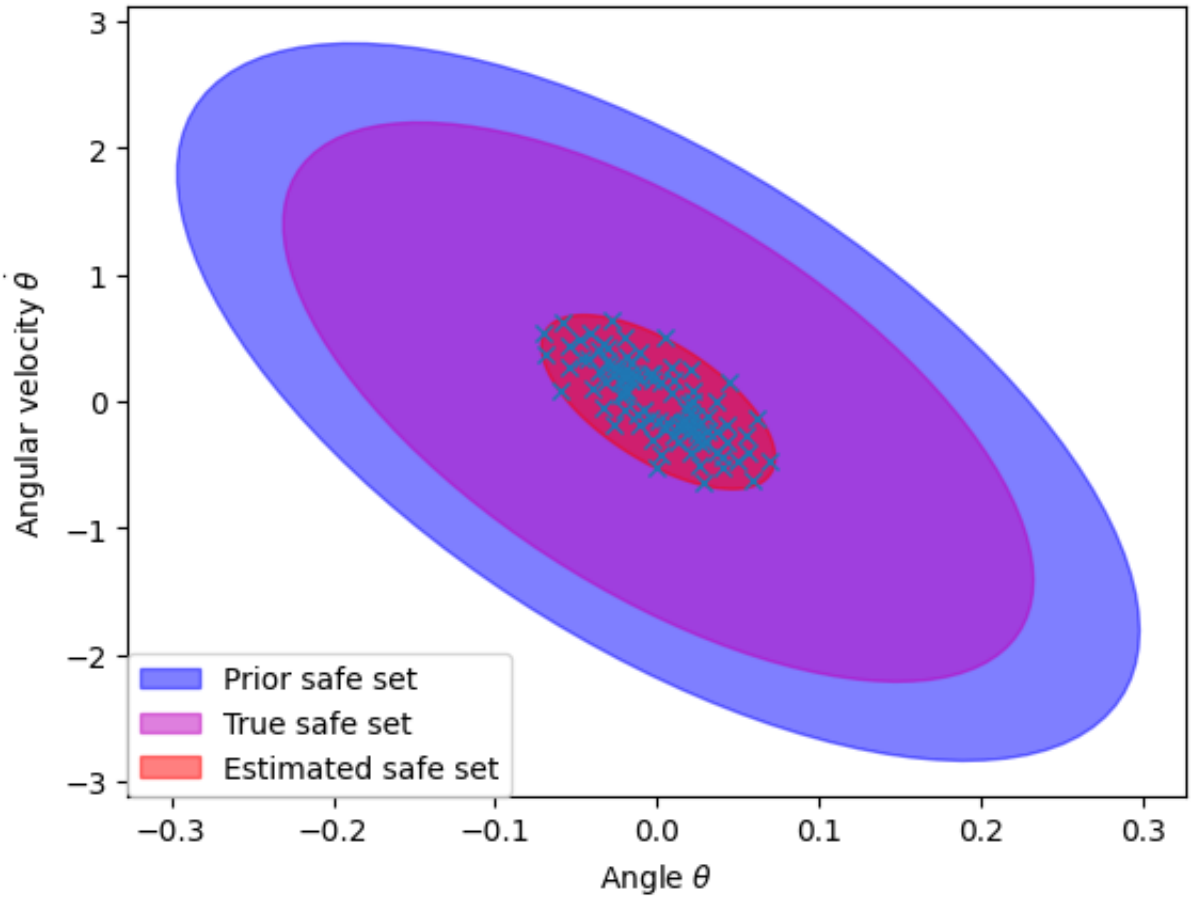


Figure 4.8: ROA learned from 100 measurements learned β

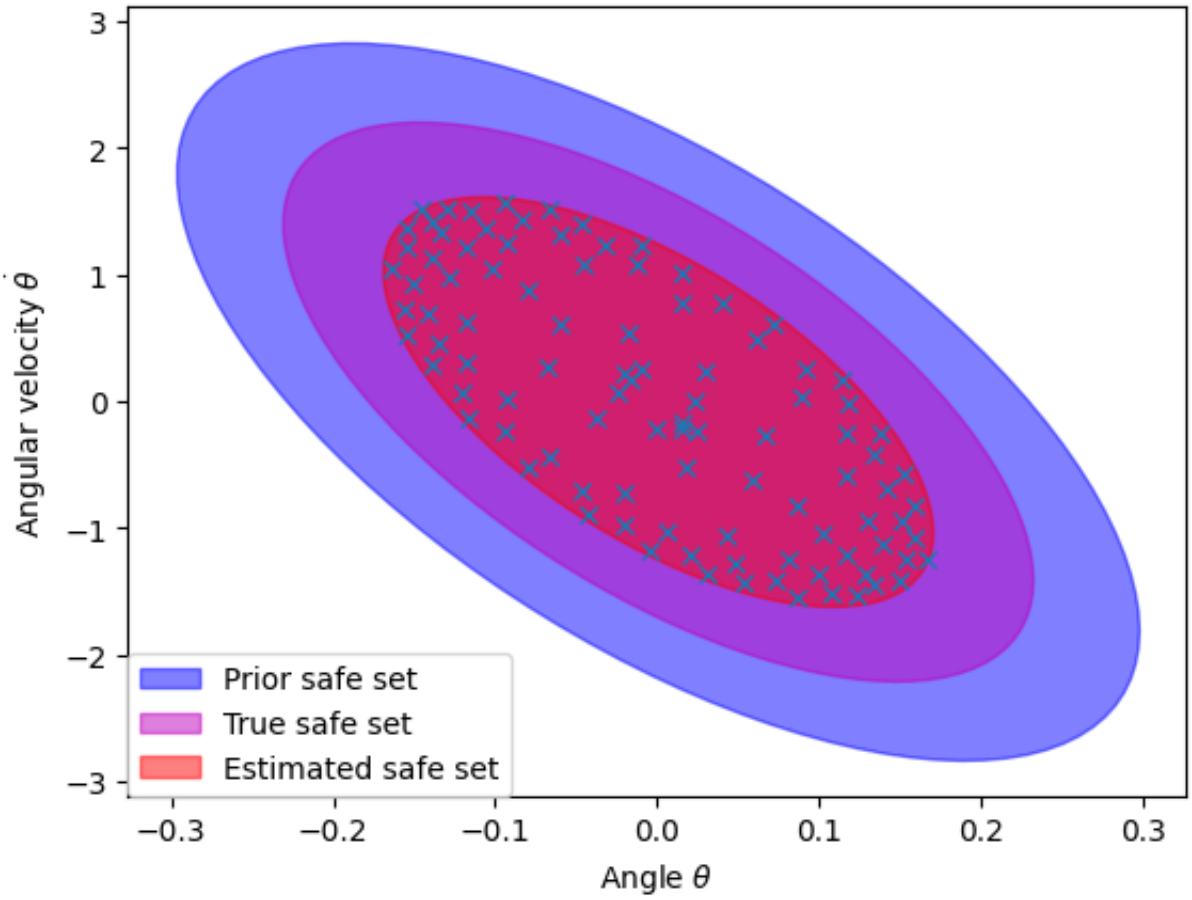


Figure 4.9: ROA learned from 100 measurements with $\beta = 2\Gamma^2$

Chapter 5

Disscussion

In this research paper, we explored error bound estimates for Gaussian Processes and kernel methods, and demonstrated their application to the stability analysis of dynamical systems. Through theoretical derivations and experimental verifications, several key observations and challenges were identified.

First, Gaussian Processes and kernel methods provide strong theoretical guarantees and interpretability for error bounds estimate. These properties are crucial for stability analysis in safety-critical environments.

Second, experimental results verify that these error bounds are not overly conservative in one-dimensional settings. The learned ROA closely matches the true ROA, and the conservative estimates obtained through the proposed methods remain meaningful.

However, challenges arise when extending these methods to higher-dimensional systems. In the inverted pendulum experiment, several significant issues were encountered when attempting to accurately learn the ROA. In particular, it proved infeasible to obtain a reliable estimate of the RKHS norm purely from data, and the resulting ROA estimates were too conservative to be practical.

In addition, this study makes several important contributions and corrections to previous research. We show that it is sufficient to apply error bounds directly to the unknown dynamics but not its lyapunov function. This insight extends the applicability of stability analysis beyond Gaussian Processes to include kernel methods. Furthermore, we optimize the error bounds estimation by introducing chi-square techniques, thereby addressing issues associated with truncated Gaussian noise and improving the robustness of the error estimates.

Overall, this study lays a theoretical foundation for safe learning using Gaussian Processes and kernel methods. It demonstrates promising results in lower-dimensional systems, identifies critical challenges in high-dimensional scenarios.

References

[1] Felix Berkenkamp, Riccardo Moriconi, Angela P. Schoellig, and Andreas Krause. Safe learning of regions of attraction for uncertain, nonlinear systems with gaussian processes. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 4661–4666, 2016.

[2] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.

[3] Ya-Chien Chang, Nima Roohi, and Sicun Gao. Neural lyapunov control. *Advances in neural information processing systems*, 32, 2019.

[4] David Duvenaud. *Automatic model construction with Gaussian processes*. PhD thesis, University of Cambridge, 2014.

[5] Christian Fiedler, Carsten W Scherer, and Sebastian Trimpe. Practical and rigorous uncertainty bounds for gaussian process regression. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 7439–7447, 2021.

[6] Kazumune Hashimoto, Adnane Saoud, Masako Kishida, Toshimitsu Ushio, and Dimos V. Dimarogonas. Learning-based symbolic abstractions for nonlinear control systems. *Automatica*, 146:110646, 2022.

[7] Hassan K Khalil. Nonlinear systems third edition. *Patience Hall*, 115, 2002.

[8] Armin Lederer, Jonas Umlauft, and Sandra Hirche. Uniform error bounds for gaussian process regression with application to safe control. *Advances in Neural Information Processing Systems*, 32, 2019.

[9] Emilio Tanowe Maddalena, Paul Scharnhorst, and Colin N. Jones. Deterministic error bounds for kernel-based learning techniques under bounded noise. *Automatica*, 134:109896, 2021.

[10] Paul Scharnhorst, Emilio T. Maddalena, Yuning Jiang, and Colin N. Jones. Robust uncertainty bounds in reproducing kernel hilbert spaces: A convex optimization approach. *IEEE Transactions on Automatic Control*, pages 1–13, 2022.

[11] Bernhard Schölkopf, Ralf Herbrich, and Alex J Smola. A generalized representer theorem. In *Computational Learning Theory: 14th Annual Conference on Computational Learning Theory, COLT 2001 and 5th European Conference on Computational Learning Theory, EuroCOLT 2001 Amsterdam, The Netherlands, July 16–19, 2001 Proceedings 14*, pages 416–426. Springer, 2001.

[12] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.

[13] Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

[14] Min Wu, Zhengfeng Yang, and Wang Lin. Exact asymptotic stability analysis and region-of-attraction estimation for nonlinear systems. In *Abstract and Applied Analysis*, volume 2013. Hindawi, 2013.

[15] Ruikun Zhou, Thanin Quartz, Hans De Sterck, and Jun Liu. Neural lyapunov control of unknown nonlinear systems with stability guarantees. *Advances in Neural Information Processing Systems*, 35:29113–29125, 2022.