Technische Universität München
Fakultät für Elektrotechnik und Informationstechnik
Lehrstuhl für Informationstechnische Regelung

Safe Learning Control for Gaussian Process Models

Jonas Michael Umlauft

Vollständiger Abdruck der von der promotionsführenden Einrichtung Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzende/-r: Prof. Dr. sc. techn. Reinhard Heckel

Prüfende/-r der Dissertation:

1. Prof. Dr.-Ing. Sandra Hirche
2. Prof. Dr.-Ing. Matthias Müller

Die Dissertation wurde am 01.10.2019 bei der Technischen Universität München eingereicht und durch die promotionsführende Einrichtung Fakultät für Elektrotechnik und Informationstechnik am 22.01.2020 angenommen.

Preamble

This thesis summarizes the conducted research at the Chair of Information-oriented Control (ITR) at the Technical University of Munich (TUM). I am very thankful for all the great people that supported me during this time.

First, I am truly grateful to my doctoral advisor, Prof. Sandra Hirche, who supported me all the way from my undergraduate studies to my doctorate, always giving me the freedom to drive the research in my favorite direction. Her passion for scientific challenges and her constant will to push our research in new directions always inspired me.

Second, I would like to thank all the dedicated and gifted people, I had the pleasure to work with at ITR. With his patience and his commitment to academic research, the supervisor of my bachelor's thesis, Dominik Sieber, motivated me for this doctorate. My colleagues in the con-humo project, José Ramón Medina, Satoshi Endo, Hendrik Börner, Melanie Kimmel and Thomas Beckers, not just helped me to find the topic of this dissertation, but also inspired me in many productive and fruitful discussions. A big thanks also goes to all my students, especially Yunis Fanger, Lukas Pöhler and Armin Lederer, who supported me throughout this thesis in many ways. I also highly appreciated the friendly and professional support by Ulrike Scholze, Stefan Sosnowski, Miruna Werkmeister and all other nonscientific staff with all teaching, administrative and technical matters. Furthermore, I am very grateful for the hospitality of the team of the Chair of Automatic Control Engineering after the ITR lab burned down in 2017.

Last but not least, I express my gratitude to my friends and family, who were very understanding when the work on this thesis absorbed me and who always encouraged me to pursue my goals.

Acknowledgments

The research leading to these results was supported by the EU Seventh Framework Programme FP7/2007-2013 within the ERC Starting Grant "Control based on Human Models (con-humo)", grant agreement no. 337654.

Abstract

Machine learning allows automated systems to identify structures and physical laws based on measured data and to utilize the resulting models for inference. Due to decreasing costs for measuring, processing and storing data as well as increased availability of sensors and improved algorithms, machine learning also known by the buzzword *artificial intelligence* has attracted significant attention.

Data-driven models obtained from machine learning techniques are particularly attractive in areas where an analytic derivation of a model is too tedious or not possible, because the underlying principles are not understood. As control engineering is increasingly applied in these areas, which include, e.g., complex chemical processes or physical human-robot interaction systems, the use of data-driven models has become an advantageous alternative to classical system identification for model-based control techniques.

However, to this day, data-driven models are rarely employed in safety-critical applications, because the success of a controller, which is based on these models, cannot be guaranteed. Therefore, this thesis analyzes the closed-loop behavior of learning control laws using rigorous proofs. We focus in particular on Gaussian processes (GPs), which can be interpreted as data-driven models. The advantages of GPs consist of a high level of flexibility, resulting from their nonparametric nature, an intrinsic bias variance trade-off due to the underlying Bayesian principle and an implicit model fidelity measure. The latter enables the derivation of a model error bound and therefore facilitates the application of robust and adaptive control techniques to guarantee stability. Along these lines, this thesis provides the three following major contributions in learning and control based on GPs.

We show how Gaussian processes are employed for physically consistent identification of unknown dynamical systems. Under the prior knowledge, that the true system is dissipating energy, we enforce the model to show the same asymptotic behavior. This requires to learn not only the dynamics, but also the convergence behavior, which we achieve by learning control Lyapunov functions. Conditions for asymptotic stability are derived and the minimal number of required data points is provided. A quantitative comparison based on a real-world human motion data set shows the advantages over existing methods.

Furthermore, we propose a control law based on Gaussian process models, which actively avoids uncertainties in the state space and favors trajectories along the training data, where the system is well-known. We show that this behavior is optimal in the presence of power limitations as it maximizes the probability of asymptotic stability.

Additionally, we consider an event-triggered online learning control law, which safely explores an initially unknown system. It only takes new training data whenever the uncertainty in the system becomes too large to ensure conditions for asymptotic stability. As the underlying feedback linearizing control law only requires a locally precise model, no data-expensive global model is needed. In order to increase data-efficiency, an information gain based criteria is proposed for a safe forgetting strategy of data points.

Further results in dynamic knowledge-based leader-follower control, uncertainty modeling for programming by demonstration, risk-aware path tracking and scenario-based optimal control are summarized. All major results are validated with rigorous proofs and demonstrated in simulation.

Zusammenfassung

Maschinelles Lernen ermöglicht automatisierten Systemen - basierend auf Messdaten - Strukturen und Gesetzmäßigkeiten zu modellieren und diese für Vorhersagen zu nutzen. Aufgrund sinkender Kosten für die Messung, Verarbeitung und Speicherung von Daten und einer hohen Verfügbarkeit von Sensoren und verbesserten Algorithmen, gewann maschinelles Lernen, auch unter dem Schlagwort *künstliche Intelligenz*, breite Aufmerksamkeit.

Datengetriebene Modelle aus dem Bereich des maschinellen Lernens sind vor allem in jenen Bereichen nützlich, in denen die analytische Herleitung eines Models zu aufwendig oder nicht möglich ist, weil die zugrundeliegenden Prinzipien unbekannt sind. Da Regelungstechnik zunehmend in diesen Bereichen (z.B. komplexe chemische Prozesse, physikalische Mensch-Roboter Interaktion) eingesetzt wird, bietet die Verwendung dieser datengetriebenen Modelle für die modellbasierte Regelung zunehmend Vorteile gegenüber der klassischen Systemidentifikation.

Bis heute werden datengetriebene Modelle jedoch kaum in sicherheitsrelevanten Anwendungen eingesetzt, weil keine Garantien für die erfolgreiche Regelung (basierend auf diesen Modellen) gegeben werden können. Daher untersucht diese Arbeit das Verhalten geschlossener Regelkreise mit lernenden Regelgesetzen mithilfe von Beweisen. Der Fokus liegt im besonderen auf Gaußprozessen, die als datengetriebene Modelle interpretiert werden können. Die Vorteile sind eine hohe Flexibilität, die sich aus der nicht-parametrischen Grundstruktur ergibt, eine implizite Lösung des Verzerrung-Varianz-Dilemmas aufgrund des zugrundeliegenden Prinzips von Bayes und ein inhärentes Maß für die Modellsicherheit. Letzteres ermöglicht die Herleitung einer Schranke für den Modellfehler und erlaubt damit den Einsatz von Techniken der robusten und adaptiven Regelung, um Stabilität zu garantieren. In diesem Sinne liefert diese Arbeit drei wesentliche Beiträge zu der lernbasierten Regelung mithilfe von Gaußprozessen.

Zunächst zeigen wir, wie Gaußprozesse eingesetzt werden können, um eine physikalisch konsistente Identifizierung von unbekannten dynamischen Systemen zu erreichen. Unter dem Vorwissen, dass das echte System Energie dissipiert, erzwingen wir das gleiche asymptotische Verhalten im Modell. Dafür erforderlich ist es neben der Dynamik auch das Konvergenzverhalten zu lernen. Dies wird erreicht, indem eine regelnde Lyapunovfunktion gelernt wird. Bedingungen für die asymptotische Stabilität und die minimale Anzahl an Datenpunkten werden hergeleitet. Ein quantitativer Vergleich basierend auf einem praxisnahen Datensatz mit menschlichen Bewegungen zeigt die Vorteile gegenüber existierenden Methoden.

Des Weiteren wird ein Regelgesetz basierend auf Gaußprozessen vorgestellt, welches aktiv Unsicherheiten im Zustandsraum vermeidet und Trajektorien bevorzugt, die entlang der Trainingsdaten liegen und damit durch Bereiche verlaufen, in denen das System wohlbekannt ist. Wir zeigen, dass dieses Verhalten optimal ist, falls eine Leistungsbeschränkung für das Steuersignal vorliegt, da es die Wahrscheinlichkeit für asymptotische Stabilität maximiert.

Zusätzlich wird ein ereignisbasiertes online lernendes Regelgesetz betrachtet, welches sicher ein zunächst unbekanntes System exploriert. Es nimmt nur dann neue Trainingsdaten auf, wenn die Unsicherheit im Modell zu groß wird, um die Bedingungen für die asymptotische Stabilität zu erfüllen. Da die zugrundeliegende Linearisierung durch Rückführung nur lokal ein präzises Model benötigt, kann auf ein globales Model, welches ineffizient im Bezug auf die Menge der Datenpunkte ist, verzichtet werden. Um die Dateneffizienz zu erhöhen, wird ein sicherer Vergessensmechanismus von Datenpunkten vorgestellt, welcher den Informationszu-

gewinn als Kriterium verwendet.

Darüber hinaus werden weitere Ergebnisse in den Bereichen der wissensbasierten Zuweisung der Führer- und der Folgerrolle, der Unsicherheitsmodellierung in der Programmierung durch Nachahmen, der risikoabhängigen Pfadverfolgung und der Szenario-basierten optimalen Regelung kurz zusammengefasst und vorgestellt.

Alle Resultate werden durch grundlegende Beweise belegt und in Simulationen veranschaulicht.

Contents

1 Introduction

As automated systems advance to new fields of applications, the environment in which they operate becomes increasingly complex to model. Examples include physical human robot interaction, autonomous driving, the process industry and many others. In these applications, an analytic derivation of a model is very complex or not even possible because the underlying physical principles are not understood or unknown. Thus, an analytic representation for the system's behavior is difficult to derive, but would be necessary for controlling these processes. Furthermore, autonomous devices, like unmanned aerial/underwater vehicles (UAVs/UUVs) are often deployed in unknown conditions requiring an online modeling of the surrounding environment. In such complex scenarios, classical control techniques often show unsatisfactory performance, which triggered the utilization of data-driven models developed in the field of machine learning.

The wide spread of machine learning in recent years is favored by the simplified collection, processing and storing of data caused by high availability of sensors, faster processors and cheaper storage. More particular, supervised learning is well suited to obtain precise models from data if prior knowledge of the system is barely available. Neural networks (NNs) are the most popular technique for supervised learning due to their impressive performance on various tasks including playing the game of Go [1]. Nevertheless, a large amount of training data (more than 10^7 data points in [1]) is often required until NNs reach high precision. In contrast, Gaussian processes (GPs) are well known for their data-efficiency and generalize well in untrained areas also for small data sets [2].

Despite the success in various fields including image classification [3], user recommendation [4] and artificial intelligence for gaming [1], learning algorithms are still rarely found in safety-critical applications such as robotic control. In particular, reinforcement learning algorithms which aim to autonomously acquire skills by interacting with the environment, are barely analyzed with respect to safety. Highly critical is the autonomous exploration, which these algorithm perform to discover better strategies, as this implies operation in areas without training data and results in unpredictable behavior.

These unsolved problems reveal the present difficulty of learning-based control: Engineers hesitate for good reasons to employ data-driven control in safety-critical functions (e.g., the interaction with humans) where they are needed the most due to the missing alternatives to data-driven approaches. This leaves a high potential of technological advancement unused and thereby motivates the work of this thesis.

1.1 Challenges in data-driven control

Before data-driven control can be applied in safety-critical applications, a rigorous analysis of the closed-loop behavior is required. From the author's perspective, the following fundamental questions must be answered:

Challenge 1. *How can data-driven models be designed to be logically or physically consistent, but yet flexible to model the high complexity of the system?*

It is generally not ensured that the behavior predicted by a data-driven model does not violate any logical or physical principles, e.g., the conservation or dissipation of energy or Kirchhoff's law, because no fixed structure is imposed. Ignoring such properties of the real system in the modeling process is a loss of available information and hence, this knowledge should be used. However, this topic has only received very little attention in the literature [5] and is a significant step for making data-driven models more reliable.

Challenge 2. *Which safety guarantees can control laws, which are based on data-driven models, provide and which assumptions are necessary in return?*

On the one side, if nothing about the controlled system is initially known, guarantees for the closed-loop behavior of the system under any control law cannot be expected. This follows from the no-free-lunch theorems [6] stating that observations in one situation allow no conclusion about similar ones given that no correlation between the two exists. On the other side, assuming a fixed parametric structure of a real system simplifies the analysis of the closed-loop behavior. However, this strictly limits the domains of applications and ignores the fact that parametric models are usually inaccurate as they can not capture all real-world effects.

Therefore, nonparametric models, which will be used here as equivalent term for *data-driven* models, avoid assumptions regarding the parametric structure but make implications on high-level properties of the real system. This includes the smoothness of the dynamic function or the convergence behavior. It can be concluded that there is a trade-off between the imposed assumptions and the safety certificates which can be derived for the closed-loop system. The goal of this thesis is to explore this trade-off and push it towards stronger guarantees with less restrictive assumptions.

Challenge 3. *How can control laws, which are based on data-driven models, avoid areas where the data is lacking and, vice versa, favor actions for which the model has a high fidelity?*

Data-driven models mainly rely on measurements taken from the real system, but the available data will always be finite and not cover the entire input space densely. In unknown areas, Bayesian approaches like Gaussian processes fall back to the prior, but the generalization outside the training set is not as reliable as in trained areas. It is therefore a well-known principle, also applied by humans, to preferably operate in scenarios where experience is available. This risk-awareness is crucial for control laws operating on data-driven models as the model cannot be trusted far away from training data. A controller which internalizes such uncertainty-avoiding strategies is key for the application of data-driven control in safety-critical fields.

Challenge 4. *How can an initially unknown system be explored safely and which training points should be collected for data-driven models?*

Many systems cannot be probed offline, but measurements must be taken during operation, e.g., the air drag of an aircraft can only be measured accurately while flying. However, if no stabilizing controller is available, training data cannot be collected safely. Solving this chicken and egg problem is crucial to employ learning-based controllers in various domains. The exploration of initially unknown dynamics during operation is not a trivial task since stability must be guaranteed at all times to prevent damage to the system or harm to its environment.

Furthermore, controllers, which constantly adapt the model to newly taken measurements, are considered as life-long learning systems. For conventional, parametric models, the constantly growing set of measurements can be handled straight forward: Incrementally, the parameters are updated with every new data point, which is then deleted (as all model information is stored in the parameters). In contrast, for a data-driven nonparametric model, the data points *are* the parameters and must therefore be stored. The complexity of the model grows with the number of available data points. This property is generally desired because it is the reason for the infinite expressive power. But, considering computational limits and storage constraints, this life-long learning imposes critical questions: When should measurements be taken to generate new data points? Which data points must be kept and which can be discarded?

Addressing these challenges is essential to successfully employ data-driven control approaches in new fields of applications and are therefore addressed in this thesis.

1.2 Main contributions and outline

This thesis develops control laws for unknown nonlinear dynamical system which - based on data-driven identification - guarantee stability of the closed-loop system. Employing Gaussian processes, we ensure physical consistency in the modeling, uncertainty-aware behavior and data-efficient online learning of an initially unknown system.

First, Chapter 2 provides a brief introduction to Gaussian processes and a general review of the literature on GP-based control. The Chapters 3 to 5 present in detail our approaches addressing the Challenges 1 to 4. Each of the chapters contains a brief and specific overview of the most relevant related work, and their main contribution is outlined in the following. Furthermore, Chapter 6 provides an overview of additional work by the author to apply data-driven control concepts in robotics. A conclusion and possible directions for future research are presented in Chapter 7.

Chapter 3: Identification of Stable Systems Addressing Challenge 1, the chapter focuses on transferring the knowledge about stable equilibria of the real system into the GP-based model which thereby achieves consistent dissipation of energy. We augment the GP model by a control Lyapunov function. This function is also learned from data and ensures a consistent convergence behavior between the model and the real system. The analysis is performed for the deterministic and the stochastic interpretation of the GP model (in detail introduced in Sec. 2.3) and validated on a real-world data set. The results presented in this chapter have been partially published in [7] and in [8].

Chapter 4: Uncertainty-based Control Lyapunov Design This chapter addresses Challenges 2 and 3. With the goal to stabilize an unknown system based on a GP model under a given input power constraint, we propose an approach which maximizes the probability for asymptotic convergence. Based on principles from path planing, a novel uncertainty-aware control Lyapunov function is proposed which drives the system away from areas with low model fidelity. It thereby reveals an equivalence between dynamic programming and the maximization of the probability to converge which has not been studied before. This allows to employ computationally efficient tools from optimal control for the implementation of a risk-aware control strategy. The results presented in this chapter have been partially published in [9].

Chapter 5: Feedback Linearization with event-triggered Online Learning This chapter addresses Challenges 2 and 4. Considering the case that initially no data is given, an online learning control algorithm is presented which safely explores the unknown systems. We propose an event-triggered approach which adds data to the training data set whenever the uncertainty of the model exceeds a Lyapunov stability criteria. Thus, new data points are only added if necessary, which makes the approach highly data-efficient. Furthermore, we derive a safe forgetting strategy to keep the number of data points under a given data budget. The approach allows an asymptotically stable tracking of smooth reference trajectories under mild assumptions regarding the true system. The results presented in this chapter have been published in [10], [11] and [12].

Gaussian Processes in Identification and Control

2

This chapter reviews Gaussian process modeling and its application in system identification and control. It provides a broad overview of the related literature and states existing properties of Gaussian processes, which are utilized in Chapters 3 to 5.

2.1 Gaussian process regression

A Gaussian process (GP) is a stochastic process defined in [2, Chapter 2] as follows.

Definition 2.1. *A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.*

The definition does not state this explicitly, but the GP describes a set of infinitely many Gaussian distributed random variables. Each of them is assigned a mean and (as they are jointly distributed) a covariance, given by a prior mean function $\mathfrak{m}\colon \mathbb{X} \to \mathbb{R}$ and a prior covariance function $\mathfrak{k}\colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}$. Both together fully specify a GP. The set $\mathbb{X} \subseteq \mathbb{R}^n$, with $n \in \mathbb{N}$ is an Euclidean input space where each element is assigned a Gaussian distributed random variable. This leads to the notation of a GP

$$f_{\mathcal{GP}}(\boldsymbol{x}) \sim \mathcal{GP}\left(\mathfrak{m}(\boldsymbol{x}), \mathfrak{k}(\boldsymbol{x}, \boldsymbol{x}')\right)$$

and its interpretation as *distribution over functions* [2]. Accordingly, the GP is frequently applied in regression tasks, where noisy measurements of an unknown function $f\colon \mathbb{X} \to \mathbb{R}$ are given in a data set

$$\mathbb{D} = \left\{\boldsymbol{x}^{(i)}, y^{(i)}\right\}_{i=1}^{N}, \quad \text{with} \quad y^{(i)} = f\left(\boldsymbol{x}^{(i)}\right) + \omega^{(i)}, \qquad \omega \sim \mathcal{N}(0, \sigma_{\text{on}}^2),$$

where $\sigma_{\text{on}}^2 \in \mathbb{R}_{+,0}$ is the variance of the observation noise and $\omega^{(i)} \in \mathbb{R}$, $i = 1, \ldots, N$ are independent and identically distributed (i.i.d.). Given an arbitrary test point $\boldsymbol{x} \in \mathbb{X}$, the inferred function value must be jointly distributed according to Definition 2.1

$$\begin{bmatrix} f(\boldsymbol{x}) \\ \boldsymbol{y}^{(1:N)} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathfrak{m}(\boldsymbol{x}) \\ \mathfrak{m}\left(\boldsymbol{x}^{(1:N)}\right) \end{bmatrix}, \begin{bmatrix} \mathfrak{k}(\boldsymbol{x}, \boldsymbol{x}) & \boldsymbol{k}(\boldsymbol{x})^{\mathsf{T}} \\ \boldsymbol{k}(\boldsymbol{x}) & \boldsymbol{K} + \sigma_{\text{on}}^2 \boldsymbol{I}_n \end{bmatrix}\right), \tag{2.1}$$

where

$$\boldsymbol{k}(\boldsymbol{x}) = \mathfrak{k}\left(\boldsymbol{x}^{(1:N)}, \boldsymbol{x}\right) = \begin{bmatrix} \mathfrak{k}\left(\boldsymbol{x}^{(1)}, \boldsymbol{x}\right) & \cdots & \mathfrak{k}\left(\boldsymbol{x}^{(N)}, \boldsymbol{x}\right) \end{bmatrix}^{\mathsf{T}} \in \mathbb{R}^N \tag{2.2}$$

and

$$\boldsymbol{K} = \begin{bmatrix} \mathfrak{k}\left(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(1)}\right) & \cdots & \mathfrak{k}\left(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(N)}\right) \\ \vdots & \ddots & \vdots \\ \mathfrak{k}\left(\boldsymbol{x}^{(N)}, \boldsymbol{x}^{(1)}\right) & \cdots & \mathfrak{k}\left(\boldsymbol{x}^{(N)}, \boldsymbol{x}^{(N)}\right) \end{bmatrix} \in \mathbb{R}^{N \times N}. \tag{2.3}$$

Accordingly, the inference is obtained by conditioning on the test input and the training data

$$f(\boldsymbol{x})|\mathbb{D} \sim \mathcal{N}\Big(\underbrace{\mathfrak{m}(\boldsymbol{x}) + \boldsymbol{k}(\boldsymbol{x})^\intercal \boldsymbol{K}_{\text{on}}^{-1} \left(\boldsymbol{y}^{(1:N)} - \mathfrak{m}\left(\boldsymbol{x}^{(1:N)}\right)\right)}_{=:\mu(\boldsymbol{x})}, \underbrace{\mathfrak{k}(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}(\boldsymbol{x})^\intercal \boldsymbol{K}_{\text{on}}^{-1} \boldsymbol{k}(\boldsymbol{x})}_{=:\sigma^2(\boldsymbol{x})} \Big), \tag{2.4}$$

where $\boldsymbol{K}_{\text{on}} = \boldsymbol{K} + \sigma_{\text{on}}^2 \boldsymbol{I}_N$. The functions $\mu \colon \mathbb{X} \to \mathbb{R}$ and $\sigma^2 \colon \mathbb{X} \to \mathbb{R}_{+,0}$ are the posterior mean and posterior variance functions, respectively.

Remark 2.1. *The choice of the kernel significantly determines the properties of the function over which the GP describes a distribution. It allows to set high-level properties, like the differentiability or boundedness of the posterior mean function.*

A common choice for the covariance function, also considered as the kernel (function), which leads to infinite differentiable and bounded functions is the squared exponential (SE) kernel with automatic relevance determination

$$\mathfrak{k}_{\text{SE}}(\boldsymbol{x}, \boldsymbol{x}') = \zeta^2 \exp\left(-\sum_{j=1}^{n} \frac{(x_j - x_j')^2}{2\ell_j^2} \right), \tag{2.5}$$

where the signal variance $\zeta^2 \in \mathbb{R}_{+,0}$ and the lengthscales $\ell_j^2 \in \mathbb{R}_+$ with $j = 1, \ldots, n$ are so-called hyperparameters. These are typically concatenated in $\boldsymbol{\psi}$ (here: $\boldsymbol{\psi} \in \mathbb{R}^{n+1}$) and determined from a likelihood maximization

$$\max_{\boldsymbol{\psi}} \log p\left\{ \boldsymbol{y}^{(1:N)} | \boldsymbol{x}^{(1:N)}, \boldsymbol{\psi} \right\} = \max_{\boldsymbol{\psi}} -\frac{1}{2} \left(\boldsymbol{y}^{(1:N)^T} \boldsymbol{K}_{\text{on}}^{-1} \boldsymbol{y}^{(1:N)} + \log\det \boldsymbol{K}_{\text{on}} + N \log(2\pi) \right). \tag{2.6}$$

Even though this optimization is generally non-convex, it is commonly solved with gradient-based approaches [2]. Nevertheless, [13] provides an analysis for not properly chosen hyper-parameters.

Remark 2.2. *If GP regression is employed to model functions with multidimensional outputs $\boldsymbol{f} \colon \mathbb{X} \to \mathbb{R}^m$, $m \in \mathbb{N}$ the observation noise is typically assumed to be independent*

$$\boldsymbol{y}^{(i)} = \boldsymbol{f}\left(\boldsymbol{x}^{(i)}\right) + \boldsymbol{\omega}, \qquad \boldsymbol{\omega} \sim \mathcal{N}(\boldsymbol{0}, \sigma_{on}^2 \boldsymbol{I}_m), \qquad i = 1, \ldots, N.$$

This allows a separate modeling of each of the m output dimension with an independent GP, with corresponding mean and kernel functions $\mathfrak{m}_j(\cdot), \mathfrak{k}_j(\cdot,\cdot)$, with hyperparameters $\boldsymbol{\psi}_j$ $j = 1, \ldots, m$. The resulting posterior mean and covariance functions

$$\begin{aligned} \mu_j(\boldsymbol{x}) &= \mathfrak{m}_j(\boldsymbol{x}) + \boldsymbol{k}_j(\boldsymbol{x})^\intercal (\boldsymbol{K}_j + \sigma_{on}^2 \boldsymbol{I}_n)^{-1} \left(\boldsymbol{y}_j^{(1:N)} - \mathfrak{m}_j\left(\boldsymbol{x}^{(1:N)}\right)\right) \\ \sigma_j^2(\boldsymbol{x}) &= \mathfrak{k}_j(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}_j(\boldsymbol{x})^\intercal (\boldsymbol{K}_j + \sigma_{on}^2 \boldsymbol{I}_n)^{-1} \boldsymbol{k}_j(\boldsymbol{x}) \end{aligned}$$

are concatenated and the following notation will be used

$$\boldsymbol{\mu}(\boldsymbol{x}) = \begin{bmatrix} \mu_1(\boldsymbol{x}) \\ \vdots \\ \mu_m(\boldsymbol{x}) \end{bmatrix}, \ \boldsymbol{\sigma}^2(\boldsymbol{x}) = \begin{bmatrix} \sigma_1^2(\boldsymbol{x}) \\ \vdots \\ \sigma_m^2(\boldsymbol{x}) \end{bmatrix}, \ \boldsymbol{\sigma}(\boldsymbol{x}) = \begin{bmatrix} \sigma_1(\boldsymbol{x}) \\ \vdots \\ \sigma_m(\boldsymbol{x}) \end{bmatrix}, \ \boldsymbol{\Sigma}(\boldsymbol{x}) = \begin{bmatrix} \sigma_1^2(\boldsymbol{x}) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \sigma_m^2(\boldsymbol{x}) \end{bmatrix}. \tag{2.7}$$

Remark 2.3. *If there exists prior knowledge of a noise free function value of the unknown function $f(\cdot)$, thus $y_{kn} = f(\boldsymbol{x}_{kn})$, then this can be included in the GP regression. The joint distribution in* (2.1) *is extended by the additional training point as follows*

$$\begin{bmatrix} f(\boldsymbol{x}) \\ \boldsymbol{y}^{(1:N)} \\ y_{kn} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathfrak{m}(\boldsymbol{x}) \\ \mathfrak{m}\left(\boldsymbol{x}^{(1:N)}\right) \\ \mathfrak{m}(\boldsymbol{x}_{kn}) \end{bmatrix}, \begin{bmatrix} \mathfrak{k}(\boldsymbol{x},\boldsymbol{x}) & \boldsymbol{k}(\boldsymbol{x})^\intercal & \mathfrak{k}(\boldsymbol{x},\boldsymbol{x}_{kn}) \\ \boldsymbol{k}(\boldsymbol{x}) & \boldsymbol{K} + \sigma_{on}^2 \boldsymbol{I}_n & \boldsymbol{k}(\boldsymbol{x}_{kn}) \\ \mathfrak{k}(\boldsymbol{x}_{kn},\boldsymbol{x}) & \boldsymbol{k}(\boldsymbol{x}_{kn})^\intercal & \mathfrak{k}(\boldsymbol{x}_{kn},\boldsymbol{x}_{kn}) \end{bmatrix} \right), \tag{2.8}$$

thus it will be handled like an additional training point without noise. The regression is performed equivalently to (2.4).

2.2 Gaussian processes in control

This literature review provides a general overview on Gaussian processes in control. It particularly focuses on works which derive formal guarantees for the behavior of the closed-loop controller but (due to the extensive research in this field in the past years) does not claim completeness. For the major contributions in Chapters 3 to 5 a more specific review of the existing work will be presented in the beginning of each chapter. A basic tutorial on GPs in control can be found in [14] and [15] also provides an overview on existing literature.

2.2.1 System identification based on Gaussian processes

Classical system identification focuses on model selection and parameter tuning based on an analytical derivation of possible model structures [16]. The idea to utilize supervised learning techniques to model dynamical systems started around 1990 and initially focused mainly on the identification based on neural networks (NNs), e.g., [17] and [18]. An overview on NNs in control is provided in [19].

Around 2000, a case based comparison [20] showed that GPs perform similarly to NNs for modeling nonlinear dynamical systems, with advantages for the GP on small training sets. It was accompanied by further work along these lines in [21], [22] and [23] which identified the advantage of GPs to propagate the uncertainty of the state over multiple time steps in predictive control schemes. A summary of this early work is found in [24] and [25].

The stability analysis of Gaussian process dynamical models was first approached in [26] and [27] investigating how the convergence properties (number of equilibria, boundedness of the state) relates to the choice of the kernel.

2.2.2 Learning control with Gaussian processes

Gaussian processes are also extensively used in model-based reinforcement learning (RL). Starting in [28] and [29] the parameters of the policy / control law are updated based on an

internal simulation using the GP model before the controller is applied to the real system, see [30] and [31]. The PILCO (Probabilistic Inference for Learning and COntrol) algorithm introduced in [32]–[35] iteratively optimizes the controller and the GP model (based on new measurements). It achieves an impressively high data-efficiency and this minimizes time to interact with the environment, e.g., the double pendulum swing up task is learned with less than 100 seconds of experience [36]. A survey of these techniques in the field of robotics is available in [37].

To predict the future evolution of the state, these RL algorithms rely on the approximation through moment matching as explained in Sec. 2.3 and corresponds to the belief state interpretation. A computational efficient solution to avoid this approximation is presented in [38] based on numerical quadrature. It allows to construct a region in which stability of a controller can be shown [39], [40].

2.2.3 Gaussian processes for optimal control

The work in [41] uses Bellman residual elimination based on Gaussian processes for an approximate dynamic programming algorithm. A GP represents an approximation to the cost-to-go function and it is shown that - based on a proper kernel - the Bellman residual becomes zero. Similarly, [42] and [43] use a GP to model the value function for a finite time horizon. The work in [44] uses differential dynamic programming based on a Gaussian process dynamics and belief-space propagation of the state. In [45] a probabilistic approach for value iteration is developed to learn the value function using a GP. In [46] iterative linear quadratic regulator (LQR) is applied to obtain a (locally) optimal control law in real-time execution.

2.2.4 Gaussian processes in model predictive control

While most of the approaches previously introduced also use a predictive scheme to derive the controller, this section presents works which explicitly formulate a receding horizon model predictive control (MPC) problem as defined in [47]. In [48] and [49] a MPC algorithm based on GPs for state and input constraints is proposed which formulates an upper bound on the uncertainty as additionally constraint. However, a feasibility and stability analysis is missing. The work in [50] presents a piece-wise linear approximation to the MPC problem and [51] uses MPC in an iterative learning setting performing an analysis of feasibility and safety. The work in [52] directly learns the optimal control law based on supervised learning and analyzes the robustness of the approach. It thereby allows to circumvent the high computational complexity of MPC in the online phase, by performing it offline during the training of the GP. Further approaches are presented in[53], [54] and [55] with applications in autonomous racing, quadcopter control and path tracking, respectively.

2.2.5 Gaussian processes for internal model control

While most control techniques for GPs are based on a state space representation of the dynamical system, the work in [56] considers an input-output model. It utilizes internal model control, which is equivalent to a one step model predictive controller [57]. A variant, which makes use of the variance prediction to constraint the system to areas with high model fidelity, is presented in [58].

2.2.6 Adaptive control and safe exploration

A large body of the literature on GP-based control operates on a constant training set during the run-time of the controller. The model-based RL algorithms (Sec. 2.2.2) record new training data during each interaction with the environment, but the controller is not updated until the interaction is stopped. In [59], the first GP-based controller which updates the model during operation is proposed. It is extended in [60] to avoid areas with high model uncertainties. A similar approach for control affine system is proposed in [29]. However, none of these approaches analyzes the asymptotic behavior of the closed-loop system.

In [61] and [62] the residual dynamics of a control affine system are modeled and analyzed in a model reference adaptive control (MRAC) setting. A probabilistic boundedness guarantee for the control law is shown in [63]. The work in [64] uses a generative network for GPs to predict uncertainties, while [65] focuses on the online estimation of kernel hyperparameters.

The work in [66] and [67] develops an optimization based feedback control law to guarantee robust stability of the resulting closed-loop system. This allows to improve the performance while safely collecting new training points. A more general case is addressed in [68] to safely explore the region of attraction, which is then applied in [69] in a RL setting. It thereby addresses the first part of Challenge 4, but does not answer which training points are valuable to keep.

2.2.7 Gaussian processes in robotics

Gaussian process regression becomes computationally inefficient for a large number of training points. Therefore, the GP-based robotic control faces the challenge of real-time constraints requiring the GP inference to be performed within milliseconds [70]. High dimensional state-spaces require a high-number of training points, which reinforces this difficulty. This led to the development of local GP models for computed torque control, which trains multiple GP models covering the state space [71]–[74].

An alternative approach is the incremental sparsification of training data, see [75] and [76]. A combination of both techniques is proposed in [77], [78]. A survey on this development is provided in [79].

More recent developments focus on semi-parametric modeling to combine the strength of nonparametric approaches (like GPs), with classical (parametric) mechanical models [80]. The work in [81] shows stability for the compute torque semi-parametric control approach.

While most previous approaches ignore the uncertainty estimate provided by the GP, the work in [82] proposes an adaptation of feedback gains based on this measure. In [83] and [84] this idea is further explored and a stability proof for the gain adaptation is provided. Even though it does not directly prevent the controller from approaching regions with high model uncertainty, it is a first step to address Challenge 3.

2.2.8 Practical applications of Gaussian process-based control

The work in [85] employs GPs in a RL setting to control the arm of an octopus, however, only a (rather small) discrete action space is considered. The industrial process plant for gas-liquid separator is considered in [86], [87] and [88]. In [89] the goal is to reduce air pollution through optimal GP-based nonlinear MPC for combustion plants. In [90] a probabilistic approach is presented to regulate the blood glucose level and [91] considers the perimeter

patrol problem. The work in [92] and [93] evaluates GP-based control for locomotion and low-cost robotic manipulators.

2.2.9 Extensions to Gaussian processes

Despite faster processors and improved algorithms, the high computational complexity still prevents the GP-based control techniques from being applied extensively in practical problems. This triggered research on the sparsification of GPs, see [94]–[97] and [98], which either sort out non-informative training points or replace training points by more informative pseudo inputs. A survey of these developments is provided in [99]. Further developments include local GP models [100] (see also Sec. 2.2.7) or leveraged GPs [101].

2.2.10 Summary of previous works

As this overview of the literature shows, Gaussian processes have already been applied extensively in the field of data-driven control. Nevertheless, key challenges remain unaddressed in particular considering safety-critical applications. First, available prior knowledge about the real system with respect to its convergence behavior is often ignored in literature as outlined by Challenge 1. Second, only a small fraction of the existing work, e.g. [69], [81], formally analyzes the resulting closed-loop behavior and control designs which actively avoid high-risk areas due to scarce data do not exist (Challenges 2 and 3). Furthermore, most existing approaches assume the availability of a high quality training data set without justification how this can be obtained safely (except [63]). We will approach this shortcoming (Challenge 4) with a general online learning framework.

2.3 Interpretations of Gaussian processes for modeling dynamical systems

As the review on the literature shows, there exists a wide range of applications of GPs in control, but - due to its complex nature - the interpretations of a GP model are not uniform. This section presents different views that have been employed and provides a quick discussion. For illustration, we consider an autonomous time-discrete Gaussian process state space model (GPSSM) with a one dimensional state, thus the unknown system is assumed to be of the form

$$x_{\kappa+1} = f(x_\kappa) \qquad \text{with initial state } x_0 \in \mathbb{X},$$

where $x \in \mathbb{X} \subseteq \mathbb{R}$ and $f\colon \mathbb{X} \to \mathbb{X}$.

2.3.1 Deterministic interpretation

The most simplified view on Gaussian process dynamic models is to utilize its posterior mean function as transition function, thus the model is

$$x_{\kappa+1} = \mu(x_\kappa).$$

A visualization is provided in Fig. 2.1 on the left. It ignores the uncertainty in the model (represented by $\sigma^2(\cdot)$) like e.g., in [70], [71] or utilizes it for parallel tasks, e.g., [84] adjusts the feedback gain in a computed torque controller or [102] balances a leader-follower approach. A stability analysis of this interpretation is performed in [26].

An advantage of this interpretation, is its simplicity and the identical form to the true system. Nevertheless, for a pure function approximation / regression task, alternative methods, e.g., NNs, exist, which are computationally more efficient. Therefore, the uncertainty measure should be utilized - as shown in the following section - to justify this computationally complex dynamical model.

2.3.2 Robust interpretation

Introducing assumptions regarding the complexity of $f(\cdot)$ allows to derive confidence bounds for the model prediction as shown in [103] and [104]. These bounds will in Sec. 2.4.1 be presented in detail. They guarantee that the model error $|f(\cdot)-\mu(\cdot)|$ does not exceed $\beta\sigma(\boldsymbol{x})$ with a specific probability $\delta \in (0;1)$, where $\beta \in \mathbb{R}_+$ is a function of $\mathbb{X}$, the probability δ, the number of training points N and properties of $f(\cdot)$. This allows to write the dynamical system in a robust form

$$x_{\kappa+1} = \mu(x_\kappa) \pm \beta\sigma(x_\kappa),$$

which has similarities to systems in differential inclusion [105]. This view on GPs triggered many works in control which guarantee safety, e.g., [10] and [67]. Figure 2.1 provides an illustration on the right side.

This interpretation takes full advantage of the model uncertainty measure provided by the GP and therefore allows the derivation of stability results, which is a significant step towards safe learning control. Nevertheless, the assumptions regarding the unknown function $f(\cdot)$ are often not trivial to verify and an exact computation of β is not possible efficiently. It also drops the information about the probability inside the interval and might thereby lead to a rather conservative analysis.

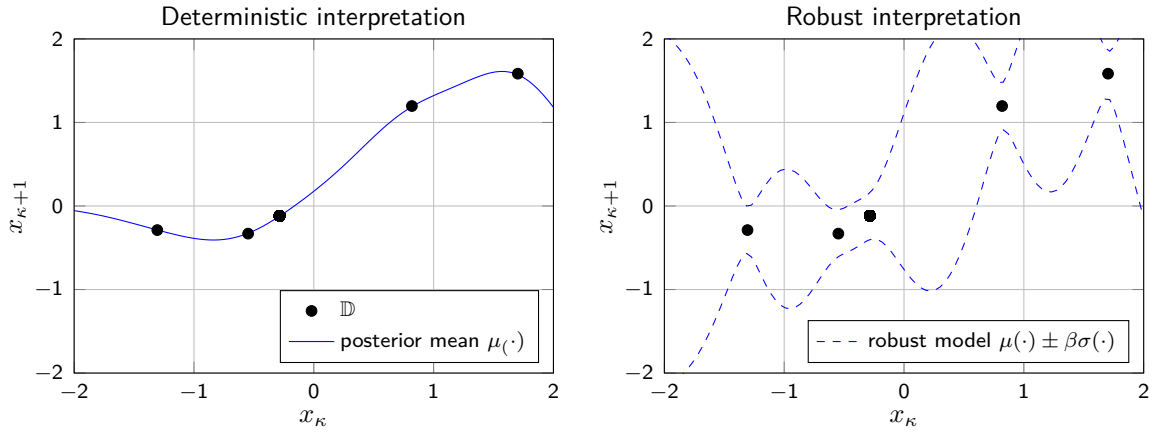

Figure 2.1: The deterministic (left) and the robust (right) interpretation of a GP model.

2.3.3 Belief-space interpretation

In the reinforcement learning literature, the *belief-space interpretation* is frequently employed, which augments the state to consist of a mean $\mu_{x_\kappa} \in \mathbb{X}$ and a variance $\sigma^2_{x_\kappa} \in \mathbb{R}_+$ and approximates it to be a Gaussian distributed random variable. This leads to the notation

$$\begin{bmatrix} \mu_{x_{\kappa+1}} \\ \sigma^2_{x_{\kappa+1}} \end{bmatrix} = \boldsymbol{f}_{\mathrm{BS}} \left(\begin{bmatrix} \mu_{x_\kappa} \\ \sigma^2_{x_\kappa} \end{bmatrix} \right),$$

where the transition function $\boldsymbol{f}_{\mathrm{BS}} : \mathbb{X} \times \mathbb{R}_+ \to \mathbb{X} \times \mathbb{R}_+$ can be computed analytically from a GP model [33]. However, this only approximates propagation of the distribution over the state for multiple time steps because the output distribution of a Gaussian process for an uncertain input is generally non-Gaussian (due to the non-linearity). As an alternative to an exact analytic solution, moment matching computes the first and second moment of the output distribution for a Gaussian input distribution exactly and thereby derives $\boldsymbol{f}_{\mathrm{BS}}(\cdot, \cdot)$. A visualization is provided in Fig. 2.2.

An improved approximation technique is presented in [40], but there does not exist any guarantee on how far the approximated propagated distribution is from the real distribution. It is therefore not suitable for a profound stability analysis of control laws. Furthermore, the approximation is computationally very demanding and its probabilistic nature does match the true system $f(\cdot)$, which is deterministic.

2.3.4 Stochastic interpretation

Given the current state x_κ, the GP assigns a Gaussian distribution to the next state, based on the inferred mean and variance. This leads to the interpretation as a stochastic dynamical system, which can be written in two ways

$$x_{\kappa+1} \sim \mathcal{N}\left(\mu(x_\kappa), \sigma^2(x_\kappa)\right), \qquad x_{\kappa+1} = \mu(x_\kappa) + \sigma(x_\kappa)\omega, \text{ with i.i.d } \omega \sim \mathcal{N}(0, 1).$$

This formulation allows an analysis of the stability of GP models using stochastic stability tools as it is performed in [27] and [7].

However, since the sampling in each step is independent from the realizations in previous steps, it can lead to different function values for the same input. This ignores the fact that

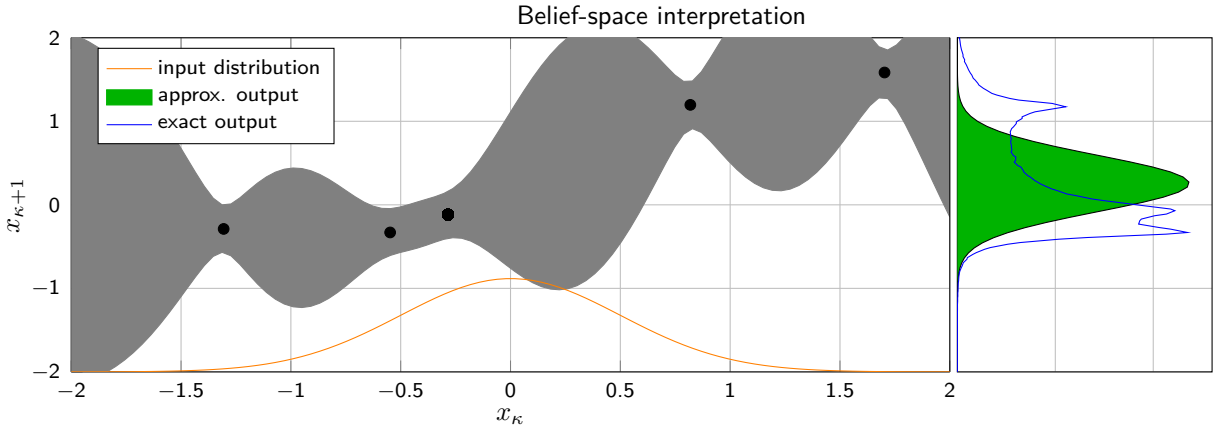

Figure 2.2: The belief-space interpretation of a GP model.

the real system is a single true function. Furthermore, the covariance between states is ignored and therefore the sampled function does not fulfill smoothness assumptions for the GP imposed by the kernel. This is illustrated in Fig. 2.3 on the left.

2.3.5 Scenario interpretation

Following the interpretation of the GP as distribution over functions, [106] proposes to sample a single realization of the GP in advance which is then used to propagate the state deterministically. This is formalized as

$$f_{\mathcal{GP}} \sim \mathcal{GP}(\mathfrak{m}(\cdot), \mathfrak{k}(\cdot,\cdot)), \qquad x_{\kappa+1} = f_{\mathcal{GP}}(x_\kappa).$$

Since each sample of a GP can be considered as a scenario, this concept is named *scenario interpretation* and is analyzed in [107]. Furthermore, sampling one function as a scenario overcomes the difficulties mentioned for the stochastic interpretation in Sec. 2.3.4 because the drawn function is deterministic and follows the smoothness assumption imposed by the kernel. A visualization is provided in Fig. 2.3 on the right.

However - due to its nonparametric nature - sampling a realization of the GP at once and storing it is impossible in practice because it is an infinite dimensional object. The sequential sampling proposed in [106], which iteratively conditions the sampling of the next state on all previous states, leads to a non-Markovian system for which applicable analysis tools are currently not available.

This overview shows that the various interpretations of GP dynamical models are used in literature and all have their right to exist. A side by side comparison has so far only been performed by the author in [107] and therefore delivers a notable contribution to the GP literature for control. For each main contributions of this thesis, we will clearly indicate on which interpretation of GPs we base our design and analysis.

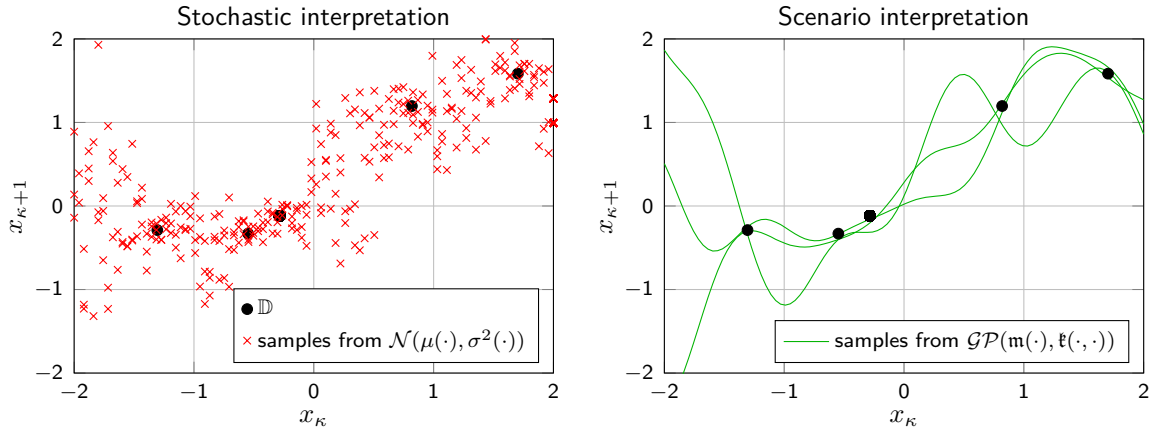

Figure 2.3: The stochastic (left) and the scenario (right) interpretation of a GP model.

2.4 Properties and bounds for Gaussian processes

Gaussian processes are well researched and there exist numerous literature on their properties from the fields of machine learning, statistics and information theory. However, only a few results are relevant for this thesis, which are reviewed in the following.

2.4.1 Error bounds for Gaussian process models

Gaussian processes are particularly appealing for applications in control of safety-critical systems because a formal analysis is analytically possible. As mentioned in Sec. 2.3.2, it is possible to bound the error between the posterior mean function $\mu(\cdot)$ and the function $f(\cdot)$ of which the measurements are taken with high probability. However, such a guarantee does not come without any assumptions on $f(\cdot)$ as the no-free-lunch theorem suggest [108].

The assumption does - thanks to the nonparametric nature of the GP - not require any structural knowledge on $f(\cdot)$, like parametric regression techniques do. But it limits the complexity of the function as measured by the reproducing kernel Hilbert space (RKHS) as formulated in the following.

Assumption 2.1. *The function $f\colon \mathbb{X} \to \mathbb{R}$ has a bounded RKHS norm with respect to a known kernel $\mathfrak{k}(\cdot,\cdot)$, denoted by $\|f(\cdot)\|_{\mathfrak{k}}^2 \leq B_f$.*

A RKHS is a complete subspace of the $\mathbb{L}_2$, for which the inner product $\langle \cdot, \cdot \rangle_{\mathfrak{k}}$ fulfills the reproducing property $\langle f, \mathfrak{k}(\boldsymbol{x}, \cdot)\rangle_{\mathfrak{k}} = f(\boldsymbol{x})$. The induced norm $\|f\|_{\mathfrak{k}} = \sqrt{\langle f, f\rangle_{\mathfrak{k}}}$ measures the smoothness of $f(\cdot)$. In most cases (for universal differentiable kernels), Assumption 2.1 translates in practice to assuming a continuously differentiable dynamic behavior. This excludes for example impacts which lead to switching dynamics e.g. a bouncing ball or robotic interaction tasks.

Furthermore the information gain which can be obtained on a compact set $\mathbb{X}$ from $N+1$ noisy measurements $\boldsymbol{x}_{(1)}, \ldots, \boldsymbol{x}_{(N+1)}$ is defined as

$$\gamma = \max_{\boldsymbol{x}^{(1:N+1)} \in \mathbb{X}} \frac{1}{2} \log\det\left(\boldsymbol{I}_{N+1} + \frac{1}{\sigma_{\text{on}}^2}\boldsymbol{K}\right), \quad \text{with } \boldsymbol{K} = \mathfrak{k}\left(\boldsymbol{x}^{(1:N+1)}, \boldsymbol{x}^{(1:N+1)}\right). \tag{2.9}$$

This allows to derive a bound for the model error as following.

Theorem 2.1. *Consider a compact set $\mathbb{X} \subset \mathbb{R}^n$, a function $f\colon \mathbb{X} \to \mathbb{R}$ under Assumption 2.1 and a probability $\delta \in (0;1)$. Then,*

$$\mathcal{P}\left\{|\mu(\boldsymbol{x}) - f(\boldsymbol{x})| \leq \beta\sigma(\boldsymbol{x}), \forall \boldsymbol{x} \in \mathbb{X}, N \in \mathbb{N}_0\right\} \geq 1 - \delta, \tag{2.10}$$

where $\beta = \sqrt{2B_f + 300\gamma \log^3((N+1)/\delta)}$ and $\mu(\cdot)$, $\sigma(\cdot)$ are posterior mean / variance function of a GP for N data points as defined in (2.4)

Proof. This result is stated and proven in [104, Theorem 6][1] □

[1] The constant 300 is not the exact number as it results from the derivation of the bound but is utilized by the authors of [104] as conservative abbreviation of a constant factor for notational convenience.

The maximum information gain γ grows sublinearly (for most common kernels) with the number of data points N and so does in consequence β, see [104] for more details. Thus, the question whether the error bound becomes tighter, i.e. $\sigma(\cdot)$ decreases faster than β increases with more training data points, cannot be answered in general but depends on the distribution of the data points across $\mathbb{X}$.

Theorem 2.1 can be extended to multiple dimensions using the union bound.

Proposition 2.1. *Consider a compact set $\mathbb{X} \subset \mathbb{R}^n$, a function $\boldsymbol{f}\colon \mathbb{X} \to \mathbb{R}^m$ for which the RKHS norms are bounded $\|f_j(\cdot)\|_{\mathfrak{k}_j}^2 \leq B_{f_j}$, $\forall j = 1, \dots, m$ and probabilities $\delta_j \in (0;1)$ for which $\sum_{j=1}^m \delta_j < 1$. Then,*

$$\mathcal{P}\left\{|\boldsymbol{\mu}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x})| \leq \|\boldsymbol{\beta}\|\|\boldsymbol{\sigma}(\boldsymbol{x})\|, \forall \boldsymbol{x} \in \mathbb{X}, N \in \mathbb{N}_0\right\} \geq 1 - \sum_{j=1}^{m} \delta_j, \tag{2.11}$$

where $\boldsymbol{\beta} = [\beta_1 \ \cdots \ \beta_m]$ with $\beta_j = \sqrt{2B_{f_j} + 300\gamma_j \log^3((N+1)/\delta_j)}$, $\boldsymbol{\mu}(\cdot)$, $\boldsymbol{\sigma}(\cdot)$ are posterior mean / standard deviation functions of a GP for N data points as defined in (2.7) and γ_j is the information gain defined in (2.9) for each kernel $\mathfrak{k}_j(\cdot,\cdot)$, $j = 1, \dots, m$.

Proof. With the result from Theorem 2.1, the union bound yields

$$\begin{aligned} 1 - \sum_{j=1}^{m} \delta_j &\leq \mathcal{P}\left\{\bigcap_{j=1}^{m} |\mu_j(\boldsymbol{x}) - f_j(\boldsymbol{x})| \leq \beta_j \sigma_j(\boldsymbol{x}), \forall \boldsymbol{x} \in \mathbb{X}\right\} \\ &\leq \mathcal{P}\left\{\|\boldsymbol{\mu}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x})\| \leq \left\| \begin{bmatrix} \beta_1 \sigma_1(\boldsymbol{x}) \\ \vdots \\ \beta_m \sigma_m(\boldsymbol{x}) \end{bmatrix} \right\|, \forall \boldsymbol{x} \in \mathbb{X}\right\} \end{aligned}$$

and using the triangle and the Cauchy-Schwarz inequality

$$\left\| \begin{bmatrix} \beta_1 \sigma_1(\boldsymbol{x}) & \cdots & \beta_m \sigma_m(\boldsymbol{x}) \end{bmatrix}^\intercal \right\| \leq \sum_{j=1}^{m} \beta_j \sigma_j(\boldsymbol{x}) \leq \|\boldsymbol{\beta}\|\|\boldsymbol{\sigma}(\boldsymbol{x})\|,$$

yields the presented result. □

Remark 2.4. *This bound is very powerful, since the probability δ not just holds for individual $\boldsymbol{x}$ values, but addresses the entire function sample $f(\cdot)$ and thereby all $\boldsymbol{x} \in \mathbb{X}$. Furthermore, it holds for all numbers of training data N, thus adding or removing data points does not affect the probability that the bound is violated.*

However, it must be admitted that the RKHS norm of an unknown is not trivial to obtain for a real system and computing an exact value for γ is difficult. Nevertheless, to verify Assumption 2.1, only a bound of the RKHS norm is required and upper limits on γ can be obtained computationally efficient.

Furthermore, alternative bounds for which the conditions are simpler to verify are developed in [103], where the author is also contributing.

2.4.2 Posterior variance limits

As shown in Theorem 2.1, the variance function plays a crucial role to bound the model error. We therefore review here a property of the variance function, which will be useful in the remainder of this thesis.

Lemma 2.1. *Consider a Gaussian process with noise free observations $\sigma_{on}^2 = 0$ and SE kernel (2.5). Then, for the posterior variance function $\sigma^2(\cdot)$ in (2.4) it holds that*

$$\sigma^2(\boldsymbol{x}) \leq \zeta^2 \left(1 - \exp\left(\frac{-\|\boldsymbol{x} - \boldsymbol{x}^{(i)}\|^2}{\underline{\ell}^2}\right)\right) \coloneqq \bar{\sigma}^2_{\boldsymbol{x}^{(i)}}(\boldsymbol{x}) \qquad \begin{matrix} \forall i = 1, \ldots, N \\ \forall \boldsymbol{x} \in \mathbb{X}, \end{matrix}$$

where $\underline{\ell}^2 = \min_i \ell_i^2$ is the shortest lengthscale and $\sigma^2_{\boldsymbol{x}^{(i)}}\left(\boldsymbol{x}^{(i)}\right) = 0$, $\forall i = 1, \ldots, N$.

Proof. Given constant hyperparameters, $\sigma^2(\cdot)$ decreases globally point-wise with any additional training point as derived in [109, Section 2.4]. Therefore, the posterior variance function of GP with a dataset which contains the point $\boldsymbol{x}^{(i)}$ can be upper bounded by the posterior variance function for a dataset which *only* contains the point $\boldsymbol{x}^{(i)}$. Substituting $\mathfrak{k}_{\mathrm{SE}}(\boldsymbol{x}, \boldsymbol{x}) = \zeta^2$, which holds $\forall \boldsymbol{x}$, results in

$$\sigma^2(\boldsymbol{x}) \leq \zeta^2 - \zeta^2 \exp\left(-\sum_{j=1}^{n} \frac{\left(x_j - x_j^{(i)}\right)^2}{2\ell_j^2}\right)^2 \leq \zeta^2 \left(1 - \exp\left(\frac{-\|\boldsymbol{x} - \boldsymbol{x}^{(i)}\|^2}{\underline{\ell}^2}\right)\right)$$

for all $j = 1, \ldots, n$. This yields the proposed upper bound. □

Example 2.1. *For $n = 1$, consider a SE kernel with $\zeta^2 = 1$ and $\ell^2 = 0.16$ and the training data set $\mathbb{D} = \{(-0.1, 0), (0.5, 0), (1, 0)\}$. Then, the resulting posterior variance function of a GP and the upper bound derived in Lemma 2.1 is illustrated in Fig. 2.4.*

Further analysis of the posterior variance is provided in [110], where the author is also contributing.

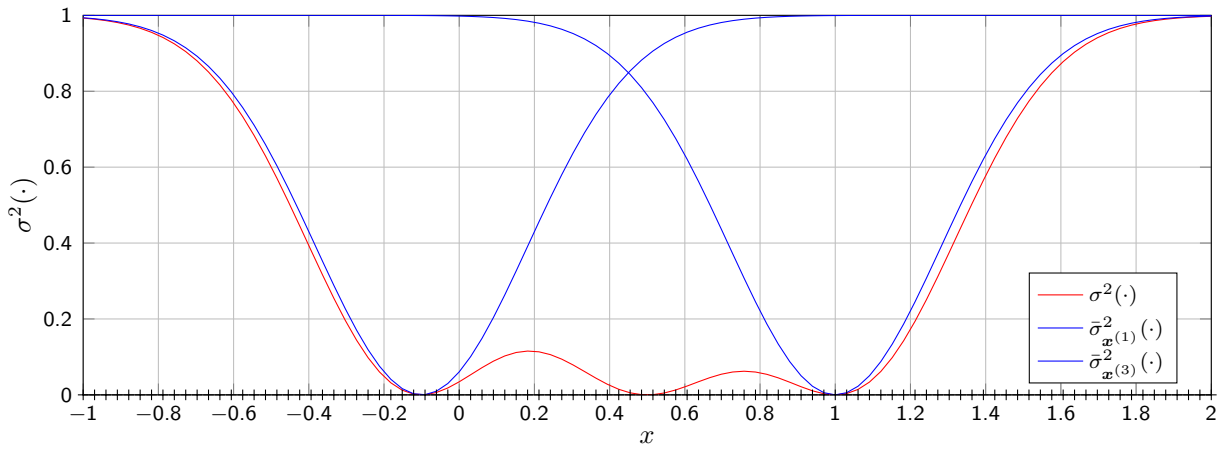

Figure 2.4: Illustration of Lemma 2.1: The posterior variance function $\sigma^2(\cdot)$ is shown in red, two possible upper bounds, $\bar{\sigma}^2_{\boldsymbol{x}^{(1)}}(\cdot)$ and $\bar{\sigma}^2_{\boldsymbol{x}^{(3)}}(\cdot)$ are shown in blue. The training data is $\boldsymbol{x}^{(1)} = -0.1$, $\boldsymbol{x}^{(2)} = 0.5$ and $\boldsymbol{x}^{(3)} = 1$.

3 Identification of Stable Systems

Data-driven modeling techniques are very powerful for systems where no analytic model can be derived. However, the lack of an analytic description does not imply that no prior knowledge about the system is given. In particular, for many physical systems, so called high-level knowledge is accessible, e.g., balance of in and out flow in a node, smoothness of the dynamics or the energy dissipation [111]. These fundamental properties are often not considered in data-driven modeling and the resulting models thereby make physically inconsistent predictions.

This work focuses on the consistent energy dissipation between the true system and the data-driven model, which is closely entangled with the convergence behavior of a dynamical system. Utilizing such high-level prior knowledge is not only helpful for increasing the model precision but also crucial as without any prior knowledge no generalization outside of the training points can be expected (see no-free-lunch theorems [108]).

The presented techniques can not only be employed for physically consistent modeling, but are also applicable in human-like motion generation: Assume a goal-directed human movement is modeled by a dynamical system in a robotic learning by demonstration scenario. Then, the training data converges to the desired goal point and this behavior must also be represented by the model. Thus, the model must represent the demonstrated motion accurately and ensure that all generated trajectories converge to the goal point.

Ensuring physically consistent prediction or stability of a parametric model is rather simple to verify, see [112] and [113]. Other classical system identification techniques, e.g. autoregressive–moving-average (ARMA) models rely on subspace methods to ensure stability [114], [115]. The deconvolution problem, to find the impulse response given input-output data, is approached using regularization techniques as discussed in [116] and recent overviews are given in [117] and [118]. This problem is also considered by the machine learning community using kernel-based techniques to identify the impulse response, see [119] and [120].

For the nonlinear case, Volterra series or Wiener-Hammerstein models exist, which consider a very limited structure of the model [16]. Therefore, supervised learning methods, e.g., NNs [121] or GPSSMs have gained attention, see [122] and [21]. However, an analysis of the system stability is missing in these studies. A first GPSSM stability analysis is performed in [26] and [27], for the deterministic and the stochastic interpretation, respectively. Enforcing stability to Gaussian mixture models (GMMs) is studied in [123] and [124] and more general techniques are developed in [125], [126] and [127].

While these approaches aim to incorporate stability into a model, none of them deals with the inherent challenge of data-driven approaches that data is usually sparse and the resulting models are imprecise. The uncertainty resulting from finite data is commonly ignored. Therefore, this chapter develops a framework to deal with these uncertainties

in form of a stochastic dynamical system and ensures its asymptotic convergence using stochastic stability theory [128].

The main contribution is a novel identification algorithm to learn asymptotically stable GPSSMs using control Lyapunov functions. For the deterministic interpretation of the GP, we show a realization for arbitrary datasets and prove that the model is improved through the stabilization. For the stochastic interpretation, we derive conditions for almost sure asymptotic convergence and show how many additional training data are required to ensure asymptotic stability if these conditions are not fulfilled on the initial dataset. To learn the convergence behavior in a data-driven fashion, we propose the use of a sum of squares (SOS) control Lyapunov function. This allows a computationally efficient estimation of unknown equilibria and we demonstrate its advantages in simulation over alternative Lyapunov candidates on a real-world dataset.

The chapter is based on the work published in [7] and [8]. It is structured as follows: After defining the problem setting in Sec. 3.1, this chapter proposes in Sec. 3.2 an optimization-based stabilization of a GPSSM for the deterministic and the stochastic interpretation (as introduced in Sec. 2.3). A data-driven search for a suitable control Lyapunov function is presented in Sec. 3.3 followed by a numerical evaluation in Sec. 3.4 and a discussion in Sec. 3.5.

3.1 Problem formulation

Here, we consider an unknown discrete-time system with state $\boldsymbol{x} \in \mathbb{X} \subseteq \mathbb{R}^n$, $n \in \mathbb{N}$, given by

$$\boldsymbol{x}_{\kappa+1} = \boldsymbol{f}(\boldsymbol{x}_\kappa), \tag{3.1}$$

with initial condition $\boldsymbol{x}_0 \in \mathbb{X}$, where $\kappa \in \mathbb{N}$ and $\boldsymbol{f} : \mathbb{X} \to \mathbb{X}$. The following is assumed.

Assumption 3.1. *The function $\boldsymbol{f}(\cdot)$ is continuously differentiable but unknown.*

Remark 3.1. *The smoothness of the function $\boldsymbol{f}(\cdot)$ is a quite natural assumption and holds for a large class of systems. It is also an essential one because the generalization outside of a training data set becomes very difficult for discontinuous functions [108].*

The available training data set is assumed to take the following form

Assumption 3.2. *The training set of N data pairs consists of consecutive measurements of the state*

$$\mathbb{D} = \left\{ \left(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)} \right) \right\}_{i=1}^{N},$$

where $\boldsymbol{y}^{(i)} = \boldsymbol{x}_{\kappa+1}^{(i)}$ is the consecutive state to $\boldsymbol{x}_\kappa^{(i)}$![1]

Remark 3.2. *The data does not necessarily stem from a single trajectory. Each pair can be taken independently from other pairs and the order in the training set is not decisive. A visualization of a possible data set is provided in Fig. 3.1*

Furthermore, we make an assumption on the asymptotic behavior of the system (3.1).

[1] An extension to noisy measurements of the consecutive state is directly possible. However, we consider it as an unrealistic setting to have noise on the consecutive state but not on the current state (as the former becomes the latter in the next time step).

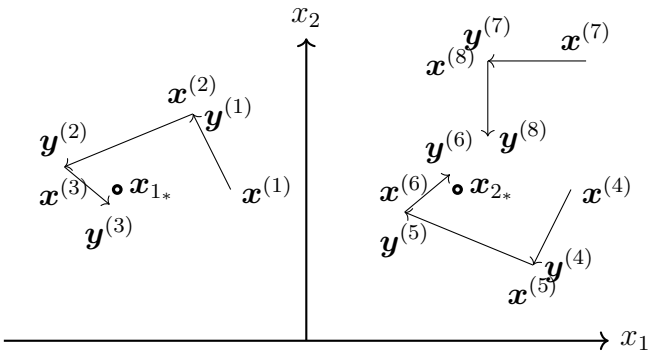

Figure 3.1: An illustration of a training dataset $\mathbb{D}$ with two equilibria $\boldsymbol{x}_{1_*}$,$\boldsymbol{x}_{2_*}$. The data originate from three different trajectories (1-2-3,4-5-6,7-8). Within one trajectory, the end point of one step is the starting point of the next, e.g., $\boldsymbol{y}^{(1)} = \boldsymbol{x}^{(2)}$, $\boldsymbol{y}^{(4)} = \boldsymbol{x}^{(5)}$, etc., but this is not necessarily the case.

Assumption 3.3. *There exist N_* unknown equilibria, denoted as $\boldsymbol{x}_{i_*} \in \mathbb{X}_{i_*} \subseteq \mathbb{X}$, where $i_* = 1, \ldots, N_*$ and $\boldsymbol{f}(\boldsymbol{x}_{i_*}) = \boldsymbol{x}_{i_*}$. Each of the equilibria is asymptotically stable with corresponding known domain of attraction $\mathbb{X}_{i_*} \subseteq \mathbb{X}$ for which holds*

$$\mathbb{X}_{i_*} \cap \mathbb{X}_{i'_*} = \emptyset, \quad \forall i'_* \neq i_*, \qquad \textit{and} \qquad \bigcup_{i_*=1}^{N_*} \mathbb{X}_{i_*} = \mathbb{X}.$$

This formulates the main motivation of this chapter. A physical system whose dynamics are unknown, dissipates energy and will eventually reach an equilibrium point. This equilibrium is not known and it is also tedious to determine it experimentally (since it takes usually infinite time to be reached). In contrast, the region of attraction is commonly easier to find.

Example 3.1. *Consider a ball filled with water, making its dynamics difficult to describe analytically. This ball is dropped over hilly ground as illustrated in Fig. 3.2. It is obvious in which valley the ball will role, given the starting position, $\mathbb{X}_{1_*}$, $\mathbb{X}_{2_*}$ or $\mathbb{X}_{3_*}$ (ignoring the unstable equilibria on top the hills). But it cannot be observed (in finite time) where it comes to rest due to the friction (in particular for the equilibrium $\boldsymbol{x}_{2_*}$).*

Assumption 3.3 therefore comes with two fundamental challenges to a data-driven identification. First, the convergence behavior must be estimated from data, which includes to find the equilibria and a function which describes the dissipation of the energy (here, a Lyapunov function). Second, this convergence behavior must be enforced in the model to ensure it is consistent in terms of this prior knowledge on stability. This concept is visualized in Fig. 3.3.

The goal is to learn a GPSSM $\hat{\boldsymbol{f}} \colon \mathbb{X} \to \mathbb{X}$ for the unknown true dynamics (3.1), denoted by

$$\boldsymbol{x}_{\kappa+1} = \hat{\boldsymbol{f}}(\boldsymbol{x}_\kappa). \tag{3.2}$$

We consider two different cases here according to the *deterministic* and the *stochastic* interpretation as presented in Sec. 2.3. For both, the prior knowledge given by Assumption 3.3 on the stability is generally not fulfilled by standard GP models [26], [27]. Therefore, we propose

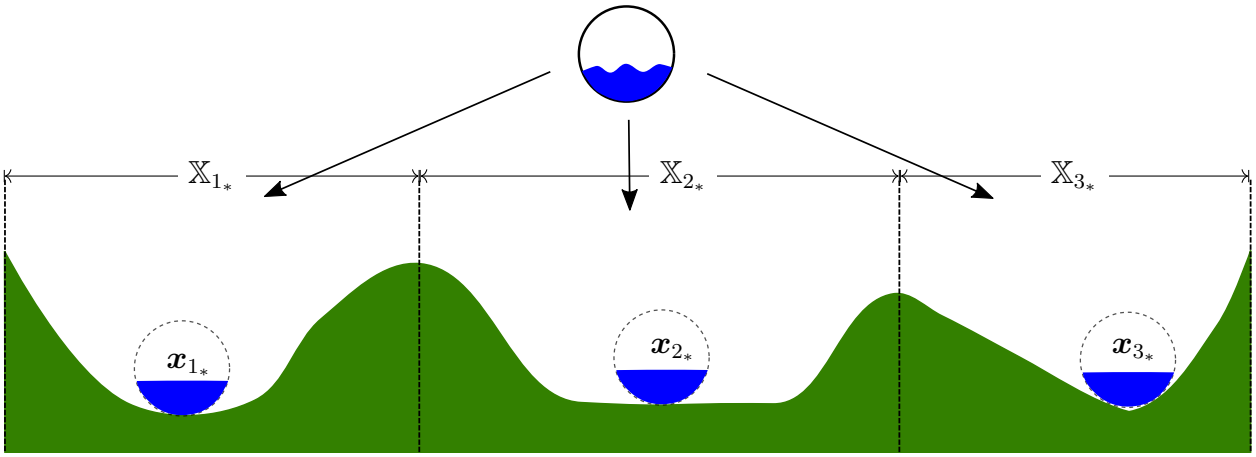

Figure 3.2: An illustration of multiple stable equilibria.

to augment the GPSSM by a stabilization signal $\boldsymbol{u}\colon \mathbb{X} \to \mathbb{R}^n$. Despite the notation, $\boldsymbol{u}(\cdot)$ is not an external control input, but belongs internally to the model

$$\hat{\boldsymbol{f}}(\boldsymbol{x}_\kappa) \coloneqq \boldsymbol{f}_{\mathcal{GP}}(\boldsymbol{x}_\kappa) + \boldsymbol{u}(\boldsymbol{x}_\kappa), \tag{3.3}$$

where $\boldsymbol{f}_{\mathcal{GP}}\colon \mathbb{X} \to \mathbb{X}$ denotes a standard GPSSM without any stabilization and $\hat{\boldsymbol{f}}\colon \mathbb{X} \to \mathbb{X}$ the proposed stabilized GPSSM. We aim to ensure the model $\hat{\boldsymbol{f}}(\cdot)$ follows Assumption 3.3. This requires the estimation of the equilibria, which we will denote here as $\hat{\boldsymbol{x}}_{i_*}$, $i_* = 1, \ldots, N_*$. Furthermore, it must be ensured that these equilibria are stable with the given region of attractions $\mathbb{X}_{i_*}$. We therefore formulate the objective for the *deterministic* GP as follows.

Objective 3.1. *Find a stabilization signal $\boldsymbol{u}(\cdot)$ which ensures that the model*

$$\boldsymbol{x}_{\kappa+1} = \boldsymbol{\mu}(\boldsymbol{x}_\kappa) + \boldsymbol{u}(\boldsymbol{x}_\kappa) =: \bar{\boldsymbol{f}}(\boldsymbol{x}_\kappa), \tag{3.4}$$

has N_* asymptotically stable equilibria $\hat{\boldsymbol{x}}_{i_*} \in \mathbb{X}_{i_*}$ with respective domains of attraction $\mathbb{X}_{i_*}$.

Thus, we aim for a model, which has the same asymptotic convergence behavior as the true system (3.1). Taking the posterior mean function of the GP, we obtain the GP model, which is the most likely one after observing the data $\mathbb{D}$. However, since the data is sparse, a high model precision cannot be expected everywhere in the state space $\mathbb{X}$. Therefore, we want to make use of the model fidelity provided by the GP model. To do so we investigate the *probabilistic* GP model by adding process noise into the model (which is not present in the true system).

Remark 3.3. *Including noise into the model ensures that it does not provide overconfident predictions when it operates far from the training data. Such a stochastic model thereby increases the safety when transferring simulation results to the real-world experiment. Furthermore, it allows to apply risk-sensitive [129], uncertainty aware (Chapter 4) or path integral [130] control techniques.*

Nevertheless, we aim to implement the prior knowledge on the convergence behavior (Assumption 3.3) into this stochastic model, which is formulated as following.

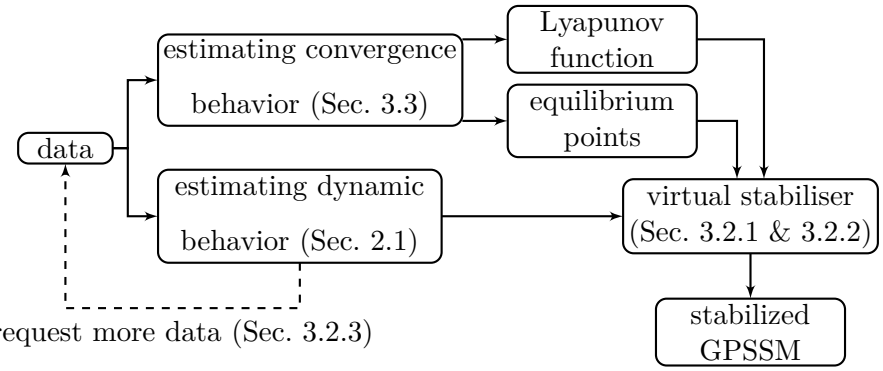

Figure 3.3: An overview of the proposed scheme for a data-driven stabilization of GPSSMs.

Objective 3.2. *Find a stabilization signal* $\boldsymbol{u}(\cdot)$ *such that the model*

$$\boldsymbol{x}_{\kappa+1} = \boldsymbol{\mu}(\boldsymbol{x}_\kappa) + \boldsymbol{u}(\boldsymbol{x}_\kappa) + \sqrt{\boldsymbol{\Sigma}(\boldsymbol{x}_\kappa)}\boldsymbol{\omega}_\kappa = \bar{\boldsymbol{f}}(\boldsymbol{x}_\kappa) + \sqrt{\boldsymbol{\Sigma}(\boldsymbol{x}_\kappa)}\boldsymbol{\omega}_\kappa \quad \boldsymbol{\omega}_\kappa \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_n), \tag{3.5}$$

has N_* *almost surely (a.s.) asymptotically stable equilibria* $\hat{\boldsymbol{x}}_{i_*} \in \mathbb{X}_{i_*}$, $i_* = 1, \ldots, N_*$. *Thus, for all* $\boldsymbol{x}_0 \in \mathbb{X}_{i_*}$ *it holds that*

$$\lim_{k\to\infty} \mathcal{P}(\|\boldsymbol{x}_\kappa - \hat{\boldsymbol{x}}_{i_*}\| = 0) = 1, \quad \forall i_* = 1, \ldots, N_*. \tag{3.6}$$

The i.i.d. random variable $\boldsymbol{\omega}_\kappa \in \mathbb{R}^n$ originates from the probability space $(\Omega, \mathcal{F}, \mathfrak{P})$, which has the sample space $\Omega = \mathbb{R}^n$ and the sigma-algebra $\mathcal{F}$ of Borel sets on Ω. The probability measure $\mathfrak{P}$ is a normal distribution.

This chapter presents an algorithm which fulfills Objectives 3.1 and 3.2 using the augmented GP model (3.3).

3.2 Stabilized Gaussian process state space models

In this section, we elaborate how stability of a GPSSM is enforced with the stabilization signal $\boldsymbol{u}(\cdot)$ using a given control Lyapunov function.

The first step towards Objectives 3.1 and 3.2 is to ensure that the GP mean function estimate $\boldsymbol{\mu}(\cdot)$ has a fixed point at the given equilibria estimates $\hat{\boldsymbol{x}}_{i_*}$, thus $\hat{\boldsymbol{x}}_{i_*} = \boldsymbol{\mu}(\hat{\boldsymbol{x}}_{i_*})$ for all $i_* = 1, \ldots, N_*$. This can be achieved as described in Remark 2.3, and will therefore not further be discussed here. However, this will not ensure the asymptotic stability of these equilibria or ensure the proper domain of attraction. To make the convergence behavior of the model match the true system, this section discusses the choice of the internal stabilizing signal $\boldsymbol{u} \in \mathbb{R}^n$ in $\hat{\boldsymbol{f}}(\boldsymbol{x})$.

We assume to be given N_* control Lyapunov functions $V^{i_*}_{\boldsymbol{\theta}_{i_*}} : \mathbb{X} \to \mathbb{R}_{+,0}$ which are all parameterized by $\boldsymbol{\theta}_{i_*} \in \boldsymbol{\Theta}_{i_*} \subseteq \mathbb{R}^{n_{\theta_i}}$, $n_{\theta_i} \in \mathbb{N}$ and an estimated equilibrium $\hat{\boldsymbol{x}}_{i_*} \in \mathbb{X}_{i_*}$ (see Sec. 3.3 how the parameters are obtained from data). The following assumptions are made.

Assumption 3.4. *For all parameter choices* $\boldsymbol{\theta}_{i_*} \in \boldsymbol{\Theta}_{i_*}$, *the functions* $V^{i_*}_{\boldsymbol{\theta}_{i_*}}(\cdot)$ *are continuous and positive definite, thus*

$$V^{i_*}_{\boldsymbol{\theta}_{i_*}}(\boldsymbol{x}) > 0, \ \forall \boldsymbol{x} \in \mathbb{X}\backslash\{\hat{\boldsymbol{x}}_{i_*}\} \quad \textit{and} \quad V^{i_*}_{\boldsymbol{\theta}_{i_*}}(\hat{\boldsymbol{x}}_{i_*}) = 0, \qquad \forall i_* = 1, \ldots, N_*.$$

Assumption 3.5. *The functions $V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\cdot)$ are radially unbounded*

$$\lim_{\|\boldsymbol{x}-\hat{\boldsymbol{x}}_{i_*}\|\to\infty} V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\boldsymbol{x}) = \infty, \quad \forall \boldsymbol{\theta}_{i_*} \in \boldsymbol{\Theta}_{i_*}.$$

These assumptions make $V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\cdot)$ Lyapunov candidates. We consider the *deterministic* interpretation of the GP first in Sec. 3.2.1 before dealing with the *stochastic* interpretation in Sec. 3.2.2 and 3.2.3.

3.2.1 Deterministic interpretation

For the deterministic interpretation of a GPSSM, the next state, given the current state, is obtained from $\boldsymbol{x}_{\kappa+1} = \boldsymbol{\mu}(\boldsymbol{x}_\kappa)$. Using Remark 2.3 the estimated equilibria are incorporated, but their asymptotic stability is not ensured. Therefore, the following optimization-based stabilization is proposed.

Theorem 3.1. *Consider the GP* (2.7)/(2.8) *with SE kernel* (2.5) *and the stabilizing command $\boldsymbol{u}^*(\cdot)$ obtained from the optimization*

$$\boldsymbol{u}^*(\boldsymbol{x}_\kappa) = \arg\min_{\boldsymbol{u}} \frac{1}{2}\boldsymbol{u}^\intercal \boldsymbol{u}, \tag{3.7a}$$

$$\text{s.t. } V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\boldsymbol{\mu}(\boldsymbol{x}_\kappa)+\boldsymbol{u}) - V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\boldsymbol{x}_\kappa) < 0 \quad \text{if } \boldsymbol{x}_\kappa \neq \hat{\boldsymbol{x}}_{i_*}, \quad \text{and } \boldsymbol{u} = \boldsymbol{0} \quad \text{if } \boldsymbol{x}_\kappa = \hat{\boldsymbol{x}}_{i_*}, \tag{3.7b}$$

where i_* is chosen such that $\boldsymbol{x}_0 \in \mathbb{X}_{i_*}$ and $V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\cdot)$ is a Lyapunov candidate which fulfills Assumption 3.4. Then, the model

$$\boldsymbol{x}_{\kappa+1} = \bar{\boldsymbol{f}}(\boldsymbol{x}_\kappa) = \boldsymbol{\mu}(\boldsymbol{x}_\kappa) + \boldsymbol{u}^*(\boldsymbol{x}_\kappa), \tag{3.8}$$

converges asymptotically to the equilibrium $\hat{\boldsymbol{x}}_{i_*}$ for all $\boldsymbol{x}_0 \in \mathbb{X}_{i_*}$.

Proof. The optimization (3.7) ensures that the Lyapunov function $V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\cdot)$ decreases in every step $V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\bar{\boldsymbol{f}}(\boldsymbol{x}_\kappa)) - V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\boldsymbol{x}_\kappa) < 0$, $\forall \boldsymbol{x}_\kappa \in \mathbb{X} \setminus \{\hat{\boldsymbol{x}}_{i_*}\}$. The constraint set is not empty $\forall \boldsymbol{x}_\kappa \in \mathbb{X}$, since a feasible solution $\boldsymbol{u} = \hat{\boldsymbol{x}}_{i_*} - \boldsymbol{\mu}(\hat{\boldsymbol{x}}_{i_*})$ always exists with $V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\hat{\boldsymbol{x}}_{i_*}) - V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\boldsymbol{x}_\kappa) = -V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\boldsymbol{x}_\kappa)$ being negative definite. □

An illustration of this optimization-based stabilization is provided in Fig. 3.4 on the left side. For a single equilibrium point, this result can directly be extended to global stability.

Corollary 3.1. *Let $N_* = 1$, $V_{\boldsymbol{\theta}}(\cdot)$ is radially unbounded (Assumption 3.5), and $\mathbb{X} = \mathbb{R}^n$. Furthermore, consider a GP* (2.7)/(2.8) *with SE kernel* (2.5) *and the stabilizing command $\boldsymbol{u}^*(\cdot)$ proposed in* (3.7)*. Then, the equilibrium $\hat{\boldsymbol{x}}_*$ of the model* (3.8) *is globally asymptotically stable (GAS).*

Proof. Since the Lyapunov function is radially unbounded and decreasing over time (compare Theorem 3.1) the necessary conditions for *global* stability hold [131]. □

Thus, with the optimization-based choice in (3.7), Objective 3.1 is achieved.

Remark 3.4. *There are many choices for $\boldsymbol{u}(\cdot)$ which would fulfill Objective 3.1 and the most trivial is $\boldsymbol{u} = \hat{\boldsymbol{x}}_{i_*} - \boldsymbol{\mu}(\hat{\boldsymbol{x}}_{i_*})$. However, we are not just trying to stabilize the model, but we aim to replicate the true system* (3.1) *as precise as possible. Therefore the GPSSM $\boldsymbol{\mu}(\cdot)$ should be distorted only minimally because it represents the data optimal (according to the likelihood optimization).*

It can be shown that (for a convex Lyapunov function) the GPSSM without stabilization $\boldsymbol{\mu}(\cdot)$ performs never better - in terms of prediction precision - than the proposed stabilized GPSSM $\bar{\boldsymbol{f}}(\cdot) = \boldsymbol{\mu}(\cdot) + \boldsymbol{u}(\cdot)$.

Proposition 3.1. *Consider the GP* (2.7)/(2.8) *with SE kernel* (2.5) *and convex Lyapunov functions $V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\cdot)$ which fulfill Assumption 3.4 and $V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\boldsymbol{f}(\boldsymbol{x})) - V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\boldsymbol{x}) < 0$, $\forall \boldsymbol{x} \in \mathbb{X}_{i_*} \setminus \{\hat{\boldsymbol{x}}_{i_*}\}$, and $\forall i_*$. Then, the prediction error of the GPSSM without stabilization $\boldsymbol{\mu}(\cdot)$ is never smaller than that of the stabilized GPSSM $\bar{\boldsymbol{f}}(\cdot)$, thus*

$$\|\bar{\boldsymbol{f}}(\boldsymbol{x}_\kappa) - \boldsymbol{f}(\boldsymbol{x}_\kappa)\| \leq \|\boldsymbol{\mu}(\boldsymbol{x}_\kappa) - \boldsymbol{f}(\boldsymbol{x}_\kappa)\|, \qquad \forall \boldsymbol{x}_\kappa \in \mathbb{X}. \tag{3.9}$$

Proof. Since the true system (3.1) is asymptotically stable (Assumption 3.3), the Lyapunov function $V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\cdot)$ decreases with every step. Thus the next step $\boldsymbol{f}(\boldsymbol{x}_\kappa)$ lies within the set $\mathcal{V}_{\boldsymbol{x}_\kappa} = \left\{\boldsymbol{x} \in \mathbb{X} \middle| V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\boldsymbol{x}) < V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\boldsymbol{x}_\kappa)\right\}$, which is convex due to the convexity of $V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\cdot)$. For all $\boldsymbol{x}_\kappa$ for which $\boldsymbol{\mu}(\boldsymbol{x}_\kappa) \in \mathcal{V}_{\boldsymbol{x}_\kappa}$, holds $\boldsymbol{u}(\boldsymbol{x}_\kappa) = 0$ and thus $\bar{\boldsymbol{f}}(\boldsymbol{x}_\kappa) = \boldsymbol{\mu}(\boldsymbol{x}_\kappa)$, which results in equality in (3.9). For all $\boldsymbol{x}_\kappa$ for which $\boldsymbol{\mu}(\boldsymbol{x}_\kappa) \notin \mathcal{V}_{\boldsymbol{x}_\kappa}$, the stabilized GPSSM $\bar{\boldsymbol{f}}(\boldsymbol{x}_\kappa)$ results in a projection onto the convex set $\mathcal{V}_{\boldsymbol{x}_\kappa}$, thus

$$\bar{\boldsymbol{f}}(\boldsymbol{x}_\kappa) = \min_{\boldsymbol{x}_{\kappa+1} \in \mathcal{V}_{\boldsymbol{x}_\kappa}} \|\boldsymbol{x}_{\kappa+1} - \boldsymbol{\mu}(\boldsymbol{x}_\kappa)\|.$$

The projection $\bar{\boldsymbol{f}}(\boldsymbol{x}_\kappa)$ is closer to any point in the convex set $\mathcal{V}_{\boldsymbol{x}_\kappa}$ than $\boldsymbol{\mu}(\boldsymbol{x}_\kappa)$. Therefore, it is also closer to $\boldsymbol{f}(\boldsymbol{x}_\kappa)$. □

Remark 3.5. *Consider that Proposition 3.1 implies the assumption, that $V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\cdot)$ are Lyapunov functions of the unknown system. These are typically unknown and therefore this imposes a quite strict assumption.*

For an infinite number of training points, it can be shown that the proposed model (3.8) converges to the true system.

Proposition 3.2. *Consider the GP* (2.7)/(2.8) *on a compact set $\bar{\mathbb{X}} \subset \mathbb{R}^n$ with SE kernel* (2.5) *and Lyapunov functions $V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\cdot)$ which fulfill Assumption 3.4 and the condition $V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\boldsymbol{f}(\boldsymbol{x})) - V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\boldsymbol{x}) < 0$, for all $\boldsymbol{x} \in \bar{\mathbb{X}} \setminus \{\hat{\boldsymbol{x}}_{i_*}\}$, $\forall i_*$. Let $\boldsymbol{f}(\cdot)$ be a sample from the GP from which infinitely many training points are generated using a dense distribution on $\bar{\mathbb{X}}$, then the model $\bar{\boldsymbol{f}}(\cdot)$ approaches the true system $\boldsymbol{f}(\cdot)$ almost surely*

$$\mathcal{P}\left\{\lim_{N\to\infty} \sup_{\boldsymbol{x}\in\bar{\mathbb{X}}} \left\|\boldsymbol{f}(\boldsymbol{x}_\kappa) - \bar{\boldsymbol{f}}(\boldsymbol{x}_\kappa)\right\| = 0\right\} = 1$$

for a stabilizing command $\boldsymbol{u}^*(\boldsymbol{x}_\kappa) = \boldsymbol{0}$ $\forall \boldsymbol{x}_\kappa \in \bar{\mathbb{X}}$.

Proof. Under the specified conditions, the maximum difference between the mean function $\boldsymbol{\mu}(\cdot)$ and the true function $\boldsymbol{f}(\cdot)$ becomes arbitrarily small almost surely. This is a well established result from scattered data interpolation [132, Eq. 2.11], where the error is bounded by a power function (which corresponds to the posterior standard deviation of a GP [133, Sec. 5.2]). This converges to zero for $N \to \infty$ [110, Corollary 3.2.]. Since $V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\cdot)$ is a continuous function the condition

$$V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\boldsymbol{\mu}(\boldsymbol{x}_\kappa)) - V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\boldsymbol{x}_\kappa) < 0, \quad \forall \boldsymbol{x}_\kappa \in \bar{\mathbb{X}} \setminus \{\hat{\boldsymbol{x}}_{i_*}\}$$

is fulfilled in the limit $N \to \infty$, and therefore $\boldsymbol{u}^*(\boldsymbol{x}_\kappa) = \boldsymbol{0}$, $\forall \boldsymbol{x}_\kappa \in \bar{\mathbb{X}}$, which yields the provided result. □

This result uses simply the fact that a GP converges to the function from which the data is taken. In this case, the stabilization becomes inactive and therefore does not distort the model.

Regarding the optimization (3.7), the following is concluded.

Proposition 3.3. *The optimization problem* (3.7) *is convex if and only if* $V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\boldsymbol{x})$ *is convex.*

Proof. In the definition of the constraint set (3.7b), $\boldsymbol{\mu}(\boldsymbol{x}_\kappa)$ and $V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\boldsymbol{x}_\kappa)$ are constant with respect to the optimization variable $\boldsymbol{u}$. The addition of $\boldsymbol{u}$ preserves the convexity. Therefore, the constraint set is convex if and only if $V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\cdot)$ is convex, which results in convexity of the optimization problem as defined in [134]. □

This allows to employ efficient solvers for the optimization (3.7) in case of convex Lyapunov functions.

Remark 3.6. *For non-convex Lyapunov functions, finding a global minimum efficiently cannot be expected. Nevertheless, the stability of the model (Theorem 3.1) is not affected, because any feasible* $\boldsymbol{u}$ *results in a stable system. Suboptimal solutions will only potentially result in behavior different from the training data, but convergence remains guaranteed.*

This also facilitates the implementation in real-time critical applications: Since the optimization is initialized at a feasible (i.e. stability guaranteeing) value, the optimization can be interrupted at any time without affecting the convergence (assuming the solver stays in the feasible region).

3.2.2 Probabilistic interpretation

For the probabilistic interpretation, the GPSSM infers a Gaussian distribution over the next state given the current state as it is formalized in (3.5). The resulting system is analyzed using tools from stochastic dynamical systems as reviewed in the following.

Lemma 3.1 (Stability of stochastic systems [135])**.** *Consider a system of the form* (3.5)*. If there exists a positive definite function* $V \colon \mathbb{X} \to \mathbb{R}_{+,0}$ *for which holds*

$$\mathbb{E}[V(\boldsymbol{x}_{\kappa+1})|\boldsymbol{x}_\kappa] - V(\boldsymbol{x}_\kappa) < 0, \quad \forall \boldsymbol{x}_\kappa \in \mathbb{X} \setminus \{\boldsymbol{x}_*\}, \tag{3.10}$$

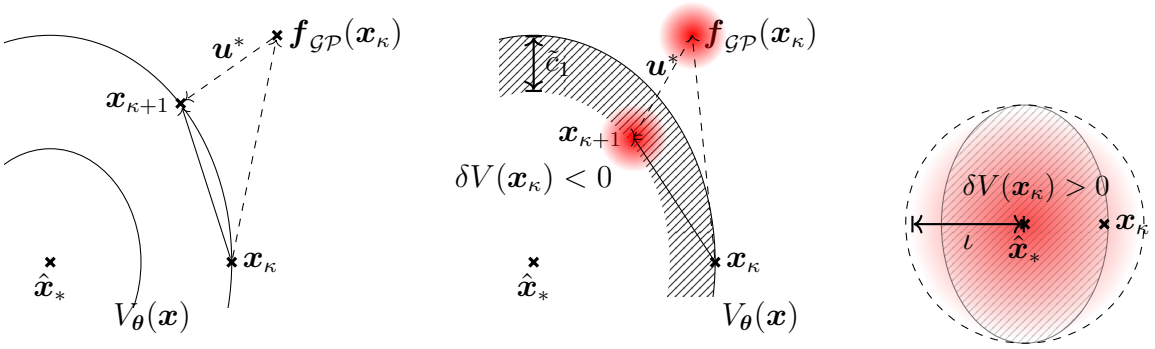

Figure 3.4: The optimizations (3.7) and (3.13) are illustrated for the deterministic and the probabilistic case on the left and in the middle, respectively. The infeasible situation with $n\bar{\zeta}^2 \geq \underline{\ell}^2$ in (3.13c) is visualized on the right, where $\boldsymbol{x}_\kappa \in \mathbb{X}_{i_*}$. The striped area indicates the set $\tilde{\mathbb{X}}_{i_*}$ for which holds $\delta V(\boldsymbol{x}_\kappa) > 0$ for any $\boldsymbol{u}$.

then the equilibrium $\boldsymbol{x}_*$ is a.s. asymptotically stable. If there exists a $V(\cdot)$ for which holds $\alpha_1(\|\boldsymbol{x}\|) \leq V(\boldsymbol{x}) \leq \alpha_2(\|\boldsymbol{x}\|)$, $\forall \boldsymbol{x} \in \mathbb{X}$ (where $\alpha_1, \alpha_2 \colon \mathbb{R}_{+,0} \to \mathbb{R}_{+,0}$ are arbitrary class $\mathcal{K}$ functions) and for which

$$\mathbb{E}[V(\boldsymbol{x}_{\kappa+1})|\boldsymbol{x}_\kappa] - V(\boldsymbol{x}_\kappa) < 0, \quad \forall \boldsymbol{x}_\kappa \in \mathbb{X} \setminus \mathbb{B}_\iota, \tag{3.11}$$

with $\mathbb{B}_\iota = \{\boldsymbol{x} \in \mathbb{X} \,|\, \|\boldsymbol{x} - \boldsymbol{x}_*\| < \iota\}$ holds, then the system is a.s. ultimately bounded (UB) to the set

$$\mathbb{B}_{\tilde{\iota}} = \{\boldsymbol{x} \in \mathbb{X} \,|\, \|\boldsymbol{x} - \boldsymbol{x}_*\| < \tilde{\iota}\}$$

with ultimate bound $\tilde{\iota} = \alpha_1^{-1}(\alpha_2(\iota))$.

Based on this Lyapunov analysis, the following stabilization of a GPSSM is proposed, where

$$\bar{\zeta}^2 = \max_{j=1,\dots,n} \zeta_j^2 \quad \text{and} \quad \underline{\ell}^2 = \min_{\substack{i=1,\dots,N \\ j=1,\dots,n}} \ell_{i,j}^2 \tag{3.12}$$

are the largest signal variance and the smallest lengthscale across all SE kernels, respectively.

Theorem 3.2. *Consider the GP (2.7)/(2.8) with SE kernel (2.5) and the stabilizing command $\boldsymbol{u}^*(\cdot)$ obtained from the optimization,*

$$\boldsymbol{u}^*(\boldsymbol{x}_\kappa) = \arg\min_{\boldsymbol{u}} \frac{1}{2}\boldsymbol{u}^\intercal \boldsymbol{u}, \tag{3.13a}$$

$$\begin{aligned} \text{s.t. } \delta V(\boldsymbol{x}_\kappa) < 0 \quad &\text{if} \quad \boldsymbol{x}_\kappa \neq \hat{\boldsymbol{x}}_{i_*}, \\ \delta V(\boldsymbol{x}_\kappa) = 0 \quad &\text{if} \quad \boldsymbol{x}_\kappa = \hat{\boldsymbol{x}}_{i_*}, \end{aligned} \tag{3.13b}$$

$$\text{and } \boldsymbol{u}^* = \hat{\boldsymbol{x}}_{i_*} - \boldsymbol{\mu}(\boldsymbol{x}_\kappa) \text{ if } \delta V(\boldsymbol{x}_\kappa) > 0 \quad \forall \boldsymbol{u}, \tag{3.13c}$$

where $\delta V(\boldsymbol{x}_\kappa) = \mathbb{E}\left[V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\boldsymbol{f}_{\mathcal{GP}}(\boldsymbol{x}_\kappa) + \boldsymbol{u})|\boldsymbol{x}_\kappa\right] - V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\boldsymbol{x}_\kappa)$, and $V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\cdot)$ is the i_*-th Lyapunov function satisfying Assumption 3.4 with i_* being chosen such that $\boldsymbol{x}_0 \in \mathbb{X}_{i_*}$. Then, the resulting model

$$\boldsymbol{x}_{\kappa+1} = \hat{\boldsymbol{f}}(\boldsymbol{x}_\kappa) := \boldsymbol{f}_{\mathcal{GP}}(\boldsymbol{x}_\kappa) + \boldsymbol{u}^*(\boldsymbol{x}_\kappa) \tag{3.14}$$

converges a.s. to the hyperball

$$\mathbb{B}_{\iota}^{i_*} = \{\boldsymbol{x} \in \mathbb{X} \,|\, \|\boldsymbol{x} - \hat{\boldsymbol{x}}_{i_*}\| < \iota\}$$

with ultimate bound

$$\iota = \sqrt{\underline{\ell}^2 W_0\left(\frac{n\bar{\zeta}^2}{-\underline{\ell}^2}\exp\left(-n\bar{\zeta}^2/\underline{\ell}^2\right)\right) + n\bar{\zeta}^2}, \tag{3.15}$$

where $W_0 : [-\exp(-1); \infty) \to [-1; \infty)$ is the principle branch of the Lambert W function. If $n\bar{\zeta}^2 < \underline{\ell}^2$ holds, it converges a.s. asymptotically to $\hat{\boldsymbol{x}}_{i_*}$, $\forall \boldsymbol{x}_0 \in \mathbb{X}_{i_*}$.

Proof. By Assumption 3.4 $V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\cdot)$ is a valid Lyapunov candidate. The optimization (3.13) is designed to decrease the expected value of the Lyapunov function in every step $\delta V(\boldsymbol{x}_\kappa) < 0$, whenever the optimization is feasible. Let $\tilde{\mathbb{X}}_{i_*} = \{\boldsymbol{x}_\kappa \in \mathbb{X}_{i_*} | \delta V(\boldsymbol{x}_\kappa) > 0 \ \forall \boldsymbol{u}\}$ denote the states for which the constraint set $\delta V(\boldsymbol{x}_\kappa) < 0$ is empty, thus no $\boldsymbol{u}$ exists which fulfills the Lyapunov condition. For all states outside of the infeasible region, $\boldsymbol{x} \in \mathbb{X} \backslash \tilde{\mathbb{X}}_{i_*}$, the conditions in Lemma 3.1 hold and the system converges to $\tilde{\mathbb{X}}_{i_*}$. The case $\boldsymbol{x}_\kappa \in \tilde{\mathbb{X}}_{i_*}$ is covered in (3.13c) and leads to the stochastic dynamics

$$\boldsymbol{x}_{\kappa+1} \sim \mathcal{N}(\hat{\boldsymbol{x}}_{i_*}, \boldsymbol{\Sigma}(\boldsymbol{x}_\kappa)). \tag{3.16}$$

Without loss of generality and for notational convenience, we set $\hat{\boldsymbol{x}}_{i_*} = \boldsymbol{0}$. For the analysis of (3.16), the quadratic Lyapunov function $V_{\mathrm{sq}}(\boldsymbol{x}) = \boldsymbol{x}^\intercal \boldsymbol{x}$ is employed.[2] The control Lyapunov function $V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\cdot)$ ensures convergence to the set $\tilde{\mathbb{X}}_{i_*}$, but inside it is no longer applied, since $\boldsymbol{u}^*$ is chosen independently of $V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\cdot)$ in (3.13c). Consider $\delta V_{\mathrm{sq}}(\cdot)$ as follows

$$\begin{aligned}\delta V_{\mathrm{sq}}(\boldsymbol{x}_\kappa) &= \mathbb{E}[V_{\mathrm{sq}}(\boldsymbol{x}_{\kappa+1})|\boldsymbol{x}_\kappa] - V_{\mathrm{sq}}(\boldsymbol{x}_\kappa) = \mathbb{E}[\boldsymbol{x}_{\kappa+1}^\intercal \boldsymbol{x}_{\kappa+1}] - \boldsymbol{x}_\kappa^\intercal \boldsymbol{x}_\kappa = \operatorname{trace}(\boldsymbol{\Sigma}(\boldsymbol{x}_\kappa)) - \boldsymbol{x}_\kappa^\intercal \boldsymbol{x}_\kappa \\ &= \sum_{j=1}^{n} \sigma_j^2(\boldsymbol{x}_\kappa) - \boldsymbol{x}_\kappa^\intercal \boldsymbol{x}_\kappa \leq \underbrace{n\bar{\zeta}^2\left(1 - \exp\left(\frac{-\|\boldsymbol{x}_\kappa\|^2}{\underline{\ell}^2}\right)\right)}_{:=\underline{V}(\|\boldsymbol{x}_\kappa\|)} - \underbrace{\|\boldsymbol{x}_\kappa\|^2}_{:=\bar{V}(\|\boldsymbol{x}_\kappa\|)},\end{aligned}$$

where $\mathbb{E}[\boldsymbol{x}_{\kappa+1}] = \boldsymbol{0}$ from (3.16) and the variance upper bound from Lemma 2.1 are employed. It can be seen that $\delta V_{\mathrm{sq}}(\boldsymbol{0}) = 0$ and

$$\lim_{\|\boldsymbol{x}_\kappa\| \to \infty} \delta V_{\mathrm{sq}}(\boldsymbol{x}_\kappa) = -\infty$$

since $\underline{V}(\cdot)$ is bounded. Thus, if $\delta V_{\mathrm{sq}}(\cdot)$ has no further roots ($\delta V_{\mathrm{sq}}(\boldsymbol{x}_\kappa)$ is negative definite), the system is asymptotically stable. In case there are roots outside the origin, the outer most is the ultimate bound.

Comparing the derivatives of $\bar{V}(\cdot)$ and $\underline{V}(\cdot)$ yields

$$\frac{\partial \underline{V}(\|\boldsymbol{x}\|)}{\partial \|\boldsymbol{x}\|} = \frac{2n\bar{\zeta}^2}{\underline{\ell}^2}\exp\left(\frac{-\|\boldsymbol{x}\|^2}{\underline{\ell}^2}\right)\|\boldsymbol{x}\|, \qquad \frac{\partial \bar{V}(\|\boldsymbol{x}\|)}{\partial \|\boldsymbol{x}\|} = 2\|\boldsymbol{x}\|.$$

Since $\exp\left(-\|\boldsymbol{x}\|^2/\underline{\ell}^2\right) \leq 1$, it can be seen that $\bar{V}(\cdot)$ increases faster than $\underline{V}(\cdot)$ $\forall \boldsymbol{x} \in \mathbb{X}$ if $n\bar{\zeta}^2/\underline{\ell}^2 < 1$. It is concluded that $\delta V_{\mathrm{sq}}(\boldsymbol{x}) < 0$, $\forall \boldsymbol{x} \in \mathbb{X} \setminus \{\boldsymbol{0}\}$ for $n\bar{\zeta}^2 < \underline{\ell}^2$, which proves that the equilibrium is a.s. asymptotically stable.

[2] For the special case $V(\boldsymbol{x}) = \boldsymbol{x}^\intercal \boldsymbol{x}$, Lemma 3.1 simplifies to $\tilde{\iota} = \iota$.

If $n\bar{\zeta}^2 > \underline{\ell}^2$, consider that $\frac{\partial \underline{V}(\|\boldsymbol{x}\|)}{\partial \|\boldsymbol{x}\|} - \frac{\partial \bar{V}(\|\boldsymbol{x}\|)}{\partial \|\boldsymbol{x}\|}$ changes its sign at most once. Thus, only the proposed ultimate bound (3.15) must be confirmed as the single root of $\delta V_{\mathrm{sq}}(\cdot)$. We define $z = \frac{n\bar{\zeta}^2}{-\underline{\ell}^2} \exp\left(-n\bar{\zeta}^2/\underline{\ell}^2\right)$ and make use of the principle branch $W_0(\cdot)$ of the Lambert W function. Substituting $\|\boldsymbol{x}\| = \iota$ from (3.15) yields

$$\begin{aligned}
\underline{V}(\iota) - \bar{V}(\iota) &= n\bar{\zeta}^2 \left(1 - \exp\left(-W_0(z) - n\bar{\zeta}^2/\underline{\ell}^2\right)\right) - \underline{\ell}^2 W_0(z) - n\bar{\zeta}^2 \\
&= -n\bar{\zeta}^2 \exp\left(-W_0(z)\right) \exp\left(-n\bar{\zeta}^2/\underline{\ell}^2\right) - \underline{\ell}^2 W_0(z) \\
&= \left(\frac{n\bar{\zeta}^2}{-\underline{\ell}^2} \exp\left(-n\bar{\zeta}^2/\underline{\ell}^2\right) - W_0(z) \exp\left(W_0(z)\right)\right) \underline{\ell}^2 \exp\left(-W_0(z)\right) \\
&= (z - z) \exp\left(-W_0(z)\right) = 0,
\end{aligned}$$

where $W_0(z) \exp\left(W_0(z)\right) = z$ is applied, which is a known identity of the Lambert W function. This allows to conclude

$$\delta V_{\mathrm{sq}}(\boldsymbol{x}) < 0, \quad \forall \boldsymbol{x} \in \{\boldsymbol{x} \in \mathbb{X} \,|\, \|\boldsymbol{x}\| > \iota\}$$

and shows a.s. UB with the specified bound ι according to Lemma 3.1. □

Remark 3.7. *The infeasible case* (3.13c) *typically occurs in close proximity to the equilibrium point, where uncertainty is too large with respect to the small "step size". The stabilization command cannot be chosen such that the expected decrease of the Lyapunov function $\delta V(\cdot)$ dominates the variance term trace$(\boldsymbol{\Sigma}(\cdot))$.*

Figure 3.4 (center) visualizes the optimization-based stabilization (3.13)*. Figure 3.4 (right) illustrates the infeasible case* (3.16)*.*

Corollary 3.2. *Let $N_* = 1$, $V_{\boldsymbol{\theta}}(\cdot)$ is radially unbounded (Assumption 3.5), and $\mathbb{X} = \mathbb{R}^n$. Furthermore, consider a GP* (2.7)/(2.8) *with SE kernel* (2.5) *and the stabilizing command $\boldsymbol{u}^*(\cdot)$ obtained from the optimization in* (3.13)*. Then, the system* (3.14) *is a.s. globally UB to the set $\mathbb{B}_\iota$. If $n\bar{\zeta}^2 < \underline{\ell}^2$, the equilibrium $\hat{\boldsymbol{x}}_*$ of the model* (3.14) *is a.s. GAS.*

Proof. Since the Lyapunov function is radially unbounded and decreasing over time (compare Theorem 3.2) the necessary conditions for *global* stability hold [135]. □

Remark 3.8. *For a polynomial Lyapunov candidate, the moments of $V_{\boldsymbol{\theta}}(\boldsymbol{f}_{\mathcal{GP}}(\boldsymbol{x}_\kappa) + \boldsymbol{u})$ are polynomial in the mean and variance of $\boldsymbol{f}_{\mathcal{GP}}(\boldsymbol{x}_\kappa)$. For the general computation of the expected value $\mathbb{E}[V_{\boldsymbol{\theta}}(\boldsymbol{f}_{\mathcal{GP}}(\boldsymbol{x}_\kappa) + \boldsymbol{u})]$ in* (3.13b)*, we refer to [136].*

Consider the following example for the results in Theorem 3.1.

Example 3.2. *For two dimensions ($n = 2$), the system* (3.16) *is simulated with $\underline{\ell}^2 = 1$ for two different cases:*

i) Choose $\bar{\zeta}^2 = 1$, thus $n\bar{\zeta}^2 > \underline{\ell}^2$ resulting in a.s. UB with bound $\iota \approx 1.26$.

ii) Choose $\bar{\zeta}^2 = 1/3$, thus $n\bar{\zeta}^2 \leq \underline{\ell}^2$ resulting in a.s. asymptotic stability.

Figure 3.5 visualizes the Lyapunov argumentation in the proof of Theorem 3.2, showing $\bar{V}(\cdot)$ and $\underline{V}(\cdot)$.

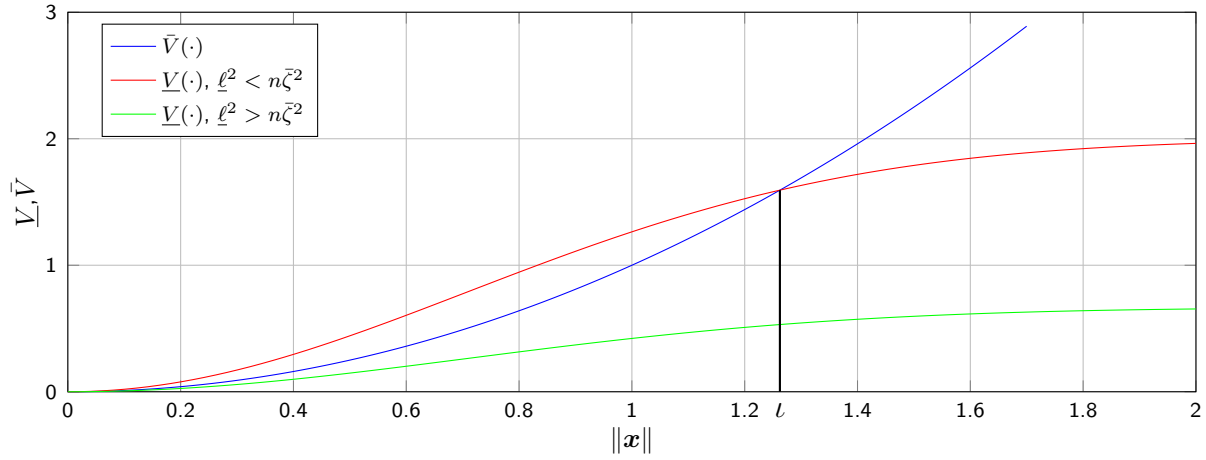

Figure 3.5: Visualization for the proof of Theorem 3.2: For $\underline{\ell}^2 > n\bar{\zeta}^2$ the function $\underline{V}(\cdot)$ grows too slow for an intersection with $\bar{V}(\cdot)$ and therefore results in a.s. asymptotic stability. For $\underline{\ell}^2 < n\bar{\zeta}^2$ $\underline{V}(\cdot)$ intersects $\bar{V}(\cdot)$ at ι, resulting in a.s. UB.

For both cases, the simulations are initialized at the same $N_{init} = 10^3$ randomly sampled (uniformly from the interval $[-5;5]^2$) state $\boldsymbol{x}_0$. In each step, the next state is drawn from an unbiased normal distribution with state dependent variance

$$\boldsymbol{x}_{\kappa+1} \sim \mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}\bar{\sigma}^2(\boldsymbol{x}_\kappa) & 0\\ 0 & \bar{\sigma}^2(\boldsymbol{x}_\kappa)\end{bmatrix}\right),$$

where $\bar{\sigma}^2(\cdot)$ serves as an upper limit for any posterior variance of a GP with an arbitrary training data set according to Lemma 2.1. The subscript of $\bar{\sigma}^2_{\boldsymbol{x}^{(i)}}$ is dropped for notational convenience.

The simulations in Fig. 3.6 quickly decrease for the asymptotically stable case. In contrast, for the ultimately bounded case, the states drop below the ultimate bound, but do not decrease further. The bound is exceeded irregularly because it only holds with probability 1 for $\kappa \to \infty$. To visualize this, Fig. 3.7 shows the number of trajectories violating the bound over a longer time horizon.

Considering the interpretations of GPs in Sec. 2.3, the result in Theorem 3.2 is based on the *stochastic interpretation*. However, it directly generalizes to the sequential sampling for the *scenario interpretation*, proposed in [106], as shown in the following.

Corollary 3.3. *Consider the SE kernel* (2.5)*, the stabilizing command $\boldsymbol{u}^*(\cdot)$ obtained from* (3.13) *and the model*

$$\boldsymbol{x}_{\kappa+1} = \boldsymbol{f}^\kappa_{\mathcal{GP}}(\boldsymbol{x}_\kappa) + \boldsymbol{u}^*(\boldsymbol{x}_\kappa) \quad \textit{with} \quad \boldsymbol{f}^\kappa_{\mathcal{GP}}(\boldsymbol{x}_\kappa) \sim \mathcal{N}\left(\boldsymbol{\mu}_\kappa(\boldsymbol{x}_\kappa), \boldsymbol{\sigma}^2_\kappa(\boldsymbol{x}_\kappa)\right), \tag{3.17}$$

where

$$\boldsymbol{\mu}_\kappa(\boldsymbol{x}_\kappa) = \mathbb{E}[\boldsymbol{f}_{\mathcal{GP}}(\boldsymbol{x}_\kappa)|\mathbb{D}, \mathbb{D}_\kappa], \qquad \boldsymbol{\sigma}^2_\kappa(\boldsymbol{x}_\kappa) = \mathbb{V}[\boldsymbol{f}_{\mathcal{GP}}(\boldsymbol{x}_\kappa)|\mathbb{D}, \mathbb{D}_\kappa]$$

are the posterior mean and variance functions of a GP in (2.7)/(2.8)*, which - in addition to the data points $\mathbb{D}$ - are, for $\kappa \geq 1$, conditioned on previously visited states*

$$\mathbb{D}_\kappa = \{(\boldsymbol{x}_{i-1}, \boldsymbol{x}_i)\}_{i=1}^\kappa .$$

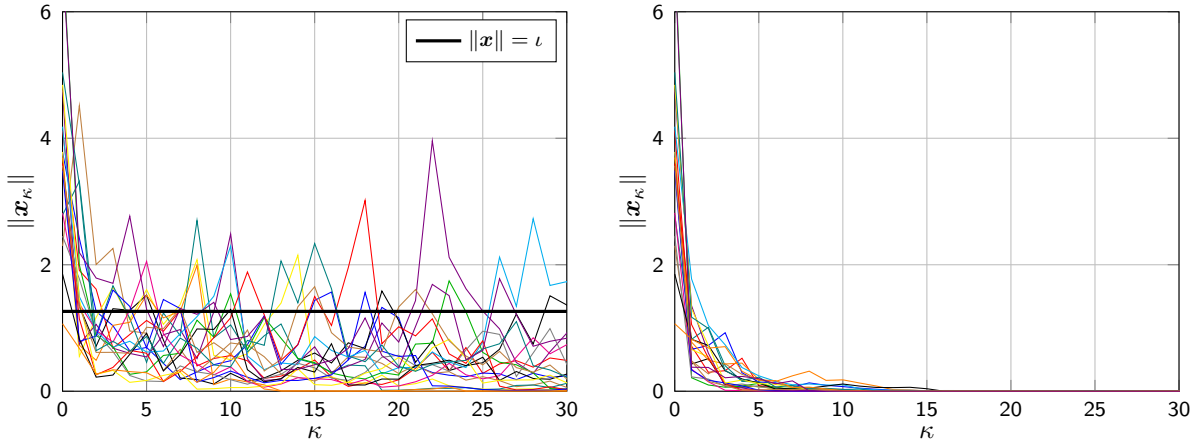

Figure 3.6: The norm of 20 out of 10^3 trajectories for different initial conditions from Example 3.2 are illustrated. The ultimate bounded case with $n\bar{\zeta}^2 > \underline{\ell}^2$ (left) is compared with the asymptotically stable case with $n\bar{\zeta}^2 \leq \underline{\ell}^2$ (right).

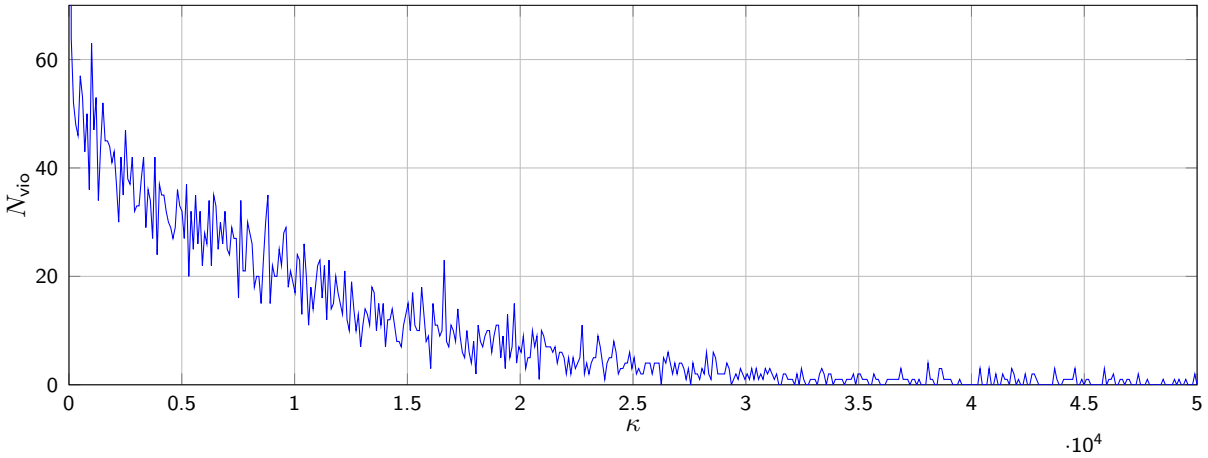

Figure 3.7: Violations of the ultimate bound ι for the simulation in Example 3.2. The number of the $N_{\text{init}} = 10^3$ trajectories which are outside the hyperball $\mathbb{B}_\iota$ ($\|\boldsymbol{x}_\kappa\| > \iota$) for each time step κ are denoted by N_{vio}. The plot suggests that the violations of the ultimate bound vanish over time.

Then, the convergence properties defined in Theorem 3.2 for (3.14) *also hold for* (3.17).

Proof. The system (3.17) corresponds to a sequential sampling of the function over which the GP describes a distribution as introduced in [106]. The convergence behavior cannot directly be analyzed with the previously employed Lyapunov-based methods, because the system is, due to the conditioning on all previously visited states, non-Markovian. However, the difference in the sequentially conditioned mean function $\boldsymbol{\mu}_\kappa(\cdot)$ and the mean function of the stochastic interpretation $\boldsymbol{\mu}(\cdot)$ will be accounted for by the stabilizing command $\boldsymbol{u}^*(\cdot)$. Furthermore, based on the results in [110] it can be concluded, that $\boldsymbol{\sigma}_\kappa^2(\cdot) \leq \boldsymbol{\sigma}^2(\cdot)$ (elementwise, $\forall \boldsymbol{x} \in \mathbb{X}$). This ensures that the analysis based on (3.16) serves as a conservative estimate and the convergence bounds for (3.14) also hold for (3.17). □

According to Theorem 3.2, Objective 3.2 is only achieved if $n\bar{\zeta}^2 < \underline{\ell}^2$ holds. This constraint could be imposed in the likelihood optimization (2.6). However, a limitation in the search space of the hyperparameters leads to a suboptimal model choice if the likelihood attains its maximum outside of the constraint set. Therefore, we want to circumvent this constraint and instead propose to add additional data to the training set to achieve asymptotic stability for arbitrary hyperparameters. Thus, the following section investigates how many additional measurements from the system are required to satisfy Objective 3.2.

3.2.3 Convergence with additional training data

This section considers the case $n\bar{\zeta}^2 > \underline{\ell}^2$, which results according to Theorem 3.2 not necessarily in asymptotic stability, and does not fulfill Objective 3.2. We therefore make an assumption on the availability of additional training data $\mathbb{D}_{\text{add}}$ and elaborate how many additional points $N_{\text{add}} = |\mathbb{D}_{\text{add}}|$ are required to fulfill Objective 3.2.

Assumption 3.6. *An additional dataset*

$$\mathbb{D}_{add} = \left\{ \left(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)} \right) \right\}_{i=N+1}^{N+N_{add}},$$

can be chosen, where $\boldsymbol{y}^{(i)} = \boldsymbol{f}\left(\boldsymbol{x}^{(i)}\right)$ *is available for arbitrary input locations* $\boldsymbol{x}^{(i)} \in \mathbb{B}_\iota$*. The hyperparameters* $\boldsymbol{\psi}_j$*, for all* $j = 1, \ldots, n$ *of the GPSSM remain unchanged.*

The assumption implies that $N \gg N_{\text{add}}$ since otherwise the optimal hyperparameters would significantly be affected by the updated data set. Taking data points at arbitrary locations is a rather strict assumption, but inevitable here. We leave it to future work to relax this assumption.

Without loss of generality, only a single equilibrium at the origin $\boldsymbol{x}_* = \boldsymbol{0}$ with domain of attraction $\mathbb{R}^n$ is considered from now on. The number of required additional datapoints is derived as follows, where $W_{-1}\colon [-\exp(-1); 0) \to [-\infty; -1)$ is the lower branch of the Lambert W function.

Theorem 3.3. *Consider a GP* (2.7)/(2.8) *with SE kernel* (2.5)*. Then, there exists a set* $\mathbb{D}_{add}$ *of size*

$$N_{add} = (1 + 2\iota/r_{tr})^n + 1,$$

where $r_{tr} \coloneqq -\underline{\ell}^2 \log\left(1 - \tilde{r}_a^2/n\bar{\zeta}^2\right)$ *and*

$$\tilde{r}_a \coloneqq \frac{l}{3\sqrt{n\bar{\zeta}^2}} \sqrt{-n\bar{\zeta}^2 W_{-1}\left(\exp\left(\frac{\underline{\ell}^2}{n\bar{\zeta}^2} - 1\right)\left(\frac{\underline{\ell}^2}{n\bar{\zeta}^2} - 1\right)\right) + \underline{\ell}^2 - n\bar{\zeta}^2}$$

such that for any (fixed) hyperparameter set $\boldsymbol{\psi}_j$, $j = 1, \ldots, n$ *the system* (3.14) *is a.s. asymptotically stable.*

Proof. Similar to the proof of Theorem 3.2, consider

$$\delta V_{\mathrm{sq}}(\boldsymbol{x}_\kappa) = \sum_{j=1}^{n} \sigma_j^2(\boldsymbol{x}_\kappa) - \boldsymbol{x}_\kappa^\intercal \boldsymbol{x}_\kappa \leq n \max_j \sigma_j^2(\boldsymbol{x}_\kappa) - \boldsymbol{x}_\kappa^\intercal \boldsymbol{x}_\kappa,$$

which is negative for $\|\boldsymbol{x}_\kappa\| > \iota$. In a first step, we analyze the area around the origin and how to ensure $\delta V_{\mathrm{sq}}(\boldsymbol{x}_\kappa) < 0$ using a single additional data point.

Since the SE kernel is stationary, we can introduce the notation $\mathfrak{k}(\boldsymbol{x}, \boldsymbol{x}') = \mathfrak{k}(\boldsymbol{x} - \boldsymbol{x}')$. Furthermore, we denote the posterior variance function of a GP with hyperparameters $\bar{\zeta}^2, \underline{\ell}^2$ and the dataset $\mathbb{D}_2 = \left\{(\mathbf{0}, \mathbf{0}), \left(\boldsymbol{x}^{(a)}, \boldsymbol{y}^{(a)}\right)\right\}$ by $\sigma_{\mathbb{D}_2}^2(\cdot)$. It serves as an upper bound for the variance function $\max_j \sigma_j^2(\cdot)$ with dataset $\mathbb{D}_2 \cup \mathbb{D}$, for the same reasons as in Lemma 2.1.

First, we analyze the single dimensional case $n = 1$ to obtain a location for a single additional data point $x^{(a)} = r_a$ which ensures $n\sigma_{\mathbb{D}_2}^2(x) < x^2$ for $x \in [0; x^{(a)}]$. According to [137], $\sigma_{\mathbb{D}_2}^2(\cdot)$ is given by

$$\sigma_{\mathbb{D}_2}^2(x) = \bar{\zeta}^2 - \frac{\bar{\zeta}^2\left(\mathfrak{k}(x)^2 + \mathfrak{k}(r_a - x)^2\right) - 2\mathfrak{k}(r_a)\mathfrak{k}(x)\mathfrak{k}(r_a - x)}{(\bar{\zeta}^2)^2 - \mathfrak{k}(r_a)^2},$$

where $\mathfrak{k}(0) = \bar{\zeta}^2$ is used. Multiplying $(\bar{\zeta}^2)^2 - \mathfrak{k}(r_a)^2$ on both sides of $\sigma_{\mathbb{D}_2}^2(x) < x^2$ yields the functions

$$\begin{aligned}
\bar{Z}(r_a, x) &\coloneqq (\bar{\zeta}^2)^3 - \bar{\zeta}^2\mathfrak{k}(r_a)^2 - \bar{\zeta}^2\mathfrak{k}(x)^2 - \bar{\zeta}^2\mathfrak{k}(r_a - x)^2 - 2\mathfrak{k}(r_a)\mathfrak{k}(x)\mathfrak{k}(r_a - x), \\
\underline{Z}(r_a, x) &\coloneqq ((\bar{\zeta}^2)^2 - \mathfrak{k}(r_a)^2)x^2,
\end{aligned}$$

for the left and right side, respectively. Their derivatives with respect to x are

$$\begin{aligned}
\frac{\partial \bar{Z}(r_a, x)}{\partial x} &= \frac{2}{\underline{\ell}^2}\left(\bar{\zeta}^2 x\mathfrak{k}(x)^2 - \bar{\zeta}^2(r_a - x)\mathfrak{k}(r_a - x)^2 + \mathfrak{k}(r_a)\mathfrak{k}(x)\mathfrak{k}(r_a - x)(2x - r_a)\right) \\
&= \frac{2}{\underline{\ell}^2}\Big(r_a\left(\mathfrak{k}(r_a)\mathfrak{k}(x)\mathfrak{k}(r_a - x) - \bar{\zeta}^2\mathfrak{k}(r_a - x)^2\right) \\
&\quad + x\left(\bar{\zeta}^2\mathfrak{k}(x)^2 + \bar{\zeta}^2\mathfrak{k}(r_a - x)^2 - 2\mathfrak{k}(r_a)\mathfrak{k}(x)\mathfrak{k}(r_a - x)\right)\Big), \\
\frac{\partial \underline{Z}(r_a, x)}{\partial x} &= 2x\left((\bar{\zeta}^2)^2 - \mathfrak{k}(r_a)^2\right).
\end{aligned}$$

Both derivatives are zero for $x = 0$. Thus, we consider the second derivatives evaluated at the origin

$$\begin{aligned}
\bar{Z}'(r_a) &\coloneqq \frac{\partial^2 \bar{Z}(r_a, 0)}{\partial x^2} = \frac{2}{\underline{\ell}^2}\left((\bar{\zeta}^2)^3 - \bar{\zeta}^2\mathfrak{k}(r_a)^2\left(1 + \frac{r_a^2}{\underline{\ell}^2}\right)\right), \\
\underline{Z}'(r_a) &\coloneqq \frac{\partial^2 \underline{Z}(r_a, 0)}{\partial x^2} = 2((\bar{\zeta}^2)^2 - \mathfrak{k}(r_a)^2).
\end{aligned}$$

With the lower branch of the Lambert W function $W_{-1}(\cdot)$, it can be shown using [138] that

$$n\bar{Z}'(r_a) - \underline{Z}'(r_a) = \frac{2}{\underline{\ell}^2}\left(n(\bar{\zeta}^2)^3 - (\bar{\zeta}^2)^2\underline{\ell}^2 - \mathfrak{k}(r_a)^2\left(n\bar{\zeta}^2\left(1+\frac{r_a^2}{\underline{\ell}^2}\right) - \underline{\ell}^2\right)\right)$$

has only one positive zero at

$$r_a = \frac{l}{\sqrt{n\bar{\zeta}^2}}\sqrt{-n\bar{\zeta}^2 W_{-1}\left(\exp\left(\frac{\underline{\ell}^2}{n\bar{\zeta}^2}-1\right)\left(\frac{\underline{\ell}^2}{n\bar{\zeta}^2}-1\right)\right) + \underline{\ell}^2 - n\bar{\zeta}^2}.$$

Thus, for any $x^{(a)} \in [0; r_a]$, it is concluded that $n\bar{Z}(r_a, x) - \underline{Z}(r_a, x) < 0$ for all $x \in \left(0; x^{(a)}\right]$. Therefore, $\delta V_{\mathrm{sq}}(x) < 0$ holds $\forall x \in (0; x^{(a)}]$ if $x^{(a)} < r$.

In the multiple dimensional case, it can be concluded from $\left\|\boldsymbol{x}^{(a)}\right\| < r_a$ that $\delta V_{\mathrm{sq}}(\boldsymbol{x}) < 0$ for all $\boldsymbol{x} = \mathfrak{a}\boldsymbol{x}^{(a)}$ with $0 < \mathfrak{a} \leq 1$. It therefore only holds on a line as visualized in Fig. 3.8 on the left.

To extend the result to a hyperball around the origin, we analyze the variance $\sigma^2_{\mathbb{D}_2}(\boldsymbol{x})$ as $\boldsymbol{x}$ is moved on a hypersphere centered at the origin. Defining the constants $c_1 = \mathfrak{k}(\|\boldsymbol{x}\|)$ and $c_2 = \mathfrak{k}\left(\left\|\boldsymbol{x}^{(\tilde{a})}\right\|\right)$ and the variable $\tau = \left\|\boldsymbol{x} - \boldsymbol{x}^{(\tilde{a})}\right\|$, the derivative of $\sigma^2_{\mathbb{D}_2}(\tau)$ with respect to τ is given by

$$\frac{\partial\sigma^2_{\mathbb{D}_2}(\tau)}{\partial\tau} = \frac{\partial\mathfrak{k}(\tau)}{\partial\tau}\frac{-\bar{\zeta}^2}{(\bar{\zeta}^2)^2 - c_2^2}(2\mathfrak{k}(\tau) - 2c_1c_2),$$

which is non-negative because $\frac{\partial\mathfrak{k}(\tau)}{\partial\tau} \leq 0$ for the SE kernel, $\bar{\zeta}^2 \geq c_2$ and $\mathfrak{k}(\tau) \geq c_1c_2$ since

$$\left\|\boldsymbol{x} - \boldsymbol{x}^{(\tilde{a})}\right\|^2 \leq \|\boldsymbol{x}\|^2 + \left\|\boldsymbol{x}^{(\tilde{a})}\right\|^2 \quad \Rightarrow \quad \exp\left(\frac{-\tau^2}{2\underline{\ell}^2}\right) \geq \exp\left(\frac{-\|\boldsymbol{x}\|^2}{2\underline{\ell}^2}\right)\exp\left(\frac{-\left\|\boldsymbol{x}^{(\tilde{a})}\right\|^2}{2\underline{\ell}^2}\right)$$

according to the triangle inequality. We conclude that the point with the highest variance on the hypersphere lies on the opposite side of the point $\boldsymbol{x}^{(\tilde{a})}$

$$\sigma^2_{\mathbb{D}_2}\left(-\boldsymbol{x}^{(\tilde{a})}\right) = \max_{\|\boldsymbol{x}\|=\left\|\boldsymbol{x}^{(\tilde{a})}\right\|} \sigma^2_{\mathbb{D}_2}(\boldsymbol{x}),$$

as illustrated in Fig. 3.8 (center).

As a result, $\boldsymbol{x}^{(\tilde{a})}$ must be chosen close enough to the origin, such that there exists a $\tilde{r}_a$ for which all points $\|\boldsymbol{x}\| < \tilde{r}_a$ are closer to $\boldsymbol{x}^{(\tilde{a})}$, than to $\boldsymbol{x}^{(a)}$,

$$\begin{aligned}
&\left\|\boldsymbol{x}^{(\tilde{a})} + \boldsymbol{x}\right\| \leq \left\|\boldsymbol{x}^{(\tilde{a})}\right\| + \|\boldsymbol{x}\| \leq \left\|\boldsymbol{x}^{(a)} - \boldsymbol{x}\right\| = \left\|\boldsymbol{x}^{(a)}\right\| - \|\boldsymbol{x}\| \\
&\left\|\boldsymbol{x}^{(\tilde{a})}\right\| + 2\|\boldsymbol{x}\| \leq \left\|\boldsymbol{x}^{(\tilde{a})}\right\| \quad \Rightarrow \quad \left\|\boldsymbol{x}^{(\tilde{a})}\right\| \leq \frac{1}{3}\left\|\boldsymbol{x}^{(a)}\right\|,
\end{aligned}$$

which yields $\tilde{r}_a \coloneqq r_a/3$.

So for the multidimensional case, it holds for all $\boldsymbol{x}$ with $0 < \|\boldsymbol{x}\| < \tilde{r}_a$ that $n\sigma^2_{\mathbb{D}_2}(\boldsymbol{x}) < \|\boldsymbol{x}\|^2$ if $\left\|\boldsymbol{x}^{(\tilde{a})}\right\| \leq \tilde{r}_a$. Thus we set $\boldsymbol{x}^{(N+1)} = \boldsymbol{x}^{(\tilde{a})}$ according to Assumption 3.6.

If $\tilde{r}_a > \iota$, the proof is completed since then $n\sigma^2_{\mathbb{D}_2}(\boldsymbol{x}) < \|\boldsymbol{x}\|^2$ holds for all $\boldsymbol{x} \in \mathbb{X}$ with one additional training point.

For, $\tilde{r}_a \leq \iota$, the region $\tilde{r}_a \leq \|\boldsymbol{x}\| \leq \iota$, must be covered with enough training data to ensure $n\sigma^2_{\mathbb{D}_2}(\boldsymbol{x}) < \|\boldsymbol{x}\|^2$ everywhere. For simplicity, we show that $n\sigma^2_{\mathbb{D}_2}(\boldsymbol{x}) < \tilde{r}_a^2$ is ensured for all $\|\boldsymbol{x}\| < \iota$. From the bound in Lemma 2.1, it follows that every additional data point $\boldsymbol{x}^{(i)}$, $i \in \{N+2, \ldots, N+N_{\text{add}}\}$ upper bounds the variance function in a radius r_{tr} given by

$$\left\|\boldsymbol{x} - \boldsymbol{x}^{(i)}\right\|^2 < r_{\text{tr}} \quad \Rightarrow \quad \sigma_j^2(\boldsymbol{x}) \leq n\bar{\zeta}^2 \left(1 - \exp\left(\frac{-\|\boldsymbol{x} - \boldsymbol{x}^{(i)}\|^2}{\underline{\ell}^2}\right)\right) \leq \tilde{r}_a^2.$$

The number of hyperballs with radius r_{tr} required to cover a (larger) hyperball with radius ι is given by the covering number. In [139] the upper bound on the covering number is given by $(1+2\iota/r_{\text{tr}})^n$. Therefore, not more than $(1+2\iota/r_{\text{tr}})^n+1$ additional data points are required to ensure $\delta V_{\text{sq}}(\boldsymbol{x}_\kappa) < 0 \; \forall \boldsymbol{x} \setminus \{\boldsymbol{0}\}$, which concludes the proof. □

Example 3.3. *Reconsider Example 3.2 with $n = 2$, $\underline{\ell}^2 = 1$ and $\bar{\zeta}^2 = 1$, for which $\iota \approx 1.26$ was obtained. According to Theorem 3.3, one training point $x^{(\tilde{a})}$ must be chosen near the origin with $\tilde{r}_a \approx 0.37$ ($r_a \approx 1.12$). Figure 3.9 compares different choices of the first additional data point $\boldsymbol{x}^{(\tilde{a})}$. With $r_{tr} \approx 0.0724$ a maximum of $N_{add} = 1290$ additional training points is required to ensure a.s. asymptotic stability.*

3.3 Learning Lyapunov functions for stabilization

For the stabilization of a GPSSM as presented in Sec. 3.2, any control Lyapunov function (subject to Assumptions 3.4 and 3.5) is applicable. However, the GPSSM $\boldsymbol{f}_{\mathcal{GP}}(\cdot)$ without stabilization is the best fit to the data and therefore should only be minimally distorted by the stabilization $\boldsymbol{u}(\cdot)$. This requires an optimal choice of the Lyapunov functions $V^{i_*}_{\boldsymbol{\theta}_{i_*}}(\cdot)$ based on the available data to properly describe the convergence behavior of the model. This section presents a general framework for a data-driven optimization-based estimation of the convergence behavior including the equilibria and discusses three specific choices for the Lyapunov candidates.

3.3.1 Optimization-based formulation

Consider Lyapunov functions $V^{i_*}_{\boldsymbol{\theta}_{i_*}}(\cdot)$ which are parameterized by $\boldsymbol{\theta}_{i_*} \in \boldsymbol{\Theta}_{i_*}$ and the estimated equilibrium $\hat{\boldsymbol{x}}_{i_*} \in \mathbb{X}_{i_*}$. For fitting these parameters to the training set $\mathbb{D}$, the following is concluded.

Proposition 3.4. *Consider N_* Lyapunov candidates $V^{i_*}_{\boldsymbol{\theta}_{i_*}}(\cdot)$ under Assumptions 3.4 and 3.5 and the dataset from Assumption 3.2. Furthermore, consider $\mathcal{I}_{i_*} = \left\{i = 1, \ldots, N \,\middle|\, \boldsymbol{x}^{(i)} \in \mathbb{X}_{i_*}\right\}$ and $\mathfrak{g} : \mathbb{R} \to \mathbb{R}_+$, which is any function of the form*

$$\mathfrak{g}(\xi) = \begin{cases} 0 & \text{for } \xi \leq 0 \\ \tilde{\mathfrak{g}}(\xi) & \text{for } \xi > 0 \end{cases} \quad \text{with} \quad \tilde{\mathfrak{g}}(\xi) > 0 \; \forall \xi. \tag{3.18}$$

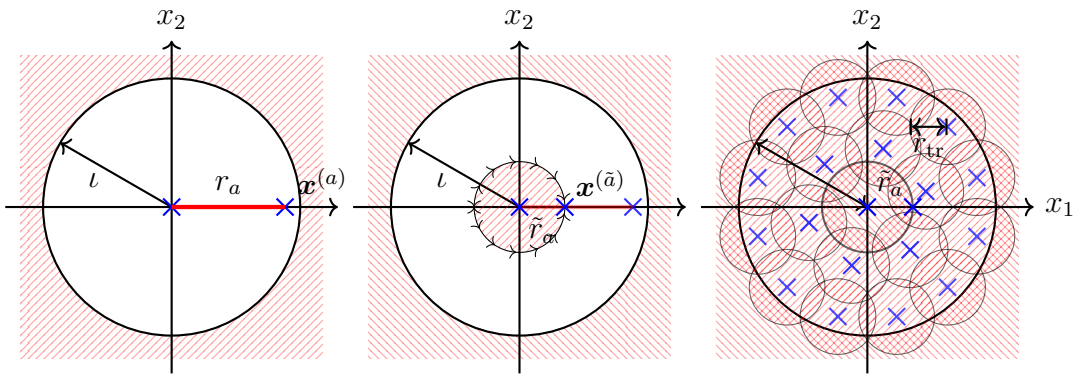

Figure 3.8: Illustration of the proof of Theorem 3.3. The red (stripped) areas indicate sets where $\delta V_{\text{sq}}(\boldsymbol{x}) < 0$ holds. Blue crosses show additional training data in $\mathbb{D}_{\text{add}}$. The one dimensional case is visualized on the left. The center shows the generalization from one dimension to multiple dimensions with $\|\boldsymbol{x}^{(\tilde{a})}\| = \tilde{r}_a < r_a/3$. The point $-\boldsymbol{x}^{(\tilde{a})}$ is the most critical because the variance $\sigma^2_{\mathbb{D}_2}(\cdot)$ decreases along the direction of the arrows along the circle. On the right, a possible choice for the covering with additional training points is shown.

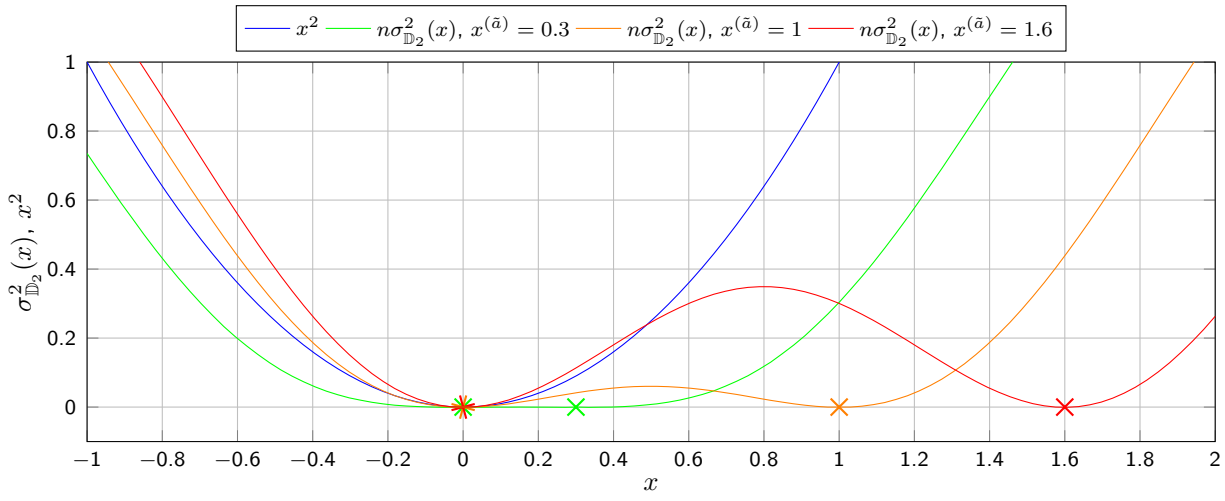

Figure 3.9: Projection of Example 3.3 on a single axis. The condition $x^2 < n\sigma^2_{\mathbb{D}_2}(x)$ holds $\forall x$ only if $x^{(\tilde{a})} = 0.3 < \tilde{r}_a$ (green). For $x^{(\tilde{a})} = 1$ (orange) it holds only in the positive domain, for $x^{(\tilde{a})} = 1.6$ (red), it is fulfilled nowhere around the origin.

If the optimization

$$\eta_{i_*} \coloneqq \min_{\substack{\boldsymbol{\theta}_{i_*} \in \boldsymbol{\Theta}_{i_*} \\ \hat{\boldsymbol{x}}_{i_*} \in \mathbb{X}_{i_*}}} \sum_{i \in \mathcal{I}_{i_*}} \mathfrak{g}\left(V_{\boldsymbol{\theta}_{i_*}}^{i_*}\left(\boldsymbol{y}^{(i)}\right) - V_{\boldsymbol{\theta}_{i_*}}^{i_*}\left(\boldsymbol{x}^{(i)}\right)\right), \tag{3.19}$$

results in $\eta_{i_*} = 0$, then the GP model $\bar{\boldsymbol{f}}(\cdot)$ in (3.8) is exact at the training data in the i_*-th region of attraction $\mathbb{X}_{i_*}$

$$\left\| \bar{\boldsymbol{f}}\left(\boldsymbol{x}^{(i)}\right) - \boldsymbol{y}^{(i)} \right\| = 0, \quad \forall i \in \mathcal{I}_{i_*}.$$

Proof. With the definition for $\mathfrak{g}(\cdot)$ in (3.18), it can be seen that $\eta_{i_*} = 0$ holds if and only if

$$V_{\boldsymbol{\theta}_{i_*}}^{i_*}\left(\boldsymbol{y}^{(i)}\right) - V_{\boldsymbol{\theta}_{i_*}}^{i_*}\left(\boldsymbol{x}^{(i)}\right) \leq 0, \quad \forall i \in \mathcal{I}_{i_*}. \tag{3.20}$$

Since the GP mean function matches the true function at the training points $\boldsymbol{\mu}\left(\boldsymbol{x}^{(i)}\right) = \boldsymbol{y}^{(i)}$ (Lemma 2.1), the constraint (3.7b) is fulfilled for $\boldsymbol{u} = \boldsymbol{0}$, which results in

$$\bar{\boldsymbol{f}}\left(\boldsymbol{x}^{(i)}\right) = \boldsymbol{\mu}\left(\boldsymbol{x}^{(i)}\right) = \boldsymbol{y}^{(i)}, \quad \forall i = 1, \ldots, N$$

which concludes the proof. □

Remark 3.9. *From the stability of the real system (Assumption 3.3) it can be concluded that there always exists a positive definite function $V : \mathbb{X}_{i_*} \to \mathbb{R}_{+,0}$ such that $\eta_{i_*} = 0$. However, it is unclear whether the chosen parameterization for $V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\cdot)$ is suitable and thus (3.20) is possibly violated for some data points. Nevertheless, the stability guarantees in Theorems 3.1 to 3.3 are not affected. The sub-optimality of the parameterization only affects the precision of the model because the stabilization signal $\boldsymbol{u}(\cdot)$ might become active where it should not, e.g. at the training data points.*

Example 3.4. *Consider an unknown system with a single equilibrium and the training data set*

$$\mathbb{D} = \left\{ \left(\begin{bmatrix} 1.3 \\ 0.8 \end{bmatrix}, \begin{bmatrix} 1.7 \\ 0.6 \end{bmatrix} \right), \left(\begin{bmatrix} 1.28 \\ 0.96 \end{bmatrix}, \begin{bmatrix} 1.58 \\ 0.66 \end{bmatrix} \right), \left(\begin{bmatrix} 1.28 \\ 0.64 \end{bmatrix}, \begin{bmatrix} 1.33 \\ 0.24 \end{bmatrix} \right) \right\}.$$

For a Lyapunov candidate $V_{Sq}(\boldsymbol{x}) = (\boldsymbol{x} - \hat{\boldsymbol{x}}_*)^\intercal \boldsymbol{P} (\boldsymbol{x} - \hat{\boldsymbol{x}}_*)$, the initial estimates are chosen to be $\boldsymbol{\theta} = \boldsymbol{P} = \boldsymbol{I}_2$ and $\hat{\boldsymbol{x}}_* = \begin{bmatrix} 0.7 & 0.8 \end{bmatrix}^\intercal$. As illustrated in Fig. 3.10, this results in a violation of condition (3.20) for two data points.

The optimal choices

$$\hat{\boldsymbol{x}}_*^* = \begin{bmatrix} 1.6 \\ 0.8 \end{bmatrix} \qquad \boldsymbol{\theta}^* = \begin{bmatrix} 1 & 0 \\ 0 & 2.5 \end{bmatrix}$$

ensure that the Lyapunov function decreases for all data points (see Fig. 3.10), thus $\eta = 0$.

Remark 3.10. *The definition of $\mathfrak{g}(\xi)$ is designed to penalize a violation of the condition (3.20), since it results in $\xi > 0$. In contrast, any data point which fulfills (3.20) does not increase (or decrease) the objective function. The form of $\tilde{\mathfrak{g}}(\cdot)$ is a design choice, where some allow to employ efficient solvers for (3.19) as discussed in Propositions 3.5 to 3.7. In general, it is advisable to choose a continuous function with $\lim_{\xi \to 0} \tilde{\mathfrak{g}}(\xi) = 0$ to avoid a discontinuous optimization problem.*

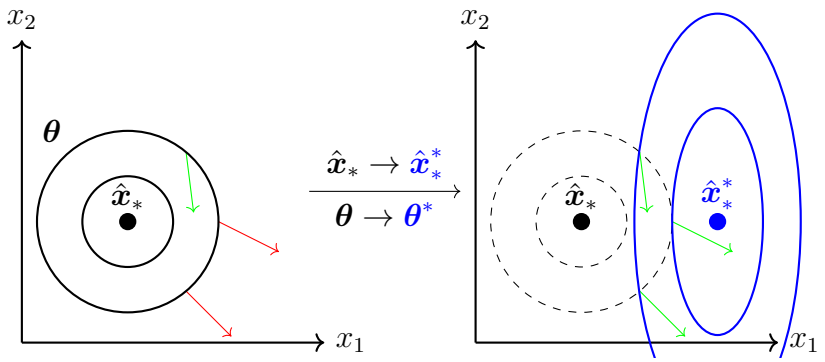

Figure 3.10: Illustration for the optimization-based learning of a Lyapunov function from data in (3.19). The optimal values for $\hat{\boldsymbol{x}}_*$ and $\boldsymbol{\theta}$ are denoted by $\hat{\boldsymbol{x}}_*^*$ and $\boldsymbol{\theta}^*$, respectively. The solid/dashed black/blue lines are level lines of a Lyapunov candidate, the red arrows denote data points violating the condition (3.20) and green arrows indicate data points in accordance to condition (3.20).

3.3.2 Specific Lyapunov candidates

This sections presents three specific Lyapunov candidates and discusses the resulting properties for the optimization (3.19). For notational convenience, we consider only a single equilibrium and use $V_{\boldsymbol{\theta}}$ for $V_{\boldsymbol{\theta}_{i_*}}^{i_*}(\cdot)$, $\hat{\boldsymbol{x}}_*$ for $\hat{\boldsymbol{x}}_{i_*}$, $\boldsymbol{\theta}$ for $\boldsymbol{\theta}_{i_*}$, $\boldsymbol{\Theta}$ for $\boldsymbol{\Theta}_{i_*}$ and $\mathbb{X}$ for $\mathbb{X}_{i_*}$.

Quadratic Lyapunov function This Lyapunov candidate is defined as

$$V_{\mathrm{Sq}}(\boldsymbol{x}) = (\boldsymbol{x} - \hat{\boldsymbol{x}}_*)^\intercal \boldsymbol{P}(\boldsymbol{x} - \hat{\boldsymbol{x}}_*), \qquad \boldsymbol{P} \succ 0, \tag{3.21}$$

with $\boldsymbol{\theta} = \boldsymbol{P}$ and $\boldsymbol{\Theta} = \mathbb{S}_+^n$. The following is concluded.

Proposition 3.5. *Consider a quadratic Lyapunov candidate* (3.21)*, a convex set* $\mathbb{X}$ *and a non-decreasing convex function* $\mathfrak{g}(\cdot)$*. Then, the optimization* (3.19) *is a biconvex optimization problem which is convex in* $\boldsymbol{\theta}$ *for constant* $\hat{\boldsymbol{x}}_*$ *and convex in* $\hat{\boldsymbol{x}}_*$ *for constant* $\boldsymbol{\theta}$*.*

Proof. For a fixed $\hat{\boldsymbol{x}}_*$, $V_{\mathrm{Sq}}(\cdot)$ is linear in $\boldsymbol{\theta}$. The difference of two linear function is linear and therefore the argument of $\mathfrak{g}(\cdot)$ is linear in $\boldsymbol{\theta}$. For a fixed $\boldsymbol{\theta}$, $V_{\mathrm{Sq}}(\cdot)$ is quadratic in $\hat{\boldsymbol{x}}_*$, but due to a cancellation of the squared terms, the argument of $\mathfrak{g}(\cdot)$ is linear in $\hat{\boldsymbol{x}}_*$. The composition of a linear function and the non-decreasing convex function $\mathfrak{g}(\cdot)$ is convex [134]. Since the sum of convex functions is convex, the objective function is convex. As the constraint sets are convex, the optimization problem is biconvex. □

Based on this property, the optimization of the parameters of $V_{\mathrm{Sq}}(\cdot)$ can be performed efficiently based on biconvex optimization algorithms [140]. However, the low flexibility with only a few parameters does not allow to represent a complex convergence behavior and η must generally be expected to be larger for the quadratic Lyapunov function than for the following Lyapunov functions with more parameters.

Sum of squares (SOS) Lyapunov functions A SOS [141] is a polynomial function defined as follows.

Definition 3.1. *For $\boldsymbol{x} \in \mathbb{R}^n$, a multivariate polynomial $p_{poly}(\boldsymbol{x})$ is a sum of squares (SOS) if there exist polynomials $r_m(\boldsymbol{x})$, $m = 1 \dots d_{poly}$ for which*

$$p_{poly}(\boldsymbol{x}) = \sum_{m=1}^{d_{poly}} r_m^2(\boldsymbol{x}). \tag{3.22}$$

Alternatively, SOS are characterized as follows [142].

Property 3.1. *A polynomial $p_{poly}(\boldsymbol{x})$ of degree $2d_{poly}$ is a SOS if and only if there exists a positive definite matrix $\boldsymbol{Q} \in \mathbb{S}_+^{n_{poly}}$, such that $p(\boldsymbol{x})$ can be formulated as*

$$p_{poly}(\boldsymbol{x}) = \boldsymbol{mon}(\boldsymbol{x})^\intercal \boldsymbol{Q} \boldsymbol{mon}(\boldsymbol{x}),$$

where $\boldsymbol{mon}(\boldsymbol{x}) \in \mathbb{R}^{n_{poly}}$ contains monomials[3] of degree greater zero and less or equal to d_{poly}.

This equivalence simplifies the construction of a SOS to a search for a positive semidefinite matrix $\boldsymbol{Q}$. For

$$V_{\mathrm{SOS}}(\boldsymbol{x}) = \mathbf{mon}(\boldsymbol{x} - \hat{\boldsymbol{x}}_*)^\intercal \boldsymbol{Q} \mathbf{mon}(\boldsymbol{x} - \hat{\boldsymbol{x}}_*), \quad \boldsymbol{Q} \succ 0, \tag{3.23}$$

with $\boldsymbol{\theta} = \boldsymbol{Q}$ and $\boldsymbol{\Theta} = \mathbb{S}_+^{n_{\mathrm{poly}}}$, the following holds.

Proposition 3.6. *Given a SOS Lyapunov candidate* (3.23) *and a non-decreasing convex function $\mathfrak{g}(\cdot)$, the optimization* (3.19) *is convex in $\boldsymbol{\theta}$ for a fixed $\hat{\boldsymbol{x}}_*$.*

Proof. The proof is analog to Proposition 3.5. □

Thus, by gaining more flexibility in the Lyapunov candidate, we have lost the computational efficiency to estimate the equilibrium point $\hat{\boldsymbol{x}}_*$ in a biconvex optimization.

Weighted sum of asymmetric quadratic functions (WSAQF) A weighted sum of asymmetric quadratic functions (WSAQF) Lyapunov function, proposed in [125] as

$$V_{\mathrm{WSAQF}}(\boldsymbol{x}) = (\boldsymbol{x} - \hat{\boldsymbol{x}}_*)^\intercal \boldsymbol{P}_0 (\boldsymbol{x} - \hat{\boldsymbol{x}}_*) + \sum_{l=1}^{L} \mathfrak{i}_l(\boldsymbol{x} - \hat{\boldsymbol{x}}_*) \left((\boldsymbol{x} - \hat{\boldsymbol{x}}_*)^\intercal \boldsymbol{P}_l \left(\boldsymbol{x} - \hat{\boldsymbol{x}}_* - \boldsymbol{x}_l^{\mathrm{WSAQF}} \right) \right)^2,$$
$$\text{with} \quad \mathfrak{i}_l(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{x}^\intercal \boldsymbol{P}_l \left(\boldsymbol{x} - \boldsymbol{x}_l^{\mathrm{WSAQF}} \right) \geq 0 \\ 0 & \text{otherwise,} \end{cases} \tag{3.24}$$

has continuous first order partial derivatives and is positive definite for $\boldsymbol{P}_0, \dots \boldsymbol{P}_L \in \mathbb{S}_+^n$ and $\boldsymbol{x}_1^{\mathrm{WSAQF}}, \dots, \boldsymbol{x}_L^{\mathrm{WSAQF}} \in \mathbb{X}$. Its parameters are accordingly

$$\boldsymbol{\theta} = \left\{ \boldsymbol{P}_0, \dots, \boldsymbol{P}_L, \boldsymbol{x}_1^{\mathrm{WSAQF}}, \dots \boldsymbol{x}_L^{\mathrm{WSAQF}} \right\}.$$

Due to a high number of parameters, it is also flexible, however the parameters $\boldsymbol{\theta}$ cannot be determined as efficiently as for SOS because (3.19) is biconvex as stated in the following.

[3] A monomial of degree d_{poly} in $\boldsymbol{x} \in \mathbb{R}^n$ is a scalar function $\prod_{j=1}^n x_j^{\Upsilon_j}$ where Υ_j are non-negative integers with $\sum_{j=1}^n \Upsilon_j = d_{\mathrm{poly}}$. The number of all possible monomials of degree $\leq d_{\mathrm{poly}}$ and > 0 is $n_{\mathrm{poly}} = \frac{(n+d_{\mathrm{poly}})!}{n! d_{\mathrm{poly}}!} - 1$.

Proposition 3.7. *Given the WSAQF Lyapunov candidate in* (3.24) *and a non-decreasing convex function* $\mathfrak{g}(\cdot)$*, the optimization* (3.19) *is a biconvex problem in* $\boldsymbol{\theta}$ *for fixed* $\hat{\boldsymbol{x}}_*$.

Proof. The proof is analogous to Proposition 3.5. □

This suggests that SOS is the most suitable choice because it allows a high degree of flexibility (compared to the quadratic Lyapunov function) and a more efficient computation of its parameters than the WSAQF Lyapunov function. This will be validated in the numerical evaluation in Sec. 3.4.

3.4 Evaluation

3.4.1 Evaluation setup

For the numerical evaluation, we work with the LASA handwriting dataset[4], which contains 24 goal-directed human motions in two dimensions ($n = 2$). There are 3 - 15 repetitions for each motion and all repetitions terminate at the same final location. Due to the generation by humans, there is no method known (to the author) to design a suitable parametric model describing the motion. Therefore, our data-driven nonparametric approach is required to model and generalize the given motions using a dynamical system.

The common property of all motions is their convergence to a single goal point and to preserve this key feature, the dynamical system model must be stable. Therefore, our idea to stabilize a GPSSM is applicable here for this dataset.

We will assume that all trajectories are generated by a system of the form (3.1), which is asymptotically stable (Assumption 3.3) with the final location being the single equilibrium point $\boldsymbol{x}_* = \mathbf{0}$ ($N_* = 1$, $\mathbb{X} = \mathbb{R}^2$). The task is to identify a model of the form (3.4) and (3.5), which generates trajectories with a high similarity to the demonstrations and also converges to (the estimated) equilibrium.

Before presenting the results, we first discuss some implementation details in Sec. 3.4.2. Then, we focus on the estimation of an unknown equilibrium based on the optimization (3.19) in Sec. 3.4.3. Second, in Sec. 3.4.4, we consider the single equilibrium point to be known ($\boldsymbol{x}_* = \hat{\boldsymbol{x}}_* = \mathbf{0}$) to perform a quantitative comparison for the deterministic GP model on the full dataset (24 motions) among the different Lyapunov candidates. Finally, in Sec. 3.4.5 we simulate the probabilistic case showing how crucial the stabilization is.

3.4.2 Implementation

Before presenting the results, we first provide an algorithmic overview of the proposed approach in Algorithm 3.1 and details regarding the implementation[5].

As already mentioned, the measurements acquired in Line 1 are taken from the LASA dataset. In the first step of the proposed approach, the GPs are trained (line 2) based on the training data using a quasi-Newton method to perform the optimization (2.6). In Line 7, the optimal Lyapunov function based on (3.19) is determined using an interior-point method. We do not explicitly exploit the (bi-)convexity with a specialized algorithm

[4] Available for download at `https://bitbucket.org/khansari/seds`

[5] Code for MATLAB is provided: `https://gitlab.lrz.de/ga68car/stablegps.git`

Algorithm 3.1 The proposed stable identification and prediction using GPSSMs.

take measurements $\mathbb{D}$ at arbitrary locations (Assumption 3.2)
optimize hyperparameters $\boldsymbol{\psi}_j$, $\forall j$ according to (2.6)
if probabilistic case $\wedge$ $n\bar{\zeta}^2 > \underline{\ell}^2$ **then**
    take N_{add} further measurements $\mathbb{D}_{\text{add}}$ at specific locations according to Theorem 3.3
end if
select region of attraction i_* s.t. $\boldsymbol{x}_0 \in \mathbb{X}_{i_*}$
optimize $\boldsymbol{\theta}_{i_*}$ of the Lyapunov function $V^{i_*}_{\boldsymbol{\theta}_{i_*}}(\cdot)$ using $\mathbb{D}$ ($\cup\, \mathbb{D}_{\text{add}}$) and estimate the equilibrium $\hat{\boldsymbol{x}}_{i_*}$ using (3.19)
initialize $\kappa = 0$
while number of desired simulation steps not exceeded **do**
    if deterministic case **then**
        determine optimal $\boldsymbol{u}(\boldsymbol{x}_\kappa)$ based on (3.7)
        set $\boldsymbol{x}_{\kappa+1} = \boldsymbol{\mu}(\boldsymbol{x}_\kappa) + \boldsymbol{u}(\boldsymbol{x}_\kappa)$
    else
        determine optimal $\boldsymbol{u}(\boldsymbol{x}_\kappa)$ based on (3.13)
        sample $\boldsymbol{\omega}_\kappa \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_n)$
        set $\boldsymbol{x}_{\kappa+1} = \boldsymbol{\mu}(\boldsymbol{x}_\kappa) + \boldsymbol{u}(\boldsymbol{x}_\kappa) + \sqrt{\boldsymbol{\Sigma}(\boldsymbol{x}_\kappa)}\boldsymbol{\omega}_\kappa$
    end if
    update $\kappa \leftarrow \kappa + 1$
end while

to allow a fair comparison of all proposed Lyapunov candidates. Nevertheless, finding the global optimum is more promising for the SOS and quadratic Lyapunov function. According to Propositions 3.5 and 3.6, we chose $\tilde{\mathfrak{g}}(\xi) = \xi$ as it results in a non-decreasing monotone function $\mathfrak{g}(\cdot)$.

For efficient implementation, we perform a Cholesky decomposition for all positive definite matrices and optimize over the nonzero entries of the lower triangular matrix. This avoids to explicitly impose the symmetry constraint. Furthermore, the optimization (3.19) also requires a regularization term to ensure that the objective function is not minimized by setting all parameters in $\boldsymbol{Q}, \boldsymbol{P}, \boldsymbol{P}_0, \dots \boldsymbol{P}_L$ close to zero. Instead, we impose a lower bound on the eigenvalues of the positive definite matrices by defining

$$\mathbb{S}^n_{\epsilon_{\underline{\varpi}}} = \left\{ \boldsymbol{P} \in \mathbb{S}^n_+ \,\middle|\, \underline{\varpi}(\boldsymbol{P}) \geq \epsilon_{\underline{\varpi}} \right\}, \quad n \in \mathbb{N}$$

and use it instead of $\mathbb{S}^m_+$ to define the constraint set $\boldsymbol{\Theta}$. This will not affect the statements in Propositions 3.5 to 3.7 because the constraint $\epsilon_{\underline{\varpi}} - \underline{\varpi}(\boldsymbol{P}) < 0$ is convex (the eigenvalue operation is linear and the minimum of a linear function is concave, its negative is convex). Here we have chosen $\epsilon_{\underline{\varpi}} = 0.01$.

Remark 3.11. *For many practical applications, the equilibrium point of the unknown system* (3.1) *might be known a priori. In this case, the Lyapunov function search is simplified as the optimization is only over* $\boldsymbol{\theta}$ *and therefore convex for* $V_{Sq}(\cdot)$ *and* $V_{SOS}(\cdot)$.

The simulation is initialized once at each starting point of the training trajectories and runs until the state reaches proximity of the (estimated) equilibrium point $\|\boldsymbol{x} - \hat{\boldsymbol{x}}_*\| < 5$ or a limit of 1000 steps (choice depends on the application scenario and the willingness to wait for the

simulation result). The optimization (3.7) and (3.13) (in Line 11 and Line 14, respectively) are performed at each step by the interior-point algorithm, initialized with $\boldsymbol{u} = \hat{\boldsymbol{x}}_* - \boldsymbol{\mu}(\boldsymbol{x})$. The constraints (3.7b) and (3.13b) are rewritten to

$$\begin{aligned} V(\boldsymbol{\mu}(\boldsymbol{x}_\kappa) + \boldsymbol{u}) - V(\boldsymbol{x}_\kappa) &\leq -\epsilon_V V(\boldsymbol{x}_\kappa) \\ \delta V(\boldsymbol{x}_\kappa) &\leq -\epsilon_V V(\boldsymbol{x}_\kappa) \end{aligned}$$

with $\epsilon_V = 0.02$ for the deterministic and probabilistic case, respectively to avoid the strict inequalities.

The Lyapunov functions presented in Sec. 3.3.2 are utilized as follows.

- A quadratic Lyapunov function $V_{\mathrm{Sq}}(\cdot)$ defined in (3.21) with $n(n+1)/2 = 3$ free parameters (the elements of the Cholesky decomposition of $\boldsymbol{P}$).
- The SOS Lyapunov function $V_{\mathrm{SOS}}(\cdot)$ defined in (3.23) with degree $2d_{\mathrm{poly}} = 4$, resulting in $n_{\mathrm{poly}} = 5$ monomials and therefore $n_{\mathrm{poly}}(n_{\mathrm{poly}}+1)/2 = 15$ free parameters (Cholesky decomposition of $\boldsymbol{Q}$).
- The WSAQF Lyapunov function $V_{\mathrm{WSAQF}}(\cdot)$ defined in (3.24) with $L = 3$ resulting in $(L+1)n(n+1)/2 + nL = 18$ free parameters for the Cholesky decompositions of $\boldsymbol{P}_0, \ldots, \boldsymbol{P}_3$ and $\boldsymbol{x}_1^{\mathrm{WSAQF}}, \ldots, \boldsymbol{x}_3^{\mathrm{WSAQF}}$.

3.4.3 Equilibrium estimation

For this first example, we assume the true equilibrium point $\boldsymbol{x}_* = \boldsymbol{0}$ to be unknown and only consider the SOS Lyapunov function. Thus, for the optimization (3.19) a total of 17 (15 for SOS, 2 for the equilibrium point) variables are optimized. We utilize the motion called *Multi-Model 1* from the LASA dataset in this example.

The result for the simulation of the deterministic case is shown Fig. 3.11. The equilibrium is identified to $\hat{\boldsymbol{x}}_* \approx \begin{bmatrix} 0.8120 & -2.3987 \end{bmatrix}^\intercal$, which is sufficiently close to the origin (the true equilibrium) as the simulation is terminated for $\|\boldsymbol{x}_\kappa - \hat{\boldsymbol{x}}_*\| < 5$.

3.4.4 Quantitative comparison

To enable a quantitative comparison of the Lyapunov functions presented in Sec. 3.3.2, we define the following measures for the precision of the learned model.

- The total area between the curve connecting the training points and the curve connecting the simulated trajectory (see Fig. 3.12)
- The average correction effort $E_{\mathrm{cor}} \in \mathbb{R}_+$ defined as $E_{\mathrm{cor}} = \sum_\kappa \|\boldsymbol{u}_\kappa\| \Big/ \sum_\kappa \|\boldsymbol{x}_\kappa\|$ over all time steps and trajectories.

The results are shown in Table 3.1. Regarding flexibility, SOS outperforms the quadratic and WSAQF Lyapunov function on the employed dataset as shown in Table 3.1 since it leads to the highest precision in both employed measures. Regarding the computational complexity, the optimization of the parameters of the quadratic and the SOS Lyapunov function (3.19) are convex problems, and therefore have significant advantages over the

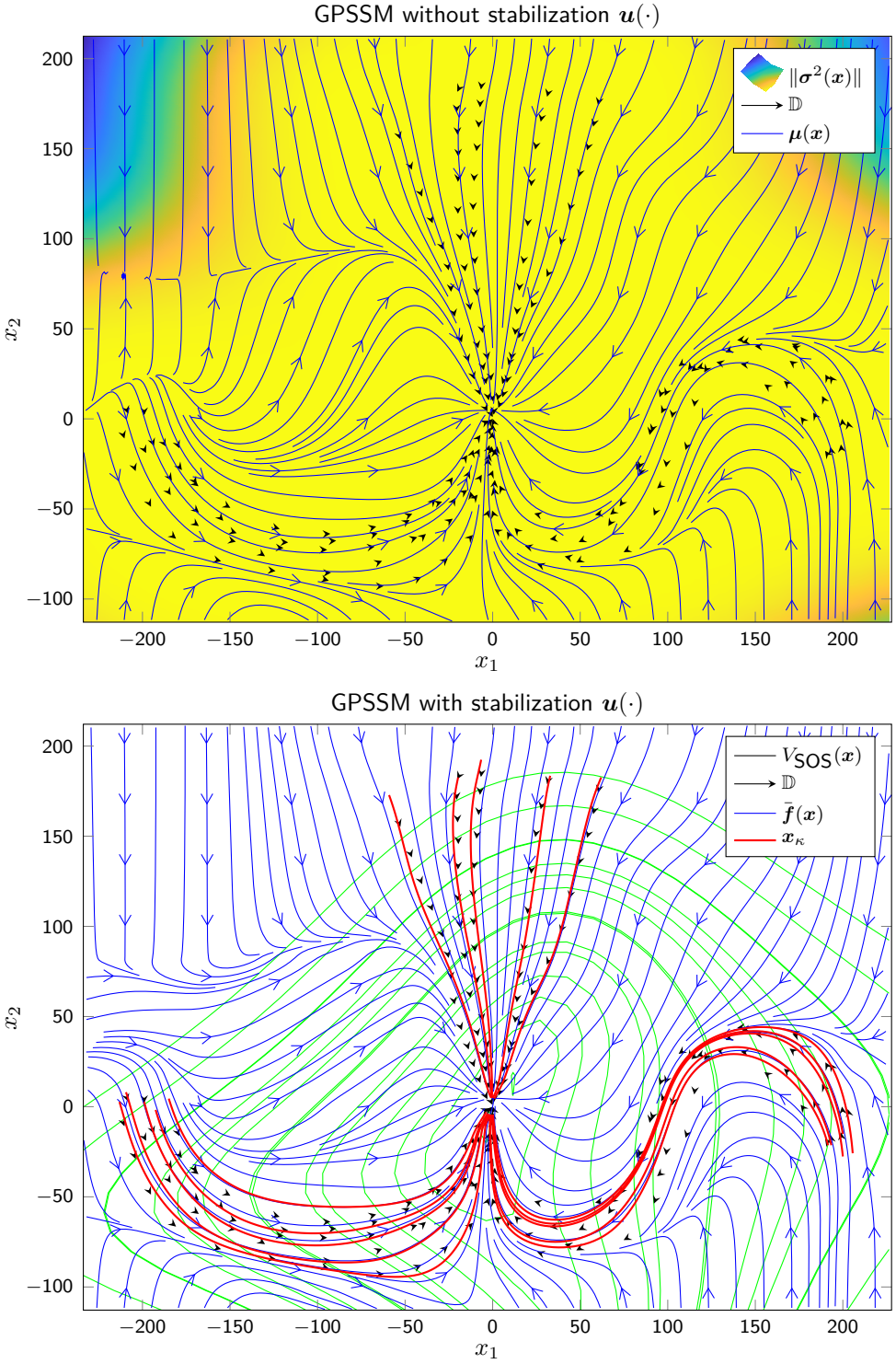

Figure 3.11: For the motion *Multi-Model 1* (training data in black), the GPSSM without stabilization $\boldsymbol{x}_{\kappa+1} = \boldsymbol{\mu}(\boldsymbol{x}_\kappa)$ (top) and the stabilized model $\boldsymbol{x}_{\kappa+1} = \boldsymbol{\mu}(\boldsymbol{x}_\kappa) + \boldsymbol{u}(\boldsymbol{x}_\kappa)$ (bottom) are visualized (blue). Without stabilization (top), there is an undesired equilibrium near $\begin{bmatrix}-200 & 80\end{bmatrix}^\mathsf{T}$. The trajectories (red) converge asymptotically to the estimated equilibrium $\hat{\boldsymbol{x}}_* = \begin{bmatrix}0.81 & -2.40\end{bmatrix}^\mathsf{T}$. Level lines of the SOS control Lyapunov function are shown in green.

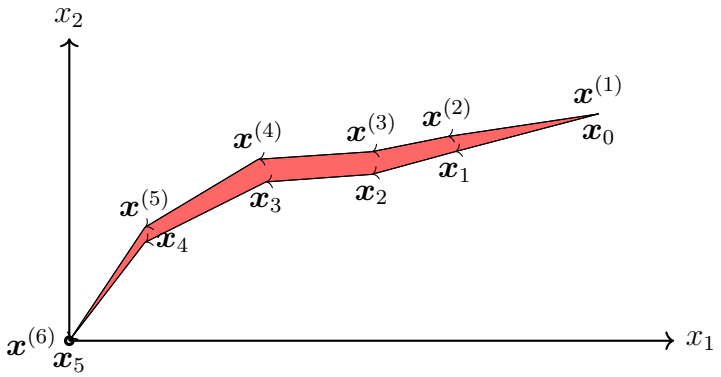

Figure 3.12: An illustration of the area error. The area shown in red measures how far the trajectory generated by the model $\boldsymbol{x}_{\kappa+1} = \bar{\boldsymbol{f}}(\boldsymbol{x}_\kappa)$, $\boldsymbol{x}_0, \dots, \boldsymbol{x}_5$ is from the training trajectory $\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(6)}$.

WSAQF. Computing the stabilizing command for SOS has a disadvantage compared to WSAQF and quadratic Lyapunov functions as (3.7) is not convex. However, this drawback is inherent to flexible Lyapunov candidates: If only convex Lyapunov candidates are permitted, systems with a complex convergence behavior cannot be represented and thereby lead to imprecise modeling.

3.4.5 Probabilistic simulation

To evaluate the probabilistic setting discussed in (3.2), we compare simulations for the following two settings.

- GPSSM without stabilization: Realizations of the trajectories are generated by drawing in each step from

$$\boldsymbol{x}_{\kappa+1} \sim \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{x}_\kappa), \boldsymbol{\Sigma}(\boldsymbol{x}_\kappa)), \tag{3.25}$$

  using the definitions in (2.7)/(2.8) for $\boldsymbol{\mu}(\cdot)$ and $\boldsymbol{\Sigma}(\cdot)$.

- GPSSM with stabilization through the SOS Lyapunov function: At each step $\boldsymbol{u}(\cdot)$ is computed according to (3.13) and the next step is drawn from

$$\boldsymbol{x}_{\kappa+1} \sim \mathcal{N}\left(\boldsymbol{\mu} + \boldsymbol{u}(\boldsymbol{x}_\kappa), \boldsymbol{\Sigma}(\boldsymbol{x}_\kappa)\right), \tag{3.26}$$

  which is equivalent to (3.5). The stabilizing command is independent of the realization of $\boldsymbol{x}_{\kappa+1}$, which is unknown when $\boldsymbol{u}$ is computed. It only depends on $\boldsymbol{\mu}(\boldsymbol{x}_\kappa)$ and $\boldsymbol{\Sigma}(\boldsymbol{x}_\kappa)$.

For both cases, the system is initialized twice with $\boldsymbol{x}_0 = \begin{bmatrix}-150 & -120\end{bmatrix}^\mathsf{T}$ as shown in Fig. 3.13. For the GPSSM without stabilization both trajectories do not converge in contrast to the trajectories of the stabilized model, which are guaranteed to converge. Here only ultimate boundedness holds because $n\bar{\zeta}^2 > \underline{\ell}^2$, with $\underline{\ell}^2 \approx 23.7266$, and $\bar{\zeta}^2 \approx 88.3054$. The ultimate bound $\iota \approx 13.2856$ is guaranteed according to Theorem 3.2. To achieve asymptotic stability, $N_{\text{add}} = 10$ additional data points would be necessary according to Theorem 3.3.

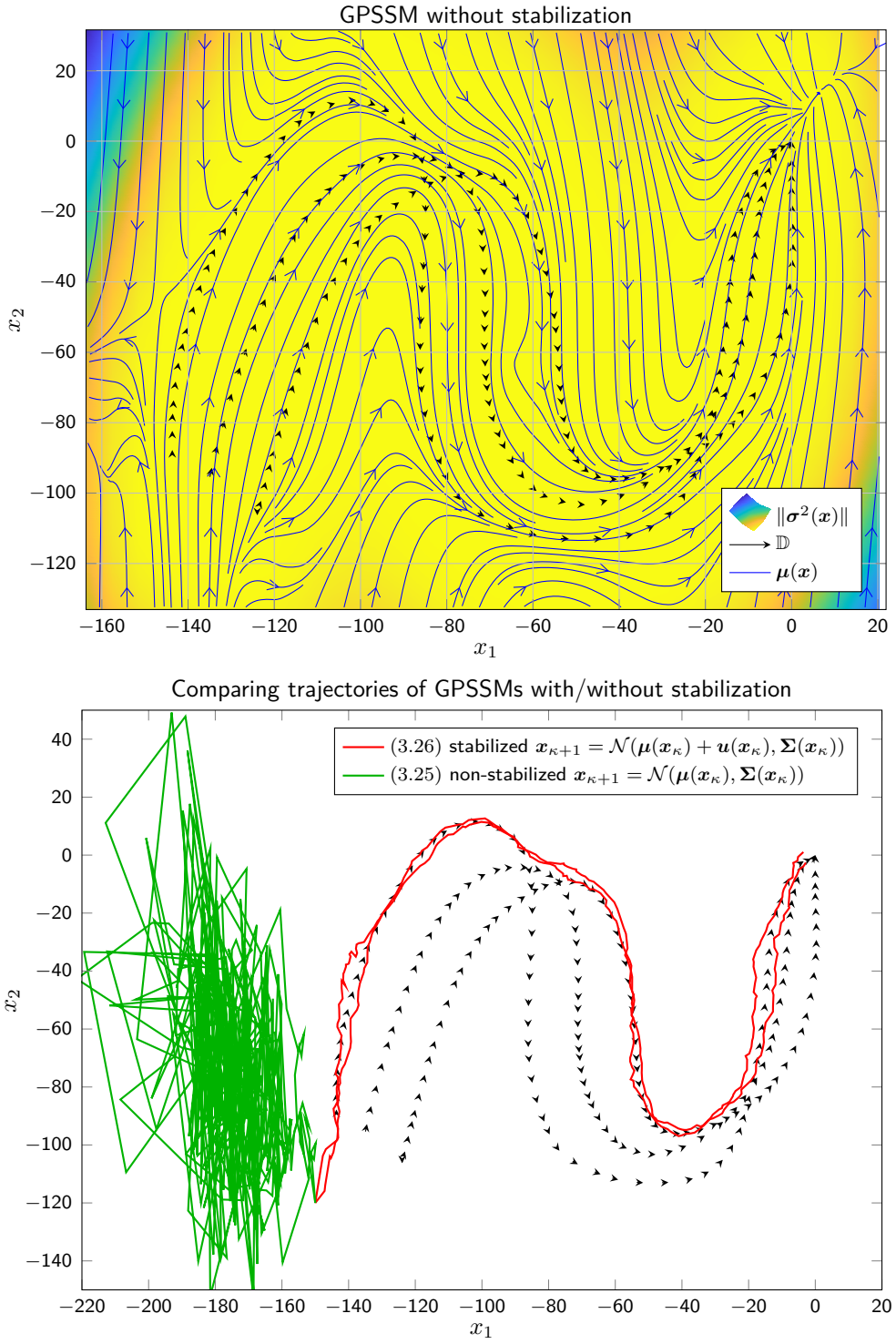

Figure 3.13: The mean and variance of a GPSSM are visualized on the top, the stochastic simulations on the bottom. The black arrows indicate the training data for the *N-Shape* motion. The stabilized trajectories are shown in red, the trajectories without stabilization in green. For both cases, the model is initialized twice at $\begin{bmatrix} -150 & -120 \end{bmatrix}^\intercal$.

$V(\cdot)$	E_{cor}	Area Error	(3.19) convex	(3.7) convex
quadratic	0.22	8.1e+03	yes	yes
SOS	**0.038**	**3.3e+03**	yes	no
WSAQF	0.074	4.0e+03	no	yes

Table 3.1: Average performance and properties of quadratic, SOS and WSAQF Lyapunov functions on the LASA handwriting dataset. The convexity of (3.19) refers to the case for which $\boldsymbol{x}_*$ is known.

3.5 Discussion

The presented analysis is mainly focused on the SE kernel. While the results for the deterministic case directly hold for other kernel classes as well, the results in the stochastic case (Theorems 3.2 and 3.3) must be reevaluated. The ultimate boundedness results should be extendable to other bounded kernels, but the derivation of the asymptotic stability conditions might not be possible. We leave this investigation to future work.

In Theorem 3.3, we derive how many data points are needed for asymptotic stability, but we did not discuss where these must be positioned for efficient covering or how to obtain these measurements. For the first, we refer to existing literature on covering, e.g., [143], [144] and [139]. For the latter, we refer to Chapter 5 and the literature [66], where the safe exploration using learning control is discussed.

This contribution has mainly focused on transferring knowledge about the convergence behavior of the real system into a non-parametric model. Nevertheless, there exist numerous other examples where high level knowledge is available and must be transferred into the model to achieve physical consistency. One of them is constrained dynamics of mechanical systems as discussed in [145], where specific kernels are chosen to ensure the model adheres to the Gauss' principle.

While the dynamics are learned by a nonparametric GP model, the convergence behavior is captured in parametric Lyapunov functions. This does not consequently follow the data-driven concept and requires prior knowledge to choose a proper structure for the Lyapunov candidates. With nonparametric Lyapunov candidates with universal approximation properties a consistent data-driven concept can be realized. The work in [146] shows promising results based on an approximate infinite horizon cost function, but is also computationally more demanding.

In addition, there are currently no guarantees on how far the estimated equilibria are from the true equilibria. The simulation suggests that the presented technique leads to acceptable estimates if the training trajectories approach the equilibrium from different directions. However, a rigorous theoretical investigation is missing.

3.6 Summary

This chapter proposes a novel approach for learning stable GPSSMs with multiple equilibria using control Lyapunov functions. By providing an algorithm which includes fundamental physical properties (here energy dissipation / stability) into the data-driven identification, we take a step to solve Challenge 1. By investigating not only the deterministic interpretation of the GP, but also considering stability of the stochastic model, we lay the foundation for

uncertainty aware control laws (Challenge 3).

For the deterministic case, we show (global) asymptotic stability of the model and prove that the stabilization will never reduce the precision of the GP model (if the Lyapunov functions are chosen correctly). For the stochastic case, we show (global) ultimate boundedness and derive an upper bound for the number of additional training data which are required to ensure asymptotic stability independently of the kernel parameters.

A generic framework to learn the convergence behavior of dynamical systems from training data is presented in this chapter. This includes data-driven optimization-based adaptation of a parametric Lyapunov candidate and an estimation of the equilibria. Furthermore, we derive (bi-)convexity results for different types of Lyapunov candidates to ensure the required optimization can be performed efficiently. The numerical analysis based on a real-world human motion data set shows that SOS Lyapunov functions are well suited for data-driven learning of the convergence behavior of dynamical systems.

Uncertainty-based Control Lyapunov Design

4

Uncertainties in a control loop can arise from various sources, e.g., measurements of the output can be corrupted by noise, the process itself can be of stochastic nature or the model of the process is imprecise. Incorporating these uncertainties into the control design becomes particularly important in safety-critical applications and various techniques have been developed to handle uncertainties properly.

Robust control is a basic concept to ensure proper operation under a bounded model error and allows to provide stability guarantees for closed-loop systems under specific assumptions [147]. However, these assumptions are often not easy to justify and lead to a conservative strategy, which sacrifices control performance [148]. Alternatively, stochastic control can also handle uncertainties in the system and perform a profound stability analysis based on the stochastic differential equations [128], [149], [150]. But this technique considers uncertainty as a random behavior, rather than ignorance of the true dynamics.

Based on optimal control, various differential dynamic programming approaches are extended to deal with uncertain systems [44]. These predictive control techniques take the future uncertainty implicitly into account for their control decision [151]. Extensions which allow to explicitly weight the future uncertainties in the cost function are presented in [129], [152], [153] known as risk-sensitive or uncertainty-dependent control design. However, stability or safety guarantees for these methods do not exist.

Since GPs provide a model fidelity measure, they are well suited for an uncertainty-based control design. Nevertheless, a large body of the GP based control literature does not take it explicitly into account for the design. The reinforcement learning approaches in [32], [33] propagate uncertainty in the prediction of the roll-outs and thereby implicitly evaluate the uncertainty in the cost function. However, uncertainties are not explicitly avoided, since exploration (visiting areas with poor models knowledge) is actively pursuit. For safety-critical applications such behavior is not acceptable.

The application in robotics utilize the GP only as function approximation and therefore drop the additional information regarding the uncertainty [72], [74]. The work in [102] controls the leader-follower behavior in multi-agent systems to ensure the agent with more knowledge (less uncertainty) takes over a leading role, while others follow (see Sec. 6.1 for more details). Tuning the gains of a feedback control law based on this uncertainty measure is proposed in [82] and conditions for ultimate boundedness are derived in [84]. While this approach increases safety in human-robot interaction due to a less aggressive controller whenever possible, it does not address the fundamental problem to avoid areas, where the model is poorly trained. Particularly in data-driven models it is essential to bypass areas without training data, since the behavior of the system is unknown and therefore possibly harmful.

This chapter's major contribution is an uncertainty-based control approach which asymptotically stabilizes an initially unknown system with high probability. The key idea is to favor areas of the state space, where the uncertainty is low using dynamic programming approaches. We show that this behavior is the safest under a given power constraint. Thereby, we reveal an equivalence of maximizing the probability for stability and the minimization of uncertainty along the trajectory, which is intuitive but - to the best of our knowledge - has not been shown so far. By using the value function of an optimal control problem as uncertainty-aware Lyapunov function, we are able to use algorithms from path planning, which ensure high computational efficiency.

The chapter is based on the work published in [9]. It is structured as follows: After defining the problem setting in Sec. 4.1, this chapter proposes the uncertainty-aware control approach in Sec. 4.2 and analyses its properties. A numerical illustration is provided in Sec. 4.3 followed by a discussion in Sec. 4.4.

4.1 Problem formulation

In this chapter, a fully actuated control-affine system is considered

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u}, \quad \boldsymbol{x}(0) = \boldsymbol{x}_0, \tag{4.1}$$

with state $\boldsymbol{x} \in \mathbb{X} \subset \mathbb{R}^n$, compact $\mathbb{X}$ and input $\boldsymbol{u} \in \mathbb{U} \subseteq \mathbb{R}^n$. The goal is an asymptotic stabilization at an arbitrary point $\boldsymbol{x}_g \in \mathbb{X}$ considering the following assumptions:

Assumption 4.1. *The function $\boldsymbol{f} : \mathbb{X} \to \mathbb{R}^n$ is unknown but the function value $\boldsymbol{f}(\boldsymbol{x}_g)$ is known.*

Without loss of generality, we take $\boldsymbol{x}_g$ as an equilibrium point at the origin, thus $\boldsymbol{f}(\boldsymbol{0}) = \boldsymbol{0}$. The assumption is easy to fulfill as only a single (noise free) measurement of the system without control input is required.

The uncontrolled dynamics $\boldsymbol{f}(\cdot)$ is unknown, however, as already discussed in Sec. 2.4.1 general prior knowledge must be assumed. Here, as already explained in Assumption 2.1, the complexity of the function in terms of its RKHS norm is limited.

Assumption 4.2. *Each element in $\boldsymbol{f}(\cdot) = \begin{bmatrix} f_1(\cdot) & \cdots & f_n(\cdot) \end{bmatrix}^\intercal$ has a bounded RKHS norm under a known kernel $\mathfrak{k}_j(\cdot,\cdot)$, thus $\|f_j\|_{\mathfrak{k}_j} < B_{f_j}$ for $j = 1, \ldots, n$.*

Furthermore, the effect of the control input on the state is known and the full actuation holds across the entire state space.

Assumption 4.3. *The function $\boldsymbol{G}\colon \mathbb{X} \to \mathbb{R}^{n\times n}$ is known, differentiable and invertible, thus rank$(\boldsymbol{G}(\boldsymbol{x})) = n$, $\forall \boldsymbol{x} \in \mathbb{X}$.*

This limits the class of systems which are considered, nevertheless it holds for Lagrangian systems, which are considered to be a quite general class. An example is a coupled multiple tank system with separate actuated inflows for each tank.

Remark 4.1. *An extension to systems which are not fully actuated is provided in Sec. 4.2.4.*

To allow data-driven modeling of the unknown dynamics $\boldsymbol{f}(\cdot)$, we assume the availability of training data.

Assumption 4.4. *A training set with $N \in \mathbb{N}$ data pairs consisting of the state and a noisy measurement of its derivative*

$$\mathbb{D} = \left\{ \left(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)} \right) \right\}_{i=1}^{N}, \quad \boldsymbol{y}^{(i)} = \boldsymbol{f}\left(\boldsymbol{x}^{(i)} \right) + \boldsymbol{\omega}^{(i)}, \tag{4.2}$$

is available, where $\boldsymbol{\omega}^{(i)}$ are i.i.d. samples of $\boldsymbol{\omega} \sim \mathcal{N}(0, \sigma_{on}^2 \boldsymbol{I}_n)$, $\sigma_{on}^2 \in \mathbb{R}_{+,0}$.

We leave it open how this training data is obtained, but we also do not impose any restriction where the data must be collected.

Remark 4.2. *Note that for a well-behaved dynamical system without finite escape time, the data can be collected in (short) uncontrolled runs of the system for a finite time interval.*

Remark 4.3. *Today's improved sensor technology justifies the assumption on perfect measurements of the state. If this cannot be argued, we refer to the literature [154] dealing with inference for noisy input. However, extending our approach to this case is left for future work. The measurements of the state's derivative can be corrupted by noise, which allows e.g., for finite difference approximations.*

The goal is to design a state feedback control law which asymptotically stabilizes the system (4.1) with high probability. Based on feedback linearization, we develop a robust control Lyapunov approach using a GP model of the system. We aim to maximize the probability for stabilization through an optimal choice of the Lyapunov function.

4.2 Control design and analysis

To realize a feedback linearizing control, a model of the unknown dynamics is required. We utilize GP regression as introduced in Sec. 2.1 and include the known equilibrium point at the origin (from Assumption 4.1) according to Remark 2.3 with $\boldsymbol{x}_{\mathrm{kn}} = \boldsymbol{0}$, $y_{\mathrm{kn}} = 0$. Thus, for the model estimate $\hat{\boldsymbol{f}}(\cdot)$, we utilize the GP posterior mean predictions for multiple dimensions (as proposed in Remark 2.2) $\hat{\boldsymbol{f}}(\cdot) = \boldsymbol{\mu}(\cdot)$, where all kernels are SE covariance functions as defined in (2.5) and the prior mean function is set to zero. An extension to other classes of kernels or prior mean functions is possible, but not discussed in this thesis.

The proposed control law is then given as

$$\boldsymbol{u}(\boldsymbol{x}) = -\boldsymbol{G}^{-1}(\boldsymbol{x}) \left(\hat{\boldsymbol{f}}(\boldsymbol{x}) + k_u \nabla_{\boldsymbol{x}}^{\mathrm{I}} V_{\mathrm{clf}}(\boldsymbol{x}) \right), \tag{4.3}$$

where $k_u \in \mathbb{R}_+$ and $V_{\mathrm{clf}} : \mathbb{X} \to \mathbb{R}_+$ is a positive definite differentiable function employed as control Lyapunov function. The term $-\boldsymbol{G}^{-1}(\boldsymbol{x})\hat{\boldsymbol{f}}(\boldsymbol{x})$ compensates the nonlinear system. However, different from the classical feedback linearization, $k_u \nabla_{\boldsymbol{x}}^{\mathrm{I}} V_{\mathrm{clf}}(\boldsymbol{x})$ is not a linear control law, but is a robustifying term derived from a control Lyapunov function.

First, in Sec. 4.2.1, we generally derive sufficient conditions on $V_{\mathrm{clf}}(\cdot)$ and k_u to ensure stability of the closed-loop system with high probability. Then, in Sec. 4.2.2, we analyze how to maximize this probability under an input power constraint. Finally, in Sec. 4.2.3, we propose an optimal control Lyapunov function, which can efficiently be computed as it is based on dynamic programming principles.

4.2.1 Conditions for asymptotic stability

For the control law (4.3), the following is concluded.

Theorem 4.1. *Consider the unknown system* (4.1) *under Assumptions 4.1 to 4.4 where* $\boldsymbol{f}(\cdot)$ *is modeled by a GP* (2.7)/(2.8) *with SE kernel* (2.5)*, denoted as* $\hat{\boldsymbol{f}}(\cdot)$*. Further consider the control law* (4.3) *with* $k_u > \|\boldsymbol{\beta}\|$ *and* $V_{clf}(\cdot)$ *is a positive definite differentiable function with* $\|\nabla_{\boldsymbol{x}} V_{clf}(\mathbf{0})\| = 0$ *and*

$$\|\boldsymbol{\sigma}(\boldsymbol{x})\| - \|\nabla_{\boldsymbol{x}} V_{clf}(\boldsymbol{x})\| \leq 0, \qquad \boldsymbol{x} \in \mathbb{X}. \tag{4.4}$$

Then, the origin of the closed-loop system is semiglobally asymptotically stable with probability at least $1 - \sum_{j=1}^{n} \delta_j$ *for all* $\boldsymbol{x}_0 \in \mathbb{X}$*, where* δ_j *and* $\boldsymbol{\beta}$ *are defined in Proposition 2.1.*

Proof. Choosing $V_{\mathrm{clf}}(\cdot)$ as Lyapunov candidate yields

$$\begin{aligned}
\dot{V}_{\mathrm{clf}}(\boldsymbol{x}) &= \nabla_{\boldsymbol{x}} V_{\mathrm{clf}}(\boldsymbol{x}) \dot{\boldsymbol{x}} = \nabla_{\boldsymbol{x}} V_{\mathrm{clf}}(\boldsymbol{x}) (\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x}) \boldsymbol{u}(\boldsymbol{x})) \\
&= \nabla_{\boldsymbol{x}} V_{\mathrm{clf}}(\boldsymbol{x}) \left(\boldsymbol{f}(\boldsymbol{x}) - \hat{\boldsymbol{f}}(\boldsymbol{x}) - k_u \nabla_{\boldsymbol{x}}^{\intercal} V_{\mathrm{clf}}(\boldsymbol{x}) \right) \\
&\leq \|\nabla_{\boldsymbol{x}} V_{\mathrm{clf}}(\boldsymbol{x})\| \left\| \boldsymbol{f}(\boldsymbol{x}) - \hat{\boldsymbol{f}}(\boldsymbol{x}) \right\| - k_u \|\nabla_{\boldsymbol{x}} V_{\mathrm{clf}}(\boldsymbol{x})\|^2,
\end{aligned}$$

where the inequality results from the Cauchy-Schwarz inequality. Proposition 2.1 yields

$$\mathcal{P}\left\{ \dot{V}_{\mathrm{clf}}(\boldsymbol{x}) \leq \|\nabla_{\boldsymbol{x}} V_{\mathrm{clf}}(\boldsymbol{x})\| \left(\|\boldsymbol{\beta}\| \|\boldsymbol{\sigma}(\boldsymbol{x})\| - k_u \|\nabla_{\boldsymbol{x}} V_{\mathrm{clf}}(\boldsymbol{x})\| \right), \forall \boldsymbol{x} \in \mathbb{X} \right\} \geq 1 - \sum_{j=1}^{n} \delta_j,$$

which results under condition (4.4) and $k_u > \|\boldsymbol{\beta}\|$ in

$$\mathcal{P}\left\{ \boldsymbol{x} \in \mathbb{X} \setminus \{\mathbf{0}\},\ \dot{V}_{\mathrm{clf}}(\boldsymbol{x}) < 0 \right\} \geq 1 - \sum_{j=1}^{n} \delta_j.$$

The strict inequality holds because $\|\nabla_{\boldsymbol{x}} V_{\mathrm{clf}}(\boldsymbol{x})\|$ is lower bounded by the positive definite function $\|\boldsymbol{\sigma}(\boldsymbol{x})\|$. Additionally, $\dot{V}_{\mathrm{clf}}(\mathbf{0}) = 0$ holds from $\|\nabla_{\boldsymbol{x}} V_{\mathrm{clf}}(\mathbf{0})\| = 0$. As $\mathbb{X}$ can be chosen arbitrarily large (however must be compact as required by Proposition 2.1) the asymptotic stability holds semiglobally. □

Figure 4.1 visualizes the idea of the proof. It imposes two sufficient conditions on the design of the control law: i) the gain k_u must be chosen large enough and ii) the norm of the gradient of the Lyapunov function $\|\nabla_{\boldsymbol{x}} V_{\mathrm{clf}}(\cdot)\|$ most be higher than the uncertainty $\|\boldsymbol{\sigma}(\cdot)\|$ for all $\boldsymbol{x} \in \mathbb{X}$.

Example 4.1. *Consider a one dimensional example* $n = 1$ *on* $\mathbb{X} = [-2; 2]$ *and a SE kernel with* $\ell^2 = 0.3$ *and* $\zeta^2 = 1$*. The dataset contains only the known equilibrium* $\mathbb{D} = \{(0, 0)\}$*. Then, the quadratic Lyapunov function* $V_{sq}^{(A)} = x^2$ *satisfies the constraint* (4.4)*, while* $V_{sq}^{(B)} = \frac{1}{2} x^2$ *does not. The theoretical derivation is analogous to the proof of Theorem 3.2, a graphical illustration is provided in Fig. 4.2.*

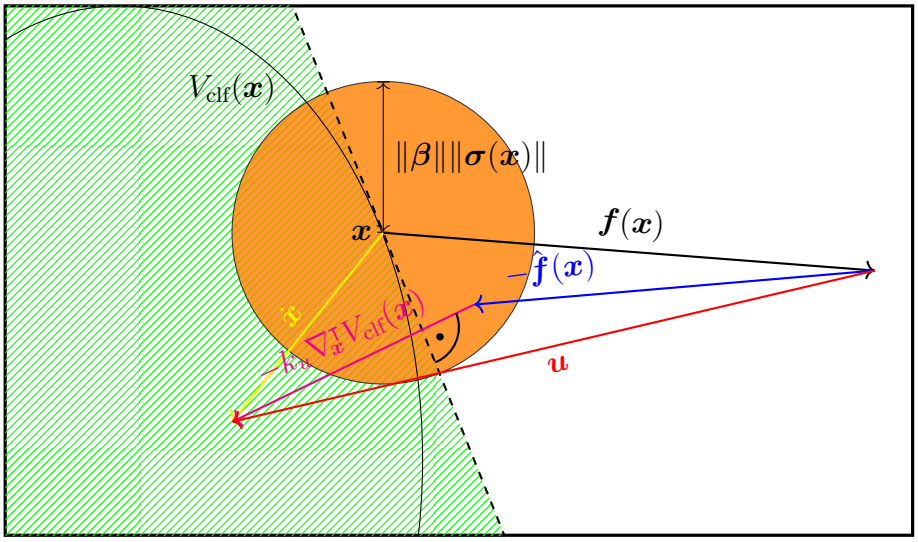

Figure 4.1: Illustration of the proof for Theorem 4.1: For asymptotic convergence, the Lyapunov function (black ellipse shows a level line) must be decreasing along the system's trajectory. Thus, $\dot{\boldsymbol{x}}$ (yellow) must point in the left half-plane of the dashed tangent line (marked with green stripes). From Proposition 2.1 it is known that $\boldsymbol{f}(\cdot) - \hat{\boldsymbol{f}}(\cdot)$ lies somewhere in the orange circle. Choosing a large enough term for $-k_u\nabla_{\boldsymbol{x}}^{\mathsf{T}}V_{\text{clf}}(\boldsymbol{x})$ (magenta), which is always perpendicular to the tangent line, will guarantee the asymptotic convergence.

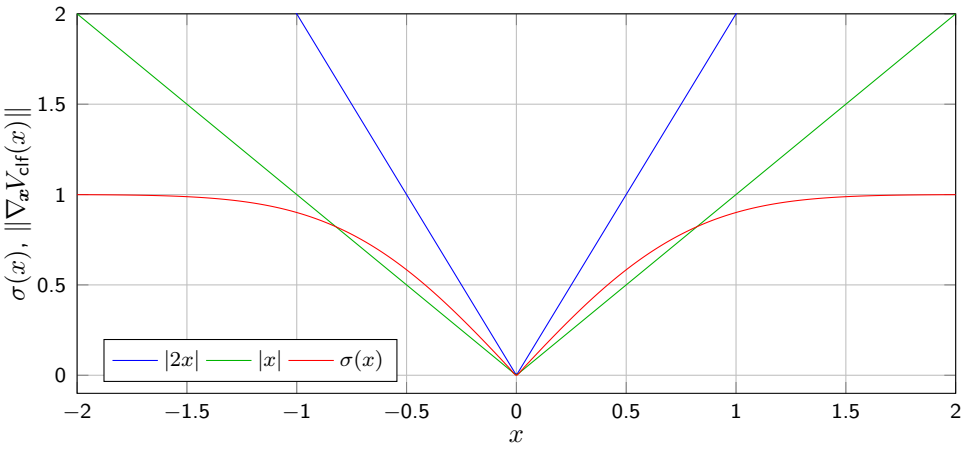

Figure 4.2: Illustration for Example 4.1 and the condition (4.4) in Theorem 4.1. For $\nabla_{\boldsymbol{x}}V_{\text{sq}}^{(A)} = 2x$ (blue) the robustification in the control law (4.3) dominates the uncertainty (red) for all $x \in \mathbb{X}$, for $\nabla_{\boldsymbol{x}}V_{\text{sq}}^{(B)} = x$ (green) it does not, which is therefore not suitable.

4.2.2 Optimality under power limitations

The conditions for stability derived in Theorem 4.1 leave large freedom in the control design because there exists are variety of control Lyapunov functions which satisfy the requirements. In Example 4.1 the quadratic Lyapunov function with properly chosen parameters meets these requirements, but leads to unnecessary large control signals, in particular for $|x| > 1$ (compare Fig. 4.2).

Considering a power constraint in the control input

$$\|\boldsymbol{u}(\boldsymbol{x})\| \leq u_{\max}, \qquad \boldsymbol{x} \in \mathbb{X} \tag{4.5}$$

the actuators can quickly saturate. Furthermore, the stability holds according to Theorem 4.1 only with a certain probability. It therefore desirable to make use of this freedom in the design of $V_{\text{clf}}(\cdot)$ to optimize this *reliability* $\upsilon \in (0;1)$, defined as $\upsilon \coloneqq 1 - \sum_{j=1}^{n} \delta_j$.

We introduce the notation $\tilde{\beta}(\upsilon) \coloneqq \left\| \begin{bmatrix} \beta_1(\delta_1) & \cdots & \beta_n(\delta_n) \end{bmatrix}^\intercal \right\|$ and its corresponding inverse $\upsilon = \mathfrak{b}(\beta_1, \ldots, \beta_n)$, which exists under the assumption $\delta = \delta_1 = \delta_2 = \cdots = \delta_n$ due to the monotonicity of $\beta_j(\delta_j)$, for $j = 1, \ldots, n$. Furthermore, we define

$$\bar{\sigma} \coloneqq \max_{\boldsymbol{x} \in \mathbb{X}} \|\boldsymbol{\sigma}(\boldsymbol{x})\|, \qquad \bar{\mu} \coloneqq \max_{\boldsymbol{x} \in \mathbb{X}} \left\| \hat{\boldsymbol{f}}(\boldsymbol{x}) \right\|, \tag{4.6}$$

which exist due to the boundedness of the SE kernel. The smallest singular value of the matrix $\boldsymbol{G}(\boldsymbol{x})$, $\boldsymbol{x} \in \mathbb{X}$ is denoted by $\underline{\varrho}_{\boldsymbol{G}} \in \mathbb{R}_{+,0}$.

This allows to find necessary conditions for the optimality of $V_{\text{clf}}(\cdot)$ and k_u to maximize the reliability υ under the power constraint (4.5).

Theorem 4.2. *Assume $u_{max} > \bar{\mu}/\underline{\varrho}_{\boldsymbol{G}}$ and consider the upper bound for the control input*

$$\|\boldsymbol{u}(\boldsymbol{x})\| \leq (\bar{\mu} + k_u \|\nabla_{\boldsymbol{x}}^{\mathrm{I}} V_{clf}(\boldsymbol{x})\|)/\underline{\varrho}_{\boldsymbol{G}} \leq u_{max}, \quad \boldsymbol{x} \in \mathbb{X}. \tag{4.7}$$

Then, the optimal reliability

$$\upsilon^* = \underset{\|\nabla_{\boldsymbol{x}} V_{clf}(\boldsymbol{x})\|, k_u}{\arg\max} \ \upsilon \tag{4.8a}$$

$$\begin{aligned} s.t.\ & (\bar{\mu} + k_u \|\nabla_{\boldsymbol{x}} V_{clf}(\boldsymbol{x})\|)/\underline{\varrho}_{\boldsymbol{G}} \leq u_{max},\ k_u > \tilde{\beta}(\upsilon), \\ & \|\nabla_{\boldsymbol{x}} V_{clf}(\boldsymbol{x})\| \geq \|\boldsymbol{\sigma}(\boldsymbol{x})\|, \quad \boldsymbol{x} \in \mathbb{X} \end{aligned} \tag{4.8b}$$

is attained for

$$\|\nabla_{\boldsymbol{x}} V_{clf}(\boldsymbol{x})\| = \|\boldsymbol{\sigma}(\boldsymbol{x})\|, \qquad k_u = (u_{max}\underline{\varrho}_{\boldsymbol{G}} - \bar{\mu})/\bar{\sigma} + \epsilon. \tag{4.9}$$

for any arbitrarily small ϵ with $(u_{max}\underline{\varrho}_{\boldsymbol{G}} - \bar{\mu})/\bar{\sigma} > \epsilon > 0$. The maximum reliability is

$$\upsilon^* = \mathfrak{b}\left((u_{max}\underline{\varrho}_{\boldsymbol{G}} - \bar{\mu})/\bar{\sigma} - \epsilon\right). \tag{4.10}$$

Proof. The function $\tilde{\beta}(\upsilon)$ is monotone according to its definition in Proposition 2.1. This allows to reformulate the optimization as a maximization of $\tilde{\beta}$. The objective is then linear and the constraint set is convex. Thus, the maximum is attained on the boundary. As k_u must not depend on $\boldsymbol{x}$, but (4.8b) must hold $\forall \boldsymbol{x} \in \mathbb{X}$, the lower bound $(u_{\max}\underline{\varrho}_{\boldsymbol{G}} - \bar{\mu})/\bar{\sigma}$ is taken for k_u. □

Remark 4.4. *Theorem 4.2 utilizes the upper bound* (4.7) *for the control input as constraint and does not maximize the reliability directly under* (4.5)*. Therefore, the solution* (4.10) *lower bounds the actually achievable reliability, since* (4.7) *is more restrictive than* (4.5)*. For the remainder of this chapter, we mainly utilize the condition for optimality* $\|\nabla_{\boldsymbol{x}} V_{clf}(\boldsymbol{x})\| = \|\boldsymbol{\sigma}(\boldsymbol{x})\|$ *which results from* (4.7)*.*

Theorem 4.2 provides insights on how the freedom in design of the control law (4.3) can be utilized to maximize the reliability. The condition $\|\nabla_{\boldsymbol{x}} V_{\mathrm{clf}}(\boldsymbol{x})\| = \|\boldsymbol{\sigma}(\boldsymbol{x})\|$ clearly indicates that the uncertainty in the model plays a crucial role in how the control Lyapunov function $V_{\mathrm{clf}}(\cdot)$ is chosen optimally, which will be discussed in more detail in the following section.

4.2.3 Uncertainty-based control Lyapunov function

The optimality condition on the Lyapunov function $V_{\mathrm{clf}}(\cdot)$ is a differential equation

$$\begin{aligned} \|\boldsymbol{\sigma}(\boldsymbol{x})\| = \|\nabla_{\boldsymbol{x}} V(\boldsymbol{x})\|, \qquad & \boldsymbol{x} \in \mathbb{X} \\ \text{with } V(\mathbf{0}) = 0, & \end{aligned} \tag{4.11}$$

where the boundary condition is obtained from Theorem 4.1. It cannot be solved analytically, however, it is a well-known problem: It takes the same form as a necessary condition for the optimal value function as derived from the Hamilton-Jacobi-Bellman (HJB) equation [155]. In a continuous path planning problem, the value function assigns to each initial state the minimum accumulated cost to reach the goal state. The following theorem exploits this relation.

Theorem 4.3. *Consider the value function* $V_{val} \colon \mathbb{X} \to \mathbb{R}_{+,0}$ *defined as*

$$V_{val}(\boldsymbol{x}_0) = \min_{\tilde{\boldsymbol{x}}(\mathfrak{s})} \int_0^{\mathfrak{S}} c_\sigma\left(\tilde{\boldsymbol{x}}(\mathfrak{s})\right) d\mathfrak{s} \qquad \text{s.t. } \tilde{\boldsymbol{x}}(0) = \boldsymbol{x}_0, \quad \tilde{\boldsymbol{x}}(\mathfrak{S}) = \mathbf{0}, \tag{4.12}$$

representing the accumulated cost along an optimal path $\tilde{\boldsymbol{x}}(\mathfrak{s})$ *from the current state* $\boldsymbol{x}_0$ *to the origin, which is parameterized by the arclength* $s \in [0; \mathfrak{S}]$*,* $\mathfrak{S} \in \mathbb{R}_+$ *for which holds* $\|\frac{d\tilde{\boldsymbol{x}}}{ds}\| = 1$*. For a stage cost* $c_\sigma : \mathbb{X} \to \mathbb{R}_{+,0}$*, given by*

$$c_\sigma(\boldsymbol{x}) = \|\boldsymbol{\sigma}(\boldsymbol{x})\|, \tag{4.13}$$

the value function $V_{val}(\cdot)$ *is a control Lyapunov function* $V_{clf}(\cdot)$ *which solves the optimization* (4.8)*.*

Proof. According to the HJB equation, the value function must satisfy the condition

$$\|\nabla_{\boldsymbol{x}} V_{\mathrm{val}}(\boldsymbol{x})\| = c_\sigma\left(\boldsymbol{x}\right), \qquad \boldsymbol{x} \in \mathbb{X} \qquad \text{with boundary condition } V_{\mathrm{val}}(\mathbf{0}) = 0. \tag{4.14}$$

From the choice (4.13), it can be directly seen that the conditions for deriving the value function (4.14) and for maximizing the achievable confidence in (4.11) are equivalent. □

As a result of Theorem 4.3, the optimal control Lyapunov function is denoted by $V_\sigma(\cdot)$ and defined as

$$V_\sigma(\boldsymbol{x}) \coloneqq \min_{\tilde{\boldsymbol{x}}(\mathfrak{s})} \int_0^{\mathfrak{S}} \|\boldsymbol{\sigma}\left(\tilde{\boldsymbol{x}}(\mathfrak{s})\right)\| d\mathfrak{s} \qquad \text{s.t. } \tilde{\boldsymbol{x}}(0) = \boldsymbol{x}, \quad \tilde{\boldsymbol{x}}(\mathfrak{S}) = \mathbf{0}. \tag{4.15}$$

Remark 4.5. *Equation* (4.14) *is also known as the inhomogeneous Eikonal equation [156]. Even though its solution is for most cost functions not differentiable, [157] developed so-called viscosity solutions. These exist for continuous, bounded and positive definite cost functions, which is fulfilled for the choice* (4.13)*.*

Remark 4.6. *It can be concluded that there is an equivalence of the solution of the Eikonal equation* (4.14) *and the optimal control Lyapunov function which maximizes the reliability. From a practical perspective, this is helpful, since it allows to use existing tools from dynamic programming, e.g., fast marching methods [158] for efficient computation. From a theoretical perspective, it verifies an intuitive understanding: By choosing the trajectory which minimizes the uncertainty along the path, the reliability, the probability for asymptotic stability of the closed-loop system is maximized.*

Note that the control law (4.3) drives the system towards parts of the state space where the uncertainty is low and avoids areas with poor model quality. If the feedback linearization was perfect, the control law would track the optimal path (least accumulated uncertainty) based on the value function descend

$$\dot{\boldsymbol{x}} = -\nabla_{\boldsymbol{x}} V_{\mathrm{val}}(\boldsymbol{x}).$$

As a summary of the results, the following proposition is stated.

Proposition 4.1. *Consider the unknown system* (4.1) *under Assumptions 4.1 to 4.4 where the function* $\boldsymbol{f}(\cdot)$ *is modeled by a GP* (2.7)/(2.8) *with SE kernel* (2.5)*, denoted as* $\hat{\boldsymbol{f}}(\cdot)$ *and the power constraint* (4.7)*. Further consider the control law* (4.3) *where the gain is* $k^c = (u_{max}\underline{\varrho}_{\boldsymbol{G}} - \bar{\mu})/\bar{\sigma} + \epsilon$ *and* $V_{clf}(\cdot) = V_\sigma(\cdot)$ *is a viscosity solution of* (4.15)*. Then, the origin of the closed-loop system is semiglobally asymptotically stable with at least probability* v^* *defined in Theorem 4.2.*

Proof. The function $V_\sigma(\cdot)$ is positive definite from the fact that it is defined by a path integral over a positive definite cost function. The condition on $V_\sigma(\cdot)$ in (4.4) is fulfilled with equality and $k^c > \beta(v^*)$ holds from (4.10). Therefore, the stability follows directly from Theorem 4.1 and the maximum confidence from Theorems 4.2 and 4.3. □

4.2.4 Extension to other system classes

While the main contribution of this chapter is derived for the fully actuated system in (4.1), we also provide some insight on how it can be extended to a more general system class. Consider a single-input control affine system in the canonical form

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= x_3 \\
&\cdots \\
\dot{x}_n &= f(\boldsymbol{x}) + g(\boldsymbol{x})u, \quad \boldsymbol{x}_0 = \boldsymbol{x}(0),
\end{aligned} \tag{4.16}$$

Equivalently to Assumptions 4.1 to 4.4 the following assumptions are made

Assumption 4.5. *The function* $f : \mathbb{X} \to \mathbb{R}$ *is unknown but the function value* $f(\boldsymbol{x}_g)$ *is known.*

Again, without loss of generality, we consider $f(\mathbf{0}) = 0$.

Assumption 4.6. *The RKHS norm of $f(\cdot)$ is bounded under the kernel $\mathfrak{k}(\cdot,\cdot)$, $\|f\|_{\mathfrak{k}} < B_f$.*

Assumption 4.7. *The function $g\colon \mathbb{X} \to \mathbb{R}_+$ is known, differentiable and strictly positive.*

Assumption 4.8. *A training set with $N \in \mathbb{N}$ data pairs consisting of the state and a noisy measurement of the highest state's derivative*

$$\mathbb{D} = \left\{ \left(\boldsymbol{x}^{(i)}, y^{(i)} \right) \right\}_{i=1}^{N}, \quad y^{(i)} = f\left(\boldsymbol{x}^{(i)}\right) + \omega^{(i)}, \tag{4.17}$$

is available, where $\omega^{(i)}$ are i.i.d. samples $\omega \sim \mathcal{N}(0, \sigma_{on}^2)$, with $\sigma_{on}^2 \in \mathbb{R}_{+,0}$.

The control law is then formulated as

$$u(\boldsymbol{x}) = -\frac{1}{g(\boldsymbol{x})} \left(\hat{f}(\boldsymbol{x}) + \frac{\nabla_{\boldsymbol{x}_{1:n-1}} V_{\mathrm{clf}}(\boldsymbol{x}) \boldsymbol{x}_{2:n}}{\nabla_{\boldsymbol{x}_n} V_{\mathrm{clf}}(\boldsymbol{x})} + k_u \nabla_{\boldsymbol{x}_n} V_{\mathrm{clf}}(\boldsymbol{x}) \right), \tag{4.18}$$

where $\nabla_{\boldsymbol{x}_{1:n-1}} V_{\mathrm{clf}}(\cdot)$ denotes the gradient of $V_{\mathrm{clf}}(\cdot)$ concatenated from the derivatives with respect to $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-1}$.

Theorem 4.4. *Consider the unknown system (4.16) under Assumptions 4.5 to 4.8 where $f(\cdot)$ is modeled by a GP (2.8) with SE kernel (2.5), denoted as $\hat{f}(\cdot)$. Further consider the control law (4.18) with $k_u > \beta$ and $V_{clf}(\cdot)$ is a positive definite differentiable function with*

$$|\nabla_{\boldsymbol{x}_n} V_{clf}(\mathbf{0})| = 0 \quad \textit{and} \quad \sigma(\boldsymbol{x}) \leq |\nabla_{\boldsymbol{x}_n} V_{clf}(\boldsymbol{x})|, \qquad \boldsymbol{x} \in \mathbb{X}. \tag{4.19}$$

Then, the origin of the closed-loop system is semiglobally asymptotically stable with probability at least $1-\delta$ for all $\boldsymbol{x}_0 \in \mathbb{X}$, where δ and β are defined in Proposition 2.1.

Proof. Choosing $V_{\mathrm{clf}}(\cdot)$ as Lyapunov candidate yields

$$\begin{aligned}
\dot{V}_{\mathrm{clf}}(\boldsymbol{x}) &= \nabla_{\boldsymbol{x}} V_{\mathrm{clf}}(\boldsymbol{x}) \dot{\boldsymbol{x}} \\
&= \nabla_{\boldsymbol{x}_{1:n-1}} V_{\mathrm{clf}}(\boldsymbol{x}) \boldsymbol{x}_{2:n} + \nabla_{\boldsymbol{x}_n} V_{\mathrm{clf}}(\boldsymbol{x}) \left(f(\boldsymbol{x}) - \hat{f}(\boldsymbol{x}) - \frac{\nabla_{\boldsymbol{x}_{1:n-1}} V_{\mathrm{clf}}(\boldsymbol{x}) \boldsymbol{x}_{2:n}}{\nabla_{\boldsymbol{x}_n} V_{\mathrm{clf}}(\boldsymbol{x})} - k_u \nabla_{\boldsymbol{x}_n} V_{\mathrm{clf}}(\boldsymbol{x}) \right) \\
&\leq |\nabla_{\boldsymbol{x}_n} V_{\mathrm{clf}}(\boldsymbol{x})| |f(\boldsymbol{x}) - \hat{f}(\boldsymbol{x})| - k_u |\nabla_{\boldsymbol{x}_n} V_{\mathrm{clf}}(\boldsymbol{x})|^2,
\end{aligned}$$

where the inequality results from the Cauchy-Schwarz inequality. Theorem 2.1 yields

$$\mathcal{P}\left\{ \dot{V}_{\mathrm{clf}}(\boldsymbol{x}) \leq |\nabla_{\boldsymbol{x}_n} V_{\mathrm{clf}}(\boldsymbol{x})| \left(\beta \sigma(\boldsymbol{x}) - k_u |\nabla_{\boldsymbol{x}_n} V_{\mathrm{clf}}(\boldsymbol{x})| \right), \forall \boldsymbol{x} \in \mathbb{X} \right\} \geq 1 - \delta,$$

which results under condition (4.19) and $k_u > \beta$ in

$$\mathcal{P}\left\{ \boldsymbol{x} \in \mathbb{X} \setminus \{\mathbf{0}\},\ \dot{V}_{\mathrm{clf}}(\boldsymbol{x}) < 0 \right\} \geq 1 - \delta.$$

Similarly to Theorem 4.1, $\mathbb{X}$ can be chosen arbitrarily large (however must be compact as required by Theorem 2.1) and therefore the asymptotic stability holds semiglobally. □

The result is analogous to Theorem 4.1, leaving again large freedom for the choice of $V_{\mathrm{clf}}(\cdot)$. We propose - for the same reasons as in Theorem 4.3 - to chose $V_{\mathrm{clf}}(\cdot)$ as the value function of an optimal control problem

$$V_\sigma(\boldsymbol{x}) \coloneqq \min_{\tilde{\boldsymbol{x}}(\mathfrak{s})} \int_0^{\mathfrak{S}} \sigma\left(\tilde{\boldsymbol{x}}(\mathfrak{s})\right) d\mathfrak{s} \quad \text{s.t.} \quad \tilde{\boldsymbol{x}}(0) = \boldsymbol{x},\ \tilde{\boldsymbol{x}}(\mathfrak{S}) = \mathbf{0},\ \frac{\partial \tilde{\boldsymbol{x}}_{1:n-1}}{\partial \mathfrak{s}} = \tilde{\boldsymbol{x}}_{2:n}, \tag{4.20}$$

where, in comparison to (4.15), an additional constraint is imposed to ensure the optimal path $\tilde{\boldsymbol{x}}(\cdot)$ complies with the dynamics (4.16).

We will not analyze this extension further in theory, since the key idea remains the same as for the fully actuated system (4.1).

4.3 Numerical evaluation

Consider the following two dimensional unstable system with two inputs

$$\begin{aligned} \dot{x}_1 &= \underbrace{x_1 + (\cos(x_1) - 1)x_2}_{f_1(\boldsymbol{x})} + u_1, \\ \dot{x}_2 &= \underbrace{-\varsigma(x_1) + x_2}_{f_2(\boldsymbol{x})} + u_2, \end{aligned} \tag{4.21}$$

where $\varsigma(x_1) = \frac{1}{1+\exp(-2x_1)} - 0.5$ is a shifted sigmoid function. The functions $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$ are both continuous and therefore have a finite RKHS norm under a universal kernel according to [104]. Choosing the SE kernel therefore fulfills Assumption 4.2. Assumptions 4.1 and 4.3 are fulfilled as $\boldsymbol{G}(\boldsymbol{x}) = \boldsymbol{I}_2$ and $f_1(\boldsymbol{0}) = f_2(\boldsymbol{0}) = 0$, where the latter is incorporated in the GP regression model according to Remark 2.3.

4.3.1 Setup and implementation

To obtain a set of training data $\mathbb{D}$, the system is initialized twice at the initial locations $\boldsymbol{x}_0 = \begin{bmatrix} 0.3 & 0 \end{bmatrix}^\intercal$ and $\boldsymbol{x}_0 = \begin{bmatrix} -0.3 & 0.1 \end{bmatrix}^\intercal$ without any control, thus $u_1 = u_2 = 0$. During a simulation for $t \in [0; 3]$ measurements are taken in constant time intervals $\Delta t = 0.3$ and with observation noise $\sigma_{\text{on}}^2 = 0.01$ according to Assumption 4.4.

The hyperparameters of the SE kernels as defined in (2.5) are obtained from a likelihood optimization (2.6) using quasi-Newton methods. The maximum input power is set to $u_{\max} = 24$. To compute the reliability, we perform a Monte Carlo experiment and sample 10^4 realizations of the GPs. It is observed that the model error bound imposed by Proposition 2.1 holds with $\upsilon^* = 99.87\%$ on the state space $\mathbb{X} = [-5.5; 5.5]^2$. As a result, we obtain $\|\boldsymbol{\beta}\| = 15$ and set $k^c \approx 17$ with $\epsilon = 2$. This empirical verification is necessary because the information gain γ_j and the RKHS norm B_{f_j}, $j = 1, \ldots, n$ cannot be computed exactly. However, it has the advantage that any sub-optimality of the hyperparameters, which might result from the local optimizer to solve (2.6), is accounted for in the value for $\|\boldsymbol{\beta}\|$.

The most evolved step in the implementation of the proposed control law is the computation of the control Lyapunov function $V_\sigma(\cdot)$ according to its definition (4.15). As an analytic solution to the value function does not exist, a numerical approximation is required, for which we use a grid-based discretization of the state space with 10^4 points. In the resulting graph only neighboring grid points are linked and each link is weighted with the average cost $\boldsymbol{\sigma}(\cdot)$ of the two connected nodes. Dijkstra algorithm is employed to find the shortest path from each node to the origin and a linear interpolation is used to obtain the value function in continuous space. Linear interpolation is chosen because it is almost everywhere differentiable, computationally efficient and preserves the positive definiteness, which would not be guaranteed for more evolved techniques, such as spline interpolation. Note that all these computations are performed offline following the training of the GP model. Thus, the online computation of the control law requires only the gradient evaluation, which is performed using first-order finite differences (with finite difference $\epsilon_{\text{fd}} = 10^{-4}$). To ensure that the numerical inaccuracies do not annul the stability guarantees, the control law online enforces the condition $\|\boldsymbol{\sigma}(\boldsymbol{x})\| < \|\nabla_{\boldsymbol{x}} V_{\text{clf}}(\boldsymbol{x})\|$ in (4.4) by scaling $\nabla_{\boldsymbol{x}} V_{\text{clf}}(\boldsymbol{x})$ accordingly, if necessary. For more evolved methods to approximate the value function in continuous space, we refer to the literature on fast marching methods [158].

All simulations are performed with an explicit Runge-Kutta variable step solver and when $\|\boldsymbol{x}\| < 0.05$, the trajectory is considered as converged[1].

4.3.2 Simulation results

Figure 4.3 visualizes the training data and the trained GPSSM model. The training trajectories indicate the diverging behavior of the uncontrolled system and is also represented in the generalization in the GP mean function $\boldsymbol{\mu}(\cdot)$. The norm of the standard deviation posterior functions of the GPs $\|\boldsymbol{\sigma}(\cdot)\|$, which is utilized as uncertainty measure and cost function, shows to increase with increasing distance to the training data.

The resulting value function, which is here used as control Lyapunov function $V_\sigma(\cdot)$, is shown in Fig. 4.4 along with the resulting trajectories of the controlled system for different initial conditions. It shows the intended behavior that the system is pushed towards the training data, where the model accuracy is high before it drives in this low uncertainty area towards the origin.

4.4 Discussion

To employ the presented approach, the stated assumption must be fulfilled. On the theoretical side, Assumption 4.2 is the most crucial. Even though it is not very restrictive, it is very difficult to verify for real-world systems. Nevertheless, to provide guarantees regarding the real system, some restrictions will always be necessary. In comparison to imposing parametric structural assumptions, we think our approach leaves more flexibility. Difficult to compute is also the maximal information gain γ_j, however previous work in [104] argues that upper bounds for it can be computed in polynomial time. We also refer to an alternative GP bound in [103], which makes less restrictive assumptions and relies on Lipschitz constants, which are easier to obtain.

The theoretical analysis shows that the maximal achievable reliability υ is according to (4.10) a trade-off between the maximum available power $u_{\max}$, the complexity of the system to control measured by B_{f_j}, the number and distribution of the training data and the considered state space $\mathbb{X}$.

One major disadvantage on the practical side is the computational complexity of the approach. The control Lyapunov function cannot be computed analytically and the numerical approximation techniques suffer from the curse of dimensionality. However, as we show equivalence to optimal control and dynamic programming problems, various techniques are applicable to improve the current implementation.

Furthermore, the heaviest computations are all performed offline, which include the Monte Carlo experiment to compute the reliability, the computation of the control Lyapunov function using a grid-based discretization of the state space and the search for the optimal path using the Dijkstra algorithm. These operations are performed for the considered example in MATLAB 2019a on a i5-6200U CPU with 2.3GHz and 8GB RAM in $\approx 30s$. The online phase (using the proposed interpolation) is thereby significantly sped up. Important to note is that the stability is guaranteed irrespectively of any numerical approximation error in the computation of the control Lyapunov function.

[1] Code for MATLAB is provided: `https://gitlab.lrz.de/ga68car/OptUCLF4GP`

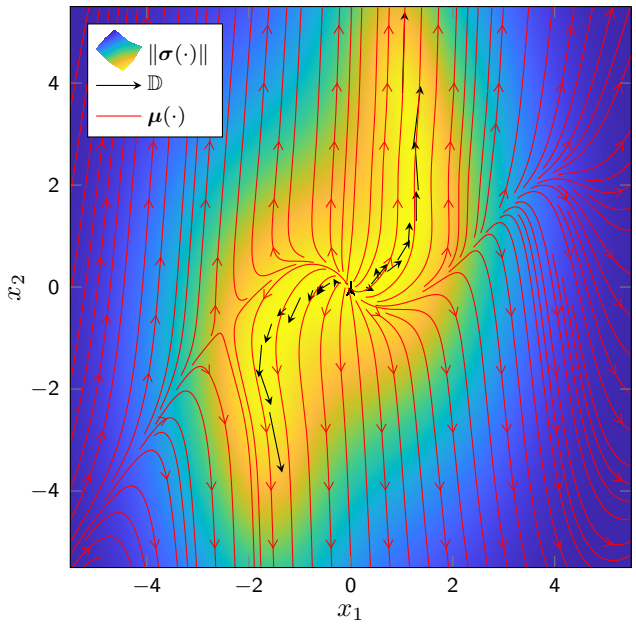

Figure 4.3: The training data $\mathbb{D}$ (black arrows) are obtained from two runs of (4.21) with zero control input $\boldsymbol{u} = 0$. The mean prediction (red streamlines) and the norm of the variance (colormap: yellow is low, blue is high) visualize the GPSSM as defined in (2.7).

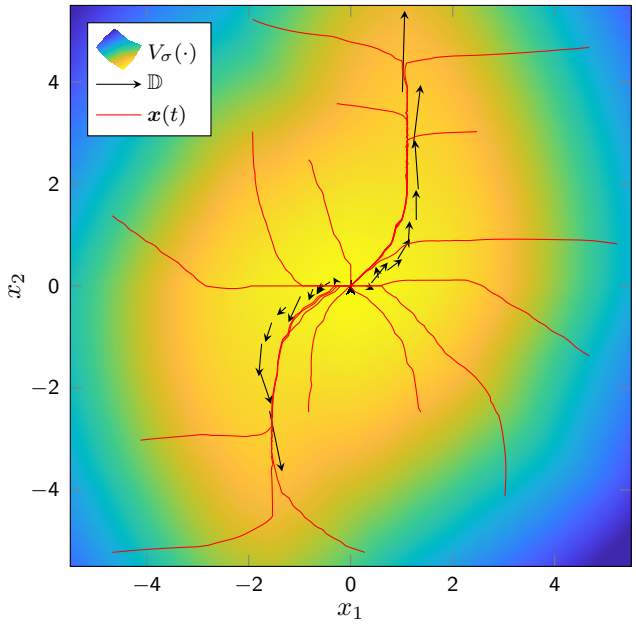

Figure 4.4: The proposed uncertainty-aware Lyapunov function (4.12), is shown as a colormap (yellow is low, blue is high) along with the training data (black arrows). Trajectories (red lines) show the closed-loop behavior under the proposed control law (4.3) for different initial states.

4.5 Summary

This chapter proposes a control law to stabilize arbitrarily complex, unknown, fully actuated systems under mild assumptions with a specified probability. To achieve robustness to model uncertainties, the employed feedback linearization uses a control Lyapunov function. We first derive general conditions for asymptotic stability of the closed-loop system before proposing a specific uncertainty-based control Lyapunov function. We show that under a power constraint on the actuators, it yields the maximum reliability (probability to converge). This reveals an intuitive equivalence: Actively avoiding regions in the state space with low model fidelity yields a controller which maximizes the probability of stability. This is not just interesting from a theoretical perspective, but also comes with practical benefits as it allows to employ computational tools from path planning and dynamic programming.

Feedback Linearization with event-triggered Online Learning

5

The online adaptation of controller or model parameters is a well understood concept in the control literature for linear systems [159]. In particular MRAC, which improves continuously an initially imprecise system model using online parameter estimation, is a popular concept [160]. Closely related is iterative learning control (ILC), which improves control performance for a repetitive task using experience from earlier executions by iteratively modulating the control input [161], [162]. However, most existing MRAC and ILC methods focus on parametric models, which suffer from limited flexibility and are not able to model complex systems.

Considering nonparametric models, the majority of the work (reviewed in Chapter 2) bases their model on one initial data set only. Thus, no training data is collected while the controller is running and there is an inherent reason for that: Classical MRAC (using parametric models) uses a continuous gradient descent in the adaptation of the parameters to minimize the model error over time. However, for nonparametric models, there exist no fixed parameters which can be improved over time. Instead, the data points itself are the parameters and can therefore not be added continuously, but are rather added at discrete time instances. This makes the analysis significantly more complex.

Most control approaches using a GP model with a fixed data set also do not discuss how it can be obtained [73], [84]. Model-based reinforcement learning algorithms collect the data during the interaction with the environment but update the model not until the trail is finished. Thus, new data points obtained during the interaction are not taken into account by the controller [32]. Furthermore, it is unclear weather an initial policy is safe to operate the system [163].

The work in [68] focuses on safe exploration of a state space by sequentially adding new points to the data set. However, the controller is designed to stay only in the region of attraction and cannot track an arbitrary trajectory in the state space. Alternatively, [63] introduces an adaptive control law to guarantee boundedness of the state but it does no quantify the size of the ultimate bound. Additionally, it is based on a time-triggered adaptation, thus data points are added to the data set irrespectively of their necessity. The resulting computational burden for large data sets is a known drawback of GPs [2]. It is commonly solved with heuristic deletion of training points or sparse approximation methods [99], which makes it more difficult to provide safety guarantees. This chapter therefore exploits the idea of *event-triggered* model updates. While event-based control is well researched [164] and frequently applied in networked control system [165] it has not been applied in learning control.

Besides GPs, NNs are also a popular identification tool which are mainly data-driven [18]. However, they have a fixed number of parameters (weights), which can therefore be adapted

continuously over time as in MRAC. This allows [166], [167] and [168] to derive proper stability and performance guarantees. Similarly, the work in [169] proposes a feedback linearizing control law and shows boundedness of the weights and the states in the closed-loop. However, it cannot quantify the ultimate bound because NNs, in contrast to GPs, do not inherently provide a measure for the precision of the model [170].

This chapter's major contribution is a novel nonparametric event-triggered online learning control law, which guarantees stochastic asymptotic stability of initially unknown, control affine systems in the canonical form under very mild assumption. Its key idea is to add training points, whenever the Lyapunov stability condition is in danger to be violated. We also address the challenge of constantly increasing computational complexity due to a growing data set and propose a novel and safe data forgetting strategy for lifelong learning; It uses an information theoretic criterion to keep the most valuable points for tracking the future desired trajectory. The proposed control scheme is illustrated in Fig. 5.1.

The chapter is based on the work published in [10], [11] and [12]. It is structured as follows: After defining the problem setting in Sec. 5.1, this chapter proposes a GP-based closed-loop identification in Sec. 5.2 and the control law in Sec. 5.3 including an analysis of the closed-loop behavior for model updates in arbitrary time instances. Section 5.4 introduces an event-triggered model update for improved convergence and discusses strategies for efficient data-handling. An evaluation in simulation and on a real-world robotic manipulator is provided in Sec. 5.6, followed by a discussion in Sec. 5.7.

5.1 Problem formulation

This chapter considers a single-input control affine system in the canonical form

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ &\cdots \\ \dot{x}_n &= f(\boldsymbol{x}) + g(\boldsymbol{x})u, \quad \boldsymbol{x}_0 = \boldsymbol{x}(0), \end{aligned} \tag{5.1}$$

with state $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^\intercal \in \mathbb{X} = \mathbb{R}^n$, input $u \in \mathbb{U} = \mathbb{R}$ and unknown functions $f \colon \mathbb{X} \to \mathbb{R}$ and $g \colon \mathbb{X} \to \mathbb{R}$. The following assumptions are made.

Assumption 5.1. *The unknown functions $f(\cdot)$ and $g(\cdot)$ are globally bounded and differentiable.*

Assuming the continuity is very natural for most physical systems and also boundedness is often true in practical systems due to saturating effects or physical limits.

Remark 5.1. *The set $\mathbb{X}$ can also alternatively be considered as bounded, which automatically implies (due to the differentiability) boundedness of $f(\cdot)$ and $g(\cdot)$. However, we decided $\mathbb{X}$ to be unbounded and assume the dynamic functions to be bounded.*

With Assumption 5.1, the following is derived.

Lemma 5.1. *Consider the system* (5.1) *under Assumption 5.1 with bounded and continuous input u. Then, the solution $\boldsymbol{x}(t)$ does not have a finite escape time, thus $\nexists t_\infty$, $0 < t_\infty < \infty$ for which*

$$\lim_{t \to t_\infty} \|\boldsymbol{x}(t)\| = \infty.$$

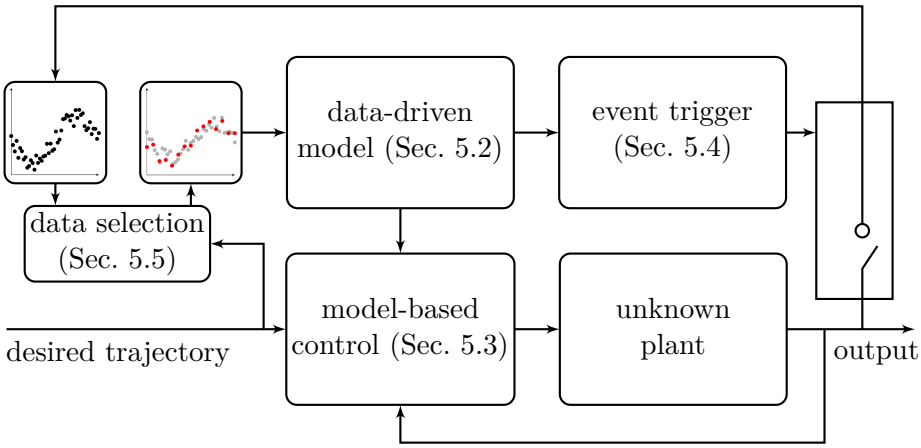

Figure 5.1: Proposed concept of an online learning control law with event-triggered model updates and data selection.

Proof. According to [131, Theorem 3.2] a unique solution $\boldsymbol{x}(t)$, is ensured under the given conditions for all $t > 0$. The finite escape time of this solution is excluded by the differentiability and the boundedness of $f(\cdot)$, $g(\cdot)$ and u since $\|\dot{\boldsymbol{x}}\| < \infty$ for all $\|\boldsymbol{x}\| < \infty$. □

Remark 5.2. *This absence of a finite escape time is important as a stabilizing controller is not known in advance (because $f(\cdot), g(\cdot)$ are unknown). This property allows to take measurements of the system in any finite time interval with a "poor" controller (or the uncontrolled system) without risking damage due to diverging states.*

In addition, the following is assumed.

Assumption 5.2. *The system's relative degree is equal to the system order n for all $\boldsymbol{x} \in \mathbb{X}$.*

Assumption 5.3. *The sign of $g(\cdot)$ is known for at least one $\boldsymbol{x} \in \mathbb{X}$.*

Assumption 5.2 excludes the existence of any internal dynamics and ensures global controllability of the system. It is a restriction of the system class, but covers a broad variety of dynamics for instance the wide-spread class of Lagrangian systems. The Assumption 5.3 is not very strong, as it concerns only a single point in the input space.

With these assumptions, the following lemma can be concluded.

Lemma 5.2. *Consider the system* (5.1) *under Assumptions 5.1 to 5.3. Then, without loss of generality, $g(\cdot)$ is taken as strictly positive, thus*

$$g(\boldsymbol{x}) > 0, \quad \boldsymbol{x} \in \mathbb{X}. \tag{5.2}$$

Proof. For $g(\cdot) = 0$, the relative degree of (5.1) is not well defined which contradicts Assumption 5.2. Thus, $g(\boldsymbol{x}) \neq 0$ holds $\forall \boldsymbol{x} \in \mathbb{X}$. With continuity from Assumption 5.1, a constant sign of $g(\cdot)$ is concluded, which can be set positive without loss of generality. □

To allow for a data-driven identification, measurements of the system must be available as described in the following assumption.

Assumption 5.4. *Noise free measurements of the state vector* $\boldsymbol{x}^{(\kappa)} = \boldsymbol{x}(t_\kappa)$ *and noisy measurements of the highest derivative* $y^{(\kappa)} \coloneqq \dot{x}_n(t_\kappa) + \omega^{(\kappa)}$ *can be taken at any discrete time instance* t_κ *with* $\kappa \in \mathbb{N}_0$*. The observation noise* $\omega \sim \mathcal{N}(0, \sigma_{on}^2)$ *is assumed to be i.i.d. Gaussian distributed. The data set*

$$\mathbb{D}_\kappa = \left\{ \boldsymbol{x}^{(i)}, y^{(i)} \right\}_{i=1}^{N_\kappa}, \tag{5.3}$$

is time-varying and updated at time t_κ*. It remains unchanged until* $t_{\kappa+1}$ *and* $N_\kappa \in \mathbb{N}_0$ *denotes the current number of data points.*

Exact measurements of the state vector is fundamental for feedback linearization control laws and thereby a common assumption. The time-derivative of the state x_n can, for practical applications, be approximated through finite differences, where the approximation error is then considered as part of the measurement noise.

The time-instance t_κ, $t_{\kappa+1}$ are not necessarily chosen equidistant, but only denote the time instances at which updates occur. With each new measurement at time t_κ, the data set $\mathbb{D}_\kappa$ is updated, but stays untouched between t_κ and $t_{\kappa+1}$. Throughout this chapter, we will also consider a limited computational power, which results in an upper bound for N_κ.

Based on these assumptions, the goal is a closed-loop identification of the unknown system, where the approximations $\hat{f}_\kappa \colon \mathbb{X} \to \mathbb{R}$, $\hat{g}_\kappa \colon \mathbb{X} \to \mathbb{R}$ for $f(\cdot)$, $g(\cdot)$, respectively, are updated at times t_κ. Based on these approximations a switching feedback linearizing control law

$$u_\kappa(\boldsymbol{x}) = \frac{1}{\hat{g}_\kappa(\boldsymbol{x})} \left(-\hat{f}_\kappa(\boldsymbol{x}) + \nu(\boldsymbol{x}) \right), \tag{5.4}$$

is proposed, where $\nu \colon \mathbb{X} \to \mathbb{R}$ is the linear control law and the input to the approximately linearized system. As the κ-th update occurs at t_κ, the control law u_κ is applied from t_κ until the next event at $t_{\kappa+1}$, more formally written as

$$u = u_\kappa(\boldsymbol{x}), \quad \forall t \in [t_\kappa; t_{\kappa+1}). \tag{5.5}$$

Consider that the model updates t_κ, $t_{\kappa+1}$ do not necessarily occur periodically. We will first consider arbitrary triggering time-instances and then introduce the updates triggered by the uncertainty of the model estimates (Sec. 5.4).

5.2 Closed-loop identification of control-affine systems

For the identification, we aim to use GP posterior mean functions as approximations $\hat{f}(\cdot)$ and $\hat{g}(\cdot)$ in our model[1]. However, the introduced regression in Sec. 2.1 cannot be applied for two reasons. First, the data points which are available according to Assumption 5.4 measure the sum of $f(\cdot)$ and $g(\cdot)u$, but the individual contribution remains unknown. Second, the positivity of $g(\cdot)$, known from Lemma 5.2, must also be ensured by the approximation $\hat{g}(\cdot)$ to generate bounded control inputs (5.5). The following sections present an extension to the already introduced GP regression, which make it applicable under these constraints.

[1]The subscript κ is dropped in this section for notational convenience.

5.2.1 Expressing structure in kernels

In Chapter 2, we discussed how the kernel of the GP reflects the properties of the resulting function. Besides its smoothness properties, the kernel can determine further characteristics, e.g., periodicity or boundedness [26]. According to [171], it can also be utilized to express prior knowledge regarding the structure of the unknown function, which will be exploited in our identification approach. The basic principle is presented in the following.

Sum of functions

Consider $f_a, f_b : \mathbb{X} \to \mathbb{R}$ both originating from independent GP priors

$$f_a(\boldsymbol{x}) \sim \mathcal{GP}\left(\mathfrak{m}_a(\boldsymbol{x}), \mathfrak{k}_a(\boldsymbol{x}, \boldsymbol{x}')\right), \qquad f_b(\boldsymbol{x}) \sim \mathcal{GP}\left(\mathfrak{m}_b(\boldsymbol{x}), \mathfrak{k}_b(\boldsymbol{x}, \boldsymbol{x}')\right),$$

which add up to $f_{\text{sum}} : \mathbb{X} \to \mathbb{R}$, thus $f_{\text{sum}}(\boldsymbol{x}) = f_a(\boldsymbol{x}) + f_b(\boldsymbol{x})$. Then,

$$f_{\text{sum}}(\boldsymbol{x}) \sim \mathcal{GP}\left(\mathfrak{m}_a(\boldsymbol{x}) + \mathfrak{m}_b(\boldsymbol{x}), \mathfrak{k}_a(\boldsymbol{x}, \boldsymbol{x}') + \mathfrak{k}_b(\boldsymbol{x}, \boldsymbol{x}')\right)$$

is also a GP with prior mean $\mathfrak{m}_a(\boldsymbol{x}) + \mathfrak{m}_b(\boldsymbol{x})$ and kernel $\mathfrak{k}_a(\boldsymbol{x}, \boldsymbol{x}') + \mathfrak{k}_b(\boldsymbol{x}, \boldsymbol{x}')$. Consider $N \in \mathbb{N}$ noisy measurements of the sum of the two function are given

$$y_{\text{sum}}^{(i)} = f_{\text{sum}}\left(\boldsymbol{x}^{(i)}\right) + \omega = f_a\left(\boldsymbol{x}^{(i)}\right) + f_b\left(\boldsymbol{x}^{(i)}\right) + \omega^{(i)},$$

with i.i.d. noise $\omega \sim \mathcal{N}(0, \sigma_{\text{on}}^2)$ and $i = 1, \ldots, N$. Then, the joint distribution of the individual functions and the observations is given by

$$\begin{bmatrix} f_a(\boldsymbol{x}) \\ f_b(\boldsymbol{x}) \\ \boldsymbol{y}_{\text{sum}}^{(1:N)} \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} \mathfrak{k}_a(\boldsymbol{x}, \boldsymbol{x}) & \boldsymbol{0} & \boldsymbol{k}_a(\boldsymbol{x})^\intercal \\ \boldsymbol{0} & \mathfrak{k}_b(\boldsymbol{x}, \boldsymbol{x}) & \boldsymbol{k}_b(\boldsymbol{x})^\intercal \\ \boldsymbol{k}_a(\boldsymbol{x}) & \boldsymbol{k}_b(\boldsymbol{x}) & \boldsymbol{K}_a + \boldsymbol{K}_b + \sigma_{\text{on}}^2 \boldsymbol{I}_N \end{bmatrix}\right),$$

where the prior mean functions are set to zero $\mathfrak{m}_a(\cdot) = \mathfrak{m}_b(\cdot) = 0$ for notational simplicity and $\boldsymbol{k}_a, \boldsymbol{k}_b, \boldsymbol{K}_a, \boldsymbol{K}_b$ are defined according to (2.2) and (2.3), respectively. Conditioning on the observed outputs $\boldsymbol{y}_{\text{sum}}^{(1:N)}$, yields an inference for the output of $f_a(\cdot)$ and $f_b(\cdot)$ as following

$$f_a(\boldsymbol{x}) | \, \boldsymbol{X}, \boldsymbol{y}_{\text{sum}}^{(1:N)} \sim \mathcal{N}\left(\boldsymbol{k}_a(\boldsymbol{x})^\intercal \boldsymbol{K}_{\text{sum}}^{-1} \boldsymbol{y}_{\text{sum}}^{(1:N)}, \mathfrak{k}_a(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}_a(\boldsymbol{x})^\intercal \boldsymbol{K}_{\text{sum}}^{-1} \boldsymbol{k}_a(\boldsymbol{x})\right), \tag{5.6}$$

$$f_b(\boldsymbol{x}) | \, \boldsymbol{X}, \boldsymbol{y}_{\text{sum}}^{(1:N)} \sim \mathcal{N}\left(\boldsymbol{k}_b(\boldsymbol{x})^\intercal \boldsymbol{K}_{\text{sum}}^{-1} \boldsymbol{y}_{\text{sum}}^{(1:N)}, \mathfrak{k}_b(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}_b(\boldsymbol{x})^\intercal \boldsymbol{K}_{\text{sum}}^{-1} \boldsymbol{k}_b(\boldsymbol{x})\right), \tag{5.7}$$

where $\boldsymbol{K}_{\text{sum}} = \boldsymbol{K}_a + \boldsymbol{K}_b + \sigma_{\text{on}}^2 \boldsymbol{I}_N$. Similar to (2.6), the extended hyperparameter vector $\boldsymbol{\psi}_{\text{sum}} = \begin{bmatrix} \boldsymbol{\psi}_a^\intercal & \boldsymbol{\psi}_b^\intercal \end{bmatrix}^\intercal$ is obtained through likelihood optimization. This allows to infer an approximation of the individual functions $f_a(\cdot)$, $f_b(\cdot)$ even though only their sum is measured.

Example 5.1. *Consider the two functions $f_a(x) = \sin(x)$ and $f_b(x) = x^2$ are unknown, but $N = 100$ data points, uniformly randomly distributed in $[-1; 1]$, are measured of $f_{sum}(x) = \sin(x) + x^2$ with an observation noise of $\sigma_{on}^2 = 10^{-2}$. Then, with zero prior mean function and SE kernels $\mathfrak{k}_a(\cdot, \cdot)$ and $\mathfrak{k}_b(\cdot, \cdot)$, the individual functions can be inferred according to* (5.6) *and* (5.7) *as shown in Fig. 5.2*

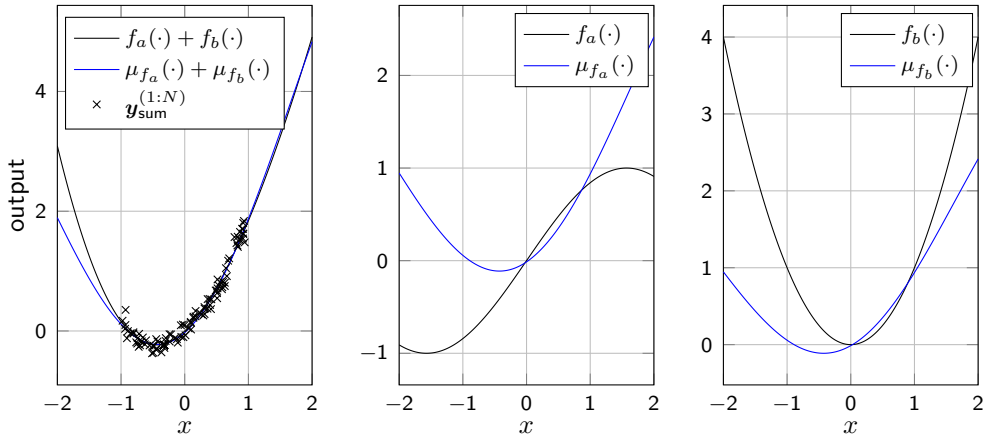

Figure 5.2: Illustration for Example 5.1. The plot on the left shows the noisy training data $\boldsymbol{y}_{\text{sum}}^{(1:N)}$, the true function $f_{\text{sum}}(\cdot)$ and the inferred posterior mean function of the GP. The plot in the middle and the right shows the individual functions $f_a(\cdot)$ and $f_b(\cdot)$ and the individual posterior mean functions $\mu_{f_a}(\cdot)$ and $\mu_{f_b}(\cdot)$. Note that no individual training points are given, but the GPs infer the functions from only observing the sum. A technique for an improved inference of the individual functions (middle and right) will be shown in Example 5.3.

Product with known function

Consider a sample of a GP $f_h : \mathbb{X} \to \mathbb{R}$ which is multiplied with a known function $h : \mathbb{X} \to \mathbb{R}$, thus

$$f_{\text{prod}}(\boldsymbol{x}) = f_h(\boldsymbol{x})h(\boldsymbol{x}) \quad \text{where} \quad f_h \sim \mathcal{GP}(0, \mathfrak{k}_h(\boldsymbol{x}, \boldsymbol{x}'))$$

and $f_{\text{prod}} \colon \mathbb{X} \to \mathbb{R}$. Then, $f_{\text{prod}}(\cdot)$ is also a GP with a scaled kernel function

$$\mathfrak{k}_{\text{prod}}(\boldsymbol{x}, \boldsymbol{x}') = h(\boldsymbol{x})\mathfrak{k}_h(\boldsymbol{x}, \boldsymbol{x}')h(\boldsymbol{x}'),$$

Given the noisy measurements of the product

$$y_{\text{prod}}^{(i)} = f_{\text{prod}}\left(\boldsymbol{x}^{(i)}\right) + \omega^{(i)} = f_h\left(\boldsymbol{x}^{(i)}\right) h\left(\boldsymbol{x}^{(i)}\right) + \omega^{(i)},$$

with $\omega \sim \mathcal{N}(0, \sigma_{\text{on}}^2)$, $i = 1, \ldots, N$, the joint distribution of the measurements and the inferred output of $f_{\text{prod}}(\cdot)$ at a test input $\boldsymbol{x}$ is given by

$$\begin{bmatrix} f_h(\boldsymbol{x}) \\ \boldsymbol{y}_{\text{prod}}^{(1:N)} \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} \mathfrak{k}_h(\boldsymbol{x}, \boldsymbol{x}) & \boldsymbol{k}_h(\boldsymbol{x})^\intercal \boldsymbol{H}^\intercal \\ \boldsymbol{H}\boldsymbol{k}_h(\boldsymbol{x}) & \boldsymbol{H}^\intercal \boldsymbol{K}_h \boldsymbol{H} + \sigma_{\text{on}}^2 \boldsymbol{I}_N \end{bmatrix}\right),$$

where $\boldsymbol{H} = \operatorname{diag}\left(h\left(\boldsymbol{x}^{(1)}\right), \ldots, h\left(\boldsymbol{x}^{(N)}\right)\right) \in \mathbb{R}^{N \times N}$ and $\boldsymbol{k}_h(\cdot)$, $\boldsymbol{K}_h$ are defined analogously to (2.2) and (2.3), respectively. By conditioning on the training data and the input, the function $f_h(\cdot)$ is inferred by

$$f_h(\boldsymbol{x}) | \, \boldsymbol{X}, \boldsymbol{y}_{\text{prod}}^{(1:N)} \sim \mathcal{N}\Big(\boldsymbol{k}_h(\boldsymbol{x})^\intercal \boldsymbol{H}^\intercal \boldsymbol{K}_{\text{prod}}^{-1} \boldsymbol{y}_{\text{prod}}^{(1:N)}, \mathfrak{k}_h(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}_h(\boldsymbol{x})^\intercal \boldsymbol{H}^\intercal \boldsymbol{K}_{\text{prod}}^{-1} \boldsymbol{H}\boldsymbol{k}_h(\boldsymbol{x})\Big), \quad (5.8)$$

where $\boldsymbol{K}_{\text{prod}} = \boldsymbol{H}^\intercal \boldsymbol{K}_h \boldsymbol{H} + \sigma_{\text{on}}^2 \boldsymbol{I}_N$.

Example 5.2. *Consider the unknown function $f_h(x) = \cos(x)$ and the known scaling function $h(x) = \exp(x)$ for which $N = 100$ data points uniformly randomly distributed in $[-1; 1]$ are measured of $f_{prod}(x) = \cos(x)\exp(x)$ with an observation noise of $\sigma_{on}^2 = 10^{-2}$. Then, with zero prior mean function and SE kernel $\mathfrak{k}_h(\cdot,\cdot)$, the unscaled function $f_h(\cdot)$ can be inferred according to (5.8) as shown in Fig. 5.3.*

Remark 5.3. *Instead of scaling the kernel, it seems more straight forward to use $y_{prod}^{(i)}/h\left(\boldsymbol{x}^{(i)}\right)$ as training data for a GP with unscaled kernel. However, this would scale the observation noise undesirably, is numerically not stable and is not compatible with the summation of kernels in the previous section.*

5.2.2 Positivity of Gaussian process posterior mean functions

From Lemma 5.2 it is known that $g(\cdot)$ is strictly positive. It is crucial that the approximation $\hat{g}_\kappa(\cdot)$ also fulfills this property for all κ, as otherwise the feedback linearizing control law (5.4) does not result in well-behaved control signals. For a GP posterior mean function, positivity can be ensured by a suitable choice of the prior mean function as shown in the following.

Lemma 5.3. *Consider the posterior mean function (2.4) with a bounded and differentiable kernel $\mathfrak{k}(\cdot,\cdot)$ and a data set $\mathbb{D} = \left\{\boldsymbol{x}^{(i)}, y^{(i)}\right\}_{i=1}^{N}$ for which $\boldsymbol{x}^{(i)} \neq \boldsymbol{x}^{(i')}$ and $y^{(i)} > 0$ hold $\forall i, i' = 1, \ldots, N$, $i \neq i'$. Then, there exists a differentiable prior mean function $\mathfrak{m}(\cdot)$ such that*

$$\mu(\boldsymbol{x}) > 0, \qquad \forall \boldsymbol{x} \in \mathbb{X}.$$

Proof. A differentiable prior mean function $\mathfrak{m}(\cdot)$ with $0 \leq \mathfrak{m}\left(\boldsymbol{x}^{(i)}\right) < \infty$, $\forall i = 1, \ldots, N$, can always be chosen such that

$$\boldsymbol{k}(\boldsymbol{x})^\intercal (\boldsymbol{K} + \sigma_{\mathrm{on}}^2 \boldsymbol{I}_N)^{-1} \left(\boldsymbol{y}^{(1:N)} - \mathfrak{m}\left(\boldsymbol{x}^{(1:N)}\right)\right) < \mathfrak{m}(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in \mathbb{X} \setminus \left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\right\},$$

holds, because the left side of the inequality is bounded. For $\boldsymbol{x} \in \left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\right\}$, strict positivity of $\mu(\cdot)$ can be ensured by the choice $\mathfrak{m}\left(\boldsymbol{x}^{(i)}\right) = \boldsymbol{y}^{(i)}$. □

Remark 5.4. *Based on Lemma 5.2, the measurements taken from $g(\cdot)$ (if directly available) would violate the assumption $\boldsymbol{y}^{(i)} > 0$ only due to the noise. This can be corrected by using suitable noise models, such as Gamma distributions, which are also compatible with GPs [2]. In most applications, is it sufficient to set $\mathfrak{m}(\cdot)$ to a positive constant and verify the strict positivity of $\mu(\cdot)$ using techniques proposed in [68].*

5.2.3 Closed-loop identification based on Gaussian processes

For the proposed open-loop identification of control affine systems, we will now combine the presented techniques. The ideas from Sec. 5.2.1 are employed to separate the effect of the control input $g(\cdot)$ and of the unforced dynamics $f(\cdot)$ as no individual measurements are available. Section 5.2.2 is used to ensure positivity according to Lemma 5.2.

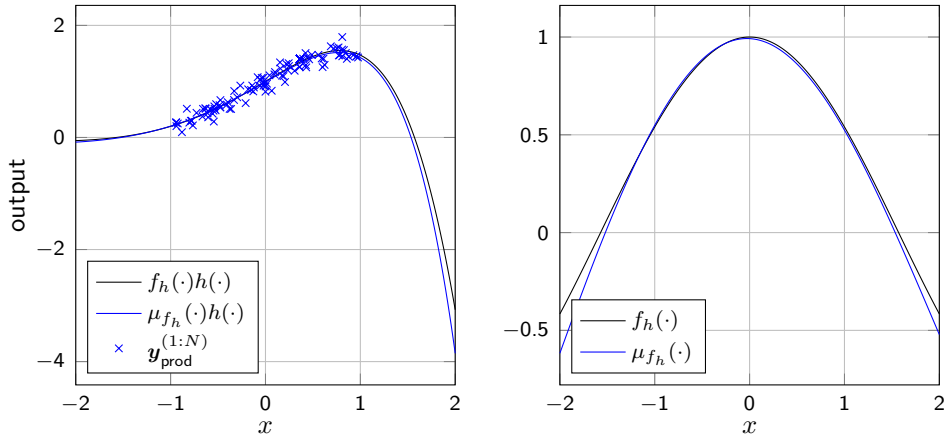

Figure 5.3: Illustration for Example 5.2. The plot on the left shows the noisy training data $\boldsymbol{y}_{\text{prod}}^{(1:N)}$, the true function $f_{\text{prod}}(\cdot)$ and the inferred mean function by the GP. The plot on the right shows the unscaled function $f_h(\cdot)$ and the posterior mean function. The scaling function $h(\cdot)$ is considered to be known.

They key idea is to utilize a composite kernel [171] which replicates the structure of the control affine structure

$$\mathfrak{k}_{fg}(\boldsymbol{x}, \boldsymbol{x}') = \mathfrak{k}_f(\boldsymbol{x}, \boldsymbol{x}') + u(\boldsymbol{x})\mathfrak{k}_g(\boldsymbol{x}, \boldsymbol{x}')u(\boldsymbol{x}'), \tag{5.9}$$

the first summand $\mathfrak{k}_f(\cdot,\cdot)$ stands for the unknown unforced dynamics $f(\cdot)$; the second summand $u(\cdot)\mathfrak{k}_g(\cdot,\cdot)u(\cdot)$ for the product of the unknown scaling of the control $g(\cdot)$ and the known control signal $u(\cdot)$. To ensure smooth approximations (according to Assumption 5.1), we utilize SE kernels with automatic relevance determination

$$\mathfrak{k}_f(\boldsymbol{x}, \boldsymbol{x}') = \zeta_f^2 \exp\left(\sum_{j=1}^{n} \frac{(x_j - x_j')^2}{-2\ell_{j,f}^2}\right), \qquad k_g(\boldsymbol{x}, \boldsymbol{x}') = \zeta_g^2 \exp\left(\sum_{j=1}^{n} \frac{(x_j - x_j')^2}{-2\ell_{j,g}^2}\right), \tag{5.10}$$

with hyperparameters $\ell_{j,f}^2, \ell_{j,g}^2 \in \mathbb{R}_+$ and $\zeta_f^2, \zeta_g^2 \in \mathbb{R}_{+,0}$ for $j = 1, \ldots, n$, which are concatenated in

$$\boldsymbol{\psi}_{fg} = \begin{bmatrix} \ell_{1,f}^2 & \ell_{1,g}^2 & \cdots & \ell_{n,f}^2 & \ell_{n,g}^2 & \zeta_f^2 & \zeta_g^2 \end{bmatrix}^\mathsf{T}.$$

Due to its *universal* property, the SE kernel allows to model any continuous function arbitrarily exact according to [172].

Remark 5.5. *Important here is not to confuse GP models with structured (composite) kernels with parametric models. The latter only has a finite and fixed number of parameters, while the former has possibly infinitely many parameters for each part of its structure. The kernels encodes knowledge of the structure, like sums and products, but each component has unlimited flexibility.*

To formulate the GP regression for the composite kernel (5.9), we define

$$\boldsymbol{U} = \operatorname{diag}\left(u_1\left(\boldsymbol{x}^{(1)}\right), \ldots, u_{N_\kappa}\left(\boldsymbol{x}^{(N_\kappa)}\right)\right) \in \mathbb{R}^{N_\kappa \times N_\kappa},$$

where u_i denotes the control law which was active during the measurement of $\left\{\boldsymbol{x}^{(i)}, y^{(i)}\right\}$ for $i = 1, \ldots, N_\kappa$. Furthermore, $\boldsymbol{k}_f(\cdot), \boldsymbol{k}_g(\cdot), \boldsymbol{K}_f, \boldsymbol{K}_g$ are analogously defined to (2.2) and (2.3), respectively, and

$$\boldsymbol{K}_{fg} = \boldsymbol{K}_f + \boldsymbol{U}^\intercal \boldsymbol{K}_g \boldsymbol{U} + \sigma_{\text{on}}^2 \boldsymbol{I}_n.$$

This allows to formulate the estimation for the control affine dynamics.

Lemma 5.4. *The GP posterior mean prediction for the functions $f(\cdot), g(\cdot)$, based on the training data $\mathbb{D}_\kappa$ with the compound kernel* (5.9) *are given by*

$$\hat{f}(\boldsymbol{x}) := \mu_f(\boldsymbol{x}) = \boldsymbol{k}_f(\boldsymbol{x})^\intercal \boldsymbol{K}_{fg}^{-1} \left(\boldsymbol{y}^{(1:N)} - \boldsymbol{U}\mathfrak{m}_g\left(\boldsymbol{x}^{(1:N)}\right)\right), \tag{5.11}$$

$$\hat{g}(\boldsymbol{x}) := \mu_g(\boldsymbol{x}) = \mathfrak{m}_g(\boldsymbol{x}) + \boldsymbol{k}_g(\boldsymbol{x})^\intercal \boldsymbol{U} \boldsymbol{K}_{fg}^{-1} \left(\boldsymbol{y}^{(1:N)} - \boldsymbol{U}\mathfrak{m}_g\left(\boldsymbol{x}^{(1:N)}\right)\right), \tag{5.12}$$

where the prior mean function for $f(\cdot)$ is set to zero, i.e. $\mathfrak{m}_f(\cdot) = 0$, and for $g(\cdot)$, $\mathfrak{m}_g(\cdot)$ is chosen according to Lemma 5.3.

Proof. The joint distribution for an input $\boldsymbol{x}$ and the kernel (5.9) is given by

$$\begin{bmatrix} f(\boldsymbol{x}) \\ g(\boldsymbol{x}) \\ \boldsymbol{y}^{(1:N)} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ \mathfrak{m}_g(\boldsymbol{x}) \\ \boldsymbol{U}\mathfrak{m}_g\left(\boldsymbol{x}^{(1:N)}\right) \end{bmatrix}, \begin{bmatrix} \mathfrak{k}_f(\boldsymbol{x}, \boldsymbol{x}) & 0 & \boldsymbol{k}_f(\boldsymbol{x})^\intercal \\ 0 & \mathfrak{k}_g(\boldsymbol{x}, \boldsymbol{x}) & \boldsymbol{k}_g(\boldsymbol{x})^\intercal \boldsymbol{U}^\intercal \\ \boldsymbol{k}_f(\boldsymbol{x}) & \boldsymbol{U}\boldsymbol{k}_g(\boldsymbol{x}) & \boldsymbol{K}_{fg} \end{bmatrix}\right), \tag{5.13}$$

similarly to (2.1). Then, the posterior mean functions (5.11) and (5.12) follow along the lines of (2.4) according to [171]. □

It can be shown that all prior knowledge on $f(\cdot)$ and $g(\cdot)$ is transferred to their estimates $\hat{f}_\kappa(\cdot), \hat{g}(\cdot)$.

Proposition 5.1. *Consider a control affine system* (5.1) *under Assumptions 5.1 to 5.4 and the compound kernel* (5.9)*. Then, the estimates $\hat{f}(\cdot)$ and $\hat{g}(\cdot)$ in Lemma 5.4 are bounded, infinitely differentiable and there exists a prior mean function $m_g(\boldsymbol{x})$ and a hyperparameter vector $\boldsymbol{\psi}_{fg}$ such that $\hat{g}(\boldsymbol{x}) > 0$ holds $\forall \boldsymbol{x} \in \mathbb{X}$.*

Proof. The differentiability and boundedness are inherited from the SE kernel to all functions represented by the GP [2]. In consequence, it also holds for the posterior mean functions used as estimates here. The strict positivity of $\hat{g}(\cdot)$ follows from the fact that ζ_g^2 can be chosen arbitrarily small. Thus, there always exists a positive function $\mathfrak{m}_g$, for which $\mathfrak{m}_g(\cdot)$ dominates the term $\boldsymbol{k}_g^\intercal \boldsymbol{U} \boldsymbol{K}_{fg}^{-1} \left(\boldsymbol{y}^{(1:N)} - \boldsymbol{U}\mathfrak{m}_g\left(\boldsymbol{x}^{(1:N)}\right)\right)$ in (5.12). □

Remark 5.6. *This thesis focuses mainly on SE kernel, however the only properties which are used for Proposition 5.1 are its differentiability and its boundedness. Thus, the conclusions can be generalized to different classes of kernels which fulfill these properties.*

The hyperparameters $\boldsymbol{\psi}_{fg}$ are obtained from a likelihood optimization as presented in (2.6).

5.2.4 Improving identification

From Lemma 5.3, it can be concluded that the state remains bounded for any time interval $[0;T]$ with $T < \infty$. As this holds for any bounded control input, it can also be set to zero $u = 0$ without risking damage. This allows to record an open-loop training point

$$y_{\mathrm{ol}}^{(i_{\mathrm{ol}})} = f\left(\boldsymbol{x}^{(i_{\mathrm{ol}})}\right) + \omega^{(i_{\mathrm{ol}})}, \qquad i_{\mathrm{ol}} = 1, \ldots, N_{\mathrm{ol}}, \quad N_{\mathrm{ol}} \in \mathbb{N}.$$

This only measures $f(\cdot)$ (with the noise ω) without any unknown contribution from $g(\cdot)u(\cdot)$. The GP framework allows to merge these N_{ol} observations with the closed-loop training points in $\mathbb{D}_\kappa$ to refine the prediction.

Consider the extension of the joint distribution (5.13), where $u = 1$ in the closed-loop (for notational convenience) and $u = 0$ in the open-loop measurements. For $\mathfrak{m}_g(\boldsymbol{x}) = 0$ (also for notional convenience)

$$\begin{bmatrix} f(\boldsymbol{x}) \\ g(\boldsymbol{x}) \\ \boldsymbol{y}^{(1:N)} \\ \boldsymbol{y}_{\mathrm{ol}}^{(1:N_{\mathrm{ol}})} \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} \mathfrak{k}_f(\boldsymbol{x},\boldsymbol{x}) & 0 & \boldsymbol{k}_f(\boldsymbol{x})^\intercal & \boldsymbol{k}_{f,\mathrm{ol}}(\boldsymbol{x})^\intercal \\ 0 & \mathfrak{k}_g(\boldsymbol{x},\boldsymbol{x}) & \boldsymbol{k}_g(\boldsymbol{x})^\intercal & \boldsymbol{0} \\ \boldsymbol{k}_f(\boldsymbol{x}) & \boldsymbol{k}_g(\boldsymbol{x}) & \boldsymbol{K}_{fg} & \boldsymbol{K}_{\mathrm{ol,cl}}^\intercal \\ \boldsymbol{k}_{f,\mathrm{ol}}(\boldsymbol{x}) & \boldsymbol{0} & \boldsymbol{K}_{\mathrm{ol,cl}} & \boldsymbol{K}_{\mathrm{ol}} \end{bmatrix}\right),$$

where $\boldsymbol{K}_{\mathrm{ol,cl}}$, $\boldsymbol{K}_{\mathrm{ol}}$ are the pairwise evaluation of $\mathfrak{k}_f(\boldsymbol{x}^{(i_{\mathrm{ol}})}, \boldsymbol{x}^{(i)})$, $\mathfrak{k}_f\left(\boldsymbol{x}^{(i_{\mathrm{ol}})}, \boldsymbol{x}^{(i'_{\mathrm{ol}})}\right)$ and $\boldsymbol{k}_{f,\mathrm{ol}}(\boldsymbol{x})$ evaluates $\mathfrak{k}_f\left(\boldsymbol{x}, \boldsymbol{x}^{(i_{\mathrm{ol}})}\right)$ for all $i = 1, \ldots, N$, $i_{\mathrm{ol}}, i'_{\mathrm{ol}} = 1, \ldots, N_{\mathrm{ol}}$. Then, the estimates are given by

$$\hat{f}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{k}_f^\intercal(\boldsymbol{x}) & \boldsymbol{k}_{f,\mathrm{ol}}^\intercal(\boldsymbol{x}) \end{bmatrix} \begin{bmatrix} \boldsymbol{K}_{fg} & \boldsymbol{K}_{\mathrm{ol,cl}}^\intercal \\ \boldsymbol{K}_{\mathrm{ol,cl}} & \boldsymbol{K}_{\mathrm{ol}} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{y}^{(1:N)} \\ \boldsymbol{y}_{\mathrm{ol}}^{(1:N_{\mathrm{ol}})} \end{bmatrix}, \tag{5.14}$$

$$\hat{g}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{k}_g^\intercal(\boldsymbol{x}) & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{K}_{fg} & \boldsymbol{K}_{\mathrm{ol,cl}}^\intercal \\ \boldsymbol{K}_{\mathrm{ol,cl}} & \boldsymbol{K}_{\mathrm{ol}} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{y}^{(1:N)} \\ \boldsymbol{y}_{\mathrm{ol}}^{(1:N_{\mathrm{ol}})} \end{bmatrix}. \tag{5.15}$$

Even though significant improvement of the identification can be expected in practice, we do not further investigate this extension theoretically, since it further complicates the analysis and - to our knowledge - does not provide additional formal guarantees regarding the convergence of the control law.

Example 5.3. *Consider the two functions $f_a(x) = \sin(x)$ and $f_b(x) = x^2$ are unknown, but $N = 100$ data points uniformly randomly distributed in $[-1;1]$ are measured of $f_{sum}(x) = \sin(x) + x^2$ with an observation noise of $\sigma_{on}^2 = 10^{-2}$. In comparison to Example 5.1, with additional $N_{ol} = 50$ measurements $y_{ol}^{(i_{ol})} = f_a(\boldsymbol{x}^{(i_{ol})})$ the individual functions can be inferred much more precisely as shown in Fig. 5.4.*

This concludes the proposed closed-loop identification based on GPs for control affine system. A numerical illustration will be provided along the simulation results for the controller in Sec. 5.6.

5.3 Feedback linearizing control law

This section introduces the feedback linearizing control law, shows ultimate boundedness for arbitrary switching sequences and quantifies the ultimate bound.

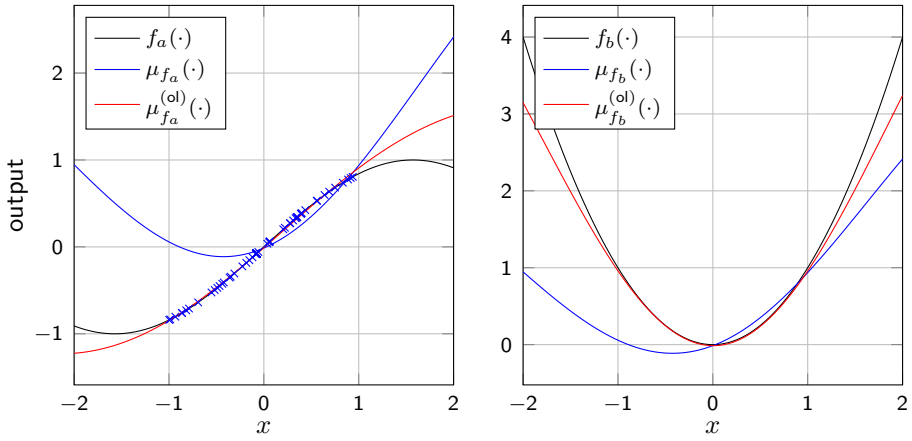

Figure 5.4: Illustration for Example 5.3. The plot on the left shows the noisy additional training data y_{ol} (crosses) and the true function $f_a(\cdot)$ in black. For $f_b(\cdot)$ (in black on the right) no separate training data is provided. Nevertheless, the inferred posterior mean function of the GP with additional training data is much more precise (red) than without (blue).

The goal is to track a desired trajectory for the state x_1, denoted as $x_d(t)$, for which the following is assumed.

Assumption 5.5. *The desired trajectory $x_d(t)$ is bounded and at least $n-1$ times differentiable, thus*

$$\boldsymbol{x}_d(t) = \begin{bmatrix} x_d & \dot{x}_d & \cdots & \frac{d^{n-1}x_d}{dt^{n-1}} \end{bmatrix}^\intercal$$

is continuous and $\frac{d^n x_d}{dt^n}$ is bounded.

The tracking error is defined as

$$\boldsymbol{e} = \boldsymbol{x} - \boldsymbol{x}_d = \begin{bmatrix} e_1 & e_2 & \cdots & e_n \end{bmatrix}^\intercal = \begin{bmatrix} x_1 - x_d & x_2 - \dot{x}_d & \cdots & x_n - \frac{d^{n-1}x_d}{dt^{n-1}} \end{bmatrix}^\intercal$$

For the control design, we aim for asymptotic stability of this tracking error, thus

$$\lim_{t\to\infty} \|\boldsymbol{e}\| = 0.$$

5.3.1 Control law

Consider the filtered state $r \in \mathbb{R}$ defined as

$$r = \begin{bmatrix} \boldsymbol{\lambda}^\intercal & 1 \end{bmatrix} \boldsymbol{e},$$

where $\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 & \lambda_2 & \cdots & \lambda_{n-1} \end{bmatrix}^\intercal \in \mathbb{R}^{n-1}$ is a vector of coefficients for which the polynomial $s^{n-1} + \lambda_{n-1}s^{n-2} + \cdots + \lambda_1$ with $s \in \mathbb{C}$ is Hurwitz. The dynamics of the filtered state are

$$\dot{r} = f(\boldsymbol{x}) + g(\boldsymbol{x})u(\boldsymbol{x}) + \rho(\boldsymbol{x}),$$

where

$$\rho(\boldsymbol{x}) = \boldsymbol{\lambda}^\intercal \boldsymbol{e}_{2:n} - \frac{d^n x_d}{dt^n},$$

with $\boldsymbol{e}_{2:n} = \begin{bmatrix} e_2 & \cdots & e_n \end{bmatrix}^\intercal \in \mathbb{R}^{n-1}$. The feedback linearizing control law (5.4) is then rewritten as

$$u_\kappa(\boldsymbol{x}) = \frac{1}{\hat{g}_\kappa(\boldsymbol{x})} \left(-\hat{f}_\kappa(\boldsymbol{x}) - k^c r - \rho(\boldsymbol{x}) \right), \tag{5.16}$$

where $\nu(\boldsymbol{x}) = -k^c r - \rho(\boldsymbol{x})$ with the control gain $k^c \in \mathbb{R}_+$ is used. The subscript $\kappa \in \mathbb{N}_0$ indicates that $u_\kappa(\cdot)$ is applied in the κ-th time interval $t \in [t_\kappa; t_{\kappa+1})$ according to (5.5). The estimates $\hat{g}_\kappa(\cdot)$, $\hat{f}_\kappa(\cdot)$ are based on the time-varying data set $\mathbb{D}_\kappa$ as introduced in Sec. 5.2.

An overview of the adaptive control scheme is visualized in Fig. 5.5 and presented in algorithmic form in Algorithm 5.1.

Algorithm 5.1 Adaptive feedback linearization control

1: initialize $\kappa = 0$, $\mathbb{D}_0 = \{\}$, $\hat{f}_0(\cdot) = 0$, $\hat{g}_0(\cdot) = \mathfrak{m}_g(\cdot)$
2: **while** simulation time not exceeded **do**
3: **while** $t < t_{\kappa+1}$ **do**
4: run $u_\kappa(\boldsymbol{x})$ in (5.16) to control the system (5.1)
5: **end while**
6: set $\kappa \leftarrow \kappa + 1$
7: measure $\boldsymbol{x}^{(\kappa)} = \boldsymbol{x}(t_\kappa)$ and $y^{(\kappa)} = \dot{x}_n(t_\kappa) + \omega^{(\kappa)}$
8: add training point $\mathbb{D}_\kappa = \mathbb{D}_{\kappa-1} \cup \left\{ \left(\boldsymbol{x}^{(\kappa)}, y^{(\kappa)} \right) \right\}$
9: update the estimates $\hat{f}_\kappa(\cdot), \hat{g}_\kappa(\cdot)$ according to (5.11), (5.12)
10: **end while**

Remark 5.7. *The adaptation of the hyperparameters is not explicitly mentioned in Algorithm 5.1 intentionally. The hyperparameters can either be considered as prior knowledge and therefore remain unchanged, compare Assumption 5.7. Alternatively, a likelihood optimization can be performed with any new data point, which we consider to be part of Line 9 in Algorithm 5.1.*

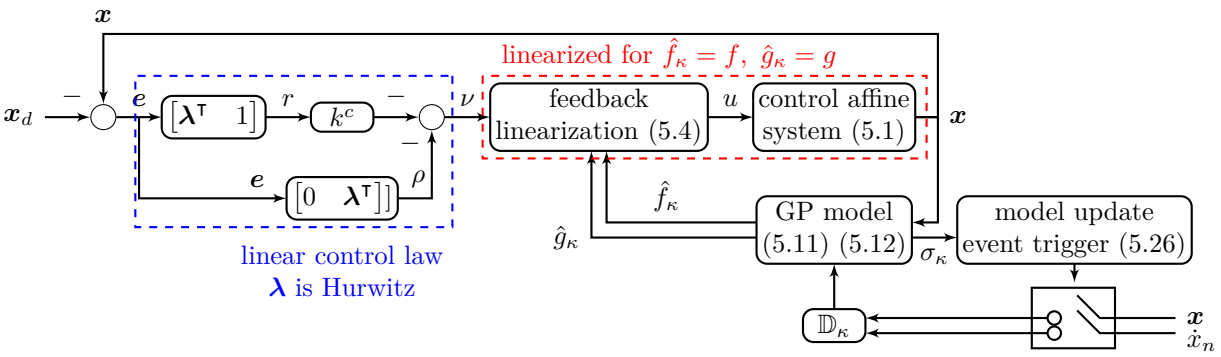

Figure 5.5: The adaptive feedback linearizing control scheme including the event trigger proposed in Sec. 5.4, to control the model update time $t_{\kappa+1}$.

Irrespective of whether Line 9 includes the hyperparameter optimization or not, it is computationally the most demanding step. The complexity of the matrix inverse in (2.4) *can be reduced (from $\mathcal{O}(N^3)$ to $\mathcal{O}(N^2)$) using a rank-1 update with the Sherman–Morrison formula [173]. In any case, we assume an instantaneous execution of this line.*

5.3.2 Convergence analysis

The time varying data set and the changing feedback control law requires an analysis of the closed loop as a switched system. In a first step, an analysis for arbitrary switching is performed, which is based on the principle of a common Lyapunov function.

Theorem 5.1. *Consider the system* (5.1) *and a desired trajectory $\boldsymbol{x}_d(t)$ under Assumptions 5.1 to 5.5. Further consider the control law* (5.16)*, where $\hat{f}_\kappa(\cdot), \hat{g}_\kappa(\cdot)$ are the GP posterior mean functions in* (5.11) *and* (5.12) *to model $f(\cdot)$ and $g(\cdot)$, respectively. The GP model is updated at arbitrary switching times t_κ according to Algorithm 5.1. Then, there exists a $k^{c*} > 0$ such that for every $k^c \geq k^{c*}$ the tracking error $\boldsymbol{e}$ is globally uniformly ultimately bounded (GUUB).*

Proof. Consider the common Lyapunov function $V(\boldsymbol{x}) = r^2/2$ and its time derivative[2]

$$\begin{aligned} \dot{V}(\boldsymbol{x}) = r\dot{r} = r\left(f + g u_\kappa + \rho\right) &= r\left(f + \frac{g}{\hat{g}_\kappa}(-\hat{f}_\kappa - k^c r - \rho) + \rho\right) \\ &= r\left(f - \bar{g}_\kappa \hat{f}_\kappa\right) - k^c \bar{g}_\kappa r^2 + (1 - \bar{g}_\kappa) r\rho, \end{aligned} \tag{5.17}$$

where $\bar{g}_\kappa := \frac{g(\boldsymbol{x})}{\hat{g}_\kappa(\boldsymbol{x})}$ is positive and bounded $\forall \kappa$ and $\boldsymbol{x} \in \mathbb{X}$ from Proposition 5.1 and Assumption 5.1. In consequence, there exist constants $\boldsymbol{c}_1, \boldsymbol{c}_3 \in \mathbb{R}^n$ and $\boldsymbol{C}_2, \boldsymbol{C}_3 \in \mathbb{S}^n_+$ for which holds

$$\left\| r\left(f - \bar{g}_\kappa \hat{f}_\kappa\right) \right\| \leq \|\boldsymbol{c}_1^\intercal \boldsymbol{e}\|, \quad \left\|\bar{g}_\kappa r^2\right\| \geq \boldsymbol{e}^\intercal \boldsymbol{C}_2 \boldsymbol{e}, \quad \|(1 - \bar{g}_\kappa) r\rho\| \leq \boldsymbol{e}^\intercal \boldsymbol{C}_3 \boldsymbol{e} + \boldsymbol{c}_3^\intercal \boldsymbol{e}, \qquad \forall \boldsymbol{e}, \kappa \tag{5.18}$$

since r, ρ depend linearly on $\boldsymbol{e}$ and $f, \hat{f}_\kappa, \bar{g}_\kappa$ are bounded. Therefore,

$$\begin{aligned} \dot{V}(\boldsymbol{x}) &\leq \|\boldsymbol{c}_1\|\|\boldsymbol{e}\| - k^c \underline{\varrho}(\boldsymbol{C}_2)\|\boldsymbol{e}\|^2 + \bar{\varrho}(\boldsymbol{C}_3)\|\boldsymbol{e}\|^2 + \|\boldsymbol{c}_3\|\|\boldsymbol{e}\| \\ &= \|\boldsymbol{e}\|\left(\|\boldsymbol{c}_1\| + \|\boldsymbol{c}_3\|\right) + \|\boldsymbol{e}\|^2\left(\bar{\varrho}(\boldsymbol{C}_3) - k^c \underline{\varrho}(\boldsymbol{C}_2)\right), \qquad \forall \kappa \end{aligned}$$

and there exists a $k^{c*} > 0$ such that

$$\bar{\varrho}(\boldsymbol{C}_3) - k^{c*} \underline{\varrho}(\boldsymbol{C}_2) \leq 0. \tag{5.19}$$

Thus, outside of the set

$$\mathbb{B} = \left\{ \boldsymbol{x} \in \mathbb{X} \,\middle|\, \|\boldsymbol{e}\| \leq \frac{\|\boldsymbol{c}_1\| + \|\boldsymbol{c}_3\|}{k^c \underline{\varrho}(\boldsymbol{C}_1) - \bar{\varrho}(\boldsymbol{C}_2)} \right\}, \tag{5.20}$$

and for every $k^c \geq k^{c*}$, the Lyapunov function decreases over time

$$\dot{V}(\boldsymbol{x}) < 0, \qquad \boldsymbol{x} \in \mathbb{X} \backslash \mathbb{B}.$$

[2]The dependencies on $\boldsymbol{x}$ are partially omitted here and in future proofs for notational convenience.

for all $\kappa \in \mathbb{N}_0$. Thus, there exists a common radially unbounded Lyapunov function $V(\cdot)$, which decreases outside of the set $\mathbb{B}$. According to [174, Theorem 2.1], this allows to conclude that the tracking error converges to $\mathbb{B}$ for arbitrary switching sequences. Since $\mathbb{B}$ is independent of the initial state, global uniform ultimate boundedness holds. □

This shows that the proposed control law bounds the tracking error for a large enough choice of the gain k^c. However, it is i) unclear how large the ultimate bound is and ii) how the critical gain k^{c*} can be determined. To investigate this further, we make the following simplifying assumption.

Assumption 5.6. *A perfect approximation of $g(\cdot)$ is given ($\hat{g}(\boldsymbol{x}) = g(\boldsymbol{x})$, $\boldsymbol{x} \in \mathbb{X}$), allowing direct measurements of $f(\cdot)$,*

$$y_f^{(i)} = f\left(\boldsymbol{x}^{(i)}\right) + \omega^{(i)} = \dot{x}_n^{(i)} - g\left(\boldsymbol{x}^{(i)}\right) u_\kappa\left(\boldsymbol{x}^{(i)}\right) + \omega^{(i)}$$

with $\omega \sim \mathcal{N}(0, \sigma_{on}^2)$ and $i = 1, \ldots, N_\kappa$. The data set becomes

$$\mathbb{D}_\kappa = \left\{\boldsymbol{x}^{(i)}, y_f^{(i)}\right\}_{i=1}^{N_\kappa}.$$

Remark 5.8. *Assumption 5.6 requires more prior knowledge of the system, but this is often available. For example in Lagrangian systems, which is a considerable large class of systems, the generalized inertia matrix can be modeled very accurately using first order principles, which corresponds to the function $g(\cdot)$. Furthermore, it is common for theoretical analysis of control affine systems [175].*

The remaining unknown function $f(\cdot)$ is now obtained from the GP posterior mean

$$\hat{f}(\boldsymbol{x}) := \boldsymbol{k}_f(\boldsymbol{x})^\intercal \boldsymbol{K}_{\text{on}}^{-1} \boldsymbol{y}^{(1:N_\kappa)}, \tag{5.21}$$

where $\boldsymbol{k}_f$ and $\boldsymbol{K}_f$ are computed according to (2.2) and (2.3) for the SE kernel. Based on the additional prior knowledge obtained from Assumption 5.6, the result in Theorem 5.1 can be improved as follows.

Corollary 5.1. *Consider the system (5.1) and a desired trajectory $\boldsymbol{x}_d(t)$ under Assumptions 5.1 to 5.6. Further consider the control law (5.16), where $\hat{f}_\kappa(\cdot)$ is the GP posterior mean function in (5.21), which is adapted at arbitrary switching times t_κ according to Algorithm 5.1. Then, the tracking error $\|\boldsymbol{e}\|$ of the closed-loop switching system is GUUB for any $k^c > 0$.*

Proof. The Lyapunov function $V(\boldsymbol{x}) = r^2/2$ and $\bar{g}_\kappa = 1$ yield the time derivative

$$\dot{V}(\boldsymbol{x}) = r\dot{r} = r\left(f + g u_\kappa + \rho\right) = r\left(f - \hat{f}_\kappa\right) - k^c r^2, \tag{5.22}$$

which is negative definite if $\left(f - \hat{f}_\kappa\right) < k^c r$ holds. Thus independent of the gain k^c, their exists a bounded set outside of which the Lyapunov function is decreasing $\forall \kappa$. This yields global uniform ultimate boundedness for all $k^c > 0$. □

For completeness, this convergence property is also formulated for a constant data set $\mathbb{D}$, thus no further measurements are taken and no data points are deleted from the set for a $t' > t_\kappa$.

Corollary 5.2. *Consider the system* (5.1) *and a desired trajectory* $\boldsymbol{x}_d(t)$ *under Assumptions 5.1 to 5.6. Further consider the control law* (5.16)*, where* $\hat{f}_\kappa(\cdot)$ *is the GP posterior mean function in* (5.21) *with a fixed data set* $\mathbb{D}$*. Then, the tracking error* $\|\boldsymbol{e}\|$ *of the closed-loop system is GUUB for any* $k^c > 0$*.*

Proof. The proof follows directly from the proof of Corollary 5.1. □

Assumption 5.6 allows to conclude the boundedness for all choices of the control gain k^c, however, the size of the ultimate bound remains unknown. This will investigated in the following section.

5.3.3 Quantifying the ultimate bound

The ultimate bound for the tracking error $\boldsymbol{e}$, which was shown to exist in Theorem 5.1 and Corollaries 5.1 and 5.2 will be quantified in this section. This requires an upper bound for the model error, which is defined as

$$\Delta f_\kappa(\boldsymbol{x}) = |f(\boldsymbol{x}) - \hat{f}_\kappa(\boldsymbol{x})|, \quad \forall \kappa. \tag{5.23}$$

For the GP posterior estimate $\hat{f}_\kappa(\cdot)$, this error can be bounded, based on Theorem 2.1, by the posterior standard deviation function $\sigma(\cdot)$ which is given according to (2.4) as

$$\sigma(\boldsymbol{x}) = \sqrt{\mathfrak{k}_f(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}_f^\intercal(\boldsymbol{x}) \boldsymbol{K}_{\text{on}}^{-1} \boldsymbol{k}_f(\boldsymbol{x})}. \tag{5.24}$$

To apply the results from Theorem 2.1, the assumption on the bounded RKHS norm is required.

Assumption 5.7. *The function* $f(\cdot)$ *has a bounded RKHS norm with respect to the employed kernel* $\mathfrak{k}_f(\cdot, \cdot)$ *with known hyperparameters denoted by* $\|f(\cdot)\|_{\mathfrak{k}_f}^2 \leq B_f$*.*

Remark 5.9. *The state space* $\mathbb{X}$ *here is unbounded and thereby not compact, which is a condition for the application of Theorem 2.1. It can therefore not directly be applied to the entire space* $\mathbb{X}$*. However, from previous analysis in Theorem 5.1 and Corollary 5.1, it is known that the state converges to a compact set, generally denoted by* $\tilde{\mathbb{X}} \subset \mathbb{X}$*. We consider therefore further in the analysis a compact state space* $\bar{\mathbb{X}}$ *which is chosen large enough to include* $\mathbb{B}$ *as defined in* (5.20)*.*

Remark 5.10. *The probability* δ*, with which the error bound in Theorem 2.1 holds, does not refer to individual* N_κ *but to all* $N_\kappa \in \mathbb{N}_0$*. To emphasize this,* (2.10) *can be rewritten as*

$$\mathcal{P}\left\{ \bigcap_{N_\kappa=0}^{\infty} |\mu_\kappa(\boldsymbol{x}) - f(\boldsymbol{x})| \leq \beta_\kappa \sigma_\kappa(\boldsymbol{x}), \forall \boldsymbol{x} \in \bar{\tilde{\mathbb{X}}} \right\} \geq 1 - \delta.$$

where $\tilde{\mathbb{X}} \subset \mathbb{X}$ *is a compact set subset of the state space.*

Based on this additional assumption, the ultimate bound can be determined as follows.

Theorem 5.2. *Consider the system* (5.1) *and a desired trajectory* $\boldsymbol{x}_d(t)$ *under Assumptions 5.1 to 5.7. Further consider the control law* (5.16)*, where* $\hat{f}_\kappa(\cdot)$ *is the GP posterior mean function in* (5.21)*, which is adapted at arbitrary switching times* t_κ *according to Algorithm 5.1. Then, with probability* $1-\delta$*,* $\delta \in (0,1)$*, the tracking error* $\|\boldsymbol{e}\|$ *is GUUB for any* $k^c > 0$ *with ultimate bound*

$$\mathbb{B}_\kappa = \left\{ \boldsymbol{x} \in \tilde{\mathbb{X}} \middle| \|\boldsymbol{e}\| \leq \frac{\beta_\kappa \bar{\sigma}_\kappa}{k^c \left\| \begin{bmatrix} \boldsymbol{\lambda}^\intercal & 1 \end{bmatrix} \right\|} \right\}, \quad \forall \boldsymbol{x}_0 \in \tilde{\mathbb{X}},$$

where $\bar{\sigma}_\kappa \coloneqq \max_{\boldsymbol{x} \in \tilde{\mathbb{X}}} \sigma_\kappa(\boldsymbol{x})$ *and* β_κ *is defined in Theorem 2.1.*

Proof. For the common Lyapunov candidate $V(\boldsymbol{x}) = r^2/2$, the time derivative is given by

$$\dot{V}(\boldsymbol{x}) = r\left(f(\boldsymbol{x}) - \hat{f}_\kappa(\boldsymbol{x})\right) - k^c r^2 \leq |r| \Delta f_\kappa(\boldsymbol{x}) - k^c r^2. \tag{5.25}$$

Applying Theorem 2.1, it can be concluded that

$$\begin{aligned} &\mathcal{P}\left\{\Delta f_\kappa(\boldsymbol{x}) \leq \beta_\kappa \bar{\sigma}_\kappa, \forall \boldsymbol{x} \in \tilde{\mathbb{X}}, \kappa \in \mathbb{N}_0\right\} \geq 1-\delta \\ \Rightarrow\ &\mathcal{P}\left\{\dot{V}(\boldsymbol{x}) < 0, \forall \boldsymbol{x} \in \tilde{\mathbb{X}} \setminus \mathbb{B}_\kappa, \kappa \in \mathbb{N}_0\right\} \geq 1-\delta, \end{aligned}$$

showing the convergence of r to a ball with radius $\frac{\beta_\kappa \bar{\sigma}_\kappa}{k^c}$ and the ultimate boundedness of the tracking error $\boldsymbol{e}$ to the set $\mathbb{B}_\kappa$ with probability larger then $1-\delta$. The attributes hold uniformly and globally from the fact that $V(\cdot)$ is a common time-independent and radially unbounded Lyapunov function [174]. □

Remark 5.11. *Theorem 5.1 shows the existence of an ultimate bound, which holds across all time intervals and is thereby the maximum of the ultimate bounds in each time step. This becomes clear in the choice of* $\boldsymbol{c}_1, \boldsymbol{c}_3, \boldsymbol{C}_2, \boldsymbol{C}_3$ *in* (5.18)*, which provides bounds for all* κ*. In contrast, Theorem 5.2 is here more specific and provides with* $\mathbb{B}_\kappa$ *a quantitative bound for each time interval* κ *individually. Note that this does not imply that the tracking error* $\boldsymbol{e}$ *converges to the set* $\mathbb{B}_\kappa$ *before the next time step* $t_{\kappa+1}$*. It would converge to this tube (in infinite time) if the control law stopped adapting after the* κ*-th update (as discussed in Corollary 5.2).*

Even though the size of the set $\mathbb{B}_\kappa$ can now be computed, it is not necessarily decreasing with any model update because β_κ is increasing with additional data points and it is unclear whether $\bar{\sigma}_\kappa$ decreases faster. If a constant ultimate bound is desired, the gain k^c can be adopted accordingly as elaborated in the following.

Corollary 5.3. *Consider the system* (5.1) *and a desired trajectory* $\boldsymbol{x}_d(t)$ *under Assumptions 5.1 to 5.7. Further consider the control law* (5.16)*, where* $\hat{f}_\kappa(\cdot)$ *is the GP posterior mean function in* (5.21)*, which is adapted at arbitrary switching times* t_κ *according to Algorithm 5.1. Additionally, the control gain* k^c *is adapted according to*

$$k^c_\kappa = \frac{\beta_\kappa \bar{\sigma}_\kappa}{b_{\mathbb{B}} \left\| \begin{bmatrix} \boldsymbol{\lambda}^\intercal & 1 \end{bmatrix} \right\|},$$

where $b_{\mathbb{B}} > 0$ *is the desired size of the bound, which can be chosen arbitrarily. Then, with probability* $1-\delta$*,* $\delta \in (0,1)$*, the tracking error* $\|\boldsymbol{e}\|$ *is GUUB by*

$$\mathbb{B}_{const} = \left\{ \boldsymbol{x} \in \tilde{\mathbb{X}} \,|\, \|\boldsymbol{e}\| \leq b_{\mathbb{B}} \right\}, \quad \forall \boldsymbol{x}_0 \in \tilde{\mathbb{X}}.$$

Proof. This directly follows from the proof of Theorem 5.2 because it already analyzes a switching control law, where now not just the model, but also the gain switches, which leads to the same conclusion. □

Remark 5.12. *The results in Theorem 5.2 and Corollary 5.3 are powerful because they allow to bound the tracking error to an arbitrarily small bound, by properly choosing k^c. This is usually sufficient for most practical applications. However, choosing large control gains is also considered as disadvantageous because it leads to aggressive feedback control, which amplifies noise and might lead to saturation of the actuators.*

Therefore, the next section aims to improve the control design to ensure tight convergence independent of the control gain k^c.

5.4 Event-triggered model update

Until now, the switching times $t_{\kappa+1}$, at which measurements are taken, were chosen arbitrarily. This section will design the instance of the model update to ensure asymptotic convergence of the closed-loop. Previous work, such as [63] and [62] are based on a periodic model update, thus the model update occurs after a fixed time-interval $\Delta t > 0$, resulting in $t_{\kappa+1} = t_\kappa + \Delta t$. While this might be the most intuitive idea and easy to implement, it has several disadvantages.

First, it is unclear, whether, at the current state $\boldsymbol{x}^{(\kappa)}$, the estimate $\hat{f}(\cdot)$ is sufficiently precise or not. In this context, "sufficiently precise" is not an absolute number but refers to the stability of the closed-loop. For a large model error $\Delta f_\kappa(\cdot)$, the condition of a decreasing Lyapunov function is violated as it can be seen from (5.25). Since the model error is not exactly known, Δt must be chosen very small to ensure the stability condition holds at any time.

Second, over an infinite time-horizon the periodic measurements result in infinitely many data points, which are impossible to store in finite memory but might also be unnecessary. Due to limited computational power and possible real-time constraints, also large finite data sets might be critical to handle. This will be discussed in more detail in Sec. 5.5.

Remark 5.13. *For time-triggered model updates, the choice of Δt results in a trade-off between precision of the trajectory tracking (size of the ultimate bound is determined by $\bar{\sigma}_\kappa$ in Theorem 5.2) and the computational complexity arising from measuring and processing the data.*

To break down this trade-off, the key idea of the proposed event-triggered online learning is to add training data points only if necessary to guarantee stability. Thus, if the uncertainty in the model about the true dynamics becomes too large and (5.25) could be violated, an event is triggered. This intuitive approach will be investigated for noiseless measurements in Sec. 5.4.1 and for noisy measurements in Sec. 5.4.2.

5.4.1 Asymptotic stability for noiseless measurements

First, the noise free measurements are formulated as an assumption.

Assumption 5.8. *The time-derivative of the highest order state $\dot{x}_n$ can be measured noise free, thus $\sigma_{on}^2 = 0$.*

The choice of the GP model as function approximation for $f(\cdot)$ yields the advantage that it provides with its posterior standard deviation function $\sigma(\cdot)$ in (5.24) a bound on the model error according to Theorem 2.1. This is utilized to define the following event

$$t_{\kappa+1} \coloneqq \{t > t_\kappa \,|\beta_\kappa \sigma_\kappa(\boldsymbol{x}) \geq k^c |r|\}, \tag{5.26}$$

where the triggering time $t_{\kappa+1}$ is the first time instance $t > t_\kappa$ when $\beta_\kappa \sigma_\kappa(\cdot)$ becomes larger or equal $k^c|r|$. The inter-event time is defined as

$$\Delta t_\kappa \coloneqq t_{\kappa+1} - t_\kappa. \tag{5.27}$$

Remark 5.14. *At $t = t_\kappa$, after an update of the model has been performed, the error of the model is zero, and so is $\sigma_\kappa(\cdot)$ according to Lemma 2.1. This implies $\sigma_\kappa(\boldsymbol{x}(t_\kappa)) < k^c|r(t_\kappa)|$ for all $r \neq 0$. Since σ_κ and r are continuous signals between two events, the event* (5.26) *will always be triggered at equality, thus $\beta_\kappa \sigma_\kappa(\boldsymbol{x}(t_{\kappa+1})) = k^c|r(t_{\kappa+1})|$.*

Based on the proposed event-trigger in (5.26) for the model update, the following conclusion is drawn.

Theorem 5.3. *Consider the system* (5.1) *and a desired trajectory $\boldsymbol{x}_d(t)$ under Assumptions 5.1 to 5.8. Further consider the control law* (5.16)*, where $\hat{f}_\kappa(\cdot)$ is the GP posterior mean function in* (5.21)*, which is updated according to the event-triggering law* (5.26) *and Algorithm 5.1. Then, the tracking error $\boldsymbol{e}$ is uniformly globally asymptotically stable (UGAS) for any $k^c > 0$ and the inter-event time Δt_κ is lower bounded by a positive constant $\underline{\Delta t} > 0$ with probability $1 - \delta$.*

Proof. Consider the common Lyapunov candidate $V(\boldsymbol{x}) = r^2/2$ and its time derivative

$$\dot{V}(\boldsymbol{x}) \leq |r| \Delta f_\kappa(\boldsymbol{x}) - k^c r^2, \tag{5.28}$$

where $\Delta f_\kappa(\cdot)$ is the model error defined in (5.23). It is zero in the time instance after the noiseless measurements and the corresponding model update $\Delta f_\kappa(\boldsymbol{x}(t_\kappa)) = 0$, which results in $\dot{V}(\boldsymbol{x}(t_\kappa)) = -k^c r^2$. For $t_\kappa < t < t_{\kappa+1}$ the estimation error $\Delta f_\kappa(\boldsymbol{x}(t))$ is generally larger than zero, but $k^c r^2 > r \Delta f_\kappa(\boldsymbol{x})$ will hold with probability $1 - \delta$ for all $t < t_{\kappa+1}$ by design of the triggering condition (5.26) and Theorem 2.1

$$\mathcal{P}\left\{\dot{V}(\boldsymbol{x}) < 0, \forall \boldsymbol{x} \in \tilde{\mathbb{X}}, \kappa \in \mathbb{N}_0\right\} \geq 1 - \delta.$$

Considering that the state converges to a compact set $\tilde{\mathbb{X}} = \mathbb{B}_\kappa$ for any initial condition $\boldsymbol{x}_0 \in \mathbb{X}$ according to Theorem 5.2, the Lyapunov function decreases globally resulting in uniform global asymptotic stability with probability $1 - \delta$.

To exclude any Zeno behavior, the Lipschitz constant $L_\sigma > 0$ is defined, such that $\dot{\sigma}_\kappa \leq L_\sigma \dot{r}$, which exists due to the differentiability of $\sigma_\kappa(\cdot)$ with respect to r. Along the lines of [176], it is derived

$$\begin{aligned}
\frac{d}{dt}\left|\frac{\sigma_\kappa}{r}\right| &= \frac{d}{dt}\frac{\sqrt{\sigma_\kappa^2}}{\sqrt{r^2}} = \frac{(\sigma_\kappa^2)^{\frac{-1}{2}}\sigma_\kappa\dot{\sigma}_\kappa\sqrt{r^2} - \sqrt{\sigma_\kappa^2}r\dot{r}(r^2)^{\frac{-1}{2}}}{r^2} = \frac{\dot{\sigma}_\kappa r - \sigma_\kappa \dot{r}}{r^2} \leq \left|\frac{\dot{\sigma}_\kappa}{r}\right| + \left|\frac{\sigma_\kappa \dot{r}}{r^2}\right| \\
&\leq \left|\frac{L_\sigma(\Delta f_\kappa - k^c r)}{r}\right| + \left|\frac{\sigma_\kappa(\Delta f_\kappa - k^c r)}{r^2}\right| \leq L_\sigma \left|\frac{\Delta f_\kappa}{r}\right| + L_\sigma k^c + \left|\frac{\Delta f_\kappa \sigma_\kappa}{r^2}\right| + k^c \left|\frac{\sigma_\kappa}{r}\right|,
\end{aligned}$$

which yields according to Theorem 2.1

$$\mathcal{P}\left\{\frac{d}{dt}\left|\frac{\sigma_\kappa}{r}\right| \leq L_\sigma\beta_\kappa\left|\frac{\sigma_\kappa}{r}\right| + L_\sigma k^c + \beta_\kappa\left|\frac{\sigma_\kappa}{r}\right|^2 + k^c\left|\frac{\sigma_\kappa}{r}\right|, \forall \boldsymbol{x} \in \tilde{\mathbb{X}}, k \in \mathbb{N}_0\right\} \geq 1-\delta.$$

For $\phi_\kappa \coloneqq \left|\frac{\sigma_\kappa}{r}\right|$, the following differential equation is obtained

$$\dot{\phi_\kappa} = \beta_\kappa\phi_\kappa^2 + \phi_\kappa(L_\sigma\beta_\kappa + k^c) + L_\sigma k^c, \tag{5.29}$$

which yields for the time interval $t \in [t_\kappa; t_{\kappa+1})$ and the initial condition $\phi_\kappa(t_\kappa) = 0$ (obtained from $\sigma_\kappa(\boldsymbol{x}(t_\kappa)) = 0$) the solution

$$\phi_\kappa(t) = \frac{1}{2\beta_\kappa}\left(c_{1,\phi_\kappa}\tan\left(\frac{1}{2}\left((t-t_\kappa)c_{1,\phi_\kappa} \pm c_{2,\phi_\kappa}\right)\right) - L_\sigma\beta_\kappa - k^c\right), \tag{5.30}$$

according to [177], where $c_{1,\phi_\kappa} = \sqrt{4\beta_\kappa L_\sigma k^c - (L_\sigma\beta_\kappa + k^c)^2}$ and $c_{2,\phi_\kappa} = 2\arccos\left(\frac{-c_1}{2\sqrt{\beta_\kappa L_\sigma k^c}}\right)$. By design, the event is triggered at $\phi = k^c/\beta_\kappa$, which yields the lower bound

$$\Delta t_\kappa \geq \left(2\arctan\left((3k^c + L_\sigma\beta_\kappa)/c_{1,\phi_\kappa}\right) + c_{2,\phi_\kappa}\right)/c_{1,\phi_\kappa} \geq (\pi + c_{2,\phi_\kappa})/c_{1,\phi_\kappa} =: \underline{\Delta t},$$

on the inter-event time, using $\arctan(\cdot) < \pi/2$. □

To resolve the probabilistic nature of Theorem 5.3, we can make an additional assumption.

Assumption 5.9. *Measurements of $\boldsymbol{x}$ and $\dot{x}_n$ are continuously available.*

This allows a redefinition of the event-trigger

$$t_{\kappa+1} \coloneqq \left\{t > t_\kappa \,|\Delta f_\kappa(\boldsymbol{x}) \geq k^c|r|\right\}, \tag{5.31}$$

which is used in the following.

Corollary 5.4. *Consider the system* (5.1) *and a desired trajectory $\boldsymbol{x}_d(t)$ under Assumptions 5.1 to 5.9. Further consider the control law* (5.16)*, where $\hat{f}_\kappa(\cdot)$ is the GP posterior mean function in* (5.21)*, which is updated according to the event-triggering law* (5.31) *and Algorithm 5.1. Then, the tracking error $\boldsymbol{e}$ is UGAS for any $k^c > 0$.*

Proof. From the proof of Theorem 5.3 this directly follows. □

Remark 5.15. *In a digital control design, Assumption 5.9 cannot be fulfilled as it would require infinite update rates of the sensor and the processing unites. Corollary 5.4 is therefore rather stated for completeness.*

5.4.2 Ultimate boundedness for noisy measurements

In case the measurements of $\dot{x}_n$ are corrupted by noise (Assumption 5.8 is not applicable), the asymptotic stability derived in Theorem 5.3 cannot be achieved. However, by redefining the event-trigger, an ultimate bound can be derived, which is (for the same control gain k^c) tighter than in Theorem 5.2 . It is proportional to the noise level and thereby consistent with Theorem 5.3 as shown in the following.

Corollary 5.5. *Consider the system* (5.1) *and a desired trajectory* $\boldsymbol{x}_d(t)$ *under Assumptions 5.1 to 5.7. Further consider the control law* (5.16)*, where* $\hat{f}_\kappa(\cdot)$ *is the GP posterior mean function in* (5.21)*, which is updated according to the event-triggering law*

$$t_{\kappa+1} \coloneqq \{t > t_\kappa \,|\beta_\kappa \sigma_\kappa(\boldsymbol{x}) \geq k^c|r| \cap \boldsymbol{e} \notin \mathbb{B}_{\sigma_{on}}\}\,. \tag{5.32}$$

and Algorithm 5.1 where

$$\mathbb{B}_{\sigma_{on}} = \left\{ \boldsymbol{x} \in \tilde{\mathbb{X}} \,\middle|\, \|\boldsymbol{e}\| \leq \frac{\sigma_{on}\beta_\kappa}{k^c \left\| \begin{bmatrix} \boldsymbol{\lambda}^\intercal & 1 \end{bmatrix} \right\|} \right\}.$$

Then, with probability $1-\delta$*, the tracking error* $\boldsymbol{e}$ *is GUUB to the set* $\mathbb{B}_{\sigma_{on}}$ *for any* $k^c > 0$ *and the inter-event time* Δt_κ *is lower bounded by a positive constant* $\underline{\Delta t}' > 0$*, for all* $\kappa \in \mathbb{N}$ *with probability* $1-\delta$*.*

Proof. In contrast to Theorem 5.3, the model update at $t = t_\kappa$ does not lead to a perfect model $\Delta f_\kappa(\boldsymbol{x}(t_\kappa)) = 0$. However, the posterior standard deviation function of the GP (5.24) is at any training point upper bounded by the measurement noise $\sigma_\kappa(\boldsymbol{x}(t_\kappa)) \leq \sigma_{\text{on}}$ according to Lemma 2.1. Considering again $V(\boldsymbol{x}) = r^2/2$ as Lyapunov function and its time derivative

$$\dot{V}(\boldsymbol{x}(t_\kappa)) \leq |r|(\beta_\kappa \sigma_{\text{on}} - k^c|r|),$$

it is obvious that $\dot{V}(\cdot)$ is possibly positive inside $\mathbb{B}_{\sigma_{\text{on}}}$. But outside it is negative definite which shows that the system is GUUB.

To exclude Zeno behavior only $\boldsymbol{e} \notin \mathbb{B}_{\sigma_{\text{on}}}$ must be analyzed, since inside $\mathbb{B}_{\sigma_{\text{on}}}$ no events are triggered. The lower bound on the inter-event time is derived along the lines of Theorem 5.3. Hence, the dynamics of $\phi_\kappa(t)$ as derived in (5.29) are the same for the noisy case, but the initial condition $\phi_\kappa(t_\kappa)$ is now unequal from zero (due to the noise). It can be upper bounded by

$$\phi_\kappa(t_\kappa) < \sqrt{\frac{\sigma_{\text{on}}^2}{1 + \sigma_{\text{on}}^2/\zeta^2}} \Big/ |r| \coloneqq \phi_0.$$

The solution for the zero initial condition in (5.30) is adapted to a nonzero initial condition $\phi_{\kappa,0}$ according to [177] by changing c_2 to

$$c'_{2,\phi_\kappa} = 2\arctan\left(\frac{2\beta_\kappa\phi_0 + L_\sigma\beta_\kappa + k^c}{c_1}\right). \tag{5.33}$$

Accordingly, the lower bound on the inter event time for the noisy case is

$$\Delta t_\kappa \geq (\pi + c'_{2,\phi_\kappa})/c_{1,\phi_\kappa} =: \underline{\Delta t}',$$

which concludes the proof. □

5.5 Efficient data handling

Using a event-triggered model update, we have designed a control law which provides the desired safety guarantees, with a vanishing tracking error for infinite time. Even though new data points are only added to $\mathbb{D}_\kappa$ if required, Algorithm 5.1 eventually keeps accumulating infinitely many data points. Since the GP is a data-driven model, whose parameters are the data-points, the computational complexity to perform a prediction grows with N_κ. This can be crucial, if any real-time requirements are in place or only limited computational power is available. Particularly, if the desired trajectory covers a large area in the state space or when high precision tracking is required, this can cause difficulties.

Remark 5.16. *This thesis explicitly only tackles the challenge of computational limits in the GP predictions $\hat{f}_\kappa(\cdot)$ and $\sigma_\kappa(\cdot)$. Since this operation must theoretically be performed in continuous time (and therefore with very high updates rates in a digital setup) to implement the proposed control law, we see this as the most crucial operation. Hence, we do not consider any further restrictions, e.g., the computational time for a measurement, a model update or finite memory.*

Remark 5.17. *There are different approaches in literature to reduce the computational complexity of GPs. However, many perform an approximation in the prediction [96], [99] and are therefore not applicable here, because we would lose the safety guarantees which were shown based on the standard GP prediction.*

We pursue the most direct approach to reduce the computational complexity of GPs, known as *subset of data*, where a set of active data points $\mathbb{D}_\kappa^{(a)} \subseteq \mathbb{D}_\kappa$ is selected for prediction, while all other points are neglected. Here, we impose a *budget* of $\bar{N}^{(a)} \in \mathbb{N}$ data points, thus we limit the active data set to $|\mathbb{D}_\kappa^{(a)}| = \bar{N}_\kappa^{(a)} \leq \bar{N}^{(a)}$. This implies the following assumption.

Assumption 5.10. *The computational power of the controller allows to perform a prediction of the GP model with $\bar{N}^{(a)}$ data points.*

In contrast to previous work, e.g., in [63], which uses a heuristic forgetting strategy, we present a data selection algorithm, which is optimal with respect to the future desired trajectory and safe as it preserves all guarantees derived in the previous section. We first prove safety for a minimal data set, then present a measure to quantify the value of a data point and finally state the resulting algorithm.

5.5.1 Safe forgetting

We first show that the asymptotic stability in Theorem 5.3 also holds if only the last data point is contained in the active training data set and all others are not utilized.

Corollary 5.6. *Consider the system* (5.1) *and a desired trajectory $\boldsymbol{x}_d(t)$ under Assumptions 5.1 to 5.8 and 5.10. Further consider the control law* (5.16)*, where $\hat{f}_\kappa(\cdot)$ is the GP posterior mean function in* (5.21) *which is based on the reduced data set*

$$\mathbb{D}_\kappa^{(a)} = \{\boldsymbol{x}(t_\kappa), \dot{x}_n(t_\kappa)\},$$

and updated according to the event-triggering law (5.31) *and Algorithm 5.1. Then, with probability $1-\delta$, the tracking error $\boldsymbol{e}$ is globally asymptotically stable for any $k^c > 0$.*

Proof. Since $\sigma_\kappa(\cdot)$ is continuous and zero at each training point, $\sigma_\kappa(\boldsymbol{x}(t_\kappa)) = 0$, it is concluded that there exists a neighborhood of $\boldsymbol{x}(t_\kappa)$ where $\sigma_\kappa(\boldsymbol{x}) < k^c|r|$ holds. Therefore, the same conclusion as in Theorem 5.3 can be drawn. □

Remark 5.18. *Corollary 5.6 shows that any memory restrictions can easily be obeyed and a single data point in the GP model is sufficient for stability. However, the rate at which events are triggered is generally increased and the control performance (convergence rate of the tracking error) might suffer.*

Since the problem setting allows to store more than a single data point, the goal is to use $\bar{N}^{(a)}$ points in the GP prediction to be as precise as possible. So for $|\mathbb{D}_\kappa| \leq \bar{N}^{(a)}$, we set $\mathbb{D}_\kappa^{(a)} = \mathbb{D}_\kappa$ and if $\|\mathbb{D}_\kappa\| > \bar{N}^{(a)}$, we aim to select the $\bar{N}^{(a)}$ most useful points. The following section will discuss how the utility of data points can be measured.

5.5.2 Information value of data points

Consider a set of states $\mathbb{X}_v \subset \mathbb{X}$ at which the output of the unknown function $f\colon \mathbb{X} \to \mathbb{R}$ should be inferred. Given is the set $\mathbb{X}_{\mathbb{D}_\kappa} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(N_\kappa)}\} \subset \mathbb{X}$ for which (noisy) measurements of the output of the function $f(\cdot)$ are available. The goal is to select $\bar{N}_\kappa^{(a)}$ data points from $\mathbb{X}_{\mathbb{D}_\kappa}$, denoted by $\mathbb{X}_{\mathbb{D}_\kappa^{(a)}} \subset \mathbb{X}_{\mathbb{D}_\kappa}$ to maximize the information about the function $f(\cdot)$ at the states $\mathbb{X}_v$.

A well understood measure of information is entropy, describing the uncertainty of random variables [178]. Here, we consider the outputs $f(\boldsymbol{x})$ for $\boldsymbol{x} \in \mathbb{X}_v$ as random variables with joint Gaussian distribution. Using $\boldsymbol{\Sigma}_{\mathbb{X}_v}$ to denote their covariance, the entropy is given by

$$H(\mathbb{X}_v) = 0.5 \log\left(2\pi \det(\boldsymbol{\Sigma}_{\mathbb{X}_v})\right),$$

which is a slight abuse of notation here, since $\mathbb{X}_v$ are not the random variables, but the function values of $f(\cdot)$ at these locations. Continuing with this notation, the quantity of interest, the conditional entropy, is formulated as

$$H(\mathbb{X}_v|\mathbb{X}_a) = H(\mathbb{X}_v, \mathbb{X}_a) - H(\mathbb{X}_a).$$

The optimization to select the most informative data set $\mathbb{D}_\kappa^{(a)} \subset \mathbb{D}_\kappa$ about the function values at the states $\mathbb{X}_v \subset \mathbb{X}$ is then formally written as

$$\mathbb{D}_\kappa^{(a^*)} = \underset{\mathbb{D}_\kappa^{(a)} \subset \mathbb{D}_\kappa, \left|\mathbb{D}_\kappa^{(a)}\right| = \bar{N}^{(a)}}{\arg\min} H(\mathbb{X}_v|\mathbb{X}_{\mathbb{D}_\kappa^{(a)}}). \tag{5.34}$$

5.5.3 Safe and optimal data selection

The conditional entropy criterion selects the optimal training set with respect to a set of states of interest $\mathbb{X}_v$, which was not further specified so far. Previous work uses here a grid over a compact input space to ensure a precise global model [178]. However, in our problem setting, the state space is not bounded and the proposed control law only requires a good model near the desired trajectory.

As a consequence, we choose the set $\mathbb{X}_v$ from the manifold of the future trajectory, thus

$$\mathbb{X}_v \subset \{\boldsymbol{x}_d(t) \in \mathbb{X} \,|\, t \in [t_\kappa; \infty)\} =: \mathbb{X}_d.$$

Example 5.4. *Consider $\Delta t_d \in \mathbb{R}_+$ as the density at which the future trajectory is sampled and $N_d \in \mathbb{N}$ determines how many data points are considered. Then, an example for the choice of $\mathbb{X}_v$ is given by*

$$\mathbb{X}_v = \{\boldsymbol{x}_d(t_\kappa), \boldsymbol{x}_d(t_\kappa + \Delta t_d), \ldots, \boldsymbol{x}_d(t_\kappa + N_d \Delta t_d))\}\,.$$

This allows to optimally chose the active data set $\mathbb{D}_\kappa^{(a)}$ for the tracking in the upcoming time interval $\Delta t_d N_d$. Consider that the time interval can be interrupted by the event-trigger if $t_\kappa + \Delta t_d N_d < t_{\kappa+1}$, which leads to a new active data set.

For most choices of $\mathbb{X}_v$, it is not guaranteed that optimization (5.34) selects the most recent training point $\boldsymbol{x}(t_\kappa)$ for the active set. This possibly leads to a violation of the stability condition, which we therefore make a constraint in the optimization

$$\mathbb{D}_\kappa^{(a^*)} = \underset{\mathbb{D}_\kappa^{(a)} \subset \mathbb{D}_\kappa, \left|\mathbb{D}_\kappa^{(a)}\right| = \bar{N}^{(a)}}{\arg\min} H(\mathbb{X}_v | \mathbb{X}_{\mathbb{D}_\kappa^{(a)}}). \qquad \text{s.t.} \quad \beta_{\mathbb{D}_\kappa^{(a)}} \sigma_{\mathbb{D}_\kappa^{(a)}}(\boldsymbol{x}) < k^c |r|, \tag{5.35}$$

where $\beta_{\mathbb{D}_\kappa^{(a)}}$ and $\sigma_{\mathbb{D}_\kappa^{(a)}}(\cdot)$ are defined as β_κ and $\sigma_\kappa(\cdot)$ using $\mathbb{D}_\kappa^{(a)}$ instead of $\mathbb{D}_\kappa$, respectively.

Remark 5.19. *According to Corollary 5.6, the stability constraint is satisfied if $\boldsymbol{x}(t_\kappa) \in \mathbb{D}_\kappa^{(a)}$. Hence, the constraint set is never empty and therefore the optimization* (5.35) *is always feasible.*

A summary using the choice of $\mathbb{X}_v$ according to Example 5.4 is provided in Algorithm 5.2.

Algorithm 5.2 Event-triggered control under computational constraints.

initialize $\kappa = 0$, $\mathbb{D}_0 = \{\}$, $\hat{f}_0(\cdot) = 0$, $\hat{g}(\cdot) = g(\cdot)$
while simulation time not exceeded **do**
  while $t < t_{\kappa+1}$ **do**
    run $u_\kappa(\boldsymbol{x})$ in (5.16) to control the system (5.1)
  end while
  set $\kappa \leftarrow \kappa + 1$
  measure $\boldsymbol{x}^{(\kappa)} = \boldsymbol{x}(t_\kappa)$ and $y^{(\kappa)} = \dot{x}_n(t_\kappa) + \omega^{(\kappa)}$
  add training point $\mathbb{D}_\kappa = \mathbb{D}_{\kappa-1} \cup \left\{\left(\boldsymbol{x}^{(\kappa)}, y^{(\kappa)}\right)\right\}$
  if $|\mathbb{D}_\kappa| > \bar{N}^{(a)}$ **then**
    set $\mathbb{X}_v = \{\boldsymbol{x}_d(t_\kappa), \boldsymbol{x}_d(t_\kappa + \Delta t_d), \ldots, \boldsymbol{x}_d(t_\kappa + N_d \Delta t_d))\}$
    determine active set $\mathbb{D}_\kappa^{(a^*)}$ according to (5.35)
  else
    set $\mathbb{D}_\kappa^{(a^*)} = \mathbb{D}_\kappa$
  end if
  update $\hat{f}_\kappa(\cdot), \hat{g}_\kappa(\cdot)$ and $\sigma_\kappa(\cdot)$ according to (5.11), (5.12) and (5.24) using $\mathbb{D}_\kappa^{(a^*)}$
end while

Remark 5.20. *From a computational perspective, the combinatorial problem* (5.35) *is NP-hard [179]. Therefore, a greedy alternative is employed, as shown in Algorithm 5.3 which*

enforces the stability constraint by always making the most recent training point $\boldsymbol{x}(t_\kappa)$ *part of the active set.*

Algorithm 5.3 Greedy approximation to minimize the conditional entropy.

1: initialize $\mathbb{X}^*_{\mathbb{D}_\kappa^{(a)}} = \{\boldsymbol{x}^{(t_\kappa)}\}$
2: **for** $\bar{n} = 1, \ldots, \bar{N}^{(a)} - 1$ **do**
3: $\quad \mathbb{X}^*_{\mathbb{D}_\kappa^{(a)}} \leftarrow \mathbb{X}^*_{\mathbb{D}_\kappa^{(a)}} \cup \underset{\boldsymbol{x}^{(i)} \in \mathbb{X}_{\mathbb{D}_\kappa}}{\arg\min}\, H(\mathbb{X}_v | \mathbb{X}^*_{\mathbb{D}_\kappa^{(a)}} \cup \boldsymbol{x}^{(i)})$
4: **end for**

5.6 Numerical evaluation

To illustrate the presented control approach and algorithms, we show simulation results for three different scenarios and present a robotic experiment[3].

5.6.1 Simulation results

For the numerical illustration, the following control affine system is considered

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= \underbrace{1 - \sin(x_1) + \varsigma(x_2)}_{=f(\boldsymbol{x})} + \underbrace{\left(1 + \frac{1}{2}\sin(x_2/2)\right)}_{=g(\boldsymbol{x})} u, \end{aligned} \tag{5.36}$$

where $\varsigma(x_2) = \frac{0.5}{1+\exp(-x_2/10)}$ is the sigmoidal function. It can directly be seen that this modified pendulum system is in line with Assumptions 5.1 and 5.2. Since Assumption 5.7 cannot be directly verified, we learn offline a GP with a high density of training data to represent $f(\cdot)$ and take a sample of this GP as the real system if Assumption 5.7 is applied. For samples of a GP, Assumption 5.7 surely holds. Assumptions 5.3, 5.4, 5.6 and 5.8 can directly be fulfilled by the setup of the simulation.

In total three scenarios are considered in simulation to illustrate different aspects of the proposed control law and an overview of the employed parameter is given in Table 5.1.

Scenario 1: Time-triggered learning

In the first scenario (S1), we focus on the identification scheme as presented in Sec. 5.2. Therefore, both functions, $f(\cdot)$ and $g(\cdot)$ are assumed to be unknown and Assumption 5.6 is not in place. We utilize the adaptive control law with a periodic switching sequence, thus $\Delta t_\kappa = 0.5$ is constant $\forall \kappa$ and only Theorem 5.1 is applicable. The event-based approach is not evaluated in S1 to illustrate the simultaneous identification of $f(\cdot)$ and $g(\cdot)$ in closed-loop. We corrupt the measurements with i.i.d. Gaussian noise and thereby Assumption 5.8 is not in place.

[3]Code for MATLAB is provided: `https://gitlab.lrz.de/ga68car/adaptFeLi4GPs`

k^c	$\boldsymbol{\lambda}$	$\boldsymbol{x}_0$	S1:	$\mathfrak{m}_g(\boldsymbol{x})$	σ^2_{on}	S2&S3:	$r_{\min}$	β	σ^2_{on}	ℓ_f^2	ζ^2	$\bar{N}^{(a)}$(S3)
1	1	$\begin{bmatrix}3 & 2\end{bmatrix}^\intercal$		$= 2, \forall \boldsymbol{x}$	10^{-6}		10^{-5}	7	10^{-14}	5	5	10

Table 5.1: Parameters for the simulation of the online learning approach

For the reference trajectory a smooth jump from $x_1 = 1$ to $x_1 = 0$ at $t = 10$ is chosen

$$x_d(t) = 1 - \frac{1}{1 + \exp(-20(t - 10))}$$

to fulfill Assumption 5.5.

Here, Assumption 5.7 is not considered to hold and therefore the hyperparameters of the SE kernel are unknown. A model update with a new data point is therefore always accompanied by hyperparmeter optimization according to (2.6). The simulation time is set to $t = 20$, which results in a total of $N = 40$ data points. Data selection as proposed in Sec. 5.5 is not considered.

Figure 5.6 compares the system dynamics $f(\cdot)$, $g(\cdot)$ in (5.36) with the corresponding estimates $\hat{f}(\cdot)$, $\hat{g}(\cdot)$ at the end for the simulation, thus based on $N = 40$ data points. The hyperparameters are fitted well, which can be seen as follows: The rate of change of $f(\cdot)$ is higher in $\boldsymbol{x}_1$ direction and this is reflected in $l_{1,f} \ll l_{2,f}$. Equivalently, $l_{1,g} \gg l_{2,g}$ shows that $g(\cdot)$ mainly depends on $\boldsymbol{x}_2$. As a result, the closed-loop identification delivers a satisfactory model precision.

The tracking performance over time is shown in Fig. 5.7 and Fig. 5.8 visualizes the resulting trajectory in the state space. Clearly, after the system has collected sufficiently many points, the state converges closely to the two stationary points $\begin{bmatrix}1 & 0\end{bmatrix}^\intercal$ and $\begin{bmatrix}0 & 0\end{bmatrix}^\intercal$ in the steady state. Since the controller uses periodic updates, unnecessary data points are added to the training set as time passes by. Therefore, the second scenario (S2) operates on an event-triggered scheme which is designed to avoid this problem.

Scenario 2: Event-triggered learning

For Scenario 2 (S2), the event-triggered design in Sec. 5.4 is illustrated, which requires further assumptions: The effect of the control signal on the system $g(\cdot)$ is perfectly known, thus Assumption 5.6 is fulfilled. The observations are taken noise free (Assumption 5.8), however a minimal noise level of $\sigma^2_{\text{on}} = 10^{-16}$ is implemented for numerical stability. Also Assumption 5.7 is put in place resulting in constant hyperparameters (no likelihood optimization) throughout the simulation.

We also set β_κ constant for all κ and refer to the discussion in Sec. 5.7 and previous work [68]. To avoid any numeric difficulties a lower bound on the filtered state is implemented $|r| > r_{\min}$. The desired trajectory is a sinusoidal function with amplitude 1, thus

$$x_d(t) = \sin(t),$$

and the simulation is manually stopped at $t = 30$. Here, no limit on the size of the data set $\mathbb{D}_\kappa$ is imposed.

Figure 5.10 compares the event- and the time-triggered schemes. The rate of convergence is approximately the same until a numerical limit is reached at $\|\boldsymbol{e}\| \approx 10^{-5}$, which is aligned

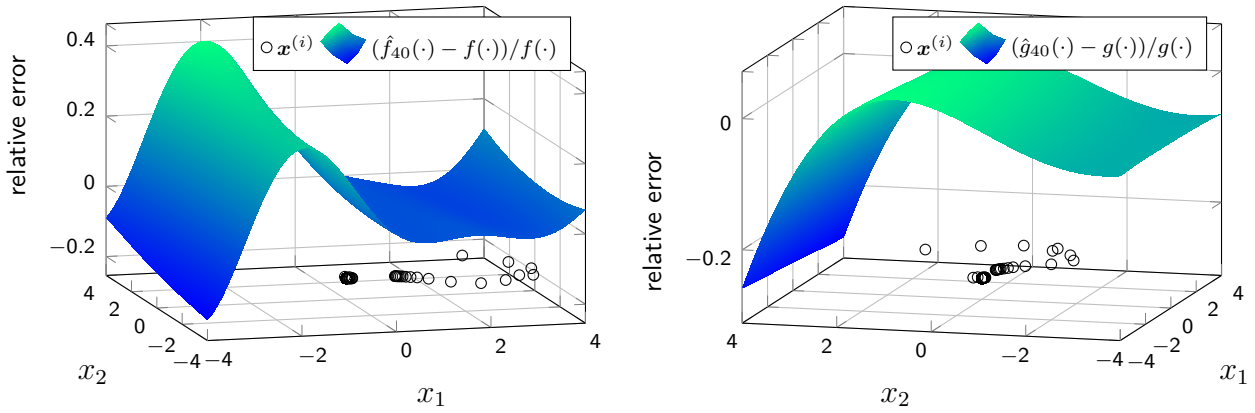

Figure 5.6: Scenario 1: The surfaces illustrate the relative error between the true function $f(\cdot)$ (left), $g(\cdot)$ (right) and the model estimates $\hat{f}_{40}(\cdot)$, $\hat{g}_{40}(\cdot)$ after taking 40 training points. The error is the lowest (in terms of absolute value) near the training data (black marks), which are plotted for illustration purposes in the x_1-x_2-plane.

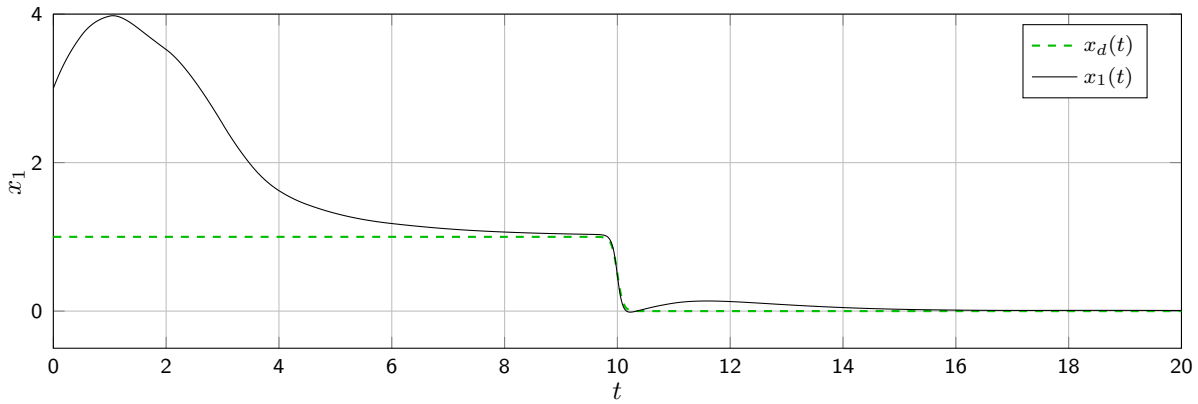

Figure 5.7: Scenario 1: The black solid line illustrates the actual, the green dashed line the desired value for the state x_1. The system converges to the desired state over time.

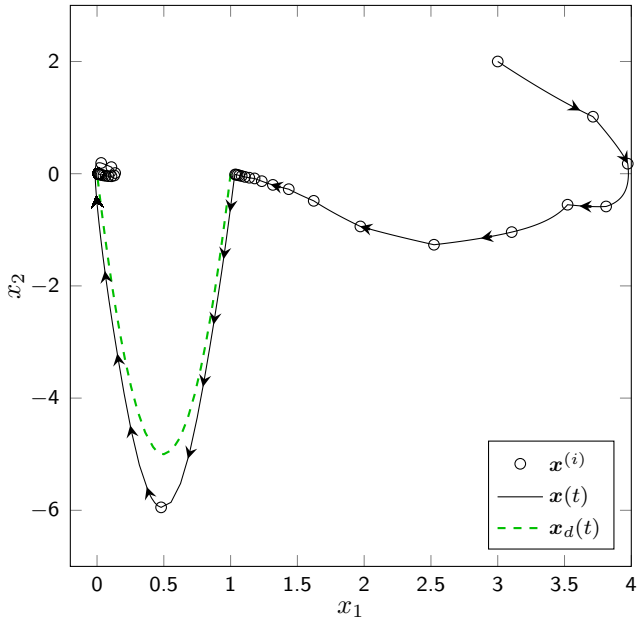

Figure 5.8: Scenario 1: Black circles indicate the collected training points, the black solid line illustrates the actual, the green dashed the desired trajectory. The system approaches the desired states as more training points are collected.

with $r_{\min}$. The event-triggered scheme stops taking new measurements after 29 data points are collected around the desired trajectory. In contrast, the time-triggered collects 60 data points (and would keep adding more) and is therefore computationally not as efficient as the event-triggered scheme.

Scenario 3: Data selection

In Scenario 3 (S3), the same setup as for S2 is chosen, but an upper limit of $\bar{N}^{(a)} = 10$ is imposed on the number of data points employed in the GP model. Therefore, Algorithm 5.2 is utilized with $N_d = 100$ and $\Delta t_d = \pi/100$. To highlight the benefits of the proposed data selection, we compare the data selection in Algorithm 5.3 with a random selection of the active data set $\mathbb{D}_{\kappa}^{(a)}$. For better illustration, we choose a sinusoidal reference trajectory with increasing amplitude

$$x_d(t) = 0.2t \sin(t)$$

and the simulation is manually stopped at $t = 12$.

Figure 5.11 shows the resulting trajectories in the state space and the active set of training data. For the proposed entropy-based selection, it can be observed that data points near $\mathbb{X}_v$ are chosen, which results in smaller values of $\sigma_{\kappa}(\cdot)$ along the future desired trajectory. This results in less events (60) compared to the random data selection (163).

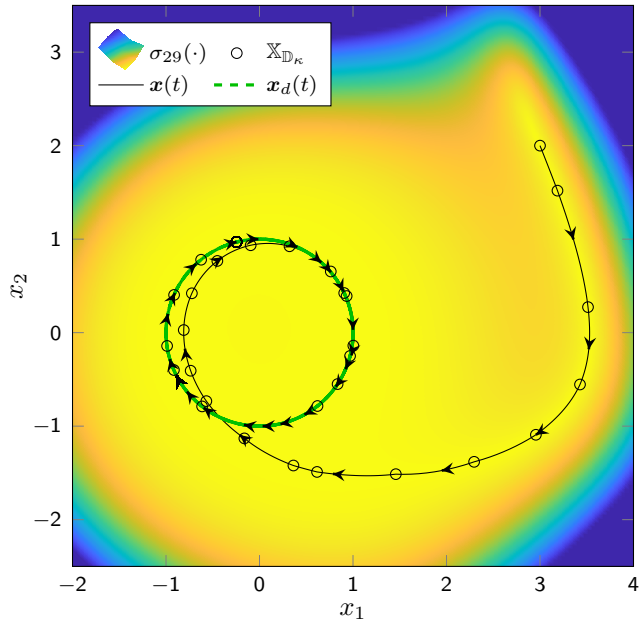

Figure 5.9: Scenario 2: Black circles indicate the collected training points, the black solid line illustrates the actual, the green dashed the desired trajectory. The colormap shows the variance function (2.4) for the GP $\sigma_{29}(\cdot)$ after the 29th update, where yellow indicates low variance and blue high variance.

5.6.2 Experimental results

Experimental Setup

For the evaluation in a robotic experiment, the two degree of freedom (DoF) manipulator CARBO as shown in Fig. 5.12 is employed. It has two rotational joints $\boldsymbol{q} = \begin{bmatrix} q_1 & q_2 \end{bmatrix}^\intercal \in \mathbb{R}^2$ and its dynamics are generally given by

$$\ddot{\boldsymbol{q}} = \underbrace{-\boldsymbol{M}_{\text{rob}}(\boldsymbol{q})^{-1} \left(\boldsymbol{C}_{\text{rob}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}_{\text{rob}}(\boldsymbol{q})\right)}_{\boldsymbol{f}(\boldsymbol{x})} + \underbrace{\boldsymbol{M}_{\text{rob}}(\boldsymbol{q})^{-1}}_{\boldsymbol{G}(\boldsymbol{x})} \boldsymbol{u},$$

where $\boldsymbol{M}_{\text{rob}} \colon \mathbb{R}^2 \to \mathbb{S}_+^2$ is the mass matrix, $\boldsymbol{C}_{\text{rob}} \colon \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^{2\times 2}$ is the Coriolis matrix and $\boldsymbol{g}_{\text{rob}} \colon \mathbb{R}^2 \to \mathbb{R}^2$ is the gravity vector. The input $\boldsymbol{u} \in \mathbb{R}^2$ is a vector of the applied torques in each joint, the state is $\boldsymbol{x} = \begin{bmatrix} q_1 & q_2 & \dot{q}_1 & \dot{q}_2 \end{bmatrix}^\intercal$

As we consider a similar scenario as in S2, $\boldsymbol{G}(\cdot)$ is assumed to be known (Assumption 5.6 holds), while $\boldsymbol{f}(\cdot)$ is unknown. Therefore, the mass matrix $\boldsymbol{M}_{\text{rob}}(\cdot)$ is given as follows

$$\boldsymbol{M}_{\text{rob}}(\boldsymbol{q}) = \begin{bmatrix} \mathfrak{I}_1 + \mathfrak{I}_2 + \mathfrak{M}_1 \tilde{\mathfrak{l}}_1^2 + \mathfrak{M}_2 \left(\mathfrak{l}_1^2 + \tilde{\mathfrak{l}}_2^2\right) + 2\mathfrak{M}_2 \mathfrak{l}_1 \tilde{\mathfrak{l}}_2 \cos(q_2) & \mathfrak{I}_2 + \mathfrak{M}_2 \tilde{\mathfrak{l}}_2^2 + 2\mathfrak{M}_2 \mathfrak{l}_1 \tilde{\mathfrak{l}}_2 \cos(q_2) \\ \mathfrak{I}_2 + \mathfrak{M}_2 \tilde{\mathfrak{l}}_2^2 + 2\mathfrak{M}_2 \mathfrak{l}_1 \tilde{\mathfrak{l}}_2 \cos(q_2) & \mathfrak{I}_2 + \mathfrak{M}_2 \tilde{\mathfrak{l}}_2^2 \end{bmatrix}$$

where the parameter values for the moments of inertia $\mathfrak{I}_1$, $\mathfrak{I}_2$, the centers of mass $\tilde{\mathfrak{l}}_1$, $\tilde{\mathfrak{l}}_2$, the lengths of the links $\mathfrak{l}_1$, $\mathfrak{l}_2$ and the masses $\mathfrak{M}_1$, $\mathfrak{M}_2$ are provided in Table 5.2.

The expression for $\boldsymbol{C}_{\text{rob}}(\cdot, \cdot)$ and $\boldsymbol{g}_{\text{rob}}(\cdot)$ are considered unknown. Therefore, $\boldsymbol{f}(\cdot)$ is identified using the proposed online learning. This covers all unknown components in the dynamics including external forces and any type of friction, which is usually very difficult to model with first order principles.

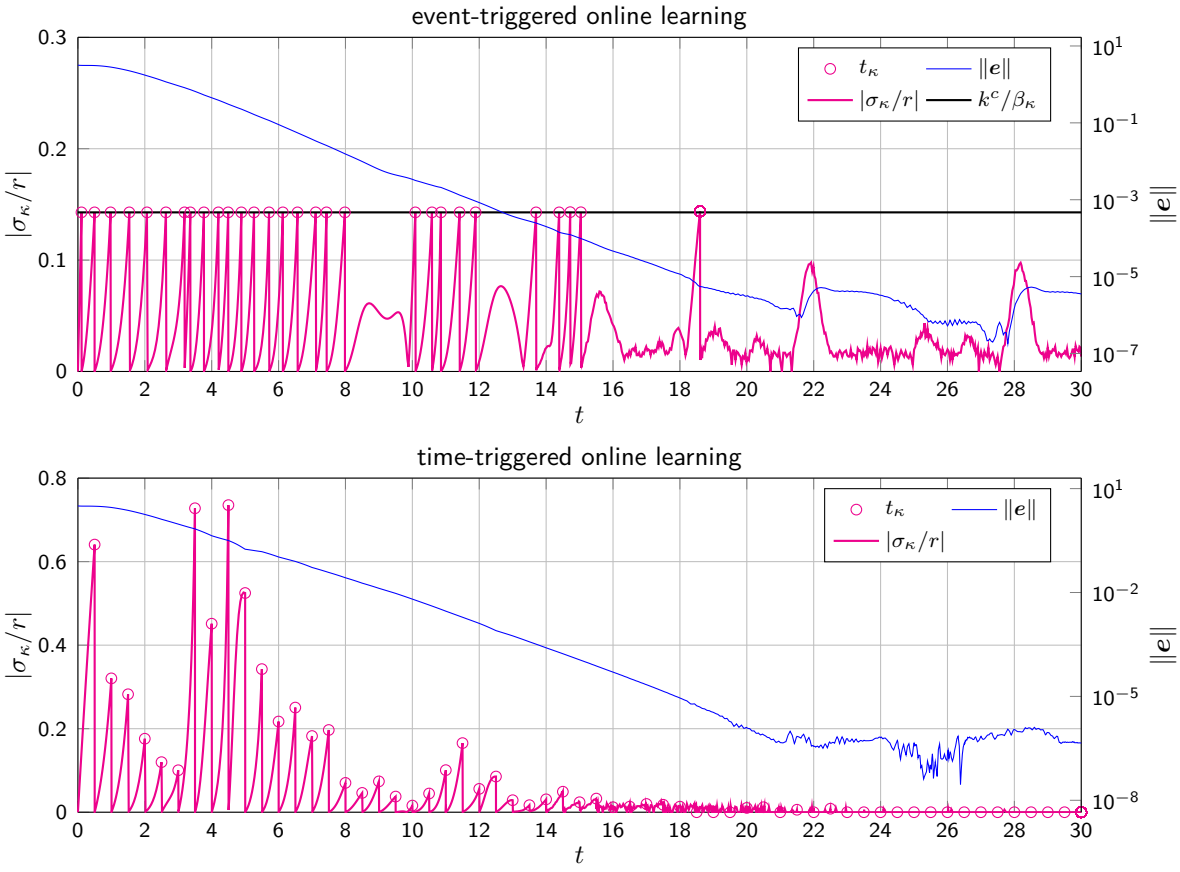

Figure 5.10: Scenario 2: Comparison of the event-triggered (top) and time-triggered (bottom) online learning. For the first, events (magenta circles) are triggered when the threshold k^c/β_κ (black horizontal line) is reached by $|\sigma_\kappa/r|$ as proposed in (5.26). For the latter, events are triggered after a fixed time interval ($\Delta t = 0.5$). The blue lines show the norm of the tracking error $\|\boldsymbol{e}\|$.

$\mathfrak{l}_1$	$\mathfrak{l}_2$	$\tilde{\mathfrak{l}}_1$	$\mathfrak{l}_2$	$\mathfrak{I}_1$	$\mathfrak{I}_2$	$\mathfrak{M}_1$	$\mathfrak{M}_2$
0.3 m	0.3 m	0.15 m	0.15 m	1 kg m^2	1 kg m^2	1.5 kg	1.5 kg

Table 5.2: Physical parameters of the two DoF robotic manipulator

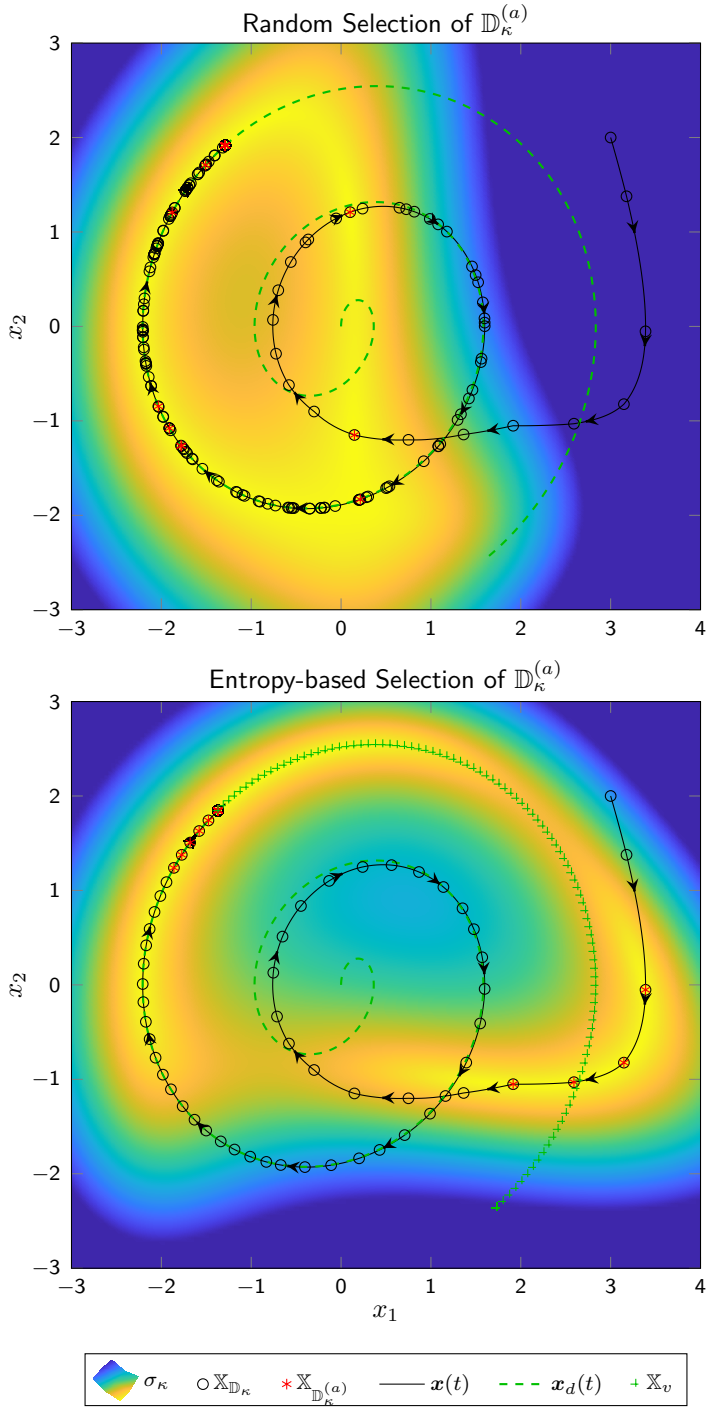

Figure 5.11: Scenario 3: Black circles indicate the collected training points $\mathbb{D}_\kappa$, the red asterisk the selected training data $\mathbb{D}_\kappa^{(a)}$, the black solid line illustrates the actual, the green dashed the desired trajectory. The colormap shows the variance function (2.4) for the GP $\sigma_\kappa(\cdot)$ after the last update, which is only based on $\bar{N}^{(a)} = 10$ data points in $\mathbb{D}_\kappa^{(a)}$ (yellow indicates low variance and blue high variance). On the top $\mathbb{D}_\kappa^{(a)}$ is chosen randomly from $\mathbb{D}_\kappa$. On the bottom, the proposed data selection Algorithm 5.3 is utilized.

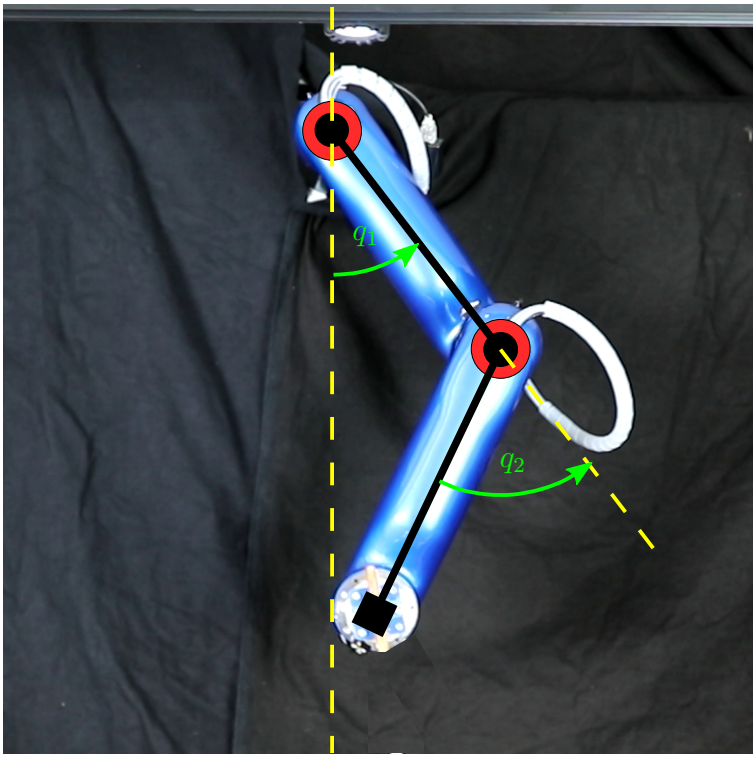

Figure 5.12: The two DoF robotic manipulator CARBO used for evaluation of the proposed event-triggered online learning approach.

Since $\boldsymbol{f}(\cdot)$ has a two-dimensional output, two GPs are employed. The input training data is for both the same, measurements of the state $\boldsymbol{x} = \begin{bmatrix}\boldsymbol{q}^\intercal & \dot{\boldsymbol{q}}^\intercal\end{bmatrix}^\intercal$, the output data are the rows of $-\boldsymbol{M}_{\text{rob}}(\boldsymbol{q})^{-1}\left(\boldsymbol{C}_{\text{rob}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}_{\text{rob}}(\boldsymbol{q})\right)$, respectively.

The even-triggering law (5.26) is extended to multiple dimensions using a logical OR-operation to trigger whenever one of the GPs has reached the threshold, thus

$$t_{\kappa+1} \coloneqq \{t > t_\kappa \,|\beta_{1,\kappa}\sigma_{1,\kappa}(\boldsymbol{x}) \geq k_1^c|r_1| \vee \beta_{2,\kappa}\sigma_{2,\kappa}(\boldsymbol{x}) \geq k_2^c|r_2|\}\,.$$

The hyperparameters of the GP models are set to fixed values as provided in Table 5.2, which are obtained from an offline learning procedure along the same reference trajectory which is used in the online learning scheme. The desired trajectory for joint 1 and 2 are

$$q_1 : x_{1,d}(t) = \frac{\pi}{6}\cos(0.5t), \qquad q_2 : x_{2,d}(t) = \frac{\pi}{6}\cos(t),$$

respectively.

The controller runs on an Ubuntu 14.04 real-time kernel with a Matlab/Simulink 2017 implementation and a sample rate of 1 kHz.

$\zeta_{f_1}^2$	$\zeta_{f_2}^2$	$\ell_{1:n,f_1}^2$	$\ell_{1:n,f_2}^2$	β_1	β_2	k_1^c	k_2^c	λ_1	λ_2
39	80	[0.8 4.9 13 11]	[13 3913 26 2.5]	29	21	4	3	10	10

Table 5.3: Parameters of the online learning controller in the robotic experiment.

Results

Similar to S2, we compare the proposed event-triggered online learning with a time-triggered online learning with $\Delta t = 0.5s$ and a model-free controller, where $\hat{f}_\kappa(\boldsymbol{x}) = 0$, $\forall \boldsymbol{x}$ is used in (5.4). The tracking error for both joints is shown in Figs. 5.13 and 5.14. The model-free controller (a PD controller with gains according to Table 5.3) shows no improvement over time, while the event-triggered and the time-triggered control laws show decreasing tracking errors.

Figure 5.15 shows the accumulative number of model updates over time. While the proposed event-triggered scheme adds most training points in the first period of the desired trajectory, the time-triggered scheme adds points at a constant rate. In the long run, the time-triggered update law adds too many training data point because the real-time constraint is violated for a data set with $N_\kappa > 80$. A problem which the event-triggered scheme circumvents because it only adds data points if required and is thereby more data-efficient.

5.7 Discussion

The proposed approach safely controls unknown systems, even if no data points are initially available. This is a crucial novelty because many previous approaches do not discuss how training data can be collected without harming the system or its environment. The identification scheme relies on Bayesian principles which enables to incorporate prior knowledge of the closed-loop system structure into the model. It is very powerful since a very large class of functions $f(\cdot)$ and $g(\cdot)$ can be represented properly using the flexibility of a nonparametric model.

The provided safety guarantees are based on the capability of the GP to measure its own fidelity and thereby to bound the model error. This property, introduced in Theorem 2.1, requires Assumption 5.7, which is very crucial. Even though it is less restrictive than parametric assumptions, it is still difficult to verify in practice. However, some assumptions are necessary because a generalization outside of the training set would otherwise not be possible [108]. Furthermore, the maximum mutual information γ_κ required in Theorem 2.1 is not trivial to obtain. However, [104] provides for the most common kernels upper bounds on γ_κ. Furthermore, alternative bound, which do not require this value have already been pointed out [103].

While this work mainly considers $g(\cdot)$, the effect of the control input on the dynamics, to be known, it is also possible to learn it from data. An analysis of the resulting convergence properties has been published recently [180] which allows to drop Assumption 5.6. While [180] only considers a static data set, we expect that an extension to the online event-triggered controller is directly possible. Furthermore, this approach does not consider a limitation on the control input. It can be shown, that u is bounded under the given assumptions, but predefined limits cannot actively be imposed.

The proposed event-triggered update law is highly data-efficient, since only necessary data points are collected. The feedback linearizing controller only requires a locally precise model, which makes it sufficient to store only a single data point (compare Corollary 5.6). This is a significant advantage over most existing approaches which require a globally precise model and thereby a large data set.

We are aware that two computational challenges remain open if the controller is applied

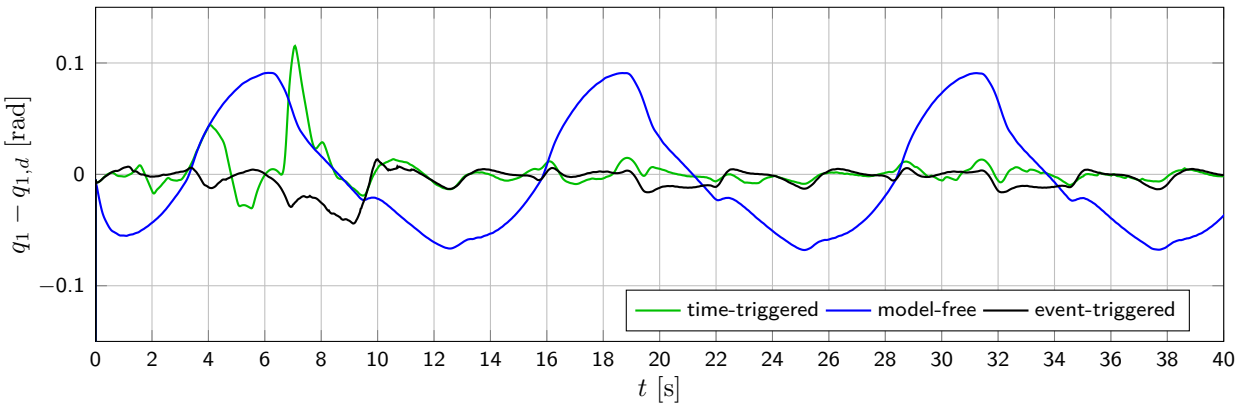

Figure 5.13: Comparison of the tracking error of the first joint $q_1 - q_{1,d}$ for the proposed event-triggered control (black), a time-triggered approach (green) and a model-free control approach (green).

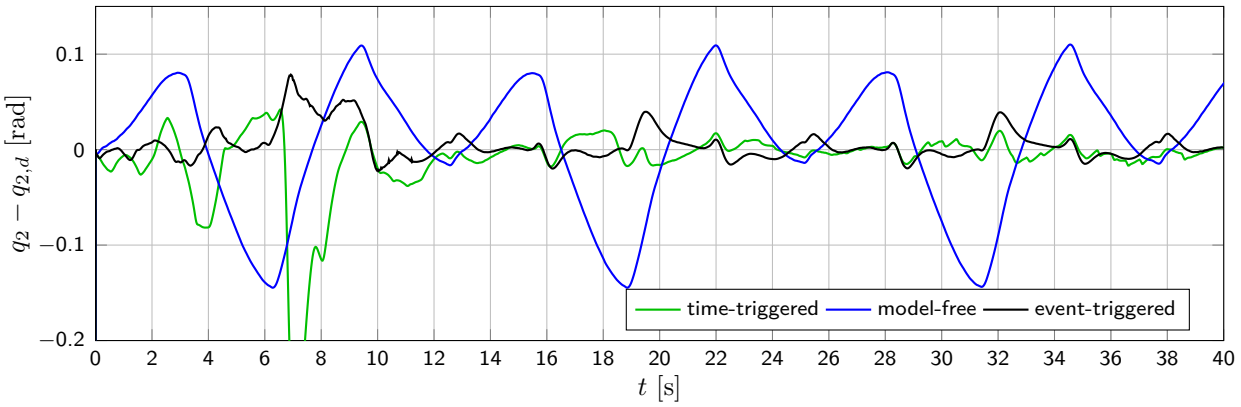

Figure 5.14: Comparison of the tracking error of the second joint $q_2 - q_{2,d}$ for the proposed event-triggered control (black), a time-triggered approach (green) and a model-free control approach (blue).

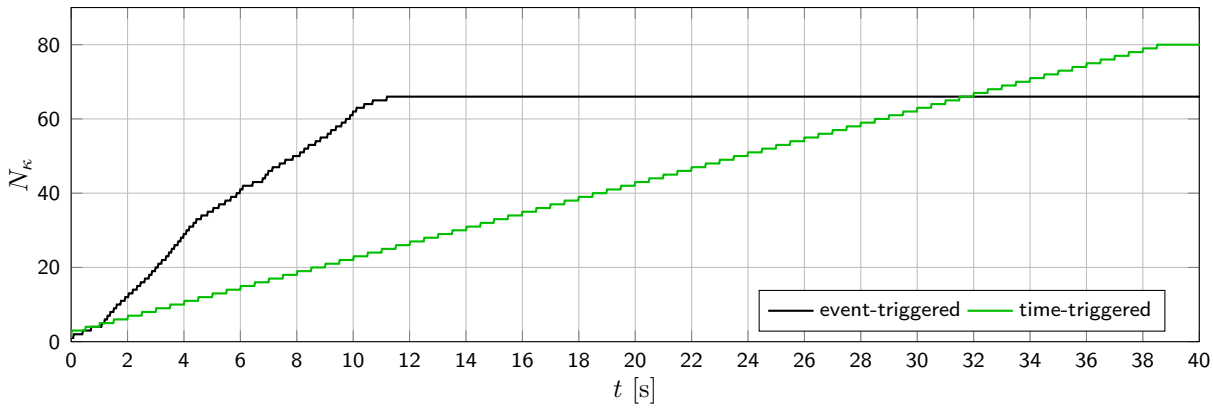

Figure 5.15: Accumulative number of model updates in comparison for the event-triggered and the time-triggered control law.

over a long time scale. First, the current algorithm requires a growing (and unbounded) amount of memory since data is never deleted. However, the presented data selection scheme has already demonstrated, how data can safely be ignored by the controller. In fact, considering the constraints (5.35), all non-active data could be deleted safely (maintaining the provided convergence guarantees) leading to a $\mathcal{O}(1)$ memory usage. Also the number of active data points can be chosen very small (as low as $\bar{N}^{(a)} = 1$) leading to minimal data storage requirements. Nevertheless, it should be considered that there is a trade-off between the number of stored / active data points (memory) and the frequency of the events, which are computationally quite expensive.

Second, the selection of the active data set adds an additional computational burden and the question whether it is beneficial requires a closer consideration: The greedy approximation proposed in Algorithm 5.3 requires $\mathcal{O}\left(N\bar{N}^{(a)}\right)$ steps, while the model update (including the matrix inversion) takes (with a rank-1 update) $\mathcal{O}\left(\bar{N}^{(a)^2}\right)$ or $\mathcal{O}\left(N^2\right)$ with or without data selection, respectively. Thus, for $N \gg \bar{N}^{(a)}$ the selection can be expected to speed up the model update step. But, even more important, the computation of the control law (GP posterior mean) requires $\mathcal{O}\left(\bar{N}^{(a)}\right)$ or $\mathcal{O}\left(N\right)$ steps and the detection of events (GP posterior variance) takes $\mathcal{O}\left(\bar{N}^{(a)^2}\right)$ or $\mathcal{O}\left(N^2\right)$ steps with or without data selection, respectively. These two operations are performed "continuously", i.e. occur much more frequently than an event and therefore the selection is expected to pay off for the overall computation time.

In conclusion, this approach contributes to solve Challenges 2 and 4 because the control law is safe from the first time instance, without requiring any knowledge of the system in advance. In comparison, to previous methods like NN-based controllers in [181], it quantifies the ultimate bound for the tracking error and in comparison to [84] asymptotically stabilizes the system. Key to this innovation is the event-triggered design of the control low which allows safe exploration of unknown areas while safely following a reference trajectory.

5.8 Summary

This chapter proposes a closed-loop nonparametric identification scheme for control-affine systems. It takes advantage of compound kernels to transfer knowledge about the system's structure into the model. Based on feedback linearization, the control law achieves global uniform ultimate boundedness of the tracking error for model updates at arbitrary time instances.

Furthermore, an event-triggered online learning scheme is proposed, which collects new training data only when necessary to ensure stability. This results in asymptotic stability guarantees for noise free measurements. To keep the control law computationally tractable this chapter presents an optimal selection of the most informative data points with respect to the future desired trajectory.

The proposed algorithms are illustrated in multiple simulation scenarios and evaluated in a robotic experiment.

6 Further Work

In addition to the contributions presented in Chapters 3 to 5, the author has published further work in the intersection of machine learning and control. This chapter summarizes these contributions very briefly. For more details, we refer to the respective publications.

6.1 Dynamic uncertainty-based leader-follower control

Humans cooperate very successfully in a variety of tasks, even if knowledge about the task is distributed asymmetrically among cooperation partners [182]. A common cooperation paradigm for asymmetric knowledge distribution is the so-called leader-follower scheme. Here, the well-informed partner adopts a leading role, while the less knowledgeable partner assumes a following role. Leader-follower schemes are well established in control [183], but the roles are rarely adapted dynamically (during task execution) and often not assigned based on knowledge.

Therefore, the work in [102] proposes a knowledge-based cooperation scheme using GPs. When an agent has high training data density in the current phase of the task, its GP posterior variance is low and it has high confidence about the task. As a result, it takes over a leading role, by being less cooperative and rather tracking its desired trajectory. When only little training data is currently available, the GP posterior variance increases and the agent focuses on cooperation (here, it strives to maintain the formation), rather than following its uncertain estimate of a desired trajectory. The comparison to a constant cooperation scheme is shown in Fig. 6.1.

6.2 Uncertainty modeling in programming by demonstration

The most appealing property of GPs is their representation of the fidelity of the model. However, most approaches ignore that the GP posterior variance function only represents the proximity to training data and thereby models uncertainty due to missing data. However, there might also be uncertainty due to contradiction in the data or multi-modality, which is not modeled by a GP.

Particularly, in learning by demonstration in robotics, where multiple demonstrations of the same task exist, the variability of the data carries crucial information about the task itself. For example, in the manipulation of an object, a phase with low demonstration variability might indicate a narrow passage in the task space.

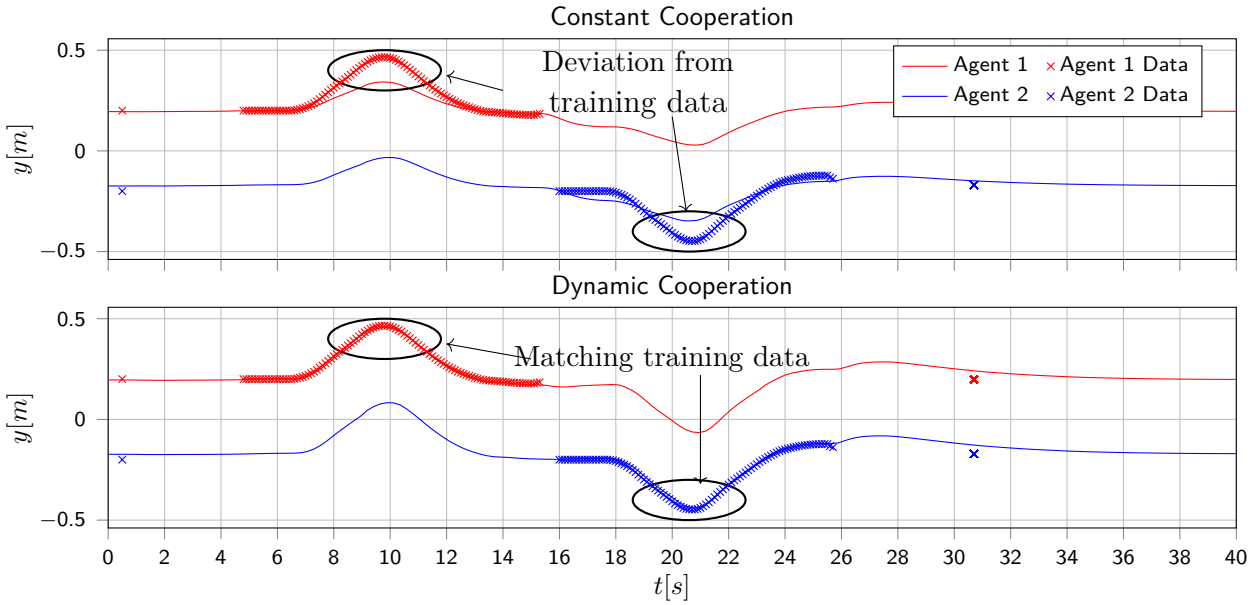

Figure 6.1: Illustration of the knowledge-based leader-follower cooperation. With constant cooperation, both agents are equal at any time. Therefore, they agree on a trajectory, which maintains the formation, but does not properly follow the training data. With the proposed dynamic cooperation, the agent, which has training data for the current phase of the task, takes over the lead and the other ensures that the formation is maintained. Here, Agent 1 leads for $t \in [5; 15]$ and Agent 2 for $t \in [16; 25.5]$.

The work in [170] proposes a GP-based modeling technique by applying Wishart-processes (presented in [184]) to a learning by demonstration scenario. In the reproduction of the presented motion, the stiffness of the robot is increased in phases with low variability in the training data and decreased where variability is high. Thereby, the controller is only aggressive when necessary and increases the safety in human-robot interaction. A visualization is provided in Fig. 6.2.

6.3 Scenario-based optimal control

Drawing (and storing) a sample from a GP is practically impossible because it is an infinite dimensional object [106]. This might be the reason why the scenario view on GPs is the least utilized among the presented interpretations in Sec. 2.3. Nevertheless, model predictive control schemes do not need to sample an entire function, but only a finite number of function values.

This insight enables the combination of a scenario-based optimal control approach and GP models in [107]. By sampling trajectories over a finite time horizon from a GP dynamical model, the variety of possible outcomes is properly represented. The work uses a differential dynamic programming approach to solve the optimal control problem. Based on the result from robust convex optimization, performance guarantees are derived. A visualization of this scenario-based approach is provided in Fig. 6.3.

6.4 Uncertainty-aware path tracking

Many programming by demonstration tasks require the replication or generalization of a presented path. In a large fraction of the literature, the desired motion is modeled as a dynamical system, due to its robustness to perturbation and real-time capability. Most works also provide guarantees for convergence to the desired final point, see e.g., dynamic movement primitives (DMP) [185] or stable estimator of dynamical systems (SEDS) [123].

However, all previous works ignore the uncertainty, which comes with sparse training data. In contrast, the work in [186] develops an uncertainty-based path tracking, which actively makes use of the capability of a GPSSM to quantify its uncertainty due to missing data. The approach actively strives into regions with more demonstration data and thus higher model certainty, which allows to reproduce human-demonstrated motions with higher precision than competitive state-of-the-art methods.

It is designed for goal-directed tasks, where it is essential to impose stability constraints on the model representing the human motion to make sure it reaches the correct final position. To guarantee this stability of the resulting trajectories, the presented approach uses a non-parametric GP-based Lyapunov function, which is visualized in Fig. 6.4.

6.5 Learning a stable state-dependent coefficient form

While most other contributions in this thesis focus on GPs, the work in [187] presents a more general framework to learn stable stochastic dynamical systems. More specifically, it considers the state-dependent coefficient form, which can represent arbitrary continuous

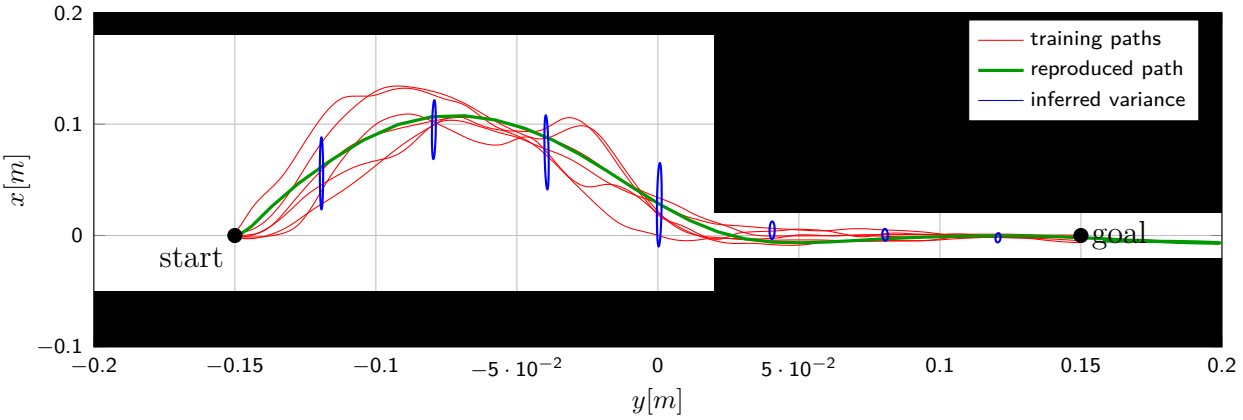

Figure 6.2: Shown are six training trajectories (red) provided by a human operator in the virtual environment and the reproduced trajectory (green). The proposed method infers the variability of the training data (blue). Based on this information, the manipulator is less stiff near the start ($y < 0.02$) which makes it safer in case of human interaction. It is more stiff near the goal ($y > 0.02$), which prevents collision with the obstructed environment in case of disturbances.

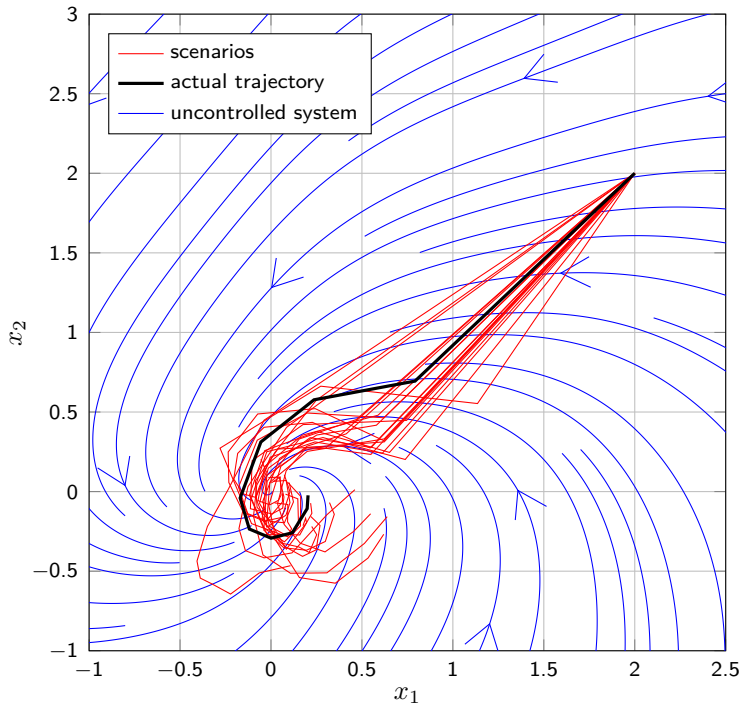

Figure 6.3: The blue arrows show the zero input behavior of an example non-linear system. The objective here is the stabilization at the origin. The red lines show the controlled behavior for different samples from the GP dynamic model. The black line shows the response of the real system to the optimal input sequence.

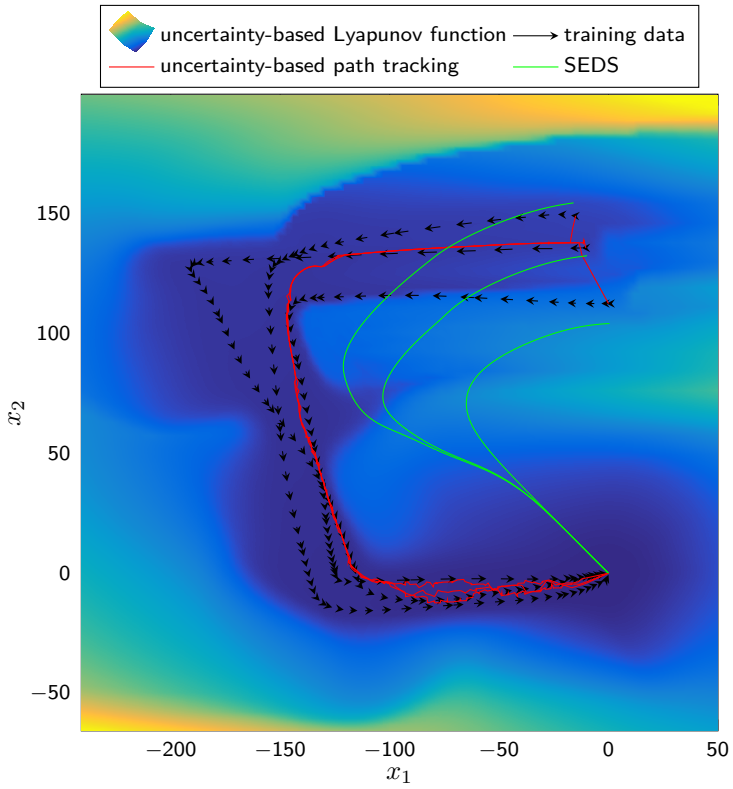

Figure 6.4: The black arrows show a training motion, which is supposed to be replicated in a learning by demonstration scenario. Previous approaches (SEDS in green) are not capable of following the training data precisely, because they do not include an incentive to stay near training data. In contrast, the proposed uncertainty-based Lyapunov function drives the generated trajectories towards the areas with low uncertainty (dark blue) and prevents the path-tracking from moving into areas without training data (light blue to yellow).

nonlinear dynamical systems. Thereby, it learns an arbitrary probability distribution as a function of the state space using a two layer learning architecture.

This is a major advantage over GP models, whose posterior variance function models the density of the data, but, e.g., cannot capture a multi-modal distribution of the data. As a main contribution, [187] derives constraints to ensure convergence with probability one of the resulting model. These constraints can directly be incorporated in the learning procedure. It applies this general framework to Beta and Dirichlet distributions and evaluates the presented techniques on a real-work human motion data set. An example is shown in Fig. 6.5.

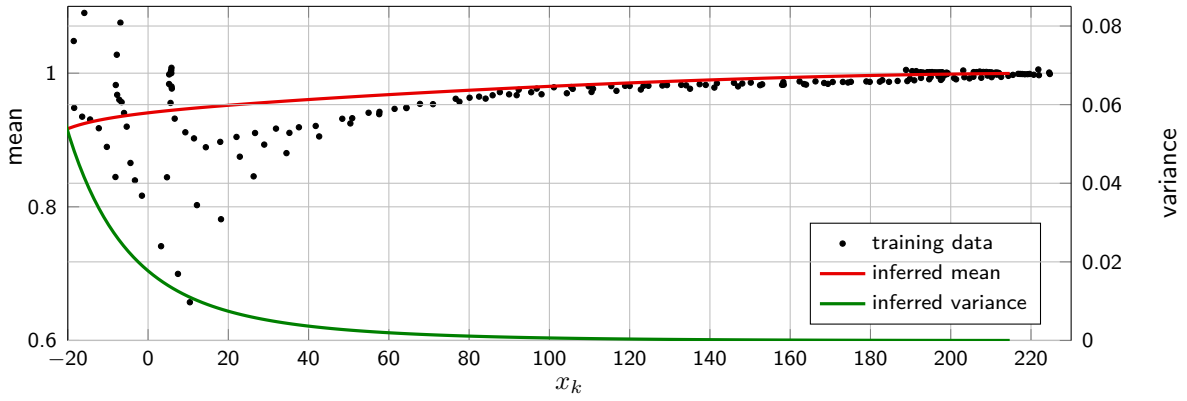

Figure 6.5: The inferred mean and variance of a Beta distribution are shown over the state space. The distribution accurately represents the high variance for $x < 50$ and the low variance for states with $x > 50$. The mean function properly averages the training data.

7 Conclusion

The application of data-driven models in control significantly progressed over the past years. More and more publications focus on formal guarantees of model-based control laws and take the present uncertainties due to imperfect modeling into consideration. This progress improves the reliability of learning-based control and supports its application in safety-critical domains, e.g., autonomous driving or physical human robot interaction.

This thesis focuses on Gaussian processes (GPs) as data-driven models and contributes novel concepts for system identification and the design of control laws which are summarized in the following.

Summary of contributions

After the introduction, we present in Chapter 2 a structured literature review on GPs in control and classify different interpretations of GPs used in existing works. Then, for the identification of dynamical systems, we propose a framework to impose consistent convergence behavior of the real unknown system and the nonparametric model. In a physical context, this ensures that the model also follows the principle of energy dissipation. The framework learns the convergence behavior from data and ensures this is mimicked by the Gaussian process state space model. We show that this approach under certain conditions never performs worse than a GP model without stabilization. In contrast to previous work, we include the uncertainty due to sparse training data in the model and derive how many data points are needed to guarantee asymptotic stability.

To take advantage of data-driven models, which include an estimate of the model fidelity, we propose a control law which actively avoids areas in the state space where the model uncertainty is high. The analysis shows that it is optimal, with respect to the probability of convergence, to steer the system along a trajectory near the training data where the model precision is high. This behavior is intuitive but has not been shown formally before.

If a training data set is not available, safe exploration of an unknown system is crucial. Therefore, we propose an online learning feedback linearizing control law, which is stable from the start and continuously improves the model over time. This event-triggered control scheme takes new measurements only if necessary and is thereby highly data-efficient. Compared to other online methods, it only operates on a local model making it unnecessary to store a large data set for a globally valid model. We show asymptotic stability of the resulting closed-loop switching system and propose an information-theoretic selection of data to reduce the computational complexity.

All results are supported by rigorous mathematical proofs along with numerical evaluations, showing that the proposed concepts are significant contributions to make data-driven models for control applicable in safety-critical domains.

Implications

We take this chance to revisit the challenges imposed for learning-based control in Sec. 1.1.

Challenge 1 Making a model logically and physically consistent with the real system requires to consider various aspects. In Chapter 3 this thesis focuses on the dissipation of energy. Learning not only the dynamics but also the convergence behavior from data allows to combine the flexibility of nonparametric models with consistent stability properties of the true system and the model. Further physical or logical verifications of the model are not studied.

Challenge 2 The question on formal guarantees and the required assumptions is addressed by the two proposed control laws, in an offline setting in Chapter 4 and an online setting in Chapter 5. Both show that despite the present uncertainties, asymptotic stability can be achieved in both cases. Two assumptions are necessary to draw this conclusion. First, the volatility of the dynamics must be restricted (here a bound on the reproducing kernel Hilbert space norm), which corresponds to a smooth dynamical system. Second, knowledge about the effect of the input on the state is required. While the first assumption is inevitable, according to the no-free-lunch theorems [108], the second originates from the chosen feedback linearizing control design.

Challenge 3 The existing control approaches which avoid uncertain areas and favor well-known areas, known as risk-sensitive control [129], are largely missing an analysis of stability. In Chapter 4 this work presents a novel controller which maximizes the probability for stability under a given power constraint and shows the desired risk-aware behavior for data-driven models.

Challenge 4 Systems for which initially no training data exists must be explored safely through an online closed-loop identification. Since the control law proposed in Chapter 5 requires only a locally precise model, the closed-loop stability can be guaranteed from the start and thereby allows safe exploration of unknown systems. Focusing on data efficiency, we develop an event-triggered update scheme which only adds training points if necessary due to a high model uncertainty. It thereby answers the question which data points should be collected. By proposing a safe forgetting strategy, which uses the information gain as criteria, we also address the question which data points should be kept in case of data processing or storage constraints.

Future directions

Although this thesis presents novel techniques to learning-based control and thereby makes progress towards the safe application in various tasks, many problems remain unsolved. We would like to point out these open research questions, which we suggest to study in the future.

Safe online learning

The simultaneous identification and control of an unknown system is, within this thesis, only discussed in Chapter 5 and is also underrepresented in the literature. However, this is one of the most urgent difficulties in the field. The assumption that an initial data set is available is hard to justify because there might be no safe method to obtain the data. Therefore, algorithms are required which start with no data points initially and safely control the system while exploring the state space. The idea of life-long learning, where data points are constantly added during operation, is important for constant improvement, however, solutions must be found to an infinitely increasing data set.

Computational and data efficiency

Gaussian processes are a computationally very heavy tool, which becomes intractable in most real-time applications for more than 10^4 data points. While there are many good reasons to choose GPs as a dynamical model (among others its very good generalization for small data sets), a solution must be found for complex systems with high dimensional state space. Since these require large data sets for precise control, the computational tractability and the data efficiency must be a focus of future work. The contribution in Sec. 5.5.1 is only a step in this direction.

Systems with unknown state spaces

An assumption, which was not explicitly stated here, is that the state space of the system is known. However, if the fundamental (physical) relations of a complex system are not understood, it might not just be impossible to derive the dynamics but also difficult to define the state space. An example which is often mentioned for the application of data-driven control is soft-robotics [188]. However, flexible structures typically require an analysis using finite element methods, which is only an approximation to an infinite dimensional state space. Therefore, future work must transfer the presented ideas to an input-output view of a dynamical system, which comes with many more challenges regarding observability and controllability of learned systems.

State and input constraints

The notion of *safety* in this thesis only considers the asymptotic behavior of the system, i.e. the derived bounds (e.g. on the tracking error) are not satisfied at all times, but are only ensured as time approaches infinity. One idea to resolve this difficulty is to analyze the rate of convergence using Lyapunov-based techniques, which guarantees a certain tracking precision for a given (finite) time interval. However, there exist various control approaches, e.g., MPC, control barrier functions and invariance control, which directly allow to impose state or input constraints in the design. Combining those methods with data-driven models and performing a rigorous analysis of the closed-loop behavior will be of great value for many safety-critical applications.

Formal guarantees with alternative regression models

The work in this thesis is focused on GPs because it allows to bound the model error under mild assumptions and thereby allows to prove formal stability results for the proposed control laws. However, GPs also have disadvantages (e.g., a high computational complexity), which is the reason why most publications in the field of machine learning focus on alternative approaches, such as neural network. Transferring the presented techniques and proofs to other supervised learning techniques would make data-driven control applicable in more complex tasks and empower learning-based control to operate large scale systems.

Notation

Acronyms & Abbreviations

a.s.	almost surely
CLF	control Lyapunov function
DoF	degree of freedom
DMP	dynamic movement primitives
GP	Gaussian process
GPSSM	Gaussian process state space model
GUUB	globally uniformly ultimately bounded
GAS	globally asymptotically stable
HJB	Hamilton-Jacobi-Bellman
i.i.d.	independent and identically distributed
ILC	iterative learning control
LQR	linear quadratic regulator
MPC	model predictive control
MRAC	model reference adaptive control
NN	neural network
RKHS	reproducing kernel Hilbert space
RL	reinforcement learning
SE	squared exponential
SEDS	stable estimator of dynamical systems
SOS	sum of squares
UAV	unmanned aerial vehicle

UGAS	uniformly globally asymptotically stable
UB	ultimately bounded
UUV	unmanned underwate vehicle
WSAQF	weighted sum of asymmetric quadratic functions

Conventions & Operators

$\lvert\cdot\rvert$	absolute value of a real number or cardinality of a set
$\boldsymbol{I}_N \in \mathbb{R}^{N\times N}$	identity matrix
$\mathbb{E}[\cdot]$	expected value of a random variable
$\mathbb{V}[\cdot]$	variance of a random variable
$.^{(i)}$	i-th training point
$.^\intercal$	transpose of a vector or matrix
$\lVert\cdot\rVert$	Euclidean norm of a vector (if not state differently)
$\mathbf{diag}(\cdot,\cdot,\cdots)$	arranges the elements on the diagonal of a diagonal matrix
$\lVert\cdot\rVert_{\mathfrak{k}}$	reproducing kernel Hilbert space norm of a function
$\langle\cdot,\cdot\rangle$	inner product of a vector space
$\cdot \sim \mathcal{N}(\cdot,\cdot)$	random variable follows a Gaussian distribution with mean and variance
$\cdot \sim \mathcal{GP}(\cdot,\cdot)$	function is distributed according to a Gaussian process with mean and variance function
$1:N$	integer indices from 1 to N, thus $1, 2, \ldots, N$
$\boldsymbol{x}^{(1:N)}$	concatenation of $\boldsymbol{x}_1$ to $\boldsymbol{x}_N$, thus $[\boldsymbol{x}_1 \ \cdots \ \boldsymbol{x}_N]^\intercal$
$f\left(\boldsymbol{x}^{(1:N)}\right)$	concatenation of function evaluations, thus $\left[f\left(\boldsymbol{x}^{(1)}\right) \ \cdots \ f\left(\boldsymbol{x}^{(N)}\right)\right]^\intercal$
$\mathcal{O}(\cdot)$	big O notation to describe limiting behavior of functions
$\mathcal{P}\{\cdot\}$	probability of an event
$p(\cdot)$	probability density function
$\det(\cdot)$	determinate of a matrix
$\log(\cdot)$	natural logarithm of a positive scalar
$\mathbf{rank}(\cdot)$	rank of a matrix

$\mathbf{trace}(\cdot)$	trace of a matrix
$x \in (a; b)$	open interval, thus $a < x < b$
$x \in [a; b]$	closed interval, thus $a \leq x \leq b$
$\nabla_{\boldsymbol{x}} \cdot$	Nabla operator, gradient of a scalar function
$\nabla_{\boldsymbol{x}}^{\intercal} \cdot$	Nabla operator, gradient transpose of a scalar fucntion
$\underline{\varrho}(\cdot), \bar{\varrho}(\cdot)$	minimal and maximal singular value of a matrix, respectively
$\underline{\varpi}(\cdot), \bar{\varpi}(\cdot)$	minimal and maximal eigenvalue value of a matrix, respectively
$\mathbf{0}$	vector/matrix of zeros of proper dimensions

Sets & Spaces

$\mathbb{B}, \mathbb{B}_{\iota}, \mathbb{B}_{\tilde{\iota}} \subseteq \mathbb{X}$	hyperball (with radius $\iota, \tilde{\iota}$)
$\mathbb{B}_{\sigma_{\mathbf{on}}} \subseteq \mathbb{X}$	hyperball whose radius depends on the observation noise σ_{on}
$\mathbb{B}_{\kappa} \subseteq \mathbb{X}$	hyperball at κ-th time step
$\mathbb{D}$, $\mathbb{D}_{\kappa}$	set of training data (after κ model updates)
$\mathbb{D}_{\kappa}^{(a)}, \mathbb{D}_{\kappa}^{(a^*)}$	(optimal) set of active training data at the κ-th time step
$\mathbb{D}_{\mathbf{add}}$	set of additional training data
$\mathbb{L}_2$	the space of square-integrable functions
$\mathbb{N}$	set of positive integers
$\mathbb{R}$, $\mathbb{R}_+$,$\mathbb{R}_{+,0}$	set of real / positive real / non-negative real numbers
$\mathbb{S}_+^n \subset \mathbb{R}^{n \times n}$	set of symmetric positive definite $n \times n$ matrices
$\mathbb{S}_{\epsilon_{\underline{\varpi}}}^n \subseteq \mathbb{S}_+^n$	set of symmetric $n \times n$ matrices, with all eigenvalues larger than $\epsilon_{\underline{\varpi}}$
$\mathbb{U} \subseteq \mathbb{R}^{n_u}$	set of control inputs
$\mathbb{X} \subseteq \mathbb{R}^n$	state or input space, refer to definition
$\mathbb{X}_{i_*} \subseteq \mathbb{X}$	region of attraction of the i_*-th equilibrium point
$\tilde{\mathbb{X}}_{i_*} \subseteq \mathbb{X}_{i_*}$	region around the i_*-th equilibrium point, refer to definition
$\mathbb{X}_d \subset \mathbb{X}$	manifold of states of the future desired trajectory
$\mathbb{X}_a \subset \mathbb{X}$	active input training points
$\mathbb{X}_v \subset \mathbb{X}$	state of interest

Functions

$\alpha_1, \alpha_2 \colon \mathbb{R}_{+,0} \to \mathbb{R}_{+,0}$	class $\mathcal{K}$ functions, i.e. strictly increasing and $\alpha(0) = 0$
$\mathfrak{b} \colon \mathbb{R}_+^n \to (0;1)$	computes the reliability for given error bounds
$\tilde{\beta} \colon (0;1) \to \mathbb{R}_+$	computes the constant β given a probability
$c_\sigma \colon \mathbb{X} \to \mathbb{R}$	cost function to plan optimal path
$\boldsymbol{C}_{\mathbf{rob}} \colon \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^{2\times 2}$	Coriolis matrix of a two DoF robotic manipulator
$\Delta f_\kappa, \Delta f \colon \mathbb{X} \to \mathbb{R}_{+,0}$	model error (in the κ-th time step)
$f \colon \mathbb{X} \to \mathbb{R}$	general scalar function, refer to definition
$\boldsymbol{f} \colon \mathbb{X} \to \mathbb{R}^m$	general vector field, refer to definition
$\bar{\boldsymbol{f}} \colon \mathbb{X} \to \mathbb{X}$	stabilized GPSSM to estimate $\boldsymbol{f}(\cdot)$
$\hat{f}, \hat{f}_\kappa \colon \mathbb{X} \to \mathbb{R}$	approximation to the function $f(\cdot)$ (at time step κ)
$\hat{\boldsymbol{f}} \colon \mathbb{X} \to \mathbb{R}^m$	approximation to the function $\boldsymbol{f}(\cdot)$, refer to definition
$\boldsymbol{f}_{\mathcal{GP}} \colon \mathbb{X} \to \mathbb{R}$	Gaussian process estimate for $\boldsymbol{f}(\cdot)$
$\boldsymbol{f}_{\mathbf{BS}} \colon \mathbb{X} \times \mathbb{S}_+ \to \mathbb{X} \times \mathbb{S}_+$	dynamics in the belief-space interpretation
$f_{\mathbf{sum}}, f_{\mathbf{prod}} \colon \mathbb{X} \to \mathbb{R}$	function, which is a sum / product of two functions
$f_a, f_b, f_h \colon \mathbb{X} \to \mathbb{R}$	functions, which are samples from a GP
$\boldsymbol{g}_{\mathbf{rob}} \colon \mathbb{R}^2 \to \mathbb{R}^2$	gravity vector of a two DoF robotic manipulator
$g \colon \mathbb{X} \to \mathbb{R}$	scaling of the input for a single input control affine system
$\boldsymbol{G} \colon \mathbb{X} \to \mathbb{R}^{n\times n}$	scaling of the input for a control affine system
$\hat{g}, \hat{g}_\kappa \colon \mathbb{X} \to \mathbb{R}$	approximation to the function $g(\cdot)$ (at time step κ)
$\bar{g}_\kappa \colon \mathbb{X} \to \mathbb{R}$	ratio of $g(\cdot)$ to $\hat{g}_\kappa(\cdot)$
$\mathfrak{g} \colon \mathbb{R} \to \mathbb{R}_+$	penalty function for fitting Lyapunov functions to data
$\tilde{\mathfrak{g}} \colon \mathbb{R}_+ \to \mathbb{R}_+$	penalty function in the positive domain for fitting Lyapunov functions to data
$h \colon \mathbb{X} \to \mathbb{R}$	scalar function in general, refer to definition
$H \colon \mathbb{X}^m \to \mathbb{R}_+$	entropy of m jointly distributed random variables
$\mathfrak{k}, \mathfrak{k}_a, \mathfrak{k}_b, \mathfrak{k}_h \colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}$	kernel functions of a GP
$\mathfrak{k}_{\mathbf{prod}} \colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}$	kernel function for product of a GP and a known function

$\mathfrak{k}_{\mathbf{SE}} \colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}$	squared exponential kernel
$\mathfrak{k}_{fg}, \mathfrak{k}_f, \mathfrak{k}_g \colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}$	kernel function for identification of control affine systems
$\boldsymbol{k}, \boldsymbol{k}_a, \boldsymbol{k}_b, \boldsymbol{k}_h, \boldsymbol{k}_g, \boldsymbol{k}_f \colon \mathbb{X} \to \mathbb{R}^N$	covariances between a test input $\boldsymbol{x}$ and the training data $\boldsymbol{x}^{(1:N)}$
$\boldsymbol{M}_{\mathbf{rob}} \colon \mathbb{R}^2 \to \mathbb{S}_+^2$	mass matrix of a two DoF robotic manipulator
$\mathbf{mon} \colon \mathbb{X} \to \mathbb{R}^{n_{\mathbf{poly}}}$	vector of monomials
$\tilde{M}, \tilde{N} \colon \mathbb{R} \to \mathbb{R}$	auxiliarily functions, refer to definition
$\mathfrak{m}, \mathfrak{m}_a, \mathfrak{m}_b \colon \mathbb{X} \to \mathbb{R}$	prior mean function of a GP
$\mu \colon \mathbb{X} \to \mathbb{R}$	posterior mean function of a GP in general
$\mu_f, \mu_g \colon \mathbb{X} \to \mathbb{R}$	posterior mean function of a GP to model functions $f(\cdot)$ and $g(\cdot)$, respectively
$\boldsymbol{\mu} \colon \mathbb{X} \to \mathbb{R}^m$	posterior mean function of m GPs concatenated as vector
$p_{\mathbf{poly}} \colon \mathbb{X} \to \mathbb{R}$	polynomial of degree d_{poly}
$p \colon \mathbb{R} \to \mathbb{R}_+$	probability density function, refer to definition
$\sigma \colon \mathbb{X} \to \mathbb{R}$	posterior standard deviation of a GP
$\boldsymbol{\sigma} \colon \mathbb{X} \to \mathbb{R}^m$	posterior standard deviation of m GPs concatenated as vector
$\sigma^2 \colon \mathbb{X} \to \mathbb{R}$	posterior variance function of a GP
$\boldsymbol{\Sigma} \colon \mathbb{X} \to \mathbb{R}^{m \times m}$	posterior variance function of m GPs concatenated as diagonal matrix
$\boldsymbol{\sigma}^2 \colon \mathbb{X} \to \mathbb{R}^m$	posterior variance function of m GPs concatenated as vector
$\sigma^2_{\mathbb{D}_2} \colon \mathbb{X} \to \mathbb{R}$	posterior variance of a GP for a dataset with two points
$\varsigma \colon \mathbb{R} \to (-0.5; 0.5))$	shifted sigmoid function
$V \colon \mathbb{X} \to \mathbb{R}_{+,0}$	general Lyapunov candidate, refer to defintion
$V_\sigma \colon \mathbb{X} \to \mathbb{R}_{+,0}$	uncertainty-aware Lyapunov function
$V_{\mathbf{clf}} \colon \mathbb{X} \to \mathbb{R}_{+,0}$	generic control Lyapunov function
$V_{\mathbf{val}} \colon \mathbb{X} \to \mathbb{R}_{+,0}$	value function for an optimal control problem
$V_{\boldsymbol{\theta}}, V^{i_*}_{\boldsymbol{\theta}_{i_*}} \colon \mathbb{X} \to \mathbb{R}_{+,0}$	parameterized Lyapunov function learned from data
$V_{\mathbf{Sq}} \colon \mathbb{X} \to \mathbb{R}_{+,0}$	quadratic Lyapunov function
$V_{\mathbf{sq}} \colon \mathbb{X} \to \mathbb{R}_{+,0}$	isotropic quadratic Lyapunov function

$V_{\mathbf{SOS}} \colon \mathbb{X} \to \mathbb{R}_{+,0}$	sum of squares Lyapunov function
$V_{\mathbf{WSAQF}} \colon \mathbb{X} \to \mathbb{R}_{+,0}$	weighted sum of asymmetric squares Lyapunov function
$W_0 \colon [-\exp(-1); \infty) \to [-1; \infty)$	principle branch of the Lambert W function
$W_{-1} \colon [-\exp(-1); 0) \to [-\infty; -1)$	lower branch of the Lambert W function
$\bar{V}, \underline{V} \colon \mathbb{R}_{+,0} \to \mathbb{R}_{+,0}$	auxiliarily functions to decompose a Lyapunov function
$\tilde{\boldsymbol{x}} \colon [0 \ S] \to \mathbb{X}$	path parameterized by the arclength $s \in [0; S]$
$\bar{Z}, \underline{Z} \colon \mathbb{R}_+ \times \mathbb{X} \to \mathbb{R}$	auxiliarily functions to analyze the posterior variance
$\bar{Z}', \underline{Z}' \colon \mathbb{R}_+ \to \mathbb{R}$	auxiliarily functions to analyze the posterior variance

Constants & Variables

$\beta \in \mathbb{R}_+$, $\boldsymbol{\beta} \in \mathbb{R}_+^m$	constants linking the model error to the variance of a GP
$b_{\mathbb{B}} \in \mathbb{R}_+$	desired size of the ultimate bound
$B_f \in \mathbb{R}$	bound for the reproducing kernel Hilbert space norm of $f(\cdot)$
$c_1, c_2 \in \mathbb{R}$	auxiliarily constants, refer to definition
$\boldsymbol{c}_1, \boldsymbol{c}_3 \in \mathbb{R}^n$	constants to bound terms of a Lyapunov function
$\boldsymbol{C}_2, \boldsymbol{C}_3 \in \mathbb{S}_+^n$	constants to bound terms of a Lyapunov function
$c_{1,\phi_\kappa}, c_{2,\phi_\kappa} \in \mathbb{R}_{+,0}$	constants defined for notational convenience
$d_{\mathbf{poly}} \in \mathbb{N}$	degree of a monomial
$\delta \in (0; 1)$	probability that the model error of a GP violates a certain bound
$E_{\mathbf{cor}} \in \mathbb{R}_{+,0}$	correction effort, refer to definition
$\boldsymbol{e} \in \mathbb{R}^n$	tracking error
$\epsilon \in \mathbb{R}_+$	arbitrarily small constant
$\epsilon_{\mathbf{fd}} \in \mathbb{R}_+$	constant to compute numerical gradients using finite differences
$\epsilon_{\underline{\varpi}} \in \mathbb{R}_+$	lower bound for eigenvalues of positive definite matrices
$\epsilon_V \in \mathbb{R}_+$	constant to resolve strict inequality constraints
$\gamma \in \mathbb{R}_+$	maximum mutual information
$g_{\mathbf{const}} \in \mathbb{R}_+$	gravity constant
$\boldsymbol{H} \in \mathbb{R}^{N \times N}$	diagonal matrix with $h\left(\boldsymbol{x}^{(i)}\right)$, $i = 1, \ldots, N$ on its diagonal

$i \in \{1, \ldots, N\}$	index (mainly used for training points)
$i_* \in \{1, \ldots, N_*\}$	index for equilibria
$\mathcal{I}_{i_*} \subseteq \{1, 2, \ldots, N\}$	indices of data points within the i_*-region of attraction
$\mathfrak{i} \in \{0, 1\}$	indicator variable for WSAQF
$\mathfrak{I}_1, \mathfrak{I}_2 \in \mathbb{R}_+$	moment of inertia of a two DoF robotic manipulator
$j \in \{1, \ldots, n\}$	index (mainly used for dimensions of the state space)
$\kappa \in \mathbb{N}_0$	time index for time-discrete event-triggered systems
$k^c \in \mathbb{R}_+$	controller gain
$\boldsymbol{K}, \boldsymbol{K}_a, \boldsymbol{K}_b, \boldsymbol{K}_h \in \mathbb{R}^{N \times N}$	covariance matrix of training data
$\boldsymbol{K}_f, \boldsymbol{K}_g \in \mathbb{R}^{N \times N}$	covariance matrix of training data for $\mathfrak{k}_f(\cdot,\cdot)$ / $\mathfrak{k}_g(\cdot,\cdot)$
$\boldsymbol{K}_{\mathbf{sum}}, \boldsymbol{K}_{\mathbf{prod}} \in \mathbb{R}^{N \times N}$	covariance matrix of training data, refer to definition
$\boldsymbol{K}_{\mathbf{on}} \in \mathbb{R}^{N \times N}$	covariance matrix of training data including observation noise
$\boldsymbol{K}_{fg} \in \mathbb{R}^{N \times N}$	covariance matrix of training data for the compound kernel $\mathfrak{k}_{fg}(\cdot,\cdot)$ including observation noise
$\boldsymbol{K}_{\mathbf{ol,cl}} \in \mathbb{R}^{N_{\mathbf{ol}} \times N}$	covariance matrix of training data for open- and closed-loop measurements
$\boldsymbol{K}_{\mathbf{ol,ol}} \in \mathbb{R}^{N_{\mathbf{ol}} \times N_{\mathbf{ol}}}$	covariance matrix of training data for open-loop measurements
$\ell^2 \in \mathbb{R}_+$	lengthscale of a SE kernel
$\ell_{j,f}^2, \ell_{j,g}^2 \in \mathbb{R}_+$	lengthscale of SE kernel $\mathfrak{k}_f(\cdot,\cdot), \mathfrak{k}_g(\cdot,\cdot)$, respectively
$\underline{\ell}^2 \in \mathbb{R}_+$	smallest lengthscale of (multiple) SE kernel(s)
$\underline{\varrho}_{\boldsymbol{G}} \in \mathbb{R}_{+,0}$	smallest singular value of a matrix $\boldsymbol{G}$
$L_\sigma \in \mathbb{R}_+$	Lipschitz constant of GP posterior standard deviation function
$l \in \{1, \ldots, L\}$	indexing variable in WSAQF
$L \in \mathbb{N}$	number of asymmetric squared functions in WSAQF
$\mathfrak{l}_1, \mathfrak{l}_2 \in \mathbb{R}_+$	length of links of two DoF robotic manipulator
$\tilde{\mathfrak{l}}_1, \tilde{\mathfrak{l}}_2 \in \mathbb{R}_+$	center of mass for each link of two DoF robotic manipulator
$\lambda_1, \ldots, \lambda_{n-1} \in \mathbb{R}$	coefficients of a linear filter
$\boldsymbol{\lambda} \in \mathbb{R}^{n-1}$	concatenation of coefficients of a linear filter
$\eta_{i_*} \in \mathbb{R}_{+,0}$	measure how well a Lyapunov function fits data

$\mathfrak{M}_1, \mathfrak{M}_2 \in \mathbb{R}_+$	mass of each link of a two DoF robotic manipulator
$\mu_{x_{\kappa+1}} \in \mathbb{X}$	mean of the next state in belief-space view
$\bar{\mu} \in \mathbb{R}_+$	upper bound for GP posterior mean function
$n \in \mathbb{N}$	dimensionality of an input or state space
$N \in \mathbb{N}$	number of training points
$N_{\mathbf{ol}} \in \mathbb{N}$	number of training points obtained from open-loop measurements
$\bar{N}_{\kappa}^{(a)} \in \mathbb{N}$	number of training points in the active data set in the κ-th time step
$\bar{N}^{(a)} \in \mathbb{N}$	budget (maximum number of training points in active data set)
$N_d \in \mathbb{N}$	number of points sampled from future desired trajectory
$N_* \in \mathbb{N}$	number of equilibria
$N_{\mathbf{init}} \in \mathbb{N}$	number of different initial points
$N_{\mathbf{vio}} \in \mathbb{N}$	number of trajectories violating the ultimate bound
$N_{\mathbf{add}} \in \mathbb{N}$	number of added data points
$n_{\mathbf{poly}} \in \mathbb{N}$	number of a monomials with a degree d_{poly} in n dimensions
$\nu \in \mathbb{R}$	linear feedback control
$\omega \in \mathbb{R}, \boldsymbol{\omega} \in \mathbb{R}^n$	independent and identically distributed noise
$\boldsymbol{\psi}, \boldsymbol{\psi}_{fg}$	hyperparameters of a Gaussian process / composite kernel $\mathfrak{k}_{fg}(\cdot,\cdot)$
$\phi \in \mathbb{R}$	upper bound for triggering condition
$\boldsymbol{P}_{0\ldots L} \in \mathbb{S}_+^n$	coefficient matrices for the WSAQF
$\boldsymbol{P} \in \mathbb{R}^{n\times n}$	coefficient matrix for a quadratic function
$\boldsymbol{Q} \in \mathbb{R}^{n\times n}$	coefficient matrix for sum of squares function
$q_1, q_2 \in \mathbb{R}, \boldsymbol{q} \in \mathbb{R}^2$	joint angles of a two DoF robot in radiants
$r_{\mathbf{tr}} \in \mathbb{R}_+$	radius around a training point
$\rho \in \mathbb{R}$	filtered tracking error
$r \in \mathbb{R}$	filtered state
$r_{\mathbf{min}} \in \mathbb{R}_+$	lower bound for the absolute value of r
$r_a, \tilde{r}_a \in \mathbb{R}_{+,0}$	distances in the state space, refer to definition

$\Upsilon_1, \ldots, \Upsilon_n \in \mathbb{N}$	exponents of monomials
$\iota, \tilde{\iota} \in \mathbb{R}_+$	radius of a ball or utlimate bound
$\sigma_{\mathbf{on}} \in \mathbb{R}_{+,0}$	standard deviation of the observation noise
$\sigma^2_{\mathbf{on}} \in \mathbb{R}_{+,0}$	variance of the observation noise
$\zeta^2 \in \mathbb{R}_{+,0}$	signal variance of a SE kernel
$\zeta_f^2, \zeta_g^2 \in \mathbb{R}_{+,0}$	signal variance of SE kernel $\mathfrak{k}_f(\cdot,\cdot), \mathfrak{k}_g(\cdot,\cdot)$, respectively
$\bar{\zeta}^2 \in \mathbb{R}_{+,0}$	maximal signal variance of mutiple SE kernels
$\bar{\sigma}^2 \in \mathbb{R}_+$	upper bound for GP posterior variance
$\bar{\sigma} \in \mathbb{R}_+$	upper bound for GP posterior standard deviation
$\sigma^2_{x_{\kappa+1}} \in \mathbb{R}_+$	variance of the state in belief-space view
$s \in \mathbb{C}$	complex variable in a characteristic polynomial
$\mathfrak{s} \in \mathbb{R}$	paramterization of a path
$\mathfrak{S} \in \mathbb{R}$	upper limit for paramterization of a path
$\boldsymbol{\theta} \in \boldsymbol{\Theta}, \boldsymbol{\theta}_{i_*} \in \boldsymbol{\Theta}_{i_*}$	parameter vector for Lyapunov function $V_{\boldsymbol{\theta}}(\cdot), V^{i_*}_{\boldsymbol{\theta}_{i_*}}(\cdot)$
$\boldsymbol{\Theta}, \boldsymbol{\Theta}_{i_*} \subseteq \mathbb{R}^{n_\theta}$	possible parameter set with dimension n_θ
$t_\kappa \in \mathbb{R}_+$	current time instance
$t_{\kappa+1} \in \mathbb{R}_+$	next time instance
$\Delta t, \Delta t_\kappa \in \mathbb{R}_+$	time interval (in the κ-th step)
$\Delta t_d \in \mathbb{R}_+$	time interval at which the future desired trajectory is sampled
$\underline{\Delta t} \in \mathbb{R}_+$	lower bound for the inter-event time
$\tau \in \mathbb{R}_+$	auxiliarily variable, refer to definition
$\upsilon \in (0;1)$	reliability (probability that a system is stable)
$u \in \mathbb{R},\ \boldsymbol{u} \in \mathbb{R}^{n_u}$	control input, refer to definition
$u_{\mathbf{max}} \in \mathbb{R}_+$	maximum control power
$\boldsymbol{U} \in \mathbb{R}^{N \times N}$	diagonal matrix with control input at training points
$\boldsymbol{x} \in \mathbb{X}$	state of a system / input to a function
$\boldsymbol{x}_\kappa, x_\kappa \in \mathbb{X}$	current state of a time discrete system
$\boldsymbol{x}_{\kappa+1}, x_{\kappa+1} \in \mathbb{X}$	next state of a time discrete system

$\boldsymbol{x}_g \in \mathbb{X}$	goal state to which a system converges
$\boldsymbol{x}_{\mathbf{kn}} \in \mathbb{X}$	state at which the function value is known
$\hat{\boldsymbol{x}}_*, \hat{\boldsymbol{x}}_{i_*} \in \mathbb{X}$	estimate of the (i_*-)equilibrium point
$\boldsymbol{x}_*, \boldsymbol{x}_{i_*} \in \mathbb{X}$	the (i_*-th) equilibrium point
$\boldsymbol{x}^{(a)}, \boldsymbol{x}^{(\tilde{a})} \in \mathbb{X}$	specific points in the state space, refer to definition
$\boldsymbol{x}_d \in \mathbb{X}$	desired trajectory
$\boldsymbol{x}_1^{\mathrm{WSAQF}}, \ldots, \boldsymbol{x}_L^{\mathrm{WSAQF}} \in \mathbb{X}$	centers in the weighted sum of asymmetric quadratic functions
$y \in \mathbb{R}, \boldsymbol{y} \in \mathbb{R}^n$	noisy measurement of the output of a function
$y_{\mathbf{kn}} \in \mathbb{R}$	known function value at $\boldsymbol{x}_{\mathrm{kn}}$
$y_{\mathbf{sum}} \in \mathbb{R}$	noisy measurements of the sum of two functions
$y_{\mathbf{prod}} \in \mathbb{R}$	noisy measurements of the product of two functions
$y_{\mathbf{ol}} \in \mathbb{R}$	noisy open loop measurements
$z \in \mathbb{R}$	auxiliarily variable

List of Figures

List of Tables

List of Algorithms

Bibliography

[1] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, *et al.*, "Mastering the game of go without human knowledge", *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[2] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", in *Advances in Neural Information Processing Systems (NeurIPS)*, Curran Associates, Inc., 2012, pp. 1097–1105.

[4] N. Du, Y. Wang, N. He, J. Sun, and L. Song, "Time-sensitive recommendation from recurrent user activities", in *Advances in Neural Information Processing Systems (NeurIPS)*, Curran Associates, Inc., 2015, pp. 3492–3500.

[5] P. Ponte and R. G. Melko, "Kernel methods for interpretable machine learning of order parameters", *Physical Review B*, vol. 96, no. 20, p. 205 146, 20 2017.

[6] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization", *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[7] J. Umlauft, A. Lederer, and S. Hirche, "Learning stable Gaussian process state space models", in *American Control Conference (ACC)*, IEEE, 2017, pp. 1499–1504.

[8] J. Umlauft and S. Hirche, "Learning stochastically stable Gaussian process state-space models", *IFAC Journal of Systems and Control*, vol. 12, p. 100 079, 2020.

[9] J. Umlauft, L. Pöhler, and S. Hirche, "An uncertainty-based control Lyapunov approach for control-affine systems modeled by Gaussian process", *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 483–488, 2018.

[10] J. Umlauft, T. Beckers, M. Kimmel, and S. Hirche, "Feedback linearization using Gaussian processes", in *Conference on Decision and Control (CDC)*, IEEE, 2017, pp. 5249–5255.

[11] J. Umlauft and S. Hirche, "Feedback linearization based on Gaussian processes with event-triggered online learning", *IEEE Transactions on Automatic Control (TAC)*, pp. 1–16, 2020.

[12] J. Umlauft, T. Beckers, A. Capone, A. Lederer, and S. Hirche, "Smart forgetting for safe online learning with Gaussian processes", in *Learning for Dynamics and Control (L4DC)*, ser. Proceedings of Machine Learning Research, vol. 120, The Cloud: PMLR, 2020, pp. 160–169.

[13] T. Beckers, J. Umlauft, and S. Hirche, "Mean square prediction error of misspecified Gaussian process models", in *Conference on Decision and Control (CDC)*, IEEE, 2018, pp. 1162–1167.

[14] M. Liu, G. Chowdhary, B. Castra da Silva, S. Y. Liu, and J. P. How, "Gaussian processes for learning and control: A tutorial with examples", *IEEE Control Systems Magazine*, vol. 38, no. 5, pp. 53–86, 2018.

[15] J. Kocijan, *Modelling and Control of Dynamic Systems Using Gaussian Process Models*. Springer, 2016.

[16] L. Ljung, *System Identification*. NJ, USA: Prentice Hall PTR, 1998.

[17] K. Narendra and K. Parthasarathy, "Adaptive identification and control of dynamical systems using neural networks", in *Conference on Decision and Control (CDC)*, vol. 2, IEEE, 1989, pp. 1737–1738.

[18] M. M. Polycarpou and P. A. Ioannou, *Identification and control of nonlinear systems using neural network models: Design and stability analysis*. University of Southern Calif., 1991.

[19] P. M. Nørgård, O. Ravn, N. K. Poulsen, and L. K. Hansen, *Neural Networks for Modelling and Control of Dynamic Systems - A Practitioner's Handbook*. London: Springer, 2000.

[20] J. Kocijan, B. Banko, B. Likar, A. Girard, R. Murray-Smith, and C. E. Rasmussen, "A case based comparison of identification with neural network and Gaussian process models.", in *International Conference on Intelligent Control Systems and Signal Processing (ICONS)*, Max-Planck-Gesellschaft, vol. 1, 2003, pp. 137–142.

[21] J. Kocijan, A. Girard, B. Banko, and R. Murray-Smith, "Dynamic systems identification with Gaussian processes", *Mathematical and Computer Modelling of Dynamical Systems*, vol. 11, no. 4, pp. 411–424, 2005.

[22] J. M. Maciejowski, *Predictive control: with constraints*. Prentice Hall, 2002.

[23] A. Girard, C. E. Rasmussen, J. Quinonero-Candela, and R. Murray-Smith, "Multiple-step ahead prediction for non linear dynamic systems–a Gaussian process treatment with propagation of the uncertainty", *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 15, pp. 529–536, 2002.

[24] J. Kocijan, "Dynamic GP models: An overview and recent developments", in *International Conference on Applied Mathematics, Simulation, Modelling (ASM)*, 2012, pp. 38–43.

[25] J. Kocijan, "Control algorithms based on Gaussian process models: A state-of-the-art survey", in *Special International Conference on Complex Systems: Synergy of Control, Communications and Computing (COSY)*, 2011, pp. 69–80.

[26] T. Beckers and S. Hirche, "Stability of Gaussian process state space models", in *European Control Conference (ECC)*, 2016, pp. 2275–2281.

[27] ——, "Equilibrium distributions and stability analysis of Gaussian process state space models", in *Conference on Decision and Control (CDC)*, IEEE, 2016, pp. 6355–6361.

[28] M. Kuss and C. E. Rasmussen, "Gaussian processes in reinforcement learning", in *Advances in Neural Information Processing Systems (NeurIPS)*, Curran Associates, Inc., 2004, pp. 751–758.

[29] D. Sbarbaro and R. Murray-Smith, "Self-tuning control of non-linear systems using Gaussian process prior models", in *Switching and Learning in Feedback Systems: European Summer School on Multi-Agent Control, Maynooth, Ireland, September 8-10, 2003, Revised Lectures and Selected Papers*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 140–157.

[30] M. P. Deisenroth, C. E. Rasmussen, and J. Peters, "Model-based reinforcement learning with continuous states and actions", in *European Symposium on Artificial Neural Networks (ESANN)*, 2008.

[31] J. Hall, C. E. Rasmussen, and J. Maciejowski, "Reinforcement learning with reference tracking control in continuous state spaces", in *Conference on Decision and Control (CDC)*, IEEE, 2011, pp. 6019–6024.

[32] M. P. Deisenroth and C. E. Rasmussen, "PILCO: A model-based and data-efficient approach to policy search", in *International Conference on Machine Learning (ICML)*, 2011, pp. 465–472.

[33] M. P. Deisenroth, *Efficient reinforcement learning using Gaussian processes*. KIT Scientific Publishing, 2010, vol. 9.

[34] J. Hall, C. E. Rasmussen, and J. Maciejowski, "Modelling and control of nonlinear systems using Gaussian processes with partial model information", in *Conference on Decision and Control (CDC)*, IEEE, 2012, pp. 5266–5271.

[35] M. P. Deisenroth, C. E. Rasmussen, and D. Fox, "Learning to control a low-cost manipulator using data-efficient reinforcement learning", in *Robotics: Science and Systems VII*. MITP, 2012, ch. 8, pp. 57–64.

[36] M. P. Deisenroth, D. Fox, and C. E. Rasmussen, "Gaussian processes for data-efficient learning in robotics and control", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 2, pp. 408–423, 2015.

[37] M. P. Deisenroth, G. Neumann, J. Peters, *et al.*, "A survey on policy search for robotics", *Foundations and Trends in Robotics*, vol. 2, no. 1–2, pp. 1–142, 2013.

[38] J. Vinogradska, B. Bischoff, J. Achterhold, T. Koller, and J. Peters, "Numerical quadrature for probabilistic policy search", *Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, pp. 164–175, 1 2018.

[39] J. Vinogradska, B. Bischoff, D. Nguyen-Tuong, H. Schmidt, A. Romer, and J. Peters, "Stability of controllers for Gaussian process forward models", in *International Conference on Machine Learning (ICML)*, JMLR, NY, USA, 2016, pp. 545–554.

[40] J. Vinogradska, "Gaussian processes in reinforcement learning: Stability analysis and efficient value propagation", PhD thesis, Technical University of Darmstadt, 2018.

[41] B. Bethke and J. P. How, "Approximate dynamic programming using bellman residual elimination and Gaussian process regression", in *American Control Conference (ACC)*, IEEE, 2009, pp. 745–750.

[42] M. P. Deisenroth, J. Peters, and C. E. Rasmussen, "Approximate dynamic programming with Gaussian processes", in *American Control Conference (ACC)*, IEEE, 2008, pp. 4480–4485.

[43] C. E. Deisenroth Marc P.and Rasmussen and J. Peters, "Gaussian process dynamic programming", *Neurocomputing*, vol. 72, no. 7-9, pp. 1508–1524, 2009.

[44] Y. Pan and E. Theodorou, "Probabilistic differential dynamic programming", in *Advances in Neural Information Processing Systems (NeurIPS)*, Curran Associates, Inc., 2014, pp. 1907–1915.

[45] B. Bischoff, D. Nguyen-Tuong, H. Markert, and A. Knoll, "Learning control under uncertainty: A probabilistic value-iteration approach", in *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, 2013.

[46] J. Boedecker, J. T. Springenberg, J. Wülfing, and M. Riedmiller, "Approximate real-time optimal control based on sparse Gaussian process models", in *Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, 2014, pp. 1–8.

[47] L. Grüne and J. Pannek, *Nonlinear model predictive control: theory and algorithms*. Springer Publishing Company, Incorporated, 2013.

[48] J. Kocijan, R. Murray-Smith, C. E. Rasmussen, and B. Likar, "Predictive control with Gaussian process models", in *The IEEE Region 8 EUROCON Computer as a Tool.*, vol. 1, 2003, pp. 352–356.

[49] J. Kocijan, R. Murray-Smith, C. E. Rasmussen, and A. Girard, "Gaussian process model based predictive control", in *American Control Conference (ACC)*, vol. 3, IEEE, 2004, pp. 2214–2219.

[50] A. Grancharova, J. Kocijan, and T. A. Johansen, "Explicit stochastic nonlinear predictive control based on Gaussian process models", in *European Control Conference (ECC)*, 2007, pp. 2340–2347.

[51] U. Rosolia and F. Borrelli, "Learning model predictive control for iterative tasks", *arXiv preprint arXiv:1609.01387*, 2016.

[52] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, "Learning-based model predictive control for safe exploration", in *Conference on Decision and Control (CDC)*, IEEE, 2018, pp. 6059–6066.

[53] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious model predictive control using Gaussian process regression", *IEEE Transactions on Control Systems Technology*, pp. 1–8, 2019.

[54] G. Cao, E. M.-K. Lai, and F. Alam, "Gaussian process model predictive control of an unmanned quadrotor", *Journal of Intelligent & Robotic Systems*, vol. 88, no. 1, pp. 147–162, 2017.

[55] C. J. Ostafew, A. P. Schoellig, T. D. Barfoot, and J. Collier, "Learning-based nonlinear model predictive control to improve vision-based mobile robot path tracking", *Journal of Field Robotics*, vol. 33, no. 1, pp. 133–152, 2016.

[56] G. Gregorcic and G. Lightbody, "Gaussian process approaches to nonlinear modelling for control", in *Intelligent Control Systems Using Computational Intelligence Techniques*. Institution of Engineering and Technology, 2005, ch. 6, pp. 177–217.

[57] G. Gregorčič and G. Lightbody, "Gaussian process internal model control", in *International PhD Workshop on Advances in Supervision and Control Systems*, Strunjan, Slovenia, 2002, pp. 39–46.

[58] G. Gregorcic and G. Lightbody, "Internal model control based on a Gaussian process prior model", in *American Control Conference (ACC)*, vol. 6, IEEE, 2003, pp. 4981–4986.

[59] R. Murray-Smith and D. Sbarbaro, "Nonlinear adaptive control using nonparametric Gaussian process prior models", *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 325–330, 2002.

[60] R. Murray-Smith, D. Sbarbaro, C. E. Rasmussen, and A. Girard, "Adaptive, cautious, predictive control with Gaussian process priors", in *IFAC Symposium on System Identification*, 2003, pp. 1195–1200.

[61] G. Chowdhary, H. A. Kingravi, J. P. How, and P. A. Vela, "Bayesian nonparametric adaptive control of time-varying systems using Gaussian processes", in *American Control Conference (ACC)*, IEEE, 2013, pp. 2655–2661.

[62] ——, "Bayesian nonparameteric model reference adaptive control using Gaussian processes", in *Guidance, Navigation, and Control Conference (GNC)*, AIAA, 2013.

[63] ——, "Bayesian nonparametric adaptive control using Gaussian processes", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 3, pp. 537–550, 2015.

[64] G. Joshi and G. Chowdhary, "Adaptive control using Gaussian-process with model reference generative network", in *Conference on Decision and Control (CDC)*, IEEE, 2018, pp. 237–243.

[65] R. C. Grande, G. Chowdhary, and J. P. How, "Nonparametric adaptive control using Gaussian processes with online hyperparameter estimation", in *Conference on Decision and Control (CDC)*, IEEE, 2013, pp. 861–867.

[66] F. Berkenkamp and A. P. Schoellig, "Learning-based robust control: Guaranteeing stability while improving performance", in *International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2014.

[67] ——, "Safe and robust learning control with Gaussian processes", in *European Control Conference (ECC)*, 2015, pp. 2496–2501.

[68] F. Berkenkamp, R. Moriconi, A. Schoellig, and A. Krause, "Safe learning of regions of attraction for uncertain, nonlinear systems with Gaussian processes", *arXiv preprint arXiv:1603.04915*, 2016.

[69] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause, "Safe model-based reinforcement learning with stability guarantees", in *Advances in Neural Information Processing Systems (NeurIPS)*, Curran Associates, Inc., 2017, pp. 908–918.

[70] D. Nguyen-Tuong, M. Seeger, and J. Peters, "Real-time local GP model learning", in *From Motor Learning to Interaction Learning in Robots*, Springer, 2010, pp. 193–207.

[71] D. Nguyen-Tuong and J. Peters, "Local Gaussian process regression for real-time model-based robot control", in *International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2008, pp. 380–385.

[72] D. Nguyen-Tuong and J. Peters, "Learning robot dynamics for computed torque control using local Gaussian processes regression", in *ECSIS Symposium on Learning and Adaptive Behaviors for Robotic Systems*, IEEE, 2008, pp. 59–64.

[73] D. Nguyen-Tuong, M. Seeger, and J. Peters, "Model learning with local Gaussian process regression", *Advanced Robotics*, vol. 23, no. 15, pp. 2015–2034, 2009.

[74] D. Nguyen-Tuong, J. R. Peters, and M. Seeger, "Local Gaussian process regression for real time online model learning", in *Advances in Neural Information Processing Systems (NeurIPS)*, Curran Associates, Inc., 2009, pp. 1193–1200.

[75] D. Nguyen-Tuong and J. Peters, "Incremental online sparsification for model learning in real-time robot control", *Neurocomputing*, vol. 74, no. 11, pp. 1859–1867, 2011.

[76] J. Schreiter, P. Englert, D. Nguyen-Tuong, and M. Toussaint, "Sparse Gaussian process regression for compliant, real-time robot control", in *International Conference on Robotics and Automation (ICRA)*, IEEE, 2015, pp. 2586–2591.

[77] S. Park, K. shabbir Mustafa, and K. Shimada, "Learning-based robot control with localized sparse online Gaussian process", in *International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2013, pp. 1202–1207.

[78] ——, "Learning based robot control with sequential Gaussian process", in *Workshop on Robotic Intelligence in Informationally Structured Space (RiiSS)*, 2013, pp. 120–127.

[79] D. Nguyen-Tuong and J. Peters, "Model learning for robot control: A survey", *Cognitive processing*, vol. 12, no. 4, pp. 319–340, 2011.

[80] R. Camoriano, S. Traversaro, L. Rosasco, G. Metta, and F. Nori, "Incremental semiparametric inverse dynamics learning", in *International Conference on Robotics and Automation (ICRA)*, IEEE, 2016, pp. 544–550.

[81] T. Beckers, J. Umlauft, and S. Hirche, "Stable model-based control with Gaussian process regression for robot manipulators", in *World Congress of the International Federation of Automatic Control (IFAC)*, vol. 50, Toulouse, France: Elsevier, 2017, pp. 3877–3884.

[82] N. T. Alberto, M. Mistry, and F. Stulp, "Computed torque control with variable gains through Gaussian process", in *International Conference on Humanoid Robots (Humanoids)*, IEEE, 2014, pp. 212–217.

[83] T. Beckers, J. Umlauft, D. Kulic, and S. Hirche, "Stable Gaussian process based tracking control of Lagrangian systems", in *Conference on Decision and Control (CDC)*, IEEE, 2017, pp. 5180–5185.

[84] T. Beckers, D. Kulić, and S. Hirche, "Stable Gaussian process based tracking control of Euler-Lagrange systems", *Automatica*, vol. 23, no. 103, pp. 390–397, 2019.

[85] Y. Engel, P. Szabo, and D. Volkinshtein, "Learning to control an octopus arm with Gaussian process temporal difference methods", in *Advances in Neural Information Processing Systems (NeurIPS)*, ser. NIPS'05, Vancouver, British Columbia, Canada: Curran Associates, Inc., 2005, pp. 347–354.

[86] B. Likar and J. Kocijan, "Predictive control of a gas–liquid separation plant based on a Gaussian process model", *Computers & Chemical Engineering*, vol. 31, no. 3, pp. 142–152, 2007.

[87] A. Grancharova and J. Kocijan, "Explicit stochastic model predictive control of gas–liquid separator based on Gaussian process model", in *International Conference on Automatics and Informatics*, vol. 188, Sofia, Bulgaria, 2011, pp. 85–88.

[88] J. Kocijan and B. Likar, "Gas–liquid separator modelling and simulation with Gaussian-process models", *Simulation Modelling Practice and Theory*, vol. 16, no. 8, pp. 910–922, 2008.

[89] A. Grancharova, J. Kocijan, and T. A. Johansen, "Explicit stochastic predictive control of combustion plants based on Gaussian process models", *Automatica*, vol. 44, no. 6, pp. 1621–1631, 2008.

[90] M. de Paula and E. Martinez, "Probabilistic optimal control of blood glucose under uncertainty", in *22nd European Symposium on Computer Aided Process Engineering*, ser. Computer Aided Chemical Engineering, vol. 30, Elsevier, 2012, pp. 1357–1361.

[91] N. Qi, Q. Sun, K. Sun, X. Liu, F. Wu, and C. Liu, "Approximate dynamic programming based on Gaussian process regression for the perimeter patrol optimization problem", in *International Conference on Mechatronics and Control (ICMC)*, 2014, pp. 1750–1754.

[92] M. P. Deisenroth, R. Calandra, A. Seyfarth, and J. Peters, "Toward fast policy search for learning legged locomotion", in *International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2012, pp. 1787–1792.

[93] O. M. Cliff, T. Sildomar, and Monteiro, "Evaluating techniques for learning a feedback controller for low-cost manipulators", in *International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2013, pp. 704–709.

[94] L. Csato and M. Opper, "Sparse online Gaussian processes", *Neural computation*, vol. 14, no. 3, pp. 641–668, 2002.

[95] N. Lawrence, M. Seeger, and R. Herbrich, "Fast sparse Gaussian process methods: The informative vector machine", in *Advances in Neural Information Processing Systems (NeurIPS)*, Curran Associates, Inc., 2003, pp. 609–616.

[96] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs", in *Advances in Neural Information Processing Systems (NeurIPS)*, Curran Associates, Inc., 2005, pp. 1257–1264.

[97] M. Seeger, C. Williams, and N. Lawrence, "Fast forward selection to speed up sparse Gaussian process regression", in *Artificial Intelligence and Statistics*, 2003.

[98] M. Lazaro-Gredilla, J. Quinonero-Candela, C. E. Rasmussen, and A. R. Figueiras-Vidal, "Sparse spectrum Gaussian process regression", *Journal of Machine Learning Research*, vol. 11, no. Jun, pp. 1865–1881, 2010.

[99] J. Quinonero-Candela and C. E. Rasmussen, "A unifying view of sparse approximate Gaussian process regression", *The Journal of Machine Learning Research*, vol. 6, pp. 1939–1959, 2005.

[100] E. Snelson and Z. Ghahramani, "Local and global sparse Gaussian process approximations", in *International Conference on Artificial Intelligence and Statistics*, 2007, pp. 524–531.

[101] S. Choi, K. Lee, and S. Oh, "Robust learning from demonstration using leveraged Gaussian processes and sparse-constrained optimization", in *International Conference on Robotics and Automation (ICRA)*, IEEE, 2016.

[102] Y. Fanger, J. Umlauft, and S. Hirche, "Gaussian processes for dynamic movement primitives with application in knowledge-based cooperation", in *International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2016, pp. 3913–3919.

[103] A. Lederer, J. Umlauft, and S. Hirche, "Uniform error bounds for Gaussian process regression with application to safe control", in *Advances in Neural Information Processing Systems (NeurIPS)*, Curran Associates, Inc., 2019, pp. 659–669.

[104] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Information-theoretic regret bounds for Gaussian process optimization in the bandit setting", *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 3250–3265, 2012.

[105] D. E. Stewart, *Dynamics with Inequalities: impacts and hard constraints*. SIAM, 2011, vol. 59.

[106] R. Frigola-Alcade, "Bayesian time series learning with Gaussian processes", PhD thesis, University of Cambridge, 2015.

[107] J. Umlauft, T. Beckers, and S. Hirche, "A scenario-based optimal control approach for Gaussian process state space models", in *European Control Conference (ECC)*, 2018, pp. 1386–1392.

[108] D. H. Wolpert, "The supervised learning no-free-lunch theorems", in *Soft Computing and Industry*, Springer, 2002, pp. 25–42.

[109] F. Vivarelli, "Studies on the generalisation of Gaussian processes and Bayesian neural networks", PhD thesis, Aston University, Birmingham, UK, 1998.

[110] A. Lederer, J. Umlauft, and S. Hirche, "Posterior variance analysis of Gaussian processes", *arXiv preprint: arXiv:1906.01404*, 2019.

[111] H. H. Asada, F. Wu, A. Girard, and M. Mayalu, "A data-driven approach to precise linearization of nonlinear dynamical systems in augmented latent space", in *American Control Conference (ACC)*, IEEE, 2016, pp. 7–1844.

[112] J. M. Maciejowski, "Guaranteed stability with subspace methods", *Systems & Control Letters*, vol. 26, no. 2, pp. 153–156, 1995.

[113] B. Boots, G. J. Gordon, and S. M. Siddiqi, "A constraint generation approach to learning stable linear dynamical systems", in *Advances in Neural Information Processing Systems (NeurIPS)*, Curran Associates, Inc., 2008, pp. 1329–1336.

[114] P. Van Overschee and B. L. De Moor, *Subspace identification for linear systems: Theory—Implementation—Applications*. Springer Science & Business Media, 2012.

[115] W. Favoreel, B. De Moor, and P. Van Overschee, "Subspace state space system identification for industrial processes", *Journal of process control*, vol. 10, no. 2, pp. 149–155, 2000.

[116] D. L. Phillips, "A technique for the numerical solution of certain integral equations of the first kind", *Journal of the ACM (JACM)*, vol. 9, no. 1, pp. 84–97, 1962.

[117] H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of inverse problems*. Springer Science & Business Media, 1996, vol. 375.

[118] A. Kirsch, *An introduction to the mathematical theory of inverse problems*. Springer Science & Business Media, 2011, vol. 120.

[119] G. Pillonetto and G. De Nicolao, "A new kernel-based approach for linear system identification", *Automatica*, vol. 46, no. 1, pp. 81–93, 2010.

[120] G. Pillonetto, F. Dinuzzo, T. Chen, G. De 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.

[121] O. Nelles, *Nonlinear system identification: From classical approaches to neural networks and fuzzy models*. Springer Science & Business Media, 2013.

[122] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models", in *Advances in Neural Information Processing Systems (NeurIPS)*, Curran Associates, Inc., 2005, pp. 1441–1448.

[123] S. M. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with Gaussian mixture models", *IEEE Transactions on Robotics (T-RO)*, vol. 27, no. 5, pp. 943–957, 2011.

[124] J. R. Medina and A. Billard, "Learning stable task sequences from demonstration with linear parameter varying systems and hidden markov models", in *Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 78, PMLR, 2017, pp. 175–184.

[125] S. M. Khansari-Zadeh and A. Billard, "Learning control Lyapunov function to ensure stability of dynamical system-based robot reaching motions", *Robotics and Autonomous Systems*, vol. 62, no. 6, pp. 752–765, 2014.

[126] C. Blocher, M. Saveriano, and D. Lee, "Learning stable dynamical systems using contraction theory", in *International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, IEEE, 2017, pp. 124–129.

[127] K. Neumann and J. J. Steil, "Learning robot motions with stable dynamical systems under diffeomorphic transformations", *Robotics and Autonomous Systems*, vol. 70, pp. 1–15, 2015.

[128] H. J. Kushner, *Introduction to stochastic control*. Holt, Rinehart and Winston New York, 1971.

[129] J. R. Medina, L. Dongheui, and S. Hirche, "Risk-sensitive optimal feedback control for haptic assistance", in *International Conference on Robotics and Automation (ICRA)*, IEEE, 2012, pp. 1025–1031.

[130] E. Theodorou, J. Buchli, and S. Schaal, "A generalized path integral control approach to reinforcement learning", *Journal of Machine Learning Research*, vol. 11, no. Nov, pp. 3137–3181, 2010.

[131] H. K. Khalil and J. W. Grizzle, *Nonlinear systems*. Prentice hall New Jersey, 1996, vol. 3.

[132] R. Schaback, "Improved error bounds for scattered data interpolation by radial basis functions", *Mathematics of Computation*, pp. 201–216, 1999.

[133] M. Kanagawa, P. Hennig, D. Sejdinovic, and B. K. Sriperumbudur, "Gaussian processes and kernel methods: A review on connections and equivalences", *arXiv preprint arXiv:1807.02582*, 2018.

[134] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[135] Y. Li, W. Zhang, and X. Liu, "Stability of nonlinear stochastic discrete-time systems", *Journal of Applied Mathematics*, vol. 2013, 2013.

[136] K. Triantafyllopoulos, *Moments and cumulants of the multivariate real and complex Gaussian distributions*, University Bristol, 2002.

[137] C. K. I. Williams and F. Vivarelli, "Upper and lower bounds on the learning curve for Gaussian processes", *Machine Learning*, vol. 40, no. 1, pp. 77–102, 2000.

[138] Wolfram|Alpha, *Solution based on the analytic continuation of the product log function*, Online, 2019.

[139] V. Temlyakov, "A remark on covering", *arXiv preprint arXiv:1301.3043*, 2013.

[140] J. Gorski, F. Pfeuffer, and K. Klamroth, "Biconvex sets and optimization with biconvex functions: A survey and extensions", *Mathematical Methods of Operations Research*, vol. 66, no. 3, pp. 373–407, 2007.

[141] A. Papachristodoulou and S. Prajna, "On the construction of Lyapunov functions using the sum of squares decomposition", in *Conference on Decision and Control (CDC)*, vol. 3, IEEE, 2002, pp. 3482–3487.

[142] P. A. Parrilo, "Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization", PhD thesis, California Institute of Technology, 2000.

[143] C. A. Rogers, "Covering a sphere with spheres", *Mathematika*, vol. 10, no. 2, pp. 157–164, 1963.

[144] I. Dumer, "Covering spheres and balls with smaller balls", in *International Symposium on Information Theory*, IEEE, 2006, pp. 992–996.

[145] A. Geist and S. Trimpe, "Learning constrained dynamics with Gauss' principle adhering Gaussian processes", in *Conference on Learning for Dynamics and Control (L4DC)*, ser. Proceedings of Machine Learning Research, vol. 120, The Cloud: PMLR, 2020, pp. 225–234.

[146] W. Xiao, A. Lederer, and S. Hirche, "Learning stable nonparametric dynamical systems with Gaussian process regression", in *IFAC World Congress*, Berlin, 2020.

[147] P. A. Ioannou and J. Sun, *Robust adaptive control*. PTR Prentice-Hall Upper Saddle River, NJ, 1996, vol. 1.

[148] K. Zhou and J. C. Doyle, *Essentials of Robust Control*. Prentice hall Upper Saddle River, NJ, 1998, vol. 104.

[149] R. Khasminskii, *Stochastic stability of differential equations*. Springer Science & Business Media, 2011, vol. 66.

[150] K. J. Astrom, *Introduction to Stochastic Control Theory*. Courier Corporation, 2012.

[151] E. Theodorou, Y. Tassa, and E. Todorov, "Stochastic differential dynamic programming", in *American Control Conference (ACC)*, IEEE, 2010, pp. 1125–1132.

[152] J. R. Medina, D. Sieber, and S. Hirche, "Risk-sensitive interaction control in uncertain manipulation tasks", in *International Conference on Robotics and Automation (ICRA)*, IEEE, 2013, pp. 502–507.

[153] J. R. Medina and S. Hirche, "Uncertainty-dependent optimal control for robot control considering high-order cost statistics", in *International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2015, pp. 3995–4002.

[154] A. Mchutchon and C. E. Rasmussen, "Gaussian process training with input noise", in *Advances in Neural Information Processing Systems (NeurIPS)*, Curran Associates, Inc., 2011, pp. 1341–1349.

[155] D. P. Bertsekas, *Dynamic programming and optimal control*. Athena scientific Belmont, MA, 1995, vol. 1.

[156] I. M. Mitchell and S. Sastry, "Continuous path planning with multiple constraints", in *Conference on Decision and Control (CDC)*, vol. 5, IEEE, 2003, pp. 5502–5507.

[157] M. G. Crandall and P.-L. Lions, "Viscosity solutions of Hamilton-Jacobi equations", *Transactions of the American Mathematical Society*, vol. 277, no. 1, pp. 1–42, 1983.

[158] J. A. Sethian, "Fast marching methods", *SIAM review*, vol. 41, no. 2, pp. 199–235, 1999.

[159] K. J. Åström and B. Wittenmark, *Adaptive control*. Courier Corporation, 2013.

[160] M. Krstic, I. Kanellakopoulos, P. V. Kokotovic, *et al.*, *Nonlinear and adaptive control design*. Wiley New York, 1995, vol. 222.

[161] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control", *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, 2006.

[162] R. Chi, Y. Liu, Z. Hou, and S. Jin, "Data-driven terminal iterative learning control with high-order learning law for a class of non-linear discrete-time multiple-input–multiple output systems", *IET Control Theory & Applications*, vol. 9, no. 7, pp. 1075–1082, 2015.

[163] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. Cambridge, MA, USA: MIT Press, 1998.

[164] A. Molin and S. Hirche, "On the optimality of certainty equivalence for event-triggered control systems", *IEEE Transactions on Automatic Control (TAC)*, vol. 58, no. 2, pp. 470–474, 2013.

[165] M. Mazo and P. Tabuada, "Decentralized event-triggered control over wireless sensor/actuator networks", *IEEE Transactions on Automatic Control (TAC)*, vol. 56, no. 10, pp. 2456–2461, 2011.

[166] F. L. Lewis, A. Yesildirek, and K. Liu, "Multilayer neural-net robot controller with guaranteed tracking performance", *IEEE Transactions on Neural Networks*, vol. 7, no. 2, pp. 388–399, 1996.

[167] F. L. Lewis, K. Liu, and A. Yesildirek, "Neural net robot controller with guaranteed tracking performance", *IEEE Transactions on Neural Networks*, vol. 6, no. 3, pp. 703–715, 1995.

[168] R. M. Sanner and J.-J. E. Slotine, "Stable adaptive control and recursive identification using radial Gaussian networks", in *Conference on Decision and Control (CDC)*, IEEE, 1991, pp. 2116–2123.

[169] A. Yesildirak and F. L. Lewis, "Feedback linearization using neural networks", *Automatica*, vol. 31, no. 11, pp. 1659–1664, 1995.

[170] J. Umlauft, Y. Fanger, and S. Hirche, "Bayesian uncertainty modeling for programming by demonstration", in *International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 6428–6434.

[171] D. Duvenaud, "Automatic model construction with Gaussian processes", PhD thesis, Computational and Biological Learning Laboratory, University of Cambridge, 2014.

[172] M. W. Seeger, S. M. Kakade, and D. P. Foster, "Information consistency of nonparametric Gaussian process methods", *IEEE Transactions on Information Theory*, vol. 54, no. 5, pp. 2376–2382, 2008.

[173] J. Sherman and W. J. Morrison, "Adjustment of an inverse matrix corresponding to a change in one element of a given matrix", *Ann. Math. Statist.*, vol. 21, no. 1, pp. 124–127, 1950.

[174] D. Liberzon, *Switching in systems and control*. Springer Science & Business Media, 2012.

[175] J.-J. E. Slotine and J. Karl Hedrick, "Robust input-output feedback linearization", *International Journal of Control*, vol. 57, no. 5, pp. 1133–1139, 1993.

[176] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks", *IEEE Transactions on Automatic Control (TAC)*, vol. 52, no. 9, pp. 1680–1685, 2007.

[177] Wolfram|Alpha, *Solution first-order nonlinear ordinary differential equation*, Online, 2018.

[178] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies", *Journal of Machine Learning Research*, vol. 9, no. Feb, pp. 235–284, 2008.

[179] C.-W. Ko, J. Lee, and M. Queyranne, "An exact algorithm for maximum entropy sampling", *Operations Research*, vol. 43, no. 4, pp. 684–691, 1995.

[180] A. Lederer, A. Capone, J. Umlauft, and S. Hirche, "How training data impacts performance in learning-based control", *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 905–910, 2021.

[181] F. L. Lewis, S. Jagannathan, and A. Yesildirak, *Neural network control of robot manipulators and non-linear systems*. CRC Press, 1998.

[182] C. Vesper and M. J. Richardson, "Strategic communication and behavioral coupling in asymmetric joint action", *Experimental brain research*, vol. 232, no. 9, pp. 2945–2956, 2014.

[183] L. Consolini, F. Morbidi, D. Prattichizzo, and M. Tosques, "Leader–follower formation control of nonholonomic mobile robots with input constraints", *Automatica*, vol. 44, no. 5, pp. 1343–1349, 2008.

[184] A. Wilson and Z. Ghahramani, "Generalised Wishart processes", in *Conference on Uncertainty in Artificial Intelligence*, Corvallis, Oregon: AUAI Press, 2011, pp. 736–744.

[185] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Learning rhythmic movements by demonstration using nonlinear oscillators", in *International Conference on Intelligent Robots and Systems (IROS)*, vol. 1, IEEE, 2002, pp. 958–963.

[186] L. Pöhler, J. Umlauft, and S. Hirche, "Uncertainty-based human trajectory tracking with stable Gaussian process state space models", en, in *IFAC Conference on Cyber-Physical & Human Systems (CPHS)*, IFAC, Miami, 2018.

[187] J. Umlauft and S. Hirche, "Learning stable stochastic nonlinear dynamical systems", in *International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 70, International Convention Centre, Sydney, Australia: PMLR, 2017, pp. 3502–3510.

[188] H. Shen, "Meet the soft, cuddly robots of the future.", *Nature*, vol. 530, no. 7588, pp. 24–26, 2016.