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

Gaussian Process based Modeling and Control with Guarantees

Thomas Beckers

Vollständiger Abdruck der von der 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.-Ing. Klaus Diepold

Prüfende/-r der Dissertation:

1. Prof. Dr.-Ing. Sandra Hirche
2. Prof. George Pappas, Ph.D.

Die Dissertation wurde am 04.05.2020 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 28.11.2020 angenommen.

Preamble

This thesis summarizes the research results from my work at the Chair of Information-Oriented Control (ITR) at the Technical University of Munich. I am very grateful for the support and inspiration of numerous people during this time. First of all, I deeply thank my doctoral advisor and head of the Chair of Information-Oriented Control, Prof. Dr. Sandra Hirche, for her outstanding support and motivation in accomplishing the necessary scientific work that led to the present results. I had the great pleasure to be one of the first PhD students working on her ERC starting grant "Control based on Human Models", which allowed an extensive freedom in conducting my research. Without Sandra's dedication to excellence in research and the international scientific exchange with leading experts, initiated and coordinated by her, this thesis would never have been possible. My work profited deeply from the international network of collaborators, especially with Prof. Dr. Tomlin's group at UC Berkeley and Prof. Dr. Kulić at Monash University, and the thereof resulting fruitful research discussions.

Furthermore, I would like to thank my colleagues for the scientific discussions and the enjoyable time. In particular Jonas, who was very generous with his time to find solutions for any kind of problems and Hendrik, who used his sharp sense of humor to spice up the daily office life. In addition, many thanks to Martin and Fred for an uncountable number of funny evenings at ITR. I am also indebted to the administration and technical staff for providing support whenever needed. I would also like to thank my parents and my sister who laid the early foundations for my scientific career. Most important, I am very grateful for the enormous support and help of my wife Muriel. She might be the only person who really enjoys discussions about mathematical indices during lunch.

Acknowledgments

This work 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, and the BaCaTec grant "Stability and performance guarantees for data-driven control methods", grant agreement no. 9-[2018/1]

Abstract

In modern technologies such as autonomous vehicles and service robots, control engineering plays a crucial role for the overall performance and safety of the system. However, the control design becomes often very time-consuming or unfeasible due to the increasing complexity of recent technological advancements. The classical control approaches, which are based on models of the systems using first-order principles, are not satisfactory in the presence of complex dynamics, e.g., for highly nonlinear systems or interaction with prior unknown environment. Recent findings in the area of computational intelligence and machine learning have shown that data-driven approaches lead to very promising results in a wide application range as they require only a minimal prior knowledge for the modeling of complex dynamics. Within the past two decades, Gaussian process (GP) models have been used increasingly as a data-driven technique due to many beneficial properties such as the bias-variance trade-off and the strong connection to Bayesian mathematics. However, the major drawback in data-driven approaches frequently manifests as unpredictable outcomes. Thus, guarantees about the stability and performance of the control loop are absent which is translated as compromised safety in control systems. As a consequence, the current application of GP models in control scenarios is limited to non-critical and low performance systems due to their unpredictable "blackbox" behavior.

In this thesis, we analyze the behavior of GP models and present their application in control scenarios with formal guarantees in three steps. First, we analyze the control related properties of GP dynamical models which heavily depend on the underlying kernel function. For GP state space models and nonlinear output models, conditions for the stability of the system are derived. As GP dynamical models generally lead to non-Markovian systems, we introduce approximations that achieve Markovian dynamics. For these approximations quantitative results for the stability and equilibrium points are derived to incorporate control theoretic prior knowledge into the GP model. Next, for the usage of GPs in model based control laws, we propose a Bayesian optimization based approach to select the kernel function such that the closed-loop performance is optimized. In case of a misspecified kernel, an upper bound for the model error is provided.

Last, a GP model based control law is presented which guarantees the safe control of Euler-Lagrange systems with unknown dynamics. These systems are omnipresent as they include most of electromechanical systems such as robots and electric circuits. A GP model is used for the feed-forward compensation of the unknown dynamics of the system. The gains of the feedback part are adapted based on the uncertainty of the learned model. Thus, the feedback gains are kept low as long as the learned model describes the true system with sufficient precision. We demonstrate how to select a suitable gain adaption law that incorporates the uncertainty of the model and guarantees a bounded tracking error. Additionally, a quantification of the maximum tracking error, based on the number of training samples, is derived.

In summary, the results of this thesis increase the understanding of GP models and allow to use this data-driven technique not only to improve the performance but also to guarantee the safety of control systems.

Zusammenfassung

In vielen modernen Systemen, wie autonomen Fahrzeugen oder Service Robotern, spielt die Regelungstechnik eine wichtige Rolle. Die Entwicklung eines geeigneten Reglers wird mit zunehmender Komplexität der Systeme jedoch schwieriger, da ein präzises Modell der Regelstrecke benötigt wird. Bisher wurde die Dynamik eines Systems häufig mittels Differenzialgleichungen basierend auf physikalischen Grundprinzipien hergeleitet. Auf Grund des benötigten Zeitaufwands ist dieser Ansatz für komplexe Systeme, wie zum Beispiel hochgradig nichtlineare Systeme, jedoch nicht mehr zufriedenstellend. Das Problem verstärkt sich noch durch die zunehmende Verschmelzung der Systeme mit unbekannten Umgebungen, wie zum Beispiel bei der Mensch-Roboter-Interaktion.

Datenbasierte Verfahren haben bei der Modellierung solcher Systeme überzeugende Ergebnisse erzielt und daher in den letzten Jahren signifikant an Bedeutung gewonnen. Vor allem Gauß-Prozess (GP) Modelle haben sich, auf Grund der starken Verbindung zur Bayesschen Statistik, als besonders vielversprechender Ansatz erwiesen. Hierbei werden die Trainingsdaten mit Hilfe einer Kernel-Funktion in einen hoch-dimensionalen Raum transformiert, wodurch eine flexible Regression ermöglich wird. Ein entscheidender Nachteil bei der Verwendung von datengetriebenen Modellen in dynamischen Systemen sind jedoch die fehlenden formalen Garantien bezüglich der Stabilität und Performanz des Regelkreises. Auch wenn bereits einige erfolgreiche Regelungen basierend auf datengetriebenen Modellen existieren, so sind diese daher nicht für sicherheitskritische oder hochperformante Systeme geeignet.

In dieser Arbeit analysieren wie das regelungstechnische Verhalten von GP Modellen und präsentieren die Anwendung in Regelungsverfahren mit formalen Garantien. Zunächst wird gezeigt, wie GP Modelle als dynamisches System eingesetzt werden können. Da dies zu nicht-Markov Systemen führt, stellen wir eine Markovische Näherung zur Verwendung der Modelle in Simulationen und modellbasierte Regelungsverfahren vor. Für verschiedene Arten von dynamischen GP Modellen wird die Stabilität und die Anzahl der Ruhelagen untersucht und nachgewiesen. Des Weiteren ermöglicht die Wahl des Kernels zusätzliches Vorwissen in das Modell zu integrieren. Für den Einsatz von GP Modellen in der Regelung entwickeln wir ein Verfahren, welches durch die Wahl des Kernels die Performanz des Regelkreises optimiert. Sollte der Kernel für das zu lernende System nicht geeignet sein, so wird eine obere Schranke für den Modellfehler berechnet.

Schlussendlich zeigen wir, wie GP Modelle zur Regelung von Euler-Lagrange Systeme mit unbekannter Dynamik eingesetzt werden können. Die unbekannte Dynamik kann durch externe, schwierig zu modellierende Kräfte wie den Menschen, Reibung oder die Manipulation von Flüssigkeiten entstehen. Für die Modellierung der unbekannten Dynamik mittels GP Modellen werden zunächst Trainingsdaten des Systems gesammelt. Das so erzeugte, datenbasierte Modell wird, zusammen mit eventuell vorhandenem Vorwissen über das Systemverhalten, als Vorsteuerung zur Kompensierung der unbekannten Dynamik genutzt. Um einen möglichst niedrigen Feedback Anteil im Regler zu erhalten, wird die Verstärkung des Feedbacks basierend auf der Unsicherheit des gelernten Modells angepasst. Dabei wird nicht nur die Stabilität des Regelkreises garantiert, sondern auch die Berechnung des maximalen Regelfehlers ermöglicht.

Die Resultate dieser Arbeit erzielen ein besseres Verständnis von GP Modellen und erlauben die Verwendung dieses datengetriebenen Verfahrens in der Regelung komplexer Systeme, um sowohl die Performanz zu verbessern, als auch die Sicherheit zu garantieren.

Contents

Notation

Acronyms

GP	Gaussian process
GPR	Gaussian process regression
GPDM	Gaussian process dynamical model
GP-SSM	Gaussian process state space model
GP-ASSM	Gaussian process approximated state space model
GP-NOE	Gaussian process nonlinear output error
GP-ANOE	Gaussian process approximated nonlinear output error
SVM	support vector machine
BO	Bayesian optimization
PD	proportional-derivative
MSPE	mean square prediction error
RMSE	root mean square error
RKHS	reproducing kernel Hilbert space
CTC	computed torque control
CTC-GPR	computed torque control with Gaussian process regression
CTC-SGP	computed torque control with stochastic Gaussian process model
EL	Euler-Lagrange

Mathematical Conventions

Sets and Spaces

$\mathcal{A}, \mathcal{B}$	sets
$\mathbb{N}$	set of natural numbers
$\mathbb{R}$	set of real numbers
$\mathbb{R}_{>0}$	set of positive real numbers
$\mathbb{R}_{\geq 0}$	set of non-negative real numbers
$\mathcal{C}^x$	set of x-times continuously differentiable functions
$\mathcal{A} \cup \mathcal{B}$	union of sets $\mathcal{A}$ and $\mathcal{B}$
$\mathcal{A} \cap \mathcal{B}$	intersection of sets $\mathcal{A}$ and $\mathcal{B}$
$\mathcal{A} \setminus \mathcal{B}$	set $\mathcal{A}$ without $\mathcal{B}$

Scalars, Vectors, Matrices and Functions

a, b, c	scalars (small letters)
$\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$	column vectors (bold small letters)
A, B, C	matrices (capital letters)
$I_n \in \mathbb{R}^{n \times n}$	identity matrix
$\boldsymbol{a}^\top \in \mathbb{R}^{1 \times n}, A^\top \in \mathbb{R}^{n \times m}$	transpose of $\boldsymbol{a} \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}$
$A^{-1} \in \mathbb{R}^{n \times n}$	inverse of $A^{-1} \in \mathbb{R}^{n \times n}$ with $A^{-1}A = AA^{-1} = I_n$
$A^+ = A^\top(AA^\top)^{-1} \in \mathbb{R}^{n \times m}$	Moore-Penrose pseudo inverse of $A \in \mathbb{R}^{m \times n}$ where $m < n, \operatorname{rank}(A) = m, AA^+ = I_m$
a_i	i-th element of vector $\boldsymbol{a}$
$\boldsymbol{a}_{-i}$	vector $\boldsymbol{a}$ without the i-th element
$A_{a:b,c:d}$	submatrix of A formed between the a-th and the b-th rows, and the c-th and the d-th column
$A_{:,:} = A$	index : indicates all rows and columns, respectively
$[\boldsymbol{a}, \boldsymbol{b}] \in \mathbb{R}^{n \times 2}$	horizontal concatenation of $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$
$[\boldsymbol{a}; \boldsymbol{b}] = [\boldsymbol{a}^\top, \boldsymbol{b}^\top]^\top \in \mathbb{R}^{2n}$	vertical concatenation of $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$

$\Vert \boldsymbol{x} \Vert_a$	a-norm of $\boldsymbol{x} \in \mathbb{R}^n$, without index $a = 2$
$\boldsymbol{a} \preceq \boldsymbol{b}$	component wise inequality with $a_i \leq b_i, \forall i$
$\Vert f \Vert_{\mathcal{H}}$	Hilbert space norm of function f
$\dot{f} = \frac{d}{dt_c} f$	time derivative of function f
$\frac{\partial}{\partial z} f$	partial derivative of function f with respect to z
$\nabla \boldsymbol{f}$	gradient of function $\boldsymbol{f}$
$\det(A)$	determinant of $A \in \mathbb{R}^{n \times n}$
$\mathrm{rank}(A)$	rank of $A \in \mathbb{R}^{n \times m}$
$\ker(A)$	null space of $A \in \mathbb{R}^{n \times m}$
$\mathrm{tr}(A)$	trace of $A \in \mathbb{R}^{n \times n}$
$\lambda(A)$	eigenvalue of $A \in \mathbb{R}^{n \times n}$
$\underline{\lambda}(A), \underline{\sigma}(A)$	smallest eigenvalue, singular value of $A \in \mathbb{R}^{n \times n}$
$\bar{\lambda}(A), \overline{\sigma}(A)$	largest eigenvalue, singular value of $A \in \mathbb{R}^{n \times n}$

Probability

P	probability measure
p	probability distribution
$\mathrm{P}(\mathcal{A})$	probability of $\mathcal{A}$
$\mathrm{P}(\mathcal{A} \vert \mathcal{B}) = \frac{\mathrm{P}(\mathcal{A} \cap \mathcal{B})}{\mathrm{P}(\mathcal{B})}$	conditional probability of $\mathcal{A}$ given $\mathcal{B}$
$\mathcal{N}(\boldsymbol{m}, \Sigma)$	Gaussian distribution with mean $\boldsymbol{m} \in \mathbb{R}^n$ and variance $\Sigma \in \mathbb{R}^{n \times n}$
$\mu(\cdot)$	mean
$\mathrm{var}(\cdot)$	variance
$\mathbb{E}(\cdot)$	expected value

Accents and Subscripts

$\hat{(\cdot)}$	estimate
$\tilde{(\cdot)}$	corrupted by noise
$\bar{(\cdot)}$	mean
$\check{(\cdot)}$	modified/extended
$\overline{(\cdot)}$	upper bound
$\underline{(\cdot)}$	lower bound
$\{\cdot\}$	sequence

Main Variables

$\boldsymbol{a}$	covariance vector times Gram matrix
$c, \boldsymbol{c}$	general constant
$c_{\mathbf{act}}(\cdot,\cdot)$	cost for current state and input
$\bar{c}_C$	upper bound of estimated Coriolis matrix
c_L	Lipschitz constant
$c_{\mathbf{tot}}(\cdot,\cdot)$	total cost functional for whole task
$d_{\mathrm{KL}}(\cdot,\cdot)$	Kullback-Leibler-divergence
$\boldsymbol{e} = \boldsymbol{q_d} - \boldsymbol{q}$	error of generalized coordinates
$\boldsymbol{f}$	state mapping
$\boldsymbol{f}_{\mathbf{det}}$	state mapping of a deterministic GP-SSM
$\boldsymbol{f}_{\mathbf{gen}}$	general function
$\boldsymbol{f}_{\mathbf{GP}}$	Gaussian process distribution
$\boldsymbol{f}_t$	time-dependent mean vector of GP-ASSMs
$\boldsymbol{g}(\boldsymbol{q})$	generalized gravity vector
$\underline{h}$	lower bound of estimated inertia matrix
$\bar{h}$	upper bound of estimated inertia matrix
$\boldsymbol{h}(\cdot)$	state mapping of GP-NOE models

$\boldsymbol{h}_{\mathbf{ex}}(\cdot,\cdot)$	storage function
$\boldsymbol{h}_t(\cdot,\cdot)$	time-dependent mean vector of GP-ANOE models
$k(\cdot,\cdot)$	kernel function
$\boldsymbol{k}(\cdot,\cdot)$	covariance vector function
$\underline{k}_d, \overline{k}_d$	lower / upper bound of feedback matrix function $K_d(\cdot)$
$\underline{k}_p, \overline{k}_p$	lower / upper bound of feedback matrix function $K_p(\cdot)$
$\underline{m}$	actual length of memory
$\overline{m}$	maximum length of memory
$m(\cdot)$	mean function of Gaussian process
$n_{\mathcal{D}}$	number of training points
n_i	number of discretization intervals
$n_{\mathbf{in}}$	number of input values for GP-NOE models
n_k	number of kernels
n_o	dimension of output of a kernel-based model
$n_{\mathbf{out}}$	number of output values for GP-NOE models
n_q	dimension of generalized coordinates
n_t	number of time-steps
n_u	dimension of system input
n_x	dimension of system state
$n_{y_{dat}}$	dimension of output data points
n_z	dimension of index set
n_γ	dimension of parameterization vector
n_ζ	dimension of concatenated output and input vector $\boldsymbol{\zeta}_t$
n_ξ	dimension of concatenated state and input vector $\boldsymbol{\xi}_t$
n_φ	number of hyperparamters
n_ϕ	dimension of the feature space
n_Φ	number of hyperparameter sets
$\boldsymbol{o}_{\mathbf{kbm}}(\cdot,\cdot,\cdot,\cdot)$	output function of a kernel-based model

$\boldsymbol{o}_t$	output of a kernel-based model
p	general degree
$\boldsymbol{q}$	generalized coordinates
$\boldsymbol{q_d}$	desired generalized coordinates
$\boldsymbol{q}_c$	concatenation $[\ddot{\boldsymbol{q}}_{\boldsymbol{d}}; \dot{\boldsymbol{q}}_{\boldsymbol{d}}; \ddot{\boldsymbol{q}}]$
$\overline{q}_d$	upper bound of desired generalized coordinates
$\boldsymbol{q}_q$	concatenation $[\ddot{\boldsymbol{q}}; \dot{\boldsymbol{q}}; \boldsymbol{q}]$
t	discrete time step
t_c	continuous time
t_s	sampling time
$\boldsymbol{u}$	continuous-time system input
$\boldsymbol{u}_{\mathbf{ctrl}}$	general control law
$\boldsymbol{u}_k$	discrete-time system input
$w, \boldsymbol{w}$	weighting value / vector
$\boldsymbol{x}$	continuous-time system state
$\boldsymbol{x}_{dat}$	input data point
$\boldsymbol{x}_t$	discrete-time system state
$\boldsymbol{x}_t^m$	state of GP-ASSMs with maximum length of memory $\overline{m}$
$\boldsymbol{y}$	continuous-time system output
$\boldsymbol{y}_{dat}$	output data point
$\boldsymbol{y}_t$	discrete-time system output
$\boldsymbol{y}_t^m$	output of GP-ANOE models with maximum length of memory $\overline{m}$
$\boldsymbol{z}$	helper variable
A	system matrix of linear system
$C(\dot{\boldsymbol{q}}, \boldsymbol{q})$	generalized Coriolis matrix
$F_t(\cdot, \cdot)$	time-dependent variance matrix of GP-ASSMs
$H(\boldsymbol{q})$	generalized inertia matrix
$H_t(\cdot, \cdot)$	time-dependent variance matrix of GP-ANOE models

$K(\cdot,\cdot)$	Gram matrix
$\check{K}_t$	extended Gram matrix with time-dependent size
$K_d, K_d(\cdot)$	feedback gain / adaptive feedback function for derivative term
$K_p, K_p(\cdot)$	feedback gain / adaptive feedback function for proportional term
P	diagonal matrix of lengthscales
V	Lyapunov function
X	input training matrix
$\check{X}_t$	extended input training matrix with time-dependent size
Y	output training matrix
$\check{Y}_t$	extended output training matrix with time-dependent size
$\mathfrak{I}$	integral operator
$\mathfrak{I}_n$	numerical integral operator
$\mathfrak{L}$	drift operator
$\mathfrak{P}$	backshift operator
$\mathcal{B}$	ball
$\mathcal{D}$	set of training data
$\mathcal{F}$	Hilbert space
$\mathcal{F}_\sigma$	σ-algebra
$\mathcal{H}$	reproducing kernel Hilbert space
$\mathcal{I}$	invariant set
$\mathcal{I}_n$	neighborhood set
$\mathcal{K}$	set of valid kernel functions
$\mathcal{L}(\dot{\boldsymbol{q}}, \boldsymbol{q})$	Lagrangian function
$\mathcal{T}(\dot{\boldsymbol{q}}, \boldsymbol{q})$	kinetic energy function
$\mathcal{V}(\boldsymbol{q})$	potential function
$\mathcal{X}$	state space
$\mathcal{X}^*$	set of equilibrium points
$\mathcal{Z}$	index set

$\boldsymbol{\alpha}$	weighting vector for kernel function
$\boldsymbol{\alpha}(i)$	weighting vector for i-th dimension
$\beta, \boldsymbol{\beta}$	variance scaling factor / vector
$\gamma_{\mathbf{max}}, \boldsymbol{\gamma}_{\mathbf{max}},$	maximum information gain / vector
$\boldsymbol{\gamma}_y$	parameterization vector for output mapping
δ	probability for model error
δ_m	probability for multi-output model error
ε	positive number
$\boldsymbol{\zeta}_t$	concatenation of past outputs and inputs vector
$\nu, \nu_{\mathbf{brw}}$	Gaussian distributed noise / Brownian noise
$\boldsymbol{\xi}_t$	concatenation of state vector and input vector
σ_n	standard deviation of Gaussian distributed noise
$\check{\boldsymbol{\tau}}$	residual dynamics
$\boldsymbol{\varphi}$	vector of hyperparameters
$\boldsymbol{\phi}(\cdot)$	feature map
Δ	model error
$\overline{\Delta}$	upper bound of model error
Λ_t^m	matrix of outputs and inputs with maximum length of memory $\overline{m}$
Ξ_t^m	matrix of states and inputs with maximum length of memory $\overline{m}$
Φ	set of hyperparameters
$\Phi_{\mathbf{set}}$	set of sets of hyperparameters
Ω	compact set for model error
$\Omega_{\mathbf{ss}}$	sample space

Introduction

1

In many modern technologies such as autonomous vehicles and service robots, control engineering plays a crucial role for the overall performance and safety of the system. The task of a controller includes the monitoring of the actual output of a system, e.g., the position of a robot, and compareing it with a desired value. If needed, a control action is generated to bring the system to this value. The starting point is usually to derive a reliable model of the system. This is not only necessary for a subsequent controller synthesis but also for simulation and verification purposes. The classical approach is to derive a model of the system using first-order principles, for instance, by hand or exploiting software tools to obtain the system dynamics. However, these approaches are very time-consuming and challenging in the presence of complex dynamics [Pil+14], e.g., for highly nonlinear systems or interaction with prior unknown environment. These complex dynamics can be found in, but are not limited to, human motion prediciton [WFH08; Bre97], (soft-)robotics [Mog+16; NSP08; AMS14] and chemical processes [PGK13; Arc96].

In contrast, measured process data of the systems can easily be collected due to the advancements in information and storage technology [Loh12]. Thus, making full use of the information of the data to enhance the modeling and control of complex systems will be valuable for coping with rapid technological advancements [HW13]. The data-driven approaches were originated from the machine learning community where they have led to very convincing results, e.g., in pattern recognition [Lea05] and decision making [PF13]. These data-driven models require only a minimal prior knowledge for the modeling of complex dynamics, as established in recent findings in the area of computational intelligence and machine learning [Ras06].

However, the transfer of these models to the control community rises challenges, as well established analysis and synthesis tools are not suitable. Therefore, the stability and performance of control loops with data-driven models can not be analyzed and, thus, not be guaranteed. Furthermore, data-driven models can provide additional information about the uncertainty of the model which might be worth to exploit. Due to this lack of knowledge, the current application of data-driven approaches in control is often limited to non-critical and/or low performance systems.

1.1 Challenges in Data-driven based Control

The lack of interpretability and analyzability of data-driven models narrows their applicability in control. This "blackbox" behavior manifests as the absence of guarantees about the stability and performance of the control loop [Qiu02], which is translated as compromised safety in control systems. To overcome this issue, the understanding of data-driven models must be improved, and the relation to classical control concepts as well as new approaches has to be evaluated.

Challenge 1.1. *How can data-driven models be employed in control?*

In control theory, there exist many approaches for modeling dynamical systems based on parametric models. Nonlinear modeling techniques, such as nonlinear auto regressive models, are well-established and define a strategy how a dynamical system can be modeled. In contrast, data-driven models basically provide a way to exploit data for predictions. The step towards data-driven models for dynamical systems raises a question if and how these models can be used in similar settings as parametric models.

Challenge 1.2. *What are the control theoretic properties of data-driven models?*

Once the step from data-driven models to a dynamical systems is established, the analysis of the control relevant properties is necessary. The knowledge about boundedness, stability and equilibria, for instance, are not only necessary for a successful modeling with data-driven techniques but also for further application in model based control approaches. Furthermore, if specific control properties are related to specific classes of data-driven models, this information can be used as control relevant prior knowledge to the model. Therefore, investigations in control theoretic properties are key for guarantees in applications.

Challenge 1.3. *How can the remaining degrees-of-freedom be selected and what are the consequences of an incorrect selection?*

Data-driven models normally depend on additional functions and (hyper)parameters that must be determined. This is referred to *model selection* in the machine learning community. This selection classically depends on prior knowledge, as stated in the previous challenge, and the data itself. However, if the models are used in a control setting, additional information such as the desired task can become relevant. Furthermore, the stability and performance of the closed-loop are often from a higher interest than the quality of the model itself. Therefore, it might be worth performing the model selection based on control relevant requirements. Finally, the consequences of an incorrect selected models are vague and need further investigations.

Challenge 1.4. *How can a controller, which exploits the additional information of data-driven models, be designed?*

After the identification of the unknown system, the data-driven models should be used in control settings. There are some approaches available that use these models, typically by simply replacing a parametric model. However, data-driven models often contain much more information, e.g., about the uncertainty of the model, which is worth exploiting. The classical control approaches are not tailored for these kind of models and, thus, are not sufficient. Therefore, the design of a suitable control approach that exploits the full power of data-driven models is an open challenge.

Challenge 1.5. *Is there a way to provide stability and performance guarantees for control based on data-driven models?*

Stability and performance guarantees are necessary for the application of controllers to safety critical and/or high-performance systems. Therefore, a major step for data-driven control approaches is essentially to provide rigorous guarantees. However, general assurances are hard to obtain due to the enormous flexibility of these models which makes the analysis extremely different in contrast to parametric models. Therefore, giving guarantees in data-driven control is still an open problem, which is one of the key steps for the application to a broad range of safety critical systems.

1.2 Data-driven Models in Control

Many classical control approaches are based on physical dynamic models, which describe the underlying system behavior in a sufficiently accurate fashion. For complex dynamical systems, however, such descriptions are often extremely hard to obtain, or sometimes non-existent. Even though neural networks show remarkable results in many application areas, the complexity is still limited due to its finite size of parameters (number of neurons and layers) [Kec01]. Therefore, data-driven approaches are highly attractive to overcome this issue. Data-driven models are based on observations and measurements of the true system and need a minimum amount of prior knowledge of the system. In contrast to parametric models, which are defined by a finite dimensional parameter vector, the complexity of data-driven models is not limited as their convolution grows with the amount of training data [HW13]. For this reason, data-driven models are also called non-parametric models.
A very promising idea in terms of interpretability is to combine the advantages of data-driven models with a Bayesian perspective. Bayesian probability mathematics enables an efficient quantification of uncertainty of data-driven models. The Bayesian methodology is a probabilistic construct that allows new information to be combined with existing information: using Bayes' theorem, the existing knowledge is combined with information from the new data to update the knowledge [Rad96]. A promising Bayesian data-driven model for control is the Gaussian process (GP) model as it provides an analytic solution for the predicted mean and variance - the uncertainty measure - for a new test point based on collected training data [Ras06]. A GP model uses an underlying kernel functions to perform the prediction in a high-dimensional, implicit feature space. The characteristics of the kernel directly influence control related properties of the GP [SHS06; Aro50]. The application of GP models for the control of complex systems has already led to many convincing results, especially for cases with small data sets [Koc16; Koc+03a]. So far, GPs are treated as “blackbox” models in modeling and control as the prediction behavior of data-driven Bayesian models are only poorly explored. In particular, a holistic approach to determine the properties of GP models in control is missing.
In the last ten years, learning of dynamics has propelled many areas of control forward at a high pace – except for physical systems. This lag is crucial as learning physical models is critical for the safety of control applications involving reasoning of prior actions or planning of future actions, e.g., service robotics and industrial automation. Instead, most engineering approaches focus on classical off-the-shelf modeling as it ensures physical plausibility of the model and controller – at a high cost of precise measurements and time exposure. The plausible representations are often preferred, as these models guarantee to extrapolate well to unknown samples, while the outcome of learned models are often hard to predict. Recently, first approaches to encode control related structures for modeling dynamical systems are

presented [LRP19]. However, the existing methods focus on the integration of structures for modeling dynamical systems only and neglect control theoretic properties. Furthermore, most of the existing approaches are applied to deterministic data-driven models. Hence, the probabilistic nature of Bayesian models is not considered.

The control design based on data-driven models can improve the controller's quality [HW13]. Thus, it allows a more performant control of unknown systems by making use of the data-driven plant model prediction to derive the control input for the actual plant. Once a model of the plant is learned from the data, control strategies such as model predictive control or feedback linearization can be applied, as some types of data-driven models can be integrated in existing control structures [KM05; Uml+17]. First promising results on safety with these types of control settings have been presented in [Ber+17; Fis+19]. However, GPs provide much more information than classical models. The exploitation of such information would lead to considerable improvements for the control loop. In conclusion, preserving the safety of data-driven based control strategies is challenging due to the missing structure and uncertainty quantification. As consequence, existing data-driven based control approaches lack of formal stability and performance guarantees.

1.3 Main Contributions and Outline

The thesis addresses the recent challenges in GP model based control to allow data-driven methods in safety critical applications. For this purpose, we start with the embedding of GP models in dynamical systems. For different types of structure, the control relevant properties are analyzed and we propose how to insert prior knowledge about the system. As the kernel selection is one of the key parts in GP modeling, a task-based selection algorithm is proposed which surpasses the classical data-based selection. Furthermore, error bounds for the case of a misspecified kernel are derived. In the next step, a GP model based control law is introduced. The full probabilistic capabilities of the GP are exploited as the mean predication is used for a feed-forward compensation of the unknown dynamics and the model uncertainty to adapt the feedback gains. This trade-off between feed-forward and feedback control allows to keep the gains as low as possible without losing a desired performance. For this approach, we guarantee the passivity and the boundedness of the tracking error for the closed-loop. In addition, the performance is quantified based on the number and distribution of the training data.

In the following, we summarize the main contributions of this thesis addressing the presented challenges. At the beginning of each chapter, a brief review of the relevant related work and the open problems is provided.

Chapter 2: We start with the introduction of the required background knowledge about GP models. For this purpose, the general idea of GPs and the application in regression is presented. The relation between GPs, kernel-based models and reproducing Kernel Hilbert spaces is introduced. A set of the most common kernel functions is presented and the individual properties are highlighted. Furthermore, we show the general embedding of GP models into dynamical systems. Finally, the different ways on how model uncertainty can be transformed into error bounds are summarized. The formal descriptions are supported by several examples for an intuitive understanding.

Chapter 3: In this chapter, we address Challenges 1.1 and 1.2 which deal with the control related properties of data-driven models. We start with the analysis of certain GP dynamical systems, in particular state space systems and nonlinear output error models. It is shown that the GP based dynamical model is generally non-Markovian and we present approximations to obtain Markovian dynamics. The difference between the actual behavior and the approximations is analyzed in terms of prediction error. Furthermore, we introduce a comprehensive analysis for the approximated model which includes, among others, boundedness and number/distribution of the equilibria. The analysis is separated in a deterministic and stochastic point of view on GP dynamical models. The results presented in this chapter have been published in [BH16a; BH16b; BH20] and contributed to [Yam+20; Uml+20b; Led+20].

Chapter 4: In this chapter, we address Challenge 1.3 that states the model selection problem of GP models. Here, the model selection process is transferred from a classical data-based approach to a control focused method. For this purpose, the performance of the control loop is evaluated by means of a cost functional and the model selection is optimized based on the outcome. In case of a misspecified GP model, we propose an upper bound for the modeling error. The results presented in this chapter have been published in [BUH18; Bec+19].

Chapter 5: In this chapter, we address Challenges 1.4 and 1.5 which seek GP model based control approaches with formal guarantees. For this purpose, a GP model is integrated in an extended computed torque control scheme for Euler-Lagrange systems with unknown dynamics. We use the mean prediction of the GP to cancel the unknown dynamics in a feed-forward manner and the model fidelity to adapt the feedback gains. This approach is motivated to focus on feed-forward control as long as a sufficiently accurate model of the plant is learned. Otherwise, the feedback gains are automatically increased to preserve the performance. For this control setting, we provide not only boundedness guarantees but also the performance in terms of the maximum tracking error depending on the number and distribution of the training data. Finally, simulations and an experimental evaluation show the superiority of the proposed approach. The results presented in this chapter have been published in [BKH19; BH19; BH18; Bec+17; BUH17] and contributed to [Uml+20a; Cap+20; BCH20].

Gaussian Process models

2

A Gaussian process (GP) is a stochastic process that is in general a collection of random variables indexed by time or space. Its special property is that any finite collection of these variables follows a multivariate Gaussian distribution. Thus, the GP is a distribution over infinitely many variables and, therefore, a distribution over functions with a continuous domain. Consequently, it describes a probability distribution over an infinite dimensional vector space. For engineering applications, the GP has gained increasing attention as supervised machine learning technique, where it is used as prior probability distribution over functions in Bayesian inference. The inference of continuous variables leads to Gaussian process regression (GPR) where the prior GP model is updated with training data to obtain a posterior GP distribution. Historically, GPR was used for the prediction of time series, at first presented by Wiener and Kolmogorov in the 1940's. Afterwards, it became increasingly popular in geostatistics in the 1970's, where GPR is known as *kriging*. Recently, it came back in the area of machine learning [Rad96; WR96], especially boosted by the rapidly increasing computational power.
In this chapter, we present the necessary background information about GPs and GPR, mainly based on [Ras06], focusing on the application in control. We start with an introduction of GPs, explain the role of the underlying kernel function and show its relation to reproducing kernel Hilbert spaces. Afterwards, the embedding in dynamical systems and the interpretation of the model uncertainty as error bounds is presented. Several examples are included for an intuitive understanding in addition to the formal notation.

2.1 Gaussian Processes

Let $(\Omega_{\text{ss}}, \mathcal{F}_\sigma, P)$ be a probability space with the sample space Ω_{ss}, the corresponding σ-algebra $\mathcal{F}_\sigma$ and the probability measure P. The index set is given by $\mathcal{Z} \subseteq \mathbb{R}^{n_z}$ with positive integer n_z. Then, a function $f_{\text{GP}}(\boldsymbol{z}, \omega_{\text{ss}})$, which is a measurable function of $\omega_{\text{ss}} \in \Omega_{\text{ss}}$ with index $\boldsymbol{z} \in \mathcal{Z}$, is called a stochastic process. The function $f_{\text{GP}}(\boldsymbol{z}, \omega_{\text{ss}})$ is a random variable on Ω_{ss} if $\boldsymbol{z} \in \mathcal{Z}$ is specified. It is simplified written as $f_{\text{GP}}(\boldsymbol{z})$. A GP is a stochastic process which is fully described by a mean function $m\colon \mathcal{Z} \to \mathbb{R}$ and covariance function $k\colon \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ such that

$$f_{\text{GP}}(\boldsymbol{z}) \sim \mathcal{GP}\left(m(\boldsymbol{z}), k(\boldsymbol{z}, \boldsymbol{z}')\right) \tag{2.1}$$

$$\begin{aligned} m(\boldsymbol{z}) &= \mathrm{E}\left[f_{\text{GP}}(\boldsymbol{z})\right] \\ k(\boldsymbol{z}, \boldsymbol{z}') &= \mathrm{E}\left[\left(f_{\text{GP}}(\boldsymbol{z}) - m(\boldsymbol{z})\right)\left(f_{\text{GP}}(\boldsymbol{z}') - m(\boldsymbol{z}')\right)\right] \end{aligned} \tag{2.2}$$

with $\boldsymbol{z}, \boldsymbol{z}' \in \mathcal{Z}$. The covariance function is a measure for the correlation of two states $(\boldsymbol{z}, \boldsymbol{z}')$ and is called *kernel* in combination with GPs. Even though no analytic description of the probability density function of the GP exists in general, the interesting property is that any finite collection of its random variables $\{f_{\mathrm{GP}}(\boldsymbol{z}_1), \dots, f_{\mathrm{GP}}(\boldsymbol{z}_{n_{\mathrm{GP}}})\}$ follows a n_{GP}-dimensional multivariate Gaussian distribution. As a GP defines a distribution over functions, each realization is also a function over the index set $\mathcal{Z}$.

Example 2.1. A GP $f_{\mathrm{GP}}(t_c) \sim \mathcal{GP}\left(m(t_c), k(t_c, t_c')\right)$ with time $t_c \in \mathbb{R}_{\geq 0}$, where

$$m(t_c) = 1\mathrm{A}, \quad k(t_c, t_c') = \begin{cases} (0.1\mathrm{A})^2 & t_c = t_c' \\ (0\mathrm{A})^2 & t_c \neq t_c' \end{cases}$$

describes a time-dependent electric current signal with Gaussian white noise with a standard deviation of 0.1 A and a mean of 1 A.

2.1.1 Gaussian Process Regression

The GP can be utilized as prior probability distribution in Bayesian inference, which allows to perform function regression. Following the Bayesian methodology, new information is combined with existing information: using Bayes' theorem, the prior is combined with new data to obtain a posterior distribution. The new information is expressed as training data set $\mathcal{D} = \{X, Y\}$. It contains the input values $X = [\boldsymbol{x}_{\mathrm{dat}}^{\{1\}}, \boldsymbol{x}_{\mathrm{dat}}^{\{2\}}, \dots, \boldsymbol{x}_{\mathrm{dat}}^{\{n_{\mathcal{D}}\}}] \in \mathcal{Z}^{1 \times n_{\mathcal{D}}}$ and output values $Y = [\tilde{y}_{\mathrm{dat}}^{\{1\}}, \tilde{y}_{\mathrm{dat}}^{\{2\}}, \dots, \tilde{y}_{\mathrm{dat}}^{\{n_{\mathcal{D}}\}}]^\top \in \mathbb{R}^{n_{\mathcal{D}}}$, where

$$\tilde{y}_{\mathrm{dat}}^{\{i\}} = f_{\mathrm{GP}}(\boldsymbol{x}_{\mathrm{dat}}^{\{i\}}) + \nu \tag{2.3}$$

for all $i = 1, \dots, n_{\mathcal{D}}$. The output data might be corrupted by Gaussian noise $\nu \sim \mathcal{N}(0, \sigma_n^2)$.

Remark 2.1. *Note that we always use the standard notation X for the input training data and Y for the output training data throughout this thesis.*

As any finite subset of a GP follows a multivariate Gaussian distribution, we can write the joint distribution

$$\begin{bmatrix} Y \\ \\ f_{\mathrm{GP}}(\boldsymbol{z}^*) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(\boldsymbol{x}_{\mathrm{dat}}^{\{1\}}) \\ \vdots \\ m(\boldsymbol{x}_{\mathrm{dat}}^{\{n_{\mathcal{D}}\}}) \\ m(\boldsymbol{z}^*) \end{bmatrix}, \begin{bmatrix} K(X, X) + \sigma_n^2 I_{n_{\mathcal{D}}} & \boldsymbol{k}(\boldsymbol{z}^*, X) \\ \\ \boldsymbol{k}(\boldsymbol{z}^*, X)^\top & k(\boldsymbol{z}^*, \boldsymbol{z}^*) \end{bmatrix} \right) \tag{2.4}$$

for any arbitrary test point $\boldsymbol{z}^* \in \mathcal{Z}$. The function $m\colon \mathcal{Z} \to \mathbb{R}$ denotes the mean function. The matrix function $K\colon \mathcal{Z}^{1 \times n_{\mathcal{D}}} \times \mathcal{Z}^{1 \times n_{\mathcal{D}}} \to \mathbb{R}^{n_{\mathcal{D}} \times n_{\mathcal{D}}}$ is called the covariance or *Gram matrix* with

$$K_{j,l}(X, X) = k(X_{:,l}, X_{:,j}) \text{ for all } j, l \in \{1, \dots, n_{\mathcal{D}}\} \tag{2.5}$$

where each element of the matrix represents the covariance between two elements of the training data X. The expression $X_{:,l}$ denotes the l-th column of X. For notational simplification, we shorten $K(X, X)$ to K when necessary. The vector-valued kernel function $\boldsymbol{k}\colon \mathcal{Z} \times \mathcal{Z}^{1 \times n_{\mathcal{D}}} \to \mathbb{R}^{n_{\mathcal{D}}}$ calculates the covariance between the test input $\boldsymbol{z}^*$ and the input training data X, i.e.,

$$\boldsymbol{k}(\boldsymbol{z}^*, X) = [k(\boldsymbol{z}^*, X_{:,1}), \dots, k(\boldsymbol{z}^*, X_{:,n_{\mathcal{D}}})]^\top. \tag{2.6}$$

To obtain the posterior predictive distribution of $f_{\mathrm{GP}}(\boldsymbol{z}^*)$, we condition on the test point $\boldsymbol{z}^*$ and the training data set $\mathcal{D}$ given by

$$\mathrm{p}(f_{\mathrm{GP}}(\boldsymbol{z}^*)|\boldsymbol{z}^*, \mathcal{D}) = \frac{\mathrm{p}(f_{\mathrm{GP}}(\boldsymbol{z}^*), Y|X, \boldsymbol{z}^*)}{\mathrm{p}(Y|X)}. \tag{2.7}$$

Thus, the conditional posterior Gaussian distribution is defined by the mean and the variance

$$\begin{aligned}\mu(f_{\mathrm{GP}}(\boldsymbol{z}^*)|\boldsymbol{z}^*, \mathcal{D}) &= m(\boldsymbol{z}^*) + \boldsymbol{k}(\boldsymbol{z}^*, X)^\top (K + \sigma_n^2 I_{n_\mathcal{D}})^{-1} \left(Y - [m(X_{:,1}), \dots, m(X_{:,n_\mathcal{D}})]^\top\right) \\ \mathrm{var}(f_{\mathrm{GP}}(\boldsymbol{z}^*)|\boldsymbol{z}^*, \mathcal{D}) &= k(\boldsymbol{z}^*, \boldsymbol{z}^*) - \boldsymbol{k}(\boldsymbol{z}^*, X)^\top (K + \sigma_n^2 I_{n_\mathcal{D}})^{-1} \boldsymbol{k}(\boldsymbol{z}^*, X). \end{aligned} \tag{2.8}$$

A detailed derivation of the posterior mean and variance based on the joint distribution (2.4) can be found in appendix A.1. Analyzing (2.8) we can make the following observations:
i) The mean prediction can be written as

$$\mu(f_{\mathrm{GP}}(\boldsymbol{z}^*)|\boldsymbol{z}^*, \mathcal{D}) = m(\boldsymbol{z}^*) + \sum_{j=1}^{n_\mathcal{D}} \alpha_j k(\boldsymbol{z}^*, X_{:,j}) \tag{2.9}$$

with $\boldsymbol{\alpha} = (K+\sigma_n^2 I_{n_\mathcal{D}})^{-1} \left(Y - [m(X_{:,1}), \dots, m(X_{:,n_\mathcal{D}})]^\top\right) \in \mathbb{R}^{n_\mathcal{D}}$. That formulation highlights the data-driven characteristic of the GPR as the posterior mean is a sum of kernel functions and its number grows with the number $n_\mathcal{D}$ of training data.
ii) The variance does not depend on the observed data, but only on the inputs, which is a property of the Gaussian distribution. The variance is the difference between two terms: The first term $k(\boldsymbol{z}^*, \boldsymbol{z}^*)$ is simply the prior covariance from which a (positive) term is subtracted, representing the information the observations contain about the function. The variance expresses the uncertainty about the underlying function $f_{\mathrm{GP}}(\boldsymbol{z}^*)$ and does not include the measurement noise given by the variance σ_n^2. To prediction the uncertainty of the measured output, i.e. the output of the function $f_{\mathrm{GP}}(\boldsymbol{z}^*)$ plus the measurement noise, an additional noise term $\sigma_n^2 I_{n_\mathcal{D}}$ must be added to the variance in (2.8). Finally, (2.8) clearly shows the strong dependence of the posterior mean and variance on the kernel k that we will discuss in depth in Section 2.2.

Example 2.2. We assume a GP with zero mean and a kernel function given by

$$k(z, z') = 0.3679^2 \exp\left(-\frac{(z-z')^2}{2 \cdot 2.7183^2}\right)$$

as prior distribution. The training data set $\mathcal{D}$ is assumed to be

$$X = \begin{bmatrix} 1 & 3 & 6 & 10 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -0.3 & 0.3 & -0.2 \end{bmatrix}^\top,$$

where the output is corrupted by Gaussian noise with $\sigma_n = 0.0498$ standard deviation and the test point is assumed to be $z^* = 5$. According to (2.5) to (2.8) the Gram matrix $K(X, X)$ is calculated as

$$K(X, X) = \begin{bmatrix} 0.1378 & 0.1032 & 0.0249 & 0.0006 \\ 0.1032 & 0.1378 & 0.0736 & 0.0049 \\ 0.0249 & 0.0736 & 0.1378 & 0.0458 \\ 0.0006 & 0.0049 & 0.0458 & 0.1378 \end{bmatrix}$$

and the kernel vector $\boldsymbol{k}(z^*, X)$ and $k(z^*, z^*)$ are obtained to be

$$\begin{aligned}\boldsymbol{k}(z^*, X) &= \begin{bmatrix}0.0458 & 0.1032 & 0.1265 & 0.0249\end{bmatrix}\\ k(z^*, z^*) &= 0.1378.\end{aligned}$$

Finally, with (2.8), we compute the predicted mean and variance for $f_{\mathrm{GP}}(z^*)$

$$\mu(f_{\mathrm{GP}}(z^*)|z^*, \mathcal{D}) = 0.0278, \quad \mathrm{var}(f_{\mathrm{GP}}(z^*)|z^*, \mathcal{D}) = 0.0015,$$

which is equivalent to a 2σ-standard deviation of 0.0775. Figure 2.1 shows the prior distribution (left), the posterior distribution with two training points (black crosses) in the middle, and the posterior distribution given the full training set $\mathcal{D}$ (right). The solid red line is the mean function and the gray shaded area indicates the 2σ-standard deviation. Five realizations (dashed lines) visualize the character of the distribution over functions.

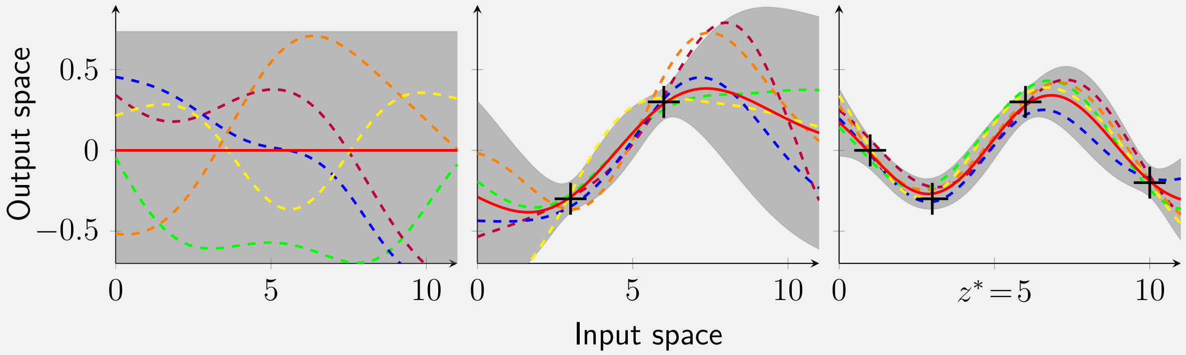


Figure 2.1: The prior distribution of a GP is updated with data that leads to the posterior distribution.

2.1.2 Marginal Variance

The computation of the variance with respect to a subset of elements of $\boldsymbol{z}^*$ can be done by marginalization. Assume $\boldsymbol{z}^* = [\boldsymbol{z}_1^*; \boldsymbol{z}_2^*]$ with $\boldsymbol{z}_1^* \in \mathbb{R}^{n_1}, \boldsymbol{z}_2^* \in \mathbb{R}^{n_2}$ and $\boldsymbol{z}^* \in \mathbb{R}^{n_z = n_1 + n_2}$. The marginal variance of the prediction based on $\boldsymbol{z}_1^*$ is given by

$$\mathrm{var}(f_{\mathrm{GP}}|\boldsymbol{z}_1^*, \mathcal{D}) = k(\boldsymbol{z}_1^*, \boldsymbol{z}_1^*) - \boldsymbol{k}(\boldsymbol{z}_1^*, X_{1:n_1,:})^\top (K(X_{1:n_1,:}, X_{1:n_1,:}) + I_{n_1}\sigma_n^2)^{-1} \boldsymbol{k}(\boldsymbol{z}_1^*, X_{1:n_1,:}). \tag{2.10}$$

The expression $X_{1:n_1,:}$ denotes the first n_1 rows of X such that $X = [X_{1:n_1,:}; X_{n_1+1:n_z,:}]$.

2.1.3 Multi-output Regression

So far, the GP regression allows functions with scalar outputs as in (2.8). For the extension to vector-valued outputs, multiple approaches exist: i) Extending the kernel to multivariate outputs [ÁRL12], ii) adding the output dimension as training data [Ber+17] or iii) using separated GPR for each output [Ras06]. While the first two approaches set a prior on the correlation between the output dimensions, the latter disregards a correlation without loss of generality. Following the approach in iii), the previous definition of the training set $\mathcal{D}$ is extended to a vector-valued output with

$$X = [\boldsymbol{x}_{\text{dat}}^{\{1\}}, \boldsymbol{x}_{\text{dat}}^{\{2\}}, \dots, \boldsymbol{x}_{\text{dat}}^{\{n_{\mathcal{D}}\}}] \in \mathcal{Z}^{1\times n_{\mathcal{D}}}, \quad Y = [\tilde{\boldsymbol{y}}_{\text{dat}}^{\{1\}}, \tilde{\boldsymbol{y}}_{\text{dat}}^{\{2\}}, \dots, \tilde{\boldsymbol{y}}_{\text{dat}}^{\{n_{\mathcal{D}}\}}]^\top \in \mathbb{R}^{n_{\mathcal{D}}\times n_{y\text{dat}}}, \tag{2.11}$$

where $n_{y\text{dat}} \in \mathbb{N}$ is the dimension of the output and the vector-valued GP is defined by

$$\boldsymbol{f}_{\text{GP}}(\boldsymbol{z}) \sim \begin{cases} \mathcal{GP}\Big(m^1(\boldsymbol{z}), k^1(\boldsymbol{z}, \boldsymbol{z}')\Big) \\ \quad\vdots \qquad \vdots \\ \mathcal{GP}\Big(m^{n_{y\text{dat}}}(\boldsymbol{z}), k^{n_{y\text{dat}}}(\boldsymbol{z}, \boldsymbol{z}')\Big) \end{cases} \tag{2.12}$$

$$\boldsymbol{m}(\boldsymbol{z}) := \Big[m^1(\boldsymbol{z}), \dots, m^{n_{y\text{dat}}}(\boldsymbol{z})\Big]^\top \tag{2.13}$$

Following (2.4) to (2.8), we obtain for the predicted mean and variance

$$\begin{aligned} \mu(f_{\text{GP},i}(\boldsymbol{z}^*)|\boldsymbol{z}^*, \mathcal{D}) &= m^i(\boldsymbol{z}^*) + \boldsymbol{k}^i(\boldsymbol{z}^*, X)^\top (K^i + \sigma_{n,i}^2 I_{n_{\mathcal{D}}})^{-1} \Big(Y_{:,i} - [m^i(X_{:,1}), \dots, m^i(X_{:,n_{\mathcal{D}}})]^\top\Big) \\ \text{var}(f_{\text{GP},i}(\boldsymbol{z}^*)|\boldsymbol{z}^*, \mathcal{D}) &= k^i(\boldsymbol{z}^*, \boldsymbol{z}^*) - \boldsymbol{k}^i(\boldsymbol{z}^*, X)^\top (K^i + \sigma_{n,i}^2 I_{n_{\mathcal{D}}})^{-1} \boldsymbol{k}^i(\boldsymbol{z}^*, X) \end{aligned} \tag{2.14}$$

for each output dimension $i \in \{1, \dots, n_{y\text{dat}}\}$ with respect to the kernels $k^1, \dots, k^{n_{y\text{dat}}}$. The variable $\sigma_{n,i}$ denotes the standard deviation of the Gaussian noise that corrupts the i-th dimension of the output measurements. The $n_{y\text{dat}}$ components of $\boldsymbol{f}_{\text{GP}}|\boldsymbol{z}^*, \mathcal{D}$ are combined into a multi-variable Gaussian distribution with

$$\begin{aligned} \boldsymbol{\mu}(\boldsymbol{f}_{\text{GP}}|\boldsymbol{z}^*, \mathcal{D}) &= [\mu(f_{\text{GP},1}|\boldsymbol{z}^*, \mathcal{D}), \dots, \mu(f_{\text{GP},n_{y\text{dat}}}|\boldsymbol{z}^*, \mathcal{D})]^\top \\ \Sigma(\boldsymbol{f}_{\text{GP}}|\boldsymbol{z}^*, \mathcal{D}) &= \text{diag}\Big(\text{var}(f_{\text{GP},1}|\boldsymbol{z}^*, \mathcal{D}), \dots, \text{var}(f_{\text{GP},n_{y\text{dat}}}|\boldsymbol{z}^*, \mathcal{D})\Big), \end{aligned} \tag{2.15}$$

where $\Sigma(\boldsymbol{f}_{\text{GP}}|\boldsymbol{z}^*, \mathcal{D})$ denotes the posterior variance matrix. This formulation allows to use a GP prior on vector-valued functions to perform predictions for test points $\boldsymbol{z}^*$. This approach treats each output dimension separately which is mostly sufficient and easy-to-handle. An alternative approach is to include the dimension as additional input, e.g., as in [Ber+17], with the benefit of a single GP at the price of loss of interpretability. For highly correlated output data, a multi-output kernel might be beneficial, see [ARL11].

Remark 2.2. *Without specific knowledge about a trend in the data, the prior mean functions $m^1, \dots, m^{n_{y\text{dat}}}$ are often set to zero, see [Ras06]. Therefore, we set the mean functions to zero for the remainder of the thesis if not stated otherwise.*

2.1.4 Kernel-based View

In Section 2.1.1, we target the GPR from a Bayesian perspective. However, for some applications of GPR a different point of view is beneficial; namely from the kernel perspective. In the following, we derive GPR from linear regression that is extended with a kernel transformation. In general, the prediction of parametric models is based on a parameter vector $\boldsymbol{w}$ which is typically learned using a set of training data points. In contrast, non-parametric models typically maintain at least a subset of the training data points in memory in order to make predictions for new data points. Many linear models can be transformed into a dual representation where the prediction is based on a linear combination of kernel functions. The idea is to transform the data points of a model to an often high-dimensional feature space where a linear regression can be applied to predict the model output, as depicted in Fig. 2.2. For a nonlinear feature map $\boldsymbol{\phi}\colon \mathcal{Z} \to \mathcal{F}$, where $\mathcal{F}$ is a $n_\phi \in \mathbb{N} \cup \{\infty\}$ dimensional Hilbert space, the kernel function is given by the inner product $k(\boldsymbol{z}, \boldsymbol{z}') = \langle \boldsymbol{\phi}(\boldsymbol{z}), \boldsymbol{\phi}(\boldsymbol{z}') \rangle, \forall \boldsymbol{z}, \boldsymbol{z}' \in \mathcal{Z}$. Thus, the kernel implicitly encodes the way the data points are transformed into a higher dimensional space. The formulation as inner product in a feature space allows to extend many standard regression methods. Also the GPR can be derived using a standard linear regression model

$$f_{\text{lin}}(\boldsymbol{z}) = \boldsymbol{z}^\top \boldsymbol{w}, \quad \tilde{y}_{\text{dat}}^{\{i\}} = f_{\text{GP}}(\boldsymbol{x}_{\text{dat}}^{\{i\}}) + \nu \tag{2.16}$$

where $\boldsymbol{z} \in \mathcal{Z}$ is the input vector, $\boldsymbol{w} \in \mathbb{R}^{n_z}$ the vector of weights with $n_z = \dim(\mathcal{Z})$ and $f_{\text{lin}}\colon \mathcal{Z} \to \mathbb{R}$ the unknown function. The observed value $\tilde{y}_{\text{dat}}^{\{i\}} \in \mathbb{R}$ for the input $\boldsymbol{x}_{\text{dat}}^{\{i\}} \in \mathcal{Z}$ is corrupted by Gaussian noise $\nu \sim \mathcal{N}(0, \sigma_n^2)$ for all $i = 1, \dots, n_{\mathcal{D}}$. The analysis of this model is analogous to the standard linear regression, i.e., we put a prior on the weights such that $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \Sigma_p)$ with $\Sigma_p \in \mathbb{R}^{n_z \times n_z}$. Based on $n_{\mathcal{D}}$ collected training data points as defined in Section 2.1.1, that leads to the well known linear Bayesian regression

$$\mathrm{p}(f_{\text{lin}}(\boldsymbol{z}^*)|\boldsymbol{z}^*, \mathcal{D}) = \mathcal{N}\Big(\underbrace{\frac{1}{\sigma_n^2} \boldsymbol{z}^{*\top} A_{\text{lin}}^{-1} X Y}_{\mu(f_{\text{lin}}(\boldsymbol{z}^*)|\boldsymbol{z}^*,\mathcal{D})},\ \underbrace{\boldsymbol{z}^{*\top} A_{\text{lin}}^{-1} \boldsymbol{z}^*}_{\text{var}(f_{\text{lin}}(\boldsymbol{z}^*)|\boldsymbol{z}^*,\mathcal{D})} \Big) \tag{2.17}$$

where $A_{\text{lin}} = \sigma_n^{-2} X X^\top + \Sigma_p^{-1}$. Now, using the feature map $\boldsymbol{\phi}(\boldsymbol{z})$ instead of $\boldsymbol{z}$ directly, leads to $f_{\text{GP}}(\boldsymbol{z}) = \boldsymbol{\phi}(\boldsymbol{z})^\top \check{\boldsymbol{w}}$ with $\check{\boldsymbol{w}} \sim \mathcal{N}(\boldsymbol{0}, \check{\Sigma}_p), \check{\Sigma}_p \in \mathbb{R}^{n_\phi \times n_\phi}$. As long as the projections are fixed functions, i.e., independent of the parameters w, the model is still linear in the parameters and, thus, analytically tractable. In particular, the Bayesian regression (2.17) with the mapping $\boldsymbol{\phi}(\boldsymbol{z})$ can be written as

$$(f_{\text{GP}}(\boldsymbol{z}^*)|\boldsymbol{z}^*, \mathcal{D}) \sim \mathcal{N}\left(\frac{1}{\sigma_n^2} \boldsymbol{\phi}(\boldsymbol{z}^*)^\top A_{\text{GP}}^{-1} \left[\boldsymbol{\phi}(X_{:,1}); \dots; \boldsymbol{\phi}(X_{:,n_{\mathcal{D}}})\right] Y, \boldsymbol{\phi}(\boldsymbol{z}^*)^\top A_{\text{GP}}^{-1} \boldsymbol{\phi}(\boldsymbol{z}^*) \right), \tag{2.18}$$

with the matrix $A_{\text{GP}} \in \mathbb{R}^{n_\phi \times n_\phi}$ given by

$$A_{\text{GP}} = \sigma_n^{-2} \left[\boldsymbol{\phi}(X_{:,1}); \dots; \boldsymbol{\phi}(X_{:,n_{\mathcal{D}}})\right] \left[\boldsymbol{\phi}(X_{:,1}); \dots; \boldsymbol{\phi}(X_{:,n_{\mathcal{D}}})\right]^\top + \check{\Sigma}_p^{-1}. \tag{2.19}$$

This equation can be simplified and rewritten to

$$(f_{\text{GP}}(\boldsymbol{z}^*)|\boldsymbol{z}^*, \mathcal{D}) \sim \mathcal{N}\Big(\underbrace{\boldsymbol{k}(\boldsymbol{z}^*, X)^\top K^{-1} Y}_{\mu(f_{\text{GP}}(\boldsymbol{z}^*)|\boldsymbol{z}^*,\mathcal{D})}, \underbrace{k(\boldsymbol{z}^*, \boldsymbol{z}^*) - \boldsymbol{k}(\boldsymbol{z}^*, X)^\top K^{-1} \boldsymbol{k}(\boldsymbol{z}^*, X)}_{\text{var}(f_{\text{GP}}(\boldsymbol{z}^*)|\boldsymbol{z}^*,\mathcal{D})} \Big), \tag{2.20}$$

with $k(\boldsymbol{z}, \boldsymbol{z}') = \boldsymbol{\phi}(\boldsymbol{z})^\top \check{\Sigma}_p \boldsymbol{\phi}(\boldsymbol{z}')$ that equals (2.8). The fact that in (2.20) the feature map $\boldsymbol{\phi}(\boldsymbol{z})$ is not needed is known as the *kernel trick*. This trick is also used in other kernel-based models, e.g., support vector machines (SVM), see [SC08] for more details.

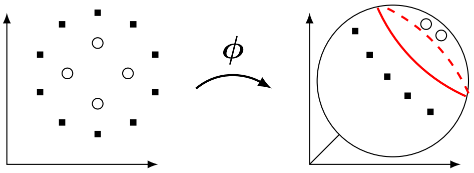


Figure 2.2: The mapping $\boldsymbol{\phi}$ transforms the data points into a feature space where linear regressors can be applied to predict the output.

2.1.5 Reproducing Kernel Hilbert Space

Even though a kernel neither uniquely defines the feature map nor the feature space, one can always construct a canonical feature space, namely the *reproducing kernel Hilbert space* (RKHS) given a certain kernel. After the introduction of the theory, illustrative examples for an intuitive understanding are presented. We will now formally present this construction procedure, starting with the concept of Hilbert spaces, following [BLG16]: A Hilbert space $\mathcal{F}$ represents all possible realizations of some class of functions, for example all functions of continuity degree i, denoted by $\mathcal{C}^i$. Moreover, a Hilbert space is a vector space such that any function $f_{\mathcal{F}} \in \mathcal{F}$ must have a non-negative norm, $\|f_{\mathcal{F}}\|_{\mathcal{F}} > 0$ for $f_{\mathcal{F}} \neq 0$. All functions $f_{\mathcal{F}}$ must additionally be equipped with an inner-product in $\mathcal{F}$. Simply speaking, a Hilbert space is an infinite dimensional vector space, where many operations behave like in the finite case. The properties of Hilbert spaces have been explored in great detail in literature, e.g., in [DM+05]. An extremely useful property of Hilbert spaces is that they are equivalent to an associated kernel function [Aro50]. This equivalence allows to simply define a kernel, instead of fully defining the associated vector space. Formally speaking, if a Hilbert space $\mathcal{H}$ is a RKHS, it will have a unique positive definite kernel $k\colon \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$, which spans the space $\mathcal{H}$.

Theorem 2.1 (Moore-Aronszajn [Aro50])**.** *Every positive definite kernel k is associated with a unique RKHS $\mathcal{H}$.*

Theorem 2.2 ([Aro50])**.** *Let $\mathcal{F}$ be a Hilbert space, $\mathcal{Z}$ a non-empty set and $\boldsymbol{\phi}\colon \mathcal{Z} \to \mathcal{F}$. Then, the inner product $\langle \boldsymbol{\phi}(\boldsymbol{z}), \boldsymbol{\phi}(\boldsymbol{z}')\rangle_{\mathcal{F}} := k(\boldsymbol{z}, \boldsymbol{z}')$ is positive definite.*

Importantly, any function $f_{\mathcal{H}}$ in $\mathcal{H}$ can be represented as a weighted linear sum of this kernel evaluated over the space $\mathcal{H}$, as

$$f_{\mathcal{H}}(\cdot) = \langle f_{\mathcal{H}}(\cdot), k(x, \cdot)\rangle_{\mathcal{H}} = \sum_{i=1}^{n_\phi} \alpha_i k\left(\boldsymbol{x}_{\text{dat}}^{\{i\}}, \cdot\right), \tag{2.21}$$

with $\alpha_i \in \mathbb{R}$ for all $i = \{1, \dots, n_\phi\}$, where $n_\phi \in \mathbb{N} \cup \{\infty\}$ is the dimension of the feature space $\mathcal{F}$. Thus, the RKHS is equipped with the inner-product

$$\langle f_{\mathcal{H}}, f'_{\mathcal{H}}\rangle_{\mathcal{H}} = \sum_{i=1}^{n_\phi}\sum_{j=1}^{n_\phi} \alpha_i \alpha'_j k(\boldsymbol{x}_{\text{dat}}^{\{i\}}, \boldsymbol{x}_{\text{dat}}^{\prime\{j\}}), \tag{2.22}$$

with $f'_{\mathcal{H}}(\cdot) = \sum_{j=1}^{n_\phi} \alpha'_j k\left(\boldsymbol{x}_{\text{dat}}^{\prime\{j\}}, \cdot\right) \in \mathcal{H}, \alpha'_j \in \mathbb{R}$. Now, the reproducing character manifests as

$$\forall \boldsymbol{z} \in \mathcal{Z}, \forall f_{\mathcal{H}} \in \mathcal{H},\ \langle f_{\mathcal{H}}, k(\boldsymbol{z}, \cdot)\rangle_{\mathcal{H}} = f_{\mathcal{H}}(\boldsymbol{z}),\ \text{in particular } k(\boldsymbol{z}, \boldsymbol{z}') = \langle k(\cdot, \boldsymbol{z}), k(\cdot, \boldsymbol{z}')\rangle_{\mathcal{H}}. \tag{2.23}$$

According to [SHS06], the RKHS is then defined as

$$\mathcal{H} = \{f_{\mathcal{H}}\colon \mathcal{Z} \to \mathbb{R} | \exists \boldsymbol{c} \in \mathcal{F}, f_{\mathcal{H}}(\boldsymbol{z}) = \langle \boldsymbol{c}, \boldsymbol{\phi}(\boldsymbol{z})\rangle_{\mathcal{F}}, \forall \boldsymbol{z} \in \mathcal{Z}\}, \tag{2.24}$$

where $\boldsymbol{\phi}(\boldsymbol{z})$ is the feature map constructing the kernel through $k(\boldsymbol{z}, \boldsymbol{z}') = \langle \boldsymbol{\phi}(\boldsymbol{z}), \boldsymbol{\phi}(\boldsymbol{z}')\rangle_{\mathcal{F}}$.

Example 2.3. We want to find the RKHS for the polynomial kernel with degree 2 that is given by

$$k(\boldsymbol{z},\boldsymbol{z}') = (\boldsymbol{z}^\top \boldsymbol{z}')^2 = (z_1 z_1')^2 + 2(z_1 z_1' z_2 z_2') + (z_2 z_2')^2.$$

for any $\boldsymbol{z},\boldsymbol{z}' \in \mathbb{R}^2$. First, we have to find a feature map $\boldsymbol{\phi}$ such that the kernel corresponds to the inner product $k(\boldsymbol{z},\boldsymbol{y}) = \langle \boldsymbol{\phi}(\boldsymbol{z}), \boldsymbol{\phi}(\boldsymbol{y})\rangle$. A possible candidate for the feature map is

$$\begin{aligned}\boldsymbol{\phi}(\boldsymbol{z}) &= \left[z_1^2, \sqrt{2}z_1 z_2, z_2^2\right]^\top, \text{ because}\\ \langle \boldsymbol{\phi}(\boldsymbol{z}), \boldsymbol{\phi}(\boldsymbol{z}')\rangle_{\mathbb{R}^3} &= \boldsymbol{\phi}(\boldsymbol{z})^\top \boldsymbol{\phi}(\boldsymbol{y}) = (z_1 z_1')^2 + 2(z_1 z_1' z_2 z_2') + (z_2 z_2')^2 = k(\boldsymbol{z},\boldsymbol{z}').\end{aligned}$$

We know that the RKHS contains all linear combinations of the form

$$\begin{aligned}f_{\mathcal{H}}(\boldsymbol{z}) &= \sum_{i=1}^{3} \alpha_i k\left(\boldsymbol{x}_{\text{dat}}^{\{i\}}, \boldsymbol{z}\right) = \sum_{i=1}^{3} \alpha_i \langle \boldsymbol{\phi}(\boldsymbol{z}'), \boldsymbol{\phi}(\boldsymbol{z})\rangle_{\mathbb{R}^3} = \sum_{i=1}^{3} \langle \boldsymbol{c}, \boldsymbol{\phi}(\boldsymbol{z})\rangle_{\mathbb{R}^3}\\ &= c_1 z_1^2 + c_2\sqrt{2}z_1 z_2 + c_3 z_2^2,\end{aligned}$$

with $\boldsymbol{\alpha}, \boldsymbol{c}, \boldsymbol{x}_{\text{dat}}^{\{i\}} \in \mathbb{R}^3$. Therefore, a possible candidate for the RKHS $\mathcal{H}$ is given by

$$\mathcal{H} = \left\{f_{\mathcal{H}} \colon \mathbb{R}^2 \to \mathbb{R} | f_{\mathcal{H}}(\boldsymbol{z}) = c_1 z_1^2 + c_2\sqrt{2}z_1 z_2 + c_3 z_2^2, \boldsymbol{c} \in \mathbb{R}^3\right\} \tag{2.25}$$

Next, it must be checked if the proposed Hilbert space is the related RKHS to the polynomial kernel with degree 2. This is achieved in two steps: i) Proving that the space is a Hilbert space and ii) confirming the reproducing property. First, we can easily proof that this is a Hilbert space rewriting $f_{\mathcal{H}}(\boldsymbol{z}) = \boldsymbol{z}^\top S \boldsymbol{z}$ with symmetric matrix $S \in \mathbb{R}^{2\times 2}$ and using the fact that $\mathcal{H}$ is euclidean and isomorphic to S. Second, the condition for an RKHS must be fulfilled, i.e., the reproducing property $f_{\mathcal{H}}(\boldsymbol{z}) = \langle f_{\mathcal{H}}(\cdot), k(\cdot,\boldsymbol{z})\rangle_{\mathcal{H}}$. Since we can write

$$\langle f_{\mathcal{H}}(\cdot), k(\cdot,\boldsymbol{z})\rangle_{\mathcal{H}} = \langle \boldsymbol{c}^\top \boldsymbol{\phi}(\cdot), k(\cdot,\boldsymbol{z})\rangle_{\mathcal{H}} = \sum_{i=1}^{3} c_i k(\cdot,\boldsymbol{z}) = \boldsymbol{c}^\top \boldsymbol{\phi}(\boldsymbol{z}) = f_{\mathcal{H}}(\boldsymbol{z}),$$

property (2.23) is fulfilled and, thus, $\mathcal{H}$ is the RKHS for the polynomial kernel with degree 2. Note that, even though the mapping $\boldsymbol{\phi}$ is not unique for the kernel k, the relation of k and the RKHS $\mathcal{H}$ is unique.

Given a function $f_{\mathcal{H}} \in \mathcal{H}$ defined by $n_{\mathcal{D}}$ observations, its RKHS norm is defined as

$$\|f_{\mathcal{H}}\|_{\mathcal{H}}^2 = \langle f_{\mathcal{H}}, f_{\mathcal{H}}\rangle_{\mathcal{H}} = \sum_{i=1}^{n_{\mathcal{D}}}\sum_{j=1}^{n_{\mathcal{D}}} \alpha_i \alpha_j k(\boldsymbol{x}_{\text{dat}}^{\{i\}}, \boldsymbol{x}_{\text{dat}}^{\prime\{j\}}) = \boldsymbol{\alpha}^\top K(X,X)\boldsymbol{\alpha}, \tag{2.26}$$

with $\boldsymbol{\alpha} \in \mathbb{R}^{n_{\mathcal{D}}}$ and $K(X,X)$ given by (2.5). We can also use the feature map such that

$$\|f_{\mathcal{H}}\|_{\mathcal{H}} = \inf\{\|\boldsymbol{c}\|_{\mathcal{F}} \colon \boldsymbol{c} \in \mathcal{F}, f_{\mathcal{H}}(\boldsymbol{z}) = \langle \boldsymbol{c}, \boldsymbol{\phi}(\boldsymbol{z})\rangle_{\mathcal{F}}, \forall \boldsymbol{z} \in \mathcal{Z}\}. \tag{2.27}$$

As there is a unique relation between the RKHS $\mathcal{H}$ and the kernel k, the norm $\|f_{\mathcal{H}}\|_{\mathcal{H}}$ can equivalently be written as $\|f_{\mathcal{H}}\|_k$. The norm of a function in the RKHS indicates how fast the function varies over $\mathcal{Z}$ with respect to the geometry defined by the kernel. Formally, it can be written as

$$\frac{|f_{\mathcal{H}}(\boldsymbol{z}) - f_{\mathcal{H}}(\boldsymbol{z}')|}{d(\boldsymbol{z}, \boldsymbol{z}')} \leq \|f_{\mathcal{H}}\|_{\mathcal{H}}, \tag{2.28}$$

with the distance $d(\boldsymbol{z}, \boldsymbol{z}')^2 = k(\boldsymbol{z}, \boldsymbol{z}) - 2k(\boldsymbol{z}, \boldsymbol{z}') + k(\boldsymbol{z}', \boldsymbol{z}')$. A function with finite RKHS norm is also element of the RKHS. A more detailed discussion about RKHS and norms is given in [Wah90].

Example 2.4. We want to find the RKHS norm of a function $f_{\mathcal{H}}$ that is an element of the RKHS of the polynomial kernel with degree 2 that is given by

$$k(\boldsymbol{z}, \boldsymbol{z}') = (\boldsymbol{z}^\top \boldsymbol{z}')^2 = (z_1 z_1')^2 + 2(z_1 z_1' z_2 z_2') + (z_2 z_2')^2.$$

Let the function be

$$f_{\mathcal{H}}(\boldsymbol{z}) = \sum_{i=1}^{3} \alpha_i k\left(\boldsymbol{x}_{\text{dat}}^{\{i\}}, \boldsymbol{z}\right), \text{ with} \tag{2.29}$$

$$\alpha_1 = 1,\ \alpha_2 = -2,\ \alpha_3 = 3 \tag{2.30}$$

$$\boldsymbol{x}_{\text{dat}}^{\{1\}} = [1, 1]^\top,\ \boldsymbol{x}_{\text{dat}}^{\{2\}} = [1, 2]^\top,\ \boldsymbol{x}_{\text{dat}}^{\{3\}} = [2, 1]^\top. \tag{2.31}$$

Hence, function (2.29) with (2.30) and (2.31) corresponds to

$$f_{\mathcal{H}}(\boldsymbol{z}) = 11z_1^2 + 6z_1 z_2 - 4z_2^2.$$

Now, we have two possibilities how to calculate the RKHS norm. First, the RKHS-norm of $f_{\mathcal{H}}$ is calculated using (2.26) by

$$\|f_{\mathcal{H}}\|_{\mathcal{H}}^2 = \boldsymbol{\alpha}^\top K(X, X)\boldsymbol{\alpha} = \begin{bmatrix} 1 & -2 & 3 \end{bmatrix} \begin{bmatrix} 4 & 9 & 9 \\ 9 & 25 & 16 \\ 9 & 16 & 25 \end{bmatrix} \begin{bmatrix} 1 \\ -2 \\ 3 \end{bmatrix} = 155$$

with $X = [\boldsymbol{x}_{\text{dat}}^{\{1\}}, \boldsymbol{x}_{\text{dat}}^{\{2\}}, \boldsymbol{x}_{\text{dat}}^{\{3\}}]$. Alternatively, we can use (2.27) that results in $\|f_{\mathcal{H}}\|_{\mathcal{H}} = \|\boldsymbol{c}\|$, where $\boldsymbol{c}$ is defined by (2.25). Thus, the norm is computed as

$$f_{\mathcal{H}}(\boldsymbol{z}) = 11z_1^2 + 6z_1 z_2 - 4z_2^2 \Rightarrow c_1 = 11,\ c_2 = \frac{6}{\sqrt{2}},\ c_3 = -4 \Rightarrow \|f_{\mathcal{H}}\|_{\mathcal{H}}^2 = 155.$$

Example 2.5. In this example, we visualize the meaning of the RKHS norm. Figure 2.3 shows different quadratic functions with the same RKHS norm (top left and top right), a smaller RKHS norm (bottom left) and a larger RKHS norm (bottom right). An identical norm indicates a similar variation of the functions, whereas a higher norm leads to a more varying function.

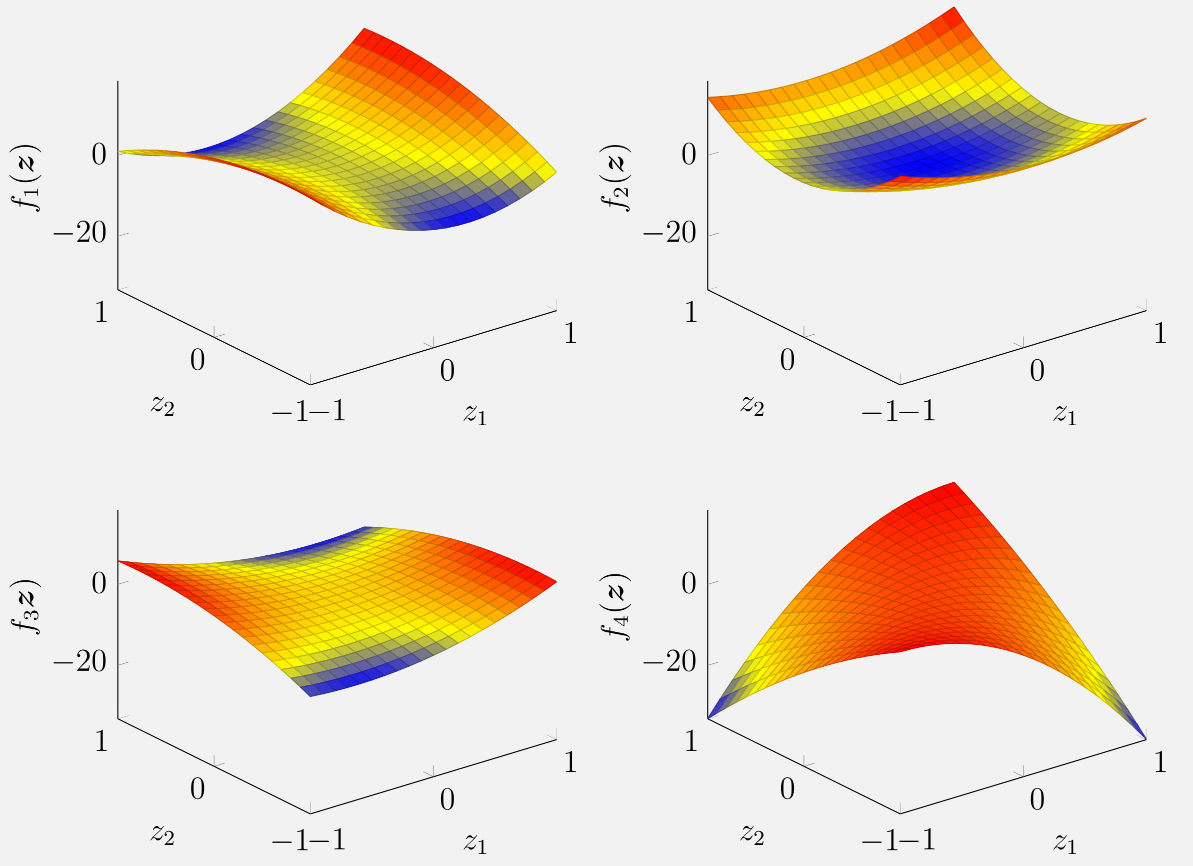


Figure 2.3: Functions with different RKHS-norms: $\|f_1\|^2_{\mathcal{H}} = \|f_2\|^2_{\mathcal{H}} = 4\|f_3\|^2_{\mathcal{H}} = \frac{1}{2}\|f_4\|^2_{\mathcal{H}}$.

In summary, we show the unique relation between the kernel and its RKHS. The reproducing property allows us to write the inner-product as a tractable function which implicitly defines a higher (or even infinite) feature dimensional space. The RKHS-norm of a function is a Lipschitz-like indicator based on the metric defined by the kernel. This view of the RKHS is related to the kernel trick in machine learning. In the next section, the RKHS-norm is exploited to determine the error between the prediction of GPR and the actual data-generating function.

2.1.6 Model Error

One of the most interesting properties of GPR is the uncertainty description encoded in the predicted variance. This uncertainty is beneficial to quantify the error between the actual underlying data generating process and the GPR. In this section, we assume that there is an unknown function $f_{\mathrm{uk}} \colon \mathbb{R}^{n_z} \to \mathbb{R}$ that generates the training data. In detail, the data set $\mathcal{D} = \{X, Y\}$ consists of

$$\begin{aligned} X &= [\boldsymbol{x}^{\{1\}}_{\mathrm{dat}}, \boldsymbol{x}^{\{2\}}_{\mathrm{dat}}, \dots, \boldsymbol{x}^{\{n_{\mathcal{D}}\}}_{\mathrm{dat}}] \in \mathbb{R}^{n_z \times n_{\mathcal{D}}} \\ Y &= [\tilde{y}^{\{1\}}_{\mathrm{dat}}, \tilde{y}^{\{2\}}_{\mathrm{dat}}, \dots, \tilde{y}^{\{n_{\mathcal{D}}\}}_{\mathrm{dat}}]^\top \in \mathbb{R}^{n_{\mathcal{D}}}, \end{aligned} \tag{2.32}$$

where the data is generated by

$$\tilde{y}_{\mathrm{dat}}^{\{i\}} = f_{\mathrm{uk}}(\boldsymbol{x}_{\mathrm{dat}}^{\{i\}}) + \nu,\ \nu \sim \mathcal{N}(0, \sigma_n^2) \tag{2.33}$$

for all $i = \{1, \ldots, n_{\mathcal{D}}\}$. Without any assumptions on f_{uk} it is obviously not possible to quantify the model error. Loosely speaking, the prior distribution of the GPR with kernel k must be suitable to learn the unknown function. More technically, f_{uk} must be an element of the RKHS spanned by the kernel as described in (2.24). This leads to the following assumption.

Assumption 2.1. *The function f_{uk} has a finite RKHS norm with respect to the kernel k, i.e., $\|f_{uk}\|_{\mathcal{H}} < \infty$, where $\mathcal{H}$ is the RKHS spanned by k.*

This sounds paradoxical as f_{uk} is assumed to be unknown. However, there exist kernels that can approximate any continuous function arbitrarily exact. Thus, for any continuous function, there exists an arbitrarily close function to f_{uk} which is element of the RKHS of an universal kernel. For more details, we refer to Section 2.1.5.

We classify the error quantification in three different approaches: i) the robust approach, ii) the scenario approach, and iii) the information-theoretical approach. The different techniques are presented in the following and visualized in Fig. 2.4. For the remainder of this section, we assume that a GPR is trained with the data set (2.32) and Assumption 2.1 holds.

Robust approach

The robust approach exploits the fact that the prediction of the GPR is Gaussian distributed. Thus, for any $\boldsymbol{z}^* \in \mathbb{R}^{n_z}$, the model error is bounded by

$$|f_{\mathrm{uk}}(\boldsymbol{z}^*) - \mu(f_{\mathrm{GP}}|\boldsymbol{z}^*, \mathcal{D})| \leq c\,\mathrm{var}(f_{\mathrm{GP}}|\boldsymbol{z}^*, \mathcal{D}) \tag{2.34}$$

with high probability where $c \in \mathbb{R}_{>0}$ adjusts the probability. However, for multiple test points $\boldsymbol{z}_1^*, \boldsymbol{z}_2^*, \ldots \in \mathbb{R}^{n_z}$, this approach neglects any correlation between $f_{\mathrm{GP}}(\boldsymbol{z}_1^*), f_{\mathrm{GP}}(\boldsymbol{z}_2^*), \ldots$. Figure 2.4 shows how for a given $\boldsymbol{z}_1^*$ and $\boldsymbol{z}_2^*$, the variance is exploited as upper bound. Thus, any prediction is handled independently, which leads to a very conservative bound, see [UBH18].

Scenario approach

Instead of using the mean and the variance as in the robust approach, the scenario approach deals with the samples of the GPR directly. In contrast to the other methods, there is no direct model error quantification but rather a sample based quantification. The idea is to draw a large number $n_{\mathrm{scen}} \in \mathbb{N}$ of sample functions $f_{\mathrm{GP}}^1, f_{\mathrm{GP}}^2, \ldots, f_{\mathrm{GP}}^{n_{\mathrm{scen}}}$ over $n_s \in \mathbb{N}$ sampling points. The sampling is performed by drawing multiple instances from f_{GP} given by the multivariate Gaussian distribution

$$\begin{bmatrix} Y \\ \\ f_{\mathrm{GP}}(\boldsymbol{z}_1^*) \\ \vdots \\ f_{\mathrm{GP}}(\boldsymbol{z}_{n_s}^*) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(\boldsymbol{x}_{\mathrm{dat}}^{\{1\}}) \\ \vdots \\ m(\boldsymbol{x}_{\mathrm{dat}}^{\{n_{\mathcal{D}}\}}) \\ m(\boldsymbol{z}_1^*) \\ \vdots \\ m(\boldsymbol{z}_{n_s}^*) \end{bmatrix}, \begin{bmatrix} K(X,X) + \sigma_n^2 I_{n_{\mathcal{D}}} & K(X^*,X) \\ \\ K(X^*,X)^\top & K(X^*,X^*) \end{bmatrix} \right), \tag{2.35}$$

where $X^* = [\boldsymbol{z}_1^*, \cdots, \boldsymbol{z}_{n_s}^*]$ contains the sampling points. Each sample can then be used in the application instead of the unknown function. For a large number of samples it is assumed that the unknown function is close to one of these samples. However, the crux of this approach is to determine, for a given model error $c \in \mathbb{R}_{>0}$, the required number of samples n_{scen} and probability $\delta_{scen} > 0$ such that

$$\mathrm{P}\left(|f_{\text{uk}}(\boldsymbol{z}^*) - f_{\text{GP}}^i(\boldsymbol{z}^*)| \leq c, i \in \{1, \dots, n_{\text{scen}}\}\right) \geq \delta_{scen} \tag{2.36}$$

for all $\boldsymbol{z}^* \in Z$. In Fig. 2.4, five different samples of a GP model are drawn as example.

Information-theoretical approach

Alternatively, the work in [Sri+12] derives an upper bound for samples of the GPR on a compact set with a specific probability. In contrast to the robust approach, the correlation between the function values are considered. We restate here the theorem.

Theorem 2.3 ([Sri+12])**.** *Given Assumption 2.1, the model error* $\Delta \in \mathbb{R}$

$$\Delta = |\,\mu(f_{GP}|\boldsymbol{z}, \mathcal{D}) - f_{uk}(\boldsymbol{z})| \tag{2.37}$$

is bounded for all $\boldsymbol{z}$ *on a compact set* $\Omega \subset \mathbb{R}^{n_z}$ *with a probability of at least* $\delta \in (0, 1)$ *by*

$$P\left\{\forall \boldsymbol{z} \in \Omega,\ \Delta \leq |\beta\, \Sigma^{\frac{1}{2}}(f_{GP}|\boldsymbol{z}, \mathcal{D})|\right\} \geq \delta, \tag{2.38}$$

where $\beta \in \mathbb{R}$ *is defined as*

$$\beta = \sqrt{2\|f_{uk}\|_k^2 + 300\gamma_{max} \ln^3\left(\frac{n_{\mathcal{D}} + 1}{1 - \delta}\right)}. \tag{2.39}$$

The variable $\gamma_{max} \in \mathbb{R}$ *is the maximum of the information gain*

$$\gamma_{max} = \max_{\boldsymbol{x}_{dat}^{\{1\}}, \dots, \boldsymbol{x}_{dat}^{\{n_{\mathcal{D}}+1\}} \in \Omega} \frac{1}{2} \log |I_{n_{\mathcal{D}}+1} + \sigma_n^{-2} K(\boldsymbol{z}, \boldsymbol{z}')| \tag{2.40}$$

with Gram matrix $K(\boldsymbol{z}, \boldsymbol{z}')$ *and the input elements* $\boldsymbol{z}, \boldsymbol{z}' \in \{\boldsymbol{x}_{dat}^{\{1\}}, \dots, \boldsymbol{x}_{dat}^{\{n_{\mathcal{D}}+1\}}\}$.

To compute this bound, the RKHS norm of f_{uk} must be known. That is in application usually not the case. However, often the norm can be upper bounded and thus, the bound in Theorem 2.3 can be upper bounded. For this purpose, the relation of the RKHS norm to the Lipschitz constant given by (2.28) is beneficial as the Lipschitz constant is more likely to be known. In general, the computation of the information gain is a non-convex optimization problem. However, the information capacity γ_{max} has a sub-linear dependency on the number of training points for many commonly used kernel functions [Sri+12]. Therefore, even though β is increasing with the number of training data, it is possible to learn the true function f_{uk} arbitrarily exactly [Ber+16]. In contrast to the other approaches, Theorem 2.3 allows to bound the error for any test point in a compact set. In Chapter 5, we will exploit this approach in GP model based control tasks. The right illustration of Fig. 2.4 visualizes the information-theoretical bound.

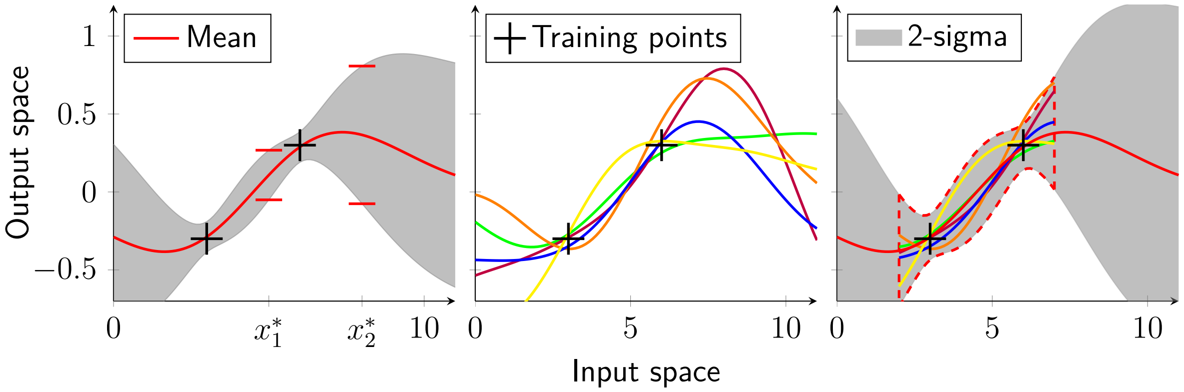


Figure 2.4: Different approaches to quantify the model error: Robust approach (left), scenario approach (middle), information-theoretical approach (right).

2.2 Model Selection

Equation (2.8) clearly shows the immense impact of the kernel on the posterior mean and variance. However, this is not surprising as the kernel is an essential part of the prior model. For practical applications that leads to the question how to choose the kernel. Additionally, most kernels depend on a set of hyperparameters that must be defined. Thus in order to turn GPR into a powerful practical tool it is essential to develop methods that address the model selection problem. We see the model selection as the determination of the kernel and its hyperparameters. We only focus on kernels that are defined on $\mathcal{Z} \subseteq \mathbb{R}^{n_z}$. In the next two subsections, we present different kernels and explain the role of the hyperparameters and their selection, mainly based on [Ras06].

Remark 2.3. *The selection of the kernel functions seems to be similar to the model selection for parametric models. However, there are two major differences: i) the selection is fully covered by the Bayesian methodology and ii) many kernels allow to model a wide range of different functions whereas parametric models a typically limited to very specific types of functions.*

2.2.1 Kernel Functions

The value of the kernel function $k(\boldsymbol{z}, \boldsymbol{z}')$ is an indicator of the interaction of two states $(\boldsymbol{z}, \boldsymbol{z}')$. Thus, an essential part of GPR is the selection of the kernel function and the estimation of its free parameters $\varphi_1, \varphi_2, \dots, \varphi_{n_\varphi}$, called hyperparameters. The number n_φ of hyperparameters depends on the kernel function. The choice of the kernel function and the determination of the corresponding hyperparameters can be seen as degrees of freedom of the regression. First of all, we start with the general properties of a function to be qualified as a kernel for GPR. A necessary and sufficient condition for the function $k\colon \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ to be a valid kernel is that the Gram matrix, see (2.5), is positive semidefinite for all possible input values [SC04].

Remark 2.4. *As shown in Section 2.1.5, the kernel function must be* positive definite *to span a unique RKHS. That seems to be contradictory to the required* positive semi-definiteness *of the Gram matrix. The solution is the definition of positive definite kernels as it is equivalent to a positive semi-definite Gram matrix. In detail, a symmetric function $k\colon \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ is* *a* positive definite *kernel on $\mathcal{Z}$ if*

$$\sum_{j=1}^{n_{\mathcal{D}}}\sum_{i=1}^{n_{\mathcal{D}}} k(\boldsymbol{x}_{dat}^{\{i\}}, \boldsymbol{x}_{dat}^{\{j\}})c_i c_j \geq 0 \tag{2.41}$$

holds for any $n_{\mathcal{D}} \in \mathbb{N}$, $\boldsymbol{x}_{dat}^{\{1\}}, \dots, \boldsymbol{x}_{dat}^{\{n_{\mathcal{D}}\}} \in \mathcal{Z}$ and $c_1, \dots, c_n \in \mathbb{R}$. Thus, there exists a positive semi-definite *matrix $A_G \in \mathbb{R}^{n_{\mathcal{D}} \times n_{\mathcal{D}}}$ such that*

$$\boldsymbol{x}_{dat}^{\top} A_G \boldsymbol{x}_{dat} = \sum_{j=1}^{n_{\mathcal{D}}}\sum_{i=1}^{n_{\mathcal{D}}} k(\boldsymbol{x}_{dat}^{\{i\}}, \boldsymbol{x}_{dat}^{\{j\}})c_i c_j \tag{2.42}$$

holds for any $n_{\mathcal{D}} \in \mathbb{N}$ and $\boldsymbol{z} \in \mathcal{Z}$.

The set of functions k which fulfill this condition is denoted with $\mathcal{K}$. Kernel functions can be separated into two classes, the *stationary* and *non-stationary* kernels. A stationary kernel is a function of the distance $\boldsymbol{z} - \boldsymbol{z}'$. Thus, it is invariant to translations in the input space. In contrast, non-stationary kernels depend directly on $\boldsymbol{z}$,$\boldsymbol{z}'$ and are often functions of a dot product $\boldsymbol{z}^{\top}\boldsymbol{z}$. In the following, we list some common kernel functions with their basic properties. Even though the number of presented kernels is limited, new kernels can be constructed easily as $\mathcal{K}$ is closed under specific operations such as addition and scalar multiplication. At the end, we summarize the equation of each kernel in Table 2.1 and provide a comparative example.

Constant Kernel

The equation for the constant kernel is given by

$$k(\boldsymbol{z}, \boldsymbol{z}') = \varphi_1^2. \tag{2.43}$$

This kernel is mostly used in addition to other kernel functions. It depends one a single hyperparameter $\varphi_1 \in \mathbb{R}_{\geq 0}$.

Linear Kernel

The equation for the linear kernel is given by

$$k(\boldsymbol{z}, \boldsymbol{z}') = \boldsymbol{z}^{\top}\boldsymbol{z}'. \tag{2.44}$$

The linear kernel is a dot-product kernel and thus, non-stationary. The kernel can be obtained from Bayesian linear regression as shown in Section 2.1.4. The linear kernel is often used in combination with the constant kernel to include a bias.

Polynomial Kernel

The equation for the polynomial kernel is given by

$$k(\boldsymbol{z}, \boldsymbol{z}') = \left(\boldsymbol{z}^{\top}\boldsymbol{z}' + \varphi_1^2\right)^p, \, p \in \mathbb{N}. \tag{2.45}$$

The polynomial kernel has an additional parameter $p \in \mathbb{N}$, that determines the degree of the polynomial. Since a dot product is contained, the kernel is also non-stationary. The prior variance grows rapidly for $\|\boldsymbol{z}\| > 1$ such that the usage for some regression problems is limited. It depends on a single hyperparameter $\varphi_1 \in \mathbb{R}_{\geq 0}$.

Matérn Kernel

The equation for the Matérn kernel is given by

$$k(\boldsymbol{z},\boldsymbol{z}')=\varphi_1^2\exp\left(-\frac{\sqrt{2\check{p}}\|\boldsymbol{z}-\boldsymbol{z}'\|}{\varphi_2}\right)\frac{p!}{(2p)!}\sum_{i=0}^{p}\frac{(p+i)!}{i!(p-i)!}\left(\frac{\sqrt{8\check{p}}\|\boldsymbol{z}-\boldsymbol{z}'\|}{\varphi_2}\right)^{p-i} \tag{2.46}$$

with $\check{p}=p+\frac{1}{2}, p\in\mathbb{N}$. The Matérn kernel is a very powerful kernel and presented here with the most common parameterization for $\check{p}$. Functions drawn from a GP model with Matérn kernel are p-times differentiable. The more general equation of this stationary kernel can be found in [Bis06]. This kernel is an *universal kernel* which is explained in the following.

Lemma 2.1 ([SC08, Lemma 4.55])**.** *Consider the RKHS $\mathcal{H}(\mathcal{Z}_c)$ of an universal kernel on any prescribed compact subset $\mathcal{Z}_c\in\mathcal{Z}$. Given any positive number ε and any function $f_{\mathcal{C}}\in\mathcal{C}^1(\mathcal{Z}_C)$, there is a function $f_{\mathcal{H}}\in\mathcal{H}(\mathcal{Z}_c)$ such that $\|f_{\mathcal{C}}-f_{\mathcal{H}}\|_{\mathcal{Z}_c}\leq\varepsilon$.*

Intuitively speaking, a GPR with an universal kernel can approximate any continuous function arbitrarily exact on a compact set. For $p\to\infty$, it results in the squared exponential kernel. The two hyperparameters are $\varphi_1\in\mathbb{R}_{\geq 0}$ and $\varphi_2\in\mathbb{R}_{>0}$.

Squared Exponential Kernel

The equation for the squared exponential kernel is given by

$$k(\boldsymbol{z},\boldsymbol{z}')=\varphi_1^2\exp\left(-\frac{\|\boldsymbol{z}-\boldsymbol{z}'\|^2}{2\varphi_2^2}\right). \tag{2.47}$$

Probably the most widely used kernel function for GPR is the squared exponential kernel, see [Ras06]. The hyperparameter φ_1 describes the signal variance which determines the average distance of the data-generating function from its mean. The lengthscale φ_2 defines how far it is needed to move along a particular axis in input space for the function values to become uncorrelated. Formally, the lengthscale determines the number of expected upcrossings of the level zero in a unit interval by a zero-mean GP. The squared exponential kernel is infinitely differentiable, which means that the GPR exhibits a smooth behavior. As limit of the Matérn kernel, it is also an universal kernel, see [MXZ06].

Example 2.6. Figure 2.5 shows the power for regression of universal kernel functions. In this example, a GPR with squared exponential kernel is used for different training data sets. The hyperparameter are optimized individually for each training data set by means of the likelihood, see Section 2.2.2. Note that all presented regressions are based on the **same** GP model, i.e. the same kernel function, but with different data sets. That highlights again the superior flexibility of GPR.

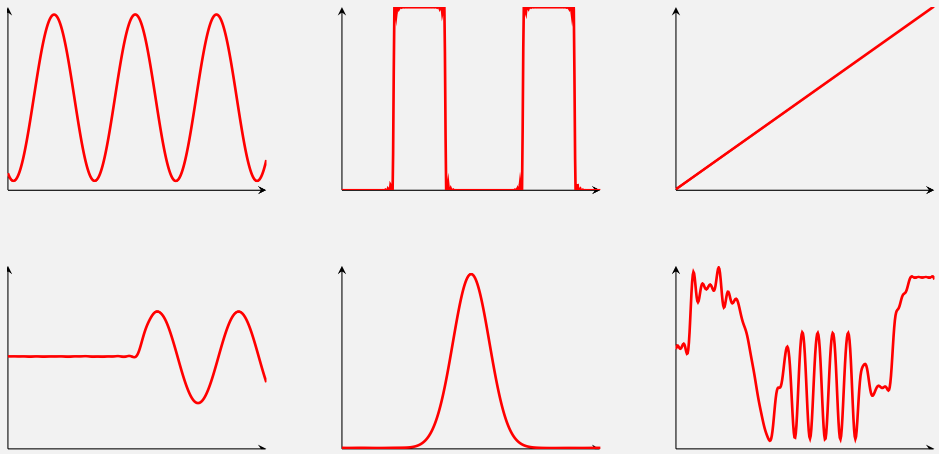


Figure 2.5: Examples for the flexibility of the regression that all are based on the same GP model.

Rational Quadratic Kernel

The equation for the rational quadratic kernel is given by

$$k(\boldsymbol{z}, \boldsymbol{z}') = \varphi_1^2 \left(1 + \frac{\|\boldsymbol{z} - \boldsymbol{z}'\|^2}{2p\varphi_2^2}\right)^{-p}, \; p \in \mathbb{N}. \tag{2.48}$$

This kernel is equivalent to summing over infinitely many squared exponential kernels with different lengthscales. Hence, GP priors with this kernel are expected to see functions which vary smoothly across many lengthscales. The parameter p determines the relative weighting of large-scale and small-scale variations. For $p \to \infty$, the rational quadratic kernel is identical to the squared exponential kernel.

Squared Exponential ARD Kernel

The equation for the squared exponential ARD kernel is given by

$$k(\boldsymbol{z}, \boldsymbol{z}') = \varphi_1^2 \exp\left(-(\boldsymbol{z} - \boldsymbol{z}')^\top P^{-1}(\boldsymbol{z} - \boldsymbol{z}')\right), \; P = \operatorname{diag}(\varphi_2^2, \dots, \varphi_{1+n_z}^2). \tag{2.49}$$

The *automatic relevance determination* (ARD) extension to the squared exponential kernel allows for independent lenghtscales $\varphi_2, \dots, \varphi_{1+n_z} \in \mathbb{R}_{>0}$ for each dimension of $\boldsymbol{z}, \boldsymbol{z}' \in \mathbb{R}^{n_z}$. The individual lenghtscales are typically larger for dimensions which are irrelevant as the covariance will become almost independent of that input. A more detailed discussion about the advantages of different kernels can be found, for instance, in [Mac97] and [Bis06].

Example 2.7. In this example, we use three GPRs with the same set of training data

$$X = [1, 3, 5, 7, 9], \; Y = [0, 1, 2, 3, 6] \tag{2.50}$$

but with different kernels, namely the squared exponential (2.47), the linear (2.44), and the polynomial (2.45) kernel. Figure 2.6 shows the different shapes of the regressions with the posterior mean (red), the posterior variance (gray shaded) and the training points (black). Even for this simple data set, the flexibility of the squared exponential kernel is already visible.

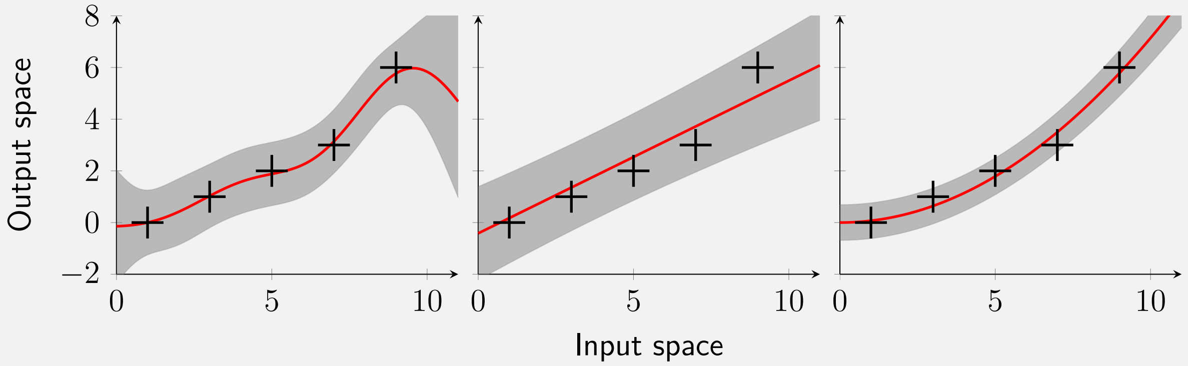


Figure 2.6: GPR with different kernels: squared exponential (left), linear (middle) and polynomial with degree 2 (right).

Kernel name	$k(\boldsymbol{z}, \boldsymbol{z}') =$
Constant	φ_1^2
Linear	$\boldsymbol{z}^\top \boldsymbol{z}' + \varphi_1^2$
Polynomial $p \in \mathbb{N}$	$\left(\boldsymbol{z}^\top \boldsymbol{z}' + \varphi_1^2\right)^p$
Matérn $\check{p} = p + \frac{1}{2}, p \in \mathbb{N}$	$\varphi_1^2 \exp\left(-\frac{\sqrt{2\check{p}}\\|\boldsymbol{z}-\boldsymbol{z}'\\|}{\varphi_2}\right) \frac{p!}{(2p)!} \sum_{i=0}^{p} \frac{(p+i)!}{i!(p-i)!} \left(\frac{\sqrt{8\check{p}}\\|\boldsymbol{z}-\boldsymbol{z}'\\|}{\varphi_2}\right)^{p-i}$
Squared exponential	$\varphi_1^2 \exp\left(-\frac{\\|\boldsymbol{z}-\boldsymbol{z}'\\|^2}{2\varphi_2^2}\right)$
Rational quadratic	$\varphi_1^2 \left(1 + \frac{\\|\boldsymbol{z}-\boldsymbol{z}'\\|^2}{2p\varphi_2^2}\right)^{-p}$
Squared exponential ARD	$\varphi_1^2 \exp\left(-(\boldsymbol{z} - \boldsymbol{z}')^\top P^{-1} (\boldsymbol{z} - \boldsymbol{z}')\right)$, $P = \operatorname{diag}(\varphi_2^2, \dots, \varphi_{1+n_z}^2)$

Table 2.1: Overview of some commonly used kernel functions.

2.2.2 Hyperparameter Optimization

In addition to the selection of a kernel function, values for any hyperparameter must be determined to perform the regression. The number of hyperparameters depends on the kernel function used. We concatenate all hyperparameters in a vector $\boldsymbol{\varphi}$ with size $n_\varphi \in \mathbb{N}$, where $\boldsymbol{\varphi} \in \Phi \subseteq \mathbb{R}^{n_\varphi}$. The hyperparameter set Φ is introduced to cover the different spaces of the individual hyperparameters as defined in the following.

Definition 2.1. The set Φ is called a hyperparameter set for a kernel function k if and only if the set Φ is a domain for the hyperparameters $\boldsymbol{\varphi}$ of k.

Often, the signal noise σ_n^2, see (2.4), is also treated as hyperparameter. For a better understanding, we keep the signal noise separated from the hyperparameters. There exist several techniques that allow computing the hyperparameters and the signal noise with respect to one optimality criterion. From a Bayesian perspective, we want to find the vector of hyperparameters $\boldsymbol{\varphi}$ which are most likely for the output data Y given the inputs X and a GP model. For this purpose, one approach is to optimize the *log marginal likelihood function* of the GP. Another idea is to split the training set into two disjoint sets, one which is actually used for training, and the other, the validation set, which is used to monitor performance. This approach is known as *cross-validation*. In the following, these two techniques for the selection of hyperparameters are presented.

Log Marginal Likelihood Approach

A very common method for the optimization of the hyperparameters is by means of the *negative log marginal likelihood function*, often simply named as (neg. log) likelihood function. It is *marginal* since it is obtained through marginalization over the function f_{GP}. The marginal likelihood is the likelihood that the output data $Y \in \mathbb{R}^{n_{\mathcal{D}}}$ fits to the input data X with the hyperparameters $\boldsymbol{\varphi}$. It is given by

$$\log p(Y|X, \boldsymbol{\varphi}) = -\frac{1}{2} Y^\top (K + \sigma_n^2 I_{n_{\mathcal{D}}})^{-1} Y - \frac{1}{2} \log |K + \sigma_n^2 I_{n_{\mathcal{D}}}| - \frac{n_{\mathcal{D}}}{2} \log 2\pi. \tag{2.51}$$

A detailed derivation can be found in [Ras06]. The three terms of the marginal likelihood in (2.51) have the following roles:

- $\frac{1}{2} Y^\top (K + \sigma_n^2 I_{n_{\mathcal{D}}})^{-1} Y$ is the only term that depends on the output data Y and represents the data-fit.
- $\frac{1}{2} \log |K + \sigma_n^2 I_{n_{\mathcal{D}}}|$ penalizes the complexity depending on the kernel function and the input data X.
- $\frac{n_{\mathcal{D}}}{2} \log 2\pi$ is a normalization constant.

Remark 2.5. *For the sake of notational simplicity, we suppress the dependency on the hyperparameters of the kernel function k whenever possible.*

The optimal hyperparameters $\boldsymbol{\varphi}^* \in \Phi$ and signal noise σ_n^* in the sense of the likelihood are obtained as the minimum of the negative log marginal likelihood function

$$\begin{bmatrix} \boldsymbol{\varphi}^* \\ \sigma_n^* \end{bmatrix} = \arg \min_{\boldsymbol{\varphi} \in \Phi, \sigma_n \in \mathbb{R}_{\geq 0}} \log p(Y|X, \boldsymbol{\varphi}). \tag{2.52}$$

Since an analytic solution of the derivation of (2.51) is impossible, a gradient based optimization algorithm is typically used to minimize the function. However, the negative log likelihood is non-convex in general such that there is no guarantee to find the optimum $\boldsymbol{\varphi}^*, \sigma_n^*$. In fact, every local minimum corresponds to a particular interpretation of the data. In the following example, we visualize how the hyperparameters affect the regression.

Example 2.8. A GPR with the squared exponential kernel is trained on eight data points. The signal variance is fixed to $\varphi_1 = 2.13$. First, we visualize the influence of the lengthscale. For this purpose, the signal noise is fixed to $\sigma_n = 0.21$. Figure 2.7 shows the posterior mean of the regression and the neg. log likelihood function. On the left side are three posterior means for different lengthscales. A short lengthscale results in overfitting whereas a large lengthscale smooths out the training data (black crosses). The dotted red function represents the mean with optimized lengthscale by a descent gradient algorithm with respect to (2.52). The right plot shows the neg. log likelihood over the signal variance φ_1 and lengthscale φ_2. The minimum is here at $\boldsymbol{\varphi}^* = [2.13, 1.58]^\top$.

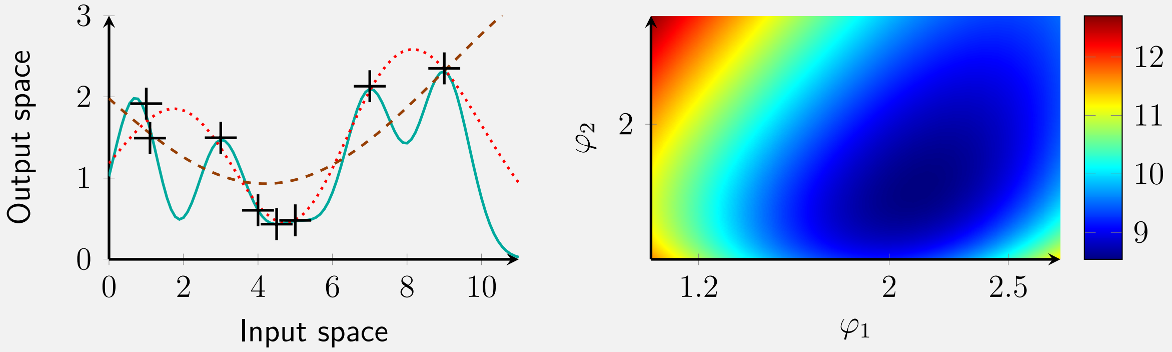


Figure 2.7: Left: Regression with different lengthscales: $\varphi_2 = 0.67$ (cyan, solid), $\varphi_2 = 7.39$ (brown, dashed), and $\varphi_2 = 1.58$ (red, dotted). Right: Neg. log likelihood function over signal variance φ_1 and lengthscale φ_2.

Next, the meaning of different interpretations of the data is visualized by varying the signal noise σ_n and the lengthscale φ_2. The right plot of Fig. 2.8 shows two minima of the negative log likelihood function. The lower left minimum at $\log(\sigma_n) = 0.73$ and $\log(\varphi_2) = -1.51$ interprets the data as slightly noisy which leads to the dotted red posterior mean in the left plot. In contrast, the upper right minimum at $\log(\sigma_n) = 5$ and $\log(\varphi_2) = -0.24$ interprets the data as very noisy without a trend, which manifests as the cyan posterior mean in the left plot. Depending on the initial value, a gradient based optimizer would terminate in one of these minima.

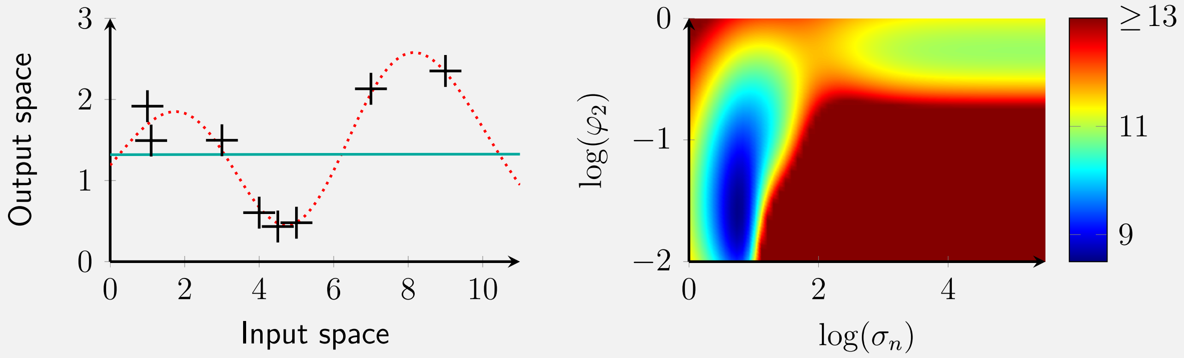


Figure 2.8: Left: Different interpretation of the data: Noisy data without a trend (cyan, solid) and slightly noisy data (red, dotted). Right: Negative log likelihood function over signal noise σ_n and lengthscale φ_2.

Cross-validation approach

This approach works with a separation of the data set $\mathcal{D}$ in two classes: one for training and one for validation. Cross-validation is almost always used in the k_{cv}-fold cross-validation setting: the k_{cv}-fold cross-validation data is split into k_{cv} disjoint, equally sized subsets; validation is done on a single subset and training is done using the union of the remaining $k_{\mathrm{cv}}-1$ subsets, the entire procedure is repeated k_{cv} times, each time with a different subset for validation. Here, without loss of generality, we present the leave-one-out cross-validation, which means $k_{\mathrm{cv}} = n_{\mathcal{D}}$. The predictive log probability when leaving out a training point $\{\boldsymbol{x}_{\mathrm{dat}}^{\{i\}}, \tilde{y}_{\mathrm{dat}}^{\{i\}}\}$ is given by

$$\log p(y_{\mathrm{dat}}^{\{i\}}|X, Y_{-i}, \boldsymbol{\varphi}) = -\frac{1}{2}\log\left(\mathrm{var}_{-i}\right) - \frac{\left(\tilde{y}_{\mathrm{dat}}^{\{i\}} - \mu_{-i}\right)^2}{2\,\mathrm{var}_{-i}} - \frac{n_{\mathcal{D}}}{2}\log 2\pi, \tag{2.53}$$

where $\mu_{-i} = \mu(f_{\mathrm{GP}}(\boldsymbol{x}_{\mathrm{dat}}^{\{i\}})|\boldsymbol{x}_{\mathrm{dat}}^{\{i\}}, X_{:,-i}, Y_{-i})$ and $\mathrm{var}_{-i} = \mathrm{var}(f_{\mathrm{GP}}(\boldsymbol{x}_{\mathrm{dat}}^{\{i\}})|\boldsymbol{x}_{\mathrm{dat}}^{\{i\}}, X_{:,-i}, Y_{-i})$. The $-i$ index indicates X and Y without the element $\boldsymbol{x}_{\mathrm{dat}}^{\{i\}}$ and $\tilde{y}_{\mathrm{dat}}^{\{i\}}$, respectively. Thus, (2.53) is the probability for the output $y_{\mathrm{dat}}^{\{i\}}$ at $\boldsymbol{x}_{\mathrm{dat}}^{\{i\}}$ but without the training point $\{\boldsymbol{x}_{\mathrm{dat}}^{\{i\}}, \tilde{y}^{\{i\}}\}$. Accordingly, the leave-one-out log predictive probability $L_{\mathrm{LOO}} \in \mathbb{R}$ is

$$L_{\mathrm{LOO}} = \sum_{i=1}^{n_{\mathcal{D}}} \log p(y_{\mathrm{dat}}^{\{i\}}|X, Y_{-i}, \boldsymbol{\varphi}). \tag{2.54}$$

In comparison to the log likelihood approach (2.52), the cross-validation is in general more computationally expensive but might find a better representation of the data set, see [GE79] for discussion and related approaches.

2.3 Gaussian Process Dynamical Models

So far, we consider GPR in non-dynamical settings where only an input-to-output mapping is considered. However, Gaussian process dynamical models (GPDMs) have recently become a versatile tool in system identification because of their beneficial properties such as the bias variance trade-off and the strong connection to Bayesian mathematics, see [FCR14]. In many works, where GPs are applied to dynamical model, only the mean function of the process is employed, e.g., in [WHB05] and [Cho+13]. This is mainly because GP models are often used to replace deterministic parametric models. However, GPDMs contain a much richer description of the underlying dynamics, but also the uncertainty about the model itself when the full probabilistic representation is considered. Therefore, one main aspect of GPDMs is to distinguish between recurrent structures and non-recurrent structures. A model is called recurrent if parts of the regression vector depend on the outputs of the model. Even though recurrent models become more complex in terms of their behavior, they allow to model sequences of data, see [Sjö+95]. If all states are fed back from the model itself, we get a simulation model, which is a special case of the recurrent structure. The advantage of such a model is its property to run independently from the real system. Thus, it is suitable for simulations, as it allows multi-step ahead predictions. In this thesis, we focus on two often-used recurrent structures: the Gaussian process state space model (GP-SSM) and the Gaussian process nonlinear error output (GP-NOE) model.

2.3.1 Gaussian Process State Space Models

Gaussian process state space models are structured as a discrete-time system. In this case, the states are the regressors, which is visualized in Fig. 2.9. This approach allows to be more efficient, since the regressors are less restricted in their internal structure as in input-output models. Thus, a very efficient model in terms of number of regressors might be possible. The mapping from the states to the output is often be assumed to be known. The situation, where the output mapping describes a known sensor model, is such an example. It is mentioned in [Fri+13] that using too flexible models for both, the state mapping $\boldsymbol{f}$ and the output mapping, can result in problems of non-identifiability. Therefore, we focus on a known output mapping. The mathematical model of the GP-SSM is thus given by

$$\begin{aligned} \boldsymbol{x}_{t+1} &= \boldsymbol{f}(\boldsymbol{\xi}_t) = \begin{cases} f_1(\boldsymbol{\xi}_t) \sim \mathcal{GP}\left(m^1(\boldsymbol{\xi}_t), k^1(\boldsymbol{\xi}_t, \boldsymbol{\xi}_t')\right) \\ \vdots \qquad \vdots \quad \vdots \\ f_{n_x}(\boldsymbol{\xi}_t) \sim \mathcal{GP}\left(m^{n_x}(\boldsymbol{\xi}_t), k^{n_x}(\boldsymbol{\xi}_t, \boldsymbol{\xi}_t')\right). \end{cases} \\ \boldsymbol{y}_t &\sim p(\boldsymbol{y}_t | \boldsymbol{x}_t, \boldsymbol{\gamma}_y), \end{aligned} \tag{2.55}$$

where $\boldsymbol{\xi}_t \in \mathbb{R}^{n_\xi}, n_\xi = n_x + n_u$ is the concatenation of the state vector $\boldsymbol{x}_t \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$ and the input $\boldsymbol{u}_t \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$ such that $\boldsymbol{\xi}_t = [\boldsymbol{x}_t; \boldsymbol{u}_t]$. The mean function is given by continuous functions $m^1, \dots, m^{n_x} \colon \mathbb{R}^{n_\xi} \to \mathbb{R}$. The output mapping is parametrized by a known vector $\boldsymbol{\gamma}_y \in \mathbb{R}^{n_\gamma}$ with $n_\gamma \in \mathbb{N}$. The system identification task for the GP-SSM mainly focuses on $\boldsymbol{f}$ in particular. It can be described as finding the state-transition probability conditioned on the observed training data.

Remark 2.6. *The potentially unknown number of regressors can be determined using established nonlinear identification techniques as presented in [KL99], or exploiting embedded techniques such as automatic relevance determination [Koc16]. A mismatch leads to similar issues as in parametric system identification.*

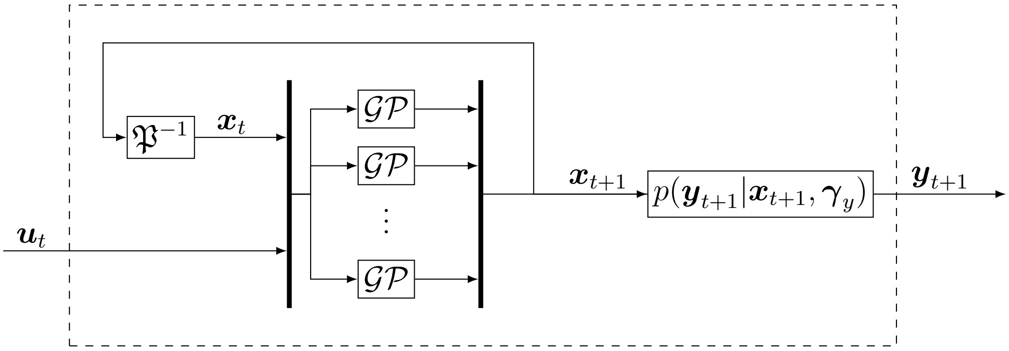


Figure 2.9: Structure of a GP-SSM with $\mathfrak{P}$ as backshift operator, such that $\mathfrak{P}^{-1}\boldsymbol{x}_{t+1} = \boldsymbol{x}_t$

2.3.2 Gaussian Process Nonlinear Output Error Models

The GP-NOE model uses the past $n_{\text{in}} \in \mathbb{N}_{>0}$ input values $\boldsymbol{u}_t \in \mathcal{U}$ and the past $n_{\text{out}} \in \mathbb{N}_{>0}$ output values $\boldsymbol{y}_t \in \mathbb{R}^{n_y}$ of the model as the regressors. Figure 2.10 shows the structure of

GP-NOE, where the outputs are fed back. Analogously to the GP-SSM, the mathematical model of the GP-NOE is given by

$$\boldsymbol{y}_{t+1} = \boldsymbol{h}(\boldsymbol{\zeta}_t) = \begin{cases} h_1(\boldsymbol{\zeta}_t) \sim \mathcal{GP}\left(m^1(\boldsymbol{\zeta}_t), k^1(\boldsymbol{\zeta}_t, \boldsymbol{\zeta}_t')\right) \\ \vdots \qquad \vdots \quad \vdots \\ h_{n_y}(\boldsymbol{\zeta}_t) \sim \mathcal{GP}\left(m^{n_y}(\boldsymbol{\zeta}_t), k^{n_y}(\boldsymbol{\zeta}_t, \boldsymbol{\zeta}_t')\right), \end{cases} \tag{2.56}$$

where $\boldsymbol{\zeta}_t \in \mathbb{R}^{n_\zeta}, n_\zeta = n_{\text{out}} n_y + n_{\text{in}} n_u$ is the concatenation of the past outputs $\boldsymbol{y}_t$ and inputs $\boldsymbol{u}_t$ such that $\boldsymbol{\zeta}_t = [\boldsymbol{y}_{t-n_{\text{out}}+1}; \dots; \boldsymbol{y}_t; \boldsymbol{u}_{t-n_{\text{in}}+1}; \dots; \boldsymbol{u}_t]$. The mean function is given by continuous functions $m^1, \dots, m^{n_y} \colon \mathbb{R}^{n_\zeta} \to \mathbb{R}$. In contrast to nonlinear autoregressive exogenous models, that focus on one-step ahead prediction, a NOE model is more suitable for simulations as it considers the multi-step ahead prediction [Nel13]. However, the drawback is a more complex training procedure that requires a nonlinear optimization scheme due to their recurrent structure [Koc16].

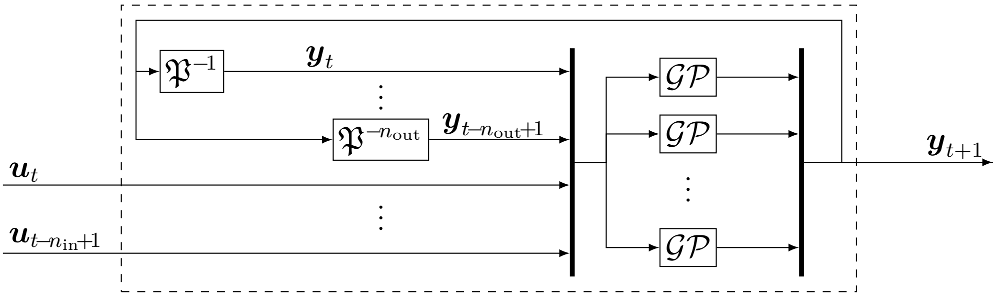


Figure 2.10: Structure of a GP-NOE model with $\mathfrak{P}$ as backshift operator ($\mathfrak{P}^{-1}\boldsymbol{y}_{t+1} = \boldsymbol{y}_t$)

Remark 2.7. *It is always possible to convert an identified input-output model into a state-space model, see[PL70]. However, focusing on state-space models only would preclude the development of a large number of useful identification results for input-output models.*

2.4 Summary

In this chapter, we introduce the GP and its usage in GPR. Based on the property, that every finite subset of a GP follows a multi-variate Gaussian distribution, a closed-form equation can be derived to predict the mean and variance for a new test point. The GPR can intrinsically handle noisy output data if it is Gaussian distributed. As GPR is a data-driven method, only little prior knowledge is necessary for the regression. Further, the complexity of GP models scales with the number of training points. A degree of freedom in the modeling is the selection of the kernel function and its hyperparameters. We present an overview of common kernels and the necessary properties to be a valid kernel function. For the hyperparameter determination, two approaches based on numerical optimization are shown. The kernel of the GP is uniquely related to a RKHS, which determines the shape of the samples of the GP. Based on this, we compare different approaches for the quantification of the model error that quantifies the error between the GPR and the actual data-generating function. Finally, we introduce how GP models can be used as dynamical systems in GP-SSMs and GP-NOE models.

3 Control Properties of GPDMs

Models of dynamical systems are key for many common control approaches, such as model predictive control, feedback linearization, and computed torque control, see [HW13]. In addition, models allow to test control approaches in simulations without time-consuming and potentially dangerous experimental evaluations. The control theoretical properties of these dynamical models are crucial for investigation in stability and performance guarantees of the closed-loop. In particular, the model must be capable to reproduce the real system behavior. For instance, a linear model might be not sufficiently precise for a highly nonlinear system. For parametric models exist many analysis tools for control theoretical properties such as stability, equilibria, and convergence rate [DFT13]. Especially for linear systems, there exists a strong and comprehensive theory, which established a relationship between the model parameters and the control properties, mainly based on the the eigenvalues and structure of the system matrix. In contrast, the analysis of these properties for GPDMs is challenging, as the non-parametric structure inhibits the application of the well-known tools for parametric models. Therefore, the control properties of the GPDMs are only sparsely researched even though the application of GPDMs become increasingly popular in control, e.g., for adaptive control [Rog+11].

Related Work and Open Problems

In order to provide rigorous guarantees on the system behavior, control properties of GPDMs need to be well-understood, see, e.g., [Koc+03b] and [AK08]. In fact, it has been widely acknowledged, e.g., in [KM05], that stability issues of GPDMs require careful attention in the future. In the following, we list the most important related approaches on the analysis of GPDMs. For linear system identification, there exists a stable kernel approach, which includes information on impulse response stability [COL12]. Chowdhary et al. presents a stability proof of an adaptive control approach with a GP model [CHK12] for nonlinear systems which is based on a bounded error model. Related to GP models are Gaussian Mixture Models (GMMs), which assume that every data point is generated from a mixture of a finite number of Gaussian distributions. Khansari-Zadeh et al. show for GMMs a synthesis approach for learning stable trajectories of a nonlinear dynamical systems [KB11a]. However, the approach is not suitable for the non-parametric GP model. A numerical stability evaluation of GPDMs is presented in [Vin+16] but it suffers from scalability issues. Hence, the fundamental stability analysis of GPDMs is still open.

In this chapter, we introduce analysis tools for a class of discrete-time GPDMs. We start with the general challenge of using GPDMs as predictive models in Section 3.1. Afterwards, boundedness properties for GP-SSMs and GP-NOE models are derived in Section 3.2. Finally, quantitative results for specific classes of GPDMs are presented in Sections 3.3 and 3.4.

3.1 The Crux of GPDM Prediction

The prediction with discrete-time GPDMs, needed for simulations and model-based control approaches, is more challenging than GPR prediction: The reason is the feedback of the model's output to the input that manifests as correlation between the current and past states defined by the GP model. Therefore, a prediction with the presented GP models in Section 2.3 would require the sampling of the probabilistic mappings $\boldsymbol{f}$ and $\boldsymbol{h}$ given by (2.55) and (2.56), respectively. Once sampled, the model could be treated as standard discrete-time system. Unfortunately, these functions are defined on the sets $\mathcal{X} \subseteq \mathbb{R}^{n_x}, \mathcal{U} \subseteq \mathbb{R}^{n_u}$ which contain infinitely many points. Thus, it would be necessary to draw an infinite-dimensional object which represents a sample of the probabilistic mappings. This is not possible without further simplifications, e.g., discrete sampling with interpolation, see (2.35). To overcome this issue, we will marginalize out the probabilistic mapping to respect the nonparametric nature of the model. Thus, the result is a joint probability distribution of the states without dependencies on the probabilistic mappings $\boldsymbol{f}, \boldsymbol{h}$. Without lack of generality, the following proposition focuses on an one-dimensional GP-SSM with $\mathcal{X} = \mathcal{U} = \mathbb{R}$. The approach can easily extended to multi-dimensional GP-SSMs due to the fact that the GPs for each state dimension are assumed to be independent (2.55). The presented approach is analogously applicable for the GP-NOE model.

Remark 3.1. *As this section focuses on the properties of GPDMs, regardless of the training procedure, we assume for the remainder of the section that a training set $\mathcal{D}$ is existent and available. For further information on the training procedure, we refer to [Koc+05; KP11] for GP-NOE models and to [WFH08; Fri+13; Ele+17] for GP-SSMs.*

Remark 3.2. *For the sake of notational simplicity, we consider GPDMs with identical kernels and noise of the training data for each output dimension. The results can easily be extended to GPDMs with different kernels and noise for each output dimension.*

Proposition 3.1. *Consider a one-dimensional GP-SSM as in (2.55) with training data set $\mathcal{D} = \{X, Y\}$, where Y is corrupted by Gaussian noise $\mathcal{N}(0, \sigma_n^2)$. For given inputs $\boldsymbol{u}_{0:t} = [u_0, \dots, u_t] \in \mathbb{R}^{1 \times t+1}$, the distribution over the states $\boldsymbol{x}_{0:t} = [x_0, \dots, x_t] \in \mathbb{R}^{1 \times t+1}$, with $t \in \mathbb{N}$, is given by*

$$p(\boldsymbol{x}_{0:t} | \boldsymbol{u}_{0:t}, \mathcal{D}) = p(x_0) \det\left((2\pi)^t \check{K}\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2} \left(\boldsymbol{x}_{1:t} - \check{\boldsymbol{m}}_{0:t-1}\right) \check{K}^{-1} \left(\boldsymbol{x}_{1:t} - \check{\boldsymbol{m}}_{0:t-1}\right)^\top\right), \tag{3.1}$$

with the Gram matrix

$$\check{K} = K(\boldsymbol{\xi}_{0:t-1}, \boldsymbol{\xi}_{0:t-1}) - K(\boldsymbol{\xi}_{0:t-1}, X)^\top \left(K + \sigma_n^2 I_{n_\mathcal{D}}\right)^{-1} K(\boldsymbol{\xi}_{0:t-1}, X), \tag{3.2}$$

where $\boldsymbol{\xi} = [x; u] \in \mathbb{R}^2$. The elements of the mean vector $\check{\boldsymbol{m}}_{0:t-1} \in \mathbb{R}^{1 \times t}$ are given by

$$\check{m}_i = m(\boldsymbol{\xi}_i) + K(\boldsymbol{\xi}_i, X)^\top (K + \sigma_n^2 I_{n_\mathcal{D}})^{-1} (Y - \boldsymbol{m}(X)) \tag{3.3}$$

for all $i = \{0, \dots, t-1\}$ with $\boldsymbol{m}(X) = [m(X_1); \dots; m(X_{n_\mathcal{D}})]$.

Before we start with the proof, note that even if the probability distribution (3.1) looks similar to a Gaussian distribution, it is not Gaussian, since the matrix $\check{K}$ depends on the past states $\boldsymbol{x}_{0:t-1}$ by $\boldsymbol{\xi}_{0:t-1}$. In Fig. 3.1, an example of a joint distribution of a GP-SSM with squared exponential kernel is shown. The initial state x_0 is Gaussian distributed, whereas the joint distribution of $p(x_{0:1})$ is clearly non-Gaussian, since x_0 is passed through the nonlinear transition function.

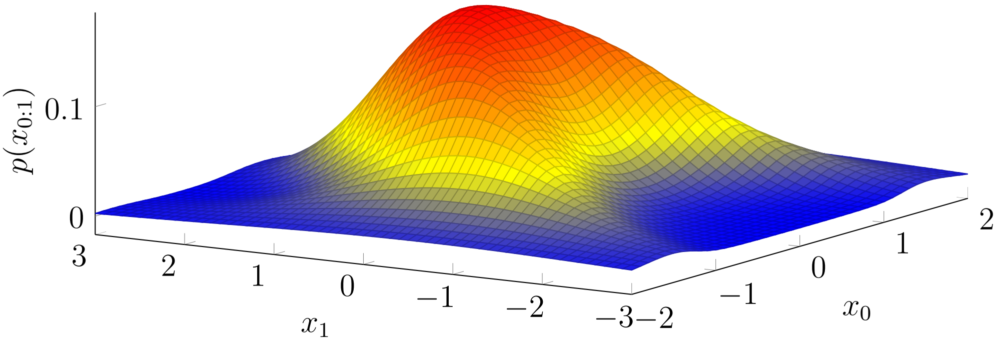


Figure 3.1: The joint probability of a GP-SSM is in general non-Gaussian.

Proof. We start with the joint probability distribution of $\boldsymbol{x}_{0:t}$ and the stochastic process $f(\cdot)$ given by

$$p(\boldsymbol{x}_{0:t}, f(\cdot)|\boldsymbol{u}_{0:t}, \mathcal{D}) = p(f(\cdot))p(x_0)\prod_{j=1}^{t} p(x_j|x_{j-1}, u_{t-1}, \mathcal{D}, f(\cdot)). \tag{3.4}$$

The intuitive goal is to integrate out the stochastic process $f(\cdot)$, such that only the joint probability of the state $\boldsymbol{x}_{0:t}$ remains. Unfortunately, we cannot integrate with respect to a function. For a better understanding, a function can be seen as infinite dimensional vector. Hence, a measure would be needed which is translation invariant and locally finite. The only measure that obeys these two properties is the zero measure, assigning zero to every input set, and thus, it is not suitable. Instead, we consider the limit

$$p(\boldsymbol{x}_{0:t}|\boldsymbol{u}_{0:t}, \mathcal{D}) = \lim_{n_f\to\infty}\int\limits_{\mathbb{R}^{n_f}} p(\boldsymbol{x}_{0:t}, \boldsymbol{u}_{0:t}, \mathcal{D}, f(\cdot))df_1 \dots df_{n_f}, \tag{3.5}$$

where $df_1, \dots, df_{n_f}$ is the function differentiation at different points, following [GZ16, Section 2.3.5]. In probabilistic terms, the relation between x_t and $f(x_{t-1})$ is a Dirac distribution

$$p(x_t|x_{t-1}, u_{t-1}, \mathcal{D}, f(\cdot)) = \delta_{\text{dirac}}(x_t - f(x_{t-1})) \tag{3.6}$$

as for a given state x_{t-1} and a realization $f(x_{t-1})$, there exist only one possible next state x_t. Based on the Dirac distribution, we integrate the stochastic process at every point except $\boldsymbol{x}_{0:t}$, denoted by $\backslash\boldsymbol{x}_{0:t}$, to obtain the joint distribution

$$p(\boldsymbol{x}_{0:t}|\boldsymbol{u}_{0:t}, \mathcal{D}) = p(x_0)\lim_{n_f\to\infty}\int\limits_{\mathbb{R}^{n_f}} p(f(\cdot)|\boldsymbol{u}_{0:t}, \mathcal{D})\prod_{j=1}^{t}\delta_{\text{dirac}}(x_j - f(x_{j-1}))df_1 \dots df_{n_f\backslash\boldsymbol{x}_{0:t}}. \tag{3.7}$$

Given the GP property that f is Gaussian distributed for all x_t, see (2.4), we obtain

$$p(\boldsymbol{x}_{0:t}|\boldsymbol{u}_{0:t},\mathcal{D}) = p(x_0)\det\Big((2\pi)^t\check{K}\Big)^{-\frac{1}{2}}\exp\left(-\frac{1}{2}\left(\boldsymbol{x}_{1:t}-\check{\boldsymbol{m}}_{0:t-1}\right)\check{K}^{-1}\left(\boldsymbol{x}_{1:t}-\check{\boldsymbol{m}}_{0:t-1}\right)^\top\right) \quad (3.8)$$

with $\check{K}$ and $\check{\boldsymbol{m}}_{0:t-1}$ as in (3.2) and (3.3), respectively. □

Remark 3.3. *In [Fri16; Mat+16], a similar result is presented but the authors disregard an external input $\boldsymbol{u}_t$ and the conditioning on a training set $\mathcal{D}$. It is our contribution to marginalize out the probabilistic mapping of GP-SSMs with external inputs, such that it is suitable for model-based control approaches.*

The result of Proposition 3.1, given in (3.1), suggests how to compute the joint distribution of a GP-SSM without the need of drawing an infinite dimensional sample. For simulation and model-based control scenarios, the conditional probability distribution for the next state ahead $\boldsymbol{x}_{t+1}$ is often required. The next theorem shows that the distribution of $\boldsymbol{x}_{t+1}$ depends on all past states starting with an initial state $\boldsymbol{x}_0 \in \mathbb{R}^{n_x}$. However, the nature of GPR allows to include the past states as noise-free training data in a way that there exists an analytic closed-form for the prediction.

Theorem 3.1. *Consider a GP-SSM (2.55) with training set $\mathcal{D} = \{X, Y\}$, where Y is corrupted by Gaussian noise $\mathcal{N}(0, \sigma_n^2 I)$. Then, the conditional distribution of the next state ahead $\boldsymbol{x}_{t+1} \in \mathbb{R}^{n_x}$ and output $\boldsymbol{y}_{t+1} \in \mathbb{R}^{n_y}$ is given by*

$$\begin{aligned}\boldsymbol{x}_{t+1}|\boldsymbol{\xi}_{0:t},\mathcal{D} &\sim \mathcal{N}\Big(\boldsymbol{\mu}(\boldsymbol{x}_{t+1}|\boldsymbol{\xi}_{0:t},\mathcal{D}),\Sigma(\boldsymbol{x}_{t+1}|\boldsymbol{\xi}_{0:t},\mathcal{D})\Big)\\ \mu_i(\cdot) &= m(\boldsymbol{\xi}_t) + \boldsymbol{k}(\boldsymbol{\xi}_t,\check{X}_t)^\top\check{K}_t^{-1}\Big([\check{Y}_t]_{:,i} - \boldsymbol{m}(\check{X}_t)\Big)\\ \Sigma_{i,i}(\cdot) &= k(\boldsymbol{\xi}_t,\boldsymbol{\xi}_t) - \boldsymbol{k}(\boldsymbol{\xi}_t,\check{X}_t)^\top\check{K}_t^{-1}\boldsymbol{k}(\boldsymbol{\xi}_t,\check{X}_t)\\ p(\boldsymbol{y}_{t+1}|\boldsymbol{\xi}_{0:t},\boldsymbol{\gamma}_y,\mathcal{D}) &= p(\boldsymbol{y}_{t+1}|\boldsymbol{x}_{t+1},\boldsymbol{\gamma}_y)p(\boldsymbol{x}_{t+1}|\boldsymbol{\xi}_{0:t},\mathcal{D})\end{aligned} \quad (3.9)$$

with $\boldsymbol{x}_0 \in \mathbb{R}^{n_x}$ for all $t \geq 0$ and extended data matrices $\check{X}_t \in \mathbb{R}^{n_\xi \times (n_\mathcal{D}+t)}$, $\check{Y}_t \in \mathbb{R}^{(n_\mathcal{D}+t)\times n_x}$

$$\begin{aligned}&\check{X}_t = X, &&\check{Y}_t = Y &&\textit{if } t = 0\\ &\check{X}_t = [X, \boldsymbol{\xi}_{0:t-1}], &&\check{Y}_t = [Y^\top, \boldsymbol{x}_{1:t}]^\top &&\textit{otherwise.}\end{aligned} \quad (3.10)$$

The Gram matrix $\check{K}_t \in \mathbb{R}^{(n_\mathcal{D}+t)\times(n_\mathcal{D}+t)}$ is defined as

$$\check{K}_t = \begin{cases}\begin{bmatrix} K + \sigma_n^2 I_{n_\mathcal{D}} & K(\boldsymbol{\xi}_{0:t-1}, X)\\ K(\boldsymbol{\xi}_{0:t-1}, X)^\top & K(\boldsymbol{\xi}_{0:t-1}, \boldsymbol{\xi}_{0:t-1})\end{bmatrix}, & \textit{if } t > 0\\ K + \sigma_n^2 I_{n_\mathcal{D}}, & \textit{otherwise.}\end{cases} \quad (3.11)$$

Proof. For the first step, i.e $t = 0$, the conditional distribution is identical to the standard GP regression with predicted mean and variance given by (2.8). For $t > 0$, the current state is fed back to the input, as shown in Fig. 2.9. Following the idea of the previous proof, we condition only on the transitions $\boldsymbol{\xi}_{0:t}$ seen up to that point, instead of conditioning on an infinite-dimensional object. Using (3.1) with the joint Gaussian distribution property (2.4) of the GP, we obtain the joint distribution

$$\begin{bmatrix}(\check{Y}_t)_{:,i}\\ (x_{t+1})_i\end{bmatrix}\sim\mathcal{N}\left(\begin{bmatrix}\boldsymbol{m}(X)\\ \boldsymbol{m}(\boldsymbol{\xi}_{0:t})\end{bmatrix},\begin{bmatrix}\check{K}_t & {K'}_t\\ {K'}_t^\top & k(\boldsymbol{\xi}_t,\boldsymbol{\xi}_t)\end{bmatrix}\right),\tag{3.12}$$

where ${K'}_t^\top=\left[\boldsymbol{k}(\boldsymbol{\xi}_t,X)^\top,\boldsymbol{k}(\boldsymbol{\xi}_t,\boldsymbol{\xi}_{0:t-1})^\top\right]$. Based on (3.12), the conditional probability distribution of the next state ahead $\boldsymbol{x}_{t+1}$ is computed following the approach in appendix A.1. □

Remark 3.4. *The challenge for the subsequent analysis of this approach is that marginalization of $\boldsymbol{f}$ will introduce dependencies across time for the state variable $\boldsymbol{x}_t$ that lead to the loss of the Markovian structure of the state space model.*

Analogously, we derive the prediction for the GP-NOE model.

Theorem 3.2. *Consider a GP-NOE model* (2.56) *with training set $\mathcal{D}=\{X,Y\}$, where the output data Y is corrupted by Gaussian noise $\mathcal{N}(0,\sigma_n^2 I_{n_\mathcal{D}})$. Then, the conditional distribution of the next output $\boldsymbol{y}_{t+1}\in\mathbb{R}^{n_y}$ is given by*

$$\begin{aligned}\boldsymbol{y}_{t+1}|\boldsymbol{\zeta}_{0:t},\mathcal{D}&\sim\mathcal{N}\Big(\boldsymbol{\mu}(\boldsymbol{y}_{t+1}|\boldsymbol{\zeta}_{0:t},\mathcal{D}),\Sigma(\boldsymbol{y}_{t+1}|\boldsymbol{\zeta}_{0:t},\mathcal{D})\Big)\\ \mu_i(\cdot)&=m(\boldsymbol{\zeta}_t)+\boldsymbol{k}(\boldsymbol{\zeta}_t,\check{X}_t)^\top\check{K}_t^{-1}\Big([\check{Y}_t]_{:,i}-\boldsymbol{m}(\check{X}_t)\Big)\\ \Sigma_{i,i}(\cdot)&=k(\boldsymbol{\zeta}_t,\boldsymbol{\zeta}_t)-\boldsymbol{k}(\boldsymbol{\zeta}_t,\check{X}_t)^\top\check{K}_t^{-1}\boldsymbol{k}(\boldsymbol{\zeta}_t,\check{X}_t)\end{aligned}\tag{3.13}$$

with $\boldsymbol{\zeta}_0\in\mathbb{R}^{n_\zeta}$ for all $t\geq 0$ and extended data matrices $\check{X}_t\in\mathbb{R}^{n_\zeta\times(n_\mathcal{D}+t)}$, $\check{Y}_t\in\mathbb{R}^{(n_\mathcal{D}+t)\times n_y}$

$$\begin{array}{lll}\check{X}_t=X, & \check{Y}_t=Y & \textit{if } t=0\\ \check{X}_t=[X,\boldsymbol{\zeta}_{0:t-1}], & \check{Y}_t=[Y^\top,\boldsymbol{y}_{1:t}]^\top & \textit{otherwise.}\end{array}$$

The Gram matrix $\check{K}_t\in\mathbb{R}^{(n_\mathcal{D}+t)\times(n_\mathcal{D}+t)}$ is defined as

$$\check{K}_t=\begin{cases}\begin{bmatrix}K+\sigma_n^2 I_{n_\mathcal{D}} & K(\boldsymbol{\zeta}_{0:t-1},X)\\ K(\boldsymbol{\zeta}_{0:t-1},X)^\top & K(\boldsymbol{\zeta}_{0:t-1},\boldsymbol{\zeta}_{0:t-1})\end{bmatrix}, & \textit{if } t>0\\ K+\sigma_n^2 I_{n_\mathcal{D}}, & \textit{otherwise.}\end{cases}\tag{3.14}$$

Proof. The proof is analogous to the proof of Theorem 3.1. □

3.1.1 The non-Markovian Structure

The previous section shows that the next step ahead state $\boldsymbol{x}_{t+1}$ of a GP-SSM is a sample drawn from a Gaussian distribution with the posterior mean and variance based on the previous states and inputs. This leads to dependencies between the states such that the dynamical model loses the Markov property, i.e., $\boldsymbol{x}_{t+1}$ depends not only on $\boldsymbol{x}_t$ but on all previous states $\boldsymbol{x}_{0:t}$. Since past states are treated as new training points without noise, the size of the extended training set increases with each time step. This results not only in an increasing computing time for the prediction but also in an intractable memory problem for long time simulations. To avoid the non-Markovian structure, GPDMs are often approximated for the simulation of dynamical systems [UBH18]. A standard way is to consider only a constant number of past states instead of the full history $\boldsymbol{x}_{0:t}$, see [BH16a]. In the next step, we introduce the formal description of this approximation for GP-SSMs. First, we define the matrix $\Xi_t^m \in \mathbb{R}^{n_\xi \times \underline{m}}$ of past states and inputs as

$$\Xi_t^m := \begin{cases} \emptyset & \text{if } \overline{m} = 0 \lor t = 0 \\ [\boldsymbol{\xi}_{t-1}, \dots, \boldsymbol{\xi}_{t-\underline{m}}] & \text{otherwise,} \end{cases} \tag{3.15}$$

which are used for the prediction. The *maximum length of memory* $\overline{m} \in \mathbb{N}$ defines how many past states and inputs are considered for the prediction of the next state. The resulting *actual length of memory* $\underline{m} = \min(t, \overline{m})$ is the number of states and inputs which are actually available. The actual length and the maximum length only differ if the number of past states beginning with $\boldsymbol{x}_0$ is less than $\overline{m}$. The prediction of the next state ahead and the output $\boldsymbol{y}_{t+1} \in \mathbb{R}^{n_y}$ is given by

$$\begin{aligned} \boldsymbol{x}_{t+1}^m &\sim \mathcal{N}\Big(\underbrace{\boldsymbol{\mu}(\boldsymbol{x}_{t+1}^m | \boldsymbol{\xi}_t, \Xi_t^m, \mathcal{D})}_{\boldsymbol{f}_t(\boldsymbol{\xi}_t, \Xi_t^m)}, \underbrace{\Sigma(\boldsymbol{x}_{t+1}^m | \boldsymbol{\xi}_t, \Xi_t^m, \mathcal{D})}_{F_t(\boldsymbol{\xi}_t, \Xi_t^m)}\Big) \\ \boldsymbol{y}_{t+1} | \boldsymbol{x}_{t+1}^m &\sim p(\boldsymbol{y}_{t+1} | \boldsymbol{x}_{t+1}^m, \boldsymbol{\gamma}_y). \end{aligned} \tag{3.16}$$

For simplicity in the notation, we introduce the helper functions $\boldsymbol{f}_t \colon \mathbb{R}^{n_\xi} \times \mathbb{R}^{n_\xi \times \underline{m}} \to \mathbb{R}^{n_x}$ and $F_t \colon \mathbb{R}^{n_\xi} \times \mathbb{R}^{n_\xi \times \underline{m}} \to \mathbb{R}^{n_x \times n_x}$. The mean and variance of the i-th element of $\boldsymbol{x}_{t+1}^m$ is given by

$$\begin{aligned} [f_t(\boldsymbol{\xi}_t, \Xi_t^m)]_i =& m(\boldsymbol{\xi}_t) + \boldsymbol{k}(\boldsymbol{\xi}_t, \check{X}_t^m)^\top (\check{K}_t^m)^{-1} \Big([\check{Y}_t]_{:,i} - \boldsymbol{m}(\check{X}_t^m)\Big) \\ [F_t(\boldsymbol{\xi}_t, \Xi_t^m)]_{i,i} =& k(\boldsymbol{\xi}_t, \boldsymbol{\xi}_t) - \boldsymbol{k}(\boldsymbol{\xi}_t, \check{X}_t^m)^\top (\check{K}_t^m)^{-1} \boldsymbol{k}(\boldsymbol{\xi}_t, \check{X}_t^m), \end{aligned} \tag{3.17}$$

respectively. The extended data matrices $\check{X}_t^m \in \mathbb{R}^{n_\xi \times (n_\mathcal{D} + \underline{m})}$, $\check{Y}_t^m \in \mathbb{R}^{(n_\mathcal{D} + \underline{m}) \times n_y}$ are

$$\begin{aligned} &\check{X}_t^m = X, && \check{Y}_t^m = Y && \text{if } \overline{m} = 0 \lor t = 0 \\ &\check{X}_t^m = [X, \boldsymbol{\xi}_{\underline{t}:t-1}], && \check{Y}_t^m = [Y^\top, \boldsymbol{x}_{\underline{t}+1:t}]^\top && \text{otherwise,} \end{aligned} \tag{3.18}$$

with $\underline{t} = \max(0, t - \overline{m})$. The Gram matrix $\check{K}_t^m \in \mathbb{R}^{(n_\mathcal{D} + \underline{m}) \times (n_\mathcal{D} + \underline{m})}$ is given by

$$\check{K}_t^m = \begin{cases} \begin{bmatrix} K(X, X) + \sigma_n^2 I & K(\boldsymbol{\xi}_{\underline{t}:t-1}, X) \\ K(\boldsymbol{\xi}_{\underline{t}:t-1}, X)^\top & K(\boldsymbol{\xi}_{\underline{t}:t-1}, \boldsymbol{\xi}_{\underline{t}:t-1}) \end{bmatrix} & \text{if } t > 0 \land \overline{m} > 0 \\ K(X, X) + \sigma_n^2 I_{n_\mathcal{D}} & \text{otherwise.} \end{cases} \tag{3.19}$$

Note that the prediction in (3.16) is based on the past states and inputs back to the time step $\underline{t}$, see (3.18). In contrast, the prediction of a GP-SSM is based on the full history, see (3.18).

Definition 3.1. We call (3.16) a *Gaussian process approximated state space model* (GP-ASSM) with maximum memory length $\overline{m}$.

Remark 3.5. *For $\overline{m} = \infty$, the prediction depends on all past states, i.e.,*

$$\boldsymbol{x}_{t+1}^\infty \sim \mathcal{N}\Big(\boldsymbol{\mu}(\boldsymbol{x}_{t+1}^\infty | \boldsymbol{\xi}_t, \dots, \boldsymbol{\xi}_0, \mathcal{D}), \Sigma(\boldsymbol{x}_{t+1}^\infty | \boldsymbol{\xi}_t, \dots, \boldsymbol{\xi}_0, \mathcal{D})\Big) \tag{3.20}$$

and thus, equals the true distribution in (3.9) *without Markovian property. The most simple approximation is for* $\overline{m} = 0$

$$\boldsymbol{x}^0_{t+1} \sim \mathcal{N}\Big(\boldsymbol{\mu}(\boldsymbol{x}^0_{t+1}|\boldsymbol{\xi}_t, \mathcal{D}), \Sigma(\boldsymbol{x}^0_{t+1}|\boldsymbol{\xi}_t, \mathcal{D})\Big), \tag{3.21}$$

where the next state ahead is independent of all past states except the current state and input $\boldsymbol{\xi}_t$*. GP-ASSMs with finite maximum length of memory* $\overline{m}$ *are* Markov chains of finite order *as they depend on a finite set of past states and input.*

Figure 3.2 visualizes the relation between actual length $\underline{m}$, the maximum length $\overline{m}$ and the time step $\underline{t}$ of the last state in the memory.

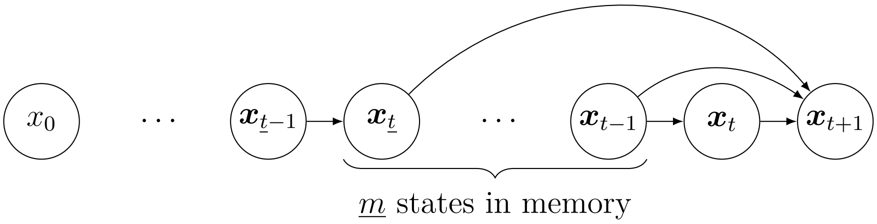

x_0
$\cdots$
$\boldsymbol{x}_{\underline{t}-1}$
$\boldsymbol{x}_{\underline{t}}$
$\cdots$
$\boldsymbol{x}_{t-1}$
$\boldsymbol{x}_t$
$\boldsymbol{x}_{t+1}$
$\underline{m}$ states in memory

Figure 3.2: Time dependencies for the next step ahead state $\boldsymbol{x}_{t+1}$ with the actual length of the memory $\underline{m} = \min(t, m)$ and the last state $\boldsymbol{x}_{\underline{t}}$ in the memory.

Example 3.1. The idea of the presented approximation is visualized in the top plot of Fig. 3.3 by a one-dimensional GP-ASSM with maximum length of memory $\overline{m} = 0$. For the sake of simplicity, the external input is set to zero $u_t = 0$ for all $t \in \mathbb{N}$. The distribution of the next state ahead depends only on the current state x^0_t as it is always sampled from a Gaussian distribution disregarding the history of the past states. Thus, for a given x^0_0, the next state x^0_1 (blue circle) is sampled from a Gaussian distribution (green line), where the mean and variance are based on x^0_0, see (3.21). In the next time step, x^0_2 (blue circle) is sampled from a Gaussian distribution (green line), where the mean and variance are solely based on x^0_1. This procedure is continued for the following time steps. As the distribution (green line) of the next state x^0_{t+1} is independent of the past states $x^0_{t-1}, \dots, x^0_0$, it is always equal to the distribution of the GP (mean and 2-sigma uncertainty) at x^0_t.

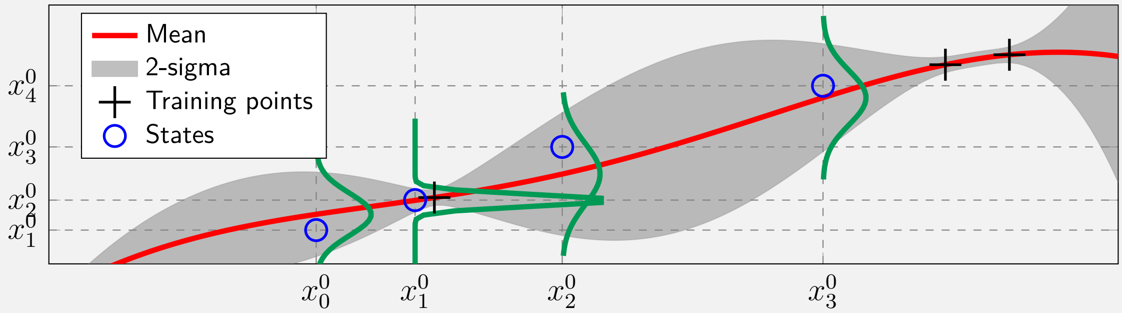


Figure 3.3: Sampling of a one-dimensional GP-ASSM with squared exponential kernel.

In contrast, the true sampling ($\overline{m} = \infty$) with a one-dimensional GP-SSM considers all past states $x^\infty_t, \dots, x^\infty_0$, see (3.20). In Fig. 3.4, we start again with a given x^∞_0. The next state x_1^∞ (blue circle) is sampled from a Gaussian distribution based on the initial state. Then, x_2^∞ is sampled based on x_1^∞ and x_0^∞. For this purpose, the pair (x_0^∞, x_1^∞) is added as noise free training data, see (3.10). Thus, for any following state where $x_t^\infty = x_0^\infty, t \geq 2$, the next state is given by $x_{t+1}^\infty = x_1^\infty$. Due to the dependency on all past states, the distribution of states, which are not yet added as training data, differ from the mean and variance of the GP. This is visualized at the distribution of x_4^∞ (green line) in contrast to the mean (red line) and the 2-sigma uncertainty (gray shaded area) of the GP.

This sampling procedure is necessary since the state mapping f, given by (2.55), can not be drawn directly due to the definition over an infinite set $\mathcal{X} \subseteq \mathbb{R}^{n_x}$. In Fig. 3.4, the mapping f is illustratively drawn (yellow line) over a finite but large number of states.

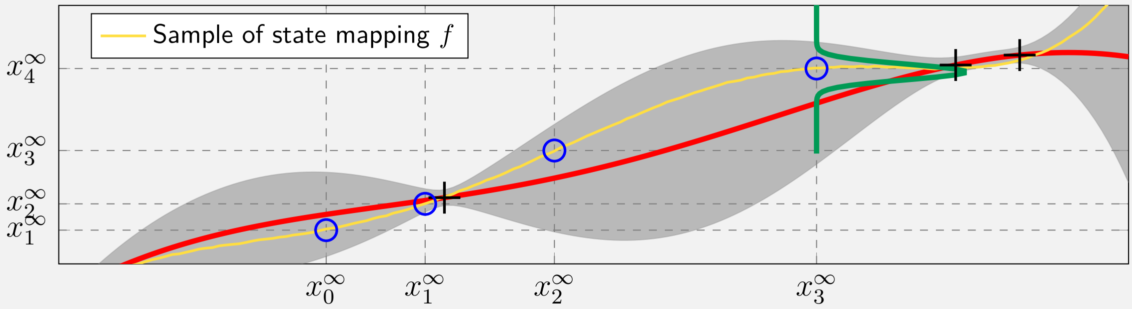


Figure 3.4: Sampling of a one-dimensional GP-SSM with squared exponential kernel.

Remark 3.6. *More advanced strategies for defining the subset of states, that are stored in the memory, are also conceivable. Namely, the same methods as for sparsification of the training data can be exploited. For instance, approaches based on the effective prior [QR05] or pseudo-inputs [SG06] have already been successfully applied for sparsification. In this case, the memory Ξ_t^m of the GP-ASSM would not contain the last $\underline{m}$ states but a selected subset of states based on the sparsification algorithm used. The presented results can be simply adapted.*

In the following, we analogously introduce the formal representation of GP-NOE models. In comparison to GP-SSMs, the GP-NOE models do not have explicitly defined states. Therefore, we define the matrix of past outputs and inputs as

$$\Lambda_t^m = \begin{cases} \emptyset & \text{if } \overline{m} = 0 \vee k = 0 \\ [\boldsymbol{\zeta}_{t-1}, \ldots, \boldsymbol{\zeta}_{t-\underline{m}}] & \text{otherwise,} \end{cases} \tag{3.22}$$

with $\overline{m} \in \mathbb{N}$ defining the maximum length of memory and $\underline{m} = \min(t, \overline{m})$, the actual length of memory. The prediction of the next step ahead output $\boldsymbol{y}_{t+1} \in \mathbb{R}^{n_y}$ is given by

$$\boldsymbol{y}_{t+1}^m \sim \mathcal{N}\Big(\underbrace{\boldsymbol{\mu}(\boldsymbol{y}_{t+1}^m | \boldsymbol{\zeta}_t, \Lambda_t^m, \mathcal{D})}_{\boldsymbol{h}_t(\boldsymbol{\zeta}_t, \Lambda_t^m)}, \underbrace{\Sigma(\boldsymbol{y}_{t+1}^m | \boldsymbol{\zeta}_t, \Lambda_t^m, \mathcal{D})}_{H_t(\boldsymbol{\zeta}_t, \Lambda_t^m)}\Big). \tag{3.23}$$

For simplicity in the notation, we introduce the helper functions $\boldsymbol{h}_t \colon \mathbb{R}^{n_\zeta} \times \mathbb{R}^{n_\zeta \times \underline{m}} \to \mathbb{R}^{n_y}$ and $H_t \colon \mathbb{R}^{n_\zeta} \times \mathbb{R}^{n_\zeta \times \underline{m}} \to \mathbb{R}^{n_y \times n_y}$. The mean $[h_t]_i$ and variance $[H_t]_{i,i}$ of the i-th output dimension is given by

$$\begin{aligned} [h_t]_i =& m(\boldsymbol{\zeta}_t) + \boldsymbol{k}(\boldsymbol{\zeta}_t, \check{X}_t^m)^\top \left(\check{K}_t^m\right)^{-1} ([\check{Y}_t]_{:,i} - \boldsymbol{m}(\check{X}_t^m)) \\ [H_t]_{i,i} =& k(\boldsymbol{\zeta}_t, \boldsymbol{\zeta}_t) - \boldsymbol{k}(\boldsymbol{\zeta}_t, \check{X}_t^m)^\top \left(\check{K}_t^m\right)^{-1} \boldsymbol{k}(\boldsymbol{\zeta}_t, \check{X}_t^m). \end{aligned} \tag{3.24}$$

For GP-NOE models, we define the extended input training set $\check{X}_t^m \in \mathbb{R}^{n_\zeta \times (n_\mathcal{D}+\underline{m})}$ and output training set $\check{Y}_t^m \in \mathbb{R}^{(n_\mathcal{D}+\underline{m}) \times n_y}$ as

$$\begin{aligned} &\check{X}_t^m = X, && \check{Y}_t^m = Y && \text{if } \overline{m} = 0 \vee t = 0 \\ &\check{X}_t^m = [X, \boldsymbol{\zeta}_{\underline{t}:t-1}], && \check{Y}_t^m = [Y^\top, \boldsymbol{y}_{\underline{t}+1:t}]^\top && \text{otherwise,} \end{aligned} \tag{3.25}$$

with $\underline{t} = \max(0, t - \overline{m})$ and the Gram matrix $\check{K}_t^m \in \mathbb{R}^{(n_\mathcal{D}+\underline{m}) \times (n_\mathcal{D}+\underline{m})}$ as

$$\check{K}_t^m = \begin{cases} \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(\boldsymbol{\zeta}_{\underline{t}:t-1}, X) \\ K(\boldsymbol{\zeta}_{\underline{t}:t-1}, X)^\top & K(\boldsymbol{\zeta}_{\underline{t}:t-1}, \boldsymbol{\zeta}_{\underline{t}:t-1}) \end{bmatrix} & \text{if } t > 0 \wedge \overline{m} > 0 \\ K(X,X) + \sigma_n^2 I_{n_\mathcal{D}} & \text{otherwise.} \end{cases} \tag{3.26}$$

Definition 3.2. We call (3.23) a *Gaussian process approximated nonlinear output error* (GP-ANOE) model with maximum memory length $\overline{m}$.

Having introduced the formal description for the approximations of the non-Markovian dynamics, we analyze the approximation error in the following.

3.1.2 Approximation Error

In this section, we present the computation of the error between the true state distribution $\boldsymbol{x}_{t+1}$ given by (3.9) and the approximated distribution $\boldsymbol{x}_{t+1}^m$ based on the maximum length of memory $\overline{m}$. As the Kullback-Leibler (KL) divergence is an important measure of how one probability distribution differs from a second, we start with the KL divergence of the GP-SSM prediction from the GP-ASSM prediction.

Proposition 3.2. *Consider a GP-ASSM with maximum length of memory $\overline{m} \in \mathbb{N}$ and data set $\mathcal{D}$ such that*

$$\boldsymbol{x}_{t+1}^m \sim \mathcal{N}\Big(\boldsymbol{\mu}(\boldsymbol{x}_{t+1}|\boldsymbol{\xi}_t, \Xi_t^m, \mathcal{D}), \Sigma(\boldsymbol{x}_{t+1}|\boldsymbol{\xi}_t, \Xi_t^m, \mathcal{D})\Big)$$

with $\boldsymbol{x}_0 \in \mathbb{R}^{n_x}$. For given past states and inputs $\boldsymbol{\xi}_{0:t}$, where $\boldsymbol{\xi}_t \neq \boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_{t-1}$, the KL-divergence of the true distribution $\boldsymbol{x}_{t+1}$ from the approximation $\boldsymbol{x}_{t+1}^m$ is given by

$$\begin{aligned} d_{\mathrm{KL}}(\boldsymbol{x}_{t+1} \| \boldsymbol{x}_{t+1}^m) = &\frac{1}{2}\Delta_t^\top F_t(\boldsymbol{\xi}_t, \Xi_t^m)^{-1}\Delta_t - n_x + \operatorname{tr}\left(F_t(\boldsymbol{\xi}_t, \Xi_t^\infty)F_t(\boldsymbol{\xi}_t, \Xi_t^m)^{-1}\right) \\ &+ \ln\left(\operatorname{tr}\left(F_t(\boldsymbol{\xi}_t, \Xi_t^\infty)^{-1}F_t(\boldsymbol{\xi}_t, \Xi_t^m)\right)\right) \end{aligned} \tag{3.27}$$

with $\Delta_t = \boldsymbol{f}_t(\boldsymbol{\xi}_t, \Xi_t^m) - \boldsymbol{f}_t(\boldsymbol{\xi}_t, \Xi_t^\infty)$.

Proof. For given past states and inputs $\boldsymbol{\xi}_{0:t}$, the next state $\boldsymbol{x}_{t+1}$ of the GP-SSM and the next state $\boldsymbol{x}_{t+1}^m$ of the GP-ASSM are Gaussian distributed such that the KL-divergence is

$$\begin{aligned} d_{\mathrm{KL}}(\boldsymbol{x}_{t+1} \| \boldsymbol{x}_{t+1}^m) = &\frac{1}{2}\Bigg[\operatorname{tr}\left(F_t(\boldsymbol{\xi}_t, \Xi_t^m)^{-1}F_t(\boldsymbol{\xi}_t, \Xi_t^\infty)\right) + \left(\boldsymbol{f}_t(\boldsymbol{\xi}_t, \Xi_t^m) - \boldsymbol{f}_t(\boldsymbol{\xi}_t, \Xi_t^\infty)\right)^\top F_t(\boldsymbol{\xi}_t, \Xi_t^m)^{-1} \\ &\left(\boldsymbol{f}_t(\boldsymbol{\xi}_t, \Xi_t^m) - \boldsymbol{f}_t(\boldsymbol{\xi}_t, \Xi_t^\infty)\right) - n_x + \ln\left(\frac{\det F_t(\boldsymbol{\xi}_t, \Xi_t^m)}{\det F_t(\boldsymbol{\xi}_t, \Xi_t^\infty)}\right)\Bigg] \end{aligned} \tag{3.28}$$

using the definition of $F_t, \boldsymbol{f}_t$ in (3.16). As the variance of each element in $\boldsymbol{x}_{t+1}$ and $\boldsymbol{x}^m_{t+1}$ is independent, see (3.17), the KL-divergence can be rewritten to

$$\begin{aligned} d_{\mathrm{KL}}(\boldsymbol{x}_{t+1} \| \boldsymbol{x}^m_{t+1}) = \frac{1}{2} \sum_{i=1}^{n_x} \Bigg(& \frac{[F_t(\boldsymbol{\xi}_t, \Xi^\infty_t)]_{i,i} + ([f_t(\boldsymbol{\xi}_t, \Xi^m_t)]_i - [f_t(\boldsymbol{\xi}_t, \Xi^\infty_t)]_i)^2}{[F_t(\boldsymbol{\xi}_t, \Xi^m_t)]_{i,i}} \\ & + \ln \left(\frac{[F_t(\boldsymbol{\xi}_t, \Xi^m_t)]_{i,i}}{[F_t(\boldsymbol{\xi}_t, \Xi^\infty_t)]_{i,i}} \right) - 1 \Bigg). \end{aligned} \tag{3.29}$$

Finally, simplifying (3.29) leads to (3.27). □

Proposition 3.2 shows that the error is quantified by the drift of mean $\boldsymbol{\mu}(\boldsymbol{x}_{t+1} | \boldsymbol{\xi}_t, \Xi^m_t, \mathcal{D})$ and variance $\Sigma(\boldsymbol{x}_{t+1} | \boldsymbol{\xi}_t, \Xi^m_t, \mathcal{D})$ with respect to the true distribution. Therefore, depending on the maximum length of memory $\overline{m}$, the approximation error is zero at the beginning as the following corollary points out.

Corollary 3.1. *For all $t \leq \overline{m}$, the approximated distribution $p(\boldsymbol{x}_{t+1} | \boldsymbol{\xi}_t, \Xi^m_t, \mathcal{D})$, given by* (3.16)*, equals the true distribution* (2.55) *with KL-divergence $d_{\mathrm{KL}}(\boldsymbol{x}_{t+1} \| \boldsymbol{x}^m_{t+1}) = 0$.*

Proof. The corollary is a direct consequence of Proposition 3.2. If the time step t is equal to or less than the maximum length of memory $\overline{m}$, the matrices of past states and inputs of the GP-SSM and the GP-ASSM is identical, i.e., $\Xi^m_t = \Xi^\infty_t$, and thus, the mean and variance of the approximated distribution equals the true distribution. In consequence, the KL-divergence is zero given by (3.29). □

The restriction of Proposition 3.2 that the current state must not be part of the past states is necessary as otherwise, the variance $F_t(\boldsymbol{\xi}_t, \Xi^\infty_t)$ or $F_t(\boldsymbol{\xi}_t, \Xi^m_t)$ would be zero. In Example 3.1, this case is explained as the past states are added to the extended data set such that the predicted variance becomes zero. Additionally, the asymmetry of the KL divergence might be obstructive in some applications. Therefore, we introduce a different measure, namely the mean square prediction error (MSPE).

Proposition 3.3. *Consider a GP-ASSM with maximum memory length $\overline{m} \in \mathbb{N}$ and data set $\mathcal{D}$ such that*

$$\boldsymbol{x}^m_{t+1} \sim \mathcal{N}\Big(\boldsymbol{\mu}(\boldsymbol{x}_{t+1} | \boldsymbol{\xi}_t, \Xi^m_t, \mathcal{D}), \Sigma(\boldsymbol{x}_{t+1} | \boldsymbol{\xi}_t, \Xi^m_t, \mathcal{D}) \Big)$$

with $\boldsymbol{x}_0 \in \mathbb{R}^{n_x}$. For given past states and inputs $\boldsymbol{\xi}_{0:t}$, the MSPE between $\boldsymbol{x}^m_{t+1}$ and $\boldsymbol{x}_{t+1}$ of the GP-SSM is given by

$$\mathrm{E}\left[\left\| \boldsymbol{x}_{t+1} - \boldsymbol{x}^m_{t+1} \right\|^2 \right] = \| \boldsymbol{f}_t(\boldsymbol{\xi}_t, \Xi^\infty_t) - \boldsymbol{f}_t(\boldsymbol{\xi}_t, \Xi^m_t) \| + \operatorname{tr}\left(F_t(\boldsymbol{\xi}_t, \Xi^\infty_t) + F_t(\boldsymbol{\xi}_t, \Xi^m_t) \right). \tag{3.30}$$

Proof. Since each element of $\boldsymbol{x}_{t+1}$ and $\boldsymbol{x}^m_{t+1}$ with a given history of past states and inputs $\boldsymbol{\xi}_{0:t}$ is Gaussian distributed, the MSPE is defined by

$$\begin{aligned} \mathrm{E}\left[\left\| \boldsymbol{x}_{t+1} - \boldsymbol{x}^m_{t+1} \right\|^2 \right] &= \sum_{i=1}^{n_x} \mathrm{E}\left[(x_{t+1,i} - x^m_{t+1,i})^2 \right] \qquad (3.31) \\ &= \sum_{i=1}^{n_x} \Big([f_t(\boldsymbol{\xi}_t, \Xi^\infty_t)]_i - [f_t(\boldsymbol{\xi}_t, \Xi^m_t)]_i \Big)^2 + [F_t(\boldsymbol{\xi}_t, \Xi^\infty_t)]_{i,i} + [F_t(\boldsymbol{\xi}_t, \Xi^m_t)]_{i,i}. \end{aligned}$$

Equation (3.31) is then rewritten to (3.30). □

With Propositions 3.2 and 3.3 the error of the approximation can be computed. Even if the error measures do not decrease in general for increasing maximum length of memory $\overline{m}$, the behavior of the variance can be quantified. The next proposition allows to overestimate the predicted variance based on the maximum length of memory.

Proposition 3.4. *Consider two GP-ASSMs with states and inputs* $\boldsymbol{\xi}_{0:t} \in \mathbb{R}^{n_\xi}$ *with* $\boldsymbol{\xi}_0 \neq \boldsymbol{\xi}_1 \neq \ldots \neq \boldsymbol{\xi}_t$ *such that*

$$\begin{aligned} \boldsymbol{x}_{t+1}^{m} &\sim \mathcal{N}\Big(\boldsymbol{f}_t(\boldsymbol{\xi}_t, \Xi_t^m), F_t(\boldsymbol{\xi}_t, \Xi_t^m)\Big) \\ \boldsymbol{x}_{t+1}^{m'} &\sim \mathcal{N}\Big(\boldsymbol{f}_t(\boldsymbol{\xi}_t, \Xi_t^{m'}), F_t(\boldsymbol{\xi}_t, \Xi_t^{m'})\Big), \end{aligned}$$

where $\overline{m}$ *and* $\overline{m}'$ *are the maximum length of memory, respectively. Then, for* $\overline{m}' > \overline{m}$

$$\operatorname{tr}\Big(\Sigma(\boldsymbol{x}_{t+1}^{m'} | \boldsymbol{\xi}_t, \Xi_t^{m'}, \mathcal{D})\Big) < \operatorname{tr}\Big(\Sigma(\boldsymbol{x}_{t+1}^{m} | \boldsymbol{\xi}_t, \Xi_t^{m}, \mathcal{D})\Big) \tag{3.32}$$

holds for all $t \in \mathbb{N}$ *with* $t > \overline{m}$.

Proof. Following (3.9), the variance for each component of the predicted state of a GP-ASSM is given by

$$\operatorname{var}(x_{t+1,i}^{m} | \boldsymbol{\xi}_t, \Xi_t^m, \mathcal{D}) = k(\boldsymbol{\xi}_t, \boldsymbol{\xi}_t) - \boldsymbol{k}(\boldsymbol{\xi}_t, \check{X}_t^m)^\top (\check{K}_t^m)^{-1} \boldsymbol{k}(\boldsymbol{\xi}_t, \check{X}_t^m). \tag{3.33}$$

The Gram matrix $\check{K}_t^m$ is positive definite and from (3.19) we know, that its dimension is $(n_\mathcal{D} + \underline{m}) \times (n_\mathcal{D} + \underline{m})$. Based on $\check{K}_t^m$, the Gram matrix $\check{K}_t^{m'} \in \mathbb{R}^{(n_\mathcal{D} + \underline{m}') \times (n_\mathcal{D} + \underline{m}')}$ is determined as

$$\check{K}_t^{m'} = \begin{bmatrix} K(\boldsymbol{\xi}_{\underline{t}':\underline{t}-1}, \boldsymbol{\xi}_{\underline{t}':\underline{t}-1}) & K(\boldsymbol{\xi}_{\underline{t}':\underline{t}-1}, X) \\ K(\boldsymbol{\xi}_{\underline{t}':\underline{t}-1}, X)^\top & \check{K}_t^m, \end{bmatrix} \tag{3.34}$$

where $\underline{t} = \max(0, t - \overline{m})$ and $\underline{t}' = \max(0, t - \overline{m}')$. Since the $\check{K}_t^{m'}$ is also positive definite and $\overline{m}' > \overline{m}$, the inequality

$$\begin{aligned} k(\boldsymbol{\xi}_t, \boldsymbol{\xi}_t) - \boldsymbol{k}(\boldsymbol{\xi}_t, \check{X}_t^{m'})^\top (\check{K}_t^{m'})^{-1} \boldsymbol{k}(\boldsymbol{\xi}_t, \check{X}_t^{m'}) &< k(\boldsymbol{\xi}_t, \boldsymbol{\xi}_t) - \boldsymbol{k}(\boldsymbol{\xi}_t, \check{X}_t^{m})^\top (\check{K}_t^{m})^{-1} \boldsymbol{k}(\boldsymbol{\xi}_t, \check{X}_t^{m}) \\ \Rightarrow \operatorname{var}(x_{t+1,i}^{m'} | \boldsymbol{\xi}_t, \Xi_t^{m'}, \mathcal{D}) &< \operatorname{var}(x_{t+1,i}^{m} | \boldsymbol{\xi}_t, \Xi_t^{m}, \mathcal{D}) \end{aligned} \tag{3.35}$$

holds for all $t \in \mathbb{N}$ with $t > \overline{m}$. Summing up (3.35) over all elements of $\boldsymbol{x}_{t+1}$ leads to (3.32). □

Proposition 3.4 verifies that the variance of the distribution for the next state ahead $\boldsymbol{x}_{t+1}^{m'}$ is less than the variance of $\boldsymbol{x}_{t+1}^{m}$ with a shorter actual length of memory. This induces that the variance is the lowest for the true sampling as it is given for $\overline{m} = \infty$. The restriction $t > m$ in Proposition 3.4 is necessary as otherwise the variances would be equal for $t \leq \overline{m}$ as explained in Corollary 3.1. The inequality of past states is necessary to ensure that the GP-ASSM with maximum length of memory $\overline{m}'$ contains not only a multiple of the same states which would not decrease the variance. For the sake of completeness, a weaker description for all $t \in \mathbb{N}$ is provided by the following corollary.

Corollary 3.2. *Consider two GP-ASSMs with states and inputs* $\boldsymbol{\xi}_{0:t} \in \mathbb{R}^{n_\xi}$ *such that*

$$\boldsymbol{x}_{t+1}^{m} \sim \mathcal{N}\Big(\boldsymbol{f}_t(\boldsymbol{\xi}_t, \Xi_t^m), F_t(\boldsymbol{\xi}_t, \Xi_t^m)\Big)$$
$$\boldsymbol{x}_{t+1}^{m'} \sim \mathcal{N}\Big(\boldsymbol{f}_t(\boldsymbol{\xi}_t, \Xi_t^{m'}), F_t(\boldsymbol{\xi}_t, \Xi_t^{m'})\Big)$$

where $\overline{m}$ *and* $\overline{m}'$ *are the maximum length of memory, respectively. Then, for* $\overline{m}' > \overline{m}$

$$\operatorname{tr}\Big(\Sigma(\boldsymbol{x}_{t+1}^{m'} | \boldsymbol{\xi}_t, \Xi_t^{m'}, \mathcal{D})\Big) \leq \operatorname{tr}\Big(\Sigma(\boldsymbol{x}_{t+1}^{m} | \boldsymbol{\xi}_t, \Xi_t^{m}, \mathcal{D})\Big)$$

holds for all $t \in \mathbb{N}$.

Proof. The corollary is a direct consequence of Proposition 3.4 since as long as the current time step t is less than the maximum length of memory $\overline{m}$, the variance of $\boldsymbol{x}_{t+1}^{m}$ and $\boldsymbol{x}_{t+1}^{m'}$ is identical as shown in Corollary 3.1. □

In the next example, a comparison of the presented error measures and the behavior of the variance is presented.

Example 3.2. In Fig. 3.5, the distributions (red shaded) for the next state ahead x_{t+1}^m depending on the maximum length of memory $\overline{m}$ for a given trajectory $x_0, \dots, x_3$ (red circles) is shown. We use here a one-dimensional GP-ASSM with squared exponential function. For sake of simplicity, the input is set to zero, i.e., $u_t = 0$ for all $t \in \mathbb{N}$. With increasing maximum length of memory $\overline{m}$, the variance of the distributions (red shaded) decreases as stated in Proposition 3.4. For $\overline{m} = 3$, the distribution is equal to the true distribution (black dashed) as stated in Corollary 3.1. Table 3.1 shows the computed KL-divergence, the MSPE and the variance of x_4^m per maximum length of memory $\overline{m}$.

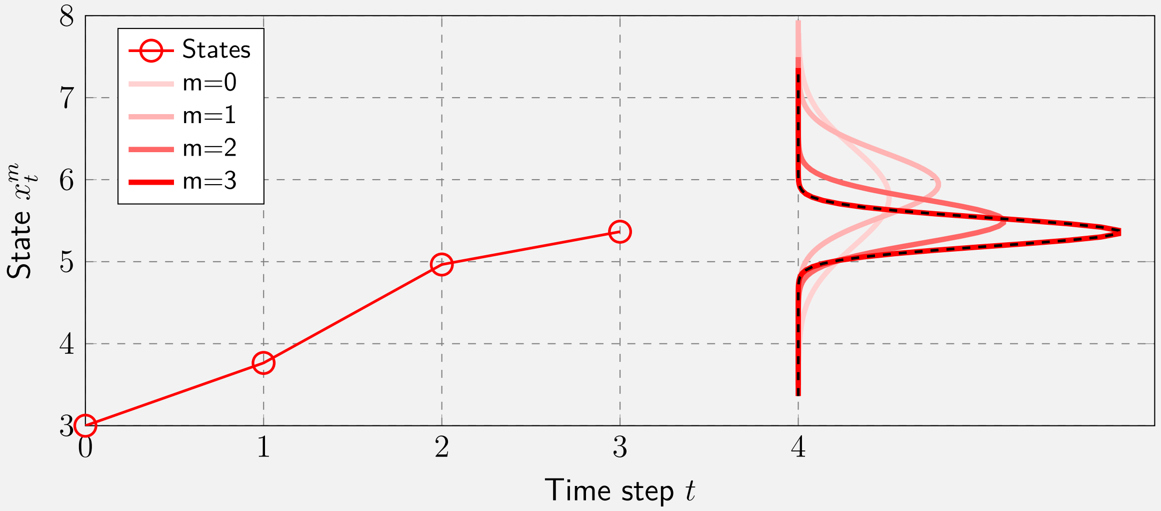


Figure 3.5: The distribution for the next state ahead x_t^m depending on the maximum length of memory $\overline{m}$.

	$\overline{m}=0$	$\overline{m}=1$	$\overline{m}=2$	$\overline{m}=3$
$d_{\mathrm{KL}}(x_4 \Vert x_4^m)$	2.1131	3.0811	0.5559	0
$\mathrm{MSPE}(x_4, x_4^m)$	0.5720	0.5190	0.1171	0.0575
$\Sigma(x_4^m \vert \boldsymbol{\xi}_3, \Xi_3^m, \mathcal{D})$	0.3620	0.1519	0.0706	0.0288

Table 3.1: Comparison of the KL-divergence, MSPE and variance Σ for different maximum lengths of memory $\overline{m}$.

So far, we obtained a method for sampling from the non-Markovian GP-SSM and introduced the approximated GP-ASSM which is a Markov chain of finite order. This approximation allows to use GP-ASSMs like parametric dynamical models since the state dependencies across time are removed. The approximation error is analyzed based on different measures and illustrated in Example 3.2.

Remark 3.7. *This section focuses on the formal development of GP-ASSMs, but the results are also directly applicable to GP-ANOE models. In this case, the proofs are analogously but with the output $\boldsymbol{y}_t$ as regressor.*

3.2 Boundedness of GPDMs

After the introduction of GP-SSMs and GP-ASSMs, the models are analyzed in terms of boundedness. Furthermore, the relation of the boundedness properties between the true and the approximated distribution are investigated.

3.2.1 GP State Space Models

We start with the general introduction of the boundedness of GP-ASSMs for bounded mean functions and kernels.

Theorem 3.3. *Consider a GP-ASSM* (3.16) *with maximum memory length $\overline{m}$ and bounded mean function and kernel, i.e., $\Vert \boldsymbol{m}(\boldsymbol{\xi})\Vert < \infty, k(\boldsymbol{\xi}, \boldsymbol{\xi}') < \infty$, respectively, for all $\boldsymbol{\xi}, \boldsymbol{\xi}' \in \mathbb{R}^{n_\xi}$. Then, the expected value of the sequence $\{\boldsymbol{x}_t^m\}, t \in \mathbb{N}$ given by*

$$\boldsymbol{x}_{t+1}^m \sim \mathcal{N}\Big(\boldsymbol{\mu}(\boldsymbol{x}_{t+1}^m \vert \boldsymbol{\xi}_t, \Xi_t^m, \mathcal{D}), \Sigma(\boldsymbol{x}_{t+1}^m \vert \boldsymbol{\xi}_t, \Xi_t^m, \mathcal{D})\Big) \tag{3.36}$$

is ultimately p-bounded *by*

$$\sup_{t \in \mathbb{N}} \mathrm{E}\left[\Vert \boldsymbol{x}_t^m \Vert^p\right] < \infty$$

for all $\boldsymbol{x}_0 \in \mathbb{R}^{n_x}, \forall p \in \mathbb{N}$.

Proof. We start with the computation of the expected value for a one-dimensional GP-SSM, which equals a GP-ASSM with $\overline{m} = \infty$, as for any other $\overline{m}$ the number of considered past states is reduced. For this purpose, we integrate over the joint probability distribution given by (3.1). That leads to

$$\begin{aligned}\sup_{t\in\mathbb{N}} \mathrm{E}[|x_t^p|] &= \sup_{t\in\mathbb{N}} \int\limits_{\mathbb{R}^t} |x_t^p| p(\boldsymbol{x}_{0:t}|\boldsymbol{u}_{0:t},\mathcal{D}) d\boldsymbol{x}_{0:t} \\ &= c_1 \sup_{t\in\mathbb{N}} \int\limits_{\mathbb{R}^t} |x_t^p| \exp\left(-\frac{1}{2}\,\boldsymbol{\mu}_t^\top\, \check{K}^{-1}\,\boldsymbol{\mu}_t\right) d\boldsymbol{x}_{0:t} \end{aligned} \tag{3.37}$$

with $\boldsymbol{\mu}_t = \boldsymbol{x}_{1:t} - \check{\boldsymbol{m}}_{0:t-1}$, constant $c_1 \in \mathbb{R}$ and $\check{\boldsymbol{m}}_{0:t-1}, \check{K}$ given by (3.2) and (3.3). The matrix $\check{K}$ is positive definite, symmetric and the eigenvalues are positive and bounded, i.e., $\underline{\lambda}(\check{K}) > 0, \bar{\lambda}(\check{K}) < \infty$ for all $\boldsymbol{\xi}_{0:t-1} \in \mathbb{R}^{n_\xi \times t}$. The expression $\underline{\lambda}(\cdot)$ and $\bar{\lambda}(\cdot)$ describe the minimum and maximum eigenvalue of a matrix, respectively. Thus, there exists a positive definite and symmetric matrix $\overline{K} \in \mathbb{R}^{t\times t}$, such that we can upper bound the expected value to

$$\sup_{t\in\mathbb{N}} \mathrm{E}[|x_t^p|] \leq c_1 \sup_{t\in\mathbb{N}} \int\limits_{\mathbb{R}^t} |x_t^p| \exp\left(-\frac{1}{2}\,\boldsymbol{\mu}_t^\top\, \overline{K}\,\boldsymbol{\mu}_t\right) d\boldsymbol{x}_{0:t}. \tag{3.38}$$

As $\check{\boldsymbol{m}}_{0:t-1}$ is given by

$$\check{\boldsymbol{m}}_{0:t-1} = \boldsymbol{m}(\boldsymbol{\xi}_{0:t-1}) + K(\boldsymbol{\xi}_{0:t-1}, X)^\top (K(X,X) + \sigma_n^2 I)^{-1}(Y - \boldsymbol{m}(X)),$$

following (3.3), and the mean function $\boldsymbol{m}$ and kernel k are bounded, it follows that each element of $\boldsymbol{m}_{0:t-1}$ is bounded, i.e., $m_i \leq c_2 \in \mathbb{R}_{>0}, \forall i = 0,\dots,t-1$. Note that this bound only depends on the previous time steps and is independent of time step t. Consequently, there exists a vector $\boldsymbol{c}_t \in \mathbb{R}^t$ such that (3.38) can be rewritten as

$$\begin{aligned}\int\limits_{\mathbb{R}^t} |x_t^p| \exp\left(-\frac{1}{2}\,\boldsymbol{\mu}_t^\top\,\overline{K}\,\boldsymbol{\mu}_t\right) d\boldsymbol{x}_{0:t} &\leq \int\limits_{\mathbb{R}^t} |x_t^p| \exp\left(-\frac{1}{2}(\boldsymbol{x}_{1:t} - \boldsymbol{c}_t)^\top \overline{K}(\boldsymbol{x}_{1:t} - \boldsymbol{c}_t)\right) d\boldsymbol{x}_{0:t} \\ &\leq c_3 \in \mathbb{R}_{>0}, \end{aligned} \tag{3.39}$$

and, thus, $\sup_{t\in\mathbb{N}} \mathrm{E}[|x_t^p|] < \infty$. As a multi-dimensional GP-SSM depends on separated GPs and the Gram matrix $\check{K}$ remains bounded, (3.39) can be extended to $\boldsymbol{x}_t \in \mathbb{R}^{n_x}$. Consequently, $\sup_{t\in\mathbb{N}} \mathrm{E}\,\|\boldsymbol{x}_t\|^p < \infty$ holds. Finally, this remains obviously true for GP-ASSMs with $\overline{m} < \infty$, since only a subset of past states is considered. This concludes the proof. □

Remark 3.8. *Note that no boundedness of the input $\boldsymbol{u}_t$ is required for Theorem 3.3.*

Remark 3.9. *Many commonly used kernels for GPDMs are bounded, for instance, the squared exponential or Matérn kernel.*

Theorem 3.3 shows the boundedness of GP-ASSMs for bounded mean function and kernel, which holds for the true as well as for the approximated distribution. However, it is also possible that a GP-ASSM with unbounded kernel leads to bounded dynamics. This mainly depends on the training data. In this case, the boundedness property might be lost for a different maximum length of memory, as the following proposition states.

Theorem 3.4. *Consider two GP-ASSMs with states* $\boldsymbol{x}_t^m$ *and* $\boldsymbol{x}_t^{m'}$*, respectively, such that*

$$\begin{aligned}\boldsymbol{x}_{t+1}^m &\sim \mathcal{N}\Big(\boldsymbol{f}_t(\boldsymbol{\xi}_t, \Xi_t^m), F_t(\boldsymbol{\xi}_t, \Xi_t^m)\Big)\\ \boldsymbol{x}_{t+1}^{m'} &\sim \mathcal{N}\Big(\boldsymbol{f}_t(\boldsymbol{\xi}_t', \Xi_t^{m'}), F_t(\boldsymbol{\xi}_t', \Xi_t^{m'})\Big),\end{aligned}$$

where $\overline{m}$ *and* $\overline{m}'$ *are the maximum length of memory. Then, for* $\overline{m} < \overline{m}'$

$$\sup_{t\in\mathbb{N}} \mathrm{E}\left[\|\boldsymbol{x}_t^m\|^p\right] < \infty \nRightarrow \sup_{t\in\mathbb{N}} \mathrm{E}\left[\|\boldsymbol{x}_t^{m'}\|^p\right] < \infty \tag{3.40}$$

holds for any $p \in \mathbb{N}$ *and* $\boldsymbol{x}_0^m = \boldsymbol{x}_0^{m'} \in \mathbb{R}^{n_x}$.

Proof. We use a counter example to prove this theorem. Consider a one-dimensional GP-ASSM with $\overline{m} = 0$ and linear kernel as introduced in (2.44). We assume two training points at $X_1 = [-1; 0], X_2 = [1; 0]$ and $Y = [Y_1, Y_2] \in \mathbb{R}^2$ with noise $\sigma_n^2 = 1$ and input $u_t = 0$. Using the definition of (3.16), the mean f_t and variance F_t of next state x_{t+1}^0 is given by

$$f_t(\boldsymbol{\xi}_t, \emptyset) = \frac{1}{3} x_t^0 (Y_2 - Y_1),\ F_t(\boldsymbol{\xi}_t, \emptyset) = \frac{1}{3} (x_t^0)^2. \tag{3.41}$$

For $|Y_2 - Y_1| \leq 3$, the sequence $\{x_t^0\}, t \in \mathbb{N}$ is p-bounded, since $x^0 = 0$ is stochastically asymptotically stable in the large. Next, in an alternative GP-ASSM, we use the same training points with $\overline{m}' \geq 1$. Starting at $x_0^{m'} \in \mathbb{R}\backslash 0$, the distribution of $x_1^{m'}$ can be computed using (3.41). With a Gaussian distributed sampled $x_1^{m'}$, the next step state $x_{t+1}^{m'}$ for $t \geq 1$ are given by

$$\begin{aligned} f_t\left(\begin{bmatrix} x_t^{m'} \\ 0 \end{bmatrix}, \Xi_t^{m'}\right) = \frac{x_1^{m'}}{x_0^{m'}} x_t^{m'},\ F_t\left(\begin{bmatrix} x_t^{m'} \\ 0 \end{bmatrix}, \Xi_t^{m'}\right) = 0 \\ x_{t+1}^{m'} = \frac{x_1^{m'}}{x_0^{m'}} x_t^{m'}. \end{aligned} \tag{3.42}$$

The predicted variance for all states in the future is zero, since the state $x_1^{m'}$ exactly defines a sample of the GP with a linear kernel. The reason is that a linear function is fully defined by one point unequal zero. Based on the Gaussian distribution of $x_1^{m'}$, the probability, that a trajectory of (3.42) is unbounded, is computed by

$$\mathrm{P}\left(\left|\frac{x_1^{m'}}{x_0^{m'}}\right| > 1\right) = 1 + \mathrm{cdf}\left(\frac{-3\left|x_0^{m'}\right| + \Delta Y}{[x_0^{m'}]^2}\right) - \mathrm{cdf}\left(\frac{3\left|x_0^{m'}\right| + \Delta Y}{[x_0^{m'}]^2}\right), \tag{3.43}$$

where $\Delta Y = Y_1 - Y_2$ and $\mathrm{cdf}(\cdot)$ denotes the standard normal cumulative distribution function. Since the probability (3.43) is greater than zero, the sequence $\{x_t^{m'}\}, t \in \mathbb{N}$ is not p-bounded. Hence, a different maximum length of memory $\overline{m}$ of a GP-ASSM might lead to a loss boundedness property as stated in Theorem 3.4. □

Example 3.3. Figure 3.6 visualizes the counter example from the proof of Theorem 3.4 based on a GP-ASSM with linear kernel as introduced in (2.44). Although the samples of the GP-ASSM with $\overline{m} = 0$ are bounded (top), a GP-ASSM with $\overline{m}' = 10$ (bottom) shows unbounded trajectories, which leads to an unbounded mean and variance.

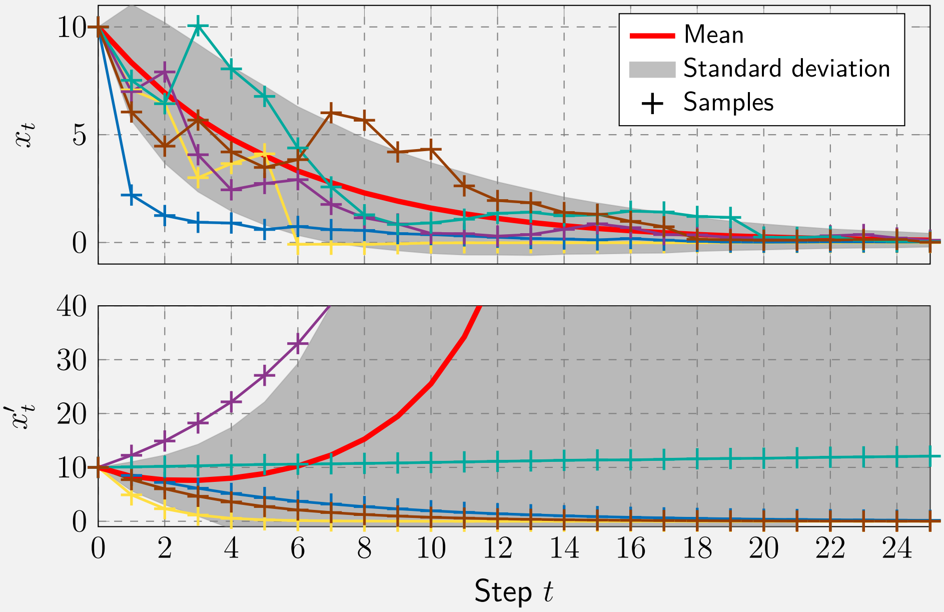


Figure 3.6: The GP-ASSM with $\overline{m} = 0$ (top) results in bounded system trajectories whereas a GP-ASSM with $\overline{m}' \geq 1$ (bottom) generates unbounded trajectories. Therefore the boundedness property is lost for different maximum lengths of memory $\overline{m}$

The following theorem shows the relationship between the boundedeness of GP-ASSMs with different length of memory. It states that the approximated dynamics given by a GP-ASSM is bounded if the dynamics of the GP-SSM is bounded. Thus, it allows to use the approximation in control settings without losing the boundedness, which is important for the robustness and stability analysis. Note that, in contrast to Theorem 3.3, the kernel is not required to be bounded.

Theorem 3.5. *Consider two GP-ASSMs with states $\boldsymbol{x}_t^m$ and $\boldsymbol{x}_t^{m'}$,respectively, such that*

$$\boldsymbol{x}_{t+1}^m \sim \mathcal{N}\Big(\boldsymbol{f}_t(\boldsymbol{\xi}_t, \Xi_t^m), F_t(\boldsymbol{\xi}_t, \Xi_t^m)\Big)$$
$$\boldsymbol{x}_{t+1}^{m'} \sim \mathcal{N}\Big(\boldsymbol{f}_t(\boldsymbol{\xi}'_t, \Xi_t^{m'}), F_t(\boldsymbol{\xi}'_t, \Xi_t^{m'})\Big)$$

where $\overline{m}$ and $\overline{m}'$ are the maximum length of memory. Then, if $\overline{m} < \overline{m}'$,

$$\sup_{t\in\mathbb{N},\boldsymbol{x}_0^{m'}\in\mathbb{R}^{n_x}} \mathrm{E}\left[\|\boldsymbol{x}_t^{m'}\|^p\right] < \infty \Rightarrow \sup_{t\in\mathbb{N},\boldsymbol{x}_0^{m}\in\mathbb{R}^{n_x}} \mathrm{E}\left[\|\boldsymbol{x}_t^{m}\|^p\right] < \infty \tag{3.44}$$

holds for all $p \in \mathbb{N}$.

Remark 3.10. *Note the swap of $\boldsymbol{x}_t^{m'}$ and $\boldsymbol{x}_t^m$ in* (3.44) *in contrast to* (3.40).

Proof. In the following, we split the proof in two parts depending on time step t.
For $t \leq \overline{m}$, the memories Ξ_t^m and $\Xi_t^{m'}$ of both GP-ASSMs are identical and, thus, the expected value is bounded by

$$\sup_{t \in \mathbb{N}, t \leq \overline{m}} \mathrm{E}[(x_t^{m'})^p] = \sup_{t \in \mathbb{N}, t \leq m} \mathrm{E}[(x_t^m)^p] < \infty. \tag{3.45}$$

For $t > \overline{m}$, we use the last point in memory $\boldsymbol{x}^{m'}_{\max(0, t-m'-1)}$ as initial point for $\boldsymbol{x}^m_{t+1}$. Thus, we can follow the above argumentation again, which leads to

$$\sup_{t \in \mathbb{N}, t > \overline{m}} \mathrm{E}[(x_t^m)^p] < \infty, \tag{3.46}$$

such that the boundedness is preserved. □

3.2.2 GP Nonlinear Output Error Models

In this section, we transfer our results about boundedness of GP-ASSMs to GP-ANOE models. In GP-ANOE models, the feedback loop is closed by the output $\boldsymbol{y}_t$ instead of the state $\boldsymbol{x}_t$ as in GP-ASSMs. Therefore, we present the following results without further explanation and refer here to Section 3.2.1.

Proposition 3.5. *Consider a GP-ANOE* (3.23) *with maximum memory length* $\overline{m}$ *and bounded mean function and kernel, i.e.,* $\|\boldsymbol{m}(\boldsymbol{\zeta})\| < \infty, k\boldsymbol{\zeta}, \boldsymbol{\zeta}') < \infty$*, respectively, for all* $\boldsymbol{\zeta}, \boldsymbol{\zeta}' \in \mathbb{R}^{n_\zeta}$*. Then, the expected value of the sequence* $\{\boldsymbol{y}_t^m\}, t \in \mathbb{N}$ *given by* (3.23) *is ultimately* p-bounded *by*

$$\sup_{t \in \mathbb{N}} \mathrm{E}\left[\|\boldsymbol{y}_t^m\|^p\right] < \infty$$

for all $\boldsymbol{\zeta}_0 \in \mathbb{R}^{n_\zeta}, \forall p \in \mathbb{N}$*.*

Proof. Analogously to the proof of Theorem 3.3 with the GP-ANOE model defined by (3.23). □

Proposition 3.6. *Consider two GP-ANOEs with outputs* $\boldsymbol{y}_t^m$ *and* $\boldsymbol{y}_t^{m'}$*, respectively, such that*

$$\boldsymbol{y}_{t+1}^m \sim \mathcal{N}\Big(\boldsymbol{h}_t(\boldsymbol{\zeta}_t^m, \Lambda_t), H_t(\boldsymbol{\zeta}_t^m, \Lambda_t)\Big)$$
$$\boldsymbol{y}_{t+1}^{m'} \sim \mathcal{N}\Big(\boldsymbol{h}_t(\boldsymbol{\zeta}_t^{m'}, \Lambda_t'), H_t(\boldsymbol{\zeta}_t^{m'}, \Lambda_t')\Big)$$

where $\overline{m}$ *and* $\overline{m}'$ *are the maximum length of memory, respectively. Then, for* $\overline{m} < \overline{m}'$*,*

$$\sup_{t \in \mathbb{N}} \mathrm{E}\left[\|\boldsymbol{y}_{t+1}^m\|^p\right] < \infty \nRightarrow \sup_{t \in \mathbb{N}} \mathrm{E}\left[\|\boldsymbol{y}_{t+1}^{m'}\|^p\right] < \infty$$

holds for $p \in \mathbb{N}$ *and* $\boldsymbol{\zeta}_0^m = \boldsymbol{\zeta}_0^{m'} \in \mathbb{R}^{n_\zeta}$*.*

Proof. Analogously to the proof of Theorem 3.4 with the GP-ANOE model defined by (3.23). □

Proposition 3.7. *Consider two GP-ANOE models with outputs $\boldsymbol{y}_t^m$ and $\boldsymbol{y}_t^{m'}$, respectively, such that*

$$\begin{aligned}\boldsymbol{y}_{t+1}^m &\sim \mathcal{N}\Big(\boldsymbol{h}_t(\boldsymbol{\zeta}_t^m, \Lambda_t), H_t(\boldsymbol{\zeta}_t^m, \Lambda_t)\Big)\\ \boldsymbol{y}_{t+1}^{m'} &\sim \mathcal{N}\Big(\boldsymbol{h}_t(\boldsymbol{\zeta}_t^{m'}, \Lambda_t'), H_t(\boldsymbol{\zeta}_t^{m'}, \Lambda_t')\Big)\end{aligned}$$

where $\overline{m}$ and $\overline{m}'$ are the maximum length of memory, respectively. Then, if $\overline{m} < \overline{m}'$,

$$\sup_{t\in\mathbb{N},\boldsymbol{\zeta}_0^{m'}\in\mathbb{R}^{n_\zeta}} \mathrm{E}\left\|\boldsymbol{y}_t^{m'}\right\|^p < \infty \Rightarrow \sup_{t\in\mathbb{N},\boldsymbol{\zeta}_0^{m}\in\mathbb{R}^{n_\zeta}} \mathrm{E}\left\|\boldsymbol{y}_t^{m}\right\|^p < \infty$$

holds for all $p \in \mathbb{N}$.

Proof. Analogously to the proof of Theorem 3.5 with the GP-ANOE model defined by (3.23). □

3.2.3 Case Study

In this case study, we demonstrate the modeling of a dynamical system with a GP-SSM and GP-ASSMs with different maximum lengths on memory. As dynamical system to be modeled, we consider the non-autonomous discrete-time predator–prey system introduced in [Liu10]. It is given by

$$\begin{aligned}x_{t+1,1}^{\mathrm{p}} &= x_{t,1}^{\mathrm{p}} \exp\left(1 - 0.4x_{t,1}^{\mathrm{p}} - \frac{(2+1.2u_{t,1})x_{t,2}^{\mathrm{p}}}{1+(x_{t,1}^{\mathrm{p}})^2}\right)\\ x_{t+1,2}^{\mathrm{p}} &= x_{t,2}^{\mathrm{p}} \exp\left(1 + 0.5u_{t,1} - \frac{(1.5-u_{t,2})x_{t,2}^{\mathrm{p}}}{x_{t,1}^{\mathrm{p}}}\right)\end{aligned} \tag{3.47}$$

$$\boldsymbol{y}_t^{\mathrm{p}} = \boldsymbol{x}_t^{\mathrm{p}} + \boldsymbol{\nu}, \quad \boldsymbol{u}_t = \begin{bmatrix}\cos(0.02\pi t)\\ \sin(0.02\pi t)\end{bmatrix}, \tag{3.48}$$

with two-dimensional state $\boldsymbol{x}_t^{\mathrm{p}} \in \mathbb{R}^2$, output $\boldsymbol{y}_t^{\mathrm{p}} \in \mathbb{R}^2$, input $\boldsymbol{u}_t \in \mathbb{R}^2$, and Gaussian distributed noise $\boldsymbol{\nu} \in \mathbb{R}^2, \boldsymbol{\nu} \sim \mathcal{N}(\boldsymbol{0}, 0.05^2 I_2)$. The states $x_{t,1}^{\mathrm{p}}$ and $x_{t,2}^{\mathrm{p}}$ represent the population size of prays and predators, respectively, but are taken to be continuous. The system dynamics (3.47) are assumed to be unknown whereas the input and output, given by (3.48), are assumed to be known. For the modeling with a GP-SSM, 33 training points of a trajectory from the predator–prey system with initial state $\boldsymbol{x}_0^{\mathrm{p}} = [0.3; 0.8]$ are collected. More detailed, every third state $\boldsymbol{x}_t^{\mathrm{p}}$, input $\boldsymbol{u}_t$ and output $\boldsymbol{y}_t^{\mathrm{p}}$ between $t = 1, \dots, 100$ is recorded. Thus, the training set $\mathcal{D} = \{X, Y\}$ consists of

$$\begin{aligned}X &= [\boldsymbol{\xi}_1, \boldsymbol{\xi}_4, \dots, \boldsymbol{\xi}_{97}] \text{ with } \boldsymbol{\xi}_t = [\boldsymbol{x}_t^{\mathrm{p}}; \boldsymbol{u}_t]\\ Y &= [\boldsymbol{y}_1, \boldsymbol{y}_4, \dots, \boldsymbol{y}_{97}]^\top.\end{aligned} \tag{3.49}$$

Following the structure of GP-SSMs in (2.55), two GPs are employed to model each element of the state $\boldsymbol{x}_t^{\mathrm{p}}$ separately. The GPs are based on a squared exponential kernel with automatic relevance detection given by

$$k(\boldsymbol{\xi}_t, \boldsymbol{\xi}_t') = \varphi_1^2 \exp\left(-(\boldsymbol{\xi}_t - \boldsymbol{\xi}_t')^\top P^{-1}(\boldsymbol{\xi}_t - \boldsymbol{\xi}_t')\right)$$

with matrix $P = \operatorname{diag}(\varphi_2^2, \dots, \varphi_5^2)$. This kernel is bounded with respect to $\boldsymbol{\xi}_t, \boldsymbol{\xi}'_t \in \mathbb{R}^4$. The hyperparameters $\varphi_1, \dots, \varphi_5$ of each GP are optimized by means of the likelihood function, see [Ras06]. In this study, we model the dynamics (3.47) with a GP-SSM, a GP-ASSM with maximum length of memory 10 and a GP-ASSM with maximum length of memory 0. For the testing of these models, we select the initial state $\boldsymbol{x}_0^{\mathrm{p}} = [0.268; 0.400]$. The top plot of Fig. 3.7 visualizes the trajectory of the predator–prey system (3.47), considered as the ground-truth. After a transition phase, the numbers of prays (red dashed) and predators (blue solid) converge to a periodic solution. The second plot shows three samples of the

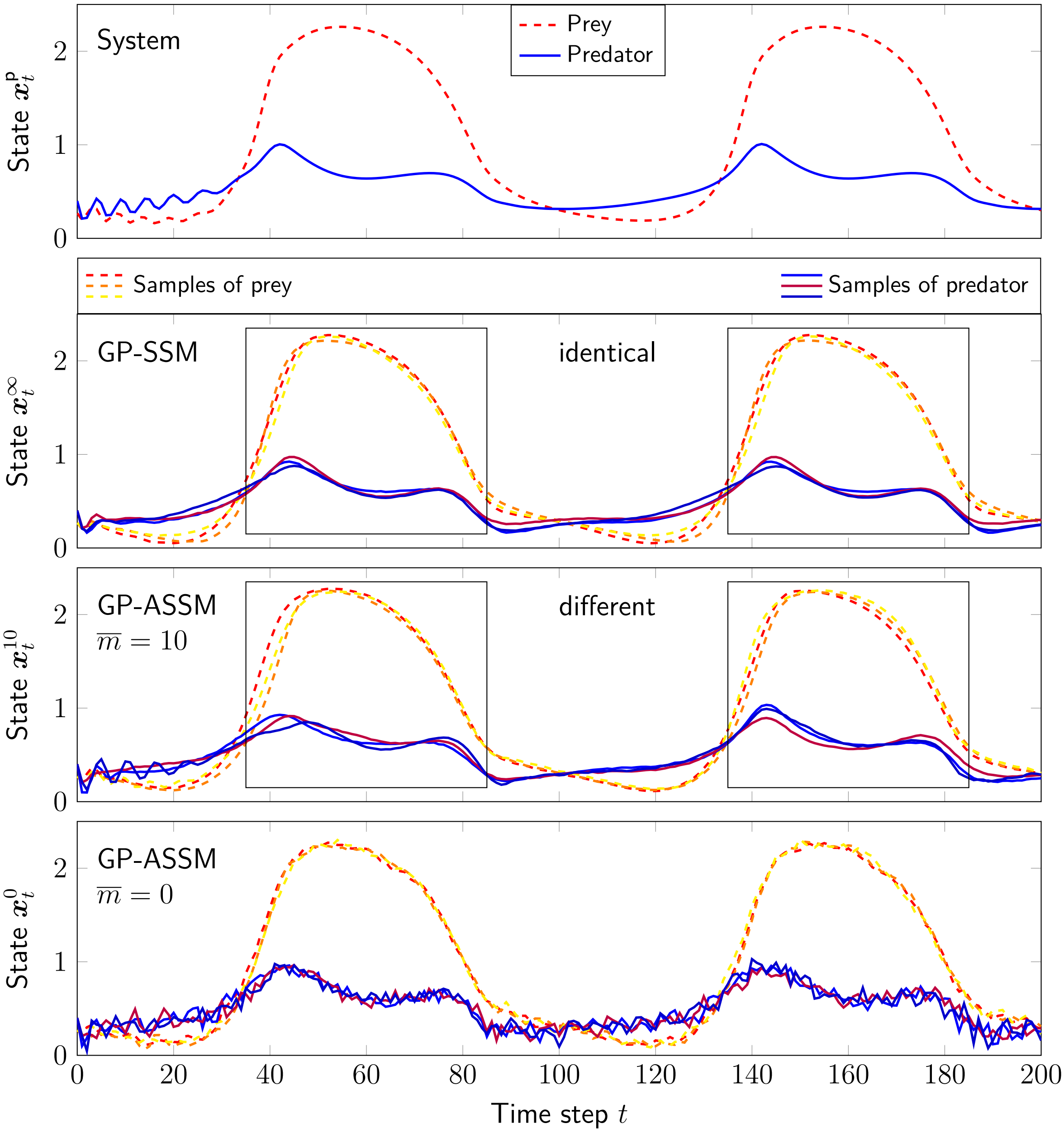


Figure 3.7: From top to bottom: Trajectory of predator–prey system, samples of GP-SSM, samples of GP-ASSM with $\overline{m} = 10$, and samples of GP-SSM with $\overline{m} = 0$. For decreasing maximum length of memory of the approximations, the variance is increasing which leads to rougher trajectories.

GP-SSM drawn by means of Theorem 3.1. Even though the training set consists only of data up to the time step $t = 97$, see (3.49), the GP-SSM precisely predicts the trajectory after the transition phase. As the GP-SSM implies $\overline{m} = \infty$, all past state transitions are added to the memory Ξ_t^∞, defined in (3.15), and used for the next state ahead prediction. Consequently, the shape of each sample is identical in periodic repetitions, as highlighted inside the boxes in the second plot of Fig. 3.7. However, the drawback of this sampling is the increasing size of the memory Ξ_t^∞ and, thus, the increasing size of the Gram matrix $\check{K}_t^\infty$, given by (3.19).

In Fig. 3.8, the size of the square matrix $\check{K}_t^\infty$ is depicted (solid black) with respect to the time step t. For $t = 0$, the matrix solely contains the covariance between each element of the input training data X. The linear slope is problematic for the computation time as the Gram matrix must be inverted for each prediction step, see (3.9).

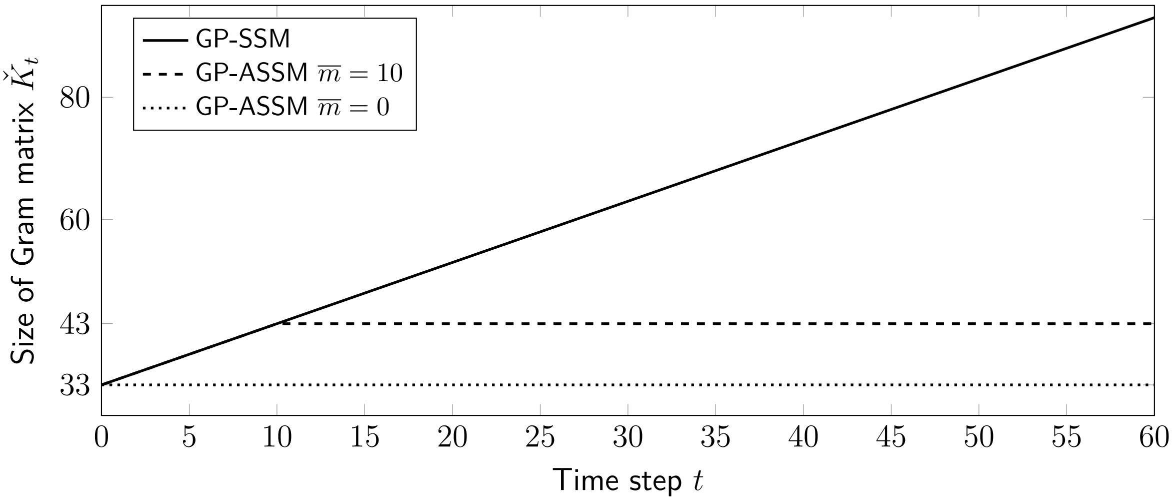


Figure 3.8: The number of rows and columns $n_\mathcal{D}+\underline{m}$ of the Gram matrix for the GP-SSM and two GP-ASSMs. The GP-ASSM allows to bound the size of the Gram matrix, which must be inverted in each time step.

Three samples of the GP-ASSM with maximum length of memory 10, given by the means of (3.16), are visualized in the third plot of Fig. 3.7. The samples are similar to the samples of the GP-SSM, since the memory Ξ_t^{10} consists of sufficiently many past states to generate a similar predictive distribution for next step state. However, the shape of the samples differs between the periodic repetitions, as indicated with the two boxes. This variation is due to the reduced memory, which induces that the evolution of the state inside the left box is not considered for the prediction of the corresponding state in the right box. In contrast to the GP-SSM, the maximum length of memory 10 bounds the size (dashed line) of the Gram matrix $\check{K}_t^{10}$ as shown in Fig. 3.8.

In the bottom plot of Fig. 3.7, three samples of the GP-ASSM with maximum length of memory 0 are drawn. The variance for each prediction step is significantly higher, as described in Proposition 3.4, such that the trajectories are rougher. However, the size of the Gram matrix $\check{K}_t^0$ remains constantly low (dotted line) as presented in Fig. 3.8.

Finally, the GP-SSM and the GP-ASSM with $\overline{m} = 0$ are tested with 50 different initial values, which are drawn from a uniform distribution between $[-5, 5]$ for both states, visualized in Fig. 3.9. All trajectories are bounded, which supports Theorems 3.3 and 3.5.

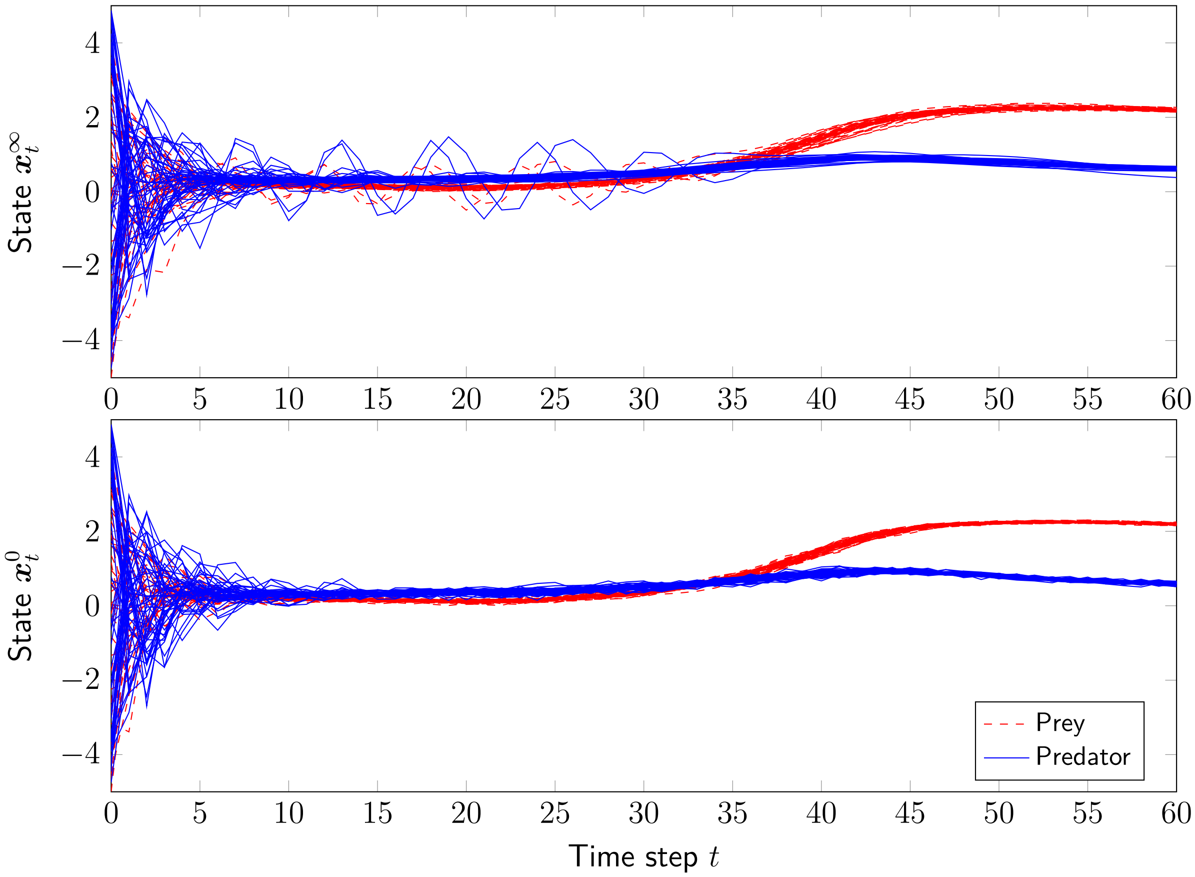


Figure 3.9: Trajectories of 50 samples starting from multiple initial points illustrate the boundedness of the GP-SSM and GP-ASSM.

3.3 Deterministic Markov Models

In many applications, where a GPDM is considered in a control setting, only the mean function of the process is employed, see, e.g., [WHB05] and [Cho+13]. This is mainly because a GPDM is often used for replacing a deterministic model in already well-known model-based control approaches. Hence, the current state is solely exploited for the prediction, such that a well-known Markov structure is achieved. Therefore, we focus in this section on more detailed properties of GP-ASSM with $\overline{m} = 0$ where only the mean prediction is fed back. We call this simplification a *deterministic GP-SSM*. The contribution of this section is the study of equilibria and their stability properties in terms of Lyapunov stability and ultimate boundedness. We analyze deterministic GP-SSMs with a linear, polynomial and squared exponential kernel, see Section 2.2 for details on the various kernels. Afterwards, we determine the number of equilibrium points and present stability conditions for these models. Additionally, the derived results are illustrated in numerical simulations.

The remainder of this section is organized as follows: In Section 3.3.2 the equilibrium points of deterministic GP-SSMs are analyzed. Stability conditions for deterministic GP-SSMs are presented in Section 3.3.3. Finally, Section 3.3.4 presents some illustrations of the derived results.

3.3.1 Deterministic GP-SSM

The GP-ASSM, introduced in Definition 3.1, is taken as basis for the deterministic GP-SSM. For the sake of simplicity, we assume that the prior mean functions $m^1, \dots, m^{n_x}$ are set to zero. However, the results can be easily extended using a definition as in (2.55). We focus on autonomous deterministic GP-SSMs, as we want to analyze its equilibrium points.

Definition 3.3. An autonomous *deterministic GP-SSM* is given by the state space equation

$$\bar{\boldsymbol{x}}_{t+1} = \boldsymbol{\mu}(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \mathcal{D}) = \begin{bmatrix} \boldsymbol{k}(\boldsymbol{x}_t, X)^\top (K + I_{n_\mathcal{D}}\sigma_{n,1}^2)^{-1} Y_{:,1} \\ \vdots \\ \boldsymbol{k}(\boldsymbol{x}_t, X)^\top (K + I_{n_\mathcal{D}}\sigma_{n,n_x}^2)^{-1} Y_{:,n_x} \end{bmatrix} =: \boldsymbol{f}_{\text{det}}(\boldsymbol{x}_t), \tag{3.50}$$

where $\bar{\boldsymbol{x}}_{t+1}$ denotes the next step ahead state and $\boldsymbol{f}_{\text{det}} \in \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ the state mapping.

Hence, an autonomous deterministic GP-SSM is a GP-ASSM with maximum length of memory $\overline{m} = 0$, where the variance and the external input $\boldsymbol{u}_t$ is suppressed. Thus, the predicted next step ahead state $\bar{\boldsymbol{x}}_{t+1}$ is not probabilistic but deterministic.

3.3.2 Equilibrium Points

In this section, we analyze the autonomous deterministic GP-SSM in terms of the existence of equilibrium points. We denote the set of equilibrium points $\mathcal{X}^*$ by

$$\mathcal{X}^* = \{\boldsymbol{x}^* \in \mathcal{X} \mid \boldsymbol{x}^* = \boldsymbol{f}_{\text{det}}(\boldsymbol{x}^*)\}.$$

The cardinality $|\mathcal{X}^*|$ denotes the number of equilibrium points. Each component of the predicted state vector of a deterministic GP-SSM, as defined in (3.50), can be written as weighted sum of kernels. The number of kernels is equal to the number of training points $n_\mathcal{D}$, such that (3.50) is rewritten to

$$\bar{x}_{t+1,i} = \sum_{j=1}^{n_\mathcal{D}} k(\boldsymbol{x}_t, X_{:,j}) \underbrace{[(K + I_{n_\mathcal{D}}\sigma_{n,i}^2)^{-1} Y_{:,i}]_j}_{\alpha_j(i)}. \tag{3.51}$$

The vector of weighting factors $\boldsymbol{\alpha}(i) = [\alpha_1(i), \dots, \alpha_{n_\mathcal{D}}(i)]^\top$ depends on the inverse of the covariance matrix with signal noise $(K(X,X) + I_{n_\mathcal{D}}\sigma_{n,i}^2)^{-1}$, the output training matrix Y and the required element i. In the following, we provide an overview about the behavior of deterministic GP-SSMs with common kernels of the listed kernels in Table 2.1.

Squared exponential kernel

The widely used squared exponential kernel, as defined in (2.47), is very powerful for nonlinear function regression. The following proposition gives a lower bound of the quantity of equilibrium points.

Proposition 3.8. *Consider an autonomous deterministic GP-SSM with squared exponential kernel* (2.47)*, as introduced in* (3.50)*. There exists at least one equilibrium point*

$$\min |\mathcal{X}^*| = 1.$$

Proof. The idea of the proof is that each component-wise equation $x^*_{t,i} = [f_{\text{det}}(\boldsymbol{x}_t)]_i$ has a solution for any fixed component $x_{t,j} \in \mathbb{R}$ with $j \in \{1, \dots, n_x\}$ and $j \neq i$. Therefore, it must exist at least one solution for the overall system of equations.

For the proof of the minimum quantity of equilibrium points, we consider the state prediction of the autonomous deterministic GP-SSM (3.51) and insert the equation of the squared exponential kernel

$$\bar{x}_{t+1,i} = \sum_{j=1}^{n_{\mathcal{D}}} \varphi^2_{1,i} \exp\left(-\frac{\|\boldsymbol{x}_t - X_{:,j}\|^2}{2\varphi^2_{2,i}}\right) \alpha_j(i). \tag{3.52}$$

The parameters $\varphi_{1,i}$ and $\varphi_{2,i}$ are the hyperparameters of the function $[f_{\text{det}}]_i$. As far as we know, it is not possible to find an analytic solution for this kind of multivariate equation system. Therefore, the system functions will be treated component-wise. Even though this view neglects the effects of the multivariate structure, it provides a valid solution for the minimum number of equilibrium points as shown at the end of the proof. An important property of the squared exponential function is the behavior at the limit given by

$$\lim_{\|\boldsymbol{z}\| \to \infty} \varphi^2_1 \exp\left(-\frac{\|\boldsymbol{z} - \boldsymbol{z}'\|^2}{2\varphi^2_2}\right) = 0, \quad \text{with } \boldsymbol{z}, \boldsymbol{z}' \in \mathbb{R}^{n_x}. \tag{3.53}$$

Since the limit of the squared exponential function is zero, the limit of the weighted sum of squared exponential functions must be also zero

$$\lim_{\|\boldsymbol{x}_t\| \to \infty} \sum_{j=1}^{n_{\mathcal{D}}} \varphi^2_{1,i} \exp\left(-\frac{\|\boldsymbol{x}_t - X_{:,j}\|^2}{2\varphi^2_{2,i}}\right) \alpha_j(i) = 0. \tag{3.54}$$

To finish the proof, we require Bolzano's theorem, which is a special case of the intermediate value theorem.

Theorem 3.6 (Bolzano, [LE13])**.** *Suppose $f_{gen}(z) : [c_1, c_2] \to \mathbb{R}$ is continuous on the closed interval $[c_1, c_2]$ and suppose that $f_{gen}(c_1)$ and $f_{gen}(c_2)$ have opposite signs. Then there exists a number c in the interval $[c_1, c_2]$ for which $f_{gen}(c) = 0$.*

Since Bolzano's theorem just holds for scalar functions, the state space equation (3.52) must be rewritten as function of a scalar variable. For this purpose, the components x^*_j with $j \in \{1, \dots, n_x\}, j \neq i$ are fixed and we obtain a function $\boldsymbol{f}^s_{\text{det}}(x_i) \colon \mathbb{R} \to \mathbb{R}^{n_x}$ given by

$$\begin{aligned} [f^s_{\text{det}}(x_{t,i})]_i &:= [f_{\text{det}}([x_{t,1}, \dots, x_{t,i-1}, x_{t,i}, x_{t,i+1}, \dots, x_{t,n_x}])]_i \\ &\quad \text{with fixed } x_{t,1}, \dots, x_{t,i-1}, x_{t,i+1}, \dots, x_{t,n_x} \in \mathbb{R}. \end{aligned} \tag{3.55}$$

Due to the fact that $[f^s_{\text{det}}(\cdot)]_i$ is continuous and its limits

$$\begin{aligned} \lim_{x_{t,i} \to \infty} [f^s_{\text{det}}(x_{t,i})]_i - x_{t,i} &= -\infty \\ \lim_{x_{t,i} \to -\infty} [f^s_{\text{det}}(x_{t,i})]_i - x_{t,i} &= \infty \end{aligned} \tag{3.56}$$

have a different sign, Bolzano's theorem guarantees at least one solution $x^*_i = [f^s_{\text{det}}(x^*_i)]_i$. Since this holds for any $x_{t,1}, \dots, x_{t,i-1}, x_{t,i+1}, \dots, x_{t,n_x} \in \mathbb{R}$, each function $[f^s_{\text{det}}(x^*_i)]_i$ has such a solution. Therefore, there must exist an equilibrium point $\boldsymbol{x}^*$ which fulfills the equilibrium equation $\boldsymbol{x}^* = \boldsymbol{f}_{\text{det}}(\boldsymbol{x}^*)$. □

Example 3.4. Figure 3.10 visualizes the idea of the proof. For an example system with two states, the top row shows color-coded on the left side the difference between $[f_{\text{det}}(\boldsymbol{x}_t)]_1$ and $x_{t,1}$ and on the right side the difference between $[f_{\text{det}}(\boldsymbol{x}_t)]_2$ and $x_{t,2}$. If the distance is zero, which is illustrated by dark color, the corresponding component of the state vector $x_{t,i}$ equals $[f_{\text{det}}(\boldsymbol{x}_t)]_i$. The second row shows the slice plane $[f^s_{\text{det}}(x_{t,i})]_i - x_{t,i}$, which should be zero for an equilibrium. On the left side, $x_{2,k}$ is fixed by three example values -5 (red), 5 (blue) and 0.93 (green). On the right side, $x_{t,1}$ is fixed by three example values -5 (red), 5 (blue) and -1.88 (green). As Bolzano's theorem predicts, each function has at least one zero crossing. Therefore, it is possible to find two values x^*_1 and x^*_2 which fulfill $[f^s_{\text{det}}(x^*_i)]_i - x^*_i = 0$ for each $i \in \{1, 2\}$. For this example system, a numerical solver determines one equilibrium point at $\boldsymbol{x}^* = [-1.88, 0.93]^\top$. The green function illustrates this value. On the left side, the function crosses zero at 0.93 and on the right side zero is crossed at -1.88.

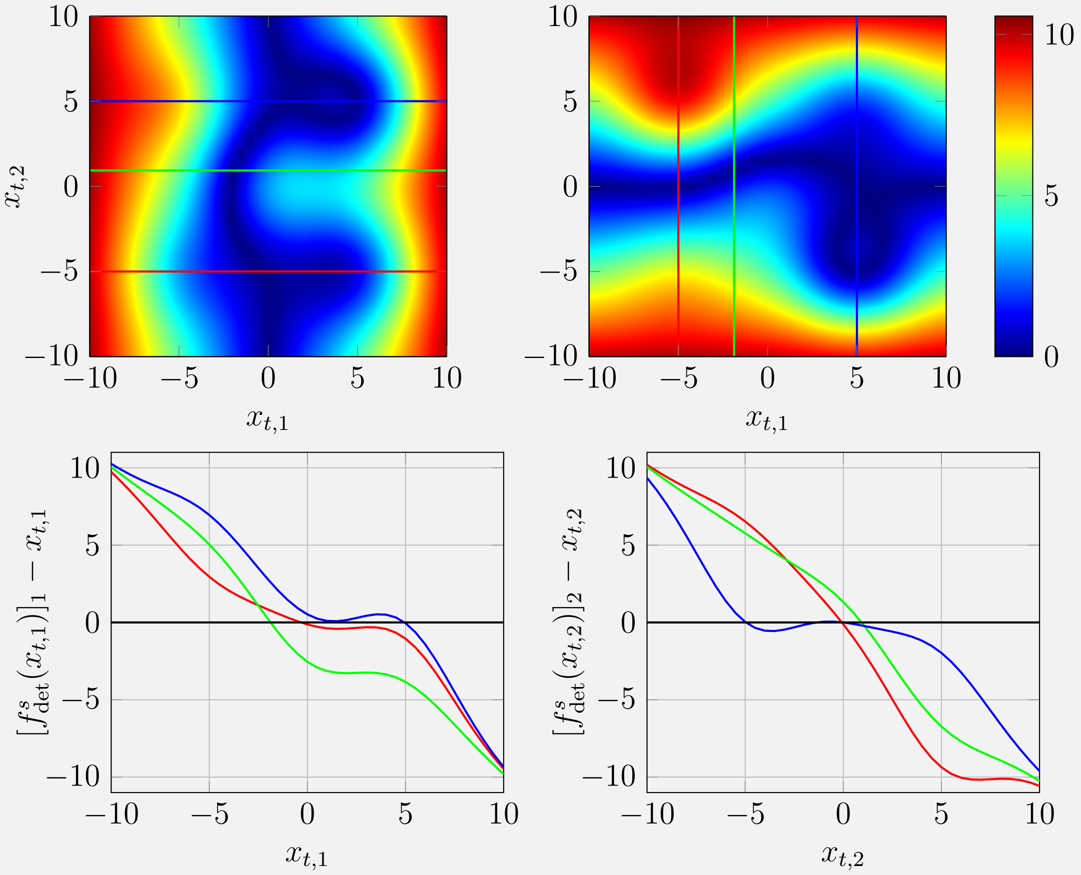


Figure 3.10: Top: Shows color-coded $|[f_{\text{det}}(\boldsymbol{x}_t)]_1 - x_{t,1}|$ on the left and $|[f_{\text{det}}(\boldsymbol{x}_t)]_2 - x_{t,2}|$ on the right against $x_{t,1}$ and $x_{t,2}$. Dark blue marks the area with possible equilibrium points. Bottom: On the left side, $x_{t,2}$ is fixed by three example values -5 (red), 5 (blue) and 0.93 (green). On the right side, $x_{t,1}$ is fixed by three example values -5 (red), 5 (blue) and -1.88 (green). As predicited by Bolzano's theorem, each function has at least one zero crossing.

Linear kernel

In the following, we investigate the equilibrium points of the linear kernel, as defined in (2.44). The following proposition classifies the number of equilibrium points into three outcomes.

Proposition 3.9. *Consider an autonomous deterministic GP-SSM with linear kernel* (2.44)*, as introduced in* (3.50)*. It has the following properties:*

$$|\mathcal{X}^*| = 0 \;\vee\; |\mathcal{X}^*| = 1 \;\vee\; |\mathcal{X}^*| = \infty$$

Proof. We start with the prediction for the next state ahead, given by (3.51), and insert the equation of the linear kernel. That leads to the next state ahead given by

$$\begin{aligned}\bar{x}_{t+1,i} &= \sum_{j=1}^{n_\mathcal{D}} k_j(\boldsymbol{x}_t, X)\alpha_j(i) = \sum_{j=1}^{m} (\boldsymbol{x}_t^\top X_{:,j} + \varphi_{1,i}^2)\alpha_j(i) \\ &= \sum_{j=1}^{n_\mathcal{D}} \boldsymbol{x}_t^\top X_{:,j}\alpha_j(i) + \varphi_{1,i}^2 \mathbf{1}^\top \boldsymbol{\alpha}(i), \end{aligned} \tag{3.57}$$

where $\mathbf{1}^\top$ denotes the vector $[1, \dots, 1]$ of appropriate dimension and $\boldsymbol{\alpha}(i)$ is given by (3.51). Since the sum of linear functions is also a linear function, the whole one step ahead state vector $\bar{\boldsymbol{x}}_{t+1}$ can be written as

$$\bar{\boldsymbol{x}}_{t+1} = \underbrace{\begin{pmatrix} X_{1,:}\boldsymbol{\alpha}(1), \dots, X_{n_x,:}\boldsymbol{\alpha}(1) \\ X_{1,:}\boldsymbol{\alpha}(2), \dots, X_{n_x,:}\boldsymbol{\alpha}(2) \\ \vdots \\ X_{1,:}\boldsymbol{\alpha}(n_x), \dots, X_{n_x,:}\boldsymbol{\alpha}(n_x) \end{pmatrix}}_{A} \boldsymbol{x}_t + \underbrace{\begin{pmatrix} \varphi_{1,1}^2 \mathbf{1}^\top \boldsymbol{\alpha}(1) \\ \varphi_{1,2}^2 \mathbf{1}^\top \boldsymbol{\alpha}(2) \\ \vdots \\ \varphi_{1,n_x}^2 \mathbf{1}^\top \boldsymbol{\alpha}(n_x) \end{pmatrix}}_{\boldsymbol{b}}, \tag{3.58}$$

and, thus, has a form of a non-homogeneous linear system with state matrix $A \in \mathbb{R}^{n_x \times n_x}$ and offset $\boldsymbol{b} \in \mathbb{R}^{n_x}$. The equilibrium points are calculated by solving the equation $\boldsymbol{x}^* = A\boldsymbol{x}^* + \boldsymbol{b}$. Therefore, we use in the following the Moore-Penrose pseudoinverse matrix $(A^I)^+$ with $A^I = I_{n_x} - A$. The system (3.58) behaves in any one of three possible ways:

(i) The system has a single unique solution
if $\operatorname{rank}(A^I) = \operatorname{rank}(A^I|\boldsymbol{b}) = n_x \Rightarrow \mathcal{X}^* = \{(A^I)^{-1}\boldsymbol{b}\}$

(ii) The system has infinitely many solutions
if $\operatorname{rank}(A^I) = \operatorname{rank}(A^I|\boldsymbol{b}) < n_x \Rightarrow \mathcal{X}^* = \{(A^I)^{+}\boldsymbol{b} + \ker(A)\}$

(iii) The system has no solution
if $\operatorname{rank}(A^I) \neq \operatorname{rank}(A^I|\boldsymbol{b}) \Rightarrow \mathcal{X}^* = \{\emptyset\}$

Thus, the number of equilibrium points is given by one of these solutions which concludes the proof. □

Remark 3.11. *Due to the fact that a system with infinitely many solutions or no solution is in practice almost impossible, the presented conditions (ii) and (iii) are only pathological cases. For example, if we assume a one dimensional system, A must be exactly* 1 *to obtain infinitely many solutions (noise free data where the input data equals the output data) or no solution (noise free data where the input data plus a constant equals the output data).*

Polynomial kernel

Next, we study the polynomial kernel, as defined in (2.45), which is more flexible than the linear kernel as it allows nonlinear function estimation. Its degree p is important for the quantity of equilibrium points as the next theorem shows.

Proposition 3.10. *Consider an autonomous deterministic GP-SSM with polynomial kernel* (2.45)*, as introduced in* (3.50)*. Its set of equilibrium points of has the property*

$$|\mathcal{X}^*| \leq \prod_{i=1}^{n_x} p_i,$$

where p_i is the degree of the polynomial kernel corresponding to the i-th component of $\bar{\boldsymbol{x}}_{t+1}$.

Proof. Once again, we employ the next state ahead prediction (3.51) and insert the polynomial kernel. Thus, we obtain

$$\bar{x}_{t+1,i} = \sum_{j=1}^{n_{\mathcal{D}}} k(\boldsymbol{x}_t, X_{:,j})\alpha_j(i) = \sum_{j=1}^{n_{\mathcal{D}}} (\boldsymbol{x}_t^\top X_{:,j} + \varphi_{1,i}^2)^{p_i}\alpha_j(i), \tag{3.59}$$

where the vector of exponents $\boldsymbol{p} = [p_1, \ldots, p_{n_x}]^\top \in \mathbb{N}^{n_x}$ contains the degree of each polynomial kernel. Employing the multinomial theorem [GZ16] and the condition for equilibrium points $\boldsymbol{x}^* = \boldsymbol{f}_{\text{det}}(\boldsymbol{x}^*)$, equation (3.59) can be rewritten as

$$x_i^* = \sum_{l_1+\ldots+l_{n_x+1}=p_i} c_{i,l_1,\ldots,l_{n_x+1}} {x_1^*}^{l_1} {x_2^*}^{l_2} \cdots {x_{n_x}^*}^{l_{n_x}} \varphi_{1,i}^{2l_{n_x+1}} \tag{3.60}$$

with $0 \leq l_1, \ldots, l_{n_x+1} \leq n_x$ and $c_{l_1,\ldots,l_{n_x+1}} \in \mathbb{R}$. The term at the left-hand side can be integrated into the right-hand side by adapting the coefficients $c_{l_1,\ldots,l_{n_x+1}}$ to $c'_{i,l_1,\ldots,l_{n_x+1}} \in \mathbb{R}$ and we obtain

$$0 = \sum_{l_1+\ldots+l_{n_x+1}=p_i} c'_{i,l_1,\ldots,l_{n_x+1}} {x_1^*}^{l_1} {x_2^*}^{l_2} \cdots {x_{n_x}^*}^{l_{n_x}} \varphi_{1,i}^{2l_{n_x+1}}. \tag{3.61}$$

The theorem of Bézout gives an upper bound for the number of roots for this polynomial system.

Theorem 3.7 (Bézout, [Stu98])**.** *Consider a square polynomial system denoted by $\boldsymbol{f}_{gen}(\boldsymbol{z})$ with degree c_i of each polynomial function $f_{gen,i}(\boldsymbol{z})$*

$$\begin{aligned} f_{gen,1}(z_1, z_2, \ldots, z_n) &= 0 \\ &\vdots \\ f_{gen,n}(z_1, z_2, \ldots, z_n) &= 0. \end{aligned}$$

Unless it has an infinite number of zeros, the number of its isolated zeros in $\mathcal{C}^n$, counting multiplicities, does not exceed the number $c = c_1 c_2 \cdots c_n$.

Due to the fact that the real numbers are a subset of the complex numbers, the number of zeros of (3.61) in $\mathbb{R}^{n_x}$ is less than or equal to the number of zeros given by Bézout's theorem. □

Given that the polynomial (3.61) is incomplete, Bernstein's theorem allows to calculate a tighter bound for the number of zeros. However, the generated polynomial functions by (3.61) are complete and, thus, Bernstein's theorem does not provide a closer boundary.

3.3.3 Stability

In this section, we analyze the stability of autonomous deterministic GP-SSMs as stability is a crucial property for the application to control scenarios. For each of the previously presented kernels the related stability condition is provided in the following theorems. We start with a theorem for the squared exponential kernel.

Theorem 3.8. *An autonomous deterministic GP-SSM* (3.50) *with squared exponential kernel, as defined in* (2.47)*, and $n_{\mathcal{D}}$ training points has the following properties:*

(i) There exists an invariant set

$$\mathcal{I} = \left\{ \boldsymbol{z} \in \mathbb{R}^{n_x} \mid |z_i| \leq \varphi_{1,i}^2 \sqrt{n_{\mathcal{D}}} \|\boldsymbol{\alpha}(i)\|, \forall i = 1, \ldots, n_x \right\},$$

with $\boldsymbol{\alpha}(i)$ defined in (3.51)*, which is globally attractive.*

(ii) The solution of (3.50) *is globally uniformly ultimately bounded by*

$$\sup_{t \in \mathbb{N}_{>0}} \|\boldsymbol{x}_t\| \leq b_{uub} = \sqrt{n_{\mathcal{D}}} \left\| \left[\varphi_{1,1}^2 \|\boldsymbol{\alpha}(1)\|, \ldots, \varphi_{1,n_x}^2 \|\boldsymbol{\alpha}(n_x)\| \right] \right\|.$$

Proof. We first prove required properties of the smooth squared exponential kernel $k(\boldsymbol{z}, \boldsymbol{z}')$: For all $\varphi_1 \in \mathbb{R}_{>0}$ and $\varphi_2 \in \mathbb{R}_{>0}$ the kernel is bounded with

$$\sup_{\boldsymbol{z}, \boldsymbol{z}' \in \mathbb{R}^{n_x}} k(\boldsymbol{z}, \boldsymbol{z}') = \varphi_1^2 \exp\left(-\frac{\|\boldsymbol{z} - \boldsymbol{z}'\|^2}{2\varphi_2^2} \right)\Bigg|_{\boldsymbol{z}=\boldsymbol{z}'} = \varphi_{1,i}^2 \tag{3.62}$$

$$\inf_{\boldsymbol{z}, \boldsymbol{z}' \in \mathbb{R}^{n_x}} k(\boldsymbol{z}, \boldsymbol{z}') = \lim_{\|\boldsymbol{z} - \boldsymbol{z}'\| \to \infty} \varphi_1^2 \exp\left(-\frac{\|\boldsymbol{z} - \boldsymbol{z}'\|^2}{2\varphi_2^2} \right) = 0. \tag{3.63}$$

According to the Cauchy-Schwarz inequality and the results above, the next step ahead can be bounded by

$$|\bar{x}_{t+1,i}| = |\boldsymbol{k}(\boldsymbol{x}_t, X)^\top \boldsymbol{\alpha}(i)| \leq \varphi_{1,i}^2 \sqrt{n_{\mathcal{D}}} \|\boldsymbol{\alpha}(i)\|. \tag{3.64}$$

As (3.64) holds for any time step $t \in \mathbb{N}$, the invariant set $\mathcal{I}$ is determined by

$$\mathcal{I} = \left\{ \boldsymbol{z} \in \mathbb{R}^{n_x} \mid |z_i| \leq \varphi_{1,i}^2 \sqrt{n_{\mathcal{D}}} \|\boldsymbol{\alpha}(i)\|, \forall i = 1, \ldots, n_x \right\}. \tag{3.65}$$

Next, we show that the set $\mathcal{I}$ is attractive, see (i) in Theorem 3.8. Since the bound in (3.64) shows that for any $\boldsymbol{x}_0 \in \mathcal{X}$ the absolute value of the next step state vector $\boldsymbol{x}_1$ is equal to or less than $\varphi_{1,i}^2\sqrt{n_\mathcal{D}}\|\boldsymbol{\alpha}(i)\|$, the state $\boldsymbol{x}_t$ approaches $\mathcal{I}$ for $t \geq 1$. This guarantees globally uniformly ultimately boundedness, see (ii) in Theorem 3.8, with ultimate bound

$$b_{\text{uub}} = \sqrt{n_\mathcal{D}}\left\|\left[\varphi_{1,1}^2\|\boldsymbol{\alpha}(1)\|, \dots, \varphi_{1,n_x}^2\|\boldsymbol{\alpha}(n_x)\|\right]\right\|. \tag{3.66}$$

Thus, we proof the existence of an invariant set $\mathcal{I}$, that is globally attractive, and we determine the ultimate bound b_{uub}. □

An important consequence of Theorem 3.8 is that it is not possible to learn unbounded system trajectories with an autonomous deterministic GP-SSM, which is based on the squared exponential kernel. We continue with the stability of a GP-SSM with linear kernel.

Theorem 3.9. *An autonomous deterministic GP-SSM* (3.50) *with linear kernel* (2.44)*, as defined in* (2.44)*, is stable if and only if the spectrum of the matrix*

$$A = \begin{pmatrix} X_{1,:}\boldsymbol{\alpha}(1), \dots, X_{n,:}\boldsymbol{\alpha}(1) \\ X_{1,:}\boldsymbol{\alpha}(2), \dots, X_{n,:}\boldsymbol{\alpha}(2) \\ \vdots \\ X_{1,:}\boldsymbol{\alpha}(n_x), \dots, X_{n,:}\boldsymbol{\alpha}(n_x) \end{pmatrix}$$

is equal to or less than one. If the absolute value of the spectrum is strictly less than one, i.e. $\sigma(A) < 1$, then the equilibrium point is asymptotically stable. Otherwise, the system is unstable.

Proof. Since the state mapping of a GP-SSM with linear kernel is a linear function, see (3.58), the well-known conditions for linear stability can be directly applied. □

In the following, the stability for a GP-SSM with polynomial kernel is analyzed.

Theorem 3.10. *An autonomous deterministic GP-SSM* (3.50) *with polynomial kernel, as defined in* (2.45)*, is locally stable in $\boldsymbol{x}^*$ if and only if the spectrum of the matrix*

$$A_{poly} = \frac{\partial}{\partial \boldsymbol{x}_t} \sum_{l_1+\ldots+l_{n_x+1}=p_i} c_{i,l_1,\ldots,l_{n_x+1}} x_{t,1}^{l_1} x_{t,1}^{l_2} \cdots x_{t,n_x}^{l_n} \varphi_{1,i}^{2l_{n_x+1}} \Bigg|_{\boldsymbol{x}_t=\boldsymbol{x}^*}$$

is equal to or less than one. If $\sigma(A_{poly}) < 1$ the equilibrium point is locally asymptotically stable.

Proof. This theorem is a direct application of Lyapunov's direct method. The derivative exists, since the polynomial kernel is a smooth function. □

3.3.4 Numerical Examples

Equilibrium points

After the formal analysis of deterministic GP-SSMs, we present illustrations for an intuitive understanding of the equilibrium sets for different kernels. For this purpose, 100 randomly linear systems are generated with

$$\boldsymbol{x}_{t+1} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \boldsymbol{x}_t + \boldsymbol{\nu}, \tag{3.67}$$

where $a_{11}, a_{12}, a_{21}, a_{22} \in (0, 1)$ are random numbers drawn from the uniform distribution on the open interval and $\boldsymbol{\nu}$ is Gaussian distributed noise $\mathcal{N}(0, 0.05^2 I_2)$. Each system is learned by a autonomous deterministic GP-SSM with 100 homogeneously distributed training points on $[-1, 1] \times [-1, 1]$ and 5 different kernels (linear, polynomial with $p = 2, 3, 5$, squared exponential, see Table 2.1).

The hyperparameters are optimized according to the likelihood function (2.51) with a conjugate gradient method. The equilibrium points are numerically estimated by local solvers starting from multiple points in $[-20, 20] \times [-20, 20]$. For comparison, the same procedure is applied to randomly generated nonlinear systems, which have a multitude of equilibrium points

$$\boldsymbol{x}_{t+1} = \begin{bmatrix} \sin(c_1 x_{2,t}) + x_{1,t} \\ \sin(c_2 x_{1,t}) + x_{2,t} \end{bmatrix} + \nu. \tag{3.68}$$

The values $c_1, c_2 \in (0, \frac{3}{2}\pi)$ are random numbers drawn from the uniform distribution on the open interval and $\boldsymbol{\nu}$ represents Gaussian distributed noise $\mathcal{N}(0, 0.05^2 I_2)$. Now, the area of initial points for each of the multiple local solvers is set to $[-5, 5] \times [-5, 5]$. Table 3.2 and Table 3.3 show the number of detected equilibrium points of the deterministic GP-SSMs, which are trained by the linear and the nonlinear systems. Since it is extremely unlikely that a deterministic GP-SSM with linear kernel has zero or infinitely many equilibrium

Kernel / Number of equilibria	0	1	2	3	4
Linear	0	100	0	0	0
Polynomial $p = 2$	0	53	44	3	0
Polynomial $p = 3$	0	54	42	3	1
Polynomial $p = 5$	0	53	42	4	1
Squared exponential	0	50	1	49	0

Table 3.2: Number of equilibrium points of 100 deterministic GP-SSMs, each trained by a randomly generated 2-dimensional linear systems.

Kernel / Number of equilibria	0	1	2	$[3, 4]$	$[5, 9]$	$[10, 19]$
Linear	0	100	0	0	0	0
Polynomial $p = 2$	0	97	3	0	0	0
Polynomial $p = 3$	0	70	0	5	25	0
Polynomial $p = 5$	0	27	2	10	27	34
Squared exponential	0	3	2	30	32	33

Table 3.3: Number of equilibrium points of 100 deterministic GP-SSMs trained by randomly generated 2-dimensional, sinusoidal systems.

points, as mentioned in Remark 3.11, only one equilibrium point for each of the 100 test systems has been found. The system with polynomial kernel has always a maximum of p^2 equilibrium points and the deterministic GP-SSMs with squared exponential kernel more than zero equilibrium points.

Stability

Due to the wide spread of the squared exponential kernel, we present here an example for the boundedness of deterministic GP-SSMs with squared exponential kernel. Our example employs with the well-known, nonlinear Van der Pol oscillator. The discretization of the oscillator is described in [VH10] with

$$\boldsymbol{x}_{t+1} = \begin{bmatrix} f_{\mathrm{VDP1}}(t_s, x_{t,1}, x_{t,2}, c_3) + (f_{\mathrm{VDP2}}(t_s, x_{t,1}, x_{t,2}, c_3) + 1)x_{t,1} + \nu_1 \\ f_{\mathrm{VDP1}}(t_s, x_{t,1}, x_{t,2}, c_3) + (f_{\mathrm{VDP2}}(t_s, x_{t,1}, x_{t,2}, c_3) + 1)x_{t,2} + \nu_2 \end{bmatrix}, \tag{3.69}$$

where $t_s \in \mathbb{R}_{>0}$ is the sample time and the parameter $c_3 \in \mathbb{R}$ is a scalar, which influences the nonlinearity of the system. Gaussian distributed noise is added by $\nu_1, \nu_2 \sim \mathcal{N}(0, 0.01^2)$. For a detailed expanation of the functions $f_{\mathrm{VDP1}}, f_{\mathrm{VDP2}}$, we refer to [VH10]. For this example c_3 is set to -0.8 and the sample time $t_s = 0.1$.

A deterministic GP-SSM with squared exponential kernel is trained over 441 homogeneously distributed points in $[-4, 4] \times [-4, 4]$. The hyperparameters are optimized by the minimization of the likelihood function (2.51) with a conjugate gradient method. We now illustrate the behavior of the trajectory generated by the deterministic GP-SSM. Figure 3.11 shows the trajectory $\boldsymbol{x}_t$ (blue and red) of the system (3.69) and the prediction of the trained deterministic GP-SSM $\bar{\boldsymbol{x}}_t$ (green and black) for $\boldsymbol{x}_0 = [-1.8; 0]$. Since the trajectory stays inside the training area, the predicted trajectory is similar to the real trajectory $\boldsymbol{x}_t$. Furthermore, the boundedness of the trained deterministic GP-SSM is demonstrated.

In contrast, we select a different initial value and present the result in Fig. 3.12. The graph shows the trajectory (blue and red) of the system (3.69) and the prediction of the trained deterministic GP-SSM $\bar{\boldsymbol{x}}_t$ (green and black) for the initial value $\boldsymbol{x}_0 = [2.2; 0]$. This initial

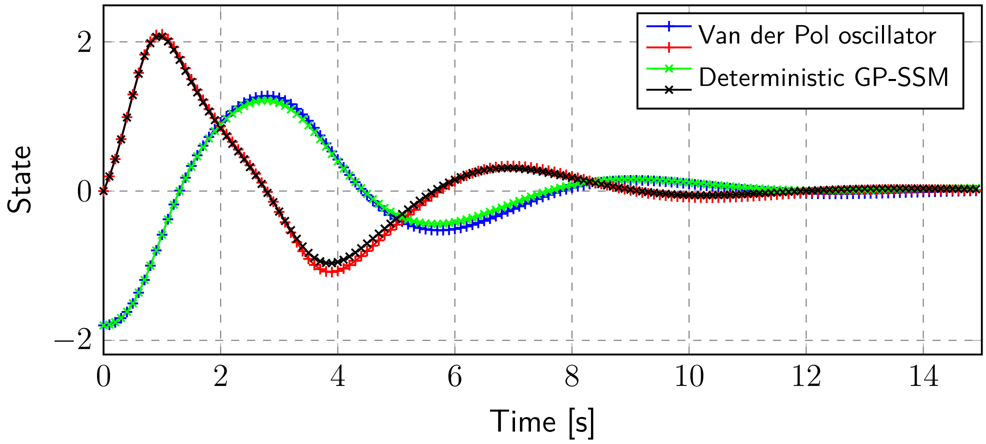


Figure 3.11: For $\boldsymbol{x}_0 = [-1.8, 0]^\top$, the prediction (green and black) of a deterministic GP-SSM with squared exponential kernel and the trajectory (blue and red) of the actual system (3.69) are close to each other.

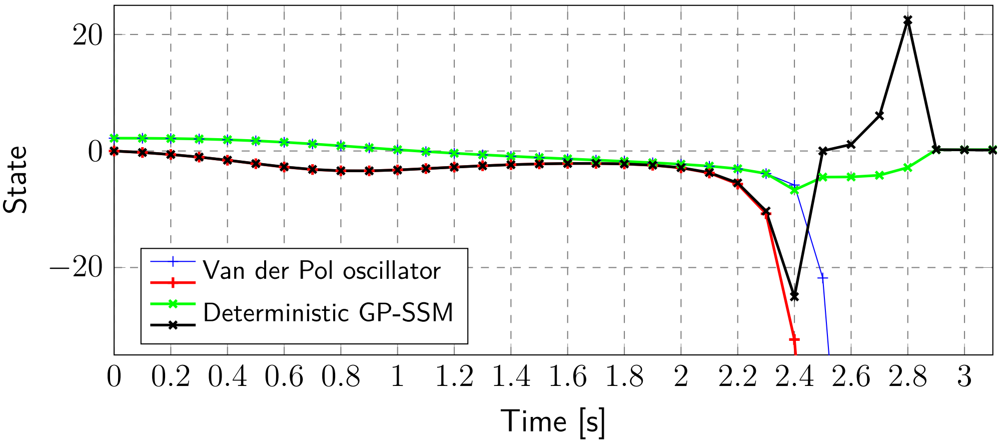


Figure 3.12: The prediction (green and black) of a deterministic GP-SSM with squared exponential kernel is bounded even if the trajectory (blue and red) of the actual system is unbounded.

point is not inside the region of attraction of the oscillator, such that the trajectory $\boldsymbol{x}_t$ of the system is not bounded. Although the original trajectory is unbounded, the prediction of the deterministic GP-SSM is bounded. Thus, only bounded trajectories can sufficiently predicted by a deterministic GP-SSM with squared exponential kernel. However, this property can also be used as prior knowledge: If we know that all trajectories of a real system are bounded, a deterministic GP-SSM with squared exponential kernel can be used for modeling as this GP-SSM is always bounded. For the modeling of unbounded system, a combination of the squared exponential kernel with a linear kernel might be a reasonable selection, see Section 2.2 and Theorem 3.9.

In this section, we have analyzed the number of equilibrium points and the stability of deterministic GP-SSMs. The results present an important limitation for the modeling with deterministic GP-SSMs but also allow to use these properties as prior knowledge for the modeling.

3.4 Stochastic Markov Models

So far, we focused on theoretical properties for deterministic GP-SSMs, which are a special class of GP-ASSMs. However, a GP-ASSM contains a much richer description of the underlying dynamics due to its probabilistic nature. The predicted uncertainty can be used, for instance, to determine a suitable control law. In [MLH15] control laws are derived, which explicitly take the uncertainty into account. In order to ensure the applicability of these control settings, classical control theoretical properties are required, see, e.g., [Koc16] and [AK08]. Such basic properties of a dynamical system are among others the existence of equilibria and stability conditions. Hence, in this section, we analyze a special class of GP-ASSMs, where the probabilistic nature is considered but the dynamics is still Markovian. We call these models *stochastic GP-SSMs*. In stochastic systems, the concept of equilibrium distribution, often also called stationary distribution, is used. Namely, the output distribution of a stochastic GP-SSM for a given input distribution must be calculated. If there

exists an input distribution which is equal to the related output distribution, it is called equilibrium distribution. Different definitions exist for the stability of stochastic systems. The most common ones are stability in probability, almost sure stability and mean square stability. Since mean square stability implies stability in probability (Markov's inequality), it is a frequently used approach. For the formal analysis of control approaches based on GP-SSMs, knowledge about the stability is essential. However, the calculation of equilibrium distributions of stochastic GP-SSMs and the derivation of stability conditions are still open problems.
In this section, we study equilibrium distributions and stability of stochastic GP-SSMs. We introduce a method to compute the equilibrium distribution, based on the solution of a Fredholm integral equation. The method is usable for arbitrary kernels. For the widespread squared exponential kernel, we present an upper bound in mean square sense and a set, which is positive recurrent. We demonstrate that it is only possible to learn bounded systems with a stochastic GP-SSM using a squared exponential kernel. The derived results are illustrated in numerical simulations. We start with the definition of stochastic GP-SSMs. In Sections 3.4.2 and 3.4.3, a method for calculating equilibrium distributions of stochastic GP-SSMs is presented. Section 3.4.4 investigates with the mean square boundedness and mean square attractive sets of stochastic GP-SSMs. Finally, Section 3.4.5 shows relevant illustrations of the previously presented methods.

3.4.1 Stochastic GP-SSM

The GP-ASSM, introduced in Definition 3.1, is taken as basis for the stochastic GP-SSM. For the sake of simplicity, we assume that the prior mean functions $m^1, \dots, m^{n_x}$ are set to zero. However, the results can be easily extended using a definition as in (2.55). We focus on autonomous stochastic GP-SSMs, as we want to analyze its equilibrium distributions.

Definition 3.4. An autonomous *stochastic GP-SSM* is given by the state space equation

$$\boldsymbol{x}_{t+1} = \boldsymbol{\mu}(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \mathcal{D}) + \Sigma(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \mathcal{D})\boldsymbol{\nu}, \tag{3.70}$$

where $\boldsymbol{\mu}$ and Σ are defined by (2.15) and $\boldsymbol{\nu} \sim \mathcal{N}(\boldsymbol{0}, I_{n_x})$ is a normally distributed random variable.

Hence, an autonomous stochastic GP-SSM is a GP-ASSM with maximum length of memory $\overline{m} = 0$, where the external input $\boldsymbol{u}_t$ is suppressed.

3.4.2 Equilibrium Distribution

The analysis of equilibrium points of stochastic systems requires first of all a definition of the stochastic equilibrium. In case the variance is disregarded, a deterministic approach can be used. To consider the probabilistic nature of the state variable, however, an equilibrium is defined by an invariant distribution of the current state $\boldsymbol{x}_t$ and the next state $\boldsymbol{x}_{t+1}$.
Assume that the current state is a random variable $\boldsymbol{x}_t$ with probability distribution $p(\boldsymbol{x}_t)$. The predictive distribution $p(\boldsymbol{x}_{t+1})$ is calculated by marginalizing over the state vector, following [Gir+03],

$$p(\boldsymbol{x}_{t+1}) = \int p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \mathcal{D})p(\boldsymbol{x}_t)d\boldsymbol{x}_t. \tag{3.71}$$

The probability distribution $p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \mathcal{D})$ is Gaussian

$$p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \mathcal{D}) = \mathcal{N}\left(\boldsymbol{\mu}(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \mathcal{D}), \Sigma(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \mathcal{D})\right), \tag{3.72}$$

as defined in (3.70). An analytic solution of the integral in (3.71) is not possible in general but still traceable for some special cases, e.g., if the distribution $p(\boldsymbol{x}_t)$ is also normal. A solution for arbitrary distributions of $\boldsymbol{x}_t$ can in general solely be found by numerical computation. To qualify as an equilibrium, the distribution of $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$ must be equal. This condition transforms the predictive distribution equation into a linear, homogeneous Fredholm integral equation of the second kind, see [Hac12]. The definition of this integral equation is given by

$$\underbrace{u_{\text{frd}}(\boldsymbol{x}_{t+1})}_{p(\boldsymbol{x}_{t+1})} = \lambda_{\text{frd}} \int \underbrace{f_{\text{frd}}(\boldsymbol{x}_{t+1}, \boldsymbol{x}_t)}_{p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t,\mathcal{D})} \underbrace{u_{\text{frd}}(\boldsymbol{x}_t)}_{p(\boldsymbol{x}_t)} \, d\boldsymbol{x}_t, \tag{3.73}$$

where $\lambda_{\text{frd}} \in \mathbb{R}$, $\boldsymbol{x}_t, \boldsymbol{x}_{t+1} \in \mathcal{X}$ and $f_{\text{frd}} \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ are known piece-wise continuous functions while $u_{\text{frd}} \colon \mathcal{X} \to \mathbb{R}$ is an unknown function. The function $f_{\text{frd}}(\cdot, \cdot)$ is known as kernel and λ_{frd} is the eigenvalue of the integral equation.

Remark 3.12. *The kernel f_{frd} of a Fredholm integral equation is not to be confused with the kernel k of a GP.*

For the sake of clarity, we assume a one-dimensional system with $\mathcal{X} = \mathbb{R}$ in the following. The extension to the multidimensional case is presented at the end of this section. A numerical solution of the integral equation (3.73) can be found by using the Nyström method, which approximates the integral by a finite sum, e.g., employing the trapezoid rule, see [Jer99]. For this approach, the integral has to be defined on a finite interval $[a, b]$ with $a, b \in \mathbb{R}$ and the function f_{frd} must be continuous. The length of the interval should be chosen sufficiently large. The interval is divided into n_i equal parts of width $\Delta z = \frac{b-a}{n_i}$. Additionally, let $z_i = a + i\Delta z$ and $z_{n_i} = b$.

Remark 3.13. *Instead of the Nyström method, a Monte-Carlo approach can be used to find a solution for the integral equation* (3.73). *However, the Monte-Carlo approach is computational demanding especially for high-dimensional state spaces.*

The solution $u_{\text{frd}}(\cdot)$ of the integral equation (3.73) can be approximated by the matrix equation $M_f \boldsymbol{u}_f = \boldsymbol{0}$ with $M_f \in \mathbb{R}^{n_i \times n_i}$ and vector $\boldsymbol{u}_f \in \mathbb{R}^{n_i}$. We obtain

$$M_f = \begin{bmatrix} \frac{1}{\lambda_{\text{frd}}} - \frac{\Delta x}{2} f_{\text{frd}}(z_0, z_0) & -\Delta z f_{\text{frd}}(z_0, z_1) & \dots & -\frac{\Delta z}{2} f_{\text{frd}}(z_0, z_{n_i}) \\ -\frac{\Delta z}{2} f_{\text{frd}}(z_1, z_0) & \frac{1}{\lambda_{\text{frd}}} - \Delta z f_{\text{frd}}(z_1, z_1) & \dots & -\frac{\Delta z}{2} f_{\text{frd}}(z_1, z_{n_i}) \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{\Delta z}{2} f_{\text{frd}}(z_{n_i}, z_0) & -\Delta z f_{\text{frd}}(z_{n_i}, z_1) & \dots & \frac{1}{\lambda_{\text{frd}}} - \frac{\Delta z}{2} f_{\text{frd}}(z_{n_i}, z_{n_i}) \end{bmatrix} \tag{3.74}$$

for $i, j = 0, 1, \dots, n_i$. The vector $\boldsymbol{u}_f$ contains the approximation of the function values of $u_{\text{frd}}(\cdot)$ at z_i. If and only if $\det M_f = 0$, there exists an infinite number of non-zero solutions. This condition must be satisfied for $\lambda_{\text{frd}} = 1$ to fulfill (3.71). Additionally, $u_{\text{frd}}(\cdot)$ must satisfy the constraints for a probability distribution, namely

$$\int u_{\text{frd}}(z) dz = 1 \text{ and } u_{\text{frd}}(z) \geq 0, \, \forall z \in \mathbb{R}. \tag{3.75}$$

To find an appropriate solution, the linear equation $M_f \boldsymbol{u}_f = \boldsymbol{0}$ with the constraints (3.75) must be solved. We use again the trapezoid rule to discretize the constraints

$$\Delta x \sum_{i=0}^{n_i} u_{f,i} - \frac{\Delta z}{2}(u_{f,0} + u_{f,n_i}) = 1, \quad u_{f,i} \geq 0,\ \forall i = 0, 1, \ldots, n_i \tag{3.76}$$

and add the constraint given by (3.76) to the matrix M_f. The result is a non-homogeneous system of linear equations which can be formulated as least square optimization problem given by

$$\begin{aligned} &\min_{\boldsymbol{u}_f} \|\check{M}_f \boldsymbol{u}_f - \boldsymbol{b}_f\|^2 \text{ with } u_{f,i} \geq 0,\ \forall i = 0, 1, \ldots, n_i \\ &\check{M}_f = \begin{bmatrix} & & M_f & & \\ \frac{\Delta z}{2} & \Delta z & \ldots & \Delta z & \frac{\Delta z}{2} \end{bmatrix}, \quad \boldsymbol{b}_f = [0, \ldots, 0, 1]^\top. \end{aligned} \tag{3.77}$$

If the residual of the optimization is sufficiently small, the vector $\boldsymbol{u}_f$ is a discrete approximation of $p(x_t)$ at $x_t = a + i\Delta z$ for $i = 0, 1, \ldots, n_i$ which solves equation (3.71).

If the system has more than one dimension, the numerical integration scheme for the Fredholm integral equation must be adapted. In general, a numerical approximation for an integral of a continuous function $f_{\text{gen}} \colon \mathcal{X} \to \mathbb{R}$ over a closed and bounded set $\mathcal{X}$ in $\mathbb{R}^{n_x}$ is given by

$$\int_{\mathcal{X}} f_{\text{gen}}(z) dz \approx \sum_{i=0}^{n_i} w_i f_{\text{gen}}(z_i), \quad \text{with } w_i \in \mathbb{R},\ z_i \in \mathcal{X}. \tag{3.78}$$

The numerical approximation, that we present earlier in (3.74), must converge to the true integral for $n_i \to \infty$ to be valid for the introduced algorithm. The multidimensional trapezoid rule satisfies the required convergence. With this approach, the integral approximation (3.74) is straightforward adapted for the application in higher dimensional systems.

Algorithm 1 describes the computation of the approximation of $p(\boldsymbol{x}_t)$ in higher dimensional systems.

3.4.3 Remarks on Convergence

For a numerical approach it is essential to analyze the convergence of the algorithm. The following proposition ensures the convergence of the proposed approach.

Proposition 3.11. *Assume a finite interval $[a, b]$ with boundaries $a, b \in \mathbb{R}$ and a continuous solution $p(x_{t+1}) = p(x_t)$ of the integral equation*

$$p(x_{t+1}) = \int_a^b p(x_{t+1}|x_t, \mathcal{D}) p(x_t) dx_t.$$

The numerical solution $p_{num}(x_t)$ given by the Nyström method with the trapezoid rule converges to the exact solution $p(x_t)$, if the step size $\Delta z = \frac{b-a}{n_i} \to 0$ with $n_i \to \infty$ and

$$\begin{aligned} &\Delta z \sum_{i=0}^{q} p_{num}(a + i\Delta z) - \frac{\Delta z}{2}(p_{num}(a) + p_{num}(b)) = \int_a^b p(x_t) dx_t \\ &\text{with } p_{num}(a + i\Delta z) \geq 0,\ \forall i = 0, 1, \ldots, n_i. \end{aligned}$$

Algorithm 1 Equilibrium distribution

```
$\boldsymbol{b}_f \leftarrow [0,\dots,0,1]^T$, $\boldsymbol{z} \leftarrow [a,\dots,b]^T$
for $z_i, i = 1,\dots,n_i$ do
    for $z_j, j = 1,\dots,n_i$ do
        $x_t \leftarrow z_j$
        $\mu(x_{t+1}) \leftarrow \boldsymbol{k}^\top (K + I_{n_\mathcal{D}}\sigma_n^2)^{-1} Y$
        $\mathrm{var}(x_{t+1}) \leftarrow k - \boldsymbol{k}^\top (K + I_{n_\mathcal{D}}\sigma_n^2)^{-1} \boldsymbol{k}$
        $(M_f)_{i,j} \leftarrow p(z_i|x_t, \mathcal{D})$
    end for
end for
$M_f = I - \mathrm{weighted}(M_f)$
if $\det M_f \neq 0$ then
    return No solution
else
    $\check{M}_f = \begin{bmatrix} M_f \\ \text{normalized weights} \end{bmatrix}$
    $\min_{\boldsymbol{u}_f} \|\check{M}_f \boldsymbol{u}_f - \boldsymbol{b}_f\|_2^2$ with $u_i \geq 0, \forall i = 0,1,\dots,n_i$
    return $\boldsymbol{u}_f$
end if
```

Proof. We start with the definition of the integral operator $\mathfrak{I}$ and the numerical integral operator $\mathfrak{I}_n$

$$\mathfrak{I}p(x_{t+1}) = \int_a^b p(x_{t+1}|x_t)p(x_t)dx_t \tag{3.79}$$

$$\mathfrak{I}_n p(x_{t+1}) = \sum_{i=0}^{n_i} w_i p(x_{t+1}|a + i\Delta z)p(a + i\Delta z) \tag{3.80}$$

with $x_{t+1} \in [a, b], w_i \in \mathbb{R}$. If the numerical integrator operator is based on the trapezoid rule with step size $\Delta z = \frac{b-a}{n_i}$, the numerical integral converges to the true integral

$$\int_a^b f_{\mathrm{gen}}(z)dz \approx \Delta z \sum_{i=0}^{n_i} f_{\mathrm{gen}}(a + i\Delta z) - \frac{\Delta z}{2}(f_{\mathrm{gen}}(a) + f_{\mathrm{gen}}(b)) \tag{3.81}$$

for any continuous function f_{gen}, see [SB13]. The speed of convergence of $p_{\mathrm{num}}(x_t)$ to the exact solution $p(x_t)$ of (3.71) depends on the numerical integration error of the trapezoid rule, given by

$$(\mathfrak{I} - \mathfrak{I}_n)p(x_t) = -\frac{\Delta z^2}{12}\left[\frac{\partial p(x_{t+1}|x_t)p(x_t)}{\partial x_t}\right]_{x_t=a}^{x_t=b} + O(\Delta z^4), \tag{3.82}$$

where O denotes the *Big O notation*. As the difference $(\mathfrak{I} - \mathfrak{I}_n)p(x_t)$ converges to zero for $n_i \to \infty$ and $\|p(x_t) - p_{\mathrm{num}}(x_t)\|_\infty \leq c_s\|(\mathfrak{I} - \mathfrak{I}_n)p(x_t)\|$ with a constant $c_s < \infty$, the numerical solution $p_{\mathrm{num}}(x_t)$ converges to the exact solution $p(x_t)$, see [Atk97]. □

3.4.4 Stability

The previous section investigates with the numerical approximation of equilibrium distributions. Another important property of dynamical systems is the stability. Several different stability measures exist for stochastic systems. In this section, we focus on the widespread mean square measure and positive recurrent sets. Further, we focus on stochastic GP-SSMs with the squared exponential kernel, since this kernel is a very common choice for GP-SSMs. The following theorem presents the mean square boundedness for such a stochastic GP-SSM.

Theorem 3.11. *Consider an autonomous stochastic GP-SSM, as defined in* (3.70)*, with squared exponential kernel*

$$k(\boldsymbol{z}, \boldsymbol{z}') = \varphi_{1,i}^2 \exp\left(-\frac{\|\boldsymbol{z} - \boldsymbol{z}'\|^2}{2\varphi_{2,i}^2}\right), \; \boldsymbol{z}, \boldsymbol{z}' \in \mathbb{R}^{n_x},$$

where $\varphi_{1,i} \in \mathbb{R}_{\geq 0}$ and $\varphi_{2,i} \in \mathbb{R}_{>0}$ for all $i \in \{1, \dots, n_x\}$ with $n_{\mathcal{D}}$ training points. Then, the state $\boldsymbol{x}_t$ is mean square bounded by

$$\sup_{t \in \mathbb{N}_{>0}} \mathbb{E}\left[\|\boldsymbol{x}_t\|^2\right] \leq \sum_{i=1}^{n_x} \varphi_{1,i}^4 n_{\mathcal{D}} \|\boldsymbol{\alpha}(i)\|^2 + \varphi_{1,i}^2$$

with $\boldsymbol{\alpha}(i)$ defined in (3.51)*.*

Before starting the proof, note that the squared exponential kernel is bounded, as presented in (3.62) and (3.63). We use this property for the following proof.

Proof. We prove the mean square boundedness by evaluating the expected value $\mathbb{E}\left[\|\boldsymbol{f}(\boldsymbol{x}_t)\|^2\right]$ for each $\boldsymbol{x}_t$. The expected value of a squared Gaussian distributed variable is calculated by

$$\mathbb{E}\left[\boldsymbol{x}_{t+1}^\top \boldsymbol{x}_{t+1}\right] = \sum_{i=1}^{n_x} \mu_i^2(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \mathcal{D}) + \operatorname{var}_i(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \mathcal{D}), \tag{3.83}$$

where $\mu_i(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \mathcal{D}) := \mu(x_{t+1,i}|\boldsymbol{x}_t, \mathcal{D})$ and $\operatorname{var}_i(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \mathcal{D}) := \operatorname{var}(x_{t+1,i}|\boldsymbol{x}_t, \mathcal{D})$. If the squared mean $\mu_i^2(\cdot)$ and the variance $\operatorname{var}_i(\cdot)$ are bounded, then (3.83) is bounded. The mean $\mu_i(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \mathcal{D})$ is bounded with

$$\begin{aligned} \|\, \mu_i(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \mathcal{D})\| \leq \varphi_{1,i}^2 \sqrt{n_{\mathcal{D}}} \|\boldsymbol{\alpha}(i)\| \Rightarrow \mu_i^2(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \mathcal{D}) \leq \varphi_{1,i}^4 n_{\mathcal{D}} \|\boldsymbol{\alpha}(i)\|^2 \\ \text{with } \boldsymbol{\alpha}_i = (K + I_{n_{\mathcal{D}}} \sigma_{n,i}^2)^{-1} Y_{:,i}, \end{aligned} \tag{3.84}$$

applying the Cauchy-Schwarz inequality and the bounds of the squared exponential kernel, given by (3.62) and (3.63). The variance

$$\operatorname{var}_i(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \mathcal{D}) = k(\boldsymbol{x}_t, \boldsymbol{x}_t) - \boldsymbol{k}(\boldsymbol{x}_t, X)^\top (K + I_{n_{\mathcal{D}}} \sigma_{n,i}^2)^{-1} \boldsymbol{k}(\boldsymbol{x}_t, X) \tag{3.85}$$

is also bounded by $0 \leq \operatorname{var}_i(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \mathcal{D}) \leq \varphi_{1,i}^2$ because of (3.62) and (3.63) and the positive definiteness of the matrix $(K(X, X) + I_{n_{\mathcal{D}}} \sigma_{n,i}^2)^{-1}$. Note, that the bound for the mean and the variance hold for any time step $t \in \mathbb{N}$. Therefore, the solution $\boldsymbol{x}_t$ with $t > 0$ of system (3.70) is mean square bounded with

$$\sup_{t \in \mathbb{N}_{>0}} \mathbb{E}\left[\|\boldsymbol{x}_t\|^2\right] \leq \sum_{i=1}^{n_x} \varphi_{1,i}^4 n_{\mathcal{D}} \|\boldsymbol{\alpha}(i)\|^2 + \varphi_{1,i}^2. \tag{3.86}$$

□

Theorem 3.11 can be interpreted as follows: Since the mean and the variance of $\boldsymbol{x}_{t+1}|\boldsymbol{x}_t$ are bounded, a stochastic GP-SSM with squared exponential kernel is solely capable of learning bounded trajectories. The upper bound (3.86) depends on the signal variance φ_1 and noise variance σ_n, the number of training points $n_{\mathcal{D}}$, and their position. The value of the upper bound increases if the number of training points $n_{\mathcal{D}}$ or the values of the output training data Y increase.
Next, we focus on the behavior of the trajectories of the stochastic GP-SSM. For this purpose, we use the theory of Markov chains because the future state of the system (3.70) only depends on the current state and, thus, it is Markovian.

Theorem 3.12. *Consider an autonomous stochastic GP-SSM, as defined in* (3.70)*, with squared exponential kernel*

$$k(\boldsymbol{z}, \boldsymbol{z}') = \varphi_{1,i}^2 \exp\left(-\frac{\|\boldsymbol{z} - \boldsymbol{z}'\|^2}{2\varphi_{2,i}^2}\right), \; \boldsymbol{z}, \boldsymbol{z}' \in \mathbb{R}^{n_x},$$

where $\varphi_{1,i} \in \mathbb{R}_{\geq 0}$ and $\varphi_{2,i} \in \mathbb{R}_{>0}$ for all $i \in \{1, \dots, n_x\}$ with $n_{\mathcal{D}}$ training points. Then, there exists a set

$$\mathcal{I} = \{\boldsymbol{z} \in \mathbb{R}^{n_x} | \; \|\boldsymbol{z}\|^2 \leq \sum_{i=1}^{n_x} \varphi_{1,i}^4 n_{\mathcal{D}} \|\boldsymbol{\alpha}(i)\|^2 + \varphi_{1,i}^2\},$$

which is positive recurrent.

Proof. First, we recall the criterion for positive recurrent sets: Positive recurrency guarantees that the system trajectory returns to a set in a finite time horizon. A condition for the existence of a positive recurrent set is given in the following lemma.

Lemma 3.1 ([Kus71])**.** *Suppose that there exists a positive definite Lyapunov function $V(\boldsymbol{x}_t)$ for $\boldsymbol{x}_t \in \mathbb{R}^{n_x}$ and positive constants $c_1, c_2, c_3 \in \mathbb{R}_{>0}$ such that*

$$\begin{aligned} &\mathbb{E}\left[V(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)\right] - V(\boldsymbol{x}_t) \leq -c_2, && \text{if } V(\boldsymbol{x}_t) > c_1 \\ &\mathbb{E}\left[V(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)\right] - V(\boldsymbol{x}_t) \leq c_3 < \infty, && \text{if } V(\boldsymbol{x}_t) \leq c_1. \end{aligned}$$

Then, the set $\mathcal{I} = \{\boldsymbol{z} \in \mathbb{R}^{n_x} \colon V(\boldsymbol{z}) \leq c_1\}$ is positive recurrent.

We assume the positive definite Lyapunov function $V(\boldsymbol{x}_t) = \boldsymbol{x}_t^\top \boldsymbol{x}_t$ with $\boldsymbol{x}_t \in \mathbb{R}^{n_x}$. The drift of $V(\boldsymbol{x}_t)$ is given by

$$\Delta V = \mathbb{E}\left[V(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)\right] - V(\boldsymbol{x}_t) = \mathbb{E}\left[\boldsymbol{x}_{t+1}^\top \boldsymbol{x}_{t+1}\right] - \boldsymbol{x}_t^\top \boldsymbol{x}_t. \tag{3.87}$$

An upper bound for $\left[\boldsymbol{x}_{t+1}^\top \boldsymbol{x}_{t+1}\right]$ is given by (3.86), which induces

$$\Delta V \leq \sum_{i=1}^{n} \varphi_{1,i}^4 n_{\mathcal{D}} \|\boldsymbol{\alpha}(i)\|^2 + \varphi_{1,i}^2 - \boldsymbol{x}_t^\top \boldsymbol{x}_t. \tag{3.88}$$

Due to the fact that $\lim_{\|\boldsymbol{x}_t\| \to \infty} \boldsymbol{x}_t^\top \boldsymbol{x}_t = \infty$ is unbounded and the upper bound (3.88) holds, there must exist a set $\mathcal{I}$ with a neighbourhood $\mathcal{I}_n = \mathbb{R}^{n_x} \setminus \{\mathcal{I}\}$ which fulfills $\Delta V < 0$, $\boldsymbol{x}_t \in \mathcal{I}_n$.

The drift ΔV is negative if $\boldsymbol{x}_t^\top \boldsymbol{x}_t > \mathbb{E}\left[\boldsymbol{x}_{t+1}^\top \boldsymbol{x}_{t+1}\right]$. Therefore, the set $\mathcal{I}$ is defined by

$$\mathcal{I} = \{\boldsymbol{z} \in \mathbb{R}^{n_x} - \Big| \left\|\boldsymbol{z}\right\|^2 \leq \sum_{i=1}^{n_x} \varphi_{1,i}^4 n_{\mathcal{D}} \|\boldsymbol{\alpha}(i)\|^2 + \varphi_{1,i}^2\}. \tag{3.89}$$

Since the drift of the Lyapunov function is negative outside the set $\mathcal{I}$, Lemma 3.1 is fulfilled and, thus, the set is positive recurrent. □

3.4.5 Numerical Examples

Equilibrium Distribution

In this section, we present examples of equilibrium distributions of a one-dimensional stochastic GP-SSM with squared exponential kernel. The solution is validated by a Monte Carlo experiment and a two-sample Kolmogorov-Smirnov test. We assume a system described by

$$x_{t+1} = 0.01x_t^3 - 0.2x_t^2 + 0.2x_t + \nu, \tag{3.90}$$

where ν is standard normal distributed. A stochastic GP-SSM with squared exponential kernel is trained with 20 input points, which are uniformly distributed on the interval $[-5, 5]$, and the corresponding output data. The output data is corrupted by a Gaussian noise with a variance of $\sigma_n = 1$. The hyperparameters are optimized by maximizing the marginal likelihood. The obtained value for the lengthscale φ_2 is 3.59 and the signal noise φ_1 is 4.21. The predicted mean function (red) and the variance (gray shaded area) of the GP-SSM are visualized in Fig. 3.13. The equilibrium distribution (black line) has a non-Gaussian shape

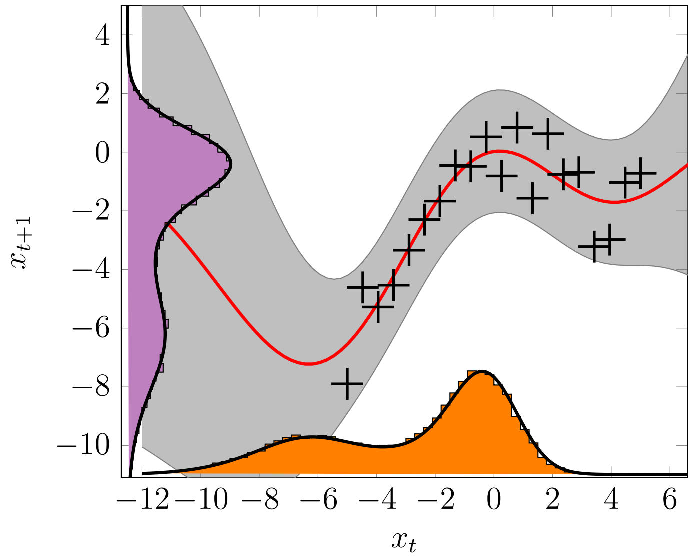


Figure 3.13: A stochastic GP-SSM with squared exponential function trained by 20 noisy data points (black crosses). The stochastic GP-SSM determines the mean function (red) and variance (gray). The black lines, at the bottom and on the left side, describe the computed equilibrium distribution. A Monte Carlo experiment with the input samples (orange) based on the equilibrium distribution and the output samples (purple) supports that the distribution is an equilibrium.

which is due to the strong nonlinear behavior of the mean function. To cover the relevant area of the distribution, we set the interval of the integral to $[-12, 8]$ and divide it in $n_i = 150$ sections. The determinant of integral approximation matrix M_f, given in (3.74), is zero and an active-set algorithm finds a minimum for $||\check{M}_f \boldsymbol{u}_f - \boldsymbol{b}_f||^2$ satisfying the constraint (3.75). The optimization results in a residual value of 2.3e-6. To validate the computed distribution, we use the inverse transform sampling method, introduced in [Dev86], to generate 30000 samples out of this distribution. Since the inverse of the cumulative distribution function is necessary for the inverse transform sampling method, the discrete points p_i of the probability density function are numerically integrated. These samples are visualized by the orange bars at the bottom of Fig. 3.13. The purple bars on the left side show the output distribution of the samples. It can be seen that the shape of the proposed equilibrium distribution (black line) coincide with the samples before (bottom) and after (left) the transition through the stochastic GP-SSM.

Furthermore, a two-sample Kolmogorov-Smirnov test returns that it is not possible to reject the null hypothesis, stating that the proposed distribution and the true equilibrium distribution are identical at the 5% significance level. This means, the Monte Carlo experiment and the Kolmogorov-Smirnov test support the assumption that the calculated distribution function $p_{\text{num}}(x_t)$ is an equilibrium distribution of the stochastic GP-SSM.

Stability

In the following example, we illustrate the boundedness of the stochastic GP-SSM with squared exponential kernel. We use again the highly nonlinear Van der Pol oscillator as training system, see (3.69), where the sample time t_s is set to 0.1 s and the additional parameter ϵ to -0.8.

Gaussian distributed noise $\nu_1, \nu_2 \sim \mathcal{N}(0, 0.01^2)$ is added to the output data set. The stochastic GP-SSM is trained with 441 uniformly distributed points on the square $[-3, 3] \times [-3, 3]$. The hyperparameters are optimized by the minimization of the log-likelihood function with a conjugate gradient method. For the multi-step ahead prediction not only the mean but also the uncertainty is considered, see [Gir+03]. Since the trajectory stays inside the training area, the predicted trajectory is very close to the true trajectory. The mean square boundedness of the stochastic GP-SSM is fulfilled.

The model is tested with two different set of initial points $\boldsymbol{x}_0$. For $\boldsymbol{x}_0 = [-1.8, 0]^\top$, Fig. 3.14 shows the trajectory of the system (3.69) and the mean of $\boldsymbol{x}_t$ with the 2-sigma standard deviation of the stochastic GP-SSM. The stochastic GP-SSM can predict sufficiently accurate the trajectory of (3.69) that manifests in a very low variance.

In the second case, the initial state of the system is changed to $\boldsymbol{x}_0 = [2.2, 0]^\top$ which generates an unstable trajectory, see Fig. 3.15. Due to the fact that this initial point is not inside the attraction area of the Van der Pol oscillator, the trajectory x_t, y_t of the system is not bounded. Nevertheless, the GP-SSM generates a bounded mean and variance function. This test case demonstrates the boundedness of the stochastic GP-SSM. The increased variance shows the uncertainty of the prediction since the model generates always bounded trajectories.

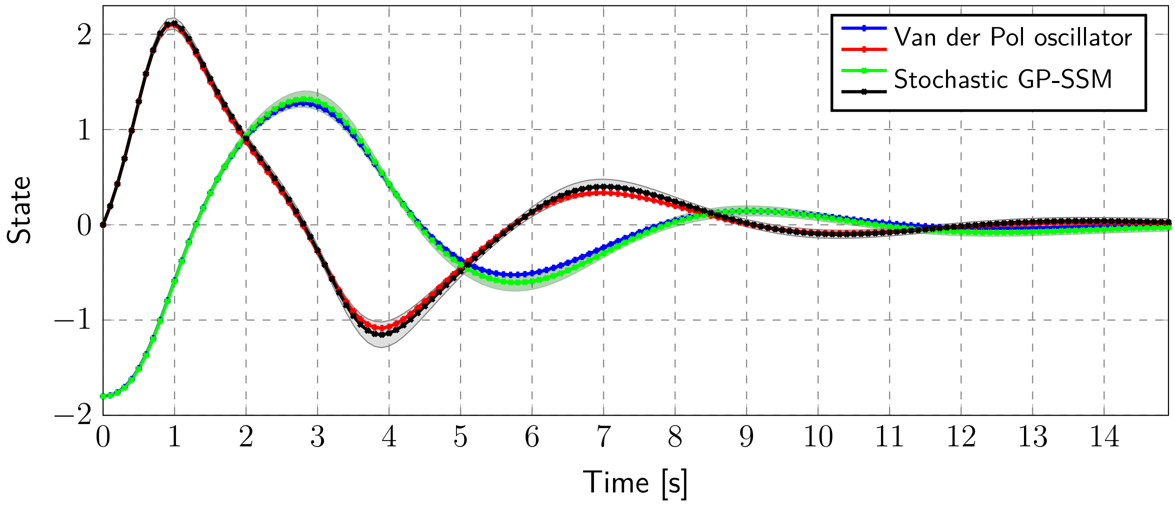


Figure 3.14: The mean (green and black) and the 2-sigma standard deviation (green and black shaded area) of the stochastic GP-SSM with squared exponential kernel. The prediction is close to the true system states.

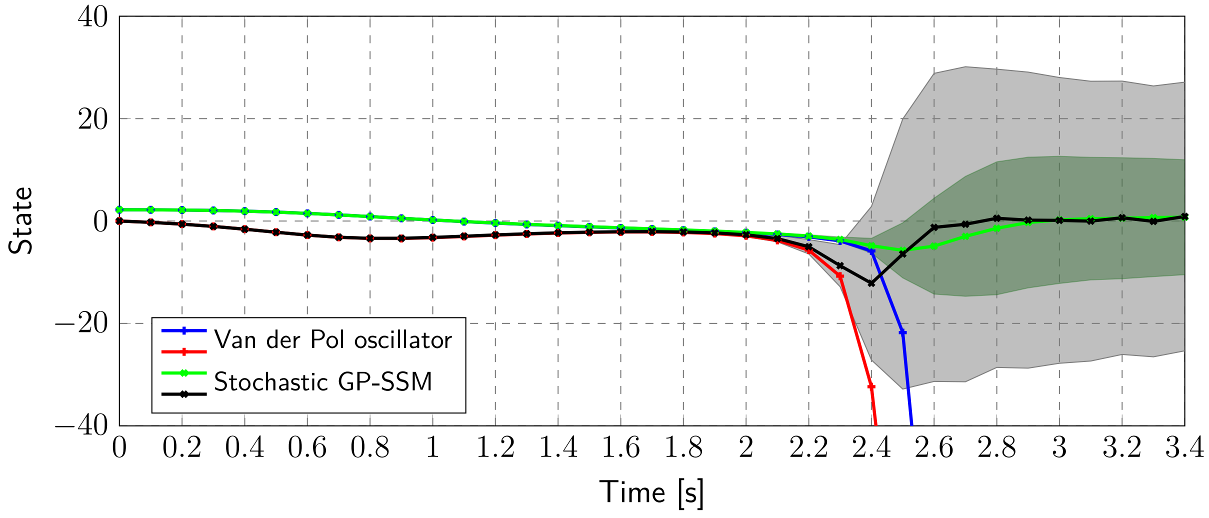


Figure 3.15: The prediction of the mean (green and black) and the 2-sigma standard deviation (green and black shaded area) of a stochastic GP-SSM with squared exponential kernel is bounded even if the trajectory of the original system is unbounded. For testing purpose, the stochastic GP-SSM should generate an unbounded trajecory. Since the stochastic GP-SSM is bounded, the trajectory of the true system is not reproduced that manifests in a high variance of the predicted states.

3.5 Discussion

In this chapter, we show that the sampling of GPDMs, avoiding the impossible sampling of infinite dimensional objects, leads to non-Markovian dynamics. This characteristic is surprising as the representation of the GP-SSM and GP-NOE model, given by (2.55) and (2.56) respectively, is based on a Markovian state space structure. However, the covariance term of the GP introduces dependencies across the states that leads to dependencies across time for GPDMs. Thus, the sampling of GP-SSMs and GP-NOE models generates non-Markovian dynamics, which we analyze from a control theoretical point of view. More precisely, a general description for approximations based on the number of included past states/outputs is presented and compared against the true sampling. The approximation error of these models is analyzed with respect to the Kullback-Leibler divergence, the mean square prediction error and the variance of the prediction. Furthermore, we prove that the true variance of the next state ahead is always less than the variance of the approximated model. This is relevant for the usage of the approximation in variance based control approaches such as risk-sensitive control approaches, e.g., [MSH13; LK07]. Additionally, the boundedness of GPDMs with bounded mean and variance functions, such as the commonly used squared exponential function, is proven. The boundedness is an important property for the identification of unknown systems with GPDMs and is likewise exploited for robustness analysis in GPDM based control approaches, see [FKK98]. The introduced characteristics about the relation between the boundedness of the true sampling and the approximations allows a safe usage of the approximation.

For the approximation with the maximum length of memory $\overline{m} = 0$, we derived more detailed results. The analysis is split into deterministic and stochastic GP-SSMs. For the first mentioned, we investigate the number of equilibrium points and stability properties for various kernels. In particular, we study deterministic GP-SSMs with squared exponential, linear, and polynomial kernel. A deterministic GP-SSM with squared exponential kernel generates always at least one equilibrium and is globally uniformly ultimately bounded. Therefore, it is not possible to learn unbounded trajectories with this approach. The linear kernel generates one equilibrium point except for pathological cases. The number of equilibrium points of a deterministic GP-SSM with polynomial function is always equal to or less than the degree of the polynomial. Two examples visualize the presented properties.

For the stochastic GP-SSM, we present an algorithm for the computation of equilibrium distributions. The method bases on the solution of a Fredholm integral equation which is obtained by numerical approximation. The result is a system of linear equations and constraints to ensure the solution to be a valid probability distribution. The second part deals with the proof of the mean square boundedness of a stochastic GP-SSM with squared exponential kernel. We also show that there exists a set which is positive recurrent. Therefore, the stochastic GP-SSM is solely capable of reproducing bounded trajectories with the squared exponential kernel. The computation of equilibrium distributions is validated in a simulation which uses input sample points that are generated by the equilibrium distribution. A simulation of a discrete Van der Pol oscillator shows the mean square boundedness. The results for the deterministic and stochastic GP-SSM show the implications of system properties based on the selected kernel function. This allows not only the safe application of these GPDMs in control approaches but also the incorporation of control theoretical prior knowledge.

Kernel Selection of GP Models

4

In the previous chapters, we point out the essential role of the kernel function in GP models. In addition, the hyperparameters must be determined before the models can be used. If prior knowledge about the system is existent, the kernel can be selected based on the known system properties as presented in Chapter 3. Without any prior knowledge, the selection of the kernel and its hyperparameter is often performed by minimizing a cost function. Commonly, these selections are data-based, for instance through minimizing a loss function that is often a trade-off between the prediction error and the complexity of the model, as presented in Section 2.2.
However, if the model is used in model-based control scenarios, the goal is typically to minimize the control error or maximizing the performance of the control-loop. In this case, the data-based selection neglects the fact that the learned model is used for the control of the actual system, which can result in reduced controller performance [AQN06; Gev05; HGD96]. Thus, the objective of this section is a kernel and hyperparameter selection, which are control related and not data-based only. Furthermore, if the selected kernel does not match to the data generating system, we aim to find an upper bound for the model error.

Related Work and Open Problems

In general, the problem is that only a finite data set is available to derive the kernel function. In addition, the kernel function typically depends on a number of hyperparameters. There exist many different methods to estimate these parameters based on the training data set, e.g., marginal likelihood optimization. However, the involved optimization problems are in general non-convex, such that the marginal likelihood may have multiple local optima [Ras06]. Alternatively, there exists the cross validation approach which deals with a validation and a test set to carry out the hyperparameter selection. From the control perspective, system identification approaches aim to obtain an open-loop dynamics model of the system by minimizing the state prediction error. This problem has been well studied in literature for both linear systems, e.g., [Lju98], as well as for nonlinear systems using the function approximators such as GP [CHK12; Uml+17; BUH17] and neural networks (NN) [NP90; Ban+16]. However, a model obtained using this open-loop procedure can result in a reduced controller performance on the actual system [HGD96]. To overcome these challenges, adaptive control mechanisms and iterative learning control have been studied where the system dynamics or control parameters are optimized based on the performance on the actual system, e.g., [ÅW13; CKM85; BTA06]. However, these approaches are mostly limited to linear systems and controllers or assume at least a parametric system model. Recently, learning-based controller tuning mechanisms have also been proposed [LL13; Cal+15], but such methods might be highly data-inefficient for general nonlinear systems as they typically completely disregard the underlying dynamics [Rec18]. A Bayesian optimization (BO) based approach is presented in [Ban+18], wherein the authors also propose a goal-driven dynamics learning approach via BO, but for linear systems. To overcome the challenges of open-loop system identification, closed-loop system identification methods have been studied that lead to more robust control performance on the actual system [AQN06; Gev05; HGD96]. However, the model selection for GP models to optimize the control performance is still an unaddressed problem.

Furthermore, the variance of GP models is exploited in many different kinds of control approaches [MLH15; Bec+17; Koc+04]. In such cases, it is essential that the GP model fits the data generating system. Otherwise, the variance as prediction error measure is not valid anymore. Although GPs with universal kernel functions often produce satisfactory results, the selection of a suitable kernel function is a nontrivial problem [See99; PD11] without any guarantee that the kernel function and its hyperparameters fit the data generating process. As a consequence, the variance of the GP model may not correctly estimate the real model confidence. A lower bound for the prediction error for GP models with a misspecified kernel is given by [Wåg+17] whereas an upper bound is still missing. Using GP models in control, the upper bound is highly interesting for stability consideration based on robust control methods.

In Section 4.1, we present a closed-loop based model selection approach to find the kernel and hyperparameters that maximize a control-performance based cost function. In the case that a kernel is selected that does not fit to the regression problem, we derive an upper error bound for the model error in Section 4.2.

4.1 Closed-loop Model Selection using Bayesian Optimization

In this section, we propose a Bayesian optimization based active learning framework to optimize the kernel and its hyperparameters directly with respect to the performance of the closed-loop. This optimization is performed in a sequential fashion where at each step of the optimization, BO takes into account all the past data points and proposes the most promising kernel and hyperparameters for the next evaluation. The outcome is used to define a kernel-based model that is utilized by a given controller. The obtained model-based controller is then applied to the actual system in a closed-loop fashion to evaluate its performance. This information is then used by BO to optimize the next evaluation. Consequently, multiple evaluations on the actual system must be performed, which is often feasible such as for systems with repetitive trajectories. BO thus does not aim to obtain the most accurate dynamics model of the system, but rather to optimize the performance of the closed-loop system. Furthermore, our approach explicitly allows to preserve the convergence properties of the initial closed-loop system. We start with the formal problem description.

4.1.1 Problem Setting

Consider a discrete-time, potentially nonlinear system

$$\begin{aligned} \boldsymbol{x}_{t+1} &= \boldsymbol{f}_{\mathrm{ukn}}(\boldsymbol{x}_t, \boldsymbol{u}_t), \quad t = \{0, \dots, n_t - 1\}, n_t \in \mathbb{N} \\ \boldsymbol{y}_t &= \boldsymbol{g}_{\mathrm{ukn}}(\boldsymbol{x}_t, \boldsymbol{u}_t) \end{aligned} \tag{4.1}$$

in which $\boldsymbol{f}_{\mathrm{ukn}}, \boldsymbol{g}_{\mathrm{ukn}}$ are unknown functions of the state $\boldsymbol{x}_t \in \mathbb{R}^{n_x}$ and input $\boldsymbol{u}_t \in \mathbb{R}^{n_u}$. For the following, we assume that the state mapping $\boldsymbol{f}_{\mathrm{ukn}} \colon \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ and the output mapping $\boldsymbol{g}_{\mathrm{ukn}} \colon \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_y}$ are such that there exist a unique state and output trajectory for all $\boldsymbol{u}_t \in \mathbb{R}^{n_u}$ and $\boldsymbol{x}_0,\ t \geq 0$. We assume that a control law $\boldsymbol{u}_{\mathrm{ctrl}} \colon \mathbb{R}^{n_y} \times \mathbb{R}^{n_o} \to \mathbb{R}^{n_u}$

$$\boldsymbol{u}_t = \boldsymbol{u}_{\mathrm{ctrl}}(\boldsymbol{y}_t - \boldsymbol{r}_t, \boldsymbol{o}_t) \tag{4.2}$$

is given for the system (4.1). The reference $\boldsymbol{r}_t \in \mathbb{R}^{n_y}$ is assumed to be zero but the framework is also applicable for a varying signal. In addition to the reference, the control law also depends on the output $\boldsymbol{o}_t \in \mathbb{R}^{n_o}$ of a kernel-based model, a regression technique that uses a kernel to perform the regression in a higher-dimensional feature space. The output of a kernel-based model, $\boldsymbol{o}_t$, depends on the kernel function k, its hyperparameters $\boldsymbol{\varphi} \in \mathbb{R}^{n_\varphi}$ and system input and output, i.e., $\boldsymbol{o}_t = \boldsymbol{o}_{\mathrm{kbm}}(\boldsymbol{u}_{0:t-1}, \boldsymbol{y}_{0:t}, k, \boldsymbol{\varphi})$, where the function $\boldsymbol{o}_{\mathrm{kbm}}$ depends on the class of the kernel-based model, such as GP or support vector machines (SVM), used for the prediction.

Remark 4.1. *For example, the output $\boldsymbol{o}_t$ can be the prediction of the next state or output of the system based on the current state and input, using the mean and the variance of a GP model. This information can then be used by the controller to compute an appropriate system input $\boldsymbol{u}_t$.*

The control law $\boldsymbol{u}_{\mathrm{ctrl}}$ might be an output tracking controller designed based on the predicted model output. For possible control laws for different classes of systems, we refer to [CHK12; Uml+17; BUH17; SVD01; Ber+16]. The goal is to optimize the kernel and its hyperparameters such that the corresponding model output $\boldsymbol{o}_t$ minimizes the cost functional

$$c_{\mathrm{tot}}(\boldsymbol{y}_{0:t}, \boldsymbol{u}_{0:t}) = \sum_{k=0}^{n-1} c_{\mathrm{act}}(\boldsymbol{y}_t, \boldsymbol{u}_t), \tag{4.3}$$

where $c_{\mathrm{act}}(\boldsymbol{y}_t, \boldsymbol{u}_t) \colon \mathbb{R}^{n_y} \times \mathbb{R}^{n_u} \to \mathbb{R}$ represents the cost incurred for the control input $\boldsymbol{u}_t$ and the system output $\boldsymbol{y}_t$. The cost function here might represent the requirements concerning the closed-loop, e.g., an accurate tracking behavior or a minimized power consumption. Note that the cost functional in (4.3) implicitly depends on the kernel-based model $\boldsymbol{o}_{\mathrm{kbm}}$ through $\boldsymbol{u}_t$, see (4.2). The optimization of (4.3) is challenging since the system dynamics in (4.1) are unknown and the kernel-based model output $\boldsymbol{o}_t$ indirectly influences the cost. To overcome this challenge, we use BO to optimize the kernel and the hyperparameters based on the direct evaluation of the control law in (4.2) on the system (4.1) to find those that minimize the cost functional in (4.3).

4.1.2 Bayesian Optimization

Bayesian Optimization is an approach to minimize an unknown objective function based on (only a few) evaluated samples. We use BO to optimize the cost function (4.3) based on the kernel-based model as this is in general a non-convex optimization problem with unknown objective function (because the system dynamics are unknown), and probably multiple local extrema. BO is well-suited for this optimization as the task evaluations can be expensive and noisy [Sha+16]. Furthermore, BO is a gradient-free optimization method which only requires that the objective function can be evaluated for any given input. Since the objective function is unknown, the Bayesian strategy is to treat it as a random function with a prior, often as GP. Note that this GP here is used for the closed-loop cost functional approximation in BO and is not related to the kernel-based model for the controller (4.2) as stated in Remark 4.1. The prior captures the beliefs about the behaviour of the function, e.g., continuity or boundedness. After gathering the cost (4.3) of the task evaluation, the prior is updated to form the posterior distribution over the objective function. The posterior distribution is used to construct an acquisition function that determines the most promising kernel/hyperparameters for the next evaluation to minimize the cost. Different acquisition functions are used in literature to trade off between exploration of unseen kernel/hyperparameters and exploitation of promising combinations during the optimization process. Common acquisition functions are expected improvement, entropy search, and upper confidence bound [Moc12]. To escape a local objective function minimum, the authors of [Bul11] propose a method to modify the acquisition function when they seem to over-exploit an area, namely expected-improvement-plus. That results in a more comprehensive and also partially random exploration of the area and, thus it is probably faster in finding the global minimum. We also use this acquisition function for BO in our simulation and the experiment.

4.1.3 Closed-loop Model Selection Procedure

Our goal is to optimize the model's kernel and its hyperparameters with respect to the cost functional $c_{\text{tot}}(\boldsymbol{y}_{0:t}, \boldsymbol{u}_{0:t})$. Thus, in contrast to the classical kernel selection problem, where the kernel is selected to minimize the state prediction error, our goal here is not to get the most accurate model but the one that achieves the best closed-loop behavior. We now describe the proposed overall procedure for the kernel selection to optimize the closed-loop behavior; we then describe each step in detail.

We start with an initial kernel k with hyperparameters $\boldsymbol{\varphi}$, and obtain the control law for the system (4.1) using (4.2) with the model output $\boldsymbol{o}_t = \boldsymbol{o}_{\text{kbm}}(\boldsymbol{u}_{0:t-1}, \boldsymbol{y}_{0:t}, k, \boldsymbol{\varphi})$. This control law is then applied to the actual system, and the cost function (4.3) is evaluated after performing the control task, see Fig. 4.1. Depending on the obtained cost value, BO suggests a new kernel and corresponding hyperparameters for the kernel-based model $\boldsymbol{o}_{\text{kbm}}$ in order to minimize the cost function on the actual system. With this model, the control task is repeated and, based on the cost evaluation, BO suggests the next kernel and hyperparameters. This procedure

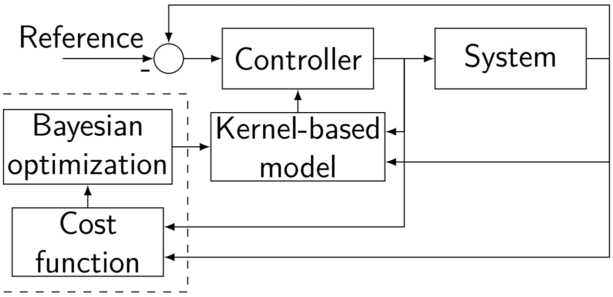


Figure 4.1: Closed-loop model selection for kernel-based models. Bayesian optimization is used to optimize the kernel and its hyperparameters directly based on the evaluation of a cost function.

is continued until a maximum number of task evaluations is reached or the user rates the closed-loop performance as sufficient enough. We now describe the above three steps, i.e., initialization, evaluation, and optimization, in detail.

Initialization

We define a set $\mathcal{K}_{\text{sub}} = \{k^1, \dots, k^{n_k}\}$ of $n_k \in \mathbb{N}$ kernel candidates k^j that we want to choose the kernel from for our kernel-based model. BO will be used to select the kernel with the best closed-loop performance in this set.

Remark 4.2. *The selection of possible kernels can be based on prior knowledge about the system, e.g., smoothness with the Matérn kernel or number of equlibria using a polynomial kernel, see [BH16b] and [Bis06] for general properties, respectively.*

In addition, each kernel depends on a set of hyperparameters. Since the number of hyperparameters could be different for each kernel, we define a set of sets $\Phi_{\text{set}} = \{\Phi^1, \dots, \Phi^{n_k}\}$ such that $\Phi^j \subset \mathbb{R}^{n_{\Phi^j}}$ is a closed set. Here, n_{Φ^j} represents the number of hyperparameters for the kernel k^j. Moreover, we assume that Φ^j is a valid *hyperparameter set*, see Definition 2.1. For the first evaluation of the closed-loop, the kernel-based model function $\boldsymbol{o}_{\text{kbm}}$ is created with an initial kernel k^j of the set $\mathcal{K}_{\text{sub}}$ and hyperparameters $\boldsymbol{\varphi}^j \in \Phi^j$ with $j \in \{1, \dots, n_k\}$.

Remark 4.3. *One potential way to select the initial kernel and hyperparameters is to set them equal to the kernel and hyperparameters of a prediction model that is optimized with respect to a loss function, e.g., using cross-validation or maximization of the likelihood function [Bis06].*

Task Evaluation

For the i-th task evaluation, BO determines an index value $j \in \{1, \dots, n_k\}$ and a $\boldsymbol{\varphi}^j \in \Phi^j$. The control law (4.2) for the kernel-based model $\boldsymbol{o}_{\text{kbm}}$, with the determined kernel k^j and hyperparameters $\boldsymbol{\varphi}^j$, is applied to the system (4.1)

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{f}_{\text{ukn}}(\boldsymbol{x}_t, \boldsymbol{u}_{\text{ctrl}}(\boldsymbol{y}_t, \boldsymbol{o}_{\text{kbm}}(\boldsymbol{u}_{0:t-1}, \boldsymbol{y}_{0:t}, k^j, \boldsymbol{\varphi}^j)) \\ \boldsymbol{y}_t &= \boldsymbol{g}_{\text{ukn}}(\boldsymbol{x}_t, \boldsymbol{u}_t) \text{ for } t = \{0, \dots, n_t - 1\} \end{aligned} \tag{4.4}$$

with fixed $\boldsymbol{x}_0 \in \mathbb{R}^{n_x}$.

Remark 4.4. *We focus here on a single, fixed initial state x_0. However, multiple (close by) initial states can be considered by using the expected cost across all initial states.*

The corresponding input and output sequences $\boldsymbol{u}_{0:t}$ and $\boldsymbol{y}_{0:t}$, respectively, are recorded. Afterwards, the cost function given by $c_{\text{tot}}(\boldsymbol{y}_{0:t}, \boldsymbol{u}_{0:t})$ is evaluated.

Model Optimization

In the next step, we use BO to minimize the cost function with respect to the kernel and its hyperparameters, i.e.,

$$[k^j, \boldsymbol{\varphi}^j] = \underset{j \in \{1,\dots,n_k\}, \boldsymbol{\varphi}^j \in \Phi^j}{\operatorname{argmin}} c_{\text{tot}}(\boldsymbol{y}_{0:t}, \boldsymbol{u}_{0:t}). \tag{4.5}$$

Thus, this problem involves continuous and discrete variables in the optimization task whereas classical BO assumes continuous variables only. To overcome this restriction, a modified kernel function for the BO is used. This kernel function is transformed in a way such that n_z continuous inputs and one integer-valued input are properly included. In detail, the modified kernel $k'_{GP}\colon (R^{n_z} \times \mathbb{N}) \times (R^{n_z} \times \mathbb{N}) \to \mathbb{R}$ is given by

$$k'_{GP}(\boldsymbol{z}, \boldsymbol{z}') = k_{GP}(T(\boldsymbol{z}), T(\boldsymbol{z}')), \tag{4.6}$$

where $k_{GP}\colon \mathbb{R}^{n_z+1} \times \mathbb{R}^{n_z+1} \to \mathbb{R}$ is a kernel over a continuous space and $T\colon (R^{n_z} \times \mathbb{N}) \to \mathbb{R}^{n_z+1}$ a transformation in which the integer-valued variable in $\boldsymbol{z}$ is rounded to the closest integer, see [GH20] for details.

Based on previous evaluations of the cost function, BO updates the prior and minimizes the acquisition function. The result is a kernel k^j and hyperparameters $\boldsymbol{\varphi}^j$ which are used in the model function $\boldsymbol{o}_{\mathrm{kbm}}(\boldsymbol{u}_{0:t-1}, \boldsymbol{y}_{0:t}, k^j, \boldsymbol{\varphi}^j)$. Then, the corresponding control law is evaluated again on the system and the procedure is repeated until a maximum number of task evaluations has been reached or a sufficient performance level has been achieved.

Example 4.1. In Fig. 4.2, we visualize a possible cost function c_{tot} over a mixed input space, which involves continuous and discrete variables. Let the set of kernel candidates $\mathcal{K}_{\mathrm{sub}}$ consists of $n_k = 4$ kernels, namely $k^1, \dots, k^4$. Then, the BO is used to minimize the discrete input variable j over the set $\{1, \dots, 4\}$ as stated in (4.5). Additionally, the input space contains continuous hyperparameters for each kernel function, as exemplary depicted in Fig. 4.2. Thus, the BO minimizes over a mixed input space which consists of one discrete variable j to select the kernel function and n_{Φ^j} continuous variables for the hyperparameters.

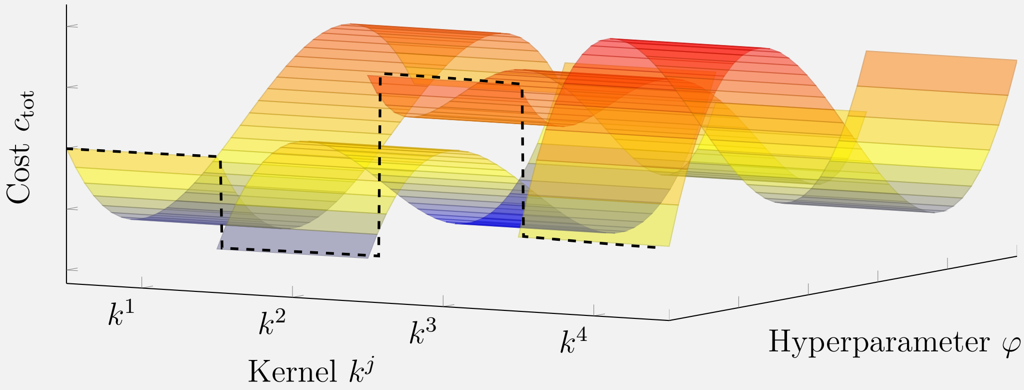


Figure 4.2: Bayesian optimization over mixed input space.

Theoretical Analysis

In this section, we show that, under some additional assumptions, the stability of the closed-loop is preserved during the task evaluation process and that BO converges to the minimum of the closed-loop cost function. Here, we focus on a class of stationary kernels which can be expressed by

$$k(\boldsymbol{z}, \boldsymbol{z}') = \varphi_1 k\Big((\boldsymbol{z} - \boldsymbol{z}')^\top P^{-1} (\boldsymbol{z} - \boldsymbol{z}')\Big),\ \boldsymbol{z}, \boldsymbol{z}' \in \mathbb{R}^{n_z} \tag{4.7}$$

with hyperparameters $\boldsymbol{\varphi} \in \mathbb{R}_{>0}^{n_\varphi = n_z + 1}$ and $P = \operatorname{diag}(\varphi_2, \dots, \varphi_{n_\varphi})$. This class of stationary kernels includes common kernels for kernel-based models [Bis06], for instance, the squared exponential kernel.

Assumption 4.1. *Let $\|\boldsymbol{f}_{ukn}\|_{k^*_{\varphi^*}}$ be finite. There exist a radius $r_y \in \mathbb{R}_{>0}$ and a time step $t_1 \in \mathbb{N}$ such that the output $\|\boldsymbol{y}_t\| \leq r_y$ of the system* (4.1) *for all $t > t_1$ with a control law* (4.2)*, based on the model $\boldsymbol{o}_{kbm}$ with stationary kernel k^* and hyperparameters $\boldsymbol{\varphi}^* \in \mathbb{R}^{n_\varphi}_{>0}$.*

The first part of the assumption, i.e., the bounded reproducing kernel Hilbert space norm, is a measure of smoothness of the function with respect to the kernel k with hyperparameters $\boldsymbol{\varphi}^* \in \mathbb{R}^{n_\varphi}_{>0}$. It is a common assumption for stabilizing controllers using kernel-based methods and is discussed in more detail in [Ber+16]. Controllers that satisfy this property for nonlinear, unknown systems are given, e.g., by [CHK12; Ber+16; BKH19]. The focus on stationary kernels is barely restrictive as many successfully applied kernels for nonlinear control are stationary.

Lemma 4.1. *With Assumption 4.1, there exists a non-empty set $\mathcal{K}_{sub}$ and a hyperparameter set $\Phi^1 \supset \{\boldsymbol{\varphi}^*\}$ such that $\forall k^j \in \mathcal{K}_{sub}$, for all $\boldsymbol{\varphi}^j \in \Phi^j$ the boundedness $\|\boldsymbol{y}_t\| \leq r_y$ of the system* (4.1) *for $t > t_1$ is preserved.*

This lemma guarantees that there exists a kernel set $\mathcal{K}_{\text{sub}}$ and a set Φ_{set} of hyperparameters that contains the stabilizing kernel k^* and the hyperparameter $\boldsymbol{\varphi}^*$ of Assumption 4.1. Thus, the proposed method can be applied to existing kernel-based control methods without loss of achieved guarantees. Before we start with the proof, the following lemma from [Bul11] is recalled.

Lemma 4.2 ([Bul11, Lemma 4])**.** *Let k be a stationary kernel as given by* (4.7)*. If $\boldsymbol{f}_{ukn} \in \mathcal{H}_{k_\varphi}$ then $\boldsymbol{f}_{ukn} \in \mathcal{H}_{k_{\varphi'}}$ holds for all $0 < \varphi'_i \leq \varphi_i, \forall i \in \{1, \dots, n_\varphi\}$, and*

$$\|\boldsymbol{f}_{ukn}\|^2_{k_{\varphi'}} \leq \left(\prod_{i=1}^{n_\varphi} \frac{\varphi_i}{\varphi'_i} \right) \|\boldsymbol{f}_{ukn}\|^2_{k_\varphi}.$$

Proof of Lemma 4.1. Assumption 4.1 inherently guarantees that at least one kernel $k^1 = k^*$ exists that preserves the boundedness of the system such that we define $\mathcal{K}_{\text{sub}} = \{k^1\}$. Since Assumption 4.1 ensures that $\|\boldsymbol{f}_{\text{ukn}}\|_{k^*_{\varphi^*}}$ is bounded and with Lemma 4.2, the mapping $\boldsymbol{f}_{\text{ukn}} \in \mathcal{H}_{k^1_{\varphi^*}}$ and, thus, also $\|\boldsymbol{f}_{\text{ukn}}\|_{k^1_{\underline{\varphi}}}$ is bounded for $\underline{\varphi}_i \in \mathbb{R}_{>0}, \forall i$, where $\underline{\varphi}_i < \varphi^*_i, \forall i$. For an upper bound, there exist $\overline{\varphi}_i \in \mathbb{R}_{>0}, \forall i$ such that $\varphi^*_i < \overline{\varphi}_i$ and $\boldsymbol{f}_{\text{ukn}} \in \mathcal{H}_{k^1_{\overline{\varphi}}}$, following Lemma 4.2. Thus, we define the set

$$\Phi^1 = \{\boldsymbol{\varphi} \colon \underline{\varphi}_i \leq \varphi_i \leq \overline{\varphi}_i, \forall i\} \tag{4.8}$$

as proper superset of $\boldsymbol{\varphi}^*$. Based on this set, the RKHS norm $\|\boldsymbol{f}_{\text{ukn}}\|_{k^1_{\varphi}}$ is finite for all $\boldsymbol{\varphi} \in \Phi^1$. As a result, the output $\|y_t\|$ is still bounded. □

Consequently, with Assumption 4.1, the stability of the control loop is preserved during the task evaluation. Furthermore, the minimum cost is not larger than the initial cost after BO as stated in the following.

Corollary 4.1. *Let $c_{tot,cl}$ be the minimum cost* (4.3) *after BO* (4.5) *with set of kernels $\mathcal{K}_{sub} = \{k^1 = k^*\}$ and hyperparameter set Φ^1 of* (4.8). *Let $c_{tot,ol}$ be the initial cost based on the control with kernel k^* and hyperparameter $\boldsymbol{\varphi}^*$. Then, $c_{tot,cl} \leq c_{tot,ol}$ holds.*

Proof. Since $c_{\mathrm{tot,cl}}$ is the minimum cost after BO that starts with the initial, data-based selected kernel k^* and hyperparameter $\boldsymbol{\varphi}^*$, it clearly follows that $c_{\mathrm{tot,cl}} \leq c_{\mathrm{tot,ol}}$ because of $k^* \in \mathcal{K}_{\mathrm{sub}}$ and $\boldsymbol{\varphi}^* \in \Phi^1$. □

We now show, that BO can converge to the global minimum of the cost function c_{tot} under specific conditions, starting with the following assumption.

Assumption 4.2. *Let k_{GP} be the kernel of the GP, that is used as prior $c_{tot} \sim \mathcal{GP}(0, k)$ for the Bayesian optimization* (4.5). *The RKHS norm of the cost function is bounded, i.e., $\|c_{tot}\|_k \leq r \in \mathbb{R}_{>0}$.*

Intuitively, Assumption 4.2 states that the kernel of the GP for BO is selected such that the GP can properly approximate the cost function. This sounds paradoxical since the cost function is unknown because of the unknown system behavior. However, there exist some kernels, so called universal kernels, which can approximate at least any continuous function arbitrarily precisely [SC08, Lemma 4.55].

Lemma 4.3 ([Sri+12]). *With Assumption 4.2, BO in* (4.5) *with upper confidence bound acquisition function [Sri+12, Eq.(6)] converges with high probability to the global minimum of c_{tot}.*

4.1.4 Numerical Evaluation

Next, we present a simple illustrative example that highlights our closed-loop model selection approach for kernel-based models. Consider the following one-dimensional system

$$\begin{aligned} x_{t+1} &= \exp(-\frac{1}{100}x_t^2)\sin(x_t) + \frac{1}{3}x_t + u_t \\ y_t &= x_t \end{aligned} \tag{4.9}$$

with state x_t and control u_t at time t. For this example, we assume that the system dynamics in (4.9) are unknown. We wish to avoid a high-gain control approach due to its unfavorable properties [Isi13], and use the proposed closed-loop model selection framework to optimize the control performance. As control law, a feedback linearization

$$u_t = -o_{\mathrm{kbm}}(\boldsymbol{u}_{0:t-1}, \boldsymbol{y}_{0:t}, k, \boldsymbol{\varphi}) + \frac{1}{2}x_t \tag{4.10}$$

is applied with the prediction of a support vector machine model $\boldsymbol{o}_{\mathrm{kbm}}$. The desired dynamics is assumed to be $x_{t+1,des} = 0.5x_{t,des}$, which would be achieved using the control law (4.10) with a perfect model of (4.9). The data set $\mathcal{D}$ consists of 11 homogeneously distributed training pairs $\{x_t^j, x_{t+1}^j\}_{j=1}^{11}$ of the system (4.9) in the interval $x_t \in [-10, 10]$ with $u_t = 0$. The linear, polynomial (cubic, p=3) and the squared exponential kernels are selected as possible kernel candidates, see Table 2.1 for details. The squared exponential kernel possesses two hyperparameters φ_1, φ_2, as defined in (2.47). In addition, the regression of the SVM depends on a hyperparameter φ_3 that defines the smoothness of the prediction and affects the number of support vectors, see [Kec01].

First, we evaluate a classical, data-based procedure which optimizes the kernel and the hyperparameters with respect to the cross-validation loss function [SC08] based on the training data only. Using BO, a minimum loss of 0.277 is found using the linear kernel with $\varphi_3 = 0.042$, Table 4.1. Using this linear model in the control loop with the nonlinear system (4.9) and control law (4.10) for $x_0 = 3$, the control error remains above zero, see Fig. 4.3. With the cost function $c_{\text{tot}} = \sum_{t=0}^{9} t^2(x_t - x_{des})^2$, the trajectory generates a cost of $c_{\text{tot}} = 1310.818$.

In comparison, the hyperparameters and the kernel are optimized with our proposed method. For this purpose, we evaluate the performance of the closed-loop system and use BO to compute the next promising kernel and hyperparameter combination. Figure 4.4 shows the mean and standard deviation of 20 repetitions over 50 trials each. The repetitions are run since BO exploration of the cost is also affected by randomness. The cost is reduced to a mean value of $c_{\text{tot}} = \overline{44.209}$ and the loss is $\overline{1.922}$. Figure 4.3 shows that the closed-loop optimization results in a reduced control error. Table 4.1 also presents the results for adding the collected data of all the 50 trials to the existing training data set to redefine the model (Data-based AT). Even with more training data, the data-based optimization favors the linear kernel.

Method	Selected kernel	$\varphi_1, \varphi_2, \varphi_3$	Loss	Cost
Data-based	Linear	$-, -, 0.042$	0.277	1310.818
Data-based AT	Linear	$-, -, 0.033$	0.274	1299.451
Closed-loop	Squared exponential	$\overline{1}, \overline{2.377}, \overline{0.027}$	$\overline{1.922}$	$\overline{44.209}$

Table 4.1: Comparison between data-based, data-based with additional training data and closed-loop optimization

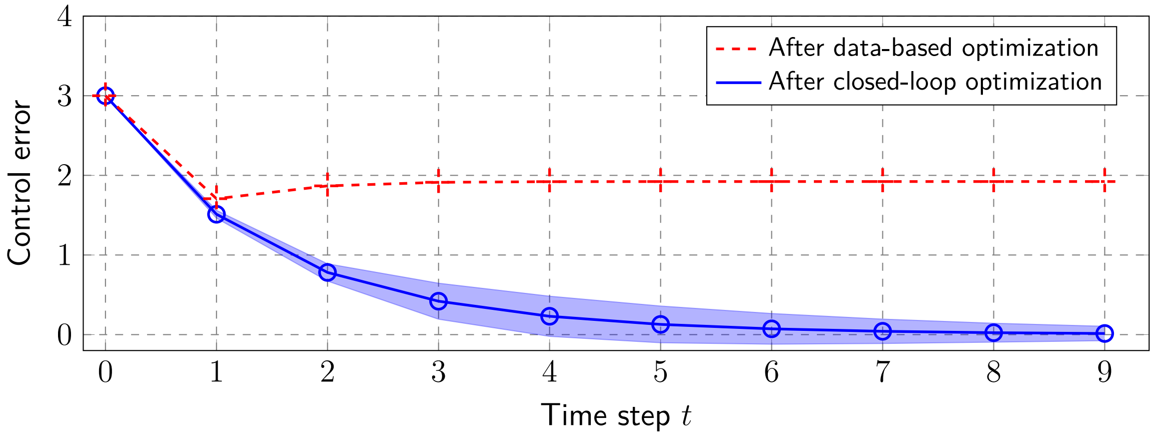


Figure 4.3: Control error using closed-loop model optimization for 20 repetitions with mean and 5-sigma deviation (blue) and data-based model selection (red).

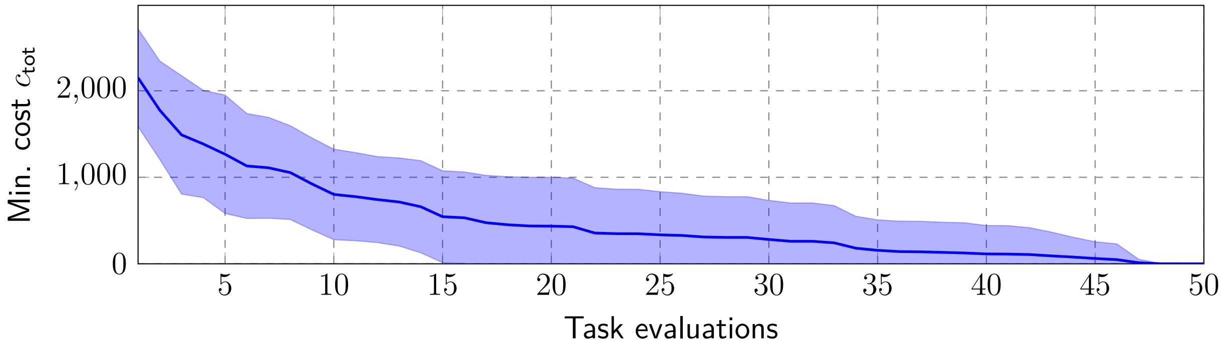


Figure 4.4: Minimum of the cost function over the number of trials for 20 repetitions for the closed-loop model selection algorithm.

Discussion

The example demonstrates that the optimization based on the training data only can lead to a reduced performance of the closed-loop system. Table 4.1 clearly shows that the data-based optimization results in a smaller loss with the linear kernel but generates a higher cost of the closed-loop system. In comparison, the closed-loop optimization finds a set of hyperparameters with the squared exponential kernel that significantly reduces the control error even if the loss of the model is higher. Thus, especially in the case of sparse data, the data-based optimization can misinterpret the data which can be avoided with the closed-loop model selection. For a fair comparison, we also collect the data, obtained during the 50 trials, to refine the kernel-based model. However, the additional training data only slightly improves the performance, but heavily increases the computational time of the kernel-based model due to the larger training data set.

An insight in one run of the BO is visualized in Fig. 4.5. The red line indicates the cost of the current task evaluation and the blue line represents the minimum cost over all past task evaluations. At the beginning of the closed-loop optimization, the Bayesian optimizer switches between different kernel functions. Then, the squared exponential kernel seems to be the most promising choice and, thus, the BO focuses on the exploration of its hyperparameters (task evaluation 3 to 14). Afterwards, the BO starts again to explore different kernel functions and hyperparameters. However, the selection in the 12th task evaluation already leads to the minimum cost.

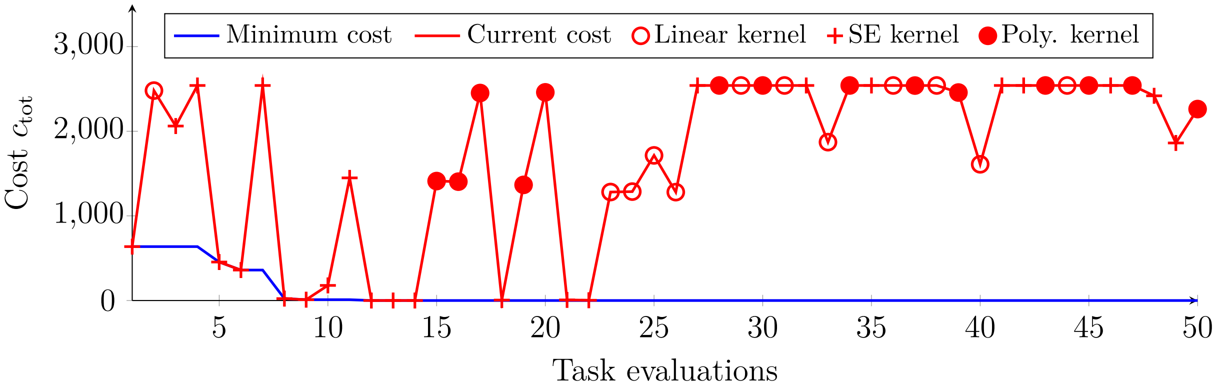


Figure 4.5: One run of the Bayesian optimization with a visualization of the selected kernel function. The configuration for the final minimum cost is obtained after 12 trials.

4.1.5 Robotic Experiment

In the following, an example with a 3-DOF robot demonstrates the applicability of the proposed approach to hardware testbeds. BO is used with the expected-improvement-plus as acquisition function because of its satisfactory performance in practical applications, see [Bul11], using a GP as the prior.

Setup

For the experimental evaluation, we use the 3-dof SCARA robot CARBO, as pictured in Fig. 4.6, with a spoon attached at the end effector. The links between the joints have a length of 0.3 m. The task goal is to follow a given trajectory as precisely as possible without using high feedback gains, at this might result in several practical disadvantages, see [NSP08]. Therefore, a precise model of the system dynamics is necessary. Since the modeling of the nonlinear fluid dynamics with a parametric model would be very time consuming, we use a computed torque control method based on a GP model. It allows high performance tracking control while also being able to guarantee the stability of the control loop, see [BKH19]. Underlying, a low level PD-controller enforces the generated torque by regulating the voltage based on a measurement of the current. The controller is implemented in MATLAB/Simulink on a Linux real-time system with a sample rate of 1 ms. For the implementation of the GP model, we use the GPML toolbox [RN10]. The desired trajectory follows a circular stirring movement through the fluid with a frequency of 0.5 hz.

Modeling

Here, we use a GP model $\boldsymbol{o}_{\text{kbm}}$ as kernel-based model technique based on 223 collected training points. The data is collected around the desired trajectory using a high gain controller. The location of the training points heavily influences the control performance. However, the proposed approach focuses on improving the performance based on the existing data. Each data pair consists of the position and velocity of all joints $[\boldsymbol{q}, \dot{\boldsymbol{q}}]^\top$ and the corresponding torque for the i-th joint. Since the GP produces one-dimensional outputs only, 3 GPs are used in total for the modeling of the robot's dynamics. Each GP $i = 1, \dots, 3$ uses a squared exponential kernel

$$k([\boldsymbol{q}, \dot{\boldsymbol{q}}]^\top, [\boldsymbol{q}', \dot{\boldsymbol{q}}']^\top) = \varphi_i^2 \exp\left(\frac{-\left\|[\boldsymbol{q}, \dot{\boldsymbol{q}}]^\top - [\boldsymbol{q}', \dot{\boldsymbol{q}}']^\top\right\|^2}{\varphi_{i+3}^2}\right), \varphi_i \in \mathbb{R}_{>0} \tag{4.11}$$

that can approximate any continuous function arbitrarily exact. With $\boldsymbol{\varphi} = [\varphi_1, \dots, \varphi_6]$ and the signal noise $\boldsymbol{\sigma}_n \in \mathbb{R}^3$, a total number of 9 parameters must be optimized. In contrast to the simulation, the kernel is fixed to reduce the optimization space and, therewith, the number of necessary task evaluations.

Control law

The control input $\boldsymbol{u}_{\text{ctrl}}$, i.e., the torque for all joints, is generated based on an estimated parametric model and the mean prediction $\boldsymbol{\mu}$ of the GP model as feed-forward component

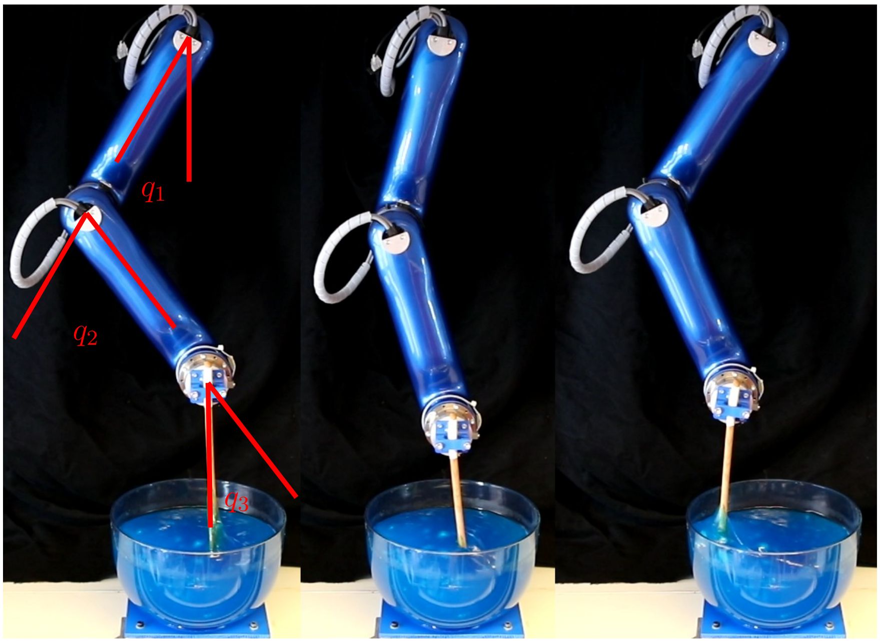


Figure 4.6: Three time steps of stirring with the 3-dof SCARA robot CARBO.

and a low gain PD-feedback part

$$\boldsymbol{u}_{\text{ctrl}} = \hat{H}\ddot{\boldsymbol{q}}_d + \hat{C}\dot{\boldsymbol{q}}_d + \hat{g} + \boldsymbol{\mu}(\dot{\boldsymbol{q}}, \boldsymbol{q}|\boldsymbol{o}_{\text{kbm}}) - K_d\dot{\boldsymbol{e}} - K_p\boldsymbol{e}. \tag{4.12}$$

Here, the desired trajectory is given by $\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d$ and $\ddot{\boldsymbol{q}}_d$ with the error $\dot{\boldsymbol{e}} = \dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}, \boldsymbol{e} = \boldsymbol{q}_d - \boldsymbol{q}$. The estimated inertia matrix $\hat{H} \in \mathbb{R}^{3\times 3}$, the estimated Coriolis matrix $\hat{C} \in \mathbb{R}^{3\times 3}$, and the estimated gravity vector $\hat{g} \in \mathbb{R}^3$ are calculated from the measured physical parameters of the robot. The feedback matrices are given by $K_p = \operatorname{diag}([60, 40, 10])$ and $K_d = \operatorname{diag}([1, 1, 0.4])$. The estimated parametric model is derived from a mathematical model where the parameters are physically measured. For the discretization of the control input, a zero-order method is used. For more details see [BKH19].

Evaluation

The evaluation of the performance of the closed-loop is based on the cost function

$$c_{\text{tot}} = \frac{1}{2000}\sum_{t=0}^{2000} \boldsymbol{e}(t_s t)^\top \boldsymbol{e}(t_s t) \tag{4.13}$$

with time step $t_s = 1\,\text{ms}$. Hence, the cost function is a measure for the tracking accuracy of the stirring movement. We consider as kernel candidate the squared exponential kernel, such that only the hyperparameters $\boldsymbol{\sigma}_n, \boldsymbol{\varphi}$ require optimization. Table 4.2 shows the comparison between the data-based and the closed-loop optimization. In the data-based case, the hyperparameters are optimized based on a gradient descent method to minimize the negative log likelihood function. In contrast, BO is used to minimize the tracking error in the closed-loop optimization. The initial values of the hyperparameters are set to the values obtained from the data-based optimization. The bounds are defined as 0.5 and 2 times of the initial values, respectively. The evolution of the minimum cost over the trials, where each trial is a single stirring movement, is shown in Fig. 4.7. The comparison of the joint

Value	Data-based	Closed-loop
$\boldsymbol{\sigma}_n$	$[0.10, 3\times10^{-3}, 6\times10^{-4}]$	$[0.20, 4\times10^{-3}, 3\times10^{-4}]$
$\varphi_{1,2,3}$	$[3.49, 1.42, 2.87]$	$[2.61, 1.68, 5.70]$
$\varphi_{4,5,6}$	$[1.21, 0.25, 0.27]$	$[0.80, 0.27, 0.29]$
Neg. log likelihood	$[89, -121, -176]$	$[115, -113, -136]$
Cost (Tracking error)	1.49	**1.05**

Table 4.2: Comparison between data-based and closed-loop optimization. The closed-loop optimization leads to a lower cost even though the neg. log likelihood is increased.

position error for the data-based and closed-loop optimization is shown in Fig. 4.8. After 100 trials, the tracking error is decreased by 30% through the optimization of the GP model only. Even though the resulting hyperparameters are sub-optimal with respect to the likelihood function, see Table 4.2, the performance of the closed-loop is significantly improved. In contrast to collecting more training data to improve the model, our proposed method does not increase the computational burden of the GP prediction, which is often critical in real-time applications. Since only the model is adapted, the properties of the closed-loop control architecture are also preserved.

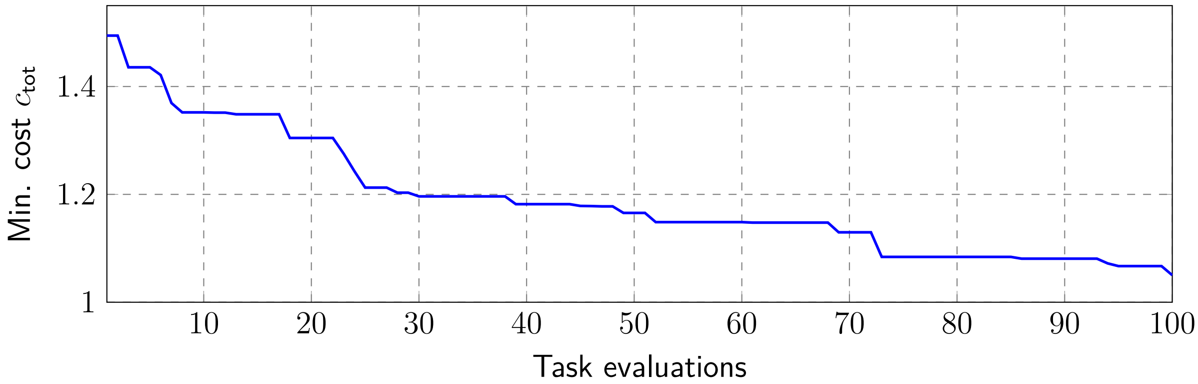


Figure 4.7: Minimum of the cost function over the number of trials. After 100 trials, the cost is significantly reduced.

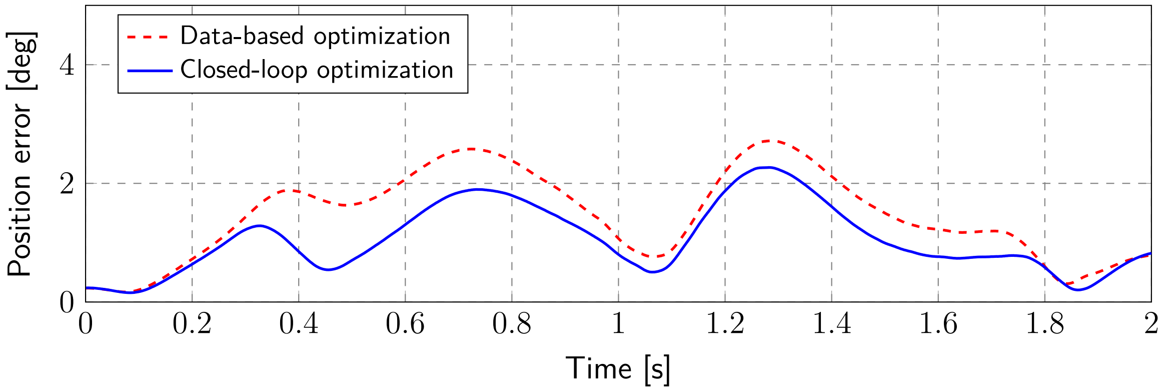


Figure 4.8: Comparison of the root square position error of all joints. Our proposed approach significantly reduces the position error.

4.2 Error of Misspecified Models

In Section 4.1, we show the selection of the kernel based on the performance of the control loop. As the variance of GP models is exploited as uncertainty measure, it is essential that the GP model fits the data generating process. Otherwise, using the variance as prediction error measure is not valid anymore, see Fig. 4.9. In this section, we derive an upper bound for the mean square prediction error (MSPE) between an estimated GP model and a GP model with unknown kernel functions and hyperparameters. For this purpose, a set of possible kernel functions with corresponding hyperparameter sets must be given. We exploit the property that many commonly used kernel functions are pseudo-concave with respect to their hyperparameters. As a consequence, the upper bound can be obtained as the solution of pseudo-concave optimization problems. Including additional assumptions, a closed-form solution is provided.

4.2.1 Problem Setting

We consider two GP models $\mathrm{GP}^1, \mathrm{GP}^2$ following (2.15), each trained with the same set $\mathcal{D} = \{X, Y\}$ of $n_\mathcal{D}$ data points. The output dimension of the data is denoted by $n_\mathrm{ydat} \in \mathbb{N}$. The model GP^1 is based on unknown kernel functions $k^1, \ldots, k^{n_\mathrm{ydat}}$ and hyperparameters $\boldsymbol{\varphi}^1, \ldots, \boldsymbol{\varphi}^{n_\mathrm{ydat}}$ whereas GP^2 uses the kernel functions $\hat{k}^1, \ldots, \hat{k}^{n_\mathrm{ydat}}$ and $\hat{\boldsymbol{\varphi}}^1, \ldots, \hat{\boldsymbol{\varphi}}^{n_\mathrm{ydat}}$, such that

$$\mathrm{GP}^1: \quad f_{\mathrm{GP}^1,i}|\boldsymbol{z}, \mathcal{D} \sim \mathcal{N}\left(\boldsymbol{k}^i(\boldsymbol{z}, X)^\top (K^i)^{-1} Y_{:,i}, k(\boldsymbol{z}, \boldsymbol{z}) - \boldsymbol{k}^i(\boldsymbol{z}, X)^\top (K^i)^{-1} \boldsymbol{k}^i(\boldsymbol{z}, X)\right) \tag{4.14}$$

$$\mathrm{GP}^2: \quad f_{\mathrm{GP}^2,i}|\boldsymbol{z}, \mathcal{D} \sim \mathcal{N}\left(\hat{\boldsymbol{k}}^i(\boldsymbol{z}, X)^\top (\hat{K}^i)^{-1} Y_{:,i}, \hat{k}(\boldsymbol{z}, \boldsymbol{z}) - \hat{\boldsymbol{k}}^i(\boldsymbol{z}, X)^\top (\hat{K}^i)^{-1} \hat{\boldsymbol{k}}^i(\boldsymbol{z}, X)\right) \tag{4.15}$$

for all $i = \{1, \ldots, n_\mathrm{ydat}\}$ and $\boldsymbol{z} \in \mathcal{Z} \subseteq \mathbb{R}^{n_z}$.

Remark 4.5. *For notational simplicity, we assume noise-free output data such that the Gram matrix $(K^i + \sigma_{n,i} I_{n_\mathcal{D}}) = K^i$ without loss of generality. All results hold for noisy output data as well by including the noise into the Gram matrix K^i. The prior mean functions are here assumed to be zero.*

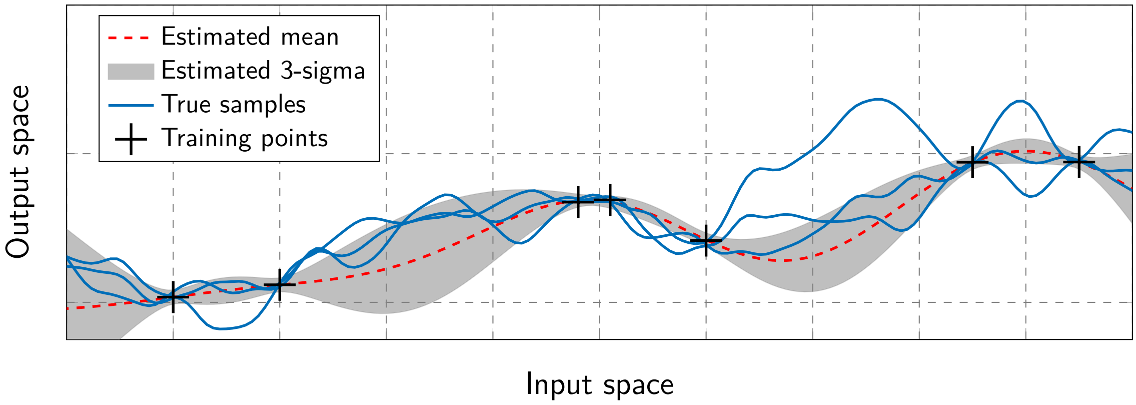


Figure 4.9: Using a misspecified GP model, the variance might be misleading in terms of the model confidence.

Our goal is to compute the MSPE between the prediction $(\boldsymbol{f}_{\mathrm{GP}^1}|\boldsymbol{z},\mathcal{D}) \in \mathbb{R}^{n_{y\mathrm{dat}}}$ of GP^1 and the mean prediction of $(\boldsymbol{f}_{\mathrm{GP}^2}|\boldsymbol{z},\mathcal{D}) \in \mathbb{R}^{n_{y\mathrm{dat}}}$ given by GP^2, i.e.,

$$\mathrm{E}\left[\left\|(\boldsymbol{f}_{\mathrm{GP}^1}|\boldsymbol{z},\mathcal{D}) - \boldsymbol{\mu}(\boldsymbol{f}_{\mathrm{GP}^2}|\boldsymbol{z},\mathcal{D})\right\|^2\right]. \tag{4.16}$$

Since the kernel functions of GP^1 are unknown, we derive an upper bound for the MSPE. The reason for using the predicted mean of GP^2 only is that we compare the MSPE with the predicted variance of GP^2 to show that the variance can be misleading. In accordance with the no-free-lunch theorem, it is not possible to give error bounds for the MSPE without any assumptions on $k^1,\dots,k^{n_{y\mathrm{dat}}}$. Thus, we assume to have knowledge about a set of kernel functions $\check{\mathcal{K}}$, which contains the kernels $k^1,\dots,k^{n_{y\mathrm{dat}}}$ of GP^1. In detail, let $\check{\mathcal{K}}$ be a set of $n_k \in \mathbb{N}$ kernel functions

$$\check{\mathcal{K}} = \{\check{k}^1,\dots,\check{k}^{n_k} \in \mathcal{K}\}, \tag{4.17}$$

which are positive and *pseudo-concave* with respect to their hyperparameters.

Remark 4.6. *A pseudo-concave function behaves like a concave function regarding the finding of local maxima, but need not actually be concave. In Section 4.2.4, we show that many common kernel functions are pseudo-concave and positive, such as the squared exponential, the rational quadratic and the polynomial for specific inputs.*

Remark 4.7. *To keep the set $\check{\mathcal{K}}$ as small as possible, statistical hypothesis testing can be used for discarding functions, which are unlikely.*

In addition, let $\check{\Phi}$ be a set of convex sets

$$\check{\Phi} = \left\{\check{\Phi}^1,\dots,\check{\Phi}^{n_k} | \check{\Phi}^j \subseteq \mathbb{R}^{n_{\Phi^j}}, n_{\Phi^j} \in \mathbb{N}, j \in \{1,\dots,n_k\}\right\}, \tag{4.18}$$

such that Φ^j is a valid hyperparameter set, see Definition 2.1, for the kernel $\check{k}^j$. Then, the following assumption must be held.

Assumption 4.3. *There exists an injective function $\varpi\colon \{1,\dots,n_{ydat}\} \to \{1,\dots,n_k\}$, such that $k^i = \check{k}^{\varpi(i)}, \boldsymbol{\varphi}^i \in \check{\Phi}^{\varpi(i)}$ for all $i \in \{1,\dots,n_{ydat}\}$.*

Interpreting this assumption, it is neither necessary to know the exact kernels $k^1,\dots,k^{n_{y\mathrm{dat}}}$ of GP^1, nor the exact corresponding hyperparameters $\boldsymbol{\varphi}^1,\dots,\boldsymbol{\varphi}^{n_{y\mathrm{dat}}}$. Instead, the kernel functions of GP^1 only need to be elements of a set of possible kernel functions given by $\check{\mathcal{K}}$. Analogously, each of the hyperparameters of GP^1 has to be in the set $\check{\Phi}^1,\dots,\check{\Phi}^{n_k}$ of possible hyperparameters. Thus, the function ϖ must exist but is unknown. The following example demonstrates a possible configuration for the sets $\check{\mathcal{K}}$ and $\check{\Phi}$.

Example 4.2. Consider GP^1 with a two-dimensional output. In the first dimension, the function follows a GP with polynomial kernel k^1 and hyperparameter $\boldsymbol{\varphi}^1 = 2$. In the second dimension, the function follows a GP with squared exponential kernel k^2 and hyperparameters $\boldsymbol{\varphi}^2 = [1.5, 0.5]^\top$. Then, valid sets $\check{\mathcal{K}}$ and $\check{\Phi}$ to satisfy Assumption 4.3 are provided by the following illustration.

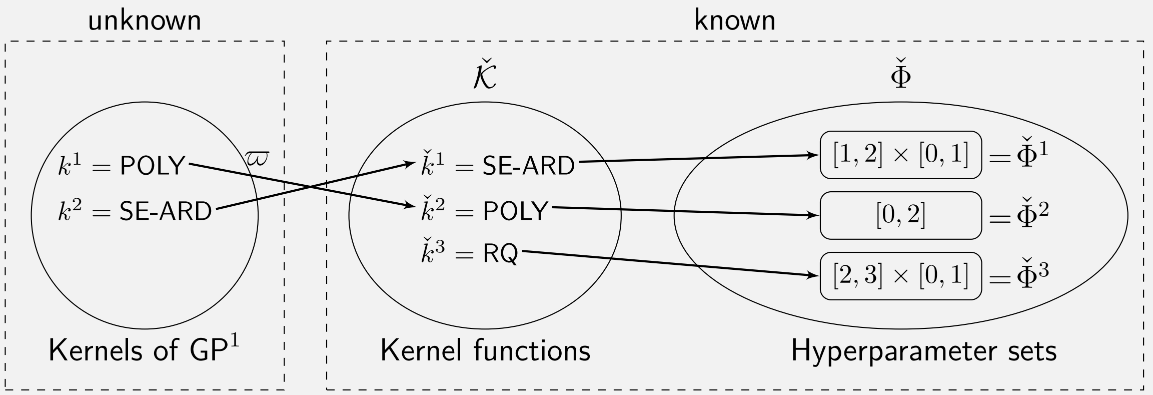


Figure 4.10: Possible configuration for set $\check{\mathcal{K}}$ of kernels and set $\check{\Phi}$ of hyperparameter sets.

Consequently, the function ϖ, which maps the unknown kernels of GP^1 to the kernels of the set $\check{\mathcal{K}}$, is defined by $\varpi(1) = 2, \varpi(2) = 1$. Normally, this function is unknown.

Application examples

Identification with GP-SSMs: The predicted variance correctly represents the model uncertainty if the reproducing kernel Hilbert space norm $\|f_i\|_{k^i}$ is bounded $\forall i \in \{1, \dots, n_x\}$, as presented in Section 2.1.6. Without knowing the exact kernel functions and hyperparameters, the predicted model uncertainty may not be correct. Our result allows to derive an upper bound for the MSPE between the correct but unknown GP-SSM and an estimated GP-SSM. Consequently, the upper bound also captures the error between the estimated GP-SSM and the original discrete-time system.
Reinforcement learning: Following [EMM05], a GP model is used for the value process $V \colon \mathbb{R}^{n_x} \to \mathbb{R}$ which connects values and rewards in a reinforcement learning scenario. It includes the assumption that the choice of the kernel function reflects the prior concerning the correlation between the values of states and rewards. Our Theorem 4.1 can be used to avoid an eventually underestimated MSPE based on the predicted variance with suboptimal hyperparameters. In this scenario, the set $\check{\mathcal{K}}$ contains the selected kernel function $\check{k}^1$ only. Thus, an upper bound for the MSPE can be computed without knowing the exact hyperparameters.

4.2.2 Mean Square Prediction Error

In this section, we introduce the computation procedure of the upper bound for the MSPE between GP^1 and the mean prediction of GP^2. The MSPE is defined by

$$\mathrm{E}\,\|\Delta\|^2 = \sum_{i=1}^{n_{y\mathrm{dat}}} k^i(\boldsymbol{\varphi}^i) - 2\hat{\boldsymbol{k}}^i(\hat{\boldsymbol{\varphi}}^i)^\top \hat{K}^{i^{-1}}(\hat{\boldsymbol{\varphi}}^i)\boldsymbol{k}^i(\boldsymbol{\varphi}^i) + \hat{\boldsymbol{k}}^i(\hat{\boldsymbol{\varphi}}^i)^\top \hat{K}^{i^{-1}}(\hat{\boldsymbol{\varphi}}^i)K^i(\boldsymbol{\varphi}^i)\hat{K}^{i^{-1}}(\hat{\boldsymbol{\varphi}}^i)\hat{\boldsymbol{k}}^i(\hat{\boldsymbol{\varphi}}^i) \tag{4.19}$$

with error $\Delta = (\boldsymbol{f}_{\mathrm{GP}^1}|\boldsymbol{z},\mathcal{D}) - \boldsymbol{\mu}(\boldsymbol{f}_{\mathrm{GP}^2}|\boldsymbol{z},\mathcal{D})$. For notational convenience, we do not write the arguments $\boldsymbol{z}$ and X but highlight the dependence on the hyperparameters $\boldsymbol{\varphi}$. Note, that the covariance vector $\boldsymbol{k}$ and the Gram matrix K are related to GP^1 and, thus, are unknown. In contrast, the covariance vector $\hat{\boldsymbol{k}}$ and the Gram matrix $\hat{K}$ are known as they belong to GP^2.

Remark 4.8. *If the estimated kernel functions and their hyperparameters are correct, i.e., let $k^i = \hat{k}^i, \boldsymbol{\varphi}^i = \hat{\boldsymbol{\varphi}}^i$ for all i, the mean square error simplifies to*

$$\mathrm{E}\left[\|\Delta\|^2\right] = \mathrm{tr}\left(\Sigma(\boldsymbol{f}_{GP^2}|\boldsymbol{z}, \mathcal{D})\right), \tag{4.20}$$

which is the trace of the posterior variance matrix.

It is obvious from (4.19), that the true kernel functions k^i of GP^1 are needed to compute the MSPE. Since the true kernel functions are unknown, we derive an upper bound based for the MSPE (4.19) on a set of kernel functions and hyperparameters. To determine this bound, the maximum of (4.19) has to be computed without knowing the kernel function k^i and the corresponding hyperparameters $\boldsymbol{\varphi}^i$. With Assumption 4.3, this problem is a non-convex, mixed-integer optimization problem. For simplicity in notation in the following derivations, parts of (4.19) are substituted to

$$\kappa_1^i(\boldsymbol{z}) = k^i(\boldsymbol{\varphi}^i) \tag{4.21}$$

$$\kappa_2^i(\boldsymbol{z}) = \hat{\boldsymbol{k}}^i(\hat{\boldsymbol{\varphi}}^i)^\top \hat{K}^{i^{-1}}(\hat{\boldsymbol{\varphi}}^i)\boldsymbol{k}^i(\boldsymbol{\varphi}^i) \tag{4.22}$$

$$\kappa_3^i(\boldsymbol{z}) = \hat{\boldsymbol{k}}^i(\hat{\boldsymbol{\varphi}}^i)^\top \hat{K}^{i^{-1}}(\hat{\boldsymbol{\varphi}}^i) K^i(\boldsymbol{\varphi}^i) \hat{K}^{i^{-1}}(\hat{\boldsymbol{\varphi}}^i)\hat{\boldsymbol{k}}^i(\hat{\boldsymbol{\varphi}}^i) \tag{4.23}$$

with $\kappa_1^i, \kappa_2^i, \kappa_3^i \colon \mathcal{Z} \to \mathbb{R}$. To achieve an upper bound for the MSPE (4.19), we compute individual bounds for the part terms given by (4.21) to (4.23) in the following. Note, that the second term κ_2^i is negative in (4.19) and, thus, we derive an lower bound for this term. We start with an upper bound for κ_1^i.

Lemma 4.4. *With Assumption 4.3, any k^i with $i \in \{1, \ldots, n_{ydat}\}$ is bounded by*

$$k^i(\boldsymbol{\varphi}^i, \boldsymbol{z}, \boldsymbol{z}') \leq \max_{j \in \{1, \ldots, n_k\}} \max_{\check{\boldsymbol{\varphi}}^j \in \check{\Phi}^j} \check{k}^j(\check{\boldsymbol{\varphi}}^j, \boldsymbol{z}, \boldsymbol{z}') \tag{4.24}$$

for all $\boldsymbol{\varphi}^i \in \check{\Phi}^{\varpi(i)}$ and $\forall \boldsymbol{z}, \boldsymbol{z}' \in \mathcal{Z}$.

Proof. Since k^i is an element of $\check{\mathcal{K}}$, the maximization over all kernel functions $\check{k}^j$ with their hyperparameter sets $\check{\Phi}^j$ is an upper bound for k^i. Hence, the optimization problem is split into an outer maximization over the finite number of kernel functions $\check{k}^j$ and an inner maximization over the convex hyperparameter sets. □

Then next lemma derives a *lower* bound for κ_2^i, since this term is negative in the MSPE (4.19).

Lemma 4.5. *With Assumption 4.3, there exists a lower bound $\underline{\kappa}_2^i(\boldsymbol{z}) \colon \mathcal{Z} \to \mathbb{R}$ for κ_2^i in (4.22) given by*

$$\underline{\kappa}_2^i(\boldsymbol{z}) = \sum_{l=1}^{n_\mathcal{D}} \min\left\{a_l^i, 0\right\} \max_{j \in \{1, \ldots, n_k\}} \max_{\check{\boldsymbol{\varphi}}^j \in \check{\Phi}^j} \check{k}^j(\check{\boldsymbol{\varphi}}^j, \boldsymbol{z}, X_{:,l}) \tag{4.25}$$

with $\boldsymbol{a}^i = \hat{\boldsymbol{k}}^i(\hat{\boldsymbol{\varphi}}^i)^\top \hat{K}^{i^{-1}}$, $\boldsymbol{a}^i \in \mathbb{R}^{n_\mathcal{D}}$

for all $\boldsymbol{z} \in \mathcal{Z}$ and $\forall i \in \{1, \ldots, n_{ydat}\}$.

Proof. We obtain a lower bound for (4.22) by

$$\kappa_2^i(\boldsymbol{z}) \geq \sum_{l=1}^{n_{\mathcal{D}}} \min\left\{a_l^i, 0\right\} \max \check{k}^j(\check{\boldsymbol{\varphi}}^j, \boldsymbol{z}, X_{:,l}) + \max\left\{a_l^i, 0\right\} \min \check{k}^j(\check{\boldsymbol{\varphi}}^j, \boldsymbol{z}, X_{:,l}) \tag{4.26}$$
$$\text{subject to } j \in \{1, \dots, n_k\}, \check{\boldsymbol{\varphi}}^j \in \check{\Phi}^j,$$

because the negative elements of $\boldsymbol{a}$ are multiplied with the maximum value of all kernel functions in $\check{\mathcal{K}}$ and vice versa. The minimum of $\check{k}^j(\check{\boldsymbol{\varphi}}^j, \boldsymbol{z}, X_{:,l})$ is always positive following Assumption 4.3, so that

$$\kappa_2^i(\boldsymbol{z}) \geq \sum_{l=1}^{n_{\mathcal{D}}} \min\left\{a_l^i, 0\right\} \max \check{k}^j(\check{\boldsymbol{\varphi}}^j, \boldsymbol{z}, X_{:,l}), \tag{4.27}$$

for $j \in \{1, \dots, n_k\}, \check{\boldsymbol{\varphi}}^j \in \check{\Phi}^j$ holds. With Lemma 4.4, we obtain the lower bound (4.25). □

Next, the following lemma presents an upper bound for the third and last term κ_3^i.

Lemma 4.6. *With Assumption 4.3, there exists an upper bound* $\bar{\kappa}_3^i(\boldsymbol{z})\colon \boldsymbol{z} \to \mathbb{R}$ *for* κ_3^i *in* (4.23) *given by*

$$\bar{\kappa}_3^i(\boldsymbol{z}) = \sum_{l,l'=1,\dots,m} \max\left\{a_l^i a_{l'}^i, 0\right\} \max_j \max_{\check{\boldsymbol{\varphi}}^j} \check{k}^j(\check{\boldsymbol{\varphi}}^j, X_{:,l'}, X_{:,l}) \tag{4.28}$$
$$\boldsymbol{a}^i = \hat{\boldsymbol{k}}^i(\hat{\boldsymbol{\varphi}}^i)^\top \hat{K}^{i^{-1}}, \; \boldsymbol{a}^i \in \mathbb{R}^{n_{\mathcal{D}}} \tag{4.29}$$

with $j \in \{1, \dots, n_k\}$ *and* $\check{\boldsymbol{\varphi}}^j \in \check{\Phi}^j, \forall \boldsymbol{z} \in \mathcal{Z}, \forall i \in \{1, \dots, n_{ydat}\}$*.*

Proof. It is analogous to the proof of Lemma 4.5. □

Given Lemmas 4.4 to 4.6, the next theorem introduce the upper bound for the MSPE (4.19) without the need of the true kernel functions k^i and hyperparameters $\boldsymbol{\varphi}^i$ of GP^1.

Theorem 4.1. *Consider the MSPE* (4.19) *between the prediction* $\boldsymbol{f}_{GP^1}|\boldsymbol{z}, \mathcal{D}$ *of GP*^{1} *and the posterior mean of* $\boldsymbol{f}_{GP^2}|\boldsymbol{z}, \mathcal{D}$ *of GP*^{2}*. With Assumption 4.3, there exists an upper bound for the MSPE given by*

$$\mathrm{E}\left[\|(\boldsymbol{f}_{GP^1}|\boldsymbol{z}, \mathcal{D}) - \boldsymbol{\mu}(\boldsymbol{f}_{GP^2}|\boldsymbol{z}, \mathcal{D})\|^2\right] \leq n_{ydat}\bar{\kappa}_1(\boldsymbol{z}) + \sum_{i=1}^{n_{ydat}} \bar{\kappa}_3^i(\boldsymbol{z}) - 2\underline{\kappa}_2^i(\boldsymbol{z}) \tag{4.30}$$
$$\bar{\kappa}_1(\boldsymbol{z}) = \max_{j \in \{1,\dots,n_k\}} \max_{\check{\boldsymbol{\varphi}}^j \in \check{\Phi}^j} \check{k}^j(\check{\boldsymbol{\varphi}}^j, \boldsymbol{z}, \boldsymbol{z}) \tag{4.31}$$

with $\underline{\kappa}_2^i, \bar{\kappa}_3^i$ *given by* (4.25) *and* (4.28)*, respectively.*

Proof. The mean square error is upper bounded by a sum of upper bounds for each term of (4.19). An upper bound of (4.21) with Assumption 4.3 can be computed by (4.31) following Lemma 4.4. The bound $\bar{\kappa}_1$ is independent of the training data $\mathcal{D}$ and thus, independent of i, so that it sums up to $n_{y\mathrm{dat}}\bar{\kappa}_1$. With Lemmas 4.5 and 4.6, the second and third terms are bounded. Finally, all three bounds together lead to the upper bound (4.31) for the MSPE. □

Remark 4.9. *The minimum in* (4.26) *for the lower bound of* κ_2^i *is set to zero because the numerical computation of* (4.26) *would be hard to obtain since* $\check{k}$ *is pseudo-concave. With this modification, the solution of (4.30) can be computed by standard gradient algorithms [HP90].*

4.2.3 Closed-form Solution

So far, we introduce an upper bound for the MSPE, given by (4.30), which can be computed numerically with an optimization algorithm. Admitting additional assumptions, a closed-form solution for (4.30) can be obtained. We start with the introduction of the assumptions.

Assumption 4.4. *Each convex set of hyperparameters* $\check{\Phi}^j \in \check{\Phi}$ *in* (4.18) *can be described by two vectors* $\underline{\boldsymbol{\varphi}}^j, \bar{\boldsymbol{\varphi}}^j \in \mathbb{R}^{n_{\Phi^j}}$

$$\check{\Phi}^j = \left\{\check{\boldsymbol{\varphi}}^j \in \mathbb{R}^{n_{\Phi^j}} | \underline{\boldsymbol{\varphi}}^j \preceq \check{\boldsymbol{\varphi}}^j \preceq \bar{\boldsymbol{\varphi}}^j\right\}, \forall j \in \{1, \dots, n_k\}. \tag{4.32}$$

Assumption 4.5. *Each kernel function* $\check{k}^j, j \in \{1, \dots, n_k\}$ *in* (4.17) *is componentwise strictly increasing with respect to its hyperparameters* $\check{\boldsymbol{\varphi}}^j$*. In detail, for all* $\check{\varphi}_i^j, \check{v}_i^j \in \check{\Phi}^j$ *with* $\check{\varphi}_i^j < \check{v}_i^j$*, one has* $\check{k}^j(\check{\boldsymbol{\varphi}}^j, \boldsymbol{z}, \boldsymbol{z}') < \check{k}^j(\check{\boldsymbol{v}}^j, \boldsymbol{z}, \boldsymbol{z}')\ \forall \boldsymbol{z}, \boldsymbol{z}' \in \mathcal{Z}$ *and for all* $i \in \{1, \dots, n_{\Phi^j}\}$.

Assumption 4.4 requires that each of the convex hyperparameter sets $\Phi^j \subseteq \mathbb{R}^{n_{\Phi^j}}$ is a n_{Φ^j}-dimensional hyperrectangle which is a weak restriction in practice. In Section 4.2.4, we show that Assumption 4.5 holds for commonly used kernel functions and, thus, it is not restrictive. Based on these assumptions, there exists a closed-form solution of Theorem 4.1 because the maximum of the kernel function $\check{k}^j$ is now always at $\bar{\boldsymbol{\varphi}}^j$. Before we present the closed-form solution, the implications of Assumptions 4.4 and 4.5 are visualized in the following.

Example 4.3. We consider a squared exponential kernel with its two hyperparameters and a hyperrectangle $\{\boldsymbol{\varphi} \in \mathbb{R}^2 | 1 \leq \varphi_1 \leq 3, 1 \leq \varphi_2 \leq 4\}$. Figure 4.11 shows the value of the kernel with respect to its hyperparameters for fixed $\boldsymbol{z}, \boldsymbol{z}$. As the squared exponential kernel is componentwise strictly increasing, its value is increasing with increasing hyperparameters. Thus, the maximum, subject to the hyperrectangle, is at the upper right corner $\bar{\boldsymbol{\varphi}} = [3; 4]$.

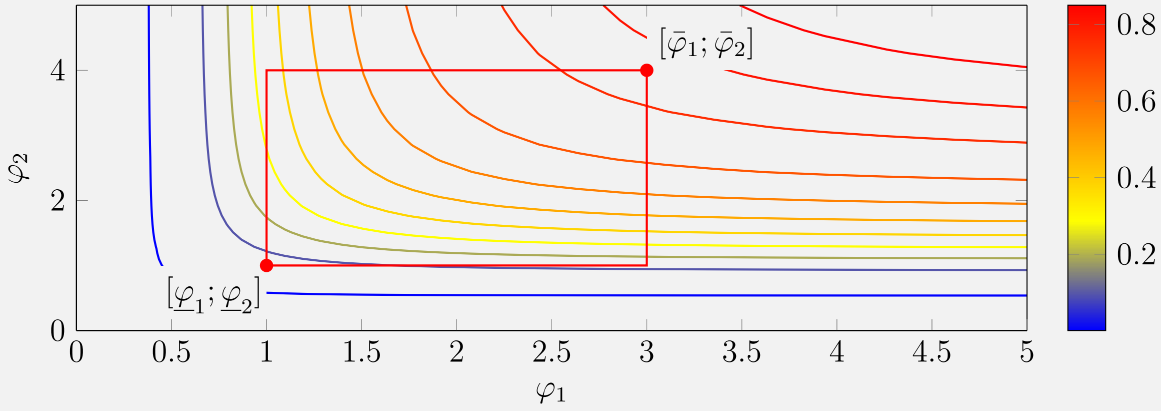


Figure 4.11: The value (color-coded, red high and blue low) of the squared exponential kernel $k(\boldsymbol{\varphi}, \boldsymbol{z}, \boldsymbol{z})$ over its hyperparameters $\boldsymbol{\varphi}$. Inside the hyperrectangle (black box), the maximum of the kernel is at its upper right corner.

Theorem 4.2. *Consider the MSPE between the prediction $\boldsymbol{f}_{GP^1}|\boldsymbol{z},\mathcal{D}$ of GP^1 and the posterior mean of $\boldsymbol{f}_{GP^2}|\boldsymbol{z},\mathcal{D}$ of GP^2. With Assumptions 4.3 to 4.5, there exists an upper bound for the MSPE given by*

$$\begin{aligned}\mathrm{E}\left\|\Delta\right\|^2 \leq \sum_{i=1}^{n_{ydat}} \max_j \Big\{ & \check{k}^j(\bar{\boldsymbol{\varphi}}^j, \boldsymbol{z}, \boldsymbol{z}) && (4.33)\\ & -2\sum_{l=1}^{n_{\mathcal{D}}} \min\left\{a_l^i, 0\right\} \check{k}^j(\bar{\boldsymbol{\varphi}}^j, \boldsymbol{z}, X_{:,l}) + \max\left\{a_l^i, 0\right\} \check{k}^j(\underline{\boldsymbol{\varphi}}^j, \boldsymbol{z}, X_{:,l}) \\ & +\sum_{l,l'=1,\dots,n_{\mathcal{D}}} \max\left\{a_l^i a_{l'}^i, 0\right\} \check{k}^j(\bar{\boldsymbol{\varphi}}^j, X_{:,l'}, X_{:,l}) + \min\left\{a_l^i a_{l'}^i, 0\right\} \check{k}^j(\underline{\boldsymbol{\varphi}}^j, X_{:,l'}, X_{:,l}) \Big\} && (4.34)\end{aligned}$$

with $\Delta = (\boldsymbol{f}_{GP^1}|\boldsymbol{z},\mathcal{D}) - \boldsymbol{\mu}(\boldsymbol{f}_{GP^2}|\boldsymbol{z},\mathcal{D})$ and $\boldsymbol{a}^i = \hat{\boldsymbol{k}}^i(\hat{\boldsymbol{\varphi}}^i)^\top \hat{K}^{i^{-1}}$.

Remark 4.10. *The solution of* (4.34) *is a closed-form expression in the sense that it can be evaluated in a finite number of operations because the maximization is over a finite set.*

Proof. Without loss of generality, we choose $j \in \{1,\dots,n_k\}$ of each maximization such that $\check{k}^j$ in (4.34) is equal to the kernel function k^i. With Assumption 4.5, the kernel function k^i with the hyperparameters $\boldsymbol{\varphi}^i$ is always equal to or less than the same kernel function with $\bar{\boldsymbol{\varphi}}^i$ and vice versa, i.e., $k^i(\boldsymbol{\varphi}^i, \boldsymbol{z}, \boldsymbol{z}') \geq k^i(\underline{\boldsymbol{\varphi}}^i, \boldsymbol{z}, \boldsymbol{z}')$. Thus, it remains to be proven that

$$\begin{aligned}\mathrm{E}\left\|\Delta\right\|^2 \leq \sum_{i=1}^{n_{y\mathrm{dat}}} \Big\{ & k^i(\bar{\boldsymbol{\varphi}}^i, \boldsymbol{z}, \boldsymbol{z}) + \sum_{p,q=1,\dots,n_{\mathcal{D}}} \max\left\{a_l^i a_{l'}^i, 0\right\} k^i(\bar{\boldsymbol{\varphi}}^i, X_{:,l'}, X_{:,l}) \\ & + \min\left\{a_l^i a_{l'}^i, 0\right\} k^i(\underline{\boldsymbol{\varphi}}^i, X_{:,l'}, X_{:,l}) - 2\sum_{l=1}^{n_{\mathcal{D}}} \min\left\{a_l^i, 0\right\} k^i(\bar{\boldsymbol{\varphi}}^i, \boldsymbol{z}, X_{:,l}) \\ & + \max\left\{a_l^i, 0\right\} k^i(\underline{\boldsymbol{\varphi}}^i, \boldsymbol{z}, X_{:,l}) \Big\}. \end{aligned} \tag{4.35}$$

Each term of (4.35) upper bounds the corresponding term of $\mathrm{E}\left\|\Delta\right\|^2$ in (4.19) analogously to the idea in the proof of Lemma 4.5. Since (4.34) maximizes over all $\check{k}^j$ and, considering Assumption 4.3, the kernel function k^i is element of $\check{\mathcal{K}}$, there exists a j such that the assumption at the beginning of the proof is fulfilled. As a consequence, the MSPE is upper bounded by (4.34). □

If the set of possible kernels $\check{\mathcal{K}}$ and hyperparameter sets $\check{\Phi}$ contain only the necessary elements to satisfy Assumption 4.3, the following special case occurs.

Corollary 4.2. *Let $k^1 = \dots = k^{n_{ydat}}$ and $\boldsymbol{\varphi}^1 = \ldots = \boldsymbol{\varphi}^{n_{ydat}}$ be the set of possible kernels and hyperparameters. Then, the closed-form solution* (4.34) *of Theorem 4.2 is equivalent to the posterior variance* (4.20) *for $\check{\mathcal{K}} = \{k^1\}$ and $\check{\Phi} = \{\boldsymbol{\varphi}^1\}$*

Proof. This is a direct result of the error bound (4.35) if the set $\check{\mathcal{K}}$ only contains the kernel functions $k^1 = \dots = k^{n_{y\mathrm{dat}}}$ and the set $\check{\Phi}$ only contains the corresponding hyperparameters $\boldsymbol{\varphi}^1 = \ldots = \boldsymbol{\varphi}^{n_{y\mathrm{dat}}}$. □

Remark 4.11. *Corollary 4.2 shows the convergence of the upper bound, given by* (4.34)*, to the true MSPE* (4.16) *between GP^1 and GP^2 for the minimum-size sets $\check{\mathcal{K}}, \check{\Phi}$.*

4.2.4 Pseudo-concave Kernel Functions

To upper bound the MSPE, the kernels in the set $\check{K}$ are assumed to be pseudo-concave. In addition, the closed-form solution requires that the kernels are componentwise monotonically increasing, see Assumption 4.5. Therefore, we show in the following that many common kernel functions fulfill these properties. Before we start, the definition of pseudo-concave functions is recalled.

Definition 4.1 ([Man75])**.** A function $\boldsymbol{f}_{\text{gen}} \in \mathcal{C}^1$, where $\boldsymbol{f}_{\text{gen}} \colon \mathcal{S} \to \mathbb{R}$, defined on a convex set $\mathcal{S} \subseteq \mathbb{R}^{n_{\mathcal{S}}}, n_{\mathcal{S}} \in \mathbb{N}$ is pseudo-concave if for all $\boldsymbol{z}, \boldsymbol{z}' \in S$

$$\nabla \boldsymbol{f}_{\text{gen}}(\boldsymbol{z})(\boldsymbol{z}' - \boldsymbol{z}) \leq 0 \Rightarrow \boldsymbol{f}_{\text{gen}}(\boldsymbol{z}') \leq \boldsymbol{f}_{\text{gen}}(\boldsymbol{z}). \tag{4.36}$$

Pseudo-concavity is of interest because a point is a local maximum of a pseudo-concave function if and only if the gradient is zero at this point. Thus, many optimization concepts which are known from concave functions can be transferred to pseudo-concave function. The next proposition states that many commonly kernel functions are pseudo-concave and componentwise monotonically increasing.

Proposition 4.1. *The following kernel functions are pseudo-concave and componentwise monotonically increasing with respect to their hyperparameters on the designated domain.*

Kernel function	*Parameters*	*Domain*
Polynomial (2.45)	$p \in \mathbb{N}, \varphi_1 \in \mathbb{R}_{\geq 0}$	$\forall \boldsymbol{z}, \boldsymbol{z}' \in \mathbb{R}^{n_z}_{\geq 0}$
Rational quadratic (2.48)	$p \in \mathbb{N}_{>0}, \boldsymbol{\varphi} \in \mathbb{R}^2_{>0}$	$\forall \boldsymbol{z}, \boldsymbol{z}' \in \mathbb{R}^{n_z}$
Squared exponential ARD (2.49)	$\boldsymbol{\varphi} \in \mathbb{R}^{n_z+1}_{>0}$	$\forall \boldsymbol{z}, \boldsymbol{z}' \in \mathbb{R}^{n_z}$
Matérn (2.46)	$p \in \{0, 1, 2\}, \boldsymbol{\varphi} \in \mathbb{R}^2_{>0}$	$\forall \boldsymbol{z}, \boldsymbol{z}' \in \mathbb{R}^{n_z}$

Proof. The following proof considers each kernel function separately.

Polynomial kernel

The polynomial kernel function (2.45) is strictly increasing on $\varphi \in \mathbb{R}_{\geq 0}$ for any $\boldsymbol{z}, \boldsymbol{z}' \in \mathbb{R}^{n_z}_{\geq 0}$ and hence, pseudo-concave, see [BSS13], and componentwise monotonically increasing.

Rational quadratic kernel

Following [BSS13], a continuous differentiable kernel function k is pseudo-concave if the kernel is quasi-concave and $\partial k / \partial \boldsymbol{\varphi} \neq \boldsymbol{0}$. The kernel function is quasi-concave if $\det Q_3(\boldsymbol{\varphi}) > 0$ and $\det Q_2(\boldsymbol{\varphi}) < 0$, where the matrix Q_r is the r-th order leading principal submatrix of the bordered Hessian Q of k with respect to $\boldsymbol{\varphi}$ given by

$$Q = \begin{bmatrix} 0 & \frac{\partial k}{\partial \varphi_1} & \frac{\partial k}{\partial \varphi_2} \\ \frac{\partial k}{\partial \varphi_1} & \frac{\partial^2 k}{\partial \varphi_1^2} & \frac{\partial^2 k}{\partial \varphi_1 \varphi_2} \\ \frac{\partial k}{\partial \varphi_2} & \frac{\partial^2 k}{\partial \varphi_2 \varphi_1} & \frac{\partial^2 k}{\partial \varphi_2^2} \end{bmatrix}. \tag{4.37}$$

For the rational quadratic kernel, the principal submatrices are

$$Q_2 = \frac{-4\|\boldsymbol{z}-\boldsymbol{z}'\|^4 p^2 \varphi_1^4}{\varphi_2^2 \left(2p\varphi_2^2 + \|\boldsymbol{z}-\boldsymbol{z}'\|^2\right)^2 \left(\frac{2p\varphi_2^2+\|\boldsymbol{z}-\boldsymbol{z}'\|^2}{2p\varphi_2^2}\right)^{2p}} < 0 \tag{4.38}$$

$$Q_3 = \frac{8\|\boldsymbol{z}-\boldsymbol{z}'\|^2 \varphi_1^4 p(p\|\boldsymbol{z}-\boldsymbol{z}'\|^2 + \|\boldsymbol{z}-\boldsymbol{z}'\|^2 + 6p\varphi_2^2)}{\varphi_2^2 \left(2p\varphi_2^2 + \|\boldsymbol{z}-\boldsymbol{z}'\|^2\right)^2 \left(\frac{2p\varphi_2^2+\|\boldsymbol{z}-\boldsymbol{z}'\|^2}{2p\varphi_2^2}\right)^{3p}} > 0, \tag{4.39}$$

for all $p \in \mathbb{N}_{>0}, \boldsymbol{\varphi} \in \mathbb{R}^2_{>0}$. Thus, the rational quadratic kernel is quasi-concave. Since the tational quadratic kernel is continuous differentiable and $\partial k/\partial \boldsymbol{\varphi} \neq \mathbf{0}$ on its domain, the kernel is also pseudo-concave. It is obvious that the kernel is also componentwise monotonically increasing.

Squared exponential kernel

The kernel function can be rewritten as

$$k(\boldsymbol{\varphi}, \boldsymbol{z}, \boldsymbol{z}') = \exp\left(\log(\varphi_1^2) + \sum_{i=1}^{n_z} -\frac{|z_i - z_i'|^2}{2\varphi_{i+1}^2}\right). \tag{4.40}$$

In the form (4.40), the argument of the exponential function is quasi-concave, since this sum of concave functions is concave on all $\boldsymbol{\varphi} \in \mathbb{R}^{n_x+1}_{>0}$ for any $\boldsymbol{z}, \boldsymbol{z}' \in \mathbb{R}^{n_x}$. The composition with the strictly increasing exponential function results in an overall quasi-concave function [Sun96, Theorem 8.5]. Since the kernel is continuous differentiable and $\partial k/\partial \boldsymbol{\varphi} \neq \mathbf{0}$ holds for $\boldsymbol{\varphi} \in \mathbb{R}^{n_z+1}_{>0}$, the kernel (4.40) is also pseudo-concave. Further, the kernel function is componentwise monotonically increasing as the exponential and the logarithm function are monotonically increasing.

Matérn kernel

Analogously to the proof for rational quadratic kernel, the principal submatrices of the bordered Hessian of k are $\det Q_2 < 0$ and $\det Q_3 > 0$. With continuous differentiable k, and $\partial k/\partial \boldsymbol{\varphi} \neq \mathbf{0}$ on its domain, the Matérn kernel is pseudo-concave. The kernel function is also componentwise monotonically increasing as the exponential function grows faster than any polynomial. □

4.2.5 Numerical Evaluation

In this section, we present a numerical example for the MSPE upper bound, given by Theorem 4.2, by means of a stochastic GP-SSMs, see (3.70). For this purpose, we assume that a discrete-time, one-dimensional system is described by GP^1 with a Matérn kernel where $p = 1$ and the hyperparameters $\boldsymbol{\varphi} = [5.2, 1.6]^\top$. The training set $\mathcal{D} = \{X, Y\}$ contains 10 uniformly distributed measurements, see Table 4.3.

Since the correct kernel function is usually unknown in real-world applications, the squared exponential kernel is a common choice. For demonstration purposes, we follow this standard approach and use GP^2 with a squared exponential kernel trained on the measurements of the system. The hyperparameters are optimized according to the log likelihood function

X	-9.6	-6.8	-4.8	-2.7	-1.0	1.3	3.4	6.4	8.4	10.7
Y	-1.9	1.2	0.2	-0.5	0.1	0.1	-0.7	1.3	-0.9	-0.6

Table 4.3: Training data set $\mathcal{D}$ of an unknown dynamics.

with a conjugate gradient method, which results in the hyperparameters $\hat{\boldsymbol{\varphi}} = [0.36, 0.32]^\top$. In Fig. 4.12, the estimated mean $\boldsymbol{\mu}(x_{t+1}|x_t, \mathcal{D})$ of GP^2 together with the mean and variance of the true generating process GP^1 are visualized. Obviously, the estimated mean (red dashed) of GP^2 does not appropriately describe the true process (blue and gray), although it represents the training data (black) accurately. As a consequence, the mean square error between the estimated mean and the correct model is significantly underestimated in the state space, as well as in the time domain, as presented in Fig. 4.13. To overcome this issue, we employ Theorem 4.2 to determine an upper bound of the MSPE without exact knowledge of the correct kernel function. More in detail, we consider a set of kernel functions with their corresponding hyperparameter sets, as presented in Table 4.4. For the comparison of different sets of hyperparameters, we use three different interval sizes around the true hyperparameters. Figure 4.13 shows the estimated (red dashed) and true mean square prediction error (blue point-dashed). Note that the true MSPE is normally unknown.The error obtained from GP^2 obviously underestimates the true MSPE. In contrast, we compute upper bounds (green solid) using Theorem 4.2 based on the set of kernels and three different hyperparameter set, given by Table 4.4. With a smaller range for the hyperparameters $\check{\boldsymbol{\varphi}}^1$, i.e., a better estimation of the unknown hyperparameters, the bound becomes tighter. In summary, our upper bound successfully confines the true MSPE without knowing the exact kernel and hyperparameters of GP^1.

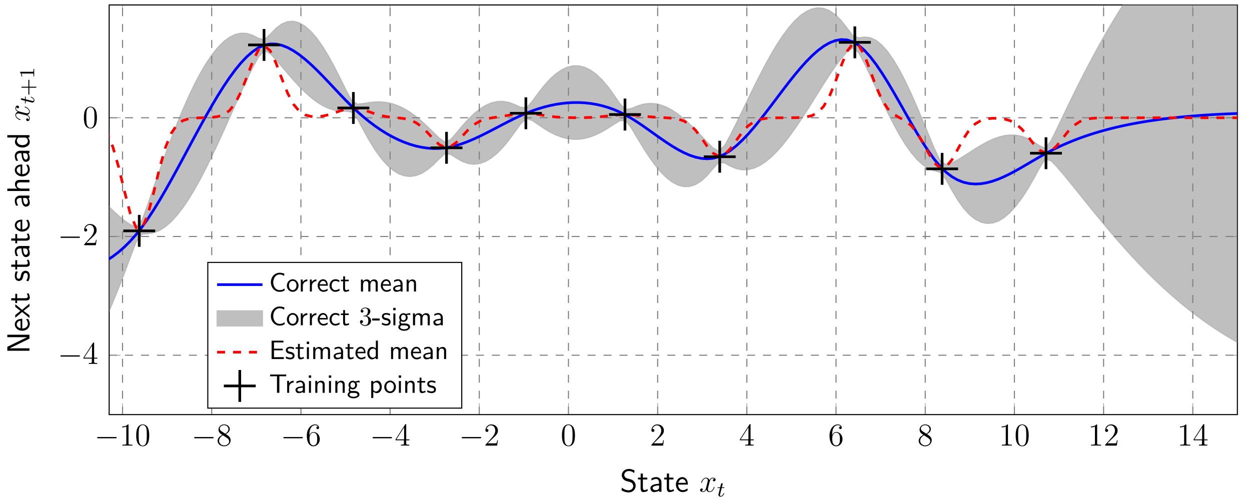


Figure 4.12: Based on the training data, the estimated mean generates a misleading impression of the underlying process.

Kernel functions	Hyperparameter sets	
$\check{k}^1$: Matérn $p = 1$	$\{0.9\boldsymbol{\varphi}^1 \preceq \check{\boldsymbol{\varphi}}^1 \preceq 1.1\boldsymbol{\varphi}^1\}$	10% error interval
	$\{[0;0] \prec \check{\boldsymbol{\varphi}}^1 \preceq 2\boldsymbol{\varphi}^1\}$	100% error interval
	$\{[0;0] \prec \check{\boldsymbol{\varphi}}^1 \preceq 3\boldsymbol{\varphi}^1\}$	200% error interval
$\check{k}^2$: Matérn $p = 0, 2$	$\{[1;1.5] \preceq \check{\boldsymbol{\varphi}}^2 \preceq [10;2]\}$	
$\check{k}^3$: Rational quadratic $p = 1$	$\{[1;0.1] \preceq \check{\boldsymbol{\varphi}}^3 \preceq [20;1]\}$	
$\check{k}^4$: Squared exponential	$\{[0.1;0.01] \preceq \check{\boldsymbol{\varphi}}^4 \preceq [10;1]\}$	

Table 4.4: Kernel functions in $\check{\mathcal{K}}$ with hyperparameter sets in $\check{\Phi}$.

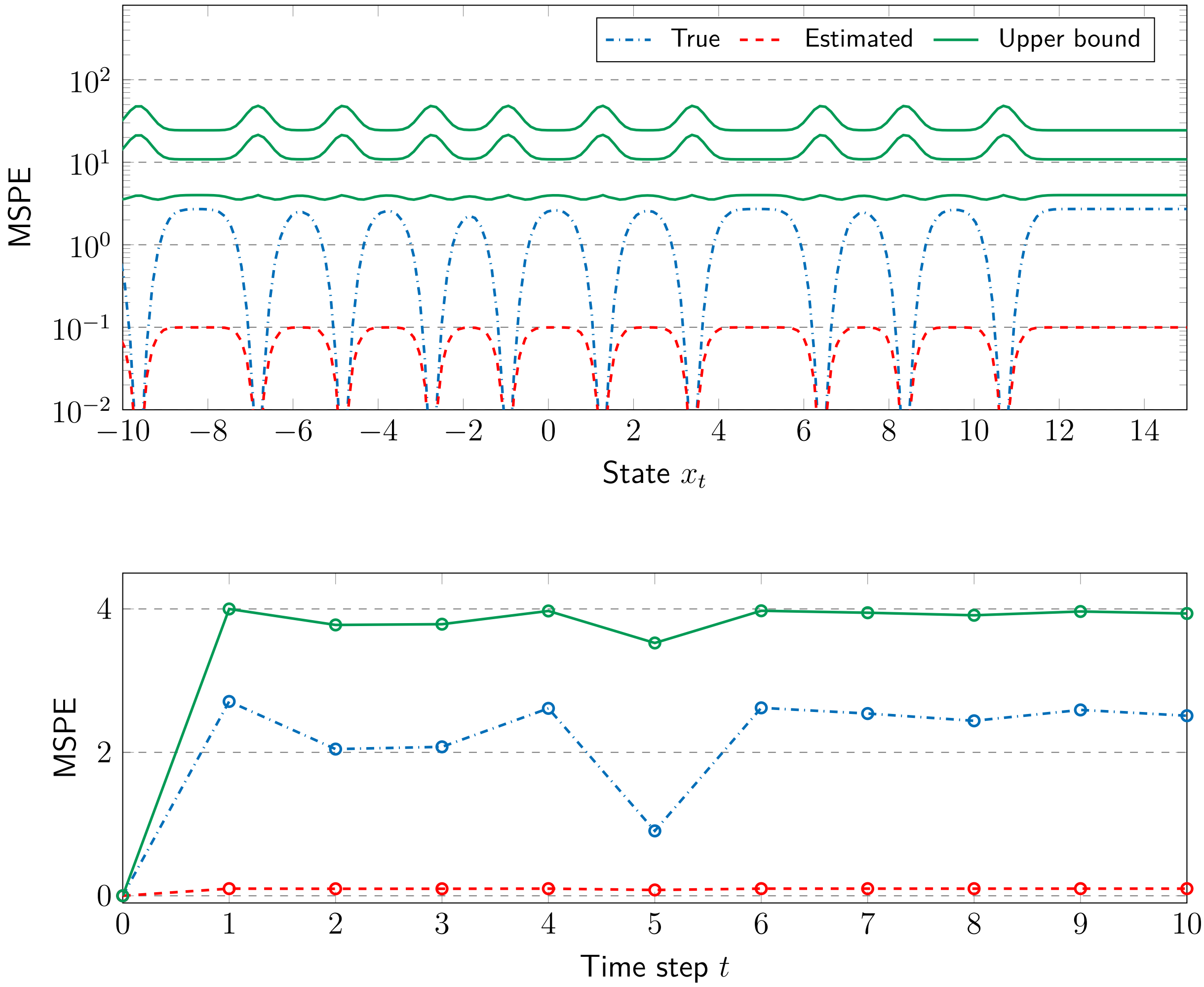


Figure 4.13: Top: The estimated, the true and the upper bound of the MSPE for a 10%, 100%, and 200% error interval (from bottom to top) around the correct hyperparameter values. Bottom: The comparison in time domain with the 10% bound.

4.3 Discussion

In this section, we focus on the kernel selection for GP models from a control point of view. First, we present a framework for the model selection for kernel-based models to directly optimize the overall closed-loop control performance. For this purpose, the kernel and its hyperparameters are optimized using Bayesian optimization with respect to a cost function that evaluates the performance of the closed-loop. Here, a modified Bayesian optimization approach is used, which integrates integer-value inputs (the kernel selection) properly. It is shown that this approach allows to preserve the control architecture properties as only the model is adapted. Two simulations and a robotic experiment demonstrate the advantages of the proposed approach in comparison to the standard approach using data-based model selection techniques. Even with a larger data set, the performance of the optimized model is still better. For large systems, however, the large amount of parameters to optimize might lead to many task repetitions before a performance increase is noticeable. Furthermore, the optimization requires a repetitive task. It is expected that, the optimized model shows similar performance for similar tasks. However, the extension and analysis for different tasks is open for future work.

In case of a misspecified kernel in the GP model, we derive an upper bound for the mean square prediction error between an estimated GP model and a GP model with unknown kernel function. For our upper bound calculus, no exact knowledge about the underlying kernel function is required. Instead, only a set of possible kernel functions with their hyperparameter sets is necessary. With additional weak assumptions, a closed-form solution is provided. A numerical example demonstrates that this bound confines the usually unknown mean square prediction error. The size of the bound depends on the number of possible kernels and set size of possible hyperparameters. Thus, for very large sets, the proposed bound might be too loose. In addition, the numerical evaluation shows that the bound becomes looser next to the training points. Finding better approximations for the individual terms of the MSPE would improve the bounds.

Control of Euler-Lagrange Systems 5

The previous chapters deal with the properties and selection of GP models. In this chapter, we present GP model based control approaches that provide stability and performance guarantees. Euler-Lagrange (EL) systems represent a crucial and large class of dynamical systems so that the control of these systems is of high interest. Various control schemes for EL systems have been proposed as discussed in [Ort+13]. Most of them can be considered as a subset of computed torque control law (CTC), which enable very effective controllers in robust, adaptive and learning control [Sic+10]. The control law is separated into a feed-forward and a feedback part. A precise model of the true system is necessary to compensate the system dynamics with the feed-forward term to keep the feedback term low. The combination of a feed-forward and a feedback part is beneficial in many ways: it avoids large errors in the presence of noise [Isi13], avoids the saturation of actuators [KG02], and enhances safety in applications such as human-robot interaction [San+08]. Since the accuracy of the compensation depends on the precision of the model, all generalized external forces, such as, for instance in robotics, friction, payload or contact forces with the environment, must be incorporated as precisely as possible. However, an accurate model of these uncertainties is hard to obtain by classical first-order principles based techniques. Especially in modern applications of Euler-Lagrange systems, such as service robotics, the interaction with unstructured and a priori unknown environments further increases the uncertainty.

Related Work and Open Problems

The uncertainties of a model can be separated into structural and parametric variations. The structural uncertainties come from the lack of knowledge of the underlying true physics. Parametric uncertainties exist since the exact values of lengths, masses, etc., are often unknown. A common approach is to derive a dynamic model from first-order physics and to increase the feedback gains to compensate the uncertainties until a desired tracking performance is achieved [SHV06]. However, high gain control is undesirable (as explained above) and deriving a more accurate model of the system is often difficult if not impossible, e.g., in soft robotics [Mog+16]. Additionally, the stability of the closed-loop system might not be guaranteed due to the uncertainties.

Computed torque control requires a parametric model of the EL system, which can be identified, e.g., for robot manipulators using [SS12]. Errors in the identified dynamics deteriorate the tracking performance and can affect the stability of the closed-loop. Several methods are presented to overcome this problem, for instance in [SL87; RWH04], and also in terms of passivation in [Fra03; BIW91] but all approaches need a parametric model of the actual system. The idea to use GP models as a data-driven approach in control of robotic systems has been presented in [NSP08; AMS14]. The inherent learning of variable loads for robotic

manipulators is provided by [Wil+09]. However, no stability guarantees for the closed-loop are given. In [Cho+15; Ber+16], the stability of a class of systems with GP models is analyzed. However, the authors do not exploit the particular structure of EL systems. Thus, high performance tracking control of EL systems with unknown dynamics and stability guarantees is still an open challenge.
In Section 5.2, we focus on the tracking control of Euler-Lagrange systems with unknown dynamics. A computed torque based control law using GP models is proposed, which guarantees the stability of the system for a class of unknown dynamics. Afterwards, we introduce how to use the uncertainty description of the GP models in computed torque based control and, additionally, provide performance guarantees in Section 5.3. Finally, we show how GP models are used for the passivation of systems with unknown dynamics in Section 5.4.

5.1 Dynamics of Euler-Lagrange Systems

In this chapter, we consider the class of non-conservative and fully-actuated systems where the equations of motion are given by

$$\frac{d}{dt_c}\left(\frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{q}}}\right) - \frac{\partial \mathcal{L}}{\partial \boldsymbol{q}} = \boldsymbol{u} + \boldsymbol{u}_d \tag{5.1}$$

with the generalized coordinates $\boldsymbol{q} \in \mathbb{R}^{n_q}$, velocities $\dot{\boldsymbol{q}} \in \mathbb{R}^{n_q}$ and the general Lagrangian function $\mathcal{L}(\dot{\boldsymbol{q}}, \boldsymbol{q}) \coloneqq \mathcal{T}(\dot{\boldsymbol{q}}, \boldsymbol{q}) - \mathcal{V}(\boldsymbol{q})$. This function depends on the kinetic energy (or co-energy) $\mathcal{T}\colon \mathbb{R}^{n_q} \times \mathbb{R}^{n_q} \to \mathbb{R}$ and the potential function $\mathcal{V}\colon \mathbb{R}^{n_q} \to \mathbb{R}$. Two types of generalized external forces are considered: The action of control $\boldsymbol{u} \in \mathbb{R}^{n_q}$ and the effect of the unknown dynamics $\boldsymbol{u}_d \in \mathbb{R}^{n_q}$.

Assumption 5.1. *The effect of the unknown dynamics $\boldsymbol{u}_d$ in*(5.1) *can be parametrized by $\boldsymbol{u}_d = \boldsymbol{f_u}\left(\boldsymbol{q}_q\right)$ with $\boldsymbol{q}_q = [\ddot{\boldsymbol{q}}; \dot{\boldsymbol{q}}; \boldsymbol{q}]$, where $\boldsymbol{f_u}\colon \mathbb{R}^{3n_q} \to \mathbb{R}^{n_q}$ is a continuous function.*

The assumption restricts $\boldsymbol{f_u}$ to be not directly time dependent, which holds in many application scenarios. For example, the common unknown dynamics in robotic systems, i.e., Columb and viscous friction, are included. The kinetic energy in the EL equation (5.1) is of the form $\mathcal{T}(\dot{\boldsymbol{q}}, \boldsymbol{q}) = \frac{1}{2}\dot{\boldsymbol{q}}^\top H(\boldsymbol{q})\dot{\boldsymbol{q}}$, where $H(\boldsymbol{q})\colon \mathbb{R}^{n_q} \to \mathbb{R}^{n_q \times n_q}$ is the symmetric and positive definite generalized inertia matrix. Based on these assumptions, the EL equation (5.1) can be written in the equivalent form

$$H(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{f_u}(\boldsymbol{q}_q) = \boldsymbol{u}(t_c), \tag{5.2}$$

where $C(\boldsymbol{q}, \dot{\boldsymbol{q}})\colon \mathbb{R}^{n_q} \times \mathbb{R}^{n_q} \to \mathbb{R}^{n_q \times n_q}$ is the generalized Coriolis matrix and the generalized vector $\boldsymbol{g}(\boldsymbol{q})\colon \mathbb{R}^{n_q} \to \mathbb{R}^{n_q}$ is given by $\boldsymbol{g}(\boldsymbol{q}) \coloneqq \frac{\partial \mathcal{V}(\boldsymbol{q})}{\partial \boldsymbol{q}}$. The time-dependency of the states is omitted for simplicity of notation and the time-dependency of the input $\boldsymbol{u}\colon \mathbb{R}_{\geq 0} \to \mathbb{R}^{n_q}$ might be also indirect, i.e., $\boldsymbol{u}(\boldsymbol{q}_q(t_c))$.

Property 5.1. Without loss of generality, we consider the non-unique matrix $C(\boldsymbol{q}, \dot{\boldsymbol{q}})$ to be defined such that $\dot{H}(\boldsymbol{q}) - 2C(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{n_q \times n_q}$ is skew-symmetric $\forall \dot{\boldsymbol{q}}, \boldsymbol{q} \in \mathbb{R}^{n_q}$ following [MLS94, Lemma 4.2].

The goal is to design a control action $\boldsymbol{u}$ such that the system (5.2) follows a desired trajectory $\ddot{\boldsymbol{q}}_{\boldsymbol{d}}(t_c), \dot{\boldsymbol{q}}_{\boldsymbol{d}}(t_c), \boldsymbol{q}_{\boldsymbol{d}}(t_c) \in \mathbb{R}^{n_q}$, which satisfies the following assumption.

Assumption 5.2. *The desired trajectory satisfies* $\|\boldsymbol{q_d}\| < \bar{q}_d$, $\|\dot{\boldsymbol{q}}_{\boldsymbol{d}}\| < \bar{\dot{q}}_d$, *and* $\|\ddot{\boldsymbol{q}}_{\boldsymbol{d}}\| < \bar{\ddot{q}}_d$ *with* $\bar{q}_d, \bar{\dot{q}}_d, \bar{\ddot{q}}_d \in \mathbb{R}_{>0}$.

The condition of bounded reference motion trajectories is a very natural assumption and does not pose any restriction in practice.

5.1.1 Hybrid Learning with GP Models

Consider the EL system in (5.2) with unknown parameterization of $H(\boldsymbol{q})$,$C(\boldsymbol{q}, \dot{\boldsymbol{q}})$, $\boldsymbol{g}(\boldsymbol{q})$ and unknown dynamics $\boldsymbol{f_u}$. If a priori knowledge of the plant is available, a hybrid learning approach can be used which is a combination of a parametric and a data-driven model. We consider the estimated model to be given by

$$\hat{\boldsymbol{u}}(t_c) = \hat{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \hat{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{g}}(\boldsymbol{q}), \tag{5.3}$$

where $\hat{H}(\boldsymbol{q}) \in \mathbb{R}^{n_q \times n_q}, \hat{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{n_q \times n_q}$ and $\hat{\boldsymbol{g}}(\boldsymbol{q}) \in \mathbb{R}^{n_q}$ are estimates of the true values which also satisfy Property 5.1. Furthermore, the estimates must fulfill the following property.

Property 5.2 (Structure of the estimates)**.** There exist constants $\underline{h}, \overline{h}, \bar{c}_C \in \mathbb{R}_{>0}$, such that $\underline{h}\|\boldsymbol{z}\|^2 \leq \boldsymbol{z}^\top \hat{H}(\boldsymbol{q})\boldsymbol{z} \leq \overline{h}\|\boldsymbol{z}\|^2$, and $\|\hat{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\| \leq \bar{c}_C\|\dot{\boldsymbol{q}}\|$, where $\hat{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{q}' = \hat{C}(\boldsymbol{q}, \boldsymbol{q}')\dot{\boldsymbol{q}}$ for all $\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{q}', \boldsymbol{z} \in \mathbb{R}^{n_q}$.

The identification of the estimates $\hat{H}, \hat{C}, \hat{\boldsymbol{g}}$ while satisfying Properties 5.1 and 5.2 can be achieved following the identification procedures from [SHV06; Koz12]. Note that Property 5.2 is required for the estimates only and not for the true system (5.2).

Remark 5.1. *Without prior knowledge of the system, the estimates are set to* $\hat{H} = I_{n_q}, \hat{C} = 0$, *and* $\hat{\boldsymbol{g}} = \boldsymbol{0}$.

After having selected the parametric model, a multi-output GP model $\boldsymbol{f}_{GP}$ following (2.13) is trained with $n_{\mathcal{D}}$ data pairs $\mathcal{D} = \{\boldsymbol{q}_q^{\{i\}}, \check{\boldsymbol{\tau}}^{\{i\}}\}_{i=1}^{n_{\mathcal{D}}}$ of the system such that

$$\boldsymbol{f}_{GP}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_q, \mathcal{D}) = \begin{bmatrix} f_{GP,1} \sim \mathcal{N}(\mu_1(\check{\boldsymbol{\tau}}|\boldsymbol{q}_q, \mathcal{D}), \mathrm{var}_1(\check{\boldsymbol{\tau}}|\boldsymbol{q}_q, \mathcal{D})) \\ \vdots \\ f_{GP,n_q} \sim \mathcal{N}(\mu_{n_q}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_q, \mathcal{D}), \mathrm{var}_{n_q}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_q, \mathcal{D})) \end{bmatrix} \tag{5.4}$$

with kernels $k^1, \ldots, k^{n_q} \in \mathcal{K}$ and zero mean functions $m^1, \ldots, m^{n_q} = 0$. The training data set consists of $\boldsymbol{q}_q = [\ddot{\boldsymbol{q}}; \dot{\boldsymbol{q}}; \boldsymbol{q}] \in \mathbb{R}^{3n_q}$ as input data, and the difference between the real system dynamics (5.2) and the estimated model (5.3) as output data $\check{\boldsymbol{\tau}}$. This residual dynamic is given by

$$\check{\boldsymbol{\tau}}(\boldsymbol{q}_q) = \check{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \check{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \check{\boldsymbol{g}}(\boldsymbol{q}) - \boldsymbol{f_u}(\boldsymbol{q}_q), \tag{5.5}$$

$$\text{with } \check{H} = H - \hat{H},\ \check{C} = C - \hat{C},\ \check{\boldsymbol{g}} = \boldsymbol{g} - \hat{\boldsymbol{g}}. \tag{5.6}$$

For the generation of training data set $\mathcal{D}$, the system (5.2) can be operated by an arbitrary controller as shown in Fig. 5.1. The only imposed requirement to the controller is that a finite sequence of training data of the system can be collected. Stability, however, is not necessarily required. It is advisable to choose the area of training points close to the desired operation area but this is not mandatory.

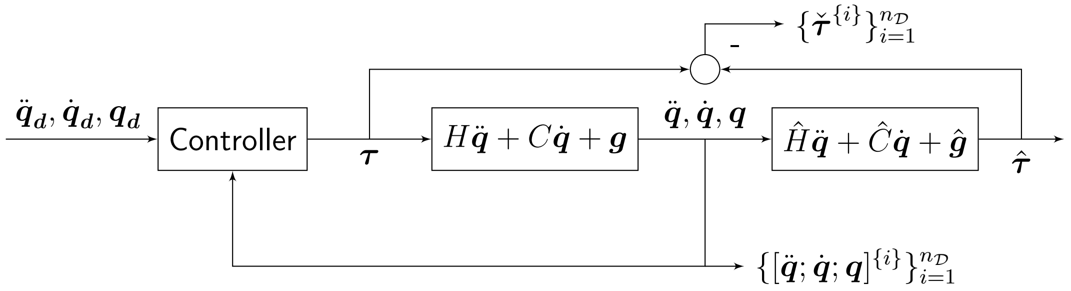

$\ddot{\boldsymbol{q}}_d, \dot{\boldsymbol{q}}_d, \boldsymbol{q}_d$
Controller
$\boldsymbol{\tau}$
$H\ddot{\boldsymbol{q}} + C\dot{\boldsymbol{q}} + \boldsymbol{g}$
$\ddot{\boldsymbol{q}}, \dot{\boldsymbol{q}}, \boldsymbol{q}$
$\hat{H}\ddot{\boldsymbol{q}} + \hat{C}\dot{\boldsymbol{q}} + \hat{\boldsymbol{g}}$
$\hat{\boldsymbol{\tau}}$
-
$\{\check{\boldsymbol{\tau}}^{\{i\}}\}_{i=1}^{n_\mathcal{D}}$
$\{[\ddot{\boldsymbol{q}}; \dot{\boldsymbol{q}}; \boldsymbol{q}]^{\{i\}}\}_{i=1}^{n_\mathcal{D}}$

Figure 5.1: The structure for generating the training data set $\mathcal{D} = \{[\ddot{\boldsymbol{q}}; \dot{\boldsymbol{q}}; \boldsymbol{q}]^{\{i\}}, \check{\boldsymbol{\tau}}^{\{i\}}\}_{i=1}^{n_\mathcal{D}}$ for the GP model.

Remark 5.2. *An appropriate choice of training points is obtained using the Bayesian optimization method where the next training point is set to the position of maximum variance, as proposed in [Sui+15]. An alternative way is presented in [Cap+20], where an information theoretical criteria is used to select the training data.*

After collecting the training pairs, the hyperparameters are optimized following Section 2.2.2.

5.2 GP based Augmented Computed Torque Control

In this section, we present an augmented computed torque control law based on GP models, which guarantees stability of the closed-loop. The control law uses a multi-output GP model (5.4) to compensate the unknown dynamics. The proposed method also abstains from feeding back the generalized accelerations (in contrast to, e.g., [AMS14]) as these are difficult to measure directly and often inject noise. Our method guarantees that the tracking error is stochastically bounded around zero, independent of the number of training data. If the number of training points tends to infinity, this bound becomes arbitrary small and the tracking error is asymptotically stable. For deriving the mathematical calculus, we consider systems with the following additional properties only

Property 5.3 (Boundedness and Linearity)**.**

- The inertia matrix H is bounded and Lipschitz continuous, such that $\|H(\boldsymbol{q})\| < \infty$ and $\|H(\boldsymbol{q}) - H(\boldsymbol{q}')\| \leq c_L \|\boldsymbol{q} - \boldsymbol{q}'\|$ with $c_L \geq 0$, for all $\boldsymbol{q}, \boldsymbol{q}' \in \mathbb{R}^{n_q}$.
- The matrix $C(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is bounded in $\boldsymbol{q}$ and linear in $\dot{\boldsymbol{q}}$, i.e., bounded by $\|C(\boldsymbol{q}, \dot{\boldsymbol{q}})\| \leq c\|\dot{\boldsymbol{q}}\|$ and $C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}' = C(\boldsymbol{q}, \dot{\boldsymbol{q}}')\dot{\boldsymbol{q}}$ for all $\dot{\boldsymbol{q}}, \dot{\boldsymbol{q}}', \boldsymbol{q} \in \mathbb{R}^{n_q}$ and $c \in \mathbb{R}_{>0}$.

Both properties hold for many real world system, e.g, for robots with rotational joints only. Furthermore, we make the following assumptions.

Assumption 5.3. *The kernel functions $k^1(\boldsymbol{q}_q, \boldsymbol{q}_q'), \ldots, k^{n_q}(\boldsymbol{q}_q, \boldsymbol{q}_q')$ of the GP model* (5.4) *are bounded for all $\boldsymbol{q}_q, \boldsymbol{q}_q' \in \mathbb{R}^{3n_q}$.*

Assumption 5.4. *There is no additional unknown dynamics, i.e., $\boldsymbol{f_u}(\boldsymbol{q}_q) = 0$.*

Assumption 5.5. *The norm of the model error is affinely bounded by the norm of the angular velocity, i.e.,* $\|H(\boldsymbol{q})\ddot{\boldsymbol{q}}_{\boldsymbol{d}}+C(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_{\boldsymbol{d}}+\boldsymbol{g}(\boldsymbol{q})-\hat{H}(\boldsymbol{q})\ddot{\boldsymbol{q}}_{\boldsymbol{d}}-\hat{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_{\boldsymbol{d}}-\hat{\boldsymbol{g}}(\boldsymbol{q})\| \leq \bar{c}_\alpha+\bar{c}_\beta\|\dot{\boldsymbol{q}}\|$ *for all* $\ddot{\boldsymbol{q}}_{\boldsymbol{d}}, \dot{\boldsymbol{q}}_{\boldsymbol{d}}, \dot{\boldsymbol{q}}, \boldsymbol{q} \in \mathbb{R}^{n_q}$ *with* $\bar{c}_\alpha, \bar{c}_\beta \in \mathbb{R}_{>0}$*, and continuous with respect to* $\ddot{\boldsymbol{q}}_{\boldsymbol{d}}, \dot{\boldsymbol{q}}_{\boldsymbol{d}}, \dot{\boldsymbol{q}}, \boldsymbol{q}$.

Assumption 5.3 is satisfied by many commonly chosen kernels such as the squared exponential kernel, see (2.47). Assumption 5.4 restricts the unknown dynamics in (5.2) to be part of the functions H, C and $\boldsymbol{g}$. For instance, the effect of unknown payload of a robot to its dynamics (5.2) can be expressed in H, C and $\boldsymbol{g}$. From a practical point of view, Assumption 5.5 states that the dynamics which are not modeled by (5.2) can at most depend linearly on the joint velocity. If there is a known range of uncertainty only in the inertia parameter, the values $\bar{c}_\alpha$ and $\bar{c}_\beta$ can be computed using the approach of [TA81]. Since the payload in mechanical systems is the major reason for the uncertainty, this approach is suitable for many application scenarios.

5.2.1 Control Law

The following theorem introduces a control law, visualized in Fig. 5.2, to ensure a bounded tracking error under the presented conditions.

Theorem 5.1. *Consider an* n_q*-dimensional Euler-Lagrange system* (5.2) *for which Property 5.3 and Assumptions 5.3 to 5.5 hold. Let* $\hat{\boldsymbol{\tau}} = \hat{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \hat{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{g}}(\boldsymbol{q})$ *be an estimated model of* (5.2) *and* $K_p, K_d \in \mathbb{R}^{n_q \times n_q}$ *two positive definite matrices. Consider the control law*

$$\boldsymbol{u} = \hat{H}(\boldsymbol{q})\ddot{\boldsymbol{q}}_{\boldsymbol{d}} + \hat{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_{\boldsymbol{d}} + \hat{\boldsymbol{g}}(\boldsymbol{q}) + \boldsymbol{f}_{GP}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_{\boldsymbol{c}}, \mathcal{D}) - K_d\dot{\boldsymbol{e}} - K_p\boldsymbol{e}, \tag{5.7}$$

with $\boldsymbol{q}_{\boldsymbol{c}} = [\ddot{\boldsymbol{q}}_{\boldsymbol{d}}; \dot{\boldsymbol{q}}_{\boldsymbol{d}}; \boldsymbol{q}]$ *and* $\underline{\sigma}(K_d) > \bar{c}_\beta$*. Then, the tracking error* $\boldsymbol{e} = \boldsymbol{q}-\boldsymbol{q}_{\boldsymbol{d}}$ *is stochastically sample path bounded.*

Details about *stochastic sample path boundedness* are provided in Definition A.11.

Remark 5.3. *The GP is trained over* $\ddot{\boldsymbol{q}}, \dot{\boldsymbol{q}}, \boldsymbol{q}$ *but receives* $\ddot{\boldsymbol{q}}_{\boldsymbol{d}}, \dot{\boldsymbol{q}}_{\boldsymbol{d}}, \boldsymbol{q}$ *as inputs in the control law. This is beneficial for practical implementation as this requires no feedback of the manipulator's acceleration and velocity. Additionally, the dependency of the Coriolis term* $\check{C}$ *on* $\dot{\boldsymbol{q}}$

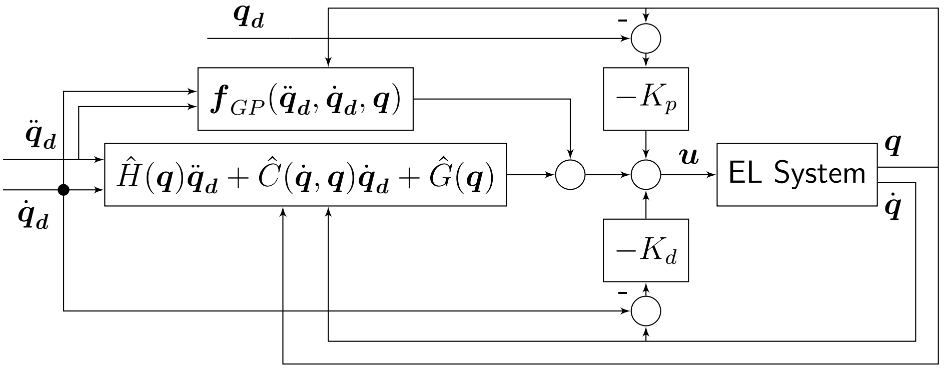

$\boldsymbol{q}_{\boldsymbol{d}}$
-
$\boldsymbol{f}_{GP}(\ddot{\boldsymbol{q}}_{\boldsymbol{d}}, \dot{\boldsymbol{q}}_{\boldsymbol{d}}, \boldsymbol{q})$
$-K_p$
$\ddot{\boldsymbol{q}}_{\boldsymbol{d}}$
$\hat{H}(\boldsymbol{q})\ddot{\boldsymbol{q}}_{\boldsymbol{d}} + \hat{C}(\dot{\boldsymbol{q}}, \boldsymbol{q})\dot{\boldsymbol{q}}_{\boldsymbol{d}} + \hat{G}(\boldsymbol{q})$
$\boldsymbol{u}$
EL System
$\boldsymbol{q}$
$\dot{\boldsymbol{q}}$
$\dot{\boldsymbol{q}}_{\boldsymbol{d}}$
$-K_d$
-

Figure 5.2: Structure of the closed-loop with proposed control law (5.7).

is problematic because it cannot be identified isolated from the angular velocity $\dot{\boldsymbol{q}}$, which is multiplied with $\check{C}$. Therefore, we use the desired velocity $\dot{\boldsymbol{q}}$ for the input of the GP which does not lead to any problems due to Assumption 5.5. So the GP model is an augmented model like the parametric model in augmented computed torque control.

Proof of Theorem 5.1. Employing the control law (5.7), the closed-loop system is given by

$$\begin{aligned}\ddot{\boldsymbol{e}} = {}& H(\boldsymbol{q})^{-1}\Big(\hat{H}(\boldsymbol{q})\ddot{\boldsymbol{q}}_{\boldsymbol{d}} + \hat{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_{\boldsymbol{d}} - C(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{g}}(\boldsymbol{q}) - \boldsymbol{g}(\boldsymbol{q})\\ &+ \boldsymbol{f}_{GP}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_{\boldsymbol{c}},\mathcal{D}) - K_d\dot{\boldsymbol{e}} - K_p\boldsymbol{e}\Big) - \ddot{\boldsymbol{q}}_{\boldsymbol{d}},\end{aligned} \tag{5.8}$$

since the matrix $H(\boldsymbol{q})$ is always non-singular. The posterior of the GP $\boldsymbol{f}_{GP}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_{\boldsymbol{c}},\mathcal{D})$ can be split in a drift $\boldsymbol{\mu}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_{\boldsymbol{c}},\mathcal{D})$ and a diffusion term $\Sigma(\check{\boldsymbol{\tau}}|\boldsymbol{q}_{\boldsymbol{c}},\mathcal{D})\boldsymbol{\nu}_{\mathrm{brw}}$, where $\boldsymbol{\mu}$ and Σ are defined as in (2.15). The vector $\boldsymbol{\nu}_{\mathrm{brw}} = [\nu_{\mathrm{brw},1}\dots\nu_{\mathrm{brw},n_q}]^\top$ denotes n_q-dimensional standard Brownian noise. We reformulate the closed-loop system with a drift and a diffusion term

$$\begin{aligned}\frac{d}{dt}\begin{bmatrix}\dot{\boldsymbol{e}}\\ \boldsymbol{e}\end{bmatrix} = {}& \underbrace{\begin{bmatrix}H^{-1}(\hat{H}\ddot{\boldsymbol{q}}_{\boldsymbol{d}} + \hat{C}\dot{\boldsymbol{q}}_{\boldsymbol{d}} - C\dot{\boldsymbol{q}} + \hat{\boldsymbol{g}} - \boldsymbol{g} + \boldsymbol{\mu}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_{\boldsymbol{c}},\mathcal{D}) - K_d\dot{\boldsymbol{e}} - K_p\boldsymbol{e}) - \ddot{\boldsymbol{q}}_{\boldsymbol{d}}\\ \dot{\boldsymbol{e}}\end{bmatrix}}_{\text{drift}}\\ &+ \underbrace{\begin{bmatrix}H^{-1}\Sigma^{\frac{1}{2}}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_{\boldsymbol{c}},\mathcal{D})\\ 0\end{bmatrix}\boldsymbol{\nu}_{\mathrm{brw}}}_{\text{diffusion}}.\end{aligned} \tag{5.9}$$

For the stability analysis of this stochastic differential equation, we use the differential generator $\mathfrak{L}$, described in Theorem A.1, which maps C^2 functions $V\colon \mathbb{R}^{2n_q} \to \mathbb{R}$ to C^0 functions $\mathfrak{L}V\colon \mathbb{R}^{2n_q} \to \mathbb{R}$. Consider the following Lyapunov function

$$V(\dot{\boldsymbol{e}},\boldsymbol{e}) = \frac{1}{2}\begin{bmatrix}\dot{\boldsymbol{e}}\\ \boldsymbol{e}\end{bmatrix}^\top \begin{bmatrix}H & \varepsilon H\\ \varepsilon H^\top & K_p\end{bmatrix}\begin{bmatrix}\dot{\boldsymbol{e}}\\ \boldsymbol{e}\end{bmatrix} \tag{5.10}$$

with an $\varepsilon > 0$. We start with the computation of $\mathfrak{L}V(\dot{\boldsymbol{e}},\boldsymbol{e})$ using the property that the matrix H^{-1} is symmetric which leads to

$$\begin{aligned}\mathfrak{L}V = {}& \begin{bmatrix}\dot{\boldsymbol{e}}^\top H + \varepsilon\boldsymbol{e}^\top H\\ \boldsymbol{e}^\top K_p + \frac{1}{2}\dot{\boldsymbol{e}}^\top\dot{H} + \varepsilon(\boldsymbol{e}^\top\dot{H} + \dot{\boldsymbol{e}}^\top H)\end{bmatrix}^\top\\ &\underbrace{\begin{bmatrix}H^{-1}(\hat{H}\ddot{\boldsymbol{q}}_{\boldsymbol{d}} + \hat{C}\dot{\boldsymbol{q}}_{\boldsymbol{d}} - C\dot{\boldsymbol{q}} + \hat{\boldsymbol{g}} - \boldsymbol{g} + \boldsymbol{\mu}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_{\boldsymbol{c}},\mathcal{D}) - K_d\dot{\boldsymbol{e}} - K_p\boldsymbol{e}) - \ddot{\boldsymbol{q}}_{\boldsymbol{d}}\\ \dot{\boldsymbol{e}}\end{bmatrix}}_{\text{drift}}\\ &+ \frac{1}{2}\operatorname{tr}\Sigma^{\frac{1}{2}}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_{\boldsymbol{c}},\mathcal{D})H^{-1}\Sigma^{\frac{1}{2}}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_{\boldsymbol{c}},\mathcal{D}).\end{aligned} \tag{5.11}$$

It can be further simplified to

$$\begin{aligned}\mathfrak{L}V = {}& \begin{bmatrix}\dot{\boldsymbol{e}}^\top & \boldsymbol{e}^\top\end{bmatrix}\underbrace{\begin{bmatrix}\underbrace{-K_d + \varepsilon H}_{M_{11}} & \underbrace{\frac{\varepsilon}{2}(-K_d^\top + \dot{H} - C)}_{M_{12}}\\ \underbrace{\frac{\varepsilon}{2}(-K_d + (\dot{H} - C)^\top)}_{M_{21}=M_{12}} & \underbrace{-\varepsilon K_p}_{M_{22}}\end{bmatrix}}_{M\in\mathbb{R}^{2n_q\times 2n_q}}\begin{bmatrix}\dot{\boldsymbol{e}}\\ \boldsymbol{e}\end{bmatrix}\\ &+ (\dot{\boldsymbol{e}} + \varepsilon\boldsymbol{e})^\top(\check{H}\ddot{\boldsymbol{q}}_{\boldsymbol{d}} + \check{C}\dot{\boldsymbol{q}}_{\boldsymbol{d}} + \check{\boldsymbol{g}} + \boldsymbol{\mu}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_{\boldsymbol{c}},\mathcal{D})) + \frac{1}{2}\operatorname{tr}\Sigma^{\frac{1}{2}}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_{\boldsymbol{c}},\mathcal{D})H^{-1}\Sigma^{\frac{1}{2}}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_{\boldsymbol{c}},\mathcal{D}),\end{aligned} \tag{5.12}$$

where $\dot{H}-2C$ is canceled out as Property 5.1 implies the skew-symmetry of $\dot{H}-2C$. The matrices $\check{H}, \check{C}, \check{\boldsymbol{g}}$ denote the difference between the EL system (5.2) and the estimated model matrices as defined in (5.6). The matrices M_{11} and M_{22} are negative definite for sufficient small ε. The non-diagonal components $\dot{\boldsymbol{e}}^\top M_{12}\boldsymbol{e}, \boldsymbol{e}^\top M_{21}\dot{\boldsymbol{e}}$ are bounded by

$$\dot{\boldsymbol{e}}^\top M_{12}\boldsymbol{e} = \boldsymbol{e}^\top M_{21}\dot{\boldsymbol{e}} \leq \frac{\varepsilon}{2}\left(\|K_d\| + \frac{5}{2}\left\|\frac{\partial H}{\partial \boldsymbol{q}}\right\|\|\dot{\boldsymbol{q}}\|\right)\|\boldsymbol{e}\|\|\dot{\boldsymbol{e}}\| \leq \frac{\varepsilon}{2}c_{V3}\|\boldsymbol{e}\|\|\dot{\boldsymbol{e}}\|, \tag{5.13}$$

with $c_{V3} \in \mathbb{R}_{>0}$ derived by using the chain rule and the dependency between $\dot{H}$ and C. The partial derivation $\left\|\frac{\partial H}{\partial \boldsymbol{q}}\right\|$ is a bounded operator since $H(\boldsymbol{q})$ is Lipschitz continuous. For the second line of (5.12), Assumption 5.5 guarantees that the term $\|\tilde{H}\ddot{\boldsymbol{q}}_{\boldsymbol{d}}+\check{C}\dot{\boldsymbol{q}}_{\boldsymbol{d}}+\check{\boldsymbol{g}}\|$ is bounded by an affine function $\overline{c}_\alpha + \overline{c}_\beta\|\dot{\boldsymbol{q}}\|$. The mean prediction $\|\boldsymbol{\mu}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_{\boldsymbol{c}}, \mathcal{D})\|$ and the corresponding variance $\|\Sigma(\check{\boldsymbol{\tau}}|\boldsymbol{q}_{\boldsymbol{c}}, \mathcal{D})\|$ is also bounded, see Assumption 5.3. Therefore, we obtain an upper bound for the second summand of (5.12) by

$$\begin{aligned}\|(\dot{\boldsymbol{e}}+\varepsilon\boldsymbol{e})^\top(\check{H}\ddot{\boldsymbol{q}}_{\boldsymbol{d}} + \check{C}\dot{\boldsymbol{q}}_{\boldsymbol{d}} + \check{\boldsymbol{g}} + \boldsymbol{\mu}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_{\boldsymbol{c}}, \mathcal{D}))\| &\leq \|\dot{\boldsymbol{e}}+\varepsilon\boldsymbol{e}\|(\overline{c}_\alpha + \overline{c}_\beta\|\dot{\boldsymbol{q}}\| + c_\mu) \\ &\leq \|\dot{\boldsymbol{e}}+\varepsilon\boldsymbol{e}\|\overline{c}_\beta\|\dot{\boldsymbol{e}}\| + \|\dot{\boldsymbol{e}}+\varepsilon\boldsymbol{e}\|\underbrace{(\overline{c}_\alpha + \overline{c}_\beta\bar{\dot{q}}_d + c_\mu)}_{c_{V4}}\end{aligned} \tag{5.14}$$

with $c_{V4} \in \mathbb{R}_{>0}$. Since $\|H(\dot{\boldsymbol{q}})\|$ is bounded and the matrix is always non-singular, the inverse $\|H^{-1}(\dot{\boldsymbol{q}})\|$ is also bounded such that

$$\frac{1}{2}\operatorname{tr}\Sigma^{\frac{1}{2}}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_{\boldsymbol{c}}, \mathcal{D})H^{-1}\Sigma^{\frac{1}{2}}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_{\boldsymbol{c}}, \mathcal{D}) \leq c_g \in \mathbb{R}_{>0} \tag{5.15}$$

holds for the last summand of (5.12). After combining the parts, the upper bound for the drift of the Lyapunov function V is given by

$$\mathfrak{L}V \leq -\dot{\boldsymbol{e}}^\top(K_d - \varepsilon H)\dot{\boldsymbol{e}} - \varepsilon\boldsymbol{e}^\top K_p\boldsymbol{e} + \overline{c}_\beta\|\dot{\boldsymbol{e}}\|^2 + c_{V4}\|\dot{\boldsymbol{e}}+\varepsilon\boldsymbol{e}\| + \varepsilon(c_{V3}+\overline{c}_\beta)\|\boldsymbol{e}\|\|\dot{\boldsymbol{e}}\| + c_g. \tag{5.16}$$

As we know from Theorem 5.1 that $\underline{\sigma}(K_d) > \overline{c}_\beta$, there exists an $\varepsilon > 0$ such that the first two summands dominate the third summand of the equation and the sum is negative. Since it is negative, the quadratic part of $\mathfrak{L}V$ dominates the linear and the constant part for $\|\dot{\boldsymbol{e}}^\top, \boldsymbol{e}^\top\| \to \infty$. Thus, the following holds

$$\lim_{\|\dot{\boldsymbol{e}}^\top, \boldsymbol{e}^\top\| \to \infty} \mathfrak{L}V(\dot{\boldsymbol{e}}, \boldsymbol{e}) = -\infty, \tag{5.17}$$

for an $\varepsilon > 0$. Owing to the continuity of $V(\dot{\boldsymbol{e}}, \boldsymbol{e})$, there exists a ball $\mathcal{B}_{\dot{e}e} = \{\|[\dot{\boldsymbol{e}}; \boldsymbol{e}]\| \leq r_{\dot{e}e}\}$ with the property that $\mathfrak{L}V < 0$ if $[\dot{\boldsymbol{e}}; \boldsymbol{e}] \in \mathbb{R}^{2n_q} \setminus \mathcal{B}_{\dot{e}e}$. In other words, $\mathfrak{L}V$ is negative outside $\mathcal{B}_{\dot{e}e}$ and, therefore, the tracking error is stochastically sample path bounded. □

Remark 5.4. *A similar result is given by [WRK93] but for parametric models only.*

In the proof, we apply Theorem A.1 to show that the tracking error is stochastically sample path bounded with the ball $\mathcal{B}_{\dot{e}e}$. For a radially unbounded, positive-definite Lyapunov function, we obtain that the drift operator is negative outside of this ball and, therefore, the tracking error enters the ball in a finite time.

Equation (5.12) shows the need of Assumption 5.5 to ensure the global negative definiteness of $\mathfrak{L}V$ outside the ball. However, with less restrictions on the model error, it is still possible to find local areas of boundedness, which require appropriate initial states. Important to note is the stochastic nature of the control law (5.7) as it is based on the stochastic process $\boldsymbol{f}_{GP}$. Nevertheless, its deterministic counterpart (the mean function) also results in a bounded tracking error.

Corollary 5.1. *The Euler-Lagrange system* (5.2) *with control law* (5.7) *results in a bounded tracking error if the stochastic process* $\boldsymbol{f}_{GP}(\check{\boldsymbol{\tau}}|\boldsymbol{q_c}, \mathcal{D})$ *is replaced by its deterministic posterior mean function* $\boldsymbol{\mu}(\check{\boldsymbol{\tau}}|\boldsymbol{q_c}, \mathcal{D})$ *as defined in* (2.8).

Proof. This corollary directly follows from Theorem 5.1 since stochastic sample path boundedness holds for all realizations of the stochastic control law. □

To achieve a asymptotically stable tracking error, the mean function $\boldsymbol{\mu}(\check{\boldsymbol{\tau}}|\boldsymbol{q_c}, \mathcal{D})$ of the GP must completely cancel out the model error $\check{\boldsymbol{\tau}}$ (5.3).

Corollary 5.2. *Consider the Euler-Lagrange system* (5.2) *with Property 5.3 and Assumptions 5.3 to 5.5, where the kernels* $k^1, \dots, k^{n_q}$ *are universal. Using control law* (5.7)*, the tracking error is stochastically asymptotically stable on a compact set* $\Omega_q \subset \mathbb{R}^{3n_q}$*, if the number of equally distributed training points on* Ω_q *for the GP model in* (5.7) *approaches infinity.*

Proof. If the number of training points tends to infinity, the regression with universal kernels leads to

$$\boldsymbol{\mu}(\boldsymbol{q_c}) = -\check{H}(\boldsymbol{q})\ddot{\boldsymbol{q}}_{\boldsymbol{d}} - \check{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}_{\boldsymbol{d}})\dot{\boldsymbol{q}}_{\boldsymbol{d}} - \check{\boldsymbol{g}}(\boldsymbol{q}), \tag{5.18}$$

as the continuity condition is fulfilled through Assumption 5.5. The upper bound for the drift operator of the Lyapunov function (5.10) can now be rewritten as

$$\mathfrak{L}V \leq -\dot{\boldsymbol{e}}^\top(K_d - \varepsilon H)\dot{\boldsymbol{e}} - \varepsilon\boldsymbol{e}^\top K_p\boldsymbol{e} + \varepsilon\boldsymbol{e}^\top(\dot{H} - C - K_d)\dot{\boldsymbol{e}} + \|\dot{\boldsymbol{e}}\|\beta\|\dot{\boldsymbol{e}}\| + \varepsilon\boldsymbol{e}^\top\check{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}_{\boldsymbol{d}})\dot{\boldsymbol{e}}. \tag{5.19}$$

With Property 5.3, the tracking error is stochastically asymptotically stable on Ω_q. □

5.2.2 Numerical Evaluation

As the safety of unmanned aerial vehicles increasingly gain importance [Gei+10], we base our numerical example on the control of an aerodynamic model of a NACA-0015 airfoil, as illustrated in Fig. 5.3. The inertia J_{wing} of the wing is assumed to be $1\,\text{kgm}^2$, the total mass $m_{\text{wing}} = 1\,\text{kg}$, the friction $c_{\text{wing}} = 0.1\,\frac{\text{kgm}^2}{\text{s}}$, and the distance between the joint and the center of mass $l_{\text{wing}} = 1\,\text{m}$. The goal is to control the angle q of the wing with an input torque τ. The wing is affected by an aerodynamic force f_{wing} which can be decomposed in lift and drag. The dynamics of the wing are given by

$$\tau = \underbrace{J_{\text{wing}}}_{H(q)}\ddot{q} + \underbrace{c_{\text{wing}}}_{C(\dot{q},q)}\dot{q} + \underbrace{m_{\text{wing}}g_{\text{const}}l_{\text{wing}}\sin(q) + f_{\text{wing}}(q)}_{\boldsymbol{g}(q)} \tag{5.20}$$

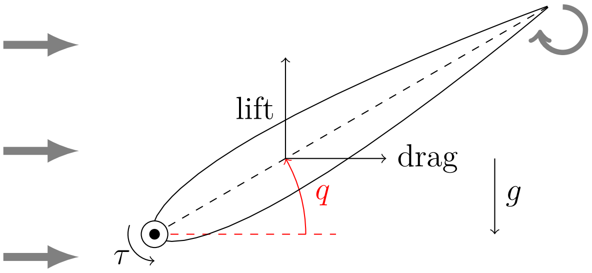


Figure 5.3: Model of torque controlled wing. Lift / drag forces are highly nonlinear functions of the angle of attack q.

These forces depend on the angle of attack, which indicates the angle between the direction of the air flow and the reference line of the wing. For a large angle of attack the lift and drag force are highly nonlinear and difficult to model analytically since air flow becomes turbulent. Our simulations are based on the measurements of the wing in a wind tunnel [SK81]. For the model shown in Fig. 5.3, the lift and drag forces are converted in the resulting torque and gravity is added. We consider the motion dynamics of a damping free pendulum for the estimated dynamics

$$\hat{J}_{\text{wing}}\ddot{q} + \hat{m}_{\text{wing}} g_{\text{const}} \hat{l}_{\text{wing}} \sin(q) = \hat{\tau}, \tag{5.21}$$

of the wing with the parameters $\hat{J}_{\text{wing}} = 0.9 J_{\text{wing}}$, $\hat{m}_{\text{wing}}\hat{l}_{\text{wing}} = 0.9 m_{\text{wing}} l_{\text{wing}}$ affected by measuring errors. Figure 5.4 shows the simulation results for the classical augmented PD control law using the estimated model. The feedback terms are set to $K_p = K_d = 5$ and the desired trajectory (dashed) is sinusoidal. Since the model contains parameter imprecision and influence of the airflow is not covered, the angle q (blue solid) differs drastically from the desired angle q_d (blue dashed). As a comparison, our proposed control law (5.7) is used. First, the GP model learns the difference between the estimated model and the real wing. For this purpose, we generate 990 homogeneous distributed pairs of torques τ and initial positions q_0 on the set $[-8, 8] \times [-\pi, \pi]$ to generate training points as shown in Fig. 5.1. The initial joint velocity and acceleration is set to zero. In this example, we do not use an extra controller but apply the torque directly for a short time interval to the manipulator. After 0.5 s the joint position, velocity and acceleration $\{\ddot{q}, \dot{q}, q\}$ are recorded. These values are inserted into the model to compute the estimated torque $\hat{\tau}$. The difference between the applied torque and the estimated torque $\check{\tau} = \tau - \hat{\tau}$ is collected. The values $\{\ddot{q}, \dot{q}, q\}$ and $\{\check{\tau}\}$ constitute a training pair.

The GP model is trained on this collection of training pairs and the hyperparameters of the squared exponential kernel are optimized with a gradient method. Afterwards, the proposed control law (5.7) with the same desired trajectory and feedback gains is used. To show the effect of the stochastic control law, we simulate 1000 realizations of the stochastic differential equation with a sample time of 1 ms. Figure 5.5 shows the mean (solid) and standard deviation (gray area) of the joint angle/velocity and the desired angles/velocity (dashed). The stochastic behavior is based on the stochastic prediction of the GP model, since the finite number of training data generates only an uncertain model. Since the GP model cancels the uncertainties of the parametric model, the mean of the joint angles converges to a tight bound around the desired angles. The size of the standard deviation depends on the certainty of the prediction of the GP model, which is influenced by the number and the distribution of the training points.

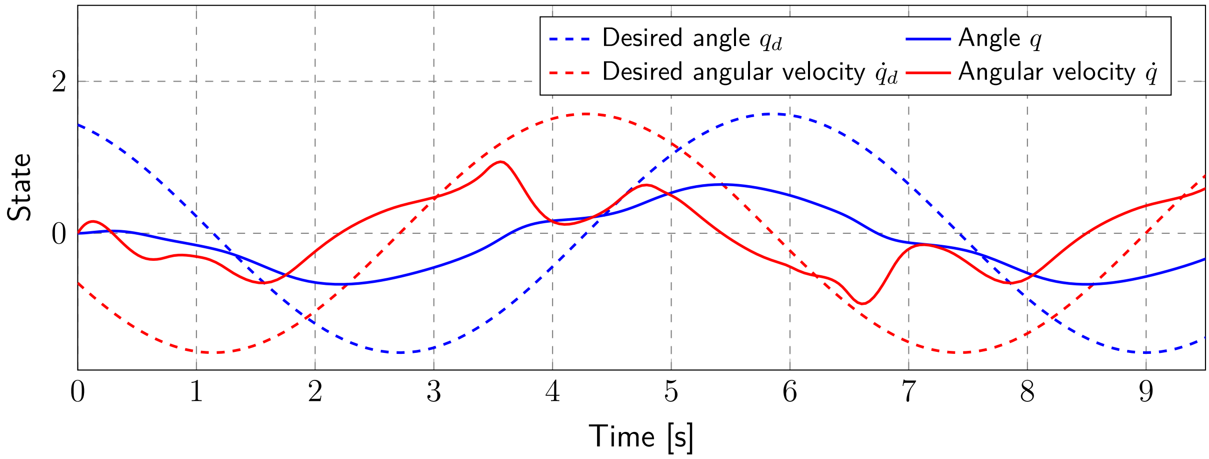


Figure 5.4: Classical augmented PD control with an estimated model does not lead to satisfactory results. The dashed lines are the desired joint position and velocity, whereas the solid lines show the true values.

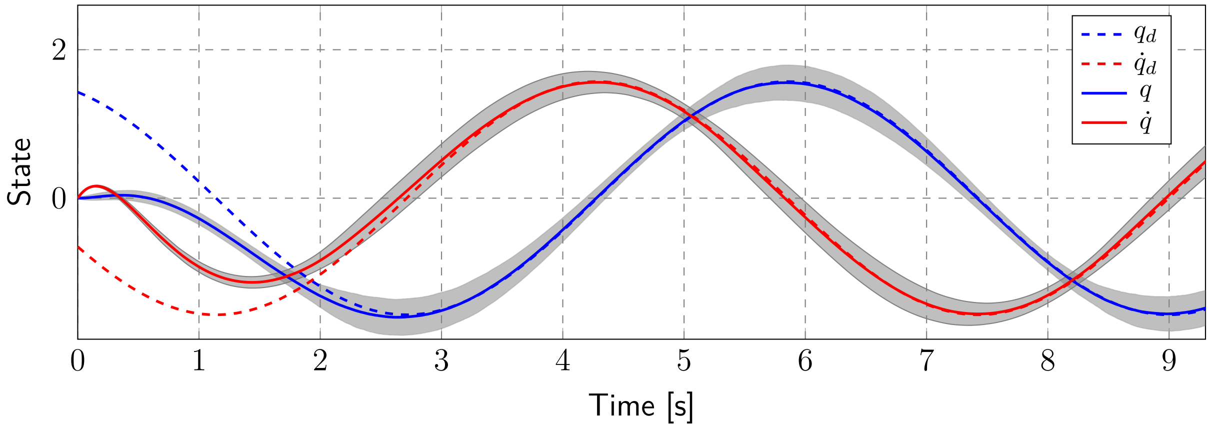


Figure 5.5: The proposed GP based control law strongly reduces the tracking error in comparison to the classical augmented PD control. The mean (solid line) of the joint angle/velocity converges to a tight bound around the desired trajectory (dashed line). The shaded area marks the 2σ interval of the 1000 simulations.

5.2.3 Experimental Evaluation

Setup

For the experimental evaluation, we use the 3-dof SCARA robot *CARBO* as pictured in Fig. 5.6. The links between the joints have a length of 0.3 m. Since the third joint just rotates a camera which is mounted as end effector, this joint is fixed for the experiment. A low level PD-controller enforces the generated torque by regulating the voltage based on a measurement of the current (which is approximately proportional to the torque). The robot manipulates a flexible rubber band. This is attached to the right side of the workspace. The task is, for example, comparable with the handling of rubber seals in the automotive manufacturing. The large time effort needed for modeling the flexible, nonlinear behavior of the

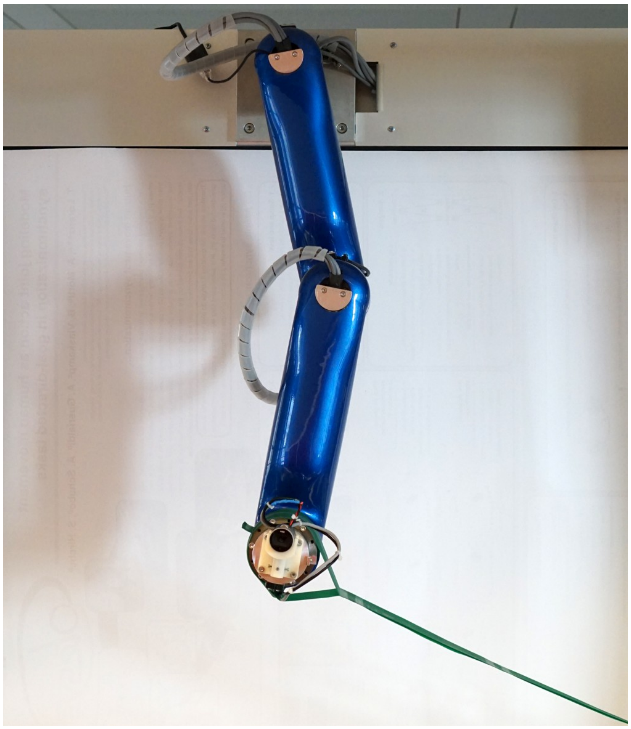


Figure 5.6: A picture of the 3-dof robot CARBO with a rubber band between the robot's end effector and the ground.

rubber band motivates the learning approach. The desired trajectory follows a sinusoidal shape with a frequency of $1\,\mathrm{s}^{-1}$ for the first, $2\,\mathrm{s}^{-1}$ for the second joint and an amplitude of $\pi/5$. The controller is implemented in MATLAB/Simulink on a Linux real-time system with a sample rate of 1 ms.

Task evaluation

As evaluation, we compare five different control approaches on the same desired trajectory.

- HG-PD: A high gain PD controller with $K_P^{(HG)} = \operatorname{diag}(800, 600)$ and $K_D^{(HG)} = \operatorname{diag}(5, 5)$ without any feed forward model.
- LG-PD: A low gain PD controller with $K_P^{(LG)} = \operatorname{diag}(20, 15)$ and $K_D^{(LG)} = \operatorname{diag}(5, 5)$ without any feed forward model.
- CTC: A computed torque controller based on a friction free model of the robot which is generated from the CAD-model combined with the LG-PD.
- CTC-SP: A computed torque controller based on a friction free model of the robot and a linear model of the rubber band combined with the LG-PD.
- CTC-SGP: A modified computed torque controller based on a friction free model of the robot and the trained GP model (our approach) in combination with the LG-PD.

The high gain approach (HG-PD) is not directly comparable to the other approach as it suffers from many disadvantages as discussed in the related work of this chapter, but it serves as a ”ground truth” here. It was also employed to generate the training data for the CTC-SGP approach by recording 351 training points corrupted by sensor noise at a rate of 30 ms while the robot follows the desired trajectory. The GPR is implemented with the GPML toolbox [RN10]. The hyperparameter of the GP model are obtained through a gradient based likelihood maximization. To obtain the best performance, we employ the deterministic version of our controller, thus using the GP’s mean function. The performance is evaluated using the root mean square error (RMSE) between the desired and the real position of the joint angles for all controllers.

Results

Figure 5.7 shows the RMSE in both joints for the different controllers. The low gain controller (LG-PD) performs very poorly, since no model knowledge is employed. This behavior is improved by adding computed torque control (CTC). Since the accuracy of the first joint increases heavily, the error of the second joint is slightly worse. If the influence of the rubber band is taken into account (CTC-SP), the accuracy for both joints is improved. Our approach (CTC-SGP) with low gain feedback clearly outperforms all other approaches with low gain and is even competitive with the high gain controller (HG-PD).
The applied torque for the first joint is visualized in Fig. 5.8. The CTC-SGP generates a torque, which is very similar to the high gain controller (HG-PD), while all others clearly differ. The influence of the amount of training data on the performance of our approach is shown in Fig. 5.9. With an increasing number of training points, the error is decreasing. The simulation and the experiment show that our approach does not only provide theoretical guarantees, but also provides performance advantages in real-world applications. The experiment illustrates that the feedback gains can be reduced by a factor of 40 while keeping the performance at a similar level. Such an significant improvement can not be achieved with our most accurate, analytically derived, physical model. Further, the simulation shows how highly nonlinear effects (turbulent airflow) can also be captured by our nonparametric modeling approach, which leads to a guaranteed diminishing tracking error.

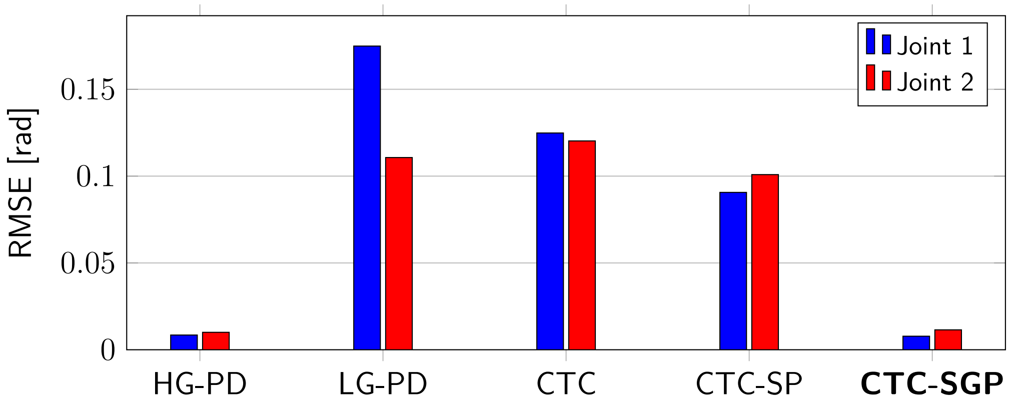
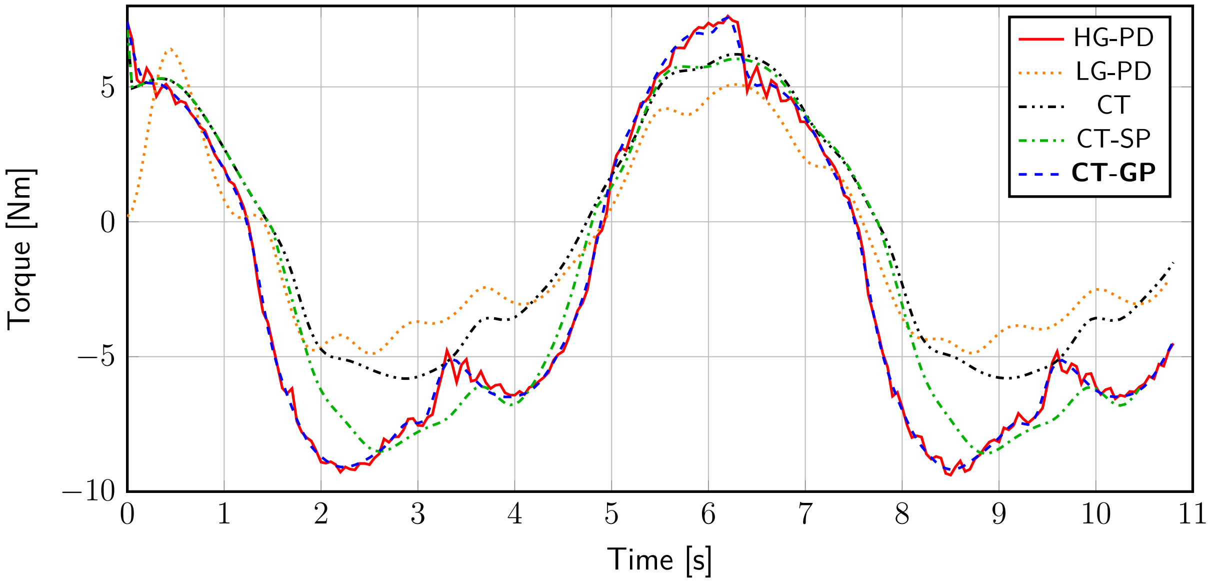


Figure 5.7: The RMSE between desired and true joint angles for the different control laws. The error of the CTC-SGP is clearly smaller than for all other approaches with low gains. The high-gain approach (HG-PD) has similar RMSE but multiple undesired properties and, therefore, should not be directly compared.



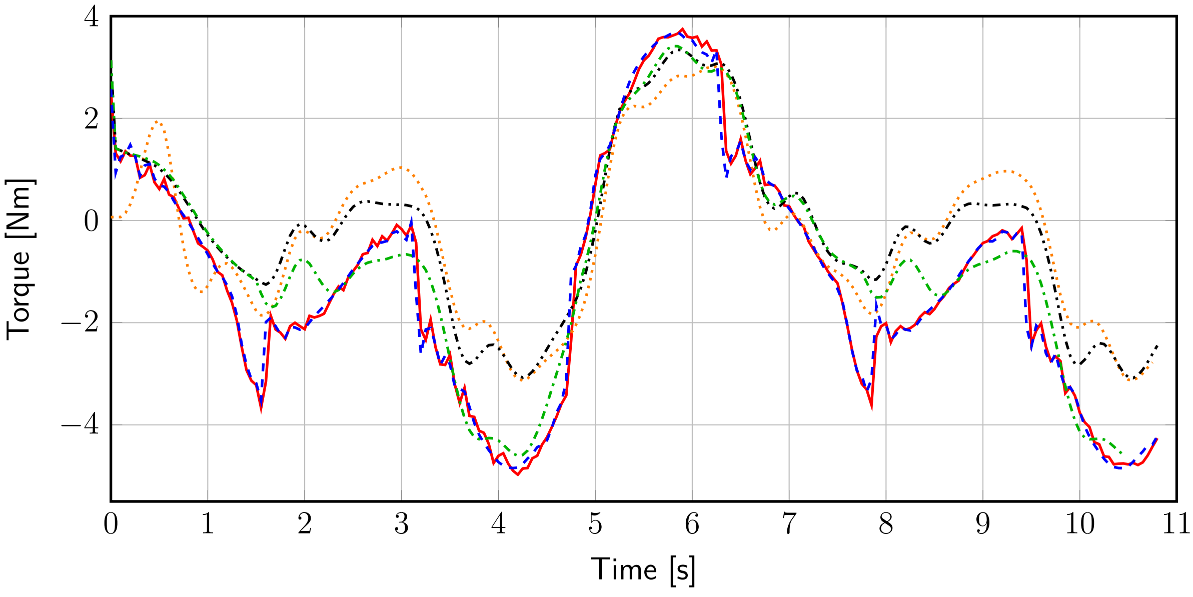


Figure 5.8: Applied torques on first joint (top) and second joint (bottom) for various state-of-the-art controllers. Our CTC-SGP approach (blue dashed) generates similar torques as the high-gain controller (red) but without the unfavorable properties of high feedback gains.

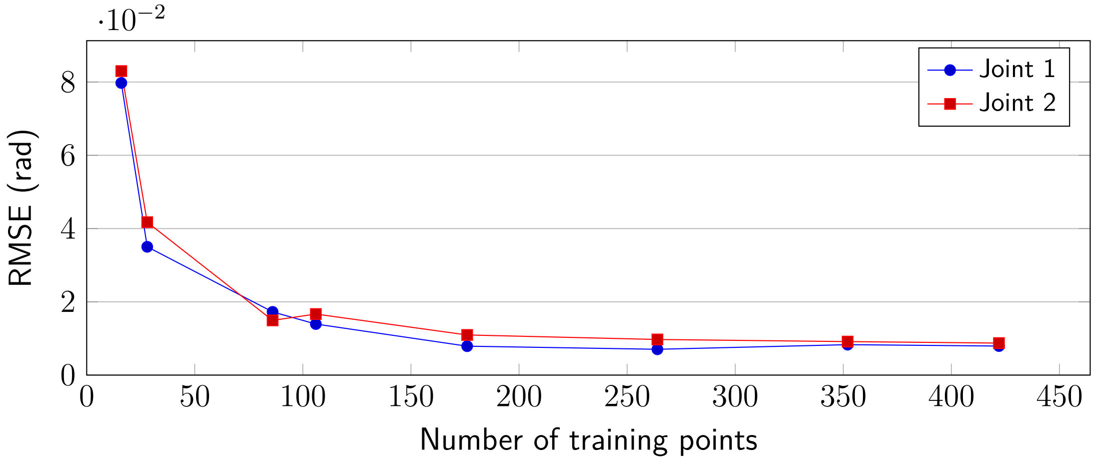


Figure 5.9: The learning curve of the CTC-SGP with increasing number of training points.

5.3 Confidence based Tracking Control

In the previous section, we need Assumptions 5.3 to 5.5 to prove the boundedness of the tracking error. Thus, the approach was restricted to a limited class of Euler-Lagrange systems where the unknown dynamics must be part of the inertia, Corilis or gravity term. In addition, we sparsely use the information about the uncertainty of the GP model. In this section, we develop a computed torque control law with GPR based feed-forward compensation (CTC-GPR) with relaxed assumptions. The proposed control law uses the mean of the GPR to compensate the unknown dynamics and the model confidence to adapt the feedback gains. The derived method guarantees that the tracking error is ultimately bounded within a ball with a specific radius and a given probability. For this purpose, we employ the error bounds of the GP model as presented in Theorem 2.3.

5.3.1 Model Error

For the computation of the model error, we assume the following for the kernel function of the GP model.

Assumption 5.6. *The kernel functions $k^1, \dots, k^{n_q}$ of a multi-output GP model* (2.13) *are chosen such that the functions $\check{\tau}_1, \dots, \check{\tau}_{n_q}$ given by* (5.5) *have a bounded RKHS norm on any compact set $\Omega \subset \mathbb{R}^{3n_q}$, i.e., $\|\check{\tau}_i\|_{k^i} < \infty$ for all $i = 1, \dots, n_q$.*

Remark 5.5. *Note that there are no further assumptions on the system and the unknown dynamics $\boldsymbol{f_u}$ of the EL system* (5.2) *in contrast to the approach in Section 5.2. Thus, any system, which can be written as* (5.2)*, can be considered as long as Assumption 5.6 holds.*

Remark 5.6. *The RKHS norm of a function is a smoothness measure relative to a kernel function k that is uniquely connected with this RKHS. In particular, the norm is a Lipschitz constant with respect to the metric of the used kernel function. For more details see Section 2.1.5.*

Assumption 5.6 requires that the kernel function must be selected in such a way that the residual $\check{\boldsymbol{\tau}}(\boldsymbol{q}_q)$ is an element of the associated RKHS. This sounds paradoxical since the residual is unknown. However, there exist universal kernel functions, see Lemma 2.1, which can approximate any continuous function arbitrarily precisely on a compact set. Therefore, any continuous residual dynamics can be approximated by a universal kernel function, i.e., this assumption is not restrictive. An upper bound for the distance between the mean prediction of the GPR and the actual function is presented in Theorem 2.3. In the following lemma, we extend this bound to multidimensional functions.

Lemma 5.1. *Consider an Euler-Lagrange system* (5.2) *and a trained GP model satisfying Assumption 5.6. Let* $\bar{\Delta}(\delta_m)$ *be a positive constant and* $\Omega \subset \mathbb{R}^{3n_q}$ *a compact set. Then, the model error is bounded by*

$$P\Big\{\Big\|\boldsymbol{\mu}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_q,\mathcal{D}) - \check{\boldsymbol{\tau}}(\boldsymbol{q}_q)\Big\| \leq \bar{\Delta}(\delta_m)\Big\} \geq \delta_m \in (0,1) \tag{5.22}$$

with $\Big\|\boldsymbol{\beta}^\top \Sigma^{\frac{1}{2}}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_q,\mathcal{D})\Big\| \leq \bar{\Delta}(\delta_m)$ *for all* $\boldsymbol{q}_q = [\ddot{\boldsymbol{q}};\dot{\boldsymbol{q}};\boldsymbol{q}] \in \Omega$*. The constants* $\boldsymbol{\beta},\boldsymbol{\gamma} \in \mathbb{R}^{n_q}$ *are given by*

$$\begin{aligned}
\beta_j &= \sqrt{2\|\check{\tau}_j\|_k^2 + 300\gamma_j \ln^3\left(\frac{n_{\mathcal{D}}+1}{1-\delta_m^{1/n_q}}\right)}\\
\gamma_j &= \max_{\boldsymbol{q}_q^{\{1\}},\ldots,\boldsymbol{q}_q^{\{n_{\mathcal{D}}+1\}}\in\Omega} \frac{1}{2}\log|I_{n_z} + \sigma_{n,j}^{-2}K(\boldsymbol{z},\boldsymbol{z}')|, \quad \boldsymbol{z},\boldsymbol{z}' \in \left\{\boldsymbol{q}_q^{\{1\}},\ldots,\boldsymbol{q}_q^{\{n_{\mathcal{D}}+1\}}\right\}
\end{aligned}$$

for all $j \in \{1,\ldots,n_q\}$*.*

Proof. The bound (5.22) is a generalization of Theorem 2.3,which concerns the one dimensional case. In the multidimensional case, Lemma 5.1, we use a GP for each dimension of $\check{\boldsymbol{\tau}}(\boldsymbol{q}_q)$ as shown in (2.15). For the calculation of (5.22), let two sets Π_A,Π_B be

$$\begin{aligned}
\Pi_A &= \Big\{\forall\boldsymbol{q}_q \in D, |\,\mu(\check{\tau}_j|\boldsymbol{q}_q,\mathcal{D}) - \check{\tau}_j(\boldsymbol{q}_q)| \leq \beta_j \operatorname{var}^{\frac{1}{2}}(\check{\tau}_j|\boldsymbol{q}_q,\mathcal{D})\Big\}\\
\Pi_B &= \Big\{\forall\boldsymbol{q}_q \in D, \Big\|\boldsymbol{\mu}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_q,\mathcal{D}) - \check{\boldsymbol{\tau}}(\boldsymbol{q}_q)\Big\| \leq \Big\|\boldsymbol{\beta}^\top \Sigma^{\frac{1}{2}}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_q,\mathcal{D})\Big\|\Big\}
\end{aligned} \tag{5.23}$$

with the multidimensional extension $\boldsymbol{\beta},\boldsymbol{\gamma} \in \mathbb{R}^{n_q}$, given by

$$\begin{aligned}
\beta_j &= \sqrt{2\|\check{\tau}_j\|_k^2 + 300\gamma_j \ln^3\left(\frac{n_{\mathcal{D}}+1}{1-\delta_m^{1/n_q}}\right)} \qquad (5.24)\\
\gamma_j &= \max_{\boldsymbol{q}_q^{\{1\}},\ldots,\boldsymbol{q}_q^{\{n_{\mathcal{D}}+1\}}\in\Omega} \frac{1}{2}\log|I_{n_z} + \sigma_j^{-2}K(\boldsymbol{z},\boldsymbol{z}')|, \quad \boldsymbol{z},\boldsymbol{z}' \in \left\{\boldsymbol{q}_q^{\{1\}},\ldots,\boldsymbol{q}_q^{\{n_{\mathcal{D}}+1\}}\right\}, j \in \{1,\ldots,n_q\}.
\end{aligned}$$

Due to $\check{\boldsymbol{\tau}}$ being uncorrelated according to (2.15), the conditional probability for the set Π_A is lower bounded by $\mathrm{P}\{\Pi_A\} \geq \delta_m$, following (2.38). With the monotony property of the probability measure P and the subset $\Pi_A \subseteq \Pi_B$, (5.22) provides an upper bound for the norm of the model error with a probability of at least $\delta_m \in (0,1)$. □

Remark 5.7. *If Assumption 5.6 is not fulfilled, due to the wrong choice of kernel function or hyperparameters, for many common kernel functions the model error is still bounded on a compact set [BH16b]. However, this upper bounds for the model error may be significantly looser. A tighter bound can be achieved by the approach presented in Section 4.2.*

The information capacity $\boldsymbol{\gamma}$ has a sub-linear dependency on the number of training points for many commonly used kernel functions [Sri+12]. Therefore, even though the values of the elements of $\boldsymbol{\beta}$ are increasing with the number of training data, it is possible to approximate the true function $\check{\boldsymbol{\tau}}(\boldsymbol{q}_q)$ arbitrarily exactly [Ber+16]. The result is an upper bound for the model error, as described in Lemma 5.1. The stochastic nature of this bound is due to the fact that just a finite number of noisy training points are available. Since the model is used for a feed-forward compensation of the unknown dynamics of the system, the model error directly effects the tracking error. More details about this relation is introduced in the next section.

5.3.2 Control Law

Before the control law is proposed, the following assumption of the confidence based feedback gain functions K_d and K_p is introduced.

Assumption 5.7. *Let the functions* $\Sigma_d : \mathbb{R}^{n_q} \times \mathbb{R}^{n_q} \to \mathbb{R}^{n_q \times n_q}$ *and* $\Sigma_p : \mathbb{R}^{n_q} \to \mathbb{R}^{n_q \times n_q}$ *be the marginal variances of the GP model* (5.4)*. These marginal variances are defined analogously to* (2.10) *by* $\Sigma_d(\dot{\boldsymbol{q}}, \boldsymbol{q}) := \Sigma(\check{\boldsymbol{\tau}}|\dot{\boldsymbol{q}}, \boldsymbol{q}, \mathcal{D})$ *and* $\Sigma_p(\boldsymbol{q}) := \operatorname{diag}(\operatorname{var}_1(\check{\boldsymbol{\tau}}|q_1, \mathcal{D}), \dots, \operatorname{var}_{n_q}(\check{\boldsymbol{\tau}}|q_n, \mathcal{D}))$*, respectively. Let* $K_d, K_p \colon \mathbb{R}^{n_q \times n_q} \to \mathbb{R}^{n_q \times n_q}$ *be symmetric matrix functions such that*

$$K_p(\Sigma_p) = \operatorname{diag}([K_p]_{1,1}([\Sigma_p]_{1,1}), \dots, [K_p]_{n_q,n_q}([\Sigma_p]_{n_q,n_q})) + K_c \tag{5.25}$$

with $K_c \in \mathbb{R}^{n_q \times n_q}$*. Additionally, let the compositions* $(K_d \circ \Sigma_d)$*,* $(K_p \circ \Sigma_p)$ *be continuous and bounded by*

$$\underline{k}_d \|\boldsymbol{z}\|^2 \le \boldsymbol{z}^\top K_d(\Sigma_d(\dot{\boldsymbol{q}}, \boldsymbol{q}))\boldsymbol{z} \le \overline{k}_d \|\boldsymbol{z}\|^2 \tag{5.26}$$

$$\underline{k}_p \|\boldsymbol{z}\|^2 \le \boldsymbol{z}^\top K_p(\Sigma_p(\boldsymbol{q}))\boldsymbol{z} \le \overline{k}_p \|\boldsymbol{z}\|^2, \tag{5.27}$$

for all $\dot{\boldsymbol{q}}, \boldsymbol{q}, \boldsymbol{z} \in \mathbb{R}^{n_q}$ *with* $\underline{k}_p, \overline{k}_p, \underline{k}_d, \overline{k}_d \in \mathbb{R}_{>0}$*.*

Remark 5.8. *Loosely speaking, the feedback gains depend on the posterior variance of the GP model* (5.4) *to achieve confidence based feedback. We use the marginal variance such that the function* K_p *implicitly depends exclusively on* $\boldsymbol{q}$ *and* K_d *on* $\boldsymbol{q}, \dot{\boldsymbol{q}}$*, which is a common approach for variable feedback gains [RWH04].*

Example 5.1. The matrix function K_p and K_d should increase with increasing variance to follow the idea that higher model uncertainty leads to higher feedback gains. Given the kernels of the multi-output GP model are bounded, a possible choice for the feedback gain functions is

$$K_p(\Sigma_p(\boldsymbol{q})) = c_1 \operatorname{diag}(\operatorname{var}(\check{\boldsymbol{\tau}}|q_1, \mathcal{D}), \dots, \operatorname{var}(\check{\boldsymbol{\tau}}|q_n, \mathcal{D})) + \underline{k}_p I_{n_q} \tag{5.28}$$

$$K_d(\Sigma_d(\dot{\boldsymbol{q}}, \boldsymbol{q})) = c_2 \, \Sigma_d(\dot{\boldsymbol{q}}, \boldsymbol{q}) + \underline{k}_d I_{n_q}. \tag{5.29}$$

The matrix functions K_p and K_d fulfill Assumption 5.7 since the variance is positive definite and bounded. The variables $c_1, c_2 \in \mathbb{R}_{>0}$ and $\underline{k}_p, \underline{k}_p \in \mathbb{R}_{>0}$ are design parameters, which define the highest and lowest feedback gains.

The next theorem introduces the confidence based control law with guaranteed boundedness of the tracking error.

Theorem 5.2 (CTC-GPR)**.** *Consider the Euler-Lagrange system* (5.2) *and the GP model* (5.4) *satisfying Assumption 5.3. Let* $\boldsymbol{e} = \boldsymbol{q} - \boldsymbol{q_d}$ *be the tracking error and* K_d, K_p *two matrix functions satisfying Assumptions 5.2 and 5.7, respectively. Then, the control law*

$$\boldsymbol{u}(t_c) = \hat{H}(\boldsymbol{q})\ddot{\boldsymbol{q}}_{\boldsymbol{d}} + \hat{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_{\boldsymbol{d}} + \hat{\boldsymbol{g}}(\boldsymbol{q}) + \boldsymbol{\mu}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_q, \mathcal{D}) - K_d(\Sigma_d)\dot{\boldsymbol{e}} - K_p(\Sigma_p)\boldsymbol{e} \tag{5.30}$$

guarantees that the tracking error is uniformly ultimately bounded and exponentially convergent to a ball with

$$P\left\{\|\dot{\boldsymbol{e}}(t_c), \boldsymbol{e}(t_c)\| \leq r(\delta_m), \forall t_c \geq t_0 + t_1(r_0)\right\} \geq \delta_m \in (0,1). \tag{5.31}$$

on a compact set Ω *for any* $\left\|\dot{\boldsymbol{e}}^\top(t_0), \boldsymbol{e}^\top(t_0)\right\| < r_0$ *with* $t_0, t_1(r_0), r_0, r(\delta_m) \in \mathbb{R}_{>0}$*.*

In preparation of the proof of this theorem, we provide a series of results on a suitable Lyapunov candidate.

Lemma 5.2. *There exists an* $\varepsilon > 0$ *such that*

$$V = \frac{1}{2}\dot{\boldsymbol{e}}^\top \hat{H}(\boldsymbol{q})\dot{\boldsymbol{e}} + \int_0^{\boldsymbol{e}} \boldsymbol{z}^\top K_p(\Sigma_p(\boldsymbol{z} + \boldsymbol{q_d}))d\boldsymbol{z} + \varepsilon \boldsymbol{e}^\top \hat{H}(\boldsymbol{q})\dot{\boldsymbol{e}} \tag{5.32}$$

is a radially unbounded Lyapunov function.

Proof. To ensure that the Lyapunov candidate is positive definite, the domain of the integral in (5.32) is analyzed. The integral is lower bounded by

$$\begin{aligned}\int_0^{\boldsymbol{e}} \boldsymbol{z}^\top K_p(\Sigma_p)d\boldsymbol{z} &\geq \int_0^{\boldsymbol{e}} \boldsymbol{z}^\top I_{n_q} \min_{i=\{1,\dots,n_q\}} \lambda_i\left(K_p(\Sigma_p)\right)d\boldsymbol{z} \\ &\geq \frac{1}{2}\boldsymbol{e}^\top\boldsymbol{e} \min_{\boldsymbol{q}\in\mathbb{R}^{n_q}, i=\{1,\dots,n_q\}} \lambda_i\left(K_p(\Sigma_p(\boldsymbol{q}))\right) \geq \frac{1}{2}\underline{k}_p\|\boldsymbol{e}\|^2,\end{aligned} \tag{5.33}$$

where λ_i denote the eigenvalues of the matrix $K_p(\cdot)$ and I_{n_q} the identity matrix. An upper quadratic bound can be found in an analogous way using the maximum eigenvalue of $K_p(\cdot)$. Since the integral is lower bounded and $\hat{H}(\boldsymbol{q})$ is always positive definite, the parameter ε can be chosen sufficiently small to obtain a positive definite and radially unbounded Lyapunov function. The valid interval for ε can be determined by the lower bound of the Lyapunov function (5.32)

$$V(\dot{\boldsymbol{e}}, \boldsymbol{e}) \geq \frac{1}{2}\underline{h}\|\dot{\boldsymbol{e}}\|^2 + \frac{1}{2}\underline{k}_p\|\boldsymbol{e}\|^2 - \frac{1}{2}\varepsilon\overline{h}\left(\|\dot{\boldsymbol{e}}\|^2 + \|\boldsymbol{e}\|^2\right). \tag{5.34}$$

which is positive for $0 < \varepsilon < \min\left\{\underline{k}_p/\overline{h}, \underline{h}/\overline{h}\right\}$. The variables $\underline{k}_p$ and $\overline{h}, \underline{h}$ are defined in Assumption 5.7 and Property 5.2, respectively. □

In the next step, we derive an upper bound for the time derivative of the Lyapunov function.

Lemma 5.3. *Given Assumptions 5.3 and 5.7, consider the Lyapunov function* (5.32) *and the system* (5.2) *with the control law* (5.30). *The derivative of the Lyapunov function is upper bounded with probability* $\delta_m \in (0,1)$ *by*

$$P\Big\{\dot{V} \leq -\frac{3}{4}w_1\|\dot{\boldsymbol{e}}\|^2 - \frac{3}{4}\varepsilon w_2\|\boldsymbol{e}\| + \varepsilon\overline{c}_C\|\dot{\boldsymbol{e}}\|^2\|\boldsymbol{e}\| + \frac{\bar{\Delta}(\delta_m)^2}{w_1} + \varepsilon\frac{\bar{\Delta}(\delta_m)^2}{w_2}\Big\} \geq \delta_m, \text{ where} \tag{5.35}$$

$$w_1 := -\varepsilon\overline{h} + \underline{k}_d - \frac{\varepsilon c_{pp}}{2}(\overline{c}_C\bar{\dot{q}}_d + \overline{k}_d)$$

$$w_2 := \underline{k}_p\frac{\varepsilon_2}{1+\varepsilon_2} \tag{5.36}$$

$$c_{pp} = (1+\varepsilon_2)\frac{\overline{c}_C\bar{\dot{q}}_d + \overline{k}_d}{2\underline{k}_p}$$

$$0 < \varepsilon < \min\left\{\frac{\underline{k}_p}{\overline{h}}, \frac{\underline{h}}{\overline{h}}, \frac{2\underline{k}_d}{2\overline{h} + c_{pp}(\overline{c}_C\bar{\dot{q}}_d + \overline{k}_d)}\right\} \tag{5.37}$$

with maximum model error $\bar{\Delta}(\delta_m)$ *given by* (5.22), *the helper variables* $w_1, w_2, \varepsilon, \varepsilon_2 \in \mathbb{R}_{>0}$ *and for all* $\boldsymbol{q}_q = [\ddot{\boldsymbol{q}}; \dot{\boldsymbol{q}}; \boldsymbol{q}] \in \Omega$.

The bounds $\underline{k}_p, \overline{k}_p, \underline{k}_d, \overline{k}_d$ for the feedback gain matrices are given by Assumption 5.7. The bounds $\underline{h}, \overline{h}$ for the estimated system matrices are given by Property 5.2. The maximum model error $\bar{\Delta}(\delta_m)$ is provided by Lemma 5.1 and $\bar{\dot{q}}_d$ denotes the maximum desired generalized velocity.

Proof. The time derivative of the Lyapunov function (5.32) is expressed by

$$\dot{V} = \begin{bmatrix} \dot{\boldsymbol{e}}^\top\hat{H} + \varepsilon\boldsymbol{e}^\top\hat{H} \\ \boldsymbol{e}^\top K_p(\Sigma_p) + \frac{1}{2}\dot{\boldsymbol{e}}^\top\dot{\hat{H}} + \varepsilon(\boldsymbol{e}^\top\dot{\hat{H}} + \dot{\boldsymbol{e}}^\top\hat{H}) \end{bmatrix}^\top \begin{bmatrix} \ddot{\boldsymbol{e}} \\ \dot{\boldsymbol{e}} \end{bmatrix}, \tag{5.38}$$

using the symmetry of $\hat{H}$ and the solution of the integral

$$\frac{\partial}{\partial\boldsymbol{e}}\int_{\boldsymbol{0}}^{\boldsymbol{e}} \boldsymbol{z}^\top K_p\left(\Sigma_p(\boldsymbol{z} + \boldsymbol{q_d})\right)d\boldsymbol{z} = \boldsymbol{e}^\top K_p(\Sigma_p(\boldsymbol{q})). \tag{5.39}$$

For the computation of $\ddot{\boldsymbol{e}}$, the closed-loop dynamics of the EL system (5.2) with input (5.30) is rewritten to

$$\ddot{\boldsymbol{q}}_{\boldsymbol{d}} = \hat{H}^{-1}\left(H\ddot{\boldsymbol{q}} + C\dot{\boldsymbol{q}} + \boldsymbol{g} - \boldsymbol{f_u}(\boldsymbol{q}_q) - \hat{C}\dot{\boldsymbol{q}}_{\boldsymbol{d}} - \hat{\boldsymbol{g}} - \boldsymbol{\mu}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_q, \mathcal{D}) + K_d(\Sigma_d)\dot{\boldsymbol{e}} + K_p(\Sigma_p)\boldsymbol{e}\right). \tag{5.40}$$

With $\hat{C}\dot{\boldsymbol{q}}_{\boldsymbol{d}} = \hat{C}\dot{\boldsymbol{q}} - \hat{C}\dot{\boldsymbol{e}}$ and the residual dynamics (5.5), the closed-loop dynamics (5.40) is rewritten to

$$\ddot{\boldsymbol{e}} = \ddot{\boldsymbol{q}} - \ddot{\boldsymbol{q}}_{\boldsymbol{d}} = \hat{H}^{-1}\Big(\boldsymbol{\mu}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_q, \mathcal{D}) - \check{\boldsymbol{\tau}}(\boldsymbol{q}_q) - K_d(\Sigma_d)\dot{\boldsymbol{e}} - K_p(\Sigma_p)\boldsymbol{e} - \hat{C}\dot{\boldsymbol{e}}\Big). \tag{5.41}$$

Using (5.41) and the Lyapunov function (5.32), the time derivative of the Lyapunov function (5.38) is expressed by

$$\dot{V} = \begin{bmatrix}\dot{\boldsymbol{e}}\\ \boldsymbol{e}\end{bmatrix}^\top \underbrace{\begin{bmatrix} \underbrace{-K_d(\Sigma_d)+\varepsilon\hat{H}}_{M_{11}} & \underbrace{\frac{\varepsilon}{2}(-K_d^\top(\Sigma_d)+\hat{C})}_{M_{12}} \\ \underbrace{\frac{\varepsilon}{2}(-K_d(\Sigma_d)+\hat{C}^\top)}_{M_{21}=M_{12}} & \underbrace{-\varepsilon K_p(\Sigma_p)}_{M_{22}}\end{bmatrix}}_{M\in\mathbb{R}^{2n_q\times 2n_q}} \begin{bmatrix}\dot{\boldsymbol{e}}\\ \boldsymbol{e}\end{bmatrix} + (\dot{\boldsymbol{e}}+\varepsilon\boldsymbol{e})^\top(\boldsymbol{\mu}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_q,\mathcal{D}) - \check{\boldsymbol{\tau}}(\boldsymbol{q}_q)), \tag{5.42}$$

where the skew-symmetry of $\dot{\hat{H}} - 2\hat{C}$ is exploited. For the following analysis, we compute bounds for the elements of the matrix M to bound the drift of the Lyapunov function, based on [RWH04]. The matrix $M_{11} \in \mathbb{R}^{n_q\times n_q}$ is negative definite for sufficiently small $\varepsilon > 0$ and upper bounded by $\dot{\boldsymbol{e}}^\top M_{11}\dot{\boldsymbol{e}} \leq (-\underline{k}_d + \varepsilon\overline{h})\|\dot{\boldsymbol{e}}\|^2$. Analogously, the submatrix $M_{22} \in \mathbb{R}^{n_q\times n_q}$ is negative definite with $\boldsymbol{e}^\top M_{22}\boldsymbol{e} \leq -\varepsilon\underline{k}_p\|\boldsymbol{e}\|^2$. With Assumption 5.2 and Property 5.2, the submatrix $M_{12} \in \mathbb{R}^{n_q\times n_q}$ is upper bounded by

$$\boldsymbol{e}^\top M_{12}\dot{\boldsymbol{e}} \leq \varepsilon\left(\overline{c}_C\|\dot{\boldsymbol{e}}\| + \overline{c}_C\bar{\dot{q}}_d + \overline{k}_d\right)\|\dot{\boldsymbol{e}}\|\|\boldsymbol{e}\|. \tag{5.43}$$

With Lemma 5.1, the overall upper bound for the time derivative of the Lyapunov function is given by

$$\begin{aligned}\mathrm{P}\Big\{\dot{V} \leq{}& (\varepsilon\overline{h} - \underline{k}_d)\|\dot{\boldsymbol{e}}\|^2 - \varepsilon\underline{k}_p\|\boldsymbol{e}\|^2 + \varepsilon\left(\overline{c}_C\|\dot{\boldsymbol{e}}\| + \overline{c}_C\bar{\dot{q}}_d + \overline{k}_d\right)\|\dot{\boldsymbol{e}}\|\|\boldsymbol{e}\| \\ &+ (\|\dot{\boldsymbol{e}}\| + \varepsilon\|\boldsymbol{e}\|)\Big\|\boldsymbol{\beta}^\top\Sigma^{\frac{1}{2}}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_q,\mathcal{D})\Big\|\Big\} \geq \delta_m.\end{aligned} \tag{5.44}$$

For the next step, we consider the Peter-Paul inequality given by

$$\|\dot{\boldsymbol{e}}\|\|\boldsymbol{e}\| \leq \frac{1}{2}\left(c_{\mathrm{pp}}\|\dot{\boldsymbol{e}}\|^2 + \|\boldsymbol{e}\|^2/c_{\mathrm{pp}}\right) \tag{5.45}$$

that holds for all $\dot{\boldsymbol{e}}, \boldsymbol{e} \in \mathbb{R}^{n_q}$ and $c_{\mathrm{pp}} \in \mathbb{R}_{\geq 0}$. With the Peter-Paul inequality, (5.44) can be rewritten as

$$\begin{aligned}\mathrm{P}\Big\{\dot{V} \leq{}& (\varepsilon\overline{h} - \underline{k}_d)\|\dot{\boldsymbol{e}}\|^2 - \varepsilon\underline{k}_p\|\boldsymbol{e}\|^2 + \frac{\varepsilon}{2}\left(\overline{c}_C\bar{\dot{q}}_d + \overline{k}_d\right)\left(c_{\mathrm{pp}}\|\dot{\boldsymbol{e}}\|^2 + \frac{\|\boldsymbol{e}\|^2}{c_{\mathrm{pp}}}\right) + \varepsilon\overline{c}_C\|\dot{\boldsymbol{e}}\|^2\|\boldsymbol{e}\| \\ &+ (\|\dot{\boldsymbol{e}}\| + \varepsilon\|\boldsymbol{e}\|)\Big\|\boldsymbol{\beta}^\top\Sigma^{\frac{1}{2}}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_q,\mathcal{D}))\Big\|\Big\} \geq \delta_m \text{ with}\end{aligned} \tag{5.46}$$

$$c_{\mathrm{pp}} = (1+\varepsilon_2)\frac{\overline{c}_C\bar{\dot{q}}_d + \overline{k}_d}{2\underline{k}_p},\ \varepsilon_2 \in \mathbb{R}_{>0}. \tag{5.47}$$

This choice of c_{pp} guarantees that the factors of the quadratic parts in (5.47) are still negative:

$$\begin{aligned}\mathrm{P}\Big\{\dot{V} \leq{}& \left(\varepsilon\overline{h} - \underline{k}_d + \frac{\varepsilon c_{\mathrm{pp}}}{2}(\overline{c}_C\bar{\dot{q}}_d + \overline{k}_d)\right)\|\dot{\boldsymbol{e}}\|^2 - \varepsilon\underline{k}_p\frac{\varepsilon_2}{1+\varepsilon_2}\|\boldsymbol{e}\|^2 + \varepsilon\overline{c}_C\|\dot{\boldsymbol{e}}\|^2\|\boldsymbol{e}\| \\ &+ (\|\dot{\boldsymbol{e}}\| + \varepsilon\|\boldsymbol{e}\|)\Big\|\boldsymbol{\beta}^\top\Sigma^{\frac{1}{2}}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_q,\mathcal{D})\Big\|\Big\} \geq \delta_m\end{aligned} \tag{5.48}$$

Since the kernel functions are continous and bounded on the compact set Ω, the posterior variance $\Sigma(\check{\boldsymbol{\tau}}|\boldsymbol{q}_q,\mathcal{D})$ is bounded, for more details see Theorem 3.3. Thus, there exists an upper bound $\bar{\Delta}(\delta_m)$ for the model error. Applying the inequality $w_1\|\boldsymbol{z}\| \leq w_1^2/w_2+w_2\|\boldsymbol{z}\|^2/4$, that holds $\forall \boldsymbol{z} \in \mathbb{R}^{n_q}$ and $w_1, w_2 \in \mathbb{R}_{\geq 0}$, the model error in (5.48) can be bounded by a quadratic function, which results in (5.35). The restrictions for ε must be extended to (5.37) to ensure that the variables $w_1, w_2 \in \mathbb{R}_{>0}$ are positive. □

The results of Lemmas 5.2 and 5.3 were presented in preparation of the proof of Theorem 5.2, which we provide in the following.

Proof. Proof of Theorem 5.2 According to [RWH04, Theorem 1], there exist a $c_3, c_4 \in \mathbb{R}_{\geq 0}$ for the time derivative of the Lyapunov function (5.44), given by Lemmas 5.2 and 5.3, such that

$$\mathrm{P}\left\{\dot{V}(\dot{\boldsymbol{e}}, \boldsymbol{e}) \leq -c_3 V(\dot{\boldsymbol{e}}, \boldsymbol{e}) + c_4\right\} \geq \delta_m. \tag{5.49}$$

Consequently, using [Cor90, Theorem 2.1], the closed-loop is uniformly ultimately bounded and exponentially convergent to a ball with a probability of at least δ_m. Thus,

$$P\left\{\|\dot{\boldsymbol{e}}(t_c), \boldsymbol{e}(t_c)\| \leq r(\delta_m), \forall t_c \geq t_0 + t_1(r_0)\right\} \geq \delta_m \tag{5.50}$$

holds on a compact set Ω for any $\left\|\dot{\boldsymbol{e}}^\top(t_0), \boldsymbol{e}^\top(t_0)\right\| < r_0$ with $t_0, t_1(r_0), r_0, r(\delta_m) \in \mathbb{R}_{>0}$. Since the desired trajectory is bounded, see Assumption 5.2, it is always possible to find a set Ω such that $\boldsymbol{q}_q \in \Omega$. That concludes the proof. □

In addition to the proof of the existence of a bound for the tracking error by Theorem 5.2, we can quantify the maximum tracking error of the closed-loop.

Proposition 5.1. *Let the allowed set of* ε *in* (5.37) *be restricted to*

$$0 < \varepsilon < \min\left\{\frac{\underline{k}_p}{\overline{h}}, \frac{\underline{h}}{\overline{h}}, \frac{2\underline{k}_d}{2\overline{h} + \frac{2\underline{k}_p c_{pp}^2}{1+\varepsilon_2} + \frac{8}{3}\overline{c}_C\sqrt{\frac{2V(\dot{\boldsymbol{e}}(t_0),\boldsymbol{e}(t_0))}{\underline{k}_p - \varepsilon\overline{h}}}}\right\}.$$

Then, the radius r *of the ball for the tracking error in* (5.31) *is*

$$r = \sqrt{\frac{2c_4}{c_3 \min\left\{\underline{k}_p - \varepsilon\overline{h}, \underline{h} - \varepsilon\overline{h}\right\}}}, \quad \textit{where} \tag{5.51}$$

$$c_3 = \frac{2}{3}\frac{\min\left\{\varepsilon w_2, w_1 - \frac{4}{3}\varepsilon\overline{c}_C\sqrt{\frac{2V(\dot{\boldsymbol{e}}(t_0),\boldsymbol{e}(t_0))}{\underline{k}_p - \varepsilon\overline{h}}}\right\}}{\max\left\{\varepsilon\overline{h} + \overline{k}_p, (1+\varepsilon)\overline{h}\right\}} \tag{5.52}$$

and $c_4 = \bar{\Delta}(\delta_m)^2/w_1 + \varepsilon\bar{\Delta}(\delta_m)^2/w_2$ *with* w_1, w_2 *defined in Lemma 5.3.*

Proof. The proof follows directly from [WB88, Lemma 2.1] and is therefore not repeated here. □

For the sake of completeness, we analyze in the following the pathological case of a perfect GP model.

Corollary 5.3. *If a perfect GP model of the unknown residual dynamics was available, such that the maximum model error* $\bar{\Delta} = 0$*, the tracking error of the closed-loop system* (5.41) *is asymptotically stable on* Ω*.*

Proof. Asymptomatic stability is given, if the drift of the Lyapunov function (5.42) is negative for all $\boldsymbol{q}_q \in \Omega$. This is fulfilled by the negative definiteness of $M \in \mathbb{R}^{2n_q \times 2n_q}$ in (5.42) since the model error $\boldsymbol{\mu}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_q, \mathcal{D}) - \check{\boldsymbol{\tau}}(\boldsymbol{q}_q))$ is zero for all $\boldsymbol{q}_q \in \Omega$. According to Schur's lemma, the matrix M in (5.42) is negative definite if the upper left block of M, given by $M_{11} = -K_d(\Sigma_d) + \varepsilon \hat{H}$, and

$$M' = -\varepsilon K_p(\Sigma_p) + \frac{\varepsilon^2}{4}(K_d(\Sigma_d) - \hat{C}^\top)(K_d(\Sigma_d) - \varepsilon \hat{H})^{-1}(K_d^\top(\Sigma_d) - \hat{C}) \tag{5.53}$$

are negative definite, where $M' \in \mathbb{R}^{n_q \times n_q}$ is the Schur complement. Since $K_d, \hat{H}$, and K_p are positive definite and bounded, ε can be chosen sufficiently small to ensure the negative definiteness of M_{11}. The second summand of the Schur complement M' is quadratic in ε and positive definite, while the first summand is linear in ε and negative. Thus, for every $\boldsymbol{q}, \dot{\boldsymbol{q}} \in \mathbb{R}^{n_q}$, an ε can be found which guarantees the negative definiteness of the Schur complement. Therefore, there exists an $\varepsilon > 0$, so that matrix M is negative definite. □

5.3.3 Design Guidelines

Theorem 5.2 provides an ultimate bound with an adherence probability depending on the gains, the system parameters and the variance of the GP. The radius of the bound depends on the upper bound of the model error $\bar{\Delta}$. Thus, the radius r shrinks, if the upper bound of the GPR variance decreases. The consequence is an improved tracking performance in terms of tracking error and the possibility to decrease the feedback gains, which is beneficial for noise attenuation. The posterior variance of the GP model is correlated to the number and distribution of the training points. It can be decreased, e.g., with a Bayesian optimization approach where the next training point is set to the position of maximum variance. For the commonly used squared exponential kernel, each new training point reduces the posterior variance [Uml+17]. The bounds of the adaptive gains additionally affect the radius of the ball. Increasing the lower bound of K_d shrinks the radius since ε can be arbitrarily small and w_1 depends linearly on $\underline{k}_d$. The influence of K_p depends on the Euler-Lagrange system. Figure 5.10 visualizes how different design goals can be addressed with our proposed control law: i) Computing the performance for specified feedback gain functions, ii) Generating a sufficiently accurate model for a predefined performance, or iii) Design feedback gain functions to obtain a predefined performance.

5.3.4 Numerical Evaluation

In this section, we present examples illustrating the properties of the proposed computed torque control law with GPR (CTC-GPR) and a detailed case study.

Noise attenuation and saturation

In the following numerical evaluation, we show the advantages of the CTC-GPR in comparison to the classical CTC. For this purpose, we consider a one dimensional EL system given

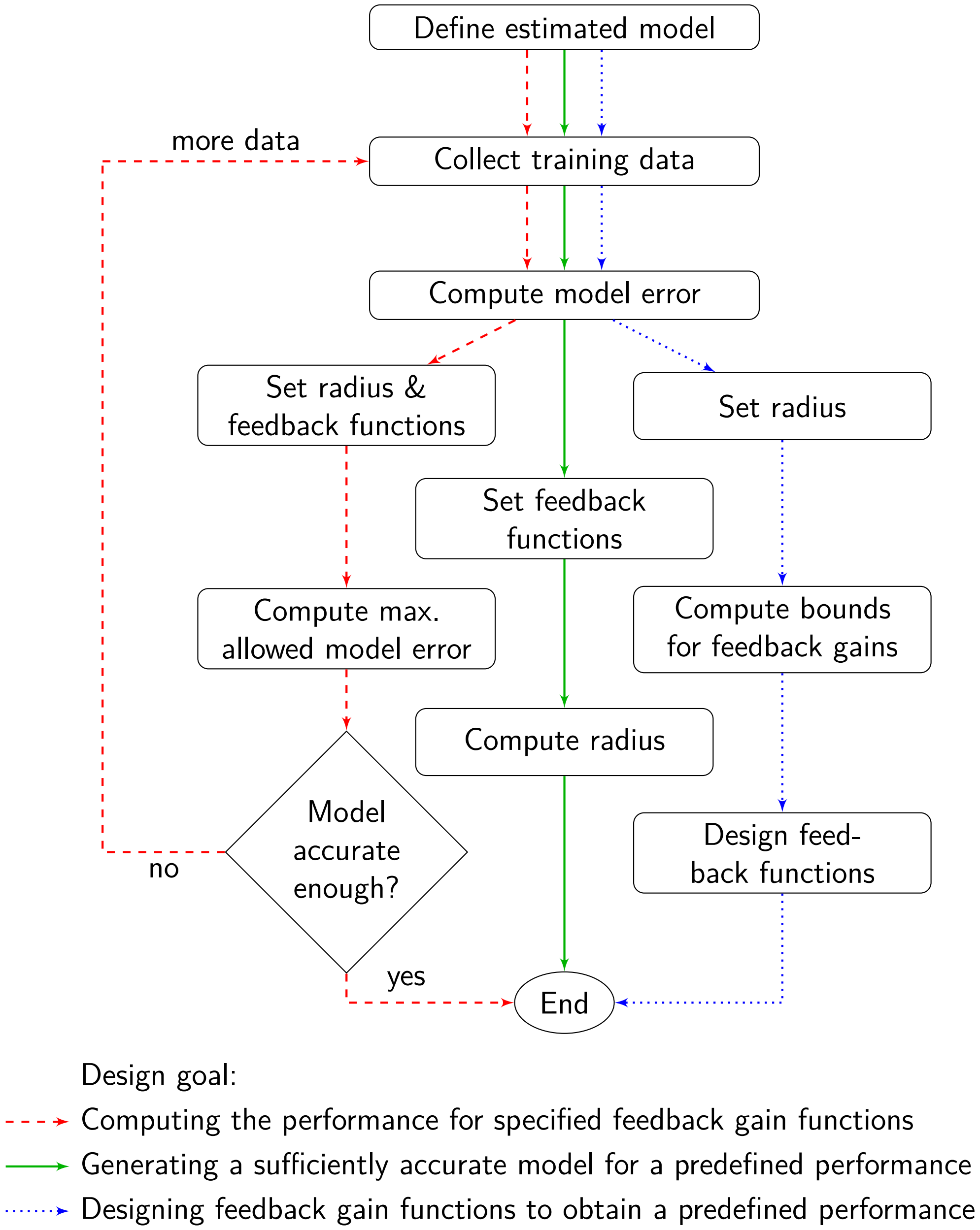

Define estimated model
more data
Collect training data
Compute model error
Set radius & feedback functions
Set radius
Set feedback functions
Compute max. allowed model error
Compute bounds for feedback gains
Compute radius
Model accurate enough?
no
Design feed-back functions
yes
End
Design goal:
Computing the performance for specified feedback gain functions
Generating a sufficiently accurate model for a predefined performance
Designing feedback gain functions to obtain a predefined performance

Figure 5.10: Guidelines for different design goals.

by $\tau = \ddot{q} + \dot{q} + q + f_u(\boldsymbol{q}_q)$ with 30 randomly generated, normally unknown, dynamics

$$f_u(\boldsymbol{q}_q) = \frac{\dot{q}^2 \sin(q-c) - \sin(c)}{\cos(q-c) - 1.1\cos^{-1}(q-c)}, \tag{5.54}$$

where each c is uniformly sampled from the set $[0, 2\pi]$. For the parametric model, we use the estimates $\hat{H} = \hat{C} = \hat{g} = 1$. A total number of 441 training data pairs $\{[\ddot{q}; \dot{q}; q], \check{\tau}\}$ for a GP model with squared exponential kernel function are equally distributed on the set $[0] \times [-1,1] \times [-1,1]$. A conjugate gradient algorithm is used to minimize the log likelihood function to find suitable hyperparameters. The desired trajectory is given by $\boldsymbol{q}_d = \sin(t_c)$ with $t_c \in [0, 2\pi]$ and the initial system value is $q_0 = 0, \dot{q}_0 = 1$. The measurements of $\ddot{q}, \dot{q}, q$ are corrupted by Gaussian noise, with zero mean and a variance of 0.04^2, for training and control. In the simulation, the CTC-GPR and the classical computed torque are compared in terms of the maximum tracking error, the noise attenuation and the maximum control action. The feedback gains of the CTC are $K_p = 100, K_d = 100$, whereas the CTC-GPR is parameterized with

$$K_p(q) = 10 + 100\,\Sigma_p(q),\; K_d(\dot{q}, q) = 10 + 100\,\Sigma_d(\dot{q}, q). \tag{5.55}$$

The results of our proposed CTC-GPR approach in percentage of the classical CTC are shown in Fig. 5.11. We visualize the median (red), the first and third quartile (blue box) and the minimum/maximum (black) for the control of the 30 randomly generated dynamics. The variation of the gains is minimal since the desired trajectory is inside the training area where the variance is low. The maximal tracking error $\max \|\dot{e}(t_c), e(t_c)\|$ of the CTC-GPR is decreased compared to the CTC for all 30 random systems with a median of 61.6%. The CTC-GPR shows remarkably better noise attenuation, as indicated by a higher signal to noise ratio (SNR) of the system trajectory. The SNR is computed as the ratio of the summed squared magnitude of the state to that of the noise. Also, the maximal control action of the CTC-GPR is reduced in comparison to the CTC, due to the lower feedback gains of the CTC-GPR, which can prevent actuator saturation.

Case study

In this case study, we contrast the approaches of the CTC-GPR and the CTC for a 2-link robotic manipulator, as depicted in Fig. 5.12. We use a CTC as reference for our performance comparison, since most of the robotic control schemes can be considered as special cases of a computed torque controller. We assume point masses for the links of $m_{\text{link},1} = m_{\text{link},2} = 1\,\text{kg}$, which are located in the center of each link. The length of the links is set to $l_{\text{link},1} = l_{\text{link},2} = 1\,\text{m}$. The joints are without mass and not influenced by any friction. Gravity is assumed to be $g = 9.81\,\text{ms}^{-2}$. As estimates, we use $\hat{m}_{\text{link},1} = 0.9\,\text{kg}, \hat{m}_{\text{link},2} = 1.1\,\text{kg}, \hat{l}_{\text{link},1} = 0.9\,\text{m}$, and $\hat{l}_{\text{link},2} = 1.1\,\text{m}$. Following [MLS94, Page 164], the estimated system matrices of the EL system (5.2) are given by

$$\hat{H} = \begin{bmatrix} 1.41 + 1.09\cos(q_2) & 0.61 + 0.54\cos(q_2) \\ 0.61 + 0.54\cos(q_2) & 0.61 \end{bmatrix}$$

$$\hat{C} = \begin{bmatrix} -0.54\sin(q_2)\dot{q}_2 & -0.54\sin(q_2)(\dot{q}_1 + \dot{q}_2) \\ 0.54\sin(q_2)\dot{q}_1 & 0 \end{bmatrix}$$

$$\hat{\boldsymbol{g}} = \begin{bmatrix} 9\sin(q_1) + 6.05\sin(q_1 + q_2) \\ 6.05\sin(q_1 + q_2) \end{bmatrix}, \tag{5.56}$$

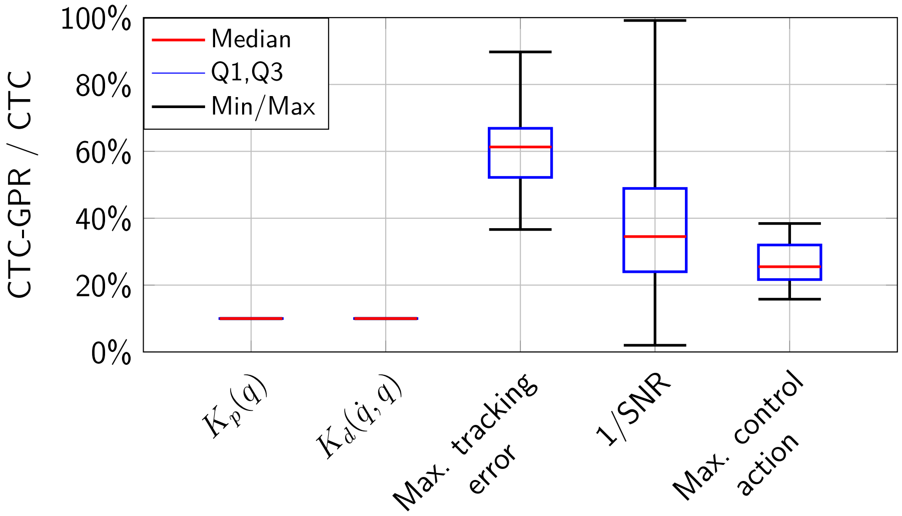


Figure 5.11: Comparison between the CTC and the proposed CTC-GPR for 30 randomly selected systems. The values of the CTC-GPR are given as a percentage of the CTC performance. Our CTC-GPR approach significantly outperforms the standard computed torque controller.

where q_1 and q_2 are the joint angles. The initial angles and velocities are set to $\boldsymbol{q}_0 = [0,1]^\top$ and $\dot{\boldsymbol{q}}_0 = [1,0]^\top$, respectively. The unknown dynamics $\boldsymbol{f_u}(\boldsymbol{q}_q)$ is simulated by an arbitrarily chosen function

$$\boldsymbol{f_u}(\boldsymbol{q}_q) = \begin{bmatrix} \sin(2\dot q_2) + \cos(2q_1) + \ddot q_1 \\ \sin(2\dot q_2) + 2\sin(\dot q_1) \end{bmatrix}. \tag{5.57}$$

A GP model with a squared exponential kernel learns the difference between the estimated model and the true system based on 576 equally distributed training pairs on the domain $\boldsymbol{q}, \ddot{\boldsymbol{q}} \in [0,1]^2$, $\dot{\boldsymbol{q}} \in [-1,1]^2$. The measurements of $\ddot{\boldsymbol{q}}, \dot{\boldsymbol{q}}, \boldsymbol{q}$ are corrupted by Gaussian noise with zero mean and a variance of 0.1^2. The hyperparameters are optimized by means of the likelihood function. The desired trajectory is a sinusoidal function with $\boldsymbol{q}_0 = [0,1]^\top$. In this example, the feedback gains are adapted with

$$\begin{aligned} K_p(\Sigma_p) &= 7I + 400\,\Sigma_p(\boldsymbol{q}) \\ K_d(\Sigma_d) &= 6I + 400\,\Sigma_d(\dot{\boldsymbol{q}}, \boldsymbol{q}). \end{aligned} \tag{5.58}$$

For comparison, we use a classical CTC with $K_{p,s} = K_{d,s} = \operatorname{diag}(10,10)$, which is a trade-off between tracking error and high feedback gains. Additionally, the CTC-GPR is compared to a CTC-GPR with static feedback gains where the values of the static gains are set to the minimum of the variable gains such that the noise attenuation is comparable. The top plot of Fig. 5.13 shows the resulting trajectory of the classical CTC for the first joint (blue dashed) along with the desired trajectory (red dashed). The trajectory of the CTC-GPR with static feedback gains (blue solid) based on the training data (black crosses) significantly improves the tracking error. However, in the area without training data (left half plane), the tracking error is still high due to the poor accuracy of the GP model. The bottom plot of Fig. 5.13

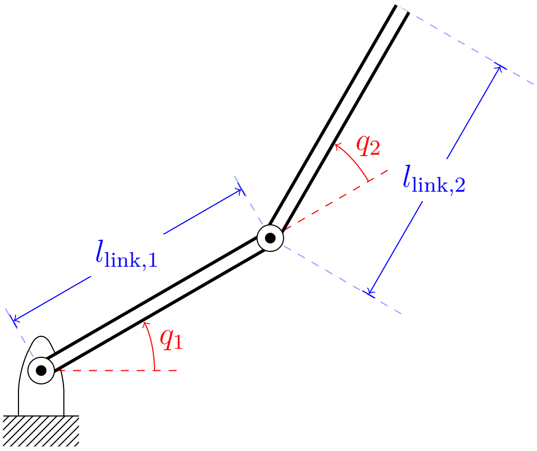


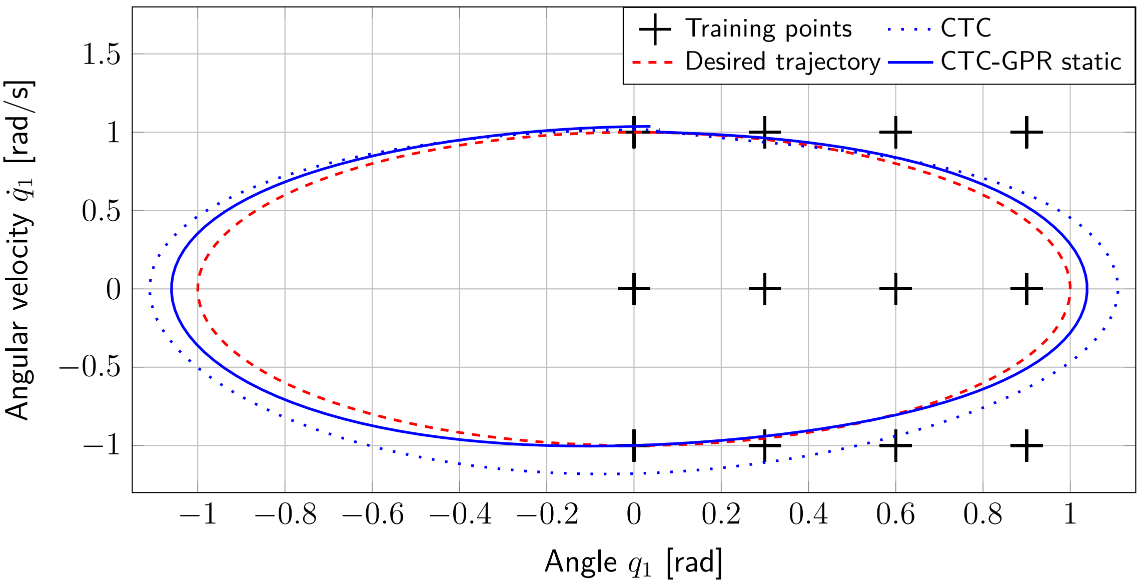
Figure 5.12: Configuration of a 2-link robotic manipulator.

visualizes the trajectory of the CTC-GPR with variable feedback gains (color-coded solid). In the area with training data (right half plane), the feedback gains are low (blue) since the model error is low. If the training data becomes sparse, the model uncertainty is increasing and, thus, the feedback gains are increasing (red) to keep the tracking error low.
An overview about the numerical performance of the CTC-GPR in contrast to the CTC are presented in Table 5.1. Both CTC-GPR approaches show a lower tracking error than the classical CTC. The reason is that the CTC-GPR uses the mean function to compensate the unknown dynamics, such that the feedback gains can be lower in comparison to the CTC. Additionally, the variable CTC-GPR outperforms (bold values) the static CTC-GPR for the absolute position and velocity error because the variable gains are increased as soon as the trajectory leaves the training area. The result is that the tracking error of the variable CTC-GPR is kept low and bounded even for areas where no training data is available. Further benefits of low feedback gains for noise attenuation are presented in Fig. 5.11.

Remark 5.9. *The MATLAB source code of the numerical evaluation and the case study is available at:* `https://github.com/TBeckers/CTC_GPR`

	CTC	Static CTC-GPR	Variable CTC-GPR
$\Vert K_p\Vert$	10	7.01	7.01 - 9.38
$\Vert K_d\Vert$	10	6.06	6.06 - 9.38
$\left\Vert \boldsymbol{e}^\top, \dot{\boldsymbol{e}}^\top \right\Vert_{L^2}$	4.7281	1.8760	**1.5118**
$\max(\Vert \boldsymbol{e}(t_c)\Vert)$	0.2420	0.1066	**0.0819**
$\max(\Vert \dot{\boldsymbol{e}}(t_c)\Vert)$	0.2377	0.1234	**0.1002**

Table 5.1: Comparision between CTC, CTC-GPR with static gains, and CTC-GPR with variable gains. Our proposed CTC-GPR approach outperforms (bold values) the classical CTC.



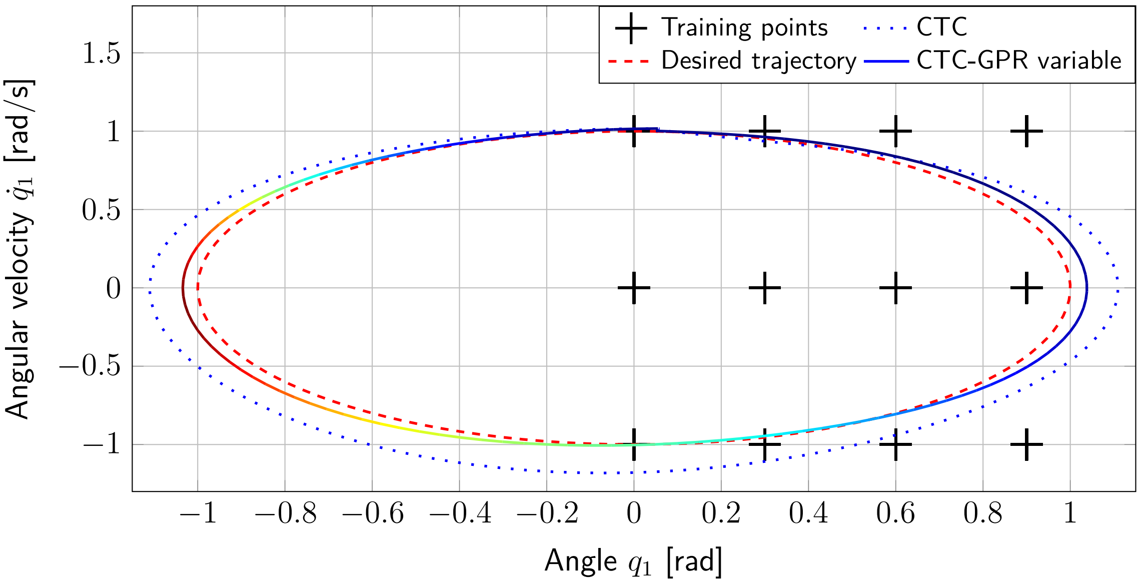


Figure 5.13: Tracking performance for the first joint. Top: A CTC-GPR with static feedback gains in comparision to a classical CTC. Bottom: The color of the CTC-GPR trajectory indicates the norm of the variable feedback gains (red high, blue low).

5.4 Passivation of Euler-Lagrange Systems with Unknown Dynamics

Mechanical and electrical systems, represented by Euler-Lagrange (EL) dynamics, are often interconnected with other systems. Passivity based techniques allow the analysis and synthesis of large and complex systems because of the particular composition properties. For instance, the parallel and feedback interconnection of passive sub-systems induces a passive overall system. The passivity property is also helpful for the interconnection with other systems, which are mostly unknown but assumed to be passive such as in telepresence systems [NBO11], robot manipulation [EH16] or physical human-robot interaction [San+08]. Hence, passive systems possess very useful and beneficial properties which make them so interesting in control theory and also in real-world applications. However, many modern EL systems are not inherently passive, e.g., high-performance aircraft. Thus, to take advantage of the passivity properties, these systems need to be rendered passive by control, which is called *passivation*. However, the dynamics of a system must be known to render it passive. In this section, we propose a control law for the passivation of EL systems with unknown dynamics.

5.4.1 Problem Setting

We consider the EL system (5.2) with output $\boldsymbol{y}_{ex} \in \mathbb{R}^{n_q}$

$$\begin{aligned} \boldsymbol{u}(t_c) &= H(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{f}_{\boldsymbol{u}}(\boldsymbol{q}_q) \\ \boldsymbol{y}_{ex} &= c_{\mathrm{ex}}\boldsymbol{q} + \dot{\boldsymbol{q}} \end{aligned} \tag{5.59}$$

with $\boldsymbol{q}_q = [\ddot{\boldsymbol{q}}; \dot{\boldsymbol{q}}; \boldsymbol{q}]$ and $c_{\mathrm{ex}} \in \mathbb{R}_{>0}$.

Remark 5.10. *The output $\boldsymbol{y}_{ex} \in \mathbb{R}^{n_q}$ is often used in interconnection scenarios of mechanical systems, where it represents a velocity plus scaled position feedback.*

The task is to find an input $\boldsymbol{u}$ such that the system (5.59) becomes passive.

5.4.2 Control Structure

For the passivation of the EL system (5.59), a closed-loop with a GP model and a feedback control law is proposed. The GP model is used as feed-forward compensation of the unknown dynamics so that the drift function of the closed-loop is bounded. Subsequently, a feedback control law is exploited to render the system *strictly semi-passive*, as defined in Definition A.7. Loosely speaking, semi-passive systems behave like passive system outside a ball $\mathcal{B}_r$ with radius $r > 0$, as depicted in Fig. 5.14. For this purpose, the input $\boldsymbol{u}$ of the system (5.59) is decomposed

$$\boldsymbol{u} = \boldsymbol{u}_c + \boldsymbol{u}_{gp} - \boldsymbol{u}_{ex} \tag{5.60}$$

into a prediction of a GP model $\boldsymbol{u}_{gp} \in \mathbb{R}^{n_q}$, a feedback control law $\boldsymbol{u}_c \in \mathbb{R}^{n_q}$, and an additional external input $\boldsymbol{u}_{ex} \in \mathcal{U} \subset \mathbb{R}^{n_q}$. The feedback control law is given by

$$\boldsymbol{u}_c = K_d\dot{\boldsymbol{q}} + K_p\boldsymbol{q} \tag{5.61}$$

with positive definite, symmetric matrices $K_p, K_d \in \mathbb{R}^{n_q \times n_q}$.

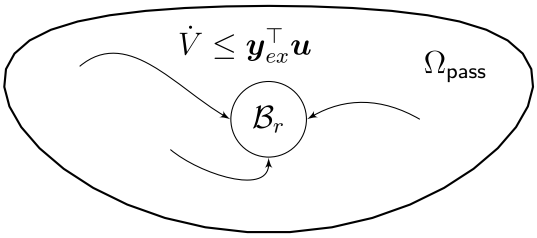


Figure 5.14: Concept of semi-passivity. The system behaves passive in $\Omega_{\text{pass}} \backslash \mathcal{B}_r$.

Remark 5.11. *For the sake of an intuitive understanding, we use static feedback gains in contrast to the CTC-GPR approach. However, the presented passivation can be extended to confidence depended feedback gains, following the idea of the proof of Theorem 5.2.*

The EL system (5.59) with input (5.60) can be rewritten as

$$\ddot{\boldsymbol{q}} = \check{\boldsymbol{\tau}}(\boldsymbol{q}_q) - K_d\dot{\boldsymbol{q}} - K_p\boldsymbol{q} - \boldsymbol{u}_{gp} + \boldsymbol{u}_{ex} \tag{5.62}$$

with $\check{\boldsymbol{\tau}}$ of the residual dynamics (5.5). The input $\boldsymbol{u}_{gp}$ is the predicted mean of the multi-output GP model (5.4) for the unknown function values $\check{\boldsymbol{\tau}}$, i.e.,

$$\boldsymbol{u}_{gp} = \boldsymbol{\mu}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_q, \mathcal{D}) \tag{5.63}$$

computed with (2.15). Thus, the prediction is based on $\boldsymbol{q}_q = [\ddot{\boldsymbol{q}}; \dot{\boldsymbol{q}}; \boldsymbol{q}]$ of the EL system and the training set $\mathcal{D}$, given by (5.5). The control structure is sketched in Fig. 5.15

5.4.3 Passivation

Preparing the main result about the passivity of the closed-loop system (5.62), the following definition and lemmas are introduced.

Definition 5.1. Consider a matrix-valued function Υ mapping $\mathbb{R}^{n_q \times n_q} \times \mathbb{R}^{n_q \times n_q} \times \mathbb{R}_{>0} \to \mathbb{R}^{2n_q \times 2n_q}$. We define Υ to be block-symmetric with

$$\Upsilon(K_d, K_p, \varepsilon) := \begin{bmatrix} K_d - \varepsilon I_{n_q} & \frac{\varepsilon}{2}K_d \\ \frac{\varepsilon}{2}K_d & \varepsilon K_p \end{bmatrix}. \tag{5.64}$$

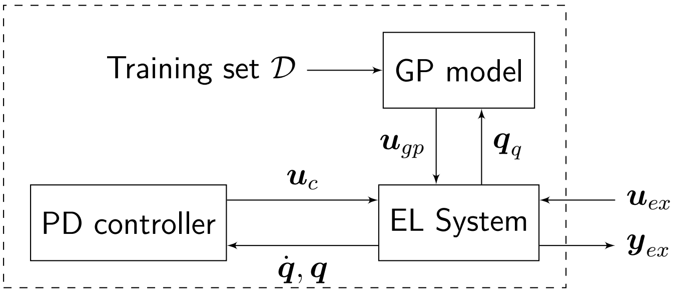


Figure 5.15: Semi-passively rendered EL System with respect to $\boldsymbol{u}_{ex}$ and $\boldsymbol{y}_{ex}$.

Lemma 5.4. *For any $\varepsilon, \underline{\lambda}_d \in \mathbb{R}_{>0}$, there exist positive definite and symmetric matrices $K_d, K_p \in \mathbb{R}^{n_q \times n_q}$, such that the smallest eigenvalue of Υ is lower bounded by*

$$\underline{\lambda}\left(\Upsilon(K_d, K_p, \varepsilon)\right) \geq \underline{\lambda}_d. \tag{5.65}$$

Proof. The goal is to prove the existence of matrices K_d, K_p such that the smallest eigenvalue of Υ is lower bounded by (5.65). Let $\check{K}_d, \check{K}_p \in \mathbb{R}^{n_q \times n_q}$ be positive definite, symmetric matrices. The matrix $\check{\Upsilon}_M \in \mathbb{R}^{2n_q \times 2n_q}$ with

$$\check{\Upsilon}_M = \begin{bmatrix} \check{K}_d & \frac{\varepsilon}{2}\check{K}_d \\ \frac{\varepsilon}{2}\check{K}_d & \varepsilon\check{K}_p \end{bmatrix} \tag{5.66}$$

is positive definite, if $\varepsilon\check{K}_p \succ 0$ and

$$\underbrace{\check{K}_d - \varepsilon\frac{\check{K}_d\check{K}_p^{-1}\check{K}_d}{4}}_{\check{\Upsilon}_S \in \mathbb{R}^{n_q \times n_q}} \succ 0 \tag{5.67}$$

using the property of the Schur complement. The eigenvalues of the matrix $\check{\Upsilon}_S$ are lower bounded by

$$\lambda_i(\check{\Upsilon}_S) \geq \underline{\lambda}(\check{K}_d) - \varepsilon\frac{\bar{\lambda}^2(\check{K}_d)}{4\underline{\lambda}(\check{K}_p)} \tag{5.68}$$

exploiting the smallest and largest eigenvalue of $\check{K}_d$ and $\check{K}_p$. Thus, it is always possible to choose a $\check{K}_p$, such that the matrix $\check{\Upsilon}_S \succ 0$. As a consequence, $\check{\Upsilon}_M \succ 0$ due to the positive definiteness of (5.67) and $\varepsilon\check{K}_p$. Now, consider a scaling factor $c_1 \in \mathbb{R}_{\geq 0}$. The eigenvalues of the overall sum $\check{\Upsilon} \in \mathbb{R}^{n_q \times n_q}$ of the two symmetric matrices

$$\check{\Upsilon} = c_1 \begin{bmatrix} \check{K}_d & \frac{\varepsilon}{2}\check{K}_d \\ \frac{\varepsilon}{2}\check{K}_d & \varepsilon\check{K}_p \end{bmatrix} + \begin{bmatrix} -\varepsilon I_{n_q} & 0 \\ 0 & 0 \end{bmatrix} \tag{5.69}$$

are lower bounded by $\underline{\lambda}(\check{\Upsilon}) \geq -\varepsilon + c_1\underline{\lambda}(\check{\Upsilon}_M)$. Since $\check{\Upsilon}_M \succ 0$, for any ε and $\underline{\lambda}_d$ there exists a c_1 such that the eigenvalue $\underline{\lambda}(\check{\Upsilon}) \geq \underline{\lambda}_d$. Finally, defining $K_d = c_1\check{K}_d$ and $K_p = c_1\check{K}_p$ concludes the proof. □

Lemma 5.5. *For any $\varepsilon \in \mathbb{R}_{>0}$, there exist positive definite and symmetric matrices $K_d, K_p \in \mathbb{R}^{n_q \times n_q}$ with*

$$\bar{\lambda}(K_d) \leq \bar{k}_d \in \mathbb{R}_{>0}, \quad \text{with } \bar{k}_d > \varepsilon \tag{5.70}$$

$$\bar{\lambda}(K_p) \leq \bar{k}_p \in \mathbb{R}_{>0}, \quad \text{with } \bar{k}_p > \frac{\varepsilon}{4}\frac{\bar{k}_d^2}{\bar{k}_d - \varepsilon}, \tag{5.71}$$

such that $\Upsilon(K_d, K_p, \varepsilon) \succ 0$.

Proof. The matrix $\Upsilon(K_d, K_p, \varepsilon)$ is positive definite, if and only if $\varepsilon K_p \succ 0$ that is fulfilled by definition, and

$$\underbrace{\check{K}_d - \varepsilon I_{n_q} - \varepsilon \frac{K_d K_p^{-1} K_d}{4}}_{\Upsilon_{SS} \in \mathbb{R}^{n_q \times n_q}} \succ 0. \tag{5.72}$$

Analogous to the proof of Lemma 5.4, the eigenvalues of Υ_S are lower bounded by

$$\lambda_i(\Upsilon_{SS}) \geq \underline{\lambda}(K_d - cI_{n_q}) - \varepsilon \frac{\bar{\lambda}^2(K_d)}{4\underline{\lambda}(K_p)}. \tag{5.73}$$

Hence, it is possible to obtain $\underline{\lambda}(\Upsilon_{SS}) > 0$ with matrices K_d, K_p, which satisfy $\bar{\lambda}(K_d) \leq \bar{k}_d$ and $\bar{\lambda}(K_p) \leq \bar{k}_p$. □

Theorem 5.3. *Given Assumption 5.6 and the closed-loop system* (5.62)*, there exist positive definite matrices K_p, K_d with $\underline{\lambda}(K_p) > c_{ex}^2$, such that the EL System* (5.59) *is rendered strictly semi-passive with probability $\delta_m \in (0,1)$ and radius*

$$r = \sqrt{\frac{(1 + c_{ex})\bar{\Delta}(\delta_m)}{\underline{\lambda}\left(\Upsilon(K_d, K_p, c_{ex})\right)}} \tag{5.74}$$

on a compact set $\Omega_{pass} \subset \mathbb{R}^{3n_q}$.

Proof. We consider the storage function

$$V(\dot{\boldsymbol{q}}, \boldsymbol{q}) = \frac{1}{2}\boldsymbol{q}^\top K_p \boldsymbol{q} + \frac{1}{2}\dot{\boldsymbol{q}}^\top \dot{\boldsymbol{q}} + c_{\text{ex}}\dot{\boldsymbol{q}}^\top \boldsymbol{q}, \tag{5.75}$$

which is positive for $\underline{\lambda}(K_p) > c_{\text{ex}}^2$ for all $\dot{\boldsymbol{q}}, \boldsymbol{q} \in \mathbb{R}^{n_q}$ and zero for $\dot{\boldsymbol{q}} = \boldsymbol{q} = \boldsymbol{0}$. With the closed-loop dynamics (5.62), the derivative of V is given by

$$\dot{V}(\dot{\boldsymbol{q}}, \boldsymbol{q}) = -\begin{bmatrix}\dot{\boldsymbol{q}}^\top & \boldsymbol{q}^\top\end{bmatrix} \Upsilon(K_d, K_p, c_{\text{ex}}) \begin{bmatrix}\dot{\boldsymbol{q}} \\ \boldsymbol{q}\end{bmatrix} + \underbrace{(\dot{\boldsymbol{q}} + c_{\text{ex}}\boldsymbol{q})^\top}_{\boldsymbol{y}_{ex}} (\check{\boldsymbol{\tau}}(\boldsymbol{q}_q) - \boldsymbol{\mu}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_q, \mathcal{D}) + \boldsymbol{u}_{ex}). \tag{5.76}$$

The first summand of (5.76) depends on the feedback gains, whereas the second summand depends on the model error $\Delta = \check{\boldsymbol{\tau}}(\boldsymbol{q}_q) - \boldsymbol{\mu}(\check{\boldsymbol{\tau}}|\boldsymbol{q}_q, \mathcal{D})$. According to Lemma 5.4, there exist matrices K_d and K_p, such that the matrix Υ is positive definite for any c_{ex}. The error between the true dynamics and the mean of the GP model in (5.76) is bounded by a constant $\bar{\Delta}(\delta_m) \in \mathbb{R}_{>0}$ with probability δ_m for all $\boldsymbol{q}_q \in \Omega_{\text{pass}}$, using Lemma 5.1. Thus, the derivative of the Lyapunov function (5.76) is upper bounded by a storage function $h_{\text{ex}} \colon \mathbb{R}^{n_q} \times \mathbb{R}^{n_q} \to \mathbb{R}$ with

$$\dot{V}(\dot{\boldsymbol{q}}, \boldsymbol{q}) \leq \boldsymbol{y}_{ex}^\top \boldsymbol{u}_{ex} - h_{\text{ex}}(\dot{\boldsymbol{q}}, \boldsymbol{q}) \tag{5.77}$$

$$h_{\text{ex}}(\dot{\boldsymbol{q}}, \boldsymbol{q}) = \underline{\lambda}(\Upsilon)\left\|\begin{matrix}\dot{\boldsymbol{q}} \\ \boldsymbol{q}\end{matrix}\right\|^2 - \bar{\Delta}(\delta_m)\|\dot{\boldsymbol{q}}\| - c_{\text{ex}}\bar{\Delta}(\delta_m)\|\boldsymbol{q}\| \tag{5.78}$$

for all $[\ddot{\boldsymbol{q}}; \dot{\boldsymbol{q}}; \boldsymbol{q}] \in \Omega_{\text{pass}}$. The function h_{ex} is positive for

$$\left\| \begin{matrix} \dot{\boldsymbol{q}} \\ \boldsymbol{q} \end{matrix} \right\| > \sqrt{\frac{(1+c_{\text{ex}})\bar{\Delta}(\delta_m)}{\underline{\lambda}(\Upsilon)}} = r, \tag{5.79}$$

i.e., outside a ball $\mathcal{B}_r$ with the radius $r \in \mathbb{R}_{>0}$. Therefore, the system (5.59) is rendered strictly semi-passive with the probability δ_m in respect to $\boldsymbol{u}_{ex}$ and $\boldsymbol{y}_{ex}$. □

Remark 5.12. *The radius of the Ball B_r can be set arbitrary small by either decreasing the maximum model error $\bar{\Delta}$ with more training data or increasing the feedback gains such that $\underline{\lambda}(\Upsilon)$.*

Remark 5.13. *The derivative of the Lyapunov function* (5.76) *is similar to the derivative* (5.42) *for the boundedness proof of CTC-GPR with $c_{ex} = \varepsilon$. However, the situation differs as c_{ex} can not be chosen arbitrarily as it is defined by the output of the system* (5.59)*.*

Remark 5.14. *We have extended the results beyond EL systems to a more general class of nonlinear systems, see [BH18].*

5.4.4 Numerical Evaluation

In this simulation, we use a modified Duffing oscillator [KB11b]

$$\ddot{q} = u^{1/3} - 0.1\dot{q} + 0.1q + 0.1q^3 + 1 \tag{5.80}$$

as sample system, where not only the parameters are unknown but also the entire parametric form of the dynamics is assumed to be unknown. This nonlinear, second-order system describes the motion of a damped oscillator with a more complex potential than in simple harmonic motion. The parameters are selected, such that the equilibrium point of the oscillator is unstable, see the top phase plane portrait of Fig. 5.16. For the demonstration of the flexibility of the proposed method, the control input is chosen to be not input affine. Now, the passivation approach of Theorem 5.3 is applied. We define $c_{\text{ex}} = 0.5$ for the passive output and set the external input $\boldsymbol{u}_{ex}$ to zero. Additionally, we set

$$\bar{k}_d := 0.9 > 0.5 = c_{\text{ex}} \tag{5.81}$$

$$\bar{k}_p := 1 > 0.253 = \frac{c_{\text{ex}}\bar{k}_d^2}{4(\bar{k}_d - c_{\text{ex}})}, \tag{5.82}$$

such that $\Upsilon(K_d, K_p, c_{\text{ex}}) \succ 0$ with the feedback gains $K_d = 0.9$ and $K_p = 1$. Since the drift function of the oscillator is continuous, we employ the squared exponential kernel for the GP model to approximate $\check{\tau}$. For this purpose, we generate 720 pairs of inputs $\{\boldsymbol{q}_q\}$ and outputs $\{\check{\boldsymbol{\tau}}\}$ as training data on $\Omega_{\text{pass}} = [-5,5]\times[-2,2]\times[-2,2]$. The hyperparameters of the squared exponential kernel function are optimized by the means of the log likelihood function. Assumption 5.6 is fulfilled as $\boldsymbol{q}_q$ is element of Ω_{pass} for the maximum model error $\bar{\Delta} = 0.045$ on the set Ω_{pass}, see Fig. 5.17. In addition, the ball $\mathcal{B}_r$ is a subset of Ω_{pass}, which is visualized in the bottom of Fig. 5.16. Thus, the Duffing oscillator is rendered strictly semi-passive. The result is that inside the set $\Omega_{\text{pass}} \backslash \mathcal{B}_r$, the closed-loop system behaves passive.

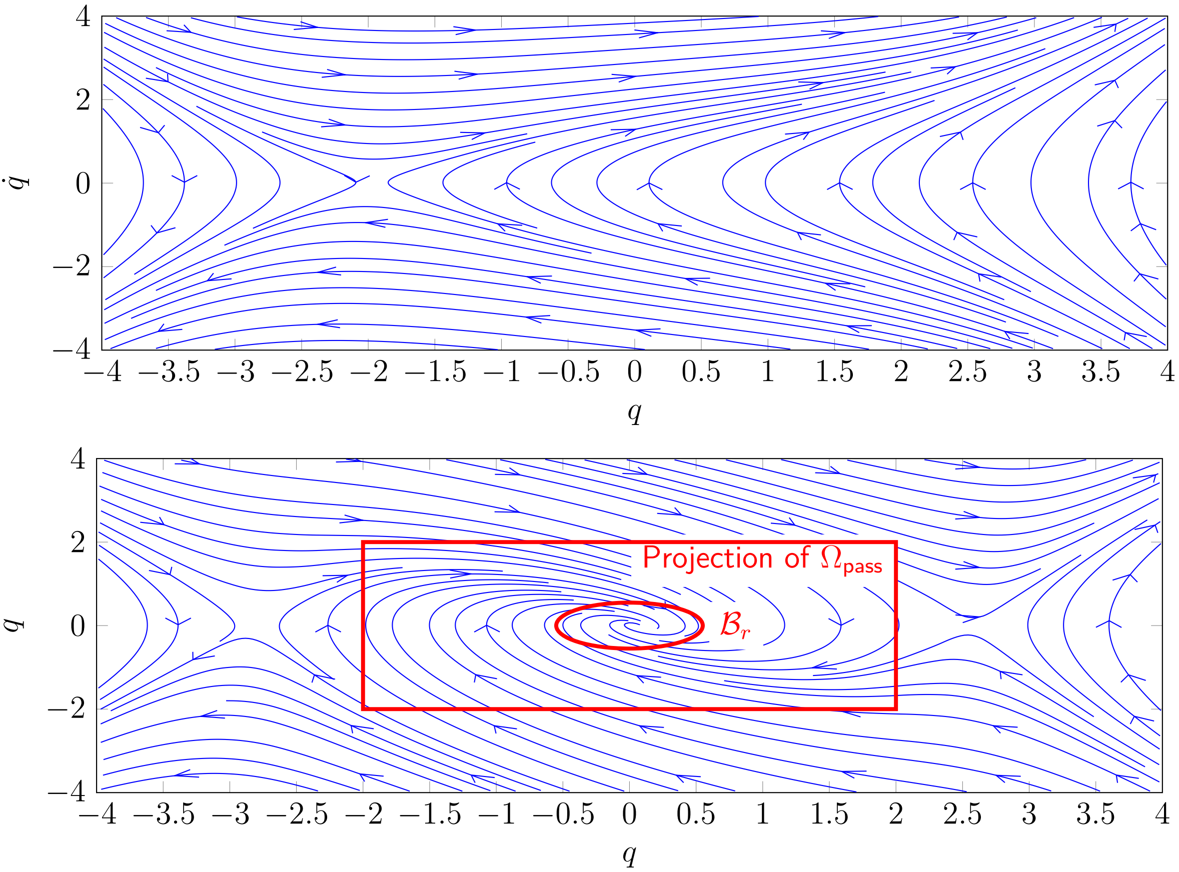


Figure 5.16: Phase plane portrait. Top: The uncontrolled Duffing oscillator. Bottom: The closed-loop system with the Duffing oscillator is strictly semi-passive in Ω_{pass} with the ball $\mathcal{B}_r$

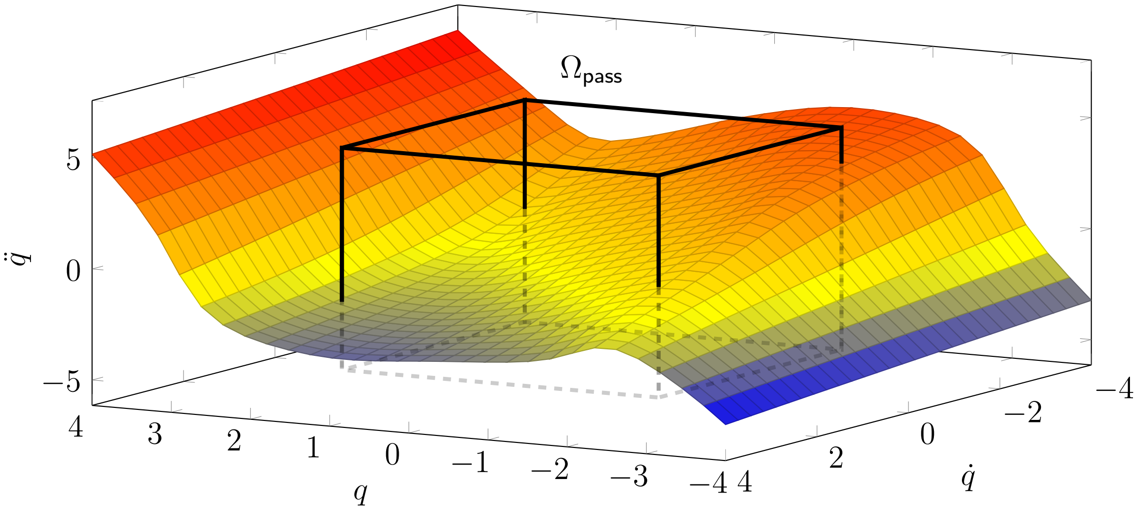


Figure 5.17: The figure demonstrates that with the selected K_p, K_d and $\bar{\Delta}$, the concatenated state vector $\boldsymbol{q}_q$ is always in Ω_{pass}

5.5 Discussion

In this chapter, we introduce GP model based control laws for Euler-Lagrange systems with unknown dynamics. For this purpose, a GP model learns the difference between an estimated, parametric model and the actual system dynamics. This learning approach enables to include prior knowledge in form of an estimated parametric model, which is often available using classical system identification. In Section 5.2, we introduce an augmented computed torque control law based on GPR for Euler-Lagrange systems. The control law uses the estimated and the GP model to compensate the unknown dynamics. The derived method guarantees stochastic sample path boundedness of the tracking error around zero. If the number of training points tends to infinity, the tracking error becomes asymptotically stable. The proposed control law is of stochastic nature and the convergence occurs in probability. Therefore, also its deterministic pendant (the GP's mean function) leads to the stable behavior. The experiment in Section 5.2.3 shows the superiority of the GP model based approach in comparison to classical approaches. However, this learning approach requires several assumptions to be satisfied that restrict the class of Euler-Lagrange systems, where the approach can be applied. In addition, the information about the uncertainty of the GP model is not exploited which leads to a poor control performance outside the area with training data.

Therefore, in Section 5.3, we propose a GP model based control approach with relaxed assumptions, where the feedback gains are adapted by the model fidelity. For this purpose, we use the mean prediction of the GP model to compensate the residual dynamics of the system and the variance to adapt the feedback gains. The idea is to increase the feedback gains if the model uncertainty is increasing to keep the tracking error low. If the model is reliable, i.e., low variance, the feedback gains are reduced such that the control law focuses on feedforward compensation. This approach keeps the feedback gains low while the tracking error of the closed-loop system is proven to be uniformly ultimately bounded and exponentially convergent to a ball for a given probability. The result shows the correlation between the bound of the tracking error, the uncertainty of the model and the value of the feedback gains. In contrast to the first approach, here the only assumption on the Euler-Lagrange system is that the RKHS norm of the residual dynamics is bounded. Even though we focus on Euler-Lagrange system, this allows the consideration of a significantly larger class of systems that are described by the dynamics (5.2).

Finally, we show how a GP model based control approach is used to render Euler-Lagrange systems with unknown dynamics strictly semi-passive. As consequence, the closed-loop system behaves passive outside a ball $\mathcal{B}_r$ on a set Ω_{pass}. It is shown, that an arbitrary small radius of the ball B_r can be achieved, depending on the model error and the feedback gains. The approach is similar to the approach in Section 5.3.2 but differs in the existence of a predefined output for the system.

6 Conclusion and Future Directions

Many classical control approaches are based on physical dynamic models, which describe the underlying system behavior in a sufficiently accurate fashion. For complex dynamical systems, however, such descriptions are often extremely hard to obtain, or even non-existent. Therefore, data-driven approaches are highly attractive to overcome this issue. Data-driven models are based on observations and measurements of the true system and need a minimum amount of prior knowledge of the system only. Especially, Gaussian process (GP) models have been used extensively for modeling dynamical systems due to their beneficial properties as the bias-variance trade-off and the adherence to Bayesian mathematics. However, they require new analytic tools and control approaches as classical methodologies are not suitable for data-driven probabilistic models. Therefore, the current application is often limited to non-safety critical and low performance systems.

In this thesis, we analyze the behavior of GP models and introduce their application in control settings including explicit mathematical guarantees for the stability and performance of the control loop.

Summary of the Contributions

The contribution of this thesis can be separated into three parts. In Chapter 3, we analyze the control theoretic properties of GP dynamical models, namely GP state space models (GP-SSM) and GP nonlinear output error (GP-NOE) models. It is shown that the sampling of GP dynamical models for simulation purposes results in non-Markovian dynamics and how Markovian approximated GP dynamics are constructed. We derive conditions for boundedness of the dynamics that heavily depend on the selected kernel function. Furthermore, the approximation error of these stochastic models is quantified, focusing on the relation between different approximations and the true, non-Markovian dynamics. For the special case of deterministic GP-SSMs, the number and stability of equilibria is determined in terms of Lyapunov stability and ultimate boundedness. GP-SSMs with a linear, polynomial, and squared exponential kernel function are analyzed. For the squared exponential kernel, we show globally uniform ultimate boundedness for the GP dynamical model. For stochastic GP-SSMs, we present a method to compute the equilibrium distribution, which is based on the solution of a Fredholm integral equation. The method is applicable to arbitrary kernel functions. Furthermore, we present an upper bound in the mean square sense and a set which is positive recurrent, for stochastic GP-SSMs with a squared exponential kernel. By exploiting these results, we prove that by the choice of the kernel control theoretic prior knowledge can be inserted into a GP dynamical model. The derived results are illustrated in numerical simulations.

In Chapter 4, we present a Bayesian optimization based framework to optimize the ker-

nel function and its hyperparameters of a kernel-based model. In contrast to the standard data-based optimization, our approach optimizes the performance of the closed-loop. Additionally, the stability properties are preserved during the optimization process. Through numerical examples and an experiment on a real 3-DoF robot, we demonstrate the advantages of the proposed approach over classical model selection methods. For the case where an unfit kernel is selected that is not suitable to represent the underlying unknown function, an upper bound for the mean square prediction error between an estimated GP model and the suitable but unknown GP model is derived. For this purpose, we exploit the property that many commonly used kernel functions are pseudo-concave with respect to their hyperparameters. As a result, the upper bound is the solution of a pseudo-concave optimization problem. For common application scenarios, a closed-form solution is provided.
In Chapter 5, a GP model-based control scheme for Euler-Lagrange systems with unknown dynamics is proposed. Formal mathematical guarantees ensure the safety, by means of stability and adherence to specified performance of the closed-loop. The mean prediction of the GP model is used for the feed-forward compensation of the unknown dynamics of the system. The gains of the feedback part are adapted based on the uncertainty of the learned model. Thus, the feedback gains are kept low as long as the learned model describes the true system with sufficient precision. We show which class of feedback gain adaption laws guarantees a globally bounded tracking error while incorporating the uncertainty of the GP model. Additionally, the performance of the closed loop is derived, i.e., the quantification of the maximum tracking error based on the number of training points. Furthermore, we provide a control law to achieve the passivation of Euler-Lagrange systems, which is beneficial for the safe interconnection of multiple systems. A robotic experiment and numerical simulations demonstrate the superiority of the presented approaches.

Implications

We summarize the implication of the presented results in terms of the challenges imposed by control based on data-driven models, as introduced in Section 1.1.

Challenge 1.1 is addressed by presenting different scenarios for GP models as dynamical system, i.e., GP-SSMs and GP-NOE models. The crux of GP dynamical models is the cross-correlation between the states. We solve this challenge with a modified training set, which includes a memory with past states and inputs. Markovian stochastic and deterministic GP dynamical models are introduced, which allow the application of classical control analysis and synthesis techniques.

Challenge 1.2 is met by the probabilistic analysis of GP dynamical models with respect to the stability, boundedness, and number and distribution of the equilibrium points. The multiple ways to use GP models as dynamical systems are considered and analyzed depending on the kernel functions used. The proposed relation between different kernel functions and the resulting control related properties of the GP dynamical model enables an appropriate selection of the kernel. Furthermore, it allows to include prior knowledge about the actual system behavior.

Challenge 1.3 is met by the introduction of Bayesian optimization (BO) based selection of the kernel function. In contrast to data-based selection, the BO aims to find the kernel and hyperparameters which optimize the control performance with respect to a cost functional of the closed-loop. During the optimization, the stability of the closed-loop is preserved due to the boundedness of the reproducing kernel Hilbert space (RKHS) norm for different kernels and sets of hyperparameters. In case an unfit kernel function is selected for a GP model, the predicted variance is misleading as measure for the model uncertainty. In this case, we provide an upper bound which successfully confines the true model error without knowing the corresponding kernel and hyperparameters.

Challenge 1.4 is addressed by a computed-torque based control law based on GP models for Euler-Lagrange systems with unknown dynamics. The mean prediction of the GP model is used as feed-forward compensation of the unknown dynamics. Furthermore, the model uncertainty is exploited to adapt the feedback gains. If the model uncertainty is increasing, the feedback gains will be increased to keep the tracking error low. Thus, the full information of the GP model is exploited to balance between feed-forward and feedback control.

Challenge 1.5 is met by the usage of the uncertainty of the GP model. The model uncertainty is transformed into a model error on a compact set, exploiting the Bayesian properties of the GP. This leads to adoption of probabilistic results not only about the boundedness of the tracking error but also about the performance given as the maximal tracking error. The presented relation between model error, training data and closed-loop performance guides different design strategies to guarantee the stability and a specific performance of the closed-loop.

Outlook

In this thesis, we take a step towards a full understanding of GP models. We investigate how they can be exploited for safety-critical control scenarios. Novel analysis techniques for GP dynamical models are presented and numerical as well as experimental results demonstrate the superiority against classical control techniques. Although a number of issues are addressed, there are still challenges in the area of data-driven control for future works.

Complex prior knowledge Although the presented approaches allow to include prior knowledge about the stability and equilibrium points, the consideration of more complex prior knowledge seems to be possible. Especially physical structures, such as conservation of energy, will help to extend the application of GP models. In general, there exist various ways to integrate prior knowledge into data-driven models, e.g., through the topology, constrained learning or specific activation/kernel functions. So far, the research on integration of physical structures is often focused on specific partial differential equations. Recently, first approaches to encode physical structures for modeling of dynamical systems into deep neural networks are presented [RK18]. This is, however, still an open problem for stochastic data-driven models.

Non-parametric control laws The presented control law for Euler-Lagrange systems with unknown dynamics show an enhanced performance of the closed-loop in contrast to classical methods. However, the control law still depends on parameters, such as the feedback gain function, that must be determined and tuned by the user. Thus, fully non-parametric control laws are an interesting direction for further research.

Online learning The extension of the training data set during the control process allows the GP model to adapt to a changing situation or to further improve the accuracy. However, this approach rises computing time issues as the inversion of the Gram matrix and the hyperparameter estimation must be performed online. Furthermore, the analysis of the closed-loop must be adapted to keep the stability and performance guarantees. First steps in this direction are recently made, for instance in [Uml+20a].

A

Appendix

A.1 Conditional Distribution

Let $\boldsymbol{\nu}_1 \in \mathbb{R}^{n_{\nu_1}}, \boldsymbol{\nu}_2 \in \mathbb{R}^{n_{\nu_2}}$ with $n_{\nu_1}, n_{\nu_1} \in \mathbb{N}$ be probability variables, which are multivariate Gaussian distribution

$$\begin{bmatrix}\boldsymbol{\nu}_1 \\ \boldsymbol{\nu}_2\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2\end{bmatrix}, \begin{bmatrix}\Sigma_{11} & \Sigma_{12}^\top \\ \Sigma_{12} & \Sigma_{22}\end{bmatrix}\right) \tag{A.1}$$

with mean $\boldsymbol{\mu}_1 \in \mathbb{R}^{n_{\nu_1}}, \boldsymbol{\mu}_2 \in \mathbb{R}^{n_{\nu_2}}$ and variance $\Sigma_{11} \in \mathbb{R}^{n_{\nu_1} \times n_{\nu_1}}, \Sigma_{12} \in \mathbb{R}^{n_{\nu_2} \times n_{\nu_1}}, \Sigma_{22} \in \mathbb{R}^{n_{\nu_2} \times n_{\nu_2}}$. The task is to determine the conditional probability

$$\mathrm{p}(\boldsymbol{\nu}_2|\boldsymbol{\nu}_1) = \frac{\mathrm{p}(\boldsymbol{\nu}_1, \boldsymbol{\nu}_2)}{\mathrm{p}(\boldsymbol{\nu}_1)}. \tag{A.2}$$

The joined probability $\mathrm{p}(\boldsymbol{\nu}_1, \boldsymbol{\nu}_2)$ is a multivariate Gaussian distribution with

$$\mathrm{p}(\boldsymbol{\nu}_1, \boldsymbol{\nu}_2) = \frac{1}{(2\pi)^{(n_{\nu_1}+n_{\nu_2})/2} \det(\Sigma)^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right) \tag{A.3}$$

$$\boldsymbol{\mu} := \begin{bmatrix}\boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2\end{bmatrix}, \quad \Sigma := \begin{bmatrix}\Sigma_{11} & \Sigma_{12}^\top \\ \Sigma_{12} & \Sigma_{22}\end{bmatrix}, \tag{A.4}$$

where $\boldsymbol{x} = [\boldsymbol{x}_1; \boldsymbol{x}_2], \boldsymbol{x}_1 \in \mathbb{R}^{n_{\nu_1}}, \boldsymbol{x}_2 \in \mathbb{R}^{n_{\nu_2}}$. The marginal distribution of $\boldsymbol{\nu}_1$ is defined by the mean $\boldsymbol{\mu}_1$ and the variance Σ_{11} such that

$$\mathrm{p}(\boldsymbol{\nu}_1) = \frac{1}{(2\pi)^{\frac{n_{\nu_1}}{2}} \det(\Sigma_{11})^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{x}_1-\boldsymbol{\mu}_1)^\top \Sigma_{11}^{-1}(\boldsymbol{x}_1-\boldsymbol{\mu}_1)\right). \tag{A.5}$$

The division of the joint distribution by the marginal distribution results again in a Gaussian distribution with

$$\mathrm{p}(\boldsymbol{\nu}_2|\boldsymbol{\nu}_1) = \underbrace{\frac{\det(\Sigma_{11})^{\frac{1}{2}}}{(2\pi)^{\frac{n_{\nu_2}}{2}} \det(\Sigma)^{\frac{1}{2}}}}_{*} \exp\Bigg(-\frac{1}{2} \underbrace{\Big[(\boldsymbol{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) - (\boldsymbol{x}_1-\boldsymbol{\mu}_1)^\top \Sigma_{11}^{-1}(\boldsymbol{x}_1-\boldsymbol{\mu}_1)\Big]\Bigg)}_{**}, \tag{A.6}$$

where the first part $*$ can be rewritten as

$$* = \frac{1}{(2\pi)^{\frac{n_{\nu_2}}{2}}} \left(\frac{\det(\Sigma_{11})}{\det(\Sigma_{11})\det(\Sigma_{22}-\Sigma_{12}\Sigma_{11}^{-1}\Sigma_{12}^\top)}\right)^{\frac{1}{2}} = \frac{1}{(2\pi)^{\frac{n_{\nu_2}}{2}} \det(\Sigma_{22}-\Sigma_{12}\Sigma_{11}^{-1}\Sigma_{12}^\top)^{\frac{1}{2}}}. \tag{A.7}$$

Thus, the covariance matrix $\Sigma_{22|1}$ of the conditional distribution $\mathrm{p}(\boldsymbol{\nu}_2|\boldsymbol{\nu}_1)$ is given by

$$\Sigma_{22|1} = \Sigma_{22} - \Sigma_{12}\Sigma_{11}^{-1}\Sigma_{12}^\top. \tag{A.8}$$

For the simplification of the second part $**$ of (A.6), we exploit the special block structure of Σ, such that its inverse is given by

$$\Sigma = \begin{bmatrix}\Sigma_{11} & \Sigma_{12}\\ \Sigma_{21} & \Sigma_{22}\end{bmatrix}, \qquad \Sigma^{-1} = \begin{bmatrix}\Sigma_{11}' & \Sigma_{12}'\\ \Sigma_{21}' & \Sigma_{22}'\end{bmatrix} \tag{A.9}$$

$$\begin{aligned}\Sigma_{11}' &= \Sigma_{11}^{-1} + \Sigma_{11}^{-1}\Sigma_{12}N\Sigma_{21}\Sigma_{11}^{-1}\\ \Sigma_{12}' &= -\Sigma_{11}^{-1}\Sigma_{12}N\\ \Sigma_{21}' &= -N\Sigma_{21}\Sigma_{11}^{-1}\\ \Sigma_{22}' &= N\end{aligned} \tag{A.10}$$

with $N = (\Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12})^{-1}$. Thus, we compute $**$ as

$$** = \begin{bmatrix}\boldsymbol{x}_1 - \boldsymbol{\mu}_1\\ \boldsymbol{x}_2 - \boldsymbol{\mu}_2\end{bmatrix}^\top \begin{bmatrix}\Sigma_{11} & \Sigma_{12}^\top\\ \Sigma_{12} & \Sigma_{22}\end{bmatrix}^{-1} \begin{bmatrix}\boldsymbol{x}_1 - \boldsymbol{\mu}_1\\ \boldsymbol{x}_2 - \boldsymbol{\mu}_2\end{bmatrix} - (\boldsymbol{x}_1 - \boldsymbol{\mu}_1)^\top\Sigma_{11}^{-1}(\boldsymbol{x}_1 - \boldsymbol{\mu}_1) \tag{A.11}$$

$$\begin{aligned}&= (\boldsymbol{x}_2 - \boldsymbol{\mu}_2)^\top\Sigma_{22|1}^{-1}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2) + 2(\boldsymbol{x}_2 - \boldsymbol{\mu}_2)^\top\left(-\Sigma_{11}^{-1}\Sigma_{12}^\top\Sigma_{22|1}^{-1}\right)(\boldsymbol{x}_1 - \boldsymbol{\mu}_1)\\ &+ (\boldsymbol{x}_1 - \boldsymbol{\mu}_1)^\top\left(\Sigma_{11}^{-1} + \Sigma_{11}^{-1}\Sigma_{12}^\top\Sigma_{22|1}^{-1}\Sigma_{12}\Sigma_{11}^{-1}\right)(\boldsymbol{x}_1 - \boldsymbol{\mu}_1) - (\boldsymbol{x}_1 - \boldsymbol{\mu}_1)^\top\Sigma_{11}^{-1}(\boldsymbol{x}_1 - \boldsymbol{\mu}_1)\end{aligned} \tag{A.12}$$

$$= \Big(\boldsymbol{x}_2 - \underbrace{\boldsymbol{\mu}_2 + \Sigma_{12}\Sigma_{11}^{-1}(\boldsymbol{x}_1 - \boldsymbol{\mu}_1)}_{\boldsymbol{\mu}_{2|1}}\Big)^\top\Sigma_{22|1}^{-1}\Big(\boldsymbol{x}_2 - \underbrace{\boldsymbol{\mu}_2 + \Sigma_{12}\Sigma_{11}^{-1}(\boldsymbol{x}_1 - \boldsymbol{\mu}_1)}_{\boldsymbol{\mu}_{2|1}}\Big) \tag{A.13}$$

Finally, the conditional probability is given with the conditional mean $\boldsymbol{\mu}_{2|1} \in \mathbb{R}^{n_{\nu_2}}$ and the conditional covariance matrix $\Sigma_{22|1} \in \mathbb{R}^{n_{\nu_2}\times n_{\nu_2}}$ by

$$p(\boldsymbol{\nu}_2|\boldsymbol{\nu}_1) = \frac{1}{(2\pi)^{\frac{n_{\nu_2}}{2}}\det(\Sigma_{22|1})^{\frac{1}{2}}}\exp\left(-\frac{1}{2}(\boldsymbol{x}_2 - \boldsymbol{\mu}_{2|1})^\top\Sigma_{22|1}^{-1}(\boldsymbol{x}_2 - \boldsymbol{\mu}_{2|1})\right) \tag{A.14}$$

$$\begin{aligned}\boldsymbol{\mu}_{2|1} &= \boldsymbol{\mu}_2 + \Sigma_{12}\Sigma_{11}^{-1}(\boldsymbol{x}_1 - \boldsymbol{\mu}_1)\\ \Sigma_{22|1} &= \Sigma_{22} - \Sigma_{12}\Sigma_{11}^{-1}\Sigma_{12}^\top.\end{aligned} \tag{A.15}$$

A.2 Stability Definitions

In this thesis, we use various types of common stability definitions for discrete-time and continuous-time systems. For the sake of clarity, a short overview about the necessary definitions is presented.

Deterministic and discrete-time

Consider a deterministic, autonomous system

$$\boldsymbol{x}_{t+1} = \boldsymbol{f}(\boldsymbol{x}_t) \tag{A.16}$$

with state $\boldsymbol{x}_t \in \mathbb{R}^{n_x}, t \in \mathbb{N}$ and a mapping $\boldsymbol{f}\colon \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$.

Definition A.1. Any state $\boldsymbol{x}^* \in \mathbb{R}^{n_x}$ that satisfies $\boldsymbol{x}^* = \boldsymbol{f}(\boldsymbol{x}^*)$ is called an *equilibrium point* of the autonomous system (A.16).

Definition A.2. Without loss of generality, an equilibrium point $\boldsymbol{x}^* = 0$ of the autonomous system (A.16) is called

- *stable*, if for each $\varepsilon > 0$, there exists a $\delta = \delta(\varepsilon) > 0$ such that $\|\boldsymbol{x}_0\| < \delta \Rightarrow \|\boldsymbol{x}_t\| < \varepsilon$ for all $t \geq 0$.
- *unstable*, if it is not stable.
- *asymptotically stable*, if it is stable and δ can be chosen such that for any $\|\boldsymbol{x}_0\| < \delta \Rightarrow \lim_{t\to\infty} \|\boldsymbol{x}_t\| = 0$.

In addition to the stability of equilibrium points, there exist more general stability definitions.

Definition A.3. The nonempty set $\mathcal{I} \subset \mathbb{R}^{n_x}$ is called *invariant* for the autonomous system (A.16) if $\forall \boldsymbol{x}_0 \in \mathcal{I}$, the system evolution satisfies $\boldsymbol{x}_t \in \mathcal{I}, \forall t \in \mathbb{N}_{>0}$.

Loosely speaking, the invariant set is a subset in space $\mathbb{R}^{n_x}$ which any trajectory of the system will never leave once it has enters the invariant set. Important classes of invariant sets are, for example, equilibrium points and limit cycles.

Definition A.4. The nonempty set $\mathcal{I} \subset \mathbb{R}^{n_x}$ is called *attractive* for the autonomous system (A.16), if there is a neighbourhood $\mathcal{I}_n$ of $\mathcal{I}$ for which for all $\boldsymbol{x}_0 \in \mathcal{I}_n$ the trajectory $\boldsymbol{x}_t \to \mathcal{I}$ with $t \to \infty$. If the neighbourhood $\mathcal{I}_n$ equals $\mathbb{R}^{n_x}$, then the set is *globally attractive.*

Definition A.5. The solutions of a discrete-time system (A.16) are

- *uniformly ultimately bounded with ultimate bound b* if there exist positive constants b and c_1 and for every $c_2 \in (0, c_1)$, there exists a $T = T(c_2, b) \geq 0$, such that

$$\|\boldsymbol{x}_0\| \leq c_2 \Rightarrow \|\boldsymbol{x}_t\| \leq b,\ \forall t \geq T. \tag{A.17}$$

- *globally uniformly ultimately bounded*, if (A.17) holds for arbitrarily large c_2.

Deterministic and continuous-time

Consider a deterministic, autonomous system

$$\dot{\boldsymbol{x}}(t_c) = \boldsymbol{f}(\boldsymbol{x}(t_c)) \tag{A.18}$$

with time $t_c \in \mathbb{R}_{>0}$, state $\boldsymbol{x} \in \mathbb{R}^{n_x}$ and a mapping $\boldsymbol{f} \colon \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$.

Definition A.6. The solution of a continuous-time system (A.18) is

- *uniformly ultimately bounded with ultimate bound b* if there exist positive constants b and c_1 and for every $c_2 \in (0, c_1)$, there is $T = T(c_2, b) \geq 0$, such that

$$\|\boldsymbol{x}(0)\| \leq c_2 \Rightarrow \|\boldsymbol{x}(t_c)\| \leq b,\ \forall t_c \geq T. \tag{A.19}$$

- *globally uniformly ultimately bounded* if (A.19) holds for arbitrarily large c_2.

In contrast to stability and boundedness, the concept of passivity depends also on the output of the system. Semi-passivity is an extension of the passivity concept and defined in the following. Consider a deterministic system

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}_{\text{ex}}) \\ \boldsymbol{y}_{\text{ex}} &= \boldsymbol{g}(\boldsymbol{x}) \end{aligned} \tag{A.20}$$

with state $\boldsymbol{x} \in \mathbb{R}^{n_x}$, input $\boldsymbol{u}_{\text{ex}} \in \mathcal{U}$, output $\boldsymbol{y}_{\text{ex}} \in \mathbb{R}^{n_y}$ and a mapping $\boldsymbol{f}\colon \mathbb{R}^{n_x} \times \mathcal{U} \to \mathbb{R}^{n_x}, \boldsymbol{g}\colon \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$.

Definition A.7. Following [Pog98], the system (A.20) is called

1. *semi-passive* in Ω, if there exists a nonnegative function $V\colon \Omega \to \mathbb{R}_{\geq 0}$ where $V(0) = 0$ such that

$$\dot{V}(\boldsymbol{x}) = \frac{\partial V}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}_{\text{ex}}) \leq \boldsymbol{y}_{\text{ex}}^{\top} \boldsymbol{u}_{\text{ex}} - h(\boldsymbol{x}). \tag{A.21}$$

   The passive output $\boldsymbol{y}_{ex} \in \mathbb{R}^{n_y}$ is state-dependent and the function $h\colon \Omega \to \mathbb{R}$ is nonnegative outside the ball $\mathcal{B}_r = \{\boldsymbol{x} \in \mathbb{R} | \|\boldsymbol{x}\| \leq r \in \mathbb{R}_{\geq 0}\} \subset \Omega$ with radius r, i.e.,

$$\exists r > 0, \|\boldsymbol{x}\| > r \Rightarrow h(\boldsymbol{x}) \geq 0. \tag{A.22}$$

2. *strictly semi-passive* in Ω, if the system is semi-passive and the function $h(\boldsymbol{x})$ is positive outside some ball B_r.

Hence, outside the ball $\mathcal{B}_r$, the behavior of semi-passive systems is identical to passive systems. Additionally, a feedback interconnection with another passive system has an ultimately bounded solution, see [Pog98], such that every trajectory of the closed-loop systems enters a compact set in finite time and remains there.

Stochastic and discrete-time

For stochastic system, the following stability definitions are used, see [LZL13]. Consider a stochastic, autonomous system

$$\boldsymbol{x}_{t+1} = \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{w}_t) \tag{A.23}$$

with state $\boldsymbol{x}_t \in \mathbb{R}^{n_x}, t \in \mathbb{N}$, a stochastic process $\boldsymbol{w}_t$ defined on the complete probability space $(\Omega_{\text{ss}}, \mathcal{F}_\sigma, P)$ and a mapping $\boldsymbol{f}\colon \mathbb{R}^{n_x} \times \Omega_{\text{ss}} \to \mathbb{R}^{n_x}$.

Definition A.8. The discrete-time dynamical system (A.23) is called *mean square bounded*, if the solution $\boldsymbol{x}_t$ for $t \in \mathbb{N}$ is bounded with $\sup_{t\in\mathbb{N}} \mathbb{E}\left[\|\boldsymbol{x}_t\|^2\right] < \infty$.

Definition A.9. The nonempty and measurable set $\mathcal{I}_r \subset \mathbb{R}^{n_x}$ is called *positive recurrent* for (A.23) if $\sup_{\boldsymbol{x}_t \in \mathcal{I}_r} \mathbb{E}[t_{\mathcal{I}}] < \infty$, where $t_{\mathcal{I}} = \inf\{t \geq 1\colon \boldsymbol{x}_t \in \mathcal{I}_r\}$ is the first return time to $\mathcal{I}_r$ if $\boldsymbol{x}_0 \in \mathcal{I}_r$ and the first hitting time, otherwise.

More general, the boundedness can be defined for an arbitrarily generated sequence of stochastic variables.

Definition A.10. A sequence of random variables $\{\boldsymbol{x}_t\}, t \in \mathbb{N}$ with is called *p-bounded*

$$\sup_{t\in\mathbb{N}} \mathrm{E}\left[\|\boldsymbol{x}_t\|^p\right] < \infty \text{ for } p \in \mathbb{N}. \tag{A.24}$$

Note that this general definition is independent of the underlying generating process of the random variables.

Stochastic and continuous-time

Consider a stochastic, autonomous system

$$d\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x}, t_c)dt + G(\boldsymbol{x}, t_c)d\boldsymbol{\nu}_{\text{brw}} \tag{A.25}$$

with $\boldsymbol{f}\colon \mathbb{R}^{n_x} \times \mathbb{R}_{>0} \to \mathbb{R}^{n_x}$, $G\colon \mathbb{R}^{n_x} \times \mathbb{R}_{>0} \to \mathbb{R}^{n_x \times n_x}$, where $\boldsymbol{\nu}_{\text{brw}}$ indicates the Brownian motion and $n_x \in \mathbb{N}$.

Definition A.11. Let there exist a ball $\mathcal{B}_r = \{\|\boldsymbol{x}\| \leq r | \boldsymbol{x} \in \mathbb{R}^{n_x}, r > 0\}$ and a time t_1, denoting the first exit time from $\mathbb{R}^{n_x} \backslash \mathcal{B}_r$ for the solution $\boldsymbol{x}(t_c)$ with $\boldsymbol{x_0} \in \mathbb{R}^{n_x} \backslash \mathcal{B}_r$. The system (A.25) is *stochastically sample path bounded*, if for each $\varepsilon > 0$ there exists a $\delta > 0$ such that

$$P\left(\sup_{0 \leq t_c \leq t_1} \|\boldsymbol{x}(t)\| \leq \delta\right) > 1 - \varepsilon. \tag{A.26}$$

The following theorem provides a way to determine stochastically sample path boundedness.

Theorem A.1 ([Gar88])**.** *Consider the stochastic system* (A.25)*. Let $V \in \mathcal{C}^2$ be a proper Lyapunov function for which the drift operator satisfies*

$$\mathfrak{L}V(\boldsymbol{x}) := \frac{\partial V}{\partial \boldsymbol{x}} \boldsymbol{f} + \frac{1}{2} \operatorname{tr}\left(G^\top \frac{\partial^2 V}{\partial \boldsymbol{x} \partial \boldsymbol{x}} G\right) \leq 0 \tag{A.27}$$

for all $\boldsymbol{x} \in \mathbb{R}^{n_x} \backslash \mathcal{B}_r$. Then, the solution of (A.25) *is stochastically sample path bounded with $\mathcal{B}_r = \{\|\boldsymbol{x}\| \leq r | \boldsymbol{x} \in \mathbb{R}^{n_x}, r > 0\}$.*

Another stability criteria is given as follows:

Definition A.12. Let Ω be a neighborhood of the trivial solution of the system (A.25). The trivial solution is *stochastically stable* on Ω, if for every pair of $\varepsilon \in (0, 1)$ and $r > 0$, there exists a $\delta = \delta(\varepsilon, r) > 0$, such that

$$P\left(\|\boldsymbol{x}(t_c)\| < r \text{ for all } t_c \geq 0\right) \geq 1 - \varepsilon \tag{A.28}$$

whenever $\boldsymbol{x}_0 \in \Omega$ and $\|\boldsymbol{x}_0\| < \delta$.

Definition A.13. Let Ω be a neighborhood of the trivial solution of the system (A.25). The trivial solution is *stochastically asymptotically stable* on Ω if it is stochastically stable on Ω, and for every $\varepsilon \in (0, 1)$, there exists a $\delta_0 = \delta_0(\varepsilon) > 0$, such that

$$P\left(\lim_{t_c \to \infty} \|\boldsymbol{x}(t_c)\| = \boldsymbol{0}\right) \geq 1 - \varepsilon \tag{A.29}$$

whenever $\boldsymbol{x}_0 \in \Omega$ and $\|\boldsymbol{x}_0\| < \delta$.

The following theorem provides a way to proof the stochastic asymptotic stability.

Theorem A.2 ([Mao07])**.** *If there exists a positive-definite radially unbounded function $V(\boldsymbol{x}) \in \mathcal{C}^2$ such that the drift operator $\mathfrak{L}V$ is negative-definite on Ω, then the trivial solution of* (A.25) *is stochastically asymptotically stable on Ω.*

List of Figures

List of Tables

Bibliography

Own publications

[BCH20] **Thomas Beckers**, Leonardo Colombo, and Sandra Hirche. *Safe learning-based trajectory tracking for underactuated vehicles with partially unknown dynamics.* 2020. arXiv: `2009.06689 [eess.SY]`.

[Bec+17] **Thomas Beckers**, Jonas Umlauft, Dana Kulić, and Sandra Hirche. “Stable Gaussian Process based Tracking Control of Lagrangian Systems”. In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC).* 2017, pp. 5180–5185. DOI: `10.1109/CDC.2017.8264427`.

[Bec+19] **Thomas Beckers**, Somil Bansal, Claire J. Tomlin, and Sandra Hirche. “Closed-loop Model Selection for Kernel-based Models Using Bayesian Optimization”. In: *2019 IEEE 58th Conference on Decision and Control (CDC).* 2019, pp. 828–834. DOI: `10.1109/CDC40024.2019.9029690`.

[BH16a] **Thomas Beckers** and Sandra Hirche. “Equilibrium distributions and stability analysis of Gaussian Process State Space Models”. In: *2016 IEEE 55th Conference on Decision and Control (CDC).* 2016, pp. 6355–6361. DOI: `10.1109/CDC.2016.7799247`.

[BH16b] **Thomas Beckers** and Sandra Hirche. “Stability of Gaussian Process State Space Models”. In: *2016 European Control Conference (ECC).* 2016, pp. 2275–2281. DOI: `10.1109/ECC.2016.7810630`.

[BH18] **Thomas Beckers** and Sandra Hirche. “Gaussian Process based Passivation of a Class of Nonlinear Systems with Unknown Dynamics”. In: *2018 European Control Conference (ECC).* 2018, pp. 1257–1262. DOI: `10.23919/ECC.2018.8550311`.

[BH19] **Thomas Beckers** and Sandra Hirche. “Keep soft robots soft - A data-driven based trade-off between feed-forward and feedback control”. In: *Workshop on Robust autonomy: Tools for safety in real-world uncertain environments (RSS 2019).* 2019.

[BH20] **Thomas Beckers** and Sandra Hirche. *Prediction with Gaussian Process Dynamical Models.* 2020. arXiv: `2006.14551 [eess.SY]`.

[BKH19] **Thomas Beckers**, Dana Kulić, and Sandra Hirche. “Stable Gaussian Process based Tracking Control of Euler-Lagrange Systems”. In: *Automatica* 103 (2019), pp. 390–397. DOI: `10.1016/j.automatica.2019.01.023`.

[BUH17] **Thomas Beckers**, Jonas Umlauft, and Sandra Hirche. “Stable Model-based Control with Gaussian Process Regression for Robot Manipulators”. In: *IFAC-PapersOnLine* 50.1 (2017). 20th IFAC World Congress, pp. 3877–3884. DOI: `10.1016/j.ifacol.2017.08.359`.

[BUH18] **Thomas Beckers**, Jonas Umlauft, and Sandra Hirche. “Mean Square Prediction Error of Misspecified Gaussian Process Models”. In: *2018 IEEE Conference on Decision and Control (CDC)*. 2018, pp. 1162–1167. DOI: 10.1109/CDC.2018.8619163.

[Cap+20] Alexandre Capone, Gerrit Noske, Jonas Umlauft, **Thomas Beckers**, Armin Lederer, and Sandra Hirche. “Efficient online closed loop exploration using receding horizon control”. In: *Learning for Dynamics and Control*. Ed. by Proceedings of Machine Learning Research. 2020.

[Gei+10] Robert Geise, Jens Schueuer, Lena Thiele, Kai Notté, **Thomas Beckers**, and Achim Enders. “A slotted waveguide setup as scaled instrument-landing-system for measuring scattering of an A380 and large objects”. In: *Proceedings of the Fourth European Conference on Antennas and Propagation*. 2010, pp. 1–5. ISBN: 9781424464319.

[Led+20] Armin Lederer, Alexandre Capone, **Thomas Beckers**, Jonas Umlauft, and Sandra Hirche. *The Value of Data in Learning-Based Control for Training Subset Selection*. 2020. arXiv: 2011.10596 [eess.SY].

[UBH18] Jonas Umlauft, **Thomas Beckers**, and Sandra Hirche. “A Scenario-based Optimal Control Approach for Gaussian Process State Space Models”. In: *2018 European Control Conference (ECC)*. 2018, pp. 1386–1392. DOI: 10.23919/ECC.2018.8550458.

[Uml+17] Jonas Umlauft, **Thomas Beckers**, Melanie Kimmel, and Sandra Hirche. “Feedback Linearization using Gaussian Processes”. In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. 2017, pp. 5249–5255. DOI: 10.1109/CDC.2017.8264435.

[Uml+20a] Jonas Umlauft, **Thomas Beckers**, Alexandre Capone, Armin Lederer, and Sandra Hirche. “Smart Forgetting for Safe Online Learning with Gaussian Processes”. In: *Learning for Dynamics and Control*. Ed. by Proceedings of Machine Learning Research. 2020.

[Uml+20b] Jonas Umlauft, Armin Lederer, **Thomas Beckers**, and Sandra Hirche. *Realtime Uncertainty Decomposition for Online Learning Control*. 2020. arXiv: 2010.02613 [cs.LG].

[Yam+20] Junya Yamauchi, **Thomas Beckers**, Marco Omainska, Takeshi Hatanaka, S. Hirche, and Masayuki Fujita. “Visual Pursuit Control with Target Motion Learning via Gaussian Process”. In: *Proceedings of the Conference of the Society of Instrument and Control Engineers of Japan*. 2020. ISBN: 9781728110905.

Other publications

[AK08] Kristjan Ažman and Juš Kocijan. “Non-linear model predictive control for models with local information and uncertainties”. In: *Transactions of the Institute of Measurement and Control* 30.5 (2008), pp. 371–396. DOI: 10.1177/0142331208095433.

[AMS14] Nicolas T. Alberto, Michael Mistry, and Freek Stulp. “Computed torque control with variable gains through Gaussian process regression”. In: *2014 IEEE-RAS International Conference on Humanoid Robots*. 2014, pp. 212–217. DOI: 10.1109/HUMANOIDS.2014.7041362.

[AQN06] Pieter Abbeel, Morgan Quigley, and Andrew Y. Ng. “Using inaccurate models in reinforcement learning”. In: *Proceedings of the 23rd International Conference on Machine Learning*. 2006, pp. 1–8. DOI: 10.1145/1143844.1143845.

[Arc96] David Archer. “A data-driven model of the global calcite lysocline”. In: *Global Biogeochemical Cycles* 10.3 (1996), pp. 511–526. DOI: 10.1029/96GB01521.

[ARL11] Mauricio A. Alvarez, Lorenzo Rosasco, and Neil D. Lawrence. *Kernels for vector-valued functions: A review*. 2011. arXiv: 1106.6251 [stat.ML].

[ÁRL12] Mauricio A. Álvarez, Lorenzo Rosasco, and Neil D. Lawrence. “Kernels for Vector-Valued Functions: A Review”. In: *Foundations and Trends in Machine Learning* 4.3 (2012), pp. 195–266. DOI: 10.1561/2200000036.

[Aro50] Nachman Aronszajn. “Theory of reproducing kernels”. In: *Transactions of the American mathematical society* 68.3 (1950), pp. 337–404. DOI: 10.2307/1990404.

[Atk97] Kendall E. Atkinson. *The numerical solution of integral equations of the second kind*. Vol. 4. Cambridge University Press, 1997. DOI: 10.1017/CBO9780511626340.

[ÅW13] Karl J. Åström and Björn Wittenmark. *Mathematical System Theory*. Springer, Berlin, Heidelberg, 2013. DOI: 10.1007/978-3-662-08546-2_24.

[Ban+16] Somil Bansal, Anayo K. Akametalu, Frank J. Jiang, Forrest Laine, and Claire J. Tomlin. “Learning quadrotor dynamics using neural network for flight control”. In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. 2016, pp. 4653–4660. DOI: 10.1109/CDC.2016.7798978.

[Ban+18] Somil Bansal, Roberto Calandra, Ted Xiao, Sergey Levine, and Claire J. Tomlin. “Goal-driven dynamics learning via Bayesian optimization”. In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. 2018, pp. 5168–5173. DOI: 10.1109/CDC.2017.8264425.

[Ber+16] Felix Berkenkamp, Riccardo Moriconi, Angela P. Schoellig, and Andreas Krause. “Safe Learning of Regions of Attraction for Uncertain, Nonlinear Systems with Gaussian Processes”. In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. 2016, pp. 4661–4666.

[Ber+17] Felix Berkenkamp, Matteo Turchetta, Angela P. Schoellig, and Andreas Krause. “Safe Model-based Reinforcement Learning with Stability Guarantees”. In: *Advances in Neural Information Processing Systems*. 2017, pp. 908–918.

[Bis06] Christopher Bishop. *Pattern recognition and machine learning*. Springer-Verlag New York, 2006. ISBN: 9780387310732.

[BIW91] Christopher I. Byrnes, Alberto Isidori, and Jan C. Willems. “Passivity, feedback equivalence, and the global stabilization of minimum phase nonlinear systems”. In: *IEEE Transactions on Automatic Control* 36.11 (1991), pp. 1228–1240. DOI: 10.1109/9.100932.

[BLG16] Yusuf Bhujwalla, Vincent Laurain, and Marion Gilson. “The impact of smoothness on model class selection in nonlinear system identification: An application of derivatives in the RKHS”. In: *2016 American Control Conference (ACC)*. 2016, pp. 1808–1813. DOI: 10.1109/ACC.2016.7525181.

[Bre97] Christoph Bregler. “Learning and recognizing human dynamics in video sequences”. In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 1997, pp. 568–574. DOI: 10.1109/CVPR.1997.609382.

[BSS13] Mokhtar S. Bazaraa, Hanif D. Sherali, and Chitharanjan M. Shetty. *Nonlinear programming: theory and algorithms*. John Wiley & Sons, 2013. DOI: 10.1002/0471787779.

[BTA06] Douglas A. Bristow, Marina Tharayil, and Andrew G. Alleyne. “A survey of iterative learning control”. In: *IEEE Control Systems Magazine* 26.3 (2006), pp. 96–114. DOI: 10.1109/MCS.2006.1636313.

[Bul11] Adam D. Bull. “Convergence rates of efficient global optimization algorithms”. In: *Journal of Machine Learning Research* 12.88 (2011), pp. 2879–2904. URL: http://jmlr.org/papers/v12/bull11a.html.

[Cal+15] Roberto Calandra, André Seyfarth, Jan Peters, and Marc P. Deisenroth. “Bayesian Optimization for Learning Gaits under Uncertainty”. In: *Annals of Mathematics and Artificial Intelligence* 76.1 (2015), pp. 5–23. DOI: 10.1007/s10472-015-9463-9.

[CHK12] Girish Chowdhary, Jonathan How, and Hassan Kingravi. “Model Reference Adaptive Control using Nonparametric Adaptive Elements”. In: *AIAA Guidance, Navigation, and Control Conference*. 2012. DOI: 10.2514/6.2012-5038.

[Cho+13] Girish Chowdhary, Hassan Kingravi, Jonathan How, and Patricio A. Vela. “Bayesian nonparametric adaptive control of time-varying systems using Gaussian processes”. In: *2013 American Control Conference*. 2013, pp. 2655–2661. DOI: 10.1109/ACC.2013.6580235.

[Cho+15] Girish Chowdhary, Hassan A. Kingravi, Jonathan P. How, and Patricio A. Vela. “Bayesian nonparametric adaptive control using Gaussian processes”. In: *IEEE Transactions on neural networks and learning systems* 26.3 (2015), pp. 537–550. DOI: 10.1109/TNNLS.2014.2319052.

[CKM85] D. W. Clarke, P. P. Kanjilal, and C. Mohtadi. “A generalized LQG approach to self-tuning control Part I. Aspects of design”. In: *International Journal of Control* 41.6 (1985), pp. 1509–1523. DOI: 10.1080/0020718508961212.

[COL12] Tianshi Chen, Henrik Ohlsson, and Lennart Ljung. “On the estimation of transfer functions, regularizations and Gaussian processes - Revisited”. In: *Automatica* 48.8 (2012), pp. 1525–1535. DOI: 10.1016/j.automatica.2012.05.026.

[Cor90] M. Corless. “Guaranteed rates of exponential convergence for uncertain systems”. In: *Journal of Optimization Theory and Applications* 64.3 (1990), pp. 481–494. DOI: 10.1007/BF00939420.

[Dev86] Luc Devroye. "Sample-based non-uniform random variate generation". In: *Proceedings of the 18th conference on Winter simulation*. 1986, pp. 260–265. DOI: 10.1145/318242.318443.

[DFT13] John C. Doyle, Bruce A. Francis, and Allen R. Tannenbaum. *Feedback control theory*. Springer, Boston, MA, 2013. DOI: 10.1007/978-0-387-85460-1_1.

[DM+05] Lokenath Debnath, Piotr Mikusinski, et al. *Introduction to Hilbert spaces with applications*. Academic press, 2005.

[EH16] Sebastian Erhart and Sandra Hirche. "Model and analysis of the interaction dynamics in cooperative manipulation tasks". In: *IEEE Transactions on Robotics* 32.3 (2016), pp. 672–683. DOI: 10.1109/TRO.2016.2559500.

[Ele+17] Stefanos Eleftheriadis, Tom Nicholson, Marc P. Deisenroth, and James Hensman. "Identification of Gaussian process state space models". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017, pp. 5309–5319. DOI: 10.5555/3295222.3295283.

[EMM05] Yaakov Engel, Shie Mannor, and Ron Meir. "Reinforcement learning with Gaussian processes". In: *Proceedings of the 22nd International Conference on Machine Learning*. 2005, pp. 201–208. DOI: 10.1145/1102351.1102377.

[FCR14] Roger Frigola, Yutian Chen, and Carl E. Rasmussen. *Variational Gaussian Process State-Space Models*. 2014. arXiv: 1406.4905 [cs.LG].

[Fis+19] Jaime F. Fisac, Anayo K. Akametalu, Melanie N. Zeilinger, Shahab Kaynama, Jeremy Gillula, and Claire J. Tomlin. "A General Safety Framework for Learning-Based Control in Uncertain Robotic Systems". In: *IEEE Transactions on Automatic Control* 64.7 (2019), pp. 2737–2752. DOI: 10.1109/TAC.2018.2876389.

[FKK98] Randy A. Freeman, M. Krstić, and P. V. Kokotović. "Robustness of adaptive nonlinear control to bounded uncertainties". In: *Automatica* 34.10 (1998), pp. 1227–1230. DOI: 10.1016/S0005-1098(98)00070-3.

[Fra03] Alexander Fradkov. "Passification of Non-square Linear Systems and Feedback Yakubovich-Kalman-Popov Lemma". In: *European Journal of Control* 9.6 (2003), pp. 577–586. DOI: 10.3166/ejc.9.577-586.

[Fri+13] Roger Frigola, Fredrik Lindsten, Thomas B. Schön, and Carl E. Rasmussen. "Bayesian inference and learning in Gaussian process state-space models with particle MCMC". In: *Advances in Neural Information Processing Systems*. 2013, pp. 3156–3164.

[Fri16] Roger Frigola-Alcalde. "Bayesian time series learning with Gaussian processes". PhD thesis. University of Cambridge, 2016. DOI: 10.17863/CAM.46480.

[Gar88] Thomas C. Gard. *Introduction to stochastic differential equations*. M. Dekker, 1988. DOI: 10.1002/zamm.19890690808.

[GE79] Seymour Geisser and William F. Eddy. "A Predictive Approach to Model Selection". In: *Journal of the American Statistical Association* 74.365 (1979), pp. 153–160. DOI: 10.1080/01621459.1979.10481632.

[Gev05] Michel Gevers. “Identification for Control: From the Early Achievements to the Revival of Experiment Design”. In: *European Journal of Control* 11.4 (2005), pp. 335–352. DOI: 10.3166/ejc.11.335-352.

[GH20] Eduardo C. Garrido-Merchán and Daniel Hernández-Lobato. “Dealing with categorical and integer-valued variables in Bayesian Optimization with Gaussian processes”. In: *Neurocomputing* 380 (2020), pp. 20–35. DOI: 10.1016/j.neucom.2019.11.004.

[Gir+03] Agathe Girard, Carl E. Rasmussen, Joaquin Quinonero-Candela, and Roderick Murray-Smith. “Gaussian Process priors with uncertain inputs? Application to multiple-step ahead time series forecasting”. In: *Advances in Neural Information Processing Systems 15*. MIT Press, 2003.

[GZ16] Tepper L. Gill and Woodford W. Zachary. *Functional analysis and the Feynman operator calculus*. Springer International Publishing, 2016. DOI: 10.1007/978-3-319-27595-6.

[Hac12] Wolfgang Hackbusch. *Integral equations: theory and numerical treatment*. Birkhäuser, 2012. DOI: 10.1007/978-3-0348-9215-5.

[HGD96] Håkan Hjalmarsson, Michel Gevers, and Franky De Bruyne. “For model-based control design, closed-loop identification gives better performance”. In: *Automatica* 32.12 (1996), pp. 1659–1673. DOI: 10.1016/S0005-1098(96)80003-3.

[HP90] Joseph E. Higgins and Elijah Polak. “Minimizing pseudoconvex functions on convex compact sets”. In: *Journal of Optimization Theory and Applications* 65.1 (1990), pp. 1–27. DOI: 10.1007/BF00941156.

[HW13] Zhong-Sheng Hou and Zhuo Wang. “From model-based control to data-driven control: Survey, classification and perspective”. In: *Information Sciences* 235 (2013), pp. 3–35. DOI: 10.1016/j.ins.2012.07.014.

[Isi13] Alberto Isidori. *Nonlinear control systems*. Springer Science & Business Media, 2013. DOI: 10.1007/978-1-84628-615-5.

[Jer99] Abdul Jerri. *Introduction to integral equations with applications*. John Wiley & Sons, 1999. ISBN: 9780471317340.

[KB11a] Seyed M. Khansari-Zadeh and Aude Billard. “Learning stable nonlinear dynamical systems with Gaussian mixture models”. In: *IEEE Transactions on Robotics* 27.5 (2011), pp. 943–957. DOI: 10.1109/TRO.2011.2159412.

[KB11b] Ivana Kovacic and Michael J. Brennan. *The Duffing equation: nonlinear oscillators and their behaviour*. John Wiley & Sons, 2011. DOI: 10.1002/9780470977859.

[Kec01] Vojislav Kecman. *Learning and soft computing: support vector machines, neural networks, and fuzzy logic models*. Vol. 47. 1. 2001, pp. 305–307. DOI: 10.1016/S0925-2312(01)00685-3.

[KG02] Vikram Kapila and Karolos M. Grigoriadis. *Actuator saturation control*. CRC Press, 2002.

[KL99] Robert Keviczky and Haber Laszlo. *Nonlinear system identification: input-output modeling approach*. Springer Netherlands, 1999. ISBN: 9780792358589.

[KM05] Juš Kocijan and Roderick Murray-Smith. *Nonlinear predictive control with a Gaussian process model.* Springer, Berlin, Heidelberg, 2005. DOI: 10.1007/978-3-540-30560-6_8.

[Koc+03a] Juš Kocijan, B. Likar, B. Banko, Agathe Girard, Roderick Murray-Smith, and Carl E. Rasmussen. "A case based comparison of identification with neural network and Gaussian process models". In: *Proceedings of the International Conference on Intelligent Control Systems and Signal Processing ICONS 2003.* 2003, pp. 137–142.

[Koc+03b] Juš Kocijan, Roderick Murray-Smith, Carl E. Rasmussen, and Bojan Likar. "Predictive control with Gaussian process models". In: *The IEEE Region 8 EUROCON 2003. Computer as a Tool.* Vol. 1. 2003, pp. 352–356.

[Koc+04] Juš Kocijan, Roderick Murray-Smith, Carl E. Rasmussen, and Agathe Girard. "Gaussian process model based predictive control". In: *Proceedings of the 2004 American Control Conference.* 2004, pp. 2214–2219. DOI: 10.23919/ACC.2004.1383790.

[Koc+05] Juš Kocijan, Agathe Girard, Blaž Banko, and Roderick Murray-Smith. "Dynamic systems identification with Gaussian processes". In: *Mathematical and Computer Modelling of Dynamical Systems* 11.4 (2005), pp. 411–424. DOI: 10.1080/13873950500068567.

[Koc16] Juš Kocijan. *Modelling and Control of Dynamic Systems Using Gaussian Process Models.* Springer International Publishing, 2016. DOI: 10.1007/978-3-319-21021-6.

[Koz12] Krzysztof R. Kozlowski. *Modelling and identification in robotics.* Springer Science & Business Media, 2012. DOI: 10.1007/978-1-4471-0429-2.

[KP11] Juš Kocijan and Dejan Petelin. "Output-error model training for Gaussian process models". In: *Adaptive and Natural Computing Algorithms.* 2011, pp. 312–321. DOI: 10.1007/978-3-642-20267-4_33.

[Kus71] Harold J. Kushner. *Introduction to stochastic control.* Holt, Rinehart and Winston New York, 1971. ISBN: 9780030849671.

[LE13] Ron Larson and Bruce Edwards. *Calculus of a single variable.* Springer, London, 2013. DOI: 10.1007/1-84628-222-5.

[Lea05] Erik G. Learned-Miller. "Data driven image models through continuous joint alignment". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.2 (2005), pp. 236–250. DOI: 10.1109/TPAMI.2006.34.

[Liu10] Xiuxiang Liu. "A note on the existence of periodic solutions in discrete predator–prey models". In: *Applied Mathematical Modelling* 34.9 (2010), pp. 2477–2483. DOI: 10.1016/j.apm.2009.11.012.

[Lju98] Lennart Ljung. "System identification". In: *Signal analysis and prediction.* Springer, 1998, pp. 163–173. DOI: 10.1007/978-1-4612-1768-8_11.

[LK07] Bojan Likar and Juš Kocijan. "Predictive control of a gas–liquid separation plant based on a Gaussian process model". In: *Computers & chemical engineering* 31.3 (2007), pp. 142–152. DOI: 10.1016/j.compchemeng.2006.05.011.

[LL13] Frank L. Lewis and Derong Liu. *Reinforcement learning and approximate dynamic programming for feedback control.* John Wiley & Sons, 2013. DOI: 10.1002/9781118453988.

[Loh12] Steve Lohr. “The age of big data”. In: *New York Times* 11.2012 (2012).

[LRP19] Michael Lutter, Christian Ritter, and Jan Peters. *Deep Lagrangian Networks: Using Physics as Model Prior for Deep Learning.* 2019. arXiv: 1907.04490 [cs.LG].

[LZL13] Yan Li, Weihai Zhang, and Xikui Liu. “Stability of nonlinear stochastic discrete-time systems”. In: *Journal of Applied Mathematics* 2013 (2013). DOI: 10.1155/2013/356746.

[Mac97] David J. MacKay. *Gaussian Processes - A Replacement for Supervised Neural Networks?* 1997.

[Man75] Olvi L. Mangasarian. “Pseudo-convex functions”. In: *Stochastic Optimization Models in Finance.* 1975, pp. 23–32. DOI: 10.1016/B978-0-12-780850-5.50009-5.

[Mao07] Xuerong Mao. *Stochastic differential equations and applications.* Elsevier, 2007. ISBN: 9781904275343.

[Mat+16] Alexander G. Matthews, James Hensman, Richard Turner, and Zoubin Ghahramani. “On sparse variational methods and the Kullback-Leibler divergence between stochastic processes”. In: *Journal of Machine Learning Research* 51 (2016), pp. 231–239.

[MLH15] José R. Medina, Tamara Lorenz, and Sandra Hirche. “Synthesizing Anticipatory Haptic Assistance Considering Human Behavior Uncertainty”. In: *IEEE Transactions on Robotics* 31.1 (2015), pp. 180–190. DOI: 10.1109/TRO.2014.2387571.

[MLS94] Richard M. Murray, Zexiang Li, and Shankar S. Sastry. *A mathematical introduction to robotic manipulation.* CRC press, 1994.

[Moc12] Jonas Mockus. *Bayesian approach to global optimization: theory and applications.* Vol. 37. Springer Science & Business Media, 2012. DOI: 10.1007/978-94-009-0909-0.

[Mog+16] Amir A. Amiri Moghadam, Keivan Torabi, Akif Kaynak, Muhd N. Alam, Abbas Kouzani, and Bobak Mosadegh. “Control-Oriented Modeling of a Polymeric Soft Robot”. In: *Soft Robotics* 3.2 (2016). DOI: 10.1089/soro.2016.0002.

[MSH13] José R. Medina, Dominik Sieber, and Sandra Hirche. “Risk-sensitive interaction control in uncertain manipulation tasks”. In: *2013 IEEE International Conference on Robotics and Automation.* 2013. DOI: 10.1109/ICRA.2013.6630621.

[MXZ06] Charles A. Micchelli, Yuesheng Xu, and Haizhang Zhang. “Universal kernels”. In: *Journal of Machine Learning Research* 7 (2006), pp. 2651–2667.

[NBO11] Emmanuel Nuño, Luis Basañez, and Romeo Ortega. “Passivity-based control for bilateral teleoperation: A tutorial”. In: *Automatica* 47.3 (2011), pp. 485–495. DOI: 10.1016/j.automatica.2011.01.004.

[Nel13] Oliver Nelles. *Nonlinear system identification: from classical approaches to neural networks and fuzzy models*. Springer-Verlag Berlin Heidelberg, 2013. DOI: 10.1007/978-3-662-04323-3.

[NP90] Kumpati S. Narendra and Kannan Parthasarathy. "Identification and control of dynamical systems using neural networks". In: *IEEE Transactions on Neural Networks* 1.1 (1990), pp. 4–27. DOI: 10.1109/72.80202.

[NSP08] Duy Nguyen-Tuong, Matthias Seeger, and Jan Peters. "Computed torque control with nonparametric regression models". In: *2008 American Control Conference*. 2008, pp. 212–217. DOI: 10.1109/ACC.2008.4586493.

[Ort+13] Romeo Ortega, Julio A. Perez, Per J. Nicklasson, and Hebertt Sira-Ramirez. *Passivity-based control of Euler-Lagrange systems: mechanical, electrical and electromechanical applications*. Springer Science & Business Media, 2013. DOI: 10.1007/978-1-4471-3603-3.

[PD11] Gianluigi Pillonetto and Giuseppe De Nicolao. "Kernel selection in linear system identification Part I: A Gaussian process perspective". In: *2011 50th IEEE Conference on Decision and Control and European Control Conference*. 2011, pp. 4318–4325. DOI: 10.1109/CDC.2011.6160606.

[PF13] Foster Provost and Tom Fawcett. "Data science and its relationship to big data and data-driven decision making". In: *Big data* 1.1 (2013), pp. 51–59. DOI: 10.1089/big.2013.1508.

[PGK13] Dejan Petelin, Alexandra Grancharova, and Juš Kocijan. "Evolving Gaussian process models for prediction of ozone concentration in the air". In: *Simulation modelling practice and theory* 33 (2013), pp. 68–80. DOI: 10.1016/j.simpat.2012.04.005.

[Pil+14] Gianluigi Pillonetto, Francesco Dinuzzo, Tianshi Chen, Giuseppe De Nicolao, and Lennart Ljung. "Kernel methods in system identification, machine learning and function estimation: A survey". In: *Automatica* 50.3 (2014), pp. 657–682. DOI: 10.1016/j.automatica.2014.01.001.

[PL70] M. Q. Phan and R. W. Longman. "Relationship between state-space and input-output models via observer Markov parameters". In: *WIT Transactions on The Built Environment* 22 (1970). DOI: 10.2495/DCSS960121.

[Pog98] A. Y. Pogromsky. "Passivity based design of synchronizing systems". In: *International Journal of Bifurcation and Chaos* 8.2 (1998), pp. 295–319. DOI: 10.1142/S0218127498000188.

[Qiu02] Li Qiu. "Essentials of robust control: Kemin Zhou, John C. Doyle". In: *Automatica* 38.5 (2002), pp. 910–912. ISSN: 0005-1098. DOI: 10.1016/S0005-1098(01)00272-2.

[QR05] Joaquin Quiñonero-Candela and Carl E. Rasmussen. "A unifying view of sparse approximate Gaussian process regression". In: *Journal of Machine Learning Research* 6.12 (2005), pp. 1939–1959. DOI: 10.5555/1046920.1194909.

[Rad96] Neal M. Radford. *Bayesian learning for neural networks*. Springer-Verlag New York, 1996. DOI: 10.1007/978-1-4612-0745-0.

[Ras06] Carl E. Rasmussen. *Gaussian processes for machine learning.* The MIT Press, 2006. ISBN: 9780262182539.

[Rec18] Benjamin Recht. “A tour of reinforcement learning: The view from continuous control”. In: *Annual Review of Control, Robotics, and Autonomous Systems* 2 (2018), pp. 253–279. DOI: `10.1146/annurev-control-053018-023825`.

[RK18] Maziar Raissi and George E. Karniadakis. “Hidden physics models: Machine learning of nonlinear partial differential equations”. In: *Journal of Computational Physics* 357 (2018), pp. 125–141. DOI: `10.1016/j.jcp.2017.11.039`.

[RN10] Carl E. Rasmussen and Hannes Nickisch. “Gaussian Processes for Machine Learning (GPML) Toolbox”. In: *Journal of machine learning research* 11 (2010), pp. 3011–3015. ISSN: 1532-4435.

[Rog+11] Alex Rogers, Sasan Maleki, Siddhartha Ghosh, and Nicholas R. Jennings. “Adaptive Home Heating Control Through Gaussian Process Prediction and Mathematical Programming”. In: *Second International Workshop on Agent Technology for Energy Systems.* 2011, pp. 71–78.

[RWH04] Thambirajah Ravichandran, David W. Wang, and Glenn R. Heppler. “Stability and robustness of a class of nonlinear controllers for robot manipulators”. In: *Proceedings of the 2004 American Control Conference.* Vol. 6. 2004, pp. 5262–5267. DOI: `10.23919/ACC.2004.1384688`.

[San+08] Agostino De Santis, Bruno Siciliano, Alessandro De Luca, and Antonio Bicchi. “An atlas of physical human–robot interaction”. In: *Mechanism and Machine Theory* 43.3 (2008), pp. 253–270. DOI: `10.1016/j.mechmachtheory.2007.03.003`.

[SB13] Josef Stoer and Roland Bulirsch. *Introduction to numerical analysis.* Vol. 12. Springer Science & Business Media, 2013. ISBN: 9781441930064.

[SC04] John Shawe-Taylor and Nello Cristianini. *Kernel methods for pattern analysis.* Cambridge university press, 2004. DOI: `10.1017/CBO9780511809682`.

[SC08] Ingo Steinwart and Andreas Christmann. *Support vector machines.* Springer Science & Business Media, 2008. DOI: `10.1007/978-0-387-77242-4`.

[See99] Matthias Seeger. “Bayesian model selection for support vector machines, Gaussian processes and other kernel classifiers”. In: *Proceedings of the 12th International Conference on Neural Information Processing Systems.* 1999, pp. 603–609. DOI: `10.5555/3009657.3009743`.

[SG06] Edward Snelson and Zoubin Ghahramani. “Sparse Gaussian processes using pseudo-inputs”. In: *Advances in Neural Information Processing Systems 18.* 2006, pp. 1257–1264.

[Sha+16] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando De Freitas. “Taking the human out of the loop: A review of Bayesian optimization”. In: *Proceedings of the IEEE* 104.1 (2016), pp. 148–175. DOI: `10.1109/JPROC.2015.2494218`.

[SHS06] Ingo Steinwart, Don Hush, and Clint Scovel. “An explicit description of the reproducing kernel Hilbert spaces of Gaussian RBF kernels”. In: *IEEE Transactions on Information Theory* 52.10 (2006), pp. 4635–4643. DOI: 10.1109/TIT.2006.881713.

[SHV06] Mark W. Spong, Seth Hutchinson, and Mathukumalli Vidyasagar. *Robot modeling and control*. Vol. 3. John Wiley & Sons, 2006. ISBN: 9780471649908.

[Sic+10] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: modelling, planning and control*. Springer Science+Business Media, 2010. DOI: 10.1007/978-1-84628-642-1.

[Sjö+95] Jonas Sjöberg, Qinghua Zhang, Lennart Ljung, Albert Benveniste, Bernard Delyon, Pierre-Yves Glorennec, Håkan Hjalmarsson, and Anatoli Juditsky. “Nonlinear black-box modeling in system identification: a unified overview”. In: *Automatica* 31.12 (1995), pp. 1691–1724. DOI: 10.1016/0005-1098(95)00120-8.

[SK81] Robert E. Sheldahl and Paul C. Klimas. *Aerodynamic characteristics of seven symmetrical airfoil sections through 180-degree angle of attack for use in aerodynamic analysis of vertical axis wind turbines*. Tech. rep. Sandia National Labs, Albuquerque, USA, 1981. DOI: 10.2172/6548367.

[SL87] Jean-Jacques E. Slotine and Weiping Li. “On the adaptive control of robot manipulators”. In: *The international journal of robotics research* 6.3 (1987), pp. 49–59. DOI: 10.1177/027836498700600303.

[Sri+12] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. “Information-theoretic regret bounds for Gaussian process optimization in the bandit setting”. In: *IEEE Transactions on Information Theory* 58.5 (2012), pp. 3250–3265. DOI: 10.1109/TIT.2011.2182033.

[SS12] Lorenzo Sciavicco and Bruno Siciliano. *Modelling and control of robot manipulators*. Springer Science & Business Media, 2012. DOI: 10.1007/978-1-4471-0449-0.

[Stu98] Bernd Sturmfels. “Polynomial Equations and Convex Polytopes”. In: *The American Mathematical Monthly* 105.10 (1998), pp. 907–922. DOI: 10.1080/00029890.1998.12004987.

[Sui+15] Yanan Sui, Alkis Gotovos, Joel Burdick, and Andreas Krause. “Safe Exploration for Optimization with Gaussian Processes”. In: *Proceedings of the 32nd International Conference on Machine Learning*. 2015, pp. 997–1005.

[Sun96] Rangarajan K. Sundaram. *A first course in optimization theory*. Cambridge university press, 1996. DOI: 10.1017/CBO9780511804526.

[SVD01] Johan A. K. Suykens, Joos Vandewalle, and Bart De Moor. “Optimal control by least squares support vector machines”. In: *Neural networks* 14.1 (2001), pp. 23–35. DOI: 10.1016/S0893-6080(00)00077-0.

[TA81] Morikazu Takegaki and Suguru Arimoto. “A new feedback method for dynamic control of manipulators”. In: *Journal of Dynamic Systems, Measurement, and Control* 103.2 (1981), pp. 119–125. DOI: doi.org/10.1115/1.3139651.

[VH10] Triet N. Van and Noriyuki Hori. “A new discrete-time model for a van del Pol Oscillator”. In: *Proceedings of SICE Annual Conference.* 2010, pp. 2699–2704. ISBN: 9784907764364.

[Vin+16] Julia Vinogradska, Bastian Bischoff, Duy Nguyen-Tuong, Anne Romer, Henner Schmidt, and Jan Peters. “Stability of controllers for Gaussian process forward models”. In: *International Conference on Machine Learning.* 2016, pp. 545–554.

[Wåg+17] Johan Wågberg, Dave Zachariah, Thomas B. Schön, and Petre Stoica. “Prediction performance after learning in Gaussian process regression”. In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics.* Vol. 54. 2017, pp. 1264–1272.

[Wah90] Grace Wahba. *Spline models for observational data.* SIAM, 1990. DOI: 10.1137/1.9781611970128.

[WB88] John T. Wen and David S. Bayard. “New class of control laws for robotic manipulators Part 1. Non–adaptive case”. In: *International Journal of Control* 47.5 (1988), pp. 1361–1385. DOI: 10.1080/00207178808906102.

[WFH08] Jack M. Wang, David J. Fleet, and Aaron Hertzmann. “Gaussian process dynamical models for human motion”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.2 (2008), pp. 283–298. DOI: 10.1109/TPAMI.2007.1167.

[WHB05] Jack Wang, Aaron Hertzmann, and David M. Blei. “Gaussian process dynamical models”. In: *Proceedings of the 18th International Conference on Neural Information Processing System.* 2005, pp. 1441–1448. DOI: 10.5555/2976248.2976429.

[Wil+09] Christopher Williams, Stefan Klanke, Sethu Vijayakumar, and Kian M. Chai. “Multi-task Gaussian Process Learning of Robot Inverse Dynamics”. In: *Advances in Neural Information Processing Systems 21.* Curran Associates, Inc., 2009, pp. 265–272.

[WR96] Christopher K. Williams and Carl E. Rasmussen. “Gaussian processes for regression”. In: *Advances in neural information processing systems.* 1996, pp. 514–520.

[WRK93] Louis L. Whitcomb, Alfred A. Rizzi, and Daniel E. Koditschek. “Comparative experiments with a new adaptive controller for robot arms”. In: *IEEE Transactions on Robotics and Automation* 9.1 (1993), pp. 59–70. DOI: 10.1109/70.210795.