

# Data-efficient and Safe Learning with Gaussian Processes

Von der Fakultät Informatik, Elektrotechnik und Informationstechnik  
der Universität Stuttgart zur Erlangung der Würde eines Doktors  
der Naturwissenschaften (Dr. rer. nat.) genehmigte Abhandlung

Vorgelegt von

**Jens Schreiter**

aus Marienberg

Hauptberichter: Prof. Dr. rer. nat. Marc Toussaint

Mitberichter: Prof. Dr. rer. nat. habil. Thomas Villmann

Tag der mündlichen Prüfung: 24.02.2020

Institut für Parallele und Verteilte Systeme (IPVS)  
der Universität Stuttgart

2020



# Declaration

I, Jens Schreiter, born on 29. August 1985 in Marienberg, hereby declare that this thesis was composed by myself and without any inadmissible external assistance, that all other assistance or aids are explicitly given in the text, and that all passages have been identified which I have directly taken literally or in the sense from other publications. This thesis has not been submitted to any other domestic or foreign university in the supplied or a similar version for any other degree or professional qualification.

Neckartenzlingen, 24. February 2020

Schreiter, Jens



# Acknowledgements

This thesis is the result of my work within the Group for Cognitive Systems in the Corporate Sector Research and Advance Engineering at the Robert Bosch GmbH, located in Renningen near Stuttgart, Germany. From the academic side, this work was supported from the University of Stuttgart.

My thanks go first of all to my supervisor Marc Toussaint, who guided me through my research and whose precious comments inspired me to think differently when considering machine learning problems. I also would like to express my gratitude to the whole Machine Learning and Robotics Laboratory at the University of Stuttgart for the useful discussions and the joint conference visits. In the same breath, I would like to thank my co-supervisor Thomas Villmann and his Computational Intelligence Group at the University of Applied Sciences Mittweida in the east of Chemnitz, Germany, for sharing their time, technical knowledge, experiences, and enabling different points of view. Moreover, I wish to thank all members of the Cognitive Systems Group (CR/AEY2) and their leader, Yasser Jadidi, from the research department at Robert Bosch GmbH. Especially, I would like to thank my Bosch internal supervisors: Heiner Markert, Duy Nguyen-Tuong, Mona Eberts, Bastian Bischoff, as well as Michael Hanselmann, who have supported me throughout my doctoral studies and always had been there for me. I would also like to thank some other members of the project Self Learning Methods in which the requirements for the topic of this thesis were established. Namely Ernst Kloppenburg, Volker Imhof, and Christoph Straehle, who discussed several machine learning and optimization issues with me. Further thanks go to the doctoral students from the CR/AEY2: Laura Beggel, Michael Herman, Julia Vinogradska, and Jörg Wagner, for proofreading and a lot of suggestions for improving this thesis. Of course, all remaining mistakes are my fault. During my Ph.D. studies I have been fortunate to benefit from inspiring collaborations with many other employees of the Robert Bosch GmbH. To list all of these people here would go beyond the scope, but in particular I thank Lothar Baum, Markus Brunk, René Diener, Christian Gosch, Benjamin Hartmann, Steffen Klug, Mark Schillinger and Holger Ulmer. I also met a lot of researchers in the field of machine learning and artificial intelligence at various conferences, workshops, and especially at the Machine Learning Summer School 2014 in Reykjavik, Iceland, who deserve my gratitude. Furthermore, I want to thank my former math teachers Ronny Künast and Christfried Siegert for their comprehensible kind of teaching, since without them I would not be where I am now. For the fast evaluations of my machine learning algorithms on many large scale data sets I thank my reliable HP Z800 Workstation. Finally, my special thanks go to my family and friends who supported me through these many years of research, especially during the final hardest one.



---

# Abstract

Data-based modeling techniques enjoy increasing popularity in many areas of science and technology where traditional approaches are limited regarding accuracy and efficiency. When employing machine learning methods to generate models of dynamic system, it is necessary to consider two important issues. Firstly, the data-sampling process should induce an informative and representative set of points to enable high generalization accuracy of the learned models. Secondly, the algorithmic part for efficient model building is essential for applicability, usability, and the quality of the learned predictive model. This thesis deals with both of these aspects for supervised learning problems, where the interaction between them is exploited to realize an exact and powerful modeling.

After introducing the non-parametric Bayesian modeling approach with Gaussian processes and basics for transient modeling tasks in the next chapter, we dedicate ourselves to extensions of this probabilistic technique to relevant practical requirements in the subsequent chapter. This chapter provides an overview on existing sparse Gaussian process approximations and propose some novel work to increase efficiency and model selection on particularly large training data sets. For example, our sparse modeling approach enables real-time capable prediction performance and efficient learning with low memory requirements. A comprehensive comparison on various real-world problems confirms the proposed contributions and shows a variety of modeling tasks, where approximate Gaussian processes can be successfully applied. Further experiments provide more insight about the whole learning process, and thus a profound understanding of the presented work.

In the fourth chapter, we focus on active learning schemes for safe and information-optimal generation of meaningful data sets. In addition to the exploration behavior of the active learner, the safety issue is considered in our work, since interacting with real systems should not result in damages or even completely destroy it. Here we propose a new model-based active learning framework to solve both tasks simultaneously. As basis for the data-sampling process we employ the presented Gaussian process techniques. Furthermore, we distinguish between static and transient experimental design strategies. Both problems are separately considered in this chapter. Nevertheless, the requirements for each active learning problem are the same. This subdivision into a static and transient setting allows a more problem-specific perspective on the two cases, and thus enables the creation of specially adapted active learning algorithms. Our novel approaches are then investigated for different applications, where a favorable trade-off between safety and exploration is always realized. Theoretical results maintain these evaluations and provide respectable knowledge about the derived model-based active learning schemes. For example, an upper bound for the probability of failure of the presented active learning methods is derived under reasonable assumptions.

Finally, the thesis concludes with a summary of the investigated machine learning problems and motivate some future research directions.





# Kurzfassung

Datenbasierte Modellierungstechniken erfreuen sich zunehmender Beliebtheit in vielen Bereichen von Wissenschaft und Technik, vor allem wenn traditionelle Ansätze an ihre Grenzen bezüglich Genauigkeit und Effizienz stoßen. Bei der Verwendung von maschinellen Lernmethoden zur Erzeugung dynamischer Systemmodelle ist die Berücksichtigung zweier wesentlicher Punkte erforderlich. Auf der einen Seite sollte der Prozess zur Gewinnung von Daten eine informative und repräsentative Menge von Punkten liefern, so dass eine hohe Generalisierungsfähigkeit der damit gelernten Modelle erzielt werden kann. Weiterhin ist auch der algorithmische Teil zur effizienten Modellbildung von großer Bedeutung für die Anwendbarkeit, Benutzerfreundlichkeit und schließlich die Qualität des gelernten Vorhersagemodells. Diese Dissertation betrachtet beide Aspekte von überwachten Lernproblemen, wobei die Wechselwirkung zwischen ihnen ausgenutzt wird, um eine exakte und leistungsfähige Modellierung zu realisieren.

In einem Grundlagenkapitel wird ein nicht-parametrischer Modellierungsansatz für Gauß-Prozesse sowie Hintergründe über transiente Modellierungstechniken vorgestellt. Danach, d.h. im ersten Hauptkapitel, widmen wir uns Erweiterungen zu diesem probabilistischen Ansatz, um relevante Praxisanforderungen zu erfüllen. Darin wird ein Überblick über vorhandene, spärliche Gauß-Prozess-Approximationen gegeben und ein neues Verfahren vorgeschlagen, um die Trainingsgeschwindigkeit und Genauigkeit bei besonders großen Trainingsdatenmengen zu erhöhen. Unser approximativer Modellierungsansatz ermöglicht beispielsweise echtzeitfähige Vorhersagezeiten und effizientes Lernen unter niedrigen Speicherplatzanforderungen. Ein umfassender Vergleich zu verschiedenen Problemstellungen aus der realen Welt bestätigt die Generalisierungsfähigkeit des vorgeschlagenen Algorithmus und zeigt eine Vielzahl von Modellierungsproblemen auf, wo spärliche Gauß-Prozesse erfolgreich angewendet wurden. Zusätzliche Experimente liefern mehr Einblick über den gesamten Lernprozess und damit ein tieferes Verständnis für die vorgestellten probabilistischen Methoden.

Das letzte Hauptkapitel dieser Arbeit konzentriert sich auf die sichere und informationsoptimale Erzeugung von aussagekräftigen Datensätzen mittels aktiver Lernverfahren. Neben dem Explorationsverhalten spielt vor allem die Sicherheit der in dieser Arbeit betrachteten aktiven Lernalgorithmen eine wichtige Rolle, denn bei der Interaktion mit echten technischen Systemen sollten diese Systeme keinen Schaden nehmen oder gar vollständig zerstört werden. Wir schlagen hier einen neuen modelbasierten Ansatz zum aktiven Lernen vor, der beide Problemstellungen gleichzeitig lösen soll. Als Grundlage für diesen Prozess zur Gewinnung von Trainingsdaten werden natürlich die vorgestellten Gauß-Prozess-Verfahren verwendet. Weiterhin können wir zwischen statischen und transienten Versuchsplänen unterscheiden, wobei beide separat innerhalb dieses Kapitels genauer erläutert werden. Dabei bleiben aber die Anforderungen, d.h. vor allem hinsichtlich sicherer Exploration, an die einzelnen aktiven Lernverfahren erhalten. Die Unterteilung in stationäre und transiente Aufgabenstellungen ermöglicht weiterhin eine problemspezifische Perspektive über beide Fälle, so dass speziell angepasste Algorithmen erstellt werden. Unsere neuen Ansätze werden dann bei verschiedenen Anwendungen getestet, wobei sich in den Resultaten stets ein

guter Kompromiss zwischen Exploration und Sicherheit einstellt. Diese Auswertungen werden durch theoretische Erkenntnisse unterstützt, die somit weitere Einblicke in die Funktionsweise unseres modellbasierten aktiven Lernalgorithmus ermöglichen. Beispielweise wird unter angemessenen Bedingungen eine obere Schranke für die Fehlerwahrscheinlichkeit der sicheren aktiven Lerner hergeleitet.

In einer Zusammenfassung werden letztendlich die betrachteten maschinellen Lernprobleme diskutiert und weitere Forschungsrichtungen motiviert.

# Table of Contents

<b>Declaration</b>	<b>i</b>
<b>Acknowledgements</b>	<b>iii</b>
<b>Abstract</b>	<b>v</b>
<b>Kurzfassung</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Outline . . . . .	3
1.3 Previously Published Work . . . . .	4
<b>2 Background</b>	<b>7</b>
2.1 Gaussian Process Models . . . . .	7
2.1.1 Regression . . . . .	8
2.1.2 Classification . . . . .	10
2.1.3 Mean Functions . . . . .	13
2.1.4 Covariance Functions . . . . .	14
2.1.5 Model Selection . . . . .	16
2.2 Error Measures . . . . .	18
2.2.1 Regression . . . . .	18
2.2.2 Classification . . . . .	19
2.3 NARX Approach for Transient Modeling . . . . .	21
2.3.1 Feature Selection . . . . .	21
2.3.2 Prediction . . . . .	22
<b>3 Sparse Gaussian Process Approximations</b>	<b>25</b>
3.1 Related Work . . . . .	25
3.2 Deterministic Training Conditional Approximation . . . . .	28
3.2.1 Expectation Maximization for Model Selection . . . . .	30
3.2.2 State-of-the-art Greedy Insertion and Deletion Criteria . . . . .	31
3.2.3 Efficient Maximum Error Insertion and Deletion . . . . .	35
3.3 Fully Independent Training Conditional Approximation . . . . .	37
3.3.1 Determining Hyperparameters and Pseudo-inputs . . . . .	39
3.4 Evaluations . . . . .	39
3.4.1 Benchmark Data Sets . . . . .	39
3.4.2 Compliant and Real-time Robot Control . . . . .	46
3.4.3 Simulation of Vehicle Power Demand . . . . .	49
3.5 Discussion . . . . .	55

<b>4</b>	<b>Safe Active Learning with Gaussian Process Models</b>	<b>59</b>
4.1	Related Work . . . . .	60
4.2	Active Learning for Stationary Environments . . . . .	61
4.2.1	Exploring Strategy . . . . .	62
4.2.2	Exploring under Uncertainty . . . . .	64
4.2.3	Introducing Safety . . . . .	67
4.2.4	The Algorithm . . . . .	69
4.2.5	Theoretical Analysis . . . . .	70
4.2.6	Evaluations . . . . .	73
4.3	Active Learning for Transient Environments . . . . .	80
4.3.1	Exploring Strategy . . . . .	81
4.3.2	Efficient Trajectory Generation with Bézier Curves . . . . .	83
4.3.3	Determining Safe Trajectories . . . . .	86
4.3.4	The Algorithm . . . . .	91
4.3.5	Evaluations . . . . .	95
4.4	Discussion . . . . .	102
<b>5</b>	<b>Conclusion</b>	<b>107</b>
5.1	Summary . . . . .	107
5.2	Future Work . . . . .	111
<b>A</b>	<b>Appendix</b>	<b>113</b>
A.1	Abbreviations . . . . .	113
A.2	Notations . . . . .	115
A.3	Mathematical Explanations . . . . .	118
A.3.1	Matrix Calculus . . . . .	118
A.3.2	Multivariate Normal Distribution . . . . .	120
A.3.3	Derivations . . . . .	122
A.3.4	Differentiations . . . . .	125
A.3.5	Proofs . . . . .	135
	<b>List of Figures</b>	<b>143</b>
	<b>List of Tables</b>	<b>145</b>
	<b>List of Algorithms</b>	<b>147</b>
	<b>Bibliography</b>	<b>149</b>

# 1 Introduction

In science and technology machine learning approaches are widely used to analyze, describe, and extract knowledge from data. The Robert Bosch GmbH as a leading global supplier of technology and services has a huge interest to establish such mathematical methods for solving problems arising in their industrial and automotive applications. These applications inspire many machine learning tasks in the whole Bosch world. A few of them are considered in this thesis.

## 1.1 Motivation

In many industrial and automotive applications, precise models of the underlying technical systems play an essential role during the research and development process. For instance, accurate models are indispensable for technical design, calibration, control, simulation, and optimization to ensure optimal results and a high level of quality. Conventionally, the modeling of systems is based on physical knowledge and a wide understanding of the system's dynamics behavior. Even today, physical modeling techniques are successfully used for applications in engineering and natural sciences. However, in some tasks physical-based modeling approaches achieve only restricted accuracy and efficiency. Possible reasons for these limitations are complex and non-descriptive system dynamics, e.g. unknown non-linearities, or a costly and time-consuming parameter identification and determination process. Here, an interesting and favored alternative is data-based model generation with machine learning methods. This is advantageous, since for many practical applications responses of the physical system to selected input parameter combinations can easily be measured. A large number of usable data points can thus be acquired in short time spans. Consequently, data-based modeling techniques, more precisely supervised learning schemes, have gained increasing popularity for many of such technical engineering problems. These modern modeling approaches entail exceptional computational challenges and technical requirements, some of which are addressed in this thesis.

Firstly, we focus on the design and capabilities of machine learning algorithms for efficient model building based on large training data sets. Such data sets typically appear when modeling transient systems, that means time-dependent systems with usually complex dynamics. To ensure an appropriate modeling for transient systems, we need high sampling frequencies to capture the systems dynamics. Additionally, for real-time control tasks it may be desirable that the model-based predictions can be made with reasonable computational effort. Therefore, it is necessary to choose a suitable prediction algorithm. Within the wide range of available supervised learning techniques from the field of machine learning, we identified a special class of suitable procedures to satisfy the prevailing conditions. Hence, we take into account probabilistic, non-parametric data-based modeling algorithms, namely Gaussian process (GP) methods. In this context, non-parametric means that inference is directly considered in a previously defined space

of functions. The class of functions is determined by a problem-specific GP prior. This is an advantageous modeling property, because already existing expert knowledge about the technical system can be used to define the prior distribution. Furthermore, standard Bayesian approaches are suitable to fit the GP model and the quality of predictions can be assessed by their standard deviation. By that, the standard GP approach has been successfully used in many industrial, economic, ecological, or automotive applications. However, its applicability is limited to small modeling problems, since its computational complexity and memory requirements grow rapidly with the size of the training data set. In the last few years, a range of various GP approximations arose to overcome these limitations. Here, we consider a special class of sparse GP approximations, because this modeling approach enables short prediction times, manageable storage requirements, and high generalization accuracy. The sparseness of this technique is induced by the fact that a representative subset of the real training data is used for the creation of the sparse GP model. However, the information contained in the whole data set is used, since an information-optimal approximation is the core of this sparse modeling approach. In this thesis, an efficient approach to select a representative subset from the whole training data set is derived and evaluated on different real-world modeling tasks.

In addition to the algorithmic side of modeling, the data-sampling process is important for subsequent model learning. To ensure a good model with high generalization capability and efficiency in model generation, the sampled data should contain as much information about the underlying system as possible. Hence, the collected data has to cover large regions of the input space. Furthermore, small redundancy within the sampled data would be beneficial, but also a higher data density for more relevant input regions, e.g. locations with particularly high non-linearities. Moreover, the process of collecting labeled data, i.e. measuring points of the input space with an associated system response, may be bounded due to constraints on time or cost for operation on the real system. All of these points are well known in the design of experiments (DoE) literature. In the machine learning community, these tasks are assigned to the field of active learning, since these frameworks deal with the problem of selective and guided generation of labeled data. In this supervised setting, an active learner guides the data generation process by choosing new informative samples to be labeled based on the knowledge obtained so far. Especially, active learning for transient systems requires great care when interacting with critical systems, since damaging or even destroying the system should be absolutely prevented. In order to ensure a safe exploration of an unknown system, the active learner additionally has to decide between safe and unsafe regions during the process of collecting new data. Therefore, additional feedback from the technical system about its health status is needed to avoid dangerous actions during exploration. This additional knowledge can be used to create a data-based model on the already explored input space, i.e. we are focusing on model-based active learning, which results in a safe and preferably information-optimal active learner. In this thesis, we present some novel model-based active learning techniques to address the safety and the other issues mentioned above when generating data for several transient and also stationary, i.e. time-invariant, applications. Even when not focusing on data-based modeling, safe active learning schemes provide a favorable basis for other data-dependent tasks, e.g. parameter estimation of physical modeling approaches. For real-world applications, it is necessary to consider practical aspects like measurement noise or sensor properties when designing active learning algorithms, since high data quality should be consistently guaranteed over time, especially when interacting with transient systems.

## 1.2 Outline

This thesis considers two strongly connected topics from the field of machine learning. At the beginning of a data-based modeling approach, the data-sampling process should receive high attention closely followed by the algorithmic constraints which give an intuition for potential modeling techniques. When both aspects are interacting, e.g. for the considered model-based active learning scenarios, the utilization of their relations are responsible for the final result.

Before the scientifically novel parts of this thesis are presented, an introductory chapter provides background information on the standard GP setting for regression and classification tasks. Mainly, the Bayesian way of model selection for these two cases and an overview with relations between the different error measures is explained. Finally, our favored approach for transient modeling of dynamic systems is described there. We hope the interested reader found this slight excursion about modeling with GPs helpful for understanding our work and already delights on these probabilistic models. Additional material with standard abbreviations, mathematical information, notations, conventions, proofs, and references are provided in the Appendix A.

In Chapter 3, we report on sparse GP approximations and their favorable properties when modeling large scale problems. The core of the considered sparsifications is the identification of a representative subset from the whole training data which induces the sparse model approximation. After a review of the state-of-the-art approximation approaches, we present our novel strategies for efficient data-based modeling. Together with a new optimization scheme for Bayesian model selection we are able to accurately adapt the method-specific parameters. A comprehensive comparison on various real-world applications demonstrate that our obtained sparse GP models are competitive with the established state-of-the-art approximations in terms of generalization accuracy. The speed-up in learning time of our method compared to other approximations is evaluated on several large data sets, i.e. with about half a million training points. Furthermore, it is shown that our approximated GP model is sufficiently fast for real-time prediction on such large modeling tasks. A detailed discussion about this remarkable result concludes the chapter.

Our GP based active learning framework is introduced in Chapter 4. Firstly, a brief overview about some previous work related to the considered active learning scenarios is given. After that, we begin with the description of the active learning problem for stationary applications, since the obtained results and experiences on stationary applications build the basis for the transient data-sampling task. For the stationary case, we present an optimal exploration criterion and, of course, the safety issue to avoid the measurement of critical and unsafe points during the sampling process. In a theoretical analysis of our proposed safe active learning algorithm is shown that we can guarantee an upper bound for the probability of failure, i.e. the probability of sampling in an unsafe region of the input space. Moreover, a strategy to select the best model is proposed when uncertainty in the parameters of the model appear. To demonstrate the efficiency and robustness of our safe exploration scheme in the stationary active learning setting, we apply the approach to a simulated toy example and a policy exploration task for the inverted pendulum stability problem. For the transient case, the stationary active learning must be

extended to consider time dependent trajectories, i.e. excitation signals for each input dimension, and their safety when exploring the input space. This dependency on time causes further requirements for an optimal and safe model-based active learning scheme. Therefore, it is necessary to consider trajectory planning algorithms, where the safety and the exploration task of the trajectory for the next time interval can be efficiently evaluated and optimized. This issue is solved with a parametrized trajectory planning approach, which is able to follow system constraints and yields a smooth excitation behavior. To guarantee a good transient experimental design, the already proposed GP approximation technique is employed for dynamic model generation. Experiments on a simulated real-world magnetic valve modeling task provide confident results for our safe and transient active learning scheme, which is presented in the second part of Chapter 4.

Finally, the chapter concludes with a summary of the proposed and experimentally confirmed contributions of the thesis. It also includes a broad overview of further research directions and other interesting questions regarding the mentioned machine learning topics.

## 1.3 Previously Published Work

Firstly, some parts of our work are used as methodical baseline strategy in three already registered patents regarding industrial applications of the Robert Bosch GmbH. This thesis summarizes the former protected contents and strongly expands our previously published work, where more technical insight is provided as well as additional experiments on benchmark data sets are shown. The whole thesis contains only work prepared by the author.

The considered transient modeling approach for dynamic systems, initial work on sparse GP approximations, and evaluations on a large scale automotive application are shown in Schreiter et al. (2013). Theoretically deeper derivations and a broader comparison between approximate GP regression techniques are provided in Schreiter et al. (2015a). In Schreiter et al. (2015b), a sparse GP algorithm for compliant and real-time capable robot control is presented. The journal article by Schreiter et al. (2016) mainly summarizes the contributions of the two former conference papers, where an extensive overview about related work is given. Regarding the active learning part of the thesis, a safe exploration approach with GPs for stationary environments is presented in Schreiter et al. (2015c) with additional results shown in a NIPS Workshop, see Schreiter et al. (2015d). Furthermore, Schillinger et al. (2016) developed an extended approach based on the herein derived safe exploration scheme.

## Publications

- Schreiter, J., Markert, H., Hanselmann, M., Nguyen-Tuong, D., and Bohne, C. (2013). Large Scale Transient Data-based Models for the Simulation of Vehicle Power Demand. In K. Röpke (Ed.),



*Proceedings of the 7th Conference on Design of Experiments (DoE) in Engine Development* (pp. 176 – 189).

- Schreiter, J., Nguyen-Tuong, D., Markert, H., Hanselmann, M., and Toussaint, M. (2015a). Fast Greedy Insertion and Deletion in Sparse Gaussian Process Regression. In M. Verleysen (Ed.), *Proceedings of the 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* (pp. 101 – 106).
- Schreiter, J., Englert, P., Nguyen-Tuong, D., and Toussaint, M. (2015b). Sparse Gaussian Process Regression for Compliant, Real-Time Robot Control. In A. Okamura, and S. Hutchinson (Eds.), *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 2586 – 2591).
- Schreiter, J., Nguyen-Tuong, D., Eberts, M., Bischoff, B., Markert, H., and Toussaint, M. (2015c). Safe Exploration for Active Learning with Gaussian Processes. In A. Bifet, B. Zadrozny, M. May, F. Bonchi, J. Cardoso, M. Spiliopoulou, R. Gavaldà, and D. Pedreschi (Eds.), *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, LNCS (Vol. 9286, pp. 133 – 149).
- Schreiter, J., Nguyen-Tuong, D., and Toussaint, M. (2016). Efficient Sparsification for Gaussian Process Regression. *Neurocomputing*, 192, pp. 29 – 37.
- Schillinger, M., Ortelt, B., Hartmann, B., Schreiter, J., Meister, M., Nguyen-Tuong, D., and Nelles, O. (2016). In K. Leiviskä (Ed.), Safe Active Learning of a High Pressure Fuel Supply System. *Proceedings of the 9th EUROSIM Congress on Modelling and Simulation* (pp. 238 – 243).

## Workshops

- Schreiter, J., Eberts, M., Nguyen-Tuong, D., and Toussaint, M. (2015d). Safe Exploration for Active Learning with Gaussian Process Models. In J. J. Williams, Y. Abbasi, and F. Doshi-Velez (Eds.), *NIPS Workshop on Machine Learning from and for Adaptive User Technologies: From Active Learning & Experimentation to Optimization & Personalization*.

## Patents

- Schreiter, J., Nguyen-Tuong, D., Markert, H., and Hanselmann, M., (2013). ROBERT BOSCH GMBH. Verfahren und Vorrichtung zum Bereitstellen von Stützstellendaten für ein datenbasiertes Funktionsmodell. German Patent, DE 10 2013 213397 A1.
- Schreiter, J., Nguyen-Tuong, D., Markert, H., and Hanselmann, M., (2013). ROBERT BOSCH

GMBH. Verfahren und Vorrichtung zum Adaptieren eines datenbasierten Funktionsmodells. German Patent, DE 10 2013 214967 A1.

- Schreiter, J., Markert, H., and Hanselmann, M., (2015). ROBERT BOSCH GMBH. Verfahren und Vorrichtung zum Berechnen eines datenbasierten Multi-Output-Funktionsmodells. German Patent, DE 10 2015 208513 A1.
- Schreiter, J., Nguyen-Tuong, D., Markert, H., and Eberts, M., Bischoff, B., (2015). ROBERT BOSCH GMBH. Verfahren und Vorrichtung zum Vermessen einer zu testenden Einheit. German Patent, DE 10 2015 216953 A1.

## 2 Background

This chapter provides substantial knowledge about Gaussian process models for regression and classification as well as error measures for both supervised learning problems. Furthermore, the preferred baseline strategy for the applied transient modeling approaches is explained. The informed reader may skip this part of the dissertation and start directly with Chapter 3, which considers sparse Gaussian process approximations.

### 2.1 Gaussian Process Models

Gaussian processes (GPs) are a widely used non-parametric Bayesian modeling technique. A nice and comprehensive overview for the variety of GP models can be found in Rasmussen and Williams (2006). In fact, the subsequent formalism and the notations used are inspired by this book. In contrast to other kernel approaches such as support vector machines (SVMs), see Schölkopf and Smola (2002) for more details, GPs offer a probabilistic framework. The latter leads to predictive distributions for test points and model selection is easy to achieve with standard Bayesian procedures. The non-parametric GP approach employed throughout this thesis considers inference directly in a space of functions. Moreover, non-parametric allows for fewer assumptions to be made about the mostly unknown structure of the learned mathematical model. For example, when modeling with artificial neural networks, cf. Haykin (2008), the basic underlying model structure must be specified. The inference problem is addressed by a GP which describes a distribution over functions. Formally, a GP is defined as a stochastic process, where every finite collection of random variables  $f(\mathbf{x})$  follows a consistent multivariate normal or Gaussian distribution, respectively, i.e.

$$f(\mathbf{x}) \sim \text{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{z})) . \quad (2.1)$$

Since a Gaussian distribution is defined by an expected value and variance, a GP is completely specified by a mean function  $m(\mathbf{x}) = \text{E}[f(\mathbf{x})]$  and a positive semi-definite covariance function  $k(\mathbf{x}, \mathbf{z}) = \text{Cov}[f(\mathbf{x}), f(\mathbf{z})] = \text{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{z}) - m(\mathbf{z}))]$ . Here, the GP (2.1) is always indexed by the set  $\mathbb{X} \subset \mathbb{R}^d$  with its  $d$ -dimensional elements  $\mathbf{x}, \mathbf{z} \in \mathbb{X}$ .

Depending on the underlying supervised modeling task, i.e. regression or classification, the way of taking inference in the associated GP model differs between the both cases. Especially, the selected training data varies due to the continuous output for regression problems and discrete labels for classification tasks, respectively. In the next section, the setting for the regression case followed by the introduction of the slightly different approach for GP classification is described. In addition to commonly used mean and covariance functions, model selection approaches for both learning tasks are presented in the end of this section about modeling with GPs.

### 2.1.1 Regression

Based on the given training data set  $\mathcal{D} = (\mathbf{y}, \mathbf{X})$ , the goal is to build an estimator for the vector of noisy observations  $\mathbf{y} = (y_1, \dots, y_n)^T \in \mathbb{R}^n$  of the underlying regression function  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  with  $f(\mathbf{x}_i) = f_i$ , obeying the relationship

$$y_i = f_i + \varepsilon_i \quad (2.2)$$

with centered Gaussian noise  $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$  and the  $n$  training input  $\mathbf{x}_i = (x_{i1}, \dots, x_{id})^T \in \mathbb{R}^d$  which are row-wise summarized in  $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^T \in \mathbb{R}^{n \times d}$ . For practical applications considered in this thesis the input dimension  $d$  is smaller than the number of training points  $n$ . Under the assumption that the generally unknown function  $f(\mathbf{x})$  follows a GP according to (2.1), the Gaussian prior distribution

$$\mathbb{P}(\mathbf{f} | \mathbf{X}) = \mathcal{N}(\mathbf{f} | \mathbf{m}, \mathbf{K}) \quad (2.3)$$

is defined for all latent function values  $\mathbf{f} = (f_1, \dots, f_n)^T \in \mathbb{R}^n$  of the training data  $\mathbf{X}$ . Here, the vector  $\mathbf{m} = (m_1, \dots, m_n)^T \in \mathbb{R}^n$  consists of the mean function values  $m(\mathbf{x}_i) = m_i$ , cf. Section 2.1.3, and  $\mathbf{K} \in \mathbb{R}^{n \times n}$  is a full covariance matrix between all available training points, which is determined by a specified covariance function  $k(\mathbf{x}_i, \mathbf{x}_j) = k_{ij}$ . The matrix  $\mathbf{K}$  is also known as Gram matrix, see Lipschutz and Lipson (2013). The positive semi-definiteness and the symmetry of  $\mathbf{K}$  are some of the beneficial properties of a covariance function, see Section 2.1.4. Derived from the noisy regression model (2.2), the conditional density

$$\mathbb{p}(\mathbf{y} | \mathbf{f}, \mathbf{X}) = \prod_{i=1}^n \mathcal{N}(y_i | f_i, \sigma^2) = \mathcal{N}(\mathbf{y} | \mathbf{f}, \sigma^2 \mathbf{I}) \quad (2.4)$$

is obtained. The density (2.4) is known as model likelihood of the given training data  $\mathcal{D}$ . Since the noise  $\varepsilon_i$  leads to conditional independence of the targets  $y_i$  given the latent function values  $f_i$ , the likelihood (2.4) factorizes over the observations. The noise-free modeling approach with GPs is described in Rasmussen and Williams (2006). For the regression case considered here, exact inference is possible and analytically tractable such that the posterior density

$$\begin{aligned} \mathbb{p}(\mathbf{f} | \mathbf{y}, \mathbf{X}) &= \frac{\mathbb{p}(\mathbf{y} | \mathbf{f}, \mathbf{X}) \mathbb{p}(\mathbf{f} | \mathbf{X})}{\mathbb{p}(\mathbf{y} | \mathbf{X})} \\ &\propto \mathbb{p}(\mathbf{y} | \mathbf{f}, \mathbf{X}) \mathbb{p}(\mathbf{f} | \mathbf{X}) = \mathcal{N}(\mathbf{y} | \mathbf{f}, \sigma^2 \mathbf{I}) \mathcal{N}(\mathbf{f} | \mathbf{m}, \mathbf{K}) \\ &\propto \mathcal{N}\left(\mathbf{f} \mid \mathbf{m} + \mathbf{K} (\sigma^2 \mathbf{I} + \mathbf{K})^{-1} (\mathbf{y} - \mathbf{m}), \sigma^2 \mathbf{K} (\sigma^2 \mathbf{I} + \mathbf{K})^{-1}\right) \end{aligned} \quad (2.5)$$

follows directly from the theorem of Bayes, cf. Press (2005), and Equation (A.19). The posterior density (2.5) is proportional to the product of the prior (2.3) and the model likelihood (2.4), where the normalization constant is given through the marginal likelihood

$$\begin{aligned} \mathbb{p}(\mathbf{y} | \mathbf{X}) &= \int_{\mathbb{R}^n} \mathbb{p}(\mathbf{y} | \mathbf{f}, \mathbf{X}) \mathbb{p}(\mathbf{f} | \mathbf{X}) \partial \mathbf{f} = \int_{\mathbb{R}^n} \mathcal{N}(\mathbf{y} | \mathbf{f}, \sigma^2 \mathbf{I}) \mathcal{N}(\mathbf{f} | \mathbf{m}, \mathbf{K}) \partial \mathbf{f} \\ &= \mathcal{N}(\mathbf{y} | \mathbf{m}, \sigma^2 \mathbf{I} + \mathbf{K}) \end{aligned} \quad (2.6)$$

of the training data  $\mathcal{D}$  under the specified regression model (2.2). Furthermore, the last step in the calculation of (2.6) follows from Equation (A.21). The goal of the trained GP model is to estimate the

distribution of test points  $\mathbf{x}_* \in \mathbb{R}^d$ . The associated predictive density for a new, so far unseen test point is given by

$$\begin{aligned} p(y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}) &= \int_{\mathbb{R}^n} p(y_* | \mathbf{x}_*, \mathbf{f}, \mathbf{X}) p(\mathbf{f} | \mathbf{y}, \mathbf{X}) \partial \mathbf{f} \\ &= \int_{\mathbb{R}^n} \mathcal{N}(y_* | m_* + \mathbf{k}_*^T \mathbf{K}^{-1} (\mathbf{f} - \mathbf{m}), k_{**} - \mathbf{k}_*^T \mathbf{K}^{-1} \mathbf{k}_* + \sigma^2) \\ &\quad \cdot \mathcal{N}(\mathbf{f} | \mathbf{m} + \mathbf{K} (\sigma^2 + \mathbf{K})^{-1} (\mathbf{y} - \mathbf{m}), \sigma^2 \mathbf{K} (\sigma^2 \mathbf{I} + \mathbf{K})^{-1}) d\mathbf{f} \\ &= \mathcal{N}(y_* | m_* + \mathbf{k}_*^T (\sigma^2 \mathbf{I} + \mathbf{K})^{-1} (\mathbf{y} - \mathbf{m}), k_{**} - \mathbf{k}_*^T (\sigma^2 \mathbf{I} + \mathbf{K})^{-1} \mathbf{k}_* + \sigma^2), \end{aligned} \quad (2.7)$$

where Equation (A.21) is used in the last step. Moreover, in the first step  $p(y_* | \mathbf{x}_*, \mathbf{f}, \mathbf{X})$  follows from the joint multivariate normal distribution

$$\begin{pmatrix} \mathbf{f} | \mathbf{X} \\ y_* | \mathbf{x}_* \end{pmatrix} \sim \mathcal{N} \left( \begin{pmatrix} \mathbf{f} \\ y_* \end{pmatrix} \middle| \begin{pmatrix} \mathbf{m} \\ m_* \end{pmatrix}, \begin{pmatrix} \mathbf{K} & \mathbf{k}_* \\ \mathbf{k}_*^T & k_{**} + \sigma^2 \end{pmatrix} \right)$$

by conditioning on the Gaussian prior (2.3) with Equation (A.22). Here  $\mathbf{k}_* \in \mathbb{R}^n$  contains the covariance function values between the test point  $\mathbf{x}_*$  and all training points  $\mathbf{X}$ . For the implementation of the GP model, it is common practice to employ the Cholesky decomposition (A.1) of the covariance matrix together with the noise term, i.e.  $\sigma^2 \mathbf{I} + \mathbf{K} = \mathbf{L} \mathbf{L}^T$ . Thus, we get the updated expressions for the posterior distribution (2.5)

$$P(\mathbf{f} | \mathbf{y}, \mathbf{X}) = \mathcal{N}(\mathbf{f} | \mathbf{y} - \sigma^2 \boldsymbol{\alpha}, \sigma^2 (\mathbf{I} - \sigma^2 \mathbf{L}^{-T} \mathbf{L}^{-1})) \quad (2.8)$$

and the predictive distribution (2.7)

$$P(y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}) = \mathcal{N}(y_* | m_* + \mathbf{k}_*^T \boldsymbol{\alpha}, k_{**} - \|\mathbf{L}^{-1} \mathbf{k}_*\|^2 + \sigma^2), \quad (2.9)$$

where the prediction vector  $\boldsymbol{\alpha} \in \mathbb{R}^n$  is defined as  $\boldsymbol{\alpha} = \mathbf{L}^{-T} \mathbf{L}^{-1} (\mathbf{y} - \mathbf{m})$ .

The determination of the prior GP (2.3) is essential for the quality of the trained GP model, since the specified mean and covariance function directly influence the posterior GP. Moreover, GPs and various machine learning methods, e.g. support vector regression (SVR) as employed in Smola and Schölkopf (1998), have in common the structure of the predictive mean from (2.9). This baseline connection is already pointed out in the representer theorem by Kimeldorf and Wahba (1970), or more generally in Schölkopf et al. (2001). It follows that the mean prediction of an unknown test point is the superposition of  $n$  weighted covariance functions which depend on the available training data.

The prediction of unknown target values for new test points is only reasonable within the region of the underlying input training data  $\mathbf{X}$ . Outside of this region, the mean prediction falls back to the mean of the prior GP (2.3). Thus, a well-chosen mean function can positively influence the extrapolation behavior. How fast the prediction of test points coincides with the specified GP prior depends further on the determined covariance function. An indicator for the uncertainty of the GP predictions for test points  $\mathbf{x}_* \in \mathbb{R}^d$  is their predictive variance. Note that the predictive variance of the distribution (2.9) is bounded by

$$\text{Var}_p[y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}] \leq \sup_{\mathbf{x}_*} (k_{**}) + \sigma^2, \quad (2.10)$$

if the supremum exists. Based on the ratio of the predicted variance and the maximal possible variance (2.10), it is estimable and finally decidable whether the mean prediction is useful or not.

The computational effort to train a GP regression (GPR) model is  $\mathcal{O}(n^3)$ , which is mainly caused by the Cholesky decomposition of the covariance matrix. Thereby, we need  $\mathcal{O}(n^2)$  memory requirements to store the full covariance matrix  $\mathbf{K}$ . Practically, this leads to challenging problems for large data sets, which are addressed in the next chapter. For the prediction of one test point, we end up with  $\mathcal{O}(dn)$  cost for mean and  $\mathcal{O}(n^2)$  cost for variance calculation, where we assumed that one evaluation of the covariance function leads to effort of  $\mathcal{O}(d)$ . The preceding assessments are crucial for the real-time usability of the GP model.

## 2.1.2 Classification

In contrast to the regression task, classification issues have to deal with discrete class labels  $c(\mathbf{x}_i) = c_i$  instead of the continuous output  $y_i$  for associated input points  $\mathbf{x}_i \in \mathbb{R}^d$ . Here, the binary classification problem is considered, where the class labels  $c_i$  can take values in  $\{+1, -1\}$  for all  $i = 1, \dots, n$ . For this problem a probabilistic classifier based on a GP model is learned. The construction of the GP classification (GPC) model is realized with the same GP prior (2.3) as in the regression case for all latent function values  $g(\mathbf{x}_i) = g_i$ , resulting in the conditional distribution  $\mathbf{g} | \mathbf{X} \sim \mathbf{N}(\mathbf{g} | \mathbf{m}, \mathbf{K})$ . Furthermore, it is assumed that the class labels  $c_i$  are conditionally independent for different function values  $g_i$  and generalized Bernoulli distributed with probability  $\pi_i = \Pr[c_i = +1 | g_i, \mathbf{x}_i]$  yielding  $c_i \sim \text{Ber}(\pi_i)$ . The goal is to model these class probabilities, which is realized by transforming the values  $g_i \in \mathbb{R}$  to the interval  $[0, 1]$  through a response function  $s(z)$  with  $z \in \mathbb{R}$ . Usually, the response function is a symmetric sigmoid function, i.e. a function with the property  $s(z) = 1 - s(-z)$ . Examples for suchlike functions to design class affiliations are the logistic fraction

$$\text{sgd}(z) = \frac{1}{1 + \exp(-z)}$$

or the probit function

$$\Phi(z) = \int_{-\infty}^z \mathcal{N}(t) \partial t . \quad (2.11)$$

In the previous equation and throughout the thesis  $\mathcal{N}(t)$  with  $t \in \mathbb{R}$  is equal to the density function of the standard normal distribution, cf. to Section A.2. The assumptions above and the probit function (2.11) is employed throughout this thesis to yield the GPC model likelihood

$$p(\mathbf{c} | \mathbf{g}, \mathbf{X}) = \prod_{i=1}^n \left( \frac{1 - c_i}{2} + c_i \pi_i \right) = \prod_{i=1}^n s(c_i g_i) = \prod_{i=1}^n \Phi(c_i g_i) , \quad (2.12)$$

where again all labels and input points are summarized in a training data set  $\mathcal{D} = (\mathbf{c}, \mathbf{X})$  of size  $n$ . The presented GPC model relies on the so-called discriminative approach, since the modeling of  $p(\mathbf{c} | \mathbf{g}, \mathbf{X})$  is considered immediately. Due to the non-Gaussian model likelihood (2.12), the exact posterior distribution  $p(\mathbf{g} | \mathbf{c}, \mathbf{X})$  is also non-Gaussian. Therefore, an analytic derivation of the posterior GP is not possible.

Thus, a variety of binary GPC approximation techniques arose in the last decades, see Nickisch and Rasmussen (2008). Throughout this thesis the Laplace approximation by Williams and Barber (1998) is employed. Alternatively, it is possible to treat the classification problem as a regression task and use standard GPR for modeling and prediction. This simple principle is called least-squares classification, cf. Rasmussen and Williams (2006). In contrast, the Laplace approximation induces an approximation of the non-Gaussian posterior with a Gaussian one resulting in

$$p(\mathbf{g} | \mathbf{c}, \mathbf{X}) \approx q(\mathbf{g} | \mathbf{c}, \mathbf{X}) = \mathcal{N}(\mathbf{g} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) , \quad (2.13)$$

which is derived by a second-order Taylor expansion around the maximum of the logarithmic posterior  $\log(p(\mathbf{g} | \mathbf{c}, \mathbf{X}))$ , where the expectation vector is defined as

$$\boldsymbol{\mu} = \arg \max_{\mathbf{g}} \left( \log(p(\mathbf{g} | \mathbf{c}, \mathbf{X})) \right) \in \mathbb{R}^n \quad (2.14)$$

and the covariance matrix through the negative inverse Hessian at the maximum, i.e.

$$\boldsymbol{\Sigma} = - \left( \frac{\partial^2 \log(p(\mathbf{g} | \mathbf{c}, \mathbf{X}))}{\partial \mathbf{g} \partial \mathbf{g}^T} \Big|_{\mathbf{g}=\boldsymbol{\mu}} \right)^{-1} \in \mathbb{R}^{n \times n} . \quad (2.15)$$

With some forethought to the upcoming derivations the function

$$\begin{aligned} \Psi(\mathbf{g}) &= \log(p(\mathbf{c} | \mathbf{g}, \mathbf{X})) + \log(p(\mathbf{g} | \mathbf{X})) \\ &= \sum_{i=1}^n \log(\Phi(c_i g_i)) - \frac{1}{2} \log(|2\pi \mathbf{K}|) - \frac{1}{2} (\mathbf{g} - \mathbf{m})^T \mathbf{K}^{-1} (\mathbf{g} - \mathbf{m}) \end{aligned} \quad (2.16)$$

is defined. The function  $\Psi(\mathbf{g})$  is based on the posterior calculation of  $p(\mathbf{g} | \mathbf{c}, \mathbf{X})$  from Equation (2.13) over the theorem of Bayes, cf. Press (2005), with a Gaussian prior for  $\mathbf{g}$  analog to (2.3) and the model likelihood (2.12), where the influence of the marginal likelihood is not considered since it does not depend on  $\mathbf{g}$ . In the following, the self-consistent equation

$$\begin{aligned} \frac{\partial \Psi(\mathbf{g})}{\partial \mathbf{g}} &= \frac{\partial \log(p(\mathbf{c} | \mathbf{g}, \mathbf{X}))}{\partial \mathbf{g}} + \frac{\partial \log(p(\mathbf{g} | \mathbf{X}))}{\partial \mathbf{g}} \\ &= \left( \bigoplus_{i=1}^n \frac{c_i \mathcal{N}(c_i g_i)}{\Phi(c_i g_i)} \right)^T - (\mathbf{g} - \mathbf{m})^T \mathbf{K}^{-1} \stackrel{!}{=} \mathbf{0}^T , \end{aligned} \quad (2.17)$$

is required to incorporate the necessary condition for the maximum. Thereby, the circled plus  $\bigoplus$  describes the concatenation of the individual partial derivatives and, for the remainder of this thesis, the gradient is always defined as a row vector. Moreover, the Hessian

$$\begin{aligned} \frac{\partial^2 \Psi(\mathbf{g})}{\partial \mathbf{g} \partial \mathbf{g}^T} &= \frac{\partial^2 \log(p(\mathbf{c} | \mathbf{g}, \mathbf{X}))}{\partial \mathbf{g} \partial \mathbf{g}^T} + \frac{\partial^2 \log(p(\mathbf{g} | \mathbf{X}))}{\partial \mathbf{g} \partial \mathbf{g}^T} \\ &= -\mathbf{W} - \mathbf{K}^{-1} \end{aligned}$$

leads to the approximated posterior covariance matrix  $\boldsymbol{\Sigma} = (\mathbf{W} + \mathbf{K}^{-1})^{-1}$  with the shorthand notation of the diagonal matrix

$$\mathbf{W} = \text{diag} \left( \bigoplus_{i=1}^n \left( \frac{\mathcal{N}(c_i g_i)^2}{\Phi(c_i g_i)^2} + \frac{c_i g_i \mathcal{N}(c_i g_i)}{\Phi(c_i g_i)} \right) \right) \in \mathbb{R}^{n \times n} . \quad (2.18)$$

To determine the posterior mean  $\boldsymbol{\mu}$ , Newton iterations based on the self-consistent Equation (2.17) are carried out and we obtain

$$\begin{aligned}\boldsymbol{\mu}_{\text{New}} &= \boldsymbol{\mu} - \left( \frac{\partial^2 \Psi(\mathbf{g})}{\partial \mathbf{g} \partial \mathbf{g}^T} \Big|_{\mathbf{g}=\boldsymbol{\mu}} \right)^{-1} \left( \frac{\partial \Psi(\mathbf{g})}{\partial \mathbf{g}} \Big|_{\mathbf{g}=\boldsymbol{\mu}} \right)^T \\ &= \mathbf{m} + \boldsymbol{\Sigma} \left( \mathbf{W}(\boldsymbol{\mu} - \mathbf{m}) + \left( \bigoplus_{i=1}^n \frac{c_i \mathcal{N}(c_i \boldsymbol{\mu}_i)}{\Phi(c_i \boldsymbol{\mu}_i)} \right) \right),\end{aligned}\quad (2.19)$$

where convergence is mostly reached within a few steps, cf. Rasmussen and Williams (2006). Due to the Gaussian posterior approximation (2.13) the integral for the calculation of the approximated predictive distribution for test points  $\mathbf{x}_* \in \mathbb{R}^d$  is now analytically tractable and results together with the equations (A.14) and (A.21) in

$$\begin{aligned}q(g_* | \mathbf{x}_*, \mathbf{c}, \mathbf{X}) &= \int_{\mathbb{R}^n} p(g_* | \mathbf{x}_*, \mathbf{g}, \mathbf{X}) q(\mathbf{g} | \mathbf{c}, \mathbf{X}) \partial \mathbf{g} \\ &= \int_{\mathbb{R}^n} \mathcal{N}\left(g_* \mid m_* + \mathbf{k}_*^T \mathbf{K}^{-1}(\mathbf{g} - \mathbf{m}), k_{**} - \mathbf{k}_*^T \mathbf{K}^{-1} \mathbf{k}_*\right) \mathcal{N}(\mathbf{g} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) \partial \mathbf{g} \\ &= \mathcal{N}\left(g_* \mid m_* + \mathbf{k}_*^T \boldsymbol{\alpha}, k_{**} - \|\mathbf{L}^{-1} \mathbf{W}^{\frac{1}{2}} \mathbf{k}_*\|^2\right),\end{aligned}\quad (2.20)$$

where  $p(g_* | \mathbf{x}_*, \mathbf{g}, \mathbf{X})$  follows from the joint multivariate normal distribution

$$\begin{pmatrix} \mathbf{g} | \mathbf{X} \\ g_* | \mathbf{x}_* \end{pmatrix} \sim \text{N} \left( \begin{pmatrix} \mathbf{g} \\ g_* \end{pmatrix} \mid \begin{pmatrix} \mathbf{m} \\ m_* \end{pmatrix}, \begin{pmatrix} \mathbf{K} & \mathbf{k}_* \\ \mathbf{k}_*^T & k_{**} \end{pmatrix} \right)$$

by conditioning on the specified Gaussian prior using Equation (A.22). Similar to the regression case, in Equation (2.20) the prediction vector is set to  $\boldsymbol{\alpha} = \mathbf{K}^{-1}(\boldsymbol{\mu} - \mathbf{m}) \in \mathbb{R}^n$  and the Cholesky factorization  $\mathbf{L}\mathbf{L}^T = \mathbf{I} + \mathbf{W}^{\frac{1}{2}} \mathbf{K} \mathbf{W}^{\frac{1}{2}} \in \mathbb{R}^{n \times n}$  is used as described in (A.1). The structure of the predictive density (2.20) is comparable to the regression task and has a connection to the representer theorem by Kimeldorf and Wahba (1970). Finally, with (2.20) the approximated predictive class probability for test points  $\mathbf{x}_*$  follows by solving the one-dimensional integral

$$\begin{aligned}\pi_* &= \Pr_q [c_* = +1 | \mathbf{x}_*, \mathbf{c}, \mathbf{X}] \\ &= \int_{\mathbb{R}} \Phi(g_*) q(g_* | \mathbf{x}_*, \mathbf{c}, \mathbf{X}) \partial g_* \\ &= \Phi \left( \frac{\mathbb{E}_q [g_* | \mathbf{x}_*, \mathbf{c}, \mathbf{X}]}{\sqrt{1 + \text{Var}_q [g_* | \mathbf{x}_*, \mathbf{c}, \mathbf{X}]}} \right) \\ &= \Phi \left( \frac{m_* + \mathbf{k}_*^T \boldsymbol{\alpha}}{\sqrt{1 + k_{**} - \|\mathbf{L}^{-1} \mathbf{W}^{\frac{1}{2}} \mathbf{k}_*\|^2}} \right),\end{aligned}\quad (2.21)$$

which is analytically tractable for the probit function (2.11), see Rasmussen and Williams (2006). Regarding the generalized Bernoulli distribution of the class labels it is common practice to write  $c_* \sim \text{Ber}(\pi_*)$ .

The quality of the predicted probabilities depends on the correctness of the Laplace approximation, which typically underestimates the posterior moments, cf. Nickisch and Rasmussen (2008). Practically,



the training of a GPC model under the Laplace approximation can be relatively fast in real-world implementations and the slight underestimation induces conservative predictions.

If the prediction vector  $\boldsymbol{\alpha}$  and the Cholesky factor  $\mathbf{L}$  are precomputed after the model training, which has the same cubic complexity as the GPR approach, the prediction of the class probabilities is feasible in  $\mathcal{O}(n^2)$ . This cost is mainly caused by the dependency of  $\pi_*$  on the predicted variance as shown in Equation (2.21). However, it suffices to evaluate the predictive mean in Equation (2.20), which is linear in the number of training points, if only the most probable class label is of interest. This behavior relies on fact that  $\pi_* = \frac{1}{2}$  exactly when  $\mathbb{E}_{\mathbf{q}}[g_* | \mathbf{x}_*, \mathbf{c}, \mathbf{X}] = 0$ , because in this case the test input  $\mathbf{x}_*$  lies directly on the classification boundary. Analogously to the regression task, when leaving the training data region, the prediction of class probabilities for unknown test points is no longer representative and the specified GP prior gains influence on the class assignment. For example, when employing a zero mean function, the class probabilities will approximately become fifty percent far away from the training data.

### 2.1.3 Mean Functions

As previously mentioned, the specified GP prior plays an essential role for the underlying modeling task. In this way, the determination of the mean function is the first option of the user to strongly influence the learning quality in both supervised learning tasks. By means of the properties of the normal distribution it is possible to center the prior GP, i.e. that the associated expectation vanishes everywhere. Thus, for the GP prior (2.1) it holds true that

$$f(\mathbf{x}) - m(\mathbf{x}) \sim \text{GP}(0, k(\mathbf{x}, \mathbf{z})) \quad (2.22)$$

for a previously determined mean function  $m(\mathbf{x})$ . This property simplifies the proofs of certain relationships without any restrictions and is frequently exploited throughout this thesis. A mean function should always be specified with subject to the modeling problem, that means according to properties of the function which will be approximated and the given training data  $\mathcal{D}$ . Thus, the mean function offers an opportunity to include expert knowledge for improving model quality and exploration behavior. If nothing is known about the functional behavior, which is generally the case in most of the modeling tasks, a constant mean function

$$m(\mathbf{x}) = a_0 \quad (2.23)$$

with  $a_0 \in \mathbb{R}$  is a good choice. Specifically, this covers the case of a zero mean function, i.e.  $a_0 = 0$ , and will thus not disturb the GP model learning. To capture linear behavior during the training process, a linear mean function

$$m(\mathbf{x}) = a_0 + \mathbf{a}^T \mathbf{x} \quad (2.24)$$

with the parameter vector  $\mathbf{a} \in \mathbb{R}^d$  and the constant  $a_0$  from Equation (2.23) can be used. Furthermore, it is possible to expand the mean function with other terms, for example with quadratic and mixed linear terms induced by  $\mathbf{x}^T \mathbf{A} \mathbf{x}$  with the parameter matrix  $\mathbf{A} \in \mathbb{R}^{d \times d}$ . For a high number of input dimensions  $d$  it is recommended to use a low-rank approximation of the matrix  $\mathbf{A}$  to reduce the number

of parameters, see Lipschutz and Lipson (2013). Another approach using weighted basis functions is presented in Rasmussen and Williams (2006). The parameters arising from the mean function, e.g.  $a_0$ ,  $\mathbf{a}$ ,  $\mathbf{A}$  and potentially others, are the so-called hyperparameters of the GP model and can be adapted as explained in Section 2.1.5. Generally, there are many ways of designing mean functions, but they should be suitable for the modeling task and compatible with the employed covariance function. Compared to Rasmussen and Ghahramani (2001), a balanced ratio between the number of all hyperparameters and the available training data points should be respected.

## 2.1.4 Covariance Functions

The covariance function  $k(\mathbf{x}, \mathbf{z})$  plays the main role in a GP model. In fact, it describes the similarity of random variables, that means between the latent function values  $f(\mathbf{x})$  and  $f(\mathbf{z})$  depending on their  $d$ -dimensional input points  $\mathbf{x}$  and  $\mathbf{z}$ . This relationship forms the basic assumption for data-based modeling techniques, where similar input points yield similar output values. From the stochastic point of view, a covariance function is per definition symmetric and positive semi-definite. More generally, a function

$$k(\mathbf{x}, \mathbf{z}) : \mathbb{V} \times \mathbb{V} \rightarrow \mathbb{C} \quad (2.25)$$

is called a kernel, if there exists a mapping  $\phi(\mathbf{x}) : \mathbb{V} \rightarrow \mathbb{H}$  to a Hilbert space  $\mathbb{H}$  such that

$$k(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle_{\mathbb{H}}$$

induces a scalar product in  $\mathbb{H}$  where  $\mathbb{V}$  is a  $d$ -dimensional metric space. As shown by Aronszajn (1950), a function  $k(\mathbf{x}, \mathbf{z})$  is a kernel, if and only if it is symmetric and positive semi-definite. Previous work for deriving this relationship is done by Mercer (1909) and Moore (1935). Thus, the description of a covariance function as Mercer kernel, or simply as kernel, is well established and synonymously used in this thesis. For the herein considered real-world modeling problems, all further described kernels are defined using  $\mathbb{V} = \mathbb{R}^d$  and result in real covariance values. Moreover, the employed kernels are highly non-linear so that they induce an infinite dimensional Hilbert space  $\mathbb{H}$  which makes them applicable to many learning problems.

A covariance function depending only on  $\mathbf{x} - \mathbf{z}$  is called stationary, which means that it is invariant under translations of the input space. A GP (2.1) with a stationary covariance function  $k(\mathbf{x} - \mathbf{z})$  and a constant mean function  $m(\mathbf{x}) = a_0$  is defined as weak stationary. An example for a stationary covariance function is the squared exponential (SE) kernel

$$k_{\text{SE}}(\mathbf{x} - \mathbf{z}) = \sigma_f^2 \exp\left(-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{2\lambda^2}\right), \quad (2.26)$$

where  $\|\mathbf{x} - \mathbf{z}\|$  denotes the Euclidean norm of the difference between the  $d$ -dimensional vectors  $\mathbf{x}$  and  $\mathbf{z}$ . The prefactor  $\sigma_f$  is known as magnitude, since it scales the values of the Gaussian, and thus the range of the GP prior. The characteristic length-scale  $\lambda$  describes the width of the Gaussian curve, and hence the distance-based similarity between the input points. The positive parameters  $\sigma_f$  and  $\lambda$  are also called hyperparameters of the GP model. Furthermore, the stationary covariance function (2.26) is isotropic, since it is only a function of the distance  $\|\mathbf{x} - \mathbf{z}\|$ . Generally, isotropic covariance functions

are independent under rotations of the input space. Hence, isotropic kernels are also known as radial basis functions (RBFs). Because the SE covariance function is very smooth and infinitely continuous differentiable, the resulting GP is infinitely mean square differentiable, see Rasmussen and Williams (2006). Practically, the high degree of smoothness of the resulting GPs may not be beneficial for many physical modeling tasks. Nevertheless, the SE kernel is the common choice for a lot of applications. For a high number of input dimensions it is advantageous to introduce a length-scale parameter for each dimension. Compared to the isotropic covariance function (2.26), a relevance scaling between all input dimensions is introduced since this aspect corresponds to a weighted Euclidean distance within the exponential term, cf. Rasmussen and Williams (2006). Finally, this extended approach results in the SE covariance function

$$k_{\text{SEARD}}(\mathbf{x} - \mathbf{z}) = \sigma_f^2 \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{z})^T \mathbf{\Lambda}^{-2}(\mathbf{x} - \mathbf{z})\right) \quad (2.27)$$

with automatic relevance determination (ARD), cf. Neal (1996). The diagonal matrix  $\mathbf{\Lambda} = \text{diag}(\boldsymbol{\lambda})$  contains the vector of characteristic length-scales  $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_d)^T \in \mathbb{R}^d$ . A more generic approach is to use the Mahalanobis distance, which results in a fully parametrized symmetric positive definite matrix  $\mathbf{\Lambda}$ , see Hastie et al. (2009). The anisotropic covariance function (2.27) preserves the high smoothness and infinitely continuous differentiability.

Another isotropic covariance function is derived by Matérn (1986) and results in

$$k_{\text{Matérn}}(\mathbf{x} - \mathbf{z}) = \sigma_f^2 \frac{2^{1-\varrho}}{\Gamma(\varrho)} \left(\sqrt{2\varrho} \frac{\|\mathbf{x} - \mathbf{z}\|}{\lambda}\right)^\varrho B_\varrho \left(\sqrt{2\varrho} \frac{\|\mathbf{x} - \mathbf{z}\|}{\lambda}\right) \quad (2.28)$$

with magnitude  $\sigma_f$  and length-scale  $\lambda$  as defined above. The gamma function  $\Gamma(\varrho)$  and the modified Bessel function of second kind  $B_\varrho$  are explained in Abramowitz and Stegun (1972). Additionally, the parameter  $\varrho$  influences the differentiability of the covariance function. Thus, the resulting stochastic process is exactly  $r$ -times mean square differentiable if and only if  $\varrho > r$ . For the limit  $\varrho \rightarrow \infty$ , the Matérn kernel converges to the isotropic SE kernel (2.26), cf. Rasmussen and Williams (2006). In the special case of  $\varrho = \frac{1}{2}$ , the Matérn kernel is equivalent to the exponential covariance function

$$k_{\text{Exp}}(\mathbf{x} - \mathbf{z}) = \sigma_f^2 \exp\left(-\frac{\|\mathbf{x} - \mathbf{z}\|}{\lambda}\right) \quad (2.29)$$

with the same hyperparameters as above, cf. Abramowitz and Stegun (1972). A centered GP with exponential covariance function is, for one-dimensional input points, known as Ornstein-Uhlenbeck process, see Uhlenbeck and Ornstein (1930). The Ornstein-Uhlenbeck process is used as mathematical model in physical applications to describe the velocity of particles under Brownian motion, cf. Grimmett and Stirzaker (2001). Obviously, it is possible to introduce the ARD framework in the covariance functions of the Matérn class.

At the end of this section, a non-stationary covariance function is discussed. The neural network (NN) covariance function with ARD is given by

$$k_{\text{NNARD}}(\mathbf{x}, \mathbf{z}) = \sigma_f^2 \arcsin\left(\frac{1 + \mathbf{x}^T \mathbf{\Lambda}^{-2} \mathbf{z}}{\sqrt{2 + \mathbf{x}^T \mathbf{\Lambda}^{-2} \mathbf{x}} \sqrt{2 + \mathbf{z}^T \mathbf{\Lambda}^{-2} \mathbf{z}}}\right), \quad (2.30)$$

where the hyperparameters  $\sigma_f$  and  $\mathbf{\Lambda}$  are equivalently defined as in Equation (2.27). Due to the non-stationary behavior of the kernel (2.30), the resulting covariance values depend on the norm of the input, i.e. a translation of the input space has great influence on the modeling behavior. This fact can be exploited to gain higher modeling flexibility when a translation vector  $\mathbf{c} \in \mathbb{R}^d$  is introduced and treated as additional hyperparameter. Thus, the scalar products in the NN covariance function tend to the value of  $(\mathbf{x} - \mathbf{c})^T \mathbf{\Lambda}^{-2} (\mathbf{z} - \mathbf{c})$ . Nevertheless, the model selection complexity is also increased in this case. The designation and structure of this covariance function comes from the Bayesian treatment of artificial neural networks (ANN), see Neal (1996). Under certain conditions, like special parameter distributions as employed by Williams (1998), Neal has shown that an ANN with one hidden layer and infinitely many units converges to a GP with this type of non-stationary covariance function.

In addition to the covariance functions mentioned above, there exist many more covariance functions that are isotropic, stationary, non-stationary, periodic and so on. For complex learning tasks a combination of different kernels can be meaningful, as for example the sum or product of numerous covariance functions. A broad summary with extensive discussions of various covariance functions is provided by Rasmussen and Williams (2006).

### 2.1.5 Model Selection

In this work, the considered GP modeling approaches are non-parametric according to the structure of the inference task. This approach is also known as function-space view, since inference is directly considered in the space of functions, cf. Rasmussen and Williams (2006). However, the mean and covariance functions are affected by a large amount of so-called hyperparameters. For notational convenience the dependence of the above formulas on the hyperparameters was neglected. Furthermore, all of the hyperparameters are summarized in the vector  $\boldsymbol{\theta}$ . Generally, the adaption of the hyperparameters based on the available training data is essential for the quality of the final GP model. In the next sections, the Bayesian model selection problem is separately solved with marginal likelihood maximization approaches for both supervised learning tasks.

### Regression

Besides the hyperparameters of the mean and covariance function, the noise  $\sigma^2$  of the regression model (2.2) is also treated as hyperparameter and added to  $\boldsymbol{\theta}$ . The marginal likelihood  $p(\mathbf{y} | \mathbf{X}, \boldsymbol{\theta})$  for the regression case is given in Equation (2.6) and optimized by using the maximum likelihood method (MLM). This results in the optimization problem

$$\varphi(\boldsymbol{\theta}) = \log(p(\mathbf{y} | \mathbf{X}, \boldsymbol{\theta})) \rightarrow \max_{\boldsymbol{\theta}} \quad (2.31)$$

for determining an optimal set of hyperparameters  $\boldsymbol{\theta}$ . The derivation of the logarithmic marginal likelihood, also known as logarithmic evidence

$$\varphi(\boldsymbol{\theta}) = -\frac{n}{2} \log(2\pi) - \sum_{i=1}^n \log(l_{ii}) - \frac{1}{2} (\mathbf{y} - \mathbf{m})^T \boldsymbol{\alpha} \quad (2.32)$$

is shown in the appendix, see Equation (A.32). Seeger (2007) proves that the logarithmic marginal likelihood (2.32) is concave and unimodal with respect to  $\mathbf{y}$ . But the influence of the hyperparameters is much more complex and can lead to hard maximization problems with many local optima depending on the given training data. However, gradient-based optimization methods, e.g. conjugate gradients (CG) solver as described by Geiger and Kanzow (1999), are often and successfully used. Thus, gradient-based optimization techniques are also employed in this thesis, where the partial derivatives of  $\varphi(\boldsymbol{\theta})$  according to the hyperparameters  $\boldsymbol{\theta}$  presented in Section A.3.4 of the appendix, cf. Rasmussen and Williams (2006). The partial derivatives concerning the positive hyperparameters, e.g.  $\sigma_f$  and  $\lambda$ , are taken with respect to their logarithm to enable unconstrained optimization techniques while simultaneously ensuring positivity. The calculation of one partial differentiation usually costs  $\mathcal{O}(n^3)$ , but the adaption of hyperparameters is essential for accurate GP modeling. Typically, a good CG solution is obtained within only a few gradient steps, if the initial hyperparameters are chosen appropriately. For example, the length-scale parameters  $\boldsymbol{\lambda}$  can be initialized according to Scott's rule of thumb, see Scott (1992). In addition, it is helpful to introduce only as many hyperparameters in the GPR model as necessary. Furthermore, an interesting connection between the local optima of the evidence maximization problem (2.31) and the entropy of the marginal likelihood (2.6) is shown in the following lemma, where the proof of the lemma is provided in Section A.3.5 of the appendix.

**Lemma 2.1.** *For a GPR model as introduced in Section 2.1.1 with a differentiable logarithmic marginal likelihood (2.32) with respect to the hyperparameters of the covariance function, defined analogously to Section 2.1.4, it holds true that*

$$\varphi(\boldsymbol{\theta}_{\text{Opt}}) = -\text{H}[\mathbf{y} | \mathbf{X}, \boldsymbol{\theta}_{\text{Opt}}] , \quad (2.33)$$

where  $\boldsymbol{\theta}_{\text{Opt}} = \arg \max_{\boldsymbol{\theta}}(\varphi(\boldsymbol{\theta}))$  is a set of hyperparameters to a possible local maxima of  $\varphi(\boldsymbol{\theta})$ .

Unfortunately, the latter lemma provides no information about the quality of the selected hyperparameters with respect to the global optimum of the logarithmic marginal likelihood. Nevertheless, the proposed relationship (2.33) is useful for the active learning part of this thesis as detailed in Chapter 4. Moreover, the entropy of the marginal Gaussian distribution plays a crucial role when defining our active data-sampling strategy.

## Classification

In analogy to the regression case, the hyperparameters of the classification model, which are summarized in  $\boldsymbol{\theta}$  as well, will be also adapted with marginal likelihood maximization strategies. Note that the GP classifier is noise-free, i.e. that the hyperparameters summarized in  $\boldsymbol{\theta}$  consist only of parameters from the used mean and covariance function. Since the exact marginal likelihood  $p(\mathbf{c} | \mathbf{X})$  is analogously to the true posterior distribution not analytically tractable, cf. Equation (2.13), the Laplace approximation to calculate an approximate Gaussian evidence  $q(\mathbf{c} | \mathbf{X})$  is used, see Rasmussen and Williams (2006). Therefore, a second-order Taylor expansion of  $\Psi(\mathbf{g})$  as given in (2.16) is considered locally around the mode  $\boldsymbol{\mu}$ , which results in  $\Psi(\mathbf{g}) \approx \Psi(\boldsymbol{\mu}) - \frac{1}{2}(\mathbf{g} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{g} - \boldsymbol{\mu})$ . Note that in the Taylor expansion of

$\Psi(\mathbf{g})$  the term depending on the gradient of  $\Psi(\mathbf{g})$  with respect to  $\boldsymbol{\mu}$  cancels out due to the necessary condition for a maximum in the self-consistent Equation (2.17). Hence, it follows

$$\begin{aligned} p(\mathbf{c} | \mathbf{X}) &= \int_{\mathbb{R}^n} p(\mathbf{c} | \mathbf{g}, \mathbf{X}) p(\mathbf{g} | \mathbf{X}) \partial \mathbf{g} = \int_{\mathbb{R}^n} \exp(\Psi(\mathbf{g})) \partial \mathbf{g} \\ &\approx q(\mathbf{c} | \mathbf{X}) = \exp(\Psi(\boldsymbol{\mu})) \int_{\mathbb{R}^n} \exp\left(-\frac{1}{2}(\mathbf{g} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{g} - \boldsymbol{\mu})\right) \partial \mathbf{g}, \end{aligned} \quad (2.34)$$

and finally the approximated logarithmic marginal likelihood

$$\psi(\boldsymbol{\theta}) = \log(q(\mathbf{c} | \mathbf{X}, \boldsymbol{\theta})) = \sum_{i=1}^n \log(\Phi(c_i \mu_i)) - \sum_{i=1}^n \log(l_{ii}) - \frac{1}{2}(\boldsymbol{\mu} - \mathbf{m})^T \boldsymbol{\alpha} \quad (2.35)$$

for the GPC model, where the detailed derivation is shown in the Appendix A.3.3, cf. (A.33), and where  $\mathbf{L}$  is the Cholesky factor and  $\boldsymbol{\alpha}$  the prediction vector as in Equation (2.20). To maximize the approximated logarithmic marginal likelihood with gradient-based optimization methods, the partial derivatives according to the hyperparameters  $\boldsymbol{\theta}$  need to be calculated. Therefore, the technical details for the gradient calculations are postponed to Section A.3.4 of the appendix. In the classification setting, the cost to adapt hyperparameters are slightly higher than in the regression case, since in each iteration a few Newton steps have to be computed to determine the approximated marginal likelihood. Nonetheless, the hyperparameter learning process is as important as for the regression task.

## 2.2 Error Measures

In this section, different error measures are introduced to assess the quality of the learned data-based models. Since the form of the regression or rather classification output induces a considerably different notion of error measures, the upcoming section is subdivided according to the considered supervised learning tasks. A more detailed discussion of error measures for regression and classification problems is presented in Olsen (2004) and Bishop (2006), respectively. Furthermore, each error measure is directly related to a loss function which specifies the penalty when the estimated model output differs from the true value.

### 2.2.1 Regression

When considering continuous model predictions as in the regression case, various error measures exist for analyzing and assessing the model quality. For notational simplicity, let  $\mathbf{y} \in \mathbb{R}^n$  be the vector of given target values and  $\mathbf{f} \in \mathbb{R}^n$  the vector of predicted system responses, respectively. In statistics, a widespread error measure is the mean square error (MSE) which is defined as the mean of the squared residuals  $(y_i - f_i)^2$ . Thus, the MSE is defined by

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - f_i)^2 = \frac{1}{n} \|\mathbf{y} - \mathbf{f}\|^2. \quad (2.36)$$

The MSE is induced by the quadratic loss and includes the bias as well as the variance of the predictions, see Hastie et al. (2009) for more details on the decomposition into bias and variance. If the predictor is unbiased, the MSE equals to the variance and has therefore the squared unit related to the underlying measurement unit of the output  $y$ . Hence, the root mean square error (RMSE) is given by

$$\text{RMSE} = \sqrt{\text{MSE}} = \frac{\|\mathbf{y} - \mathbf{f}\|}{\sqrt{n}} \quad (2.37)$$

and is related to the standard deviation. Note that the RMSE is a biased estimator for the true standard deviation. In practice, the normalized root mean square error

$$\text{NRMSE} = \frac{\text{RMSE}}{\max(\mathbf{y}) - \min(\mathbf{y})} \quad (2.38)$$

is often used since it describes the proportion of the estimated standard derivation (RMSE) on the target interval. Due to the dividing by the range of the targets it may be beneficial to present the RMSE in percent. Regarding the fact that no consistent treatment of normalization is given in the statistic literature, a normalization with the mean  $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$  would also be possible. To enable a fair comparison between different test data sets according to the same trained model with respect to the NRMSE, the range of the targets depends only on the training data even though the RMSE results from the test data. Another normalized error measure is the normalized mean square error given by

$$\text{NMSE} = \frac{\text{MSE}}{s^2} = \frac{(n-1) \|\mathbf{y} - \mathbf{f}\|^2}{n \|\mathbf{y} - \bar{y} \mathbf{1}\|^2}. \quad (2.39)$$

Dividing by the variance of the targets  $s^2 = \frac{1}{n-1} \sum_{i=1}^n (y_i - \bar{y})^2$  allows for an indication in percent. Moreover, the NMSE value is related to the coefficient of determination which can be exactly pointed out for the case of multiple linear regression, cf. Press (2005). In contrast to the error measures based on the MSE, the maximum absolute error

$$\text{MAE} = \max_{i=1, \dots, n} (|y_i - f_i|) \quad (2.40)$$

is equal to the absolute value of the largest residual and is a measure for almost half the spread of all residuals. The normalization with the range of the targets yields the normalized maximum absolute error

$$\text{NMAE} = \frac{\text{MAE}}{\max(\mathbf{y}) - \min(\mathbf{y})}, \quad (2.41)$$

which can also be given in percent. Note that all normalized error measures, i.e. the NRMSE, NMSE, and NMAE, are independent from the scaling of the targets. When analyzing trained models with the presented error measures, where a lower value is always preferable, a more detailed comparison should be supplemented by a look at the distribution of the residuals. In this way, systematic model errors can be detected and corrected.

## 2.2.2 Classification

To assess the quality of a classifier, the predictive assignment of test points to the right class is essential. In the following, only the case of binary classification is considered, where  $\mathbf{c} \in \{-1, +1\}^n$  is the vector of

true class labels and  $\mathbf{f} \in \{-1, +1\}^n$  is the associated vector of the predicted test outcomes. The result of a binary classification task can be summarized in a so-called contingency table, also known as confusion matrix, which contains the absolute values of quantitative predictions to analyze the classification model, see Table 2.1. Due to the binary classification setting, only four cases need to be distinguished. For the

		true class label	
		positive	negative
test outcome	positive	TP (true positive)	FP (false positive)
	negative	FN (false negative)	TN (true negative)

Table 2.1: Confusion matrix for the binary classification task. Generally, true refers to correctly identified instances and false to wrongly classified instances by the predictor.

upcoming definitions the index function  $\mathbb{I}(x)$  is introduced. Note that  $\mathbb{I}(x)$  is equal to 1 if  $x$  is true and zero otherwise, cf. to Section A.2. Now, the four individual quantitative prediction values are defined by

$$\begin{aligned} \text{TP} &= \sum_{i=1}^n \mathbb{I}(f_i = +1) \mathbb{I}(c_i = +1) , & \text{FN} &= \sum_{i=1}^n \mathbb{I}(f_i = -1) \mathbb{I}(c_i = +1) , \\ \text{FP} &= \sum_{i=1}^n \mathbb{I}(f_i = +1) \mathbb{I}(c_i = -1) , & \text{and} & \quad \text{TN} = \sum_{i=1}^n \mathbb{I}(f_i = -1) \mathbb{I}(c_i = -1) . \end{aligned}$$

Furthermore, these shortcuts are used to propose the most relevant error measures for classification problems. Moreover, the interpretation of these error measures is considerably dependent on the underlying population, more precisely on the given distribution of the true class labels  $\mathbf{c}$ . Let us start with the recall or sensitivity

$$\text{SEN} = \Pr[\text{positive test outcome} \mid \text{true positive class}] = \frac{\text{TP}}{\text{TP} + \text{FN}} , \quad (2.42)$$

which is related to the probability of the predictive model to identify the positive instances that are truly positive. In contrast, the specificity

$$\text{SPC} = \Pr[\text{negative test outcome} \mid \text{true negative class}] = \frac{\text{TN}}{\text{FP} + \text{TN}} , \quad (2.43)$$

also called true negative rate, coincide with the probability to correctly classify a test example as negative. Finally, the classification error

$$\text{CE} = \Pr[\text{misclassified}] = \frac{\text{FP} + \text{FN}}{n} \quad (2.44)$$

is of overall interest, since it describes the proportion of all misclassified predictions, that means all false positive and false negative test outcomes. Furthermore, the CE can be seen as one minus the accuracy of the binary classifier. Note that the CE value is no meaningful representation of the quality of a classification model if the given data is unbalanced, i.e. the number of instances in each class varies significantly. Hence, it is necessary to consider problem-specific or combined error measures when comparing classifiers. For example, the so-called receiver operating characteristic (ROC) is widely used to visualize the behavior between the SEN and  $1 - \text{SPC}$  for changing parameters of the classification model, cf. Hastie et al. (2009).



## 2.3 NARX Approach for Transient Modeling

For accurate transient modeling of non-linear dynamic systems it is necessary to consider the time dependence between the input and output values of the system. For employing data-based methods, there are two possible ways to take the time dependence into account. On the one hand, some approaches deal with a methodology to directly learn the time varying dynamics from the discretized input and output signals by means of the modeling algorithm. An example here are recurrent neural network approaches, where some details can be found in Jaeger (2003). In contrast, the modeling techniques considered in this thesis are based on the extension of the underlying time dependent training data. More precisely, the system output at time  $t$  is modeled as a noisy function of the current and time-delayed input and output values. Formally, the discretized system output  $y(\mathbf{x}_t) = y_t$  is defined by

$$y_t = f(y_{t-1}, \dots, y_{t-p}, \mathbf{x}_t, \mathbf{x}_{t-1}, \dots, \mathbf{x}_{t-q}) + \varepsilon_t, \quad (2.45)$$

where  $\mathbf{x}_t$  are the  $d$ -dimensional input points and  $\varepsilon_t$  is the disturbance for all time steps  $t$ , cf. Leontaritis and Billings (1985). The design parameters  $p$  and  $q$  describe the order of time dependent recurrences of the system's output and input, respectively.

Equation (2.45) is called a NARX( $p, q$ ) (non-linear autoregressive exogenous) model, if  $f(y_{t-1}, \dots, y_{t-p}, \mathbf{x}_t, \mathbf{x}_{t-1}, \dots, \mathbf{x}_{t-q}) = f_t$  is a continuous non-linear function. A schematic illustration of the NARX( $p, q$ ) model is presented in Figure 2.1. Consequently, the input domain of the function  $f$  has a NARX( $p, q$ ) structure and this function is non-recurrent, since the output  $f_t$  does not depend on itself. In this sense, autoregressive is related to the recurrence of time-delayed discretized output values, where  $p = 0$  means that  $f$  is only a function of the exogenous input. Hence, the NARX( $0, q$ ) model reduces to a NX( $q$ ) (non-linear exogenous) approach. Finally, the input dimension of the non-linear function  $f$  increases linearly in  $p$  and  $q$ .

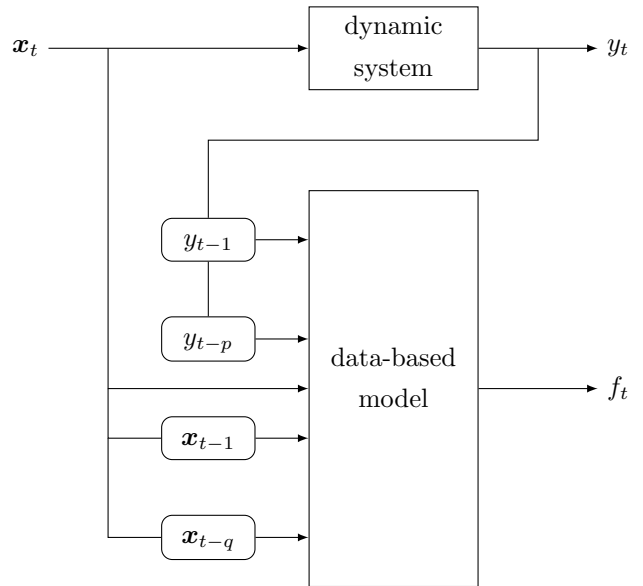


Figure 2.1: Transient NARX( $p, q$ ) model.

### 2.3.1 Feature Selection

Due to the generally unknown structure of the NARX modeling approach (2.45), it is hard to explicitly determine an appropriate parametrization of the typically non-linear function  $f$ . Nevertheless, suchlike parametric models are widely used. An extensive review for such model selection approaches is given by Hong et al. (2008). For an efficient system identification process it is important to simultaneously determine significant features, that means representative input dimensions or special historic system

output. In particular, the whole input dimension of a full NARX( $p, q$ ) model is  $p + (q + 1)d$ , see Figure 2.1. According to Bayesian model selection, cf. Hastie et al. (2009), a partial NARX( $p, q$ ) structure would be desirable to keep the modeling task small and simple. That is, only the most informative features should be selected to explain the functional relationship so that the modeling problem is not expanded too much, and thus overfitting can be avoided. Hence, feature selection approaches enable an intelligent search through the combinatorial high dimensional search space to provide a meaningful definition of the transient model (2.45), cf. Guyon and Elisseeff (2003). A comparison between various feature selection procedures with respect to dynamic modeling tasks can be found in Markert et al. (2011). Another advantage of suchlike optimization techniques is the speed-up of regression algorithms that scales badly with increasing input dimensions. In this work, the Gaussian process setting is able to perform a feature selection with the help of the ARD framework of the covariance function, see Section 2.1.4. Nevertheless, feature selection procedures can a priori reduce the learning effort if the input structure of the NARX model is not well identified. Throughout this thesis, the non-parametric GP modeling approach is considered, i.e. a class of distributions instead of an explicit representation for the function  $f$  is employed. In this way, only the structure of the input space given by the NARX approach is exploited for the GP modeling techniques, where for the sake of simplicity all features are summarized in  $\mathbf{x}$  in the following chapters.

### 2.3.2 Prediction

Now, if predictions for new test points are considered, the true system output up to  $y_t$  is not available as in the above described NARX approach for the case of model training. Therefore, the estimated target values  $f_t$  are repeatedly used to induce the needed autoregressive feedback for the underlying NARX structure. This scheme is called multiple-step ahead prediction and is visualized in Figure 2.2. Depending on the partial NARX( $p, q$ ) approach and the resulting accuracy of the learned data-based model, the prediction error will increase compared to the trained model. Hence, the benefits of an extensive historic output feedback should be analyzed. Specifically, a large design parameter  $p$  yields higher training precision at the expense of deteriorating generalization capabilities due to the distinct multiple-step ahead

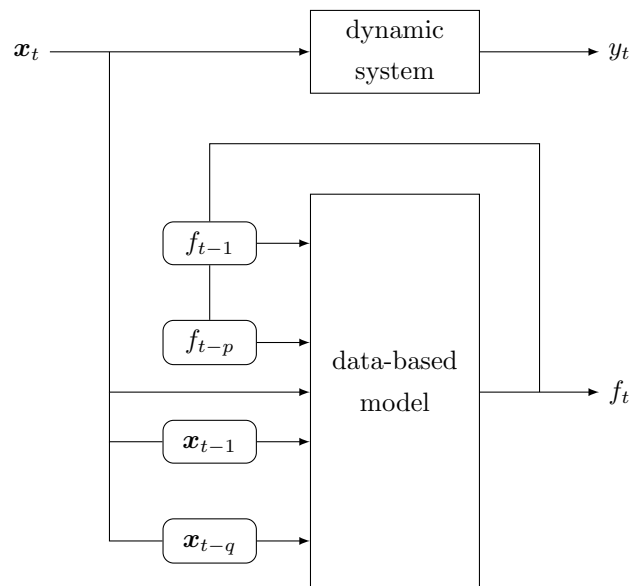


Figure 2.2: Multiple-step ahead prediction scheme for the transient NARX( $p, q$ ) model.

predictions. A GP technique which considers such an uncertainty in the input features when forecasting dynamic time series is proposed by Girard et al. (2003). They describe an approximation of the predictive distribution depending on former estimated system responses, i.e. for a real NARX structure. According

---

to our empirical experience, a ratio of one historic output feedback to three important, maybe also historic input features should not be exceeded to guarantee stable predictions. Thus, the exactly measured input features are building the core of the modeling algorithm to avoid a too strong deviation of the subsequently calculated predictions. Based on the former empirical rule of thumb and the properties of the modeling tasks considered in this thesis, like low noise of the measured system output, the slightly better but computationally much more inefficient prediction method induced by the work of Girard et al. (2003) is neglected.



# 3 Sparse Gaussian Process Approximations

Sparse GP approximations provide an efficient way for model generation on large data sets. The key idea of the approaches considered in this chapter is to select a representative subset of the available training data to introduce the sparse model approximation. A variety of selection criteria has been proposed, but they either lack accuracy or suffer from high computational costs. The main insight of this chapter is a new and straightforward criterion for successive insertion and deletion of training points in sparse GP models. Our main motivation is computational efficiency, namely, the proposed novel strategies for sparsification are as fast as the purely randomized schemes, and thus appropriate for applications in online learning. Furthermore, an efficient approach for model selection is presented. In fact, the hyperparameters as well as the representative subset of training points are simultaneously determined with an expectation maximization scheme. Without loss of generality, only sparse GP approximation techniques for regression tasks are considered in this chapter. Nevertheless, it is also possible to employ the proposed methods in a classification setting with adapted inference techniques due to the occurring non-Gaussian likelihoods as explained in Section 2.1.2. Extensive evaluations with respect to real-world benchmark data sets demonstrate that our obtained sparse regression models are competitive with the computationally intensive state-of-the-art methods in terms of generalization and accuracy. Additionally, our approach is applied to learn inverse dynamics models for compliant robot control and to the simulation of vehicle power demand using very large data sets, that means with nearly half a million training points per application.

The outline of this chapter, which represents one main part of my contribution, is as follows. After a broad review of related work in the next section, the considered sparse GP approximations are introduced in the subsequent sections. Therein, the theoretical part is based on the already published work by Schreiter et al. (2015a). In addition to several evaluations on benchmark data sets, the experimental Section 3.4 of this chapter shows results for the presented GP approximations regarding a robot control task (Schreiter et al., 2015b) and an automotive application (Schreiter et al., 2013). A short summary of the whole chapter's content is already published in Schreiter et al. (2016). Finally, a discussion of the proposed methods and results completes this chapter of the thesis.

## 3.1 Related Work

As mentioned previously in this chapter, the applicability of standard GPR to large scale problems with a high number of training points  $n$  is limited due to its unfavourable scaling in training time and memory

requirements. The dominating factors are usually  $\mathcal{O}(n^3)$  cost for inversion of a dense covariance matrix  $\mathbf{K} \in \mathbb{R}^{n \times n}$  between all available  $n$  training points and the  $\mathcal{O}(n^2)$  space required to store  $\mathbf{K}$  in memory. Since the performance of standard GPR decreases for an increasing number of training data points, an obvious solution is to use only a subset of the available data of size  $m \ll n$  to result in a manageable GP model. This straightforward model reduction is known as subset of data (SoD), cf. Rasmussen and Williams (2006). The main task of the SoD method, analogously to some of the following sparse GP approximations, is to determine a representative subset of the training data. The simplest strategy is to select the subset randomly. A more intelligent approach for subset selection is employed by Lawrence et al. (2003), where an information-theoretic scheme is used to choose the appropriate input points. Besides the considerable increase of computational efficiency, now requiring only efforts with sub-

ject to  $m$ , no information of the  $n - m$  remaining points is included in the resulting GP model. Moreover, a purely randomized subset selection can lead to excessive underfitting. To eliminate this disadvantage and to provide efficient model learning algorithms, many approximations for standard GPR have been proposed. An illustration showing different GP approximation approaches is presented in Figure 3.1. Please note that this illustration is by far not complete. For example, GP models based on network architectures, see e.g. Damianou and Lawrence (2013), are not considered in this overview. Hierarchical GP models as presented by Deisenroth and Ng (2015b) which enable distributed inference techniques for very large data sets are also not included in the figure. Local GPR approaches, as e.g. proposed by Nguyen-Tuong et al. (2009), can be used to increase modeling performance. These methods are based on a partitioning of the input space, where for each region a local GP model is trained. On the other hand, more sophisticated GPR approximation techniques consider either approximations of the dense Gram matrix  $\mathbf{K}$  or focus on sparse likelihood approximations. For example, covariance matrix approximations such as the Nyström method can be used to reduce modeling effort, see Williams and Seeger (2001). This technique leads to a low-rank approximation of the full kernel matrix given by

$$\mathbf{K} \approx \mathbf{K}_{I,N}^T \mathbf{K}_{I,I}^{-1} \mathbf{K}_{I,N} = \mathbf{V}^T \mathbf{V}, \quad (3.1)$$

where  $\mathbf{V} = \mathbf{L}^{-1} \mathbf{K}_{I,N} \in \mathbb{R}^{m \times n}$  and  $\mathbf{L} \in \mathbb{R}^{m \times m}$  is the Cholesky factor of  $\mathbf{K}_{I,I} \in \mathbb{R}^{m \times m}$  according to (A.1). The submatrices  $\mathbf{K}_{I,I}$  and  $\mathbf{K}_{I,N} \in \mathbb{R}^{m \times n}$  of the complete covariance matrix are defined over the index sets  $I \subseteq N = \{1, \dots, n\}$  with  $|I| = m$ . The selection of the index set  $I$  can be carried out randomly or by applying some of the strategies explained later in this chapter. Applying the matrix inversion

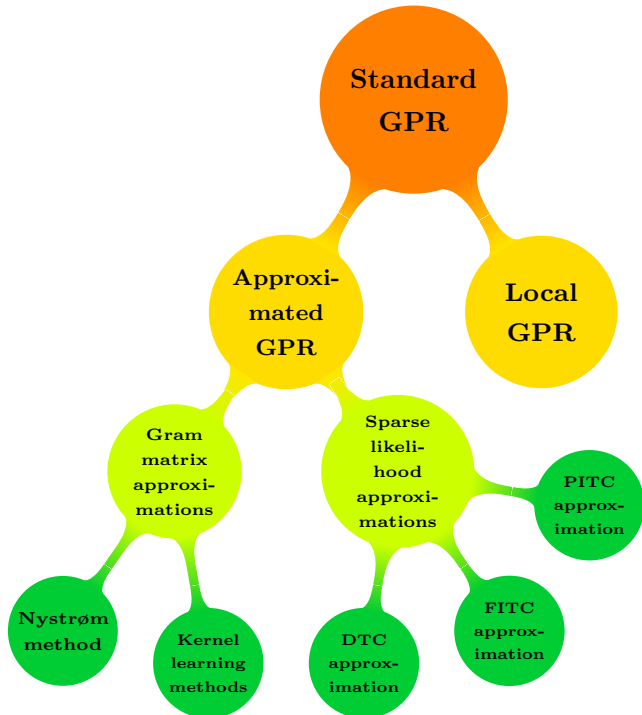


Figure 3.1: Relation between different approximation techniques for GPR. Note that this illustration of methodology relations is by far not complete.

lemma (A.14) to the computationally most expensive part of the predictive distribution (2.7) yields

$$(\sigma^2 \mathbf{I} + \mathbf{K})^{-1} \approx (\sigma^2 \mathbf{I} + \mathbf{V}^T \mathbf{V})^{-1} = \frac{1}{\sigma^2} \mathbf{I} - \frac{1}{\sigma^2} \mathbf{V}^T (\sigma^2 \mathbf{I} + \mathbf{V} \mathbf{V}^T)^{-1} \mathbf{V} , \quad (3.2)$$

where only a matrix of size  $m \times m$  needs to be inverted. Thus, the overall computational cost is reduced to  $\mathcal{O}(nm^2)$  with memory requirements in  $\mathcal{O}(mn)$ . Due to the properties of a covariance function, cf. Section 2.1.4, the Gram matrix is mostly of full rank such that the Nyström approximations lead to unsatisfactory results for subset sizes  $m$  being too small. Another class of methods which gives an approximation of the covariance matrix  $\mathbf{K}$  analogously to the Nyström method are Fourier kernel learning approaches, cf. Băzăvan et al. (2012). Therein, an approximation of the covariance function  $k(\mathbf{x}, \mathbf{z})$  by a finite Fourier series implies the same representation of the kernel matrix  $\mathbf{K}$  as in (3.1), and hence leads to a fast GPR algorithm, cf. Lázaro-Gredilla et al. (2010). As pointed out in the theorem of Bochner, see Stein (1999), this approach is only feasible for stationary covariance functions. Furthermore, a relationship between the quality of the kernel approximation, the input dimension  $d$ , and the length of the Fourier series is presented in Rahimi and Recht (2008). An additional method is the subset of regressors (SoR) approximation which is based on the representation of the prediction vector  $\boldsymbol{\alpha} \in \mathbb{R}^n$  by means of a centered Gaussian prior distribution

$$P(\boldsymbol{\alpha} | \mathbf{X}) = N(\boldsymbol{\alpha} | \mathbf{0}, \mathbf{K}^{-1}) . \quad (3.3)$$

Here, the goal is to generate a sparse prediction vector such that  $\boldsymbol{\alpha}_R = \mathbf{0}$  for the remaining index set  $R = N \setminus I$ . Thus, the prediction for new test points  $\mathbf{x}_* \in \mathbb{R}^d$  reduces to  $f(\mathbf{x}_*) = \mathbf{k}_{I,*}^T \boldsymbol{\alpha}_I$ . This scheme results in the approximation (3.1) of the covariance matrix in the same way as the Nyström method which can lead to poor predictions. The SoR approximation is related to the relevance vector machine (RVM), see Tipping (2001), which is inspired by the SVM framework as described in Schölkopf et al. (1998). However, the RVM uses a prior that factorizes over the weights  $\alpha_i$  since they are treated as independent. This improves the computational performance but also induces a deteriorated posterior variance. An improvement for the predictive distribution of the RVM through augmentation is given by Rasmussen and Quiñonero-Candela (2005). In contrast to these approaches, various sparse likelihood approximations have emerged recently, whose relations have been formalized in the unifying framework of Quiñonero-Candela and Rasmussen (2005). Their systematization together with the presented algorithmic designations are exploited throughout this thesis. The first considered sparse approximation deals with a partitioning of the given training data into disjoint sets which results in a block diagonal covariance matrix. This method arose as Bayesian committee machine (BCM) in Tresp (2000a), but the notation as partially independent training conditional (PITC) approximation is preferred. Compared to the former approximations, the PITC method has slightly higher computational effort depending on the specified partitioning of the input space. A generalized approach of the PITC setting enabling also online learning is described by Tresp (2000b). The fully independent training conditional (FITC) approximation by Snelson and Ghahramani (2006a) uses a flexible subset of virtual training points to generate a sparse GPR model and optimizes the virtual training points along with all other hyperparameters. More details about this approximation are provided in Section 3.3. Furthermore, the deterministic training conditional (DTC) approximation, as established by Csató (2002) and Seeger (2003), selects a representative subset of real training points, the so-called active points, that leads to the sparse likelihood approximation. That is the key sparse GP approximation of this thesis and explained in detail in Section 3.2. A variational formalism for the last two sparse approximation techniques is presented by Titsias (2009). This formalism leads to a

regularized logarithmic marginal likelihood for hyperparameter learning and the additional optimization of virtual training points with respect to the FITC approximation plus a new greedy selection method for the DTC approximation. Moreover, Cao et al. (2013) introduced a general optimization framework based on incomplete Cholesky decompositions for the above mentioned sparse approximations. Here, greedy schemes are employed for the DTC approximation due to the high computational complexity of the optimal subset selection problem which is NP-hard according to Natarajan (1995). A fast information gain criterion for insertion of training points to the active set is proposed by Seeger et al. (2003). Smola and Bartlett (2001) use a computationally costly selection heuristic which approximates the logarithmic marginal posterior probability. The same formalism as in Smola and Bartlett (2001) is used for the criterion by Keerthi and Chu (2006), while they improve computational performance by using a simpler approximation of the posterior probability. Quiñonero-Candela’s (2004) selection is based on the increase in the logarithmic marginal likelihood of the sparse GP by the insertion of a training point in the active subset. Csató and Opper (2001) measure the projection-induced error in the reproducing kernel Hilbert space (RKHS) and select the point which maximally extends the spanned subspace of the RKHS. Based on this idea, they also introduce a heuristic for deletion of training points from the active data set. They show that removing active points can considerably reduce the prediction times for test points with only slightly decreasing generalization accuracy. More technical details and relationships between the discussed greedy schemes are provided in Section 3.2.2.

All of the insertion and deletion methods mentioned above either lack computational speed, have high memory requirements, or lack of modeling accuracy. Moreover, if the regression model generation is based on a purely randomized selection or on a method with a small randomly selected subset of remaining training points for criteria evaluation, e.g. as done by Quiñonero-Candela (2004), Keerthi and Chu (2006) or Titsias (2009), the performance in complex and challenging regression tasks deteriorates. Our novel sparsification method, proposed in Section 3.2.3, is closely related to the inclusion heuristic by Smola and Bartlett (2001). However, by employing some reasonable assumptions, the computational cost is significantly reduced to the level of randomized selection without suffering a high loss in model accuracy. Compared to the deletion criterion by Csató and Opper (2001), our approach offers nearly the same prediction performance despite a lower computing time and less memory requirements.

## 3.2 Deterministic Training Conditional Approximation

Csató (2002) and Seeger (2003) laid the foundation for that sparse GPR model under the DTC approximation which is presented below. To facilitate the comparability of the different greedy criteria the notation by Seeger et al. (2003) is adopted. As previously, let  $I$  be the index set with size  $m$  of the so-called active training points  $\mathbf{X}_I \in \mathbb{R}^{m \times d}$  and  $R$  the index set containing the indices of the remaining data points such that  $I \cup R = N = \{1, \dots, n\}$ . Analogous to the usual GPR approach, the goal of the considered sparse GPR model is the estimation of the functional relationship according to (2.2) for given training data  $\mathcal{D} = (\mathbf{y}, \mathbf{X})$ . Thus, the way of generating predictions for not seen test points based on this



finite data set  $\mathcal{D}$  is inductive. Therefore, the centered Gaussian prior distribution

$$\mathbb{P}(\mathbf{f}_I | \mathbf{X}_I) = \mathcal{N}(\mathbf{f}_I | \mathbf{0}, \mathbf{K}_{I,I}) \quad \text{with} \quad \mathbf{K}_{I,I} = (k(\mathbf{x}_i, \mathbf{x}_j))_{i,j \in I} \in \mathbb{R}^{m \times m} \quad (3.4)$$

over the latent function values  $\mathbf{f}_I \in \mathbb{R}^m$  corresponding to the active subset  $\mathbf{X}_I$  is employed. Therein,  $\mathbf{K}_{I,I}$  is the covariance matrix over the active training points determined through the specified covariance function  $k(\mathbf{x}_i, \mathbf{x}_j) = k_{ij}$ . The sparseness of this method is introduced via a likelihood approximation that is optimized with respect to the Kullback-Leibler (KL) divergence, cf. Equation (A.25), and induced by the active training points, so that

$$q_I(\mathbf{y} | \mathbf{f}_I, \mathbf{X}) = \mathcal{N}(\mathbf{y} | \mathbf{P}_I^T \mathbf{f}_I, \sigma^2 \mathbf{I}) \quad (3.5)$$

is obtained. Here,  $\mathbf{K}_{I,N} \in \mathbb{R}^{m \times n}$  comprises the covariance function values between all training points, indexed by  $N$ , and the active subset of training points, next to the projection matrix  $\mathbf{P}_I = \mathbf{K}_{I,I}^{-1} \mathbf{K}_{I,N} \in \mathbb{R}^{m \times n}$  which maps  $\mathbf{f}_I$  to the prior conditional mean  $\mathbb{E}_{\mathbb{P}}[\mathbf{f} | \mathbf{f}_I, \mathbf{X}] = \mathbf{K}_{I,N}^T \mathbf{K}_{I,I}^{-1} \mathbf{f}_I \in \mathbb{R}^n$ , cf. Equation (A.22). Furthermore, Bayesian inference leads to the approximated posterior density

$$\begin{aligned} q_I(\mathbf{f}_I | \mathbf{y}, \mathbf{X}) &= \frac{q_I(\mathbf{y} | \mathbf{f}_I, \mathbf{X}) \mathbb{P}(\mathbf{f}_I | \mathbf{X}_I)}{q_I(\mathbf{y} | \mathbf{X})} \\ &\propto q_I(\mathbf{y} | \mathbf{f}_I, \mathbf{X}) \mathbb{P}(\mathbf{f}_I | \mathbf{X}_I) \\ &= \mathcal{N}(\mathbf{y} | \mathbf{P}_I^T \mathbf{f}_I, \sigma^2 \mathbf{I}) \mathcal{N}(\mathbf{f}_I | \mathbf{0}, \mathbf{K}_{I,I}) \\ &\propto \mathcal{N}(\mathbf{f}_I | \mathbf{L} \mathbf{M}^{-1} \mathbf{V} \mathbf{y}, \sigma^2 \mathbf{L} \mathbf{M}^{-1} \mathbf{L}^T), \end{aligned} \quad (3.6)$$

which is proportional to the product of the prior (3.4) and the approximated likelihood (3.5). In the last step of this derivation Equation (A.19) was applied, where the lower diagonal matrix  $\mathbf{L} \in \mathbb{R}^{m \times m}$  is the Cholesky factor of  $\mathbf{K}_{I,I}$  according to (A.1),  $\mathbf{V} = \mathbf{L}^{-1} \mathbf{K}_{I,N} \in \mathbb{R}^{m \times n}$ , and  $\mathbf{M} = \sigma^2 \mathbf{I} + \mathbf{V} \mathbf{V}^T \in \mathbb{R}^{m \times m}$  for fixed  $I$  of size  $m$ . The approximated marginal likelihood directly follows from the integration over the same product of Equation (3.4) and (3.5) with respect to the active function values  $\mathbf{f}_I$  and results in

$$\begin{aligned} q_I(\mathbf{y} | \mathbf{X}) &= \int_{\mathbb{R}^m} q_I(\mathbf{y} | \mathbf{f}_I, \mathbf{X}) \mathbb{P}(\mathbf{f}_I | \mathbf{X}_I) \partial \mathbf{f}_I \\ &= \int_{\mathbb{R}^m} \mathcal{N}(\mathbf{y} | \mathbf{P}_I^T \mathbf{f}_I, \sigma^2 \mathbf{I}) \mathcal{N}(\mathbf{f}_I | \mathbf{0}, \mathbf{K}_{I,I}) \partial \mathbf{f}_I \\ &= \mathcal{N}(\mathbf{y} | \mathbf{0}, \sigma^2 \mathbf{I} + \mathbf{V}^T \mathbf{V}). \end{aligned} \quad (3.7)$$

Finally, the predictive Gaussian density for the function value of a test point  $\mathbf{x}_* \in \mathbb{R}^d$  is given by

$$\begin{aligned} q_I(f_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}) &= \int_{\mathbb{R}^m} \mathbb{P}(f_* | \mathbf{x}_*, \mathbf{f}_I, \mathbf{X}_I) q_I(\mathbf{f}_I | \mathbf{y}, \mathbf{X}) \partial \mathbf{f}_I \\ &= \int_{\mathbb{R}^m} \mathcal{N}(f_* | \mathbf{k}_{I,*}^T \mathbf{K}_{I,I}^{-1} \mathbf{f}_I, \mathbf{k}_{**} - \|\mathbf{L}^{-1} \mathbf{k}_{I,*}\|^2) \mathcal{N}(\mathbf{f}_I | \mathbf{L} \mathbf{M}^{-1} \mathbf{V} \mathbf{y}, \sigma^2 \mathbf{L} \mathbf{M}^{-1} \mathbf{L}^T) \partial \mathbf{f}_I \\ &= \mathcal{N}(f_* | \mathbf{k}_{I,*}^T \mathbf{L}^{-T} \mathbf{L}_M^{-T} \boldsymbol{\beta}_I, \mathbf{k}_{**} - \|\mathbf{L}^{-1} \mathbf{k}_{I,*}\|^2 + \sigma^2 \|\mathbf{L}_M^{-1} \mathbf{L}^{-1} \mathbf{k}_{I,*}\|^2). \end{aligned} \quad (3.8)$$

In the last step Equation (A.21) is used and the Cholesky decomposition of  $\mathbf{M} = \mathbf{L}_M \mathbf{L}_M^T$  related to (A.1),  $\boldsymbol{\beta}_I = \mathbf{L}_M^{-1} \mathbf{V} \mathbf{y} \in \mathbb{R}^m$  and the covariance vector  $\mathbf{k}_{I,*} \in \mathbb{R}^m$  between the test input and the active points are

defined. Moreover, the last step in the derivation of  $q_I(f_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X})$  uses the density  $p(f_* | \mathbf{x}_*, \mathbf{f}_I, \mathbf{X}_I)$  which follows from the joint multivariate normal distribution

$$\begin{pmatrix} \mathbf{f}_I | \mathbf{X}_I \\ f_* | \mathbf{x}_* \end{pmatrix} \sim \mathcal{N} \left( \begin{pmatrix} \mathbf{f}_I \\ f_* \end{pmatrix} \middle| \begin{pmatrix} \mathbf{0} \\ 0 \end{pmatrix}, \begin{pmatrix} \mathbf{K}_{I,I} & \mathbf{k}_{I,*} \\ \mathbf{k}_{I,*}^T & k_{**} \end{pmatrix} \right)$$

by conditioning on the Gaussian prior (3.4) with Equation (A.22).

Due to the matrix-matrix multiplications, the training complexity of this sparse DTC model is  $\mathcal{O}(nm^2)$ . What is more, if only the predicted mean values are of interest, the prediction vector  $\boldsymbol{\alpha}_I = \mathbf{L}^{-T} \mathbf{L}_M^{-T} \boldsymbol{\beta}_I$  can be precomputed to perform computations of mean values with only  $\mathcal{O}(dm)$  cost. Note that this cost depends on the calculation of the vector  $\mathbf{k}_{I,*}$ , and thus on the specified covariance function which is typically proportional to the input dimension  $d$ . The predictive variance is feasible in  $\mathcal{O}(m^2)$  if  $d$  is smaller than the number of active inputs  $m$ . Finally, the approximated posterior distribution  $Q_I(\mathbf{f} | \mathbf{y}, \mathbf{X})$  for all training points induced by the active subset indicated by  $I$  results in

$$Q_I(\mathbf{f} | \mathbf{y}, \mathbf{X}) = \mathcal{N}(\mathbf{f} | \mathbf{W}^T \boldsymbol{\beta}_I, \mathbf{K} - \mathbf{V}^T \mathbf{V} + \sigma^2 \mathbf{W}^T \mathbf{W}) \quad (3.9)$$

following from (3.8) with the estimated mean vector  $\boldsymbol{\mu}_I = \mathbb{E}_{q_I}[\mathbf{f} | \mathbf{y}, \mathbf{X}] = \mathbf{W}^T \boldsymbol{\beta}_I \in \mathbb{R}^n$  and the matrix  $\mathbf{W} = \mathbf{L}_M^{-1} \mathbf{V} \in \mathbb{R}^{n \times n}$ . A strong motivation for this kind of likelihood approximation is given by Csató (2002), showing that the minimum of  $\text{KL}[Q_I(\mathbf{f} | \mathbf{y}, \mathbf{X}) \| P(\mathbf{f} | \mathbf{y}, \mathbf{X})]$  with respect to the distribution  $Q_I(\mathbf{f} | \mathbf{y}, \mathbf{X})$  is reached for  $q_I(\mathbf{f} | \mathbf{y}, \mathbf{X}) \propto r(\mathbf{f}_I) p(\mathbf{f} | \mathbf{X})$ . Obviously, the choice of  $r(\mathbf{f}_I) = q_I(\mathbf{y} | \mathbf{f}_I, \mathbf{X})$  as presented in Equation (3.5) yields the DTC approximation. Hence, this type of approximation is considered as information-optimal.

### 3.2.1 Expectation Maximization for Model Selection

The just presented sparse GPR model does not only depend on the active points  $\mathbf{X}_I$ , but also on the hyperparameters of the specified covariance function and the variance  $\sigma^2$  of the Gaussian noise model (2.2). As previously defined in Section 2.1, the vector  $\boldsymbol{\theta}$  denotes the collection of all hyperparameters including  $\sigma^2$ . For the sake of notational simplicity, the dependency of the above formulas on  $\boldsymbol{\theta}$  was neglected. Analogous to the standard GPR approach in Section 2.1.5, the adaptation of the hyperparameters can be realized by gradient-based optimization algorithms that maximize the approximated logarithmic marginal likelihood

$$\varphi_I(\boldsymbol{\theta}) = \log(q_I(\mathbf{y} | \mathbf{X}, \boldsymbol{\theta})) = (m - n) \log(\sigma) - \frac{n}{2} \log(2\pi) - \sum_{i=1}^m \log(l_{M,ii}) - \frac{\mathbf{y}^T \mathbf{y} - \boldsymbol{\beta}_I^T \boldsymbol{\beta}_I}{2\sigma^2} \quad (3.10)$$

obtained from (3.7) as shown by (A.34). Here, the values  $l_{M,ii}$  are the diagonal entries of the lower Cholesky factor  $\mathbf{L}_M$ . Titsias (2009) introduced a regularized version of the above approximated logarithmic marginal likelihood which is defined by

$$\text{VAR } \varphi_I(\boldsymbol{\theta}) = \varphi_I(\boldsymbol{\theta}) - \frac{1}{2\sigma^2} \text{trace}(\mathbf{K} - \mathbf{V}^T \mathbf{V}) \quad (3.11)$$

following from his variational (VAR) approach. Therein, the additional regularization term allows us to correct the Nyström approximation of the full covariance matrix  $\mathbf{K}$  in the hyperparameter learning

process and gives a lower bound to the approximated logarithmic marginal likelihood of the DTC approximation in Equation (3.10), cf. Titsias (2009). The gradients of both approximated logarithmic marginal likelihoods according to the hyperparameters  $\boldsymbol{\theta}$  are provided in Section A.3.4 of the appendix. One problem encountered when maximizing  $\varphi_I(\boldsymbol{\theta})$  or  $\text{VAR} \varphi_I(\boldsymbol{\theta})$  in the variational framework, respectively, is their dependence on the active subset of training points determined by the indices in  $I$ . To solve this problem, we take alternating constrained optimization steps in an expectation maximization (EM) manner employing the theory of Graça et al. (2008). Thus, the expectation step, in short E-step, for estimating the new posterior distribution  $Q_{I_{\text{New}}}(\mathbf{f}_{I_{\text{New}}} | \mathbf{y}, \mathbf{X}, \boldsymbol{\theta})$  from Equation (3.6) with fixed hyperparameters  $\boldsymbol{\theta}$  is given by

$$Q_{I_{\text{New}}}(\mathbf{f}_{I_{\text{New}}} | \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}) = \arg \min_{Q_I(\mathbf{f}_I | \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}) \in \mathcal{Q}_I(\mathbf{y}, \mathbf{X}, \boldsymbol{\theta})} \left( \text{KL} [Q_I(\mathbf{f}_I | \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}) \| Q_I(\mathbf{y}, \mathbf{f}_I | \mathbf{X}, \boldsymbol{\theta})] \right). \quad (3.12)$$

Here, the posterior distribution in the expectation step (3.12) with the KL divergence is conditioned on the family of probability distributions  $\mathcal{Q}_I(\mathbf{y}, \mathbf{X}, \boldsymbol{\theta})$ . The latter corresponds to the family of approximated posterior distributions which is induced by an active subset  $\mathbf{X}_I$  of size  $m$ . This condition is handled by means of a fixed final size  $m$  of the active subset during the greedy selection process which is explained in the next section. Furthermore, the maximization step, in short M-step, implies

$$\boldsymbol{\theta}_{\text{New}} = \arg \max_{\boldsymbol{\theta}} \left( \mathbb{E}_{q_I(\mathbf{f}_I | \mathbf{y}, \mathbf{X}, \boldsymbol{\theta})} [\log (q_I(\mathbf{y} | \mathbf{X}, \boldsymbol{\theta}))] \right) \quad (3.13)$$

to determine an updated set of hyperparameters  $\boldsymbol{\theta}_{\text{New}}$ . The M-step in (3.13) is realized with only a few gradient ascent steps on the approximated logarithmic marginal likelihoods from Equation (3.10) or rather (3.11) for a fixed active point set  $\mathbf{X}_I$ . In this case, the repeated alternating computation of the E- and M-steps leads to a generalized EM algorithm, since the approximated logarithmic marginal likelihoods are increased only. Due to the fact that a generalized EM algorithm converges to local maxima, cf. Wu (1983), the choice of the active training points is important in order to obtain a good set of hyperparameters  $\boldsymbol{\theta}$ . For the selection of the active subset, an efficient maximum error criterion is developed to ensure that the hyperparameter learning process stays fast and stable. The derivation and further details about that newly introduced criterion will be presented in Section 3.2.3.

### 3.2.2 State-of-the-art Greedy Insertion and Deletion Criteria

Most of the GP approximation techniques regarding the DTC approach differ in the way, how the active set  $\mathbf{X}_I$  is selected, cf. Quiñonero-Candela and Rasmussen (2005). Usually, the remaining point  $\mathbf{x}_i$  with  $i \in R$  that has maximum gain with respect to an insertion criterion  $\Delta_i$  will be selected. To include a remaining point in the active subset, the Cholesky factors  $\mathbf{L}$ ,  $\mathbf{L}_M$ , the matrix  $\mathbf{V}$  by means of  $\mathbf{K}_{I,N}$ , the vector  $\boldsymbol{\beta}_I$ , and the mean  $\boldsymbol{\mu}_I$  of the posterior distribution given in Equation (3.9) have to be updated, as shown in the Appendix A.3.3 in Equation (A.37), cf. Seeger et al. (2003). Thus, the cost for the sequential insertion of a training point to the active set in the  $m$ -th iteration of the DTC approximation is  $\mathcal{O}(mn)$ , where the computational effort for the criterion calculation must still be added. Analogously, the active point from the current posterior model (3.9) that has a minimal loss with respect to a deletion criterion  $\nabla_i$  will be removed. While for the inclusion strategies Cholesky updates are sufficiently fast and stable, QR downdates based on the factorization  $\mathbf{QR} = \mathbf{LML}^T$  as described in Equation (A.4) are used to

delete active points since they offer higher numerical stability. This advantageous behavior is discussed in Foster et al. (2009). The cost for deleting one active point in the  $m$ -th iteration without any criterion calculation is equal to the insertion cost, i.e.  $\mathcal{O}(mn)$ . Commonly, the final active set size  $m$  is previously fixed or combined with a stopping criterion induced through the current model quality, cf. Seeger et al. (2003). In the following, let  $I' = I \cup \{i\}$  describe the updated active index set.

One of the simplest and fastest point selection and deletion methods is to randomly select one of the training points. However, in case of complex curve fitting tasks, this method can lead to poor results, particularly for large training data sets  $\mathcal{D}$ . Nevertheless, this approach provides the baseline strategy to which all other methods are compared. For transient data sets, i.e. the inputs and the outputs are time-dependent trajectories according to the specified system stimulus, it is possible to choose the active training points sequentially. In doing so, the active set is successively determined by a fixed index distance. This strategy seems useful, if the index distance is appropriately chosen with respect to the characteristic of the considered system's output, since the index distance is related to a sample frequency. For more sophisticated information the reader is referred to the sampling theorem by Shannon (1949). However, for large training data sets an adequate index distance will not be realizable due to the usually bounded active set size  $m$ . This behavior results in the same problems as the randomized selection, namely issues related to underfitting. Despite their unattractive drawbacks, these simple selection strategies are often used in practice, in particular, due to their straightforward and efficient implementation resulting in  $\mathcal{O}(1)$  cost per point selection.

The idea of Csató (2002) is to define a greedy selection heuristic based on the projection-induced error in the reproducing kernel Hilbert space (RKHS) which is induced by the applied covariance function, see Schölkopf and Smola (2002) for more details. This results in Csató's (CS) insertion criterion

$$\text{CS}\Delta_i = k_{ii} - \|\mathbf{L}^{-1}\mathbf{k}_{I,i}\|^2 \quad (3.14)$$

for all remaining points  $\mathbf{x}_i$  with  $i \in R$  and with covariance vector  $\mathbf{k}_{I,i} \in \mathbb{R}^m$ . Due to its relatively low computational cost of  $\mathcal{O}(m)$  per remaining point, this criterion can be evaluated for all remaining points  $\mathbf{X}_R$ , which slightly increases the overall complexity of the DTC approximation to  $\mathcal{O}(nm^3)$ . Moreover, the criterion is equal to the Schur complement of the updated covariance matrix  $\mathbf{K}_{I',I'}$ , cf. Lipschutz and Lipson (2013), which yields a favorable numerical behavior, since

$$|\mathbf{K}_{I',I'}| = \text{CS}\Delta_i |\mathbf{K}_{I,I}| .$$

Note that the remaining point with highest gain according to an insertion criterion is always selected. Furthermore, be aware of the fact that the criterion (3.14) depends only on the kernel function and not on the targets  $\mathbf{y}$ . Thus, the choice of the covariance function and the determination of the associated hyperparameters is very important for the active set selection. Additionally, Csató and Opper (2001) defined a greedy deletion criterion given by

$$\text{CS}\nabla_i = |\text{CS}\Delta_i \alpha_{I',i}| . \quad (3.15)$$

This formulation is based on an already selected active subset of size  $m+1$ , i.e. with an index set  $I'$ . The main difference between the deletion and selection heuristic by Csató lies in the influence of the respective

element of the prediction vector  $\boldsymbol{\alpha}_I$ . Since  $\boldsymbol{\alpha}_I$  is known,  $\mathcal{O}(m^2)$  arithmetic operations per active point are needed for criterion evaluation. Note that the active point of the posterior model in Equation (3.9) with minimal loss in terms of a deletion criterion is always removed.

Currently, one of the best selection methods with respect to modeling accuracy is proposed by Smola and Bartlett (2001). Their greedy scheme selects the remaining point that maximizes the posterior likelihood

$$\begin{aligned} p(\boldsymbol{\alpha} | \mathbf{y}, \mathbf{X}) &\propto p(\mathbf{y} | \boldsymbol{\alpha}, \mathbf{X}) p(\boldsymbol{\alpha} | \mathbf{X}) \\ &= \mathcal{N}(\mathbf{y} | \mathbf{K}\boldsymbol{\alpha}, \sigma^2 \mathbf{I}) \mathcal{N}(\boldsymbol{\alpha} | \mathbf{0}, \mathbf{K}^{-1}) \\ &\propto \mathcal{N}\left(\boldsymbol{\alpha} \mid (\sigma^2 \mathbf{I} + \mathbf{K})^{-1} \mathbf{y}, \sigma^2 (\sigma^2 \mathbf{I} + \mathbf{K})^{-1} \mathbf{K}^{-1}\right) \end{aligned} \quad (3.16)$$

for the admission of the prediction vector  $\boldsymbol{\alpha} \in \mathbb{R}^n$  of the standard GP approach under the given data set  $\mathcal{D}$ , where Equation (A.19) and the definition of the prior density  $p(\boldsymbol{\alpha} | \mathbf{X})$  according to the SoR approximation (3.3) have been used. The model likelihood  $p(\mathbf{y} | \boldsymbol{\alpha}, \mathbf{X})$  is equal to (2.4) with the transformation  $\boldsymbol{\alpha} = \mathbf{K}^{-1} \mathbf{f}$ . The goal of the greedy selection scheme by Smola and Bartlett is to maximize the modified logarithmic likelihood

$$\begin{aligned} \tau(\boldsymbol{\alpha}) &= \sigma^2 \log(p(\boldsymbol{\alpha} | \mathbf{y}, \mathbf{X})) + \frac{\sigma^2 n}{2} \log(2\pi\sigma^2) - \frac{\sigma^2}{2} \log(|\sigma^2 \mathbf{I} + \mathbf{K}| |\mathbf{K}|) + \frac{\sigma^2}{2} \mathbf{y}^T \mathbf{K} (\sigma^2 \mathbf{I} + \mathbf{K})^{-1} \mathbf{y} \\ &= -\frac{1}{2} \boldsymbol{\alpha}^T \mathbf{K} (\sigma^2 \mathbf{I} + \mathbf{K}) \boldsymbol{\alpha} + \boldsymbol{\alpha}^T \mathbf{K} \mathbf{y} \end{aligned} \quad (3.17)$$

following from Equation (3.16) where only the terms depending on  $\boldsymbol{\alpha}$  are taken into account. This leads to the equivalent formulation

$$\tau(\boldsymbol{\alpha}_I) = -\frac{1}{2} \boldsymbol{\alpha}_I^T \mathbf{L} \mathbf{M} \mathbf{L}^T \boldsymbol{\alpha}_I + \boldsymbol{\alpha}_I^T \mathbf{L} \mathbf{V} \mathbf{y} \quad (3.18)$$

for the sparse DTC approximation since  $\boldsymbol{\alpha}_R = \mathbf{0}$ , and thus  $\mathbf{K}\boldsymbol{\alpha} = \mathbf{K}_{I,N}^T \boldsymbol{\alpha}_I = \mathbf{V}^T \mathbf{L}^T \boldsymbol{\alpha}_I$ . The gradient of (3.18) with respect to  $\boldsymbol{\alpha}_I$  is given by

$$\frac{\partial \tau(\boldsymbol{\alpha}_I)}{\partial \boldsymbol{\alpha}_I} = -\boldsymbol{\alpha}_I^T \mathbf{L} \mathbf{M} \mathbf{L}^T + \mathbf{y}^T \mathbf{V}^T \mathbf{L}^T \stackrel{!}{=} \mathbf{0}^T$$

and leads with the necessary condition for a maximum the vector  $\boldsymbol{\alpha}_I = \mathbf{L}^{-T} \mathbf{L}_M^{-T} \boldsymbol{\beta}_I$  with  $\boldsymbol{\beta}_I = \mathbf{L}_M^{-1} \mathbf{V} \mathbf{y}$  as defined in (3.8). Due to the fact that  $-\mathbf{L} \mathbf{M} \mathbf{L}^T$  is negative definite, the sufficient condition for a maximum is also fulfilled. Note that  $\boldsymbol{\alpha}_I$  exactly corresponds to the posterior prediction vector in (3.8). Thus, the maximum of (3.18) results in

$$\tau_I = \max_{\boldsymbol{\alpha}_I} (\tau(\boldsymbol{\alpha}_I)) = \frac{1}{2} \boldsymbol{\beta}_I^T \boldsymbol{\beta}_I. \quad (3.19)$$

The increase in the sparse posterior likelihood (3.16) defines the selection criterion of Smola and Bartlett (SB) given by

$$\text{SB} \Delta_i = \tau_{I'} - \tau_I = \frac{1}{2} \beta_{I',i}^2, \quad (3.20)$$

for each remaining point  $\mathbf{x}_i$  and with the new component  $\beta_{I',i}$  of the updated vector  $\boldsymbol{\beta}_{I'} \in \mathbb{R}^{m+1}$ , see (A.36). The criterion (3.20) is only evaluated for a randomly chosen subset of  $\mathbf{X}_R$  with cardinality  $\kappa$  due to the high computational cost of  $\mathcal{O}(mn)$  for the criterion calculation per remaining point. This effort is caused by the required model update to calculate  $\boldsymbol{\beta}_{I'}$ , which additionally leads to higher memory

requirements. Smola and Bartlett (2001) recommend  $\kappa = 59$  and justify it with a probabilistic argument. Nevertheless, they end up with high computational cost of  $\mathcal{O}(\kappa nm^2)$  for the whole DTC approximation. The conjugation of this selection heuristic defines the corresponding deletion criterion

$$\text{SB}\nabla_i = \text{SB}\Delta_i, \quad (3.21)$$

which leads to cost of  $\mathcal{O}(m^2)$  per active point. Equivalently to Csató's deletion method, this effort is affected by rearranging the order of the active index set with element  $i$  at the last position.

To increase the performance of the selection heuristic (3.20), a matching pursuit approach (MPA) which reduces the computational effort, but not the memory requirements is presented by Keerthi and Chu (2006). Here,  $\boldsymbol{\alpha}_I \in \mathbb{R}^m$  is fixed when maximizing  $\tau(\boldsymbol{\alpha}_{I'})$  in (3.19) and only  $\alpha_{I',i}$  with  $i \in R$  is varied. Based on

$$\tau(\boldsymbol{\alpha}_{I'}) = \tau(\boldsymbol{\alpha}_I) + \alpha_{I',i}(\mathbf{k}_{N,i}^T(\mathbf{y} - \boldsymbol{\mu}_I) - \sigma^2 \mu_{I,i}) - \frac{\alpha_{I',i}^2}{2}(\sigma^2 k_{ii} + \mathbf{k}_{N,i}^T \mathbf{k}_{N,i})$$

following from Equation (3.17) with  $\boldsymbol{\alpha}_{R \setminus i} = \mathbf{0}$ , the insertion criterion

$$\begin{aligned} \text{MPA}\Delta_i &= \max_{\alpha_{I',i}} (\tau(\boldsymbol{\alpha}_{I'}) - \tau(\boldsymbol{\alpha}_I)) \\ &= \max_{\alpha_{I',i}} \left( \alpha_{I',i}(\mathbf{k}_{N,i}^T(\mathbf{y} - \boldsymbol{\mu}_I) - \sigma^2 \mu_{I,i}) - \frac{\alpha_{I',i}^2}{2}(\sigma^2 k_{ii} + \mathbf{k}_{N,i}^T \mathbf{k}_{N,i}) \right) \\ &= \frac{(\mathbf{k}_{N,i}^T(\mathbf{y} - \boldsymbol{\mu}_I) - \sigma^2 \mu_{I,i})^2}{2(\sigma^2 k_{ii} + \mathbf{k}_{N,i}^T \mathbf{k}_{N,i})} \end{aligned} \quad (3.22)$$

is obtained. Here  $\mathbf{k}_{N,i} \in \mathbb{R}^n$  is a covariance vector and the last step in this derivation is induced by the necessary condition for the gradient given by

$$\frac{\partial \tau(\boldsymbol{\alpha}_{I'})}{\partial \alpha_{I',i}} = \mathbf{k}_{N,i}^T(\mathbf{y} - \boldsymbol{\mu}_I) - \sigma^2 \mu_{I,i} - \alpha_{I',i}(\sigma^2 k_{ii} + \mathbf{k}_{N,i}^T \mathbf{k}_{N,i}) \stackrel{!}{=} 0.$$

However, despite the lower computational cost of  $\mathcal{O}(dn)$  per remaining point, on large data sets or for high input dimensions, they also select a randomized subset of size  $\kappa$  for criteria evaluation to boost efficiency. An additional matrix cache that contains a multiple of the corresponding  $\kappa$  rows of the full covariance matrix  $\mathbf{K}$  can help to speed up the criterion evaluations, but increases the memory requirements considerably.

Seeger et al. (2003) proposed a very fast greedy criterion with computational complexity of  $\mathcal{O}(1)$  per remaining point. They measure the information gain (IG) defined by

$$\text{IG}\Delta_i = \text{KL} [\tilde{\mathcal{Q}}_{I'}(\mathbf{f} | \mathbf{y}, \mathbf{X}) \| \mathcal{Q}_I(\mathbf{f} | \mathbf{y}, \mathbf{X})] \quad (3.23)$$

with the increase in the KL divergence between the current posterior distribution  $\mathcal{Q}_I(\mathbf{f} | \mathbf{y}, \mathbf{X})$  following from (3.9) and the approximated distribution  $\tilde{\mathcal{Q}}_{I'}(\mathbf{f} | \mathbf{y}, \mathbf{X})$  after inclusion of the remaining point  $\mathbf{x}_i$ . This simplified posterior approximation satisfies

$$\begin{aligned} \tilde{\mathcal{Q}}_{I'}(\mathbf{f} | \mathbf{y}, \mathbf{X}) &\propto q_I(\mathbf{y}_{N \setminus i} | \mathbf{f}_I, \mathbf{X}_{N \setminus i}) p(y_i | f_i, \mathbf{x}_i) p(\mathbf{f} | \mathbf{X}) \\ &= \mathcal{N}(\mathbf{y}_{N \setminus i} | \mathbf{K}_{I,N \setminus i}^T \mathbf{K}_{I,I}^{-1} \mathbf{f}_I, \sigma^2 \mathbf{I}) \mathcal{N}(y_i | f_i, \sigma^2) \mathcal{N}(\mathbf{f} | \mathbf{0}, \mathbf{K}), \end{aligned}$$

where the influence of the point  $\mathbf{x}_i$  is removed in the density  $q_I(\mathbf{y}_{N \setminus i} | \mathbf{f}_I, \mathbf{X}_{N \setminus i})$ . Thus, the couplings between the latent function value  $f_i$  and the targets  $\mathbf{y}_{N \setminus i}$ , i.e. without the  $i$ -th element, are ignored to guarantee low computational costs. For the long and tedious derivation of the explicit expression for  ${}_{\text{IG}}\Delta_i$  the reader is referred to Seeger (2003).

The intention of the following greedy selection criterion by Quiñonero-Candela (QC) is to increase the logarithmic marginal likelihood  $\varphi_I(\boldsymbol{\theta})$  obtained from Equation (3.10) by the inclusion of a remaining point. This leads to the equivalent criterion

$$\begin{aligned} {}_{\text{QC}}\Delta_i &= \varphi_{I'}(\boldsymbol{\theta}) - \varphi_I(\boldsymbol{\theta}) \\ &= \log(\sigma) - \log(l_{M,ii}) + \frac{\beta_{I',i}^2}{2\sigma^2} \\ &= \log(\sigma) - \log(l_{M,ii}) + \frac{\text{SB}\Delta_i}{\sigma^2}, \end{aligned} \quad (3.24)$$

where only the change influenced by the inclusion is considered, i.e. the hyperparameters  $\boldsymbol{\theta}$  are fixed during the selection process, cf. Quiñonero-Candela (2004). More details for the adaption of hyperparameters are presented in Section 3.2.1. The heuristic (3.24) is closely related to the criterion by Smola and Bartlett (2001), since  $\beta_{I',i}^2 \propto l_{M,ii}^{-2}$ , where  $l_{M,ii}$  is the  $i$ -th new diagonal element of  $\mathbf{L}'_M$ , cf. Equation (A.36). Consequently, this criterion leads to the same computational cost of  $\mathcal{O}(mn)$  per remaining point and is also calculated for only a small randomly selected remaining subset of size  $\kappa$ .

The last insertion criterion, which is discussed here, is introduced by Titsias (2009). His idea is the same as by Quiñonero-Candela (2004), but he increases the regularized logarithmic marginal likelihood (3.11) given by his variational (VAR) framework. This results in the insertion criterion

$$\begin{aligned} {}_{\text{VAR}}\Delta_i &= \text{VAR} \varphi_{I'}(\boldsymbol{\theta}) - \text{VAR} \varphi_I(\boldsymbol{\theta}) \\ &= {}_{\text{QC}}\Delta_i + \frac{\|\mathbf{k}_{N,i} - \mathbf{V}^T \mathbf{L}^{-1} \mathbf{k}_{I,i}\|^2}{2\sigma^2 \left( k_{ii} - \|\mathbf{L}^{-1} \mathbf{k}_{I,i}\|^2 \right)} \\ &= {}_{\text{QC}}\Delta_i + \frac{\|\mathbf{k}_{N,i} - \mathbf{V}^T \mathbf{L}^{-1} \mathbf{k}_{I,i}\|^2}{2\sigma^2 {}_{\text{CS}}\Delta_i} \end{aligned} \quad (3.25)$$

for active point selection. For more details have a look at the formula for the sequential model update (A.35) in the Appendix A.3.3. The relation to the criterion by Smola and Bartlett (2001) induces the same computational complexity of  $\mathcal{O}(mn)$  per remaining point. To increase performance, the disadvantageous sub-sampling on a randomly chosen subset of remaining training points is required again.

### 3.2.3 Efficient Maximum Error Insertion and Deletion

The insertion and deletion methods discussed in the previous section either lack modeling accuracy, have high memory requirements, or lack computational speed. Additionally, if the regression model is generated based on a purely randomized selection or on a method using a small randomly selected remaining subset for criteria evaluation, e.g. as done by Smola and Bartlett (2001), Quiñonero-Candela (2004), Keerthi and Chu (2006), or Titsias (2009), the performance in hard regression tasks deteriorates.

Our novel approach aims to provide a favorable compromise between modeling accuracy, computational cost, and memory requirements.

Firstly, let us present our successive greedy criterion for inclusion of training points into the active subset. Similar to the method by Smola and Bartlett (2001), our approach maximizes the posterior probability given in Equation (3.16). The resulting greedy scheme from (3.20) successively maximizes the Euclidean norm of the vector  $\boldsymbol{\beta}_{I'}$ . This task is equivalent to iteratively minimize the scaled training error, see Equation (2.36) for further details. Thus, it is necessary to minimize

$$n \text{ MSE} = \|\mathbf{y} - \boldsymbol{\mu}_{I'}\|^2 = \|\mathbf{y}\|^2 - 2\mathbf{y}^T\boldsymbol{\mu}_{I'} + \|\boldsymbol{\mu}_{I'}\|^2$$

for the target vector  $\mathbf{y}$  with fixed norm as well as for the estimated mean vector  $\boldsymbol{\mu}_{I'}$  with approximately constant norm, since we have  $\|\boldsymbol{\beta}_{I'}\|^2 = \boldsymbol{\beta}_{I'}^T\boldsymbol{\beta}_{I'} = \mathbf{y}^T\boldsymbol{\mu}_{I'}$  after an inclusion. Due to the equivalence of norms in finite dimensional spaces, cf. Lipschutz and Lipson (2013), it holds true that

$$\|\mathbf{y} - \boldsymbol{\mu}_{I'}\| \leq n \max_{j \in N} (|y_j - \mu_{I',j}|) .$$

Looking at the limit for increasing  $m$ , it follows that  $\boldsymbol{\mu}_{I'} \approx \boldsymbol{\mu}_I$  is approximately achieved and also the norm of  $\boldsymbol{\mu}_{I'}$  converges as required above. Hence, our new insertion criterion is defined by

$$\text{ME}\Delta_i = |y_i - \mu_{I,i}| , \quad (3.26)$$

where the remaining point which has the maximal error (ME) under the current posterior model (3.9), is always selected. This computationally efficient approach has  $\mathcal{O}(1)$  cost for criterion calculation per remaining point. The convergence assumption obviates the update of the posterior model for each remaining point as required for other selection criteria, as for example by Smola and Bartlett (2001) or by Quiñonero-Candela (2004).

Secondly, we present our maximum error deletion criterion for the removal of active points. Typically, the maximum number of active points  $m$  is predefined, since it influences computing time quadratically and memory requirements linearly. If a stopping criterion for  $m$  is used, for example by monitoring the averaged square training error as suggested by Seeger et al. (2003), deletion of appropriate active points improves the predictive performance without significantly deteriorating the existing model quality. The deletion also provides a way to reduce redundancy in the greedily selected active subset. Similar to the presented insertion strategy, a greedy criterion to successively delete active points is derived. Note that deleting an active point does not necessarily lead to a state that was previously encountered when iteratively inserting training points. The reason is that the underlying assumptions for greedy insertion and deletion differ considerably. Remember, that during the deletion process the QR decomposition  $\mathbf{QR} = \mathbf{LML}^T$  is employed to receive numerical stability. For our proposed technique, the cost for the deletion of one active point is equal to its insertion cost, i.e.  $\mathcal{O}(mn)$  in the  $m$ -th iteration of the DTC approximation. Inspired by the criterion of Csató and Oppér (2001), our new deletion criterion is defined as follows. Beginning with an already selected subset determined by  $I$ , the active point with minimal value with respect to the deletion criterion

$$\text{ME}\nabla_i = |\text{ME}\Delta_i \alpha_{I,i}| , \quad (3.27)$$



is removed, where  $\boldsymbol{\alpha}_I = \mathbf{R}^{-1} \mathbf{Q}^T \mathbf{K}_{I,N} \mathbf{y} \in \mathbb{R}^m$ . Note that the maximum error  $_{\text{ME}}\Delta_i$  is used instead of the expensive projection-induced error  $_{\text{CS}}\Delta_i$  by Csató and Opper (2001). Thus, the same low complexity for a deletion criterion evaluation of  $\mathcal{O}(1)$  per active point is obtained. Thereby, the error of an active training point in the current sparse model given by Equation (3.9) is coupled with its importance under prediction in relation to the behavior (3.16). Hence, our presented deletion criterion (3.27) controls the current model accuracy and the generalization capability.

### 3.3 Fully Independent Training Conditional Approximation

Another sparse likelihood approximation for standard GPR is presented by Snelson and Ghahramani (2006a). In contrast to the DTC approximation, a flexible data set of virtual training points  $\mathcal{D}_P = (\mathbf{f}_P, \mathbf{X}_P)$ , i.e. the so-called pseudo-inputs  $\mathbf{X}_P \in \mathbb{R}^{m \times d}$  with their corresponding latent function values  $\mathbf{f}_P \in \mathbb{R}^m$ , is introduced to realize the approximation. Hence, the independence of the data set  $\mathcal{D}_P$  to the given training data  $\mathcal{D}$  induces the designation as fully independent training conditional (FITC) approximation, cf. Quiñonero-Candela and Rasmussen (2005). The prior distribution

$$P(\mathbf{f}_P | \mathbf{X}_P) = N(\mathbf{f}_P | \mathbf{0}, \mathbf{K}_{P,P}) \quad (3.28)$$

is equivalently defined as (3.4) for the DTC approximation, but only with respect to the virtual data set  $\mathcal{D}_P$  which induces the covariance matrix  $\mathbf{K}_{P,P} \in \mathbb{R}^{m \times m}$ . If the definition of the prior according to  $\mathcal{D}_P$  is well chosen, then the distribution of the virtual data set  $\mathcal{D}_P$  should be identical to the distribution of the real data. Analogously to the former regression techniques, the noisy model (2.2) is considered. Hence, the approximated model likelihood  $q_P(\mathbf{y} | \mathbf{f}_P, \mathbf{X}_P, \mathbf{X})$  is induced through the prior conditional mean  $E_P[\mathbf{f} | \mathbf{f}_P, \mathbf{X}_P, \mathbf{X}] = \mathbf{K}_{P,N}^T \mathbf{K}_{P,P}^{-1} \mathbf{f}_P \in \mathbb{R}^n$  with covariance values between the pseudo-inputs and the real training inputs summarized in  $\mathbf{K}_{P,N} \in \mathbb{R}^{m \times n}$ . That results in

$$q_P(\mathbf{y} | \mathbf{f}_P, \mathbf{X}_P, \mathbf{X}) = \mathcal{N}(\mathbf{y} | \mathbf{P}_P^T \mathbf{f}_P, \boldsymbol{\Gamma} + \sigma^2 \mathbf{I}) \quad (3.29)$$

with the diagonal covariance matrix  $\boldsymbol{\Gamma} = \text{diag}(\text{diag}(\mathbf{K} - \mathbf{K}_{P,N}^T \mathbf{K}_{P,P}^{-1} \mathbf{K}_{P,N})) \in \mathbb{R}^{n \times n}$  and the projection matrix  $\mathbf{P}_P = \mathbf{K}_{P,P}^{-1} \mathbf{K}_{P,N} \in \mathbb{R}^{m \times n}$ . The independence between the latent function values  $\mathbf{f} \in \mathbb{R}^n$  and  $\mathbf{f}_P$  leads to the diagonal structure of  $\boldsymbol{\Gamma}$ . Compared to the DTC approximation, a more exact representation of the following approximated posterior is induced by the introduction of  $\boldsymbol{\Gamma}$  in (3.29). Using Equation (A.19), the theorem of Bayes together with the prior (3.28) and the model likelihood (3.29) leads to the approximated posterior density

$$\begin{aligned} q_P(\mathbf{f}_P | \mathbf{y}, \mathbf{X}_P, \mathbf{X}) &= \frac{q_P(\mathbf{y} | \mathbf{f}_P, \mathbf{X}_P, \mathbf{X}) p(\mathbf{f}_P | \mathbf{X}_P)}{q_P(\mathbf{y} | \mathbf{X}_P, \mathbf{X})} \\ &\propto q_P(\mathbf{y} | \mathbf{f}_P, \mathbf{X}_P, \mathbf{X}) p(\mathbf{f}_P | \mathbf{X}_P) \\ &= \mathcal{N}(\mathbf{y} | \mathbf{P}_P^T \mathbf{f}_P, \boldsymbol{\Gamma} + \sigma^2 \mathbf{I}) \mathcal{N}(\mathbf{f}_P | \mathbf{0}, \mathbf{K}_{P,P}) \\ &\propto \mathcal{N}(\mathbf{f}_P | \mathbf{L} \mathbf{M}^{-1} \mathbf{V} \mathbf{D}^{-1} \mathbf{y}, \mathbf{L} \mathbf{M}^{-1} \mathbf{L}^T), \end{aligned} \quad (3.30)$$

where the lower Cholesky factor  $\mathbf{L} \in \mathbb{R}^{m \times m}$  of  $\mathbf{K}_{P,P}$  according to Equation (A.1),  $\mathbf{V} = \mathbf{L}^{-1} \mathbf{K}_{P,N} \in \mathbb{R}^{m \times n}$ , the diagonal matrix  $\mathbf{D} = \mathbf{\Gamma} + \sigma^2 \mathbf{I} \in \mathbb{R}^{n \times n}$ , and  $\mathbf{M} = \mathbf{I} + \mathbf{V} \mathbf{D}^{-1} \mathbf{V}^T \in \mathbb{R}^{m \times m}$  are defined in the same manner as for the DTC approximation. Due to the identical structure of the approximated posterior (3.30) as for the DTC approximation, the FITC approximation is also referred as information-optimal, cf. Csató (2002). The normalization constant in the posterior density (3.30) is given by the approximated marginal likelihood

$$\begin{aligned} q_P(\mathbf{y} | \mathbf{X}_P, \mathbf{X}) &= \int_{\mathbb{R}^m} q_P(\mathbf{y} | \mathbf{f}_P, \mathbf{X}_P, \mathbf{X}) p(\mathbf{f}_P | \mathbf{X}_P) d\mathbf{f}_P \\ &= \int_{\mathbb{R}^m} \mathcal{N}(\mathbf{y} | \mathbf{P}_P^T \mathbf{f}_P, \mathbf{\Gamma} + \sigma^2 \mathbf{I}) \mathcal{N}(\mathbf{f}_P | \mathbf{0}, \mathbf{K}_{P,P}) d\mathbf{f}_P \\ &= \mathcal{N}(\mathbf{y} | \mathbf{0}, \mathbf{D} + \mathbf{V}^T \mathbf{V}), \end{aligned} \quad (3.31)$$

employing the law of total probability and Equation (A.21). The predictive Gaussian density for estimating an unknown target value  $f_*$  for the so far unseen test point  $\mathbf{x}_* \in \mathbb{R}^d$  follows by marginalization over the virtual function values  $\mathbf{f}_P$  and results in

$$\begin{aligned} q_P(f_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}_P, \mathbf{X}) &= \int_{\mathbb{R}^m} p(f_* | \mathbf{x}_*, \mathbf{f}_P, \mathbf{X}_P) q_P(\mathbf{f}_P | \mathbf{y}, \mathbf{X}_P, \mathbf{X}) d\mathbf{f}_P \\ &= \int_{\mathbb{R}^m} \mathcal{N}\left(f_* \mid \mathbf{k}_{P,*}^T \mathbf{K}_{P,P}^{-1} \mathbf{f}_P, k_{**} - \|\mathbf{L}^{-1} \mathbf{k}_{P,*}\|^2\right) \mathcal{N}(\mathbf{f}_P | \mathbf{L} \mathbf{M}^{-1} \mathbf{V} \mathbf{D}^{-1} \mathbf{y}, \mathbf{L} \mathbf{M}^{-1} \mathbf{L}^T) d\mathbf{f}_P \\ &= \mathcal{N}\left(f_* \mid \mathbf{k}_{P,*}^T \boldsymbol{\alpha}_P, k_{**} - \|\mathbf{L}^{-1} \mathbf{k}_{P,*}\|^2 + \|\mathbf{L}_M^{-1} \mathbf{L}^{-1} \mathbf{k}_{P,*}\|^2\right). \end{aligned} \quad (3.32)$$

Here, the Cholesky decomposition of  $\mathbf{M} = \mathbf{L}_M \mathbf{L}_M^T$  related to Equation (A.1),  $\boldsymbol{\beta}_P = \mathbf{L}_M^{-1} \mathbf{V} \mathbf{D}^{-1} \mathbf{y} \in \mathbb{R}^m$ , the prediction vector  $\boldsymbol{\alpha}_P = \mathbf{L}^{-T} \mathbf{L}_M^{-T} \boldsymbol{\beta}_P \in \mathbb{R}^m$ , and Equation (A.21) are used. Furthermore, the conditional density  $p(f_* | \mathbf{x}_*, \mathbf{f}_P, \mathbf{X}_P)$ , which was employed in the first step of the derivation (3.32), follows from the joint Gaussian distribution

$$\begin{pmatrix} \mathbf{f}_P | \mathbf{X}_P \\ f_* | \mathbf{x}_* \end{pmatrix} \sim \mathbf{N} \left( \begin{pmatrix} \mathbf{f}_P \\ f_* \end{pmatrix} \middle| \begin{pmatrix} \mathbf{0} \\ 0 \end{pmatrix}, \begin{pmatrix} \mathbf{K}_{P,P} & \mathbf{k}_{P,*} \\ \mathbf{k}_{P,*}^T & k_{**} \end{pmatrix} \right)$$

conditioned on the Gaussian prior (3.28) by using Equation (A.22). Thereby, the covariance vector  $\mathbf{k}_{P,*} \in \mathbb{R}^m$  contains the values of the specified kernel between the pseudo-inputs and the test point.

The main computational effort of the FITC approximation is induced by storing and operating with the covariance matrix  $\mathbf{K}_{P,N}$ , which leads to cost in  $\mathcal{O}(nm^2)$  and memory requirements in  $\mathcal{O}(mn)$ . For the mean estimation of unknown target values, the prediction vector  $\boldsymbol{\alpha}_P$  should be precomputed to enable fast calculations in  $\mathcal{O}(dm)$ . The predictive variance of one test point can be computed in  $\mathcal{O}(m^2)$  if the data dimension  $d$  is smaller than the number of pseudo-inputs  $m$ . All of these computational assessments are equal to those of the DTC approximation. Additionally, a few extensions of the FITC framework are provided by Snelson and Ghahramani (2006b), where some of them can also be applied in the previously presented DTC approximation.

### 3.3.1 Determining Hyperparameters and Pseudo-inputs

In the FITC framework, the technique for model selection is the same as in the case of standard GPR by employing optimization approaches based on the MLM. Moreover, the pseudo-inputs  $\mathbf{X}_P$  are also treated as hyperparameters and were simultaneously optimized with all other hyperparameters summarized in  $\boldsymbol{\theta}$ . In contrast to the adaption of hyperparameters for the DTC approximation in Section 3.2.1, EM algorithms are not necessary. Nevertheless, the  $m$  virtual training points must be initialized where the random choice of real training points should be permitted due to the independence assumption for the approximation of the model likelihood (3.29). An initialization with prototypes resulting from data clustering algorithms like neural gas, see Martinetz et al. (1993), provide an appropriate alternative. Finally, the approximated marginal likelihood (3.31) induces the maximization problem

$$\varphi_P(\boldsymbol{\theta}) = \log(q_P(\mathbf{y} | \mathbf{X}_P, \mathbf{X}, \boldsymbol{\theta})) \rightarrow \max_{\boldsymbol{\theta}, \mathbf{X}_P} \quad (3.33)$$

to determine an optimal set of hyperparameters  $\boldsymbol{\theta}$  and pseudo-inputs  $\mathbf{X}_P$ , respectively. The derivation of the approximated logarithmic marginal likelihood

$$\varphi_P(\boldsymbol{\theta}) = -\frac{n}{2} \log(2\pi) - \frac{1}{2} \sum_{i=1}^m \log\left(k_{ii} - \|\mathbf{L}_M^{-1} \mathbf{k}_{P,i}\|^2 + \sigma^2\right) - \sum_{i=1}^m \log(l_{M,ii}) - \frac{1}{2} (\mathbf{y}^T \mathbf{D}^{-1} \mathbf{y} - \boldsymbol{\beta}_P^T \boldsymbol{\beta}_P) \quad (3.34)$$

is shown in the Appendix A.3.3 in Equation (A.38). Gradient-based optimization techniques like CG algorithms are employed to maximize  $\varphi_P(\boldsymbol{\theta})$ , where the gradient of (3.34) with respect to the hyperparameters and pseudo-inputs is provided in the Appendix A.3.4. The cost for one gradient calculation is given by  $\mathcal{O}(nm^2)$ . Thus, the adaption of the pseudo-inputs induces a more complex and expensive optimization problem since the number of all hyperparameters is increased to  $dm + |\boldsymbol{\theta}|$ . Hence, the number of steps for the CG solver should be chosen adequately, i.e. dependent of the size of the pseudo-input set  $m$ . Additionally, for large virtual pseudo sets  $\mathcal{D}_P$  the optimization can get stuck in flat regions, poor local maxima or even fail completely, cf. Snelson (2007). Nevertheless, the optimization of the pseudo-inputs is essential for the quality of the resulting sparse GPR model.

## 3.4 Evaluations

In this section, our maximum error (ME) insertion and deletion criterion is compared with the other methods for the DTC approximation presented in Section 3.2.2 on various modeling problems. Furthermore, the FITC approximation as explained in Section 3.3 is considered to present an extensive comparison.

### 3.4.1 Benchmark Data Sets

First of all, learned inverse dynamics models are compared on three types of real benchmark data sets. On the one hand, collected data from the seven Degree of Freedom (DoF) SARCOS master arm (13922

training and 5 569 test points), and on the other hand a simulation data set from the SARCOS model (14 904 training and 5 520 test points) is used as well as real robot data from the Barrett WAM arm (12 000 training and 3 000 test points), see Nguyen-Tuong et al. (2009) and Nguyen-Tuong and Peters (2011) for more details. The three robot data sets contain independent training and test points for all DoFs. Each point of the data sets has 21 input dimensions, i.e. position, velocity and acceleration of the seven DoFs, and seven targets, i.e. one torque for each DoF of the SARCOS and Barrett robot arms. Our goal is to learn the inverse dynamics models, which means the mapping from position, velocity, and acceleration to the corresponding torque for each joint. In this section, for all experiments the stationary squared exponential covariance function as defined in Equation (2.27) is used.

The first evaluation considers the quality of sparse model selection with the generalized EM algorithm from Section 3.2.1 for all insertion criteria of the DTC approximation and also for the FITC approximation on the first DoF from real SARCOS training data. For our experiments only the first DoF is chosen, because it is one of the hardest modeling tasks of the SARCOS data sets. Table 3.1 summarizes the results on the training data with respect to the NMSE (in percent) and the required learning time (in seconds). The table additionally shows the negative logarithmic marginal likelihood (NLML) for each of the various methods. The results for the DTC approximation are generated with ten EM steps (see Section 3.2.1) on an active set with  $m = 2000$  elements to reach convergence. Thus, for each selection criterion iterative model training starts with the same hyperparameter initialization, then selects an active set with 2000 elements, and performs 30 gradient ascents in the maximization step. For the FITC approximation also 2000 virtual training points are used and the resulting 42 023 hyperparameters are adapted with 650 conjugated gradient steps. Note that, for the FITC approximation, the NLML is equal to  $-\varphi_P(\boldsymbol{\theta})$  which follows from Equation (3.34). For the DTC approximation and the variational (VAR) approach by Titsias (2009), the regularized NLML depending on the negation of Equation (3.11) and, for all remaining insertion criteria, the NLML with respect to the negated Equation (3.10) is contained in Table 3.1. All experiments were repeated ten times and the averaged results over ten repetitions are reported in the presented table. Our maximum error (ME) selection outperforms all other intelligent

DoF 1	FITC	DTC insertion criterion							
		ME	IG	CS	MPA	SB	QC	VAR	Random
NMSE [%]	0.151	0.087	0.091	0.107	0.086	0.072	0.071	0.091	0.131
NLML	-1184.6	5411.1	5519.1	6752.4	5204.6	4528.6	4332.6	6472.6	7768.4
Time [s]	19957	7202	7262	18350	7640	10688	10680	12138	7172

Table 3.1: Sparse model selection for different insertion criteria of the DTC approximation and the FITC approximation for the first DoF of the real SARCOS training data. The first row describes the training accuracy after the learning process with respect to the NMSE. In the second row the negative logarithmic marginal likelihood (NLML) is considered. The last row reports on the complete model learning time in seconds for all methods. Our novel DTC insertion criterion (ME) is as fast as the randomized selection, while remaining competitive in learning performance to all other regression methods.

selection methods in computing time, yields accurate NMSE results and a stable approximation of the approximated logarithmic marginal likelihood  $\varphi_I(\boldsymbol{\theta})$ . Besides that, the high computational cost of some of the selection criteria such as  $_{\text{SB}}\Delta_i$  by Smola and Bartlett (2001),  $_{\text{QC}}\Delta_i$  by Quiñonero-Candela (2004),  $_{\text{MPA}}\Delta_i$  by Keerthi and Chu (2006), or  $_{\text{VAR}}\Delta_i$  by Titsias (2009) induced significantly longer training times. The variance of the obtained hyperparameters generated by the generalized EM algorithm is very low for all intelligent, i.e. not purely random, selection methods. Moreover, the obtained hyperparameters from the variational learning approach by Titsias (2009) nearly match the other hyperparameters, since 2000 active points are sufficient for a good Nyström approximation of the full covariance matrix. That means, that the additional regularization term in the approximated marginal likelihood (3.11) nearly cancels out in this case. Only the hyperparameters of the FITC approximation vary very much with respect to the DTC approximation, because they are influenced by the 2000 optimized virtual training points. The huge number of hyperparameters in the FITC approximation caused the low NLML results and the high learning time from their adaption with 650 gradient ascends. This sparse GPR method by Snelson and Ghahramani (2006a) has also the lowest prediction accuracy regarding the NMSE value in comparison with all other DTC type approaches.

The convergence trends with respect to the NMSE for all discussed sparse GPR approximations on the first DoF from the real SARCOS test data are shown in Figure 3.2. Here, for all DTC type approaches, the hyperparameters were learned in ten EM steps and randomized active point selection up to a final set size of  $m = 2000$  to prevent influences of the intelligent criteria. Remember that a remaining set size of  $\kappa = 59$  is always used to speed up the criteria calculation of the computationally intensive methods. To enable a fair comparison, the number of gradient steps in the FITC approximation is

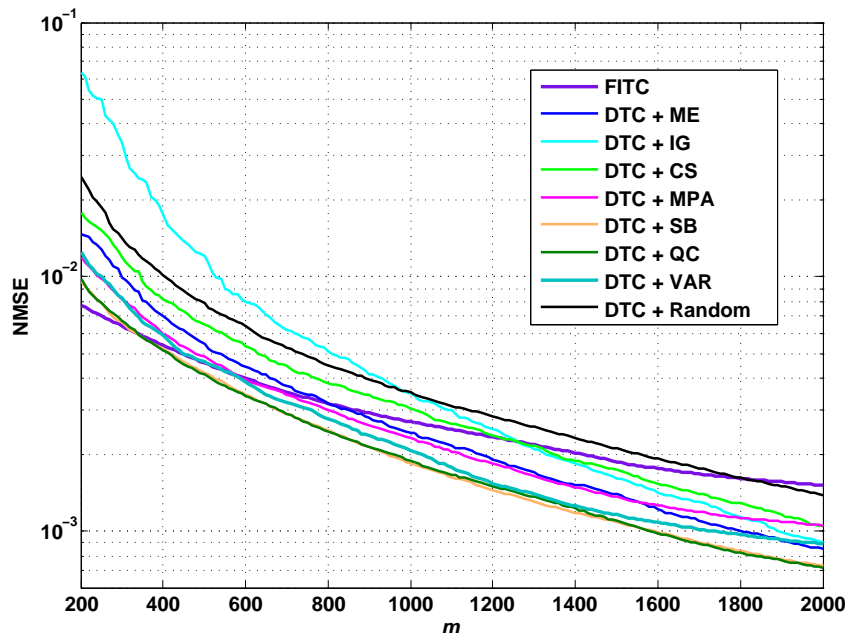


Figure 3.2: The convergence trend according to the NMSE on the first DoF from the real SARCOS test data for all discussed sparse GPR approximations. Our new developed ME sparsification strategy is closely related to the best performing methods and, for example, beats the selection techniques by Seeger (IG) and by Csató (CS).

linearly adapted with increasing virtual training points, i.e.  $150 + \frac{m}{4}$  optimization steps are used, because the number of hyperparameters in the FITC approximation also grows linearly with  $m$ . The NMSE results for randomized selection in the DTC approximation are averaged over ten runs. Furthermore, for large active set sizes nearly the same accuracy as the selection heuristic by Smola and Bartlett (2001) or Quiñonero-Candela (2004) is reached and the matching pursuit approach (MPA) by Keerthi and Chu (2006) is outperformed, cf. Figure 3.2. The complete learning times in Figure 3.3 and the NMSE values from Figure 3.2 explained before were captured every tenth active or virtual training points for all learning curves. The insertion curves by Smola and Bartlett (2001) and Quiñonero-Candela (2004) nearly match in cost and NMSE results which confirms the theoretical remarks of Section 3.2.2. The variational framework by Titsias (2009) leads to constantly higher effort in the learning process, e.g. compared to the curve by Quiñonero-Candela (2004), since the regularization term increases the effort for gradient-based optimization techniques. Our maximum error (ME) approach outperforms all DTC selection criteria with respect to the training times for low NMSE values on test data, see Figure 3.4. Since our insertion criterion yields the lowest learning curve, the best trade-off between model accuracy and computation time is provided. Interestingly, the gain in accuracy of the methods by Smola and Bartlett (2001) and Quiñonero-Candela (2004) disappear with respect to the randomized selection if only the computing times for criterion evaluation are considered.

For the evaluation of the DTC deletion schemes presented in Figure 3.5 only one random model training with fixed hyperparameters is used to demonstrate all effects in the removal process caused by the various criteria. Here, all learning curves were again captured every tenth active training points. Moreover, the designation is employed with a minus, e.g. by DTC – ME, to distinguish between deletion strategies (–) and insertion heuristics (+). Regarding Figure 3.5, our novel strategy outperforms the DTC deletion

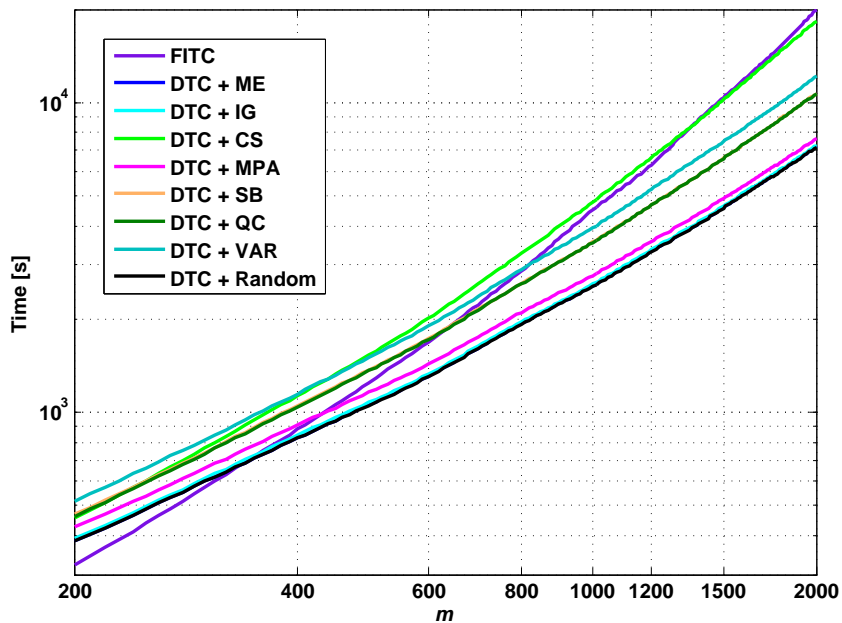


Figure 3.3: Complete learning times for the sparse GPR approximations, where our maximum error (ME) insertion method matches with Seeger’s (2003) information gain (IG) and the randomized insertion. Note that all axes are logarithmically scaled.

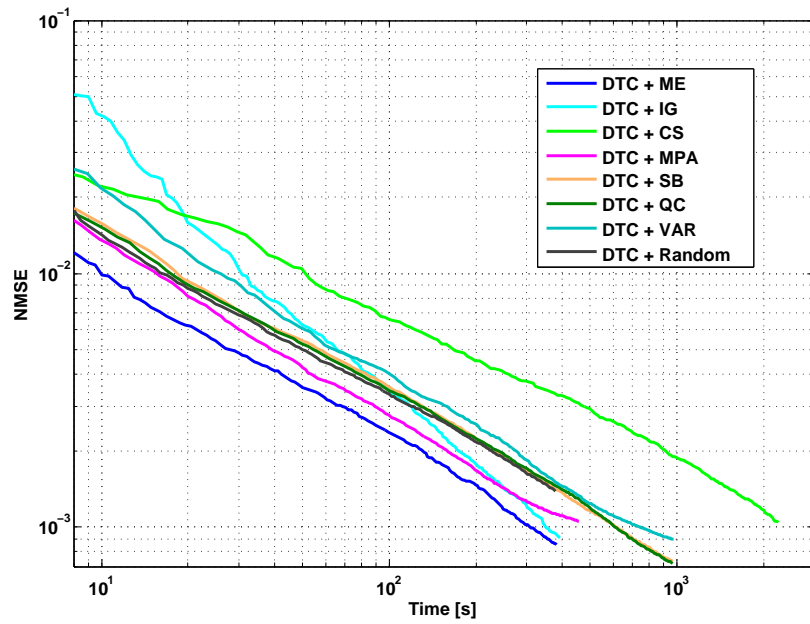


Figure 3.4: This plot shows that our maximum error (ME) selection method outperforms all other DTC type approaches in terms of the obtained accuracy level with respect to NMSE values on the real SARCOS test data for the first DoF depending on the pure insertion time.

criterion by Smola and Bartlett (2001) and the randomized version with respect to generalization accuracy. Therein, the best active point deletion algorithm is given by Csató and Opper (2001). The results of the randomized removal are again averaged over ten runs, which are only based on the same trained DTC

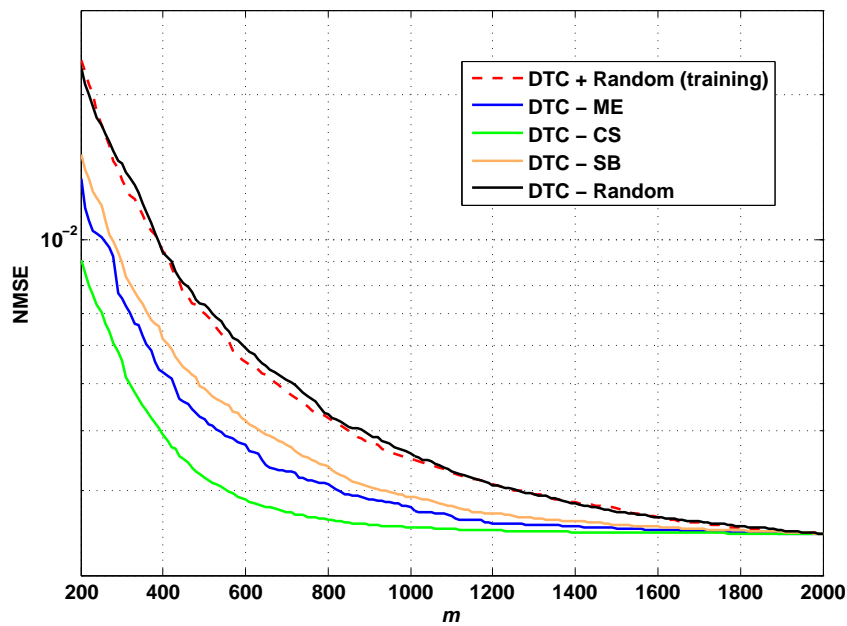


Figure 3.5: Convergence trends in NMSE on the first DoF from the real SARCOS test data for all discussed deletion criteria of the DTC approximation. Note that all deletion schemes yield better NMSE results than the simple, randomized deletion, which does not improve the resulting training error for specific active set sizes  $m$ .

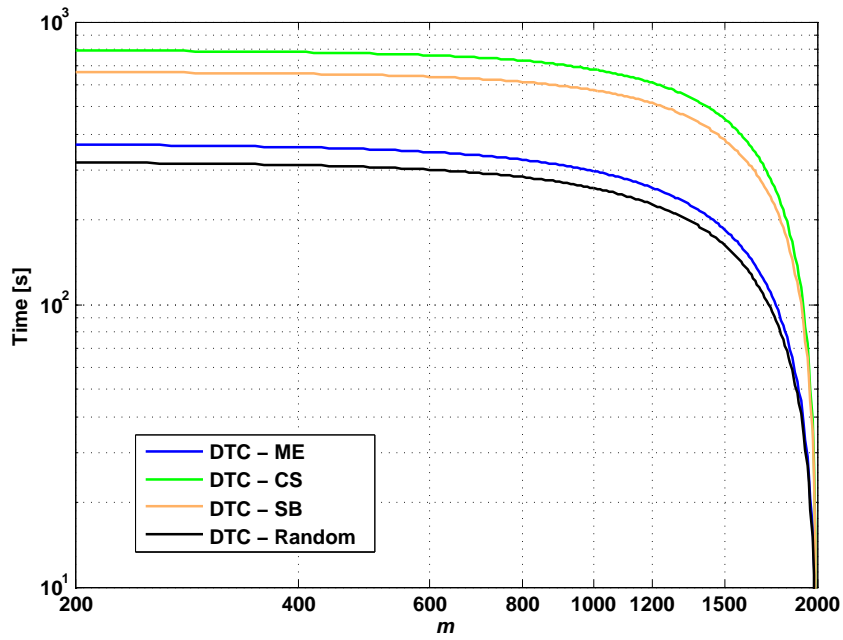


Figure 3.6: This plot illustrates the computing time of the various DTC deletion criteria depending on the current active set size  $m$  for the real SARCOS test data. It is helpful to read the curves from right to left.

regression model. The deletion criterion by Csató and Opper (2001) has the highest computational cost as shown in Figure 3.6. Our DTC – ME criterion results in a good compromise between low computational effort and high prediction precision as shown in Figure 3.7. All intelligent deletion schemes yield better

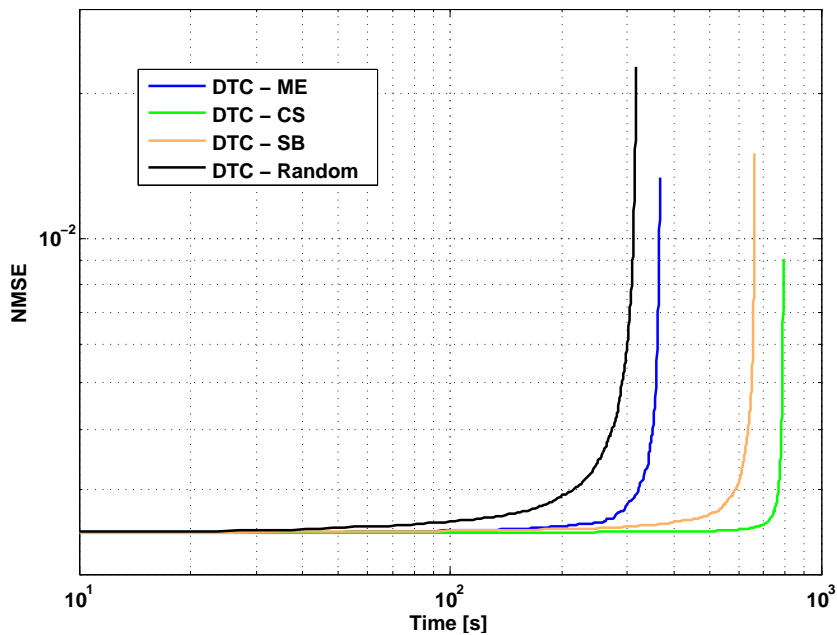


Figure 3.7: This figure demonstrate that our novel ME deletion method outperforms all other DTC deletion criteria with respect to NMSE values depending on training time of the same SARCOS test data. All the axes are logarithmically scaled.



NMSE results than the randomized deletion, and thus enable a strong increase in prediction performance since nearly half of the number of active points can be excluded from the randomly selected active set without a high decrease in model quality, cf. Figure 3.5. This indicates high redundancy in the greedily selected active set when using randomized selection strategies for model learning.

In Figure 3.9, our maximum error selection criterion for the sparse DTC approximation (DTC + ME) is compared with the FITC approximation by Snelson and Ghahramani (2006a) and other established regression procedures on all DoFs for the real and simulated SARCOS test data. Moreover, the real robot data from the Barrett WAM arm is considered and also evaluated for all seven DoFs with the presented and further mentioned techniques in Figure 3.10. The results for the other methods, i.e. local Gaussian processes (LGPs),  $\nu$ -SVR, standard GPR, and random Fourier regularized least squares (RFRLS), are taken from Nguyen-Tuong et al. (2009) and Gijbets (2011). For a fair comparison, also a final set size of  $m = 2000$  active points or pseudo-inputs is employed for both sparse GP approximations, respectively. Again, ten generalized EM steps are used for hyperparameter learning of the DTC approximation and 650 gradient ascents for optimization of virtual training points with subsequent prediction. Compared to these regression algorithms, our approach is efficient and one of the best performing methods. The higher errors for the fifth and sixth DoF on the real SARCOS test data are due to more complex non-linearities in these DoFs, e.g. induced by their joint inertia and wiring. Compared



Figure 3.8: SARCOS robot arm.

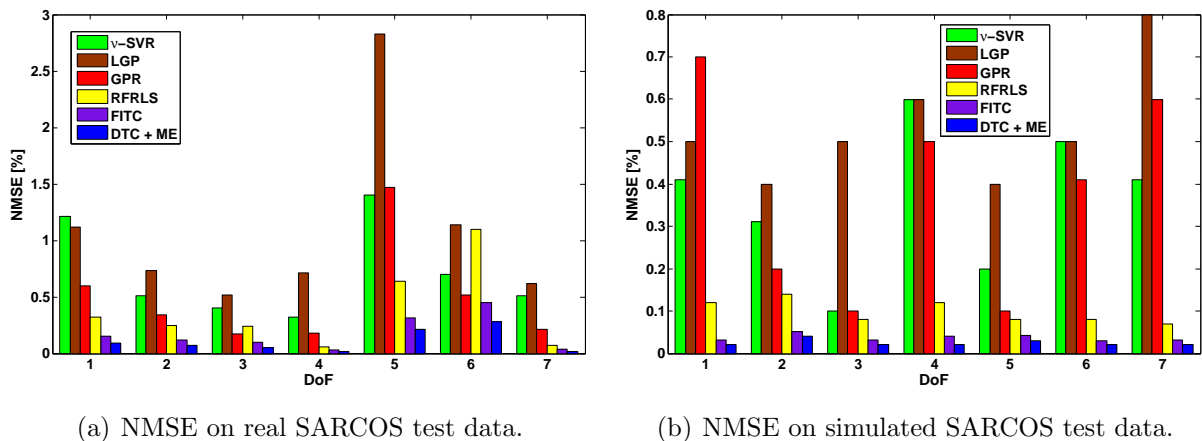
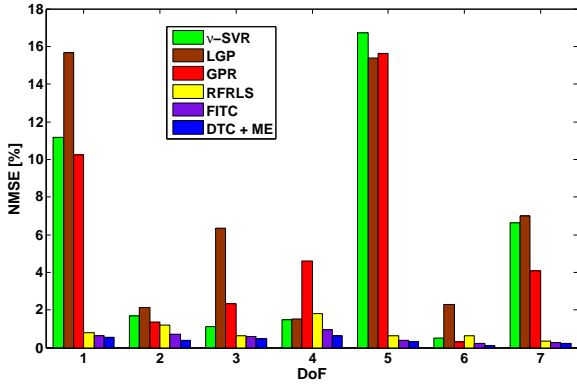


Figure 3.9: NMSE diagrams (in percent) for each degree of freedom (DoF) after prediction on the test sets with real SARCOS data (a) and simulated robot data (b) from the SARCOS master arm in Figure 3.8. Overall, the DTC approximation with the maximum error (ME) criterion performs best, closely followed by the FITC approximation. The standard GPR does not perform well on these large data sets due to suboptimal hyperparameters optimized from a subset of the training data, cf. Nguyen-Tuong et al. (2009).



(a) NMSE on real Barrett WAM test data.



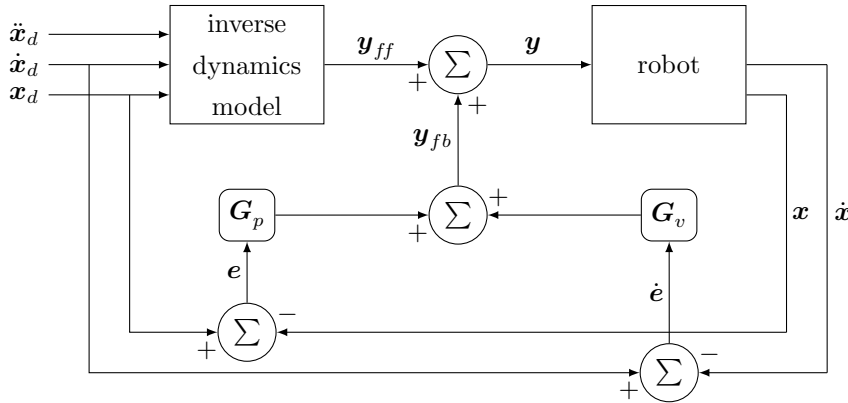
(b) Barrett WAM arm.

Figure 3.10: NMSE diagram (in percent) for each degree of freedom (DoF) after prediction on the test set with real robot data (a) from the Barrett WAM arm (b). Analogously to the results in Figure 3.9, the DTC approximation with the maximum error (ME) criterion performs best, closely followed by the FITC approximation.

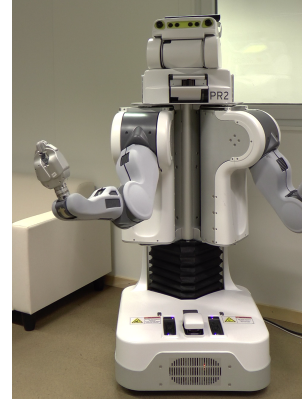
to the FITC approximation, we have much less effort for sparse model selection with higher generalization accuracy. It can be observed that DTC + ME performs well on all provided data sets, and returns better results than standard GPR. The reason for this behavior is that the hyperparameters of the standard GPR approach are optimized using only a subset of the original data sets due to the costly optimization of the marginal likelihood (2.31). Note that this approach is a common procedure in order to reduce the computational cost for hyperparameter optimization when using standard GPR models for large data sets, cf. Rasmussen and Williams (2006) and Nguyen-Tuong et al. (2009). However, depending on the subset selection for the hyperparameter optimization, the learned hyperparameters might be suboptimal and do not necessarily reflect the global data structure. The incremental training of the hyperparameters for the DTC approximation during the selection process using EM, as shown in Section 3.2.1, might represent a good alternative. Comparing DTC + ME and LGP, it should be noted that LGP is designed for applications in real-time online learning, where the learning speed is more important than accuracy, and thus it is not competitive in an offline comparison.

### 3.4.2 Compliant and Real-time Robot Control

In this section, learned inverse dynamics models for a real-time tracking control task on a PR2 robot, as shown in Figure 3.11(b), are employed. Here, the model-based tracking control law determines the joint torques  $y$  for each of the seven DoFs necessary to follow a desired joint trajectory  $\mathbf{x}_d, \dot{\mathbf{x}}_d, \ddot{\mathbf{x}}_d$ , where  $\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}$  are joint angles, velocities, and accelerations of the robot, as presented in Figure 3.11(a). This control paradigm uses an inverse dynamics model, while employing feedback in order to stabilize the system. Hence, the inverse dynamics model of the robot can be used as a feed-forward model that predicts the joint torques  $y_{ff}$  required to perform the desired trajectory, while a feedback term  $y_{fb}$  ensures the stability of the tracking control with a resulting control law of  $y = y_{ff} + y_{fb}$ . The feedback term can be a linear control law such as  $y_{fb} = \mathbf{G}_p e + \mathbf{G}_v \dot{e}$ , where  $e = \mathbf{x}_d - \mathbf{x}$  denotes the tracking error and



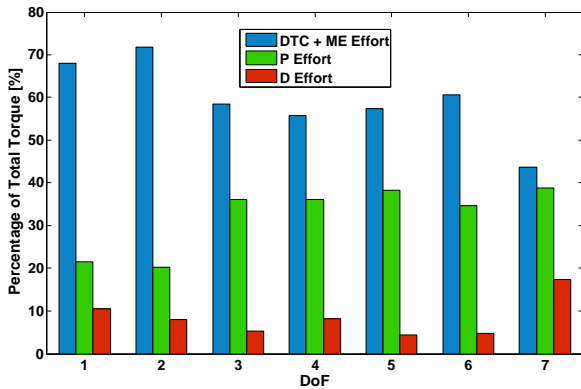
(a) Feed-forward control scheme.



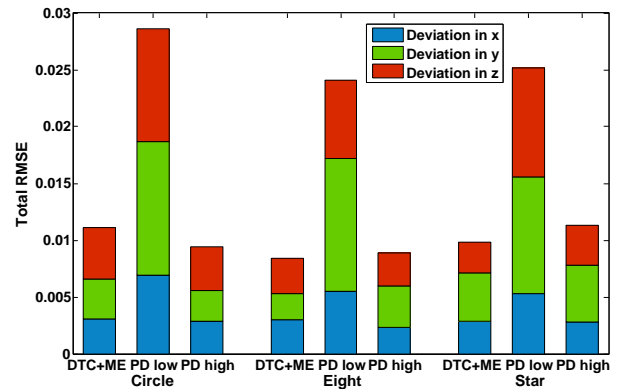
(b) The PR2 robot.

Figure 3.11: Feed-forward control scheme with inverse dynamics model (a) for compliant and real-time tracking control of the PR2 robot (b).

$\mathbf{G}_p$  as well as  $\mathbf{G}_v$  the position-gain and the velocity-gain, respectively. If an accurate inverse dynamics model can be obtained, the feed-forward term  $\mathbf{y}_{ff}$  will mostly cancel the robot's non-linearities, cf. Spong et al. (2006). In this case, the gains  $\mathbf{G}_p$  and  $\mathbf{G}_v$  can be chosen to have small values enabling compliant control performance, see Nguyen-Tuong et al. (2008) for more details. To obtain a global and precise inverse dynamics model, 517 783 data points with a frequency of 100 Hz are sampled from the right arm of the PR2 robot. Furthermore, a sparse GP model, obtained by the DTC approximation with our novel maximum error criterion (DTC + ME) and the same settings as in Section 3.4.1, was trained for each



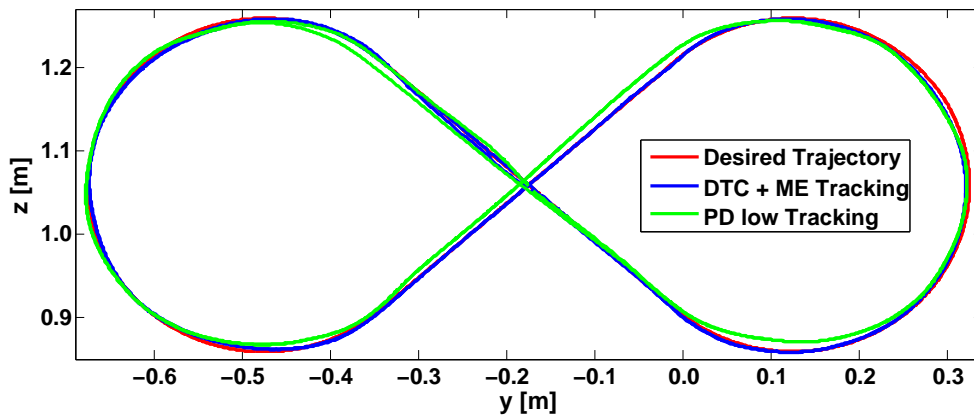
(a) Torque percentage for each DoF.



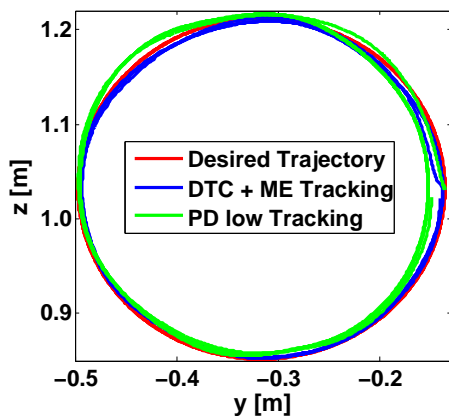
(b) Tracking error on three test trajectories.

Figure 3.12: The torque percentage for each DoF of the right PR2 arm is shown in Figure (a). The higher the torque percentage is, the more contributions have the corresponding parts of the control scheme 3.11(a). Here, the DTC + ME inverse dynamics models have usually significantly above 50% torque contribution. Figure (b) shows the tracking errors in task-space ( $x, y, z$ ) for the three test trajectories of Figure 3.13, namely, circle-, eight-, and star-shape. The RMSE value of each dimension is computed for three different control schemes, in fact, low-gain DTC + ME model-based control, PD-control with low gains, and PD-control with high gains, where about four times higher gains as in the low gain control schemes are achieved.

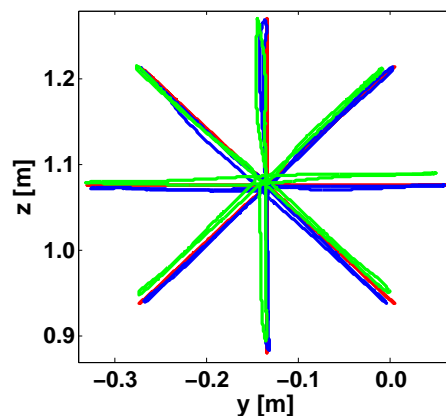
of the seven DoFs. Thereby, the hyperparameters are always learned with ten generalized EM steps, as explained in Section 3.2.1. A final active set size of  $m = 1000$  is chosen, which is sufficient to reach a good model quality and still results in prediction times of less than  $3\text{ ms}$  for all seven DoFs. Thus, the overall tracking control for the right PR2 arm can be performed in real-time at  $100\text{ Hz}$ . In Figure 3.12(a), the percentage on total torque for each DoF of the right PR2 robot arm is shown, where the gains for feedback control are chosen very small in order to enable compliant control. The contribution of our sparse GPR model, i.e. approximated with DTC + ME, to the control effort is usually far in excess of 50%. In detail, a high contribution to the control effort indicates a well approximated inverse dynamics model, since the feedback control loop does not need to strongly intervene during the control task. The corresponding tracking performance in the task-space is presented in Figure 3.13 for three test trajectories, namely, circle-, eight- and star-shape. Here, we compare the low gain feed-forward control using the learned inverse dynamics model with the standard PD-control scheme. The results with respect to the root mean square error (RMSE), cf. Equation (2.37), are shown in Figure 3.12(b). It can be seen that, applying learned inverse dynamics models, we obtain compliant tracking control, and, at the same time, have tracking accuracy comparable to the high gain PD-control scheme.



(a) Tracking on eight-shapes.



(b) Tracking on circle-shapes.



(c) Tracking on star-shapes.

Figure 3.13: Tracking performance on the three test trajectories using a low-gain DTC + ME model-based controller and PD-controller with low gains.

### 3.4.3 Simulation of Vehicle Power Demand

Modern vehicles comprise a multitude of electronic components. Especially, electric or hybrid vehicles have high requirements on the power supply of the individual electronic components. To obtain an optimal component layout and energy balance, a profound understanding of the energy needs in all operating modes is indispensable. In this section, it is shown that data-based modeling approaches like sparse GPR techniques are suitable to accurately model the dynamics of the power consumption of electronic components, in particular, of the electronic power steering (EPS) assistance systems, see Figure 3.14. This component is considered due to the fact that the conventional electro-hydraulic power steering (EHPS) systems are increasingly replaced by the purely electronic ones. The main reasons therefore lie in the large standby power consumption and the higher complexity of the EHPS system, since it consists of different technical components. Due to the modular design of the EPS assistance system, which can be adapted to constructional conditions and its better performance, it is expected to supplant the EHPS version in the future. To integrate the advanced steering assistance system with regard to the energy demand to be provided from the electrical system in the development process, simulations on substitute models are necessary. So far, the physical modeling approaches have only yielded unsatisfactory results in terms of accuracy and generalization capabilities. A more detailed analysis of this complex technical issue can be found in Schreiter et al. (2013) and the references therein. The starting point for our data-based modeling approach is a representative training data set. This involves a measurement with respect to the four main input variables of the dynamic EPS system. Unfortunately, for reasons of confidentiality it is not allowed to name and describe these input features in more detail. The four specified input variables cannot be measured on a test bench, because dynamic effects are not reproducible, for example, if they occur as response of load changes. Furthermore, the latter has the consequence that a transient DoE can not be employed. In order to receive meaningful training data, different driving maneuvers were carried out with the available vehicle on a test track. In a measurement interval of 10 *ms* the four input variables and the power consumption  $P(t)$  of the EPS system were recorded from the engine control unit (ECU). Figure 3.15 shows the power curve in dependence of the performed driving situations (all under dry conditions) on the generated data set with 415 765 training points. Among the variety of maneuvers like slalom and lane changing, the famous elk test (VDA lane change) can be found. To verify the generated regression models, additional test data from everyday drive scenarios were measured. Finally,

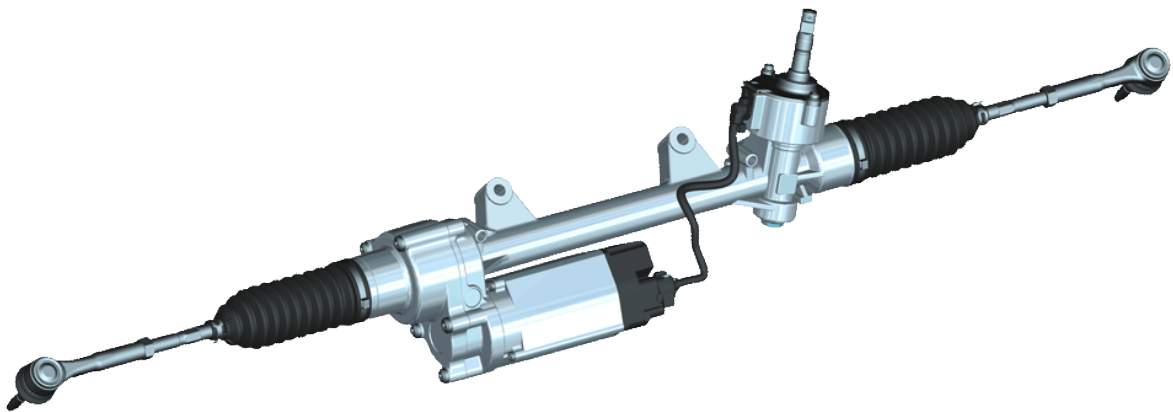


Figure 3.14: The electronic power steering (EPS) assistance system with paraxially servo unit.

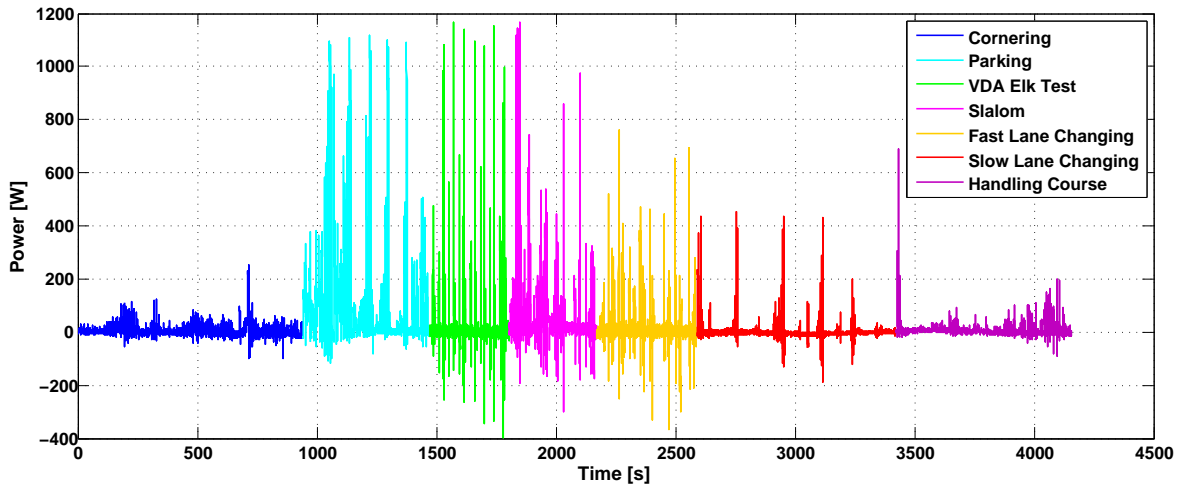


Figure 3.15: Power consumption of the EPS assistance system depending on various driving situations.

The power of the system varies strongly, particularly at the extreme driving maneuvers such as the VDA lane changing test. The negative power appears due to the steering mechanism which forces the car to drive straight.

test drives on dry and wet roads were conducted. In total, about 280 000 test points are available for evaluation. A complete NARX(0,1) approach is employed as basic modeling structure for the power consumption  $P(t)$  of the EPS assistance system. Thus, the input dimension increases to eight, because only four time-delayed input features are recirculated. A time-delayed feedback of target function values, concerning the power consumption of the EPS system, is proved to be not necessary. Hence, only an underlying nonlinear and exogenous (NX) structure for training and subsequent prediction of test points is used in all regression models. For transient modeling, the same sparse GPR techniques as before are considered, i.e. the DTC approximation with different greedy selection criteria, except for the approaches by Quiñero-Candela (2004) and Titsias (2009), because of their strong relation to the method by Smola and Bartlett (2001). Also the FITC approximation is neglected in this case due to their high training times without improving model accuracy as mentioned earlier. Instead, the SoD approach was considered with a subset size of  $m = 10\,000$ , where the points are randomly selected and the averaged results over ten repetitions are presented. Moreover, the  $\nu$ -SVR algorithm as provided by Chang and Lin (2011) is considered as an example for non-Bayesian modeling techniques. For the SoD approach and the sparse GPR models, only a linear mean function as in Equation (2.24) is used and the non-stationary neural network covariance function from Equation (2.30) is applied. All of the 19 obtained hyperparameters were adapted by gradient-based optimization techniques that are evaluated to maximize the logarithmic marginal likelihood (3.10) with our presented EM scheme, cf. Section 3.2.1. To evaluate the performance of the used methods, the NRMSE (2.38) and the NMAE (2.41) from Section 2.2.1 are utilized.

Firstly, the selected greedy insertion criteria for the creation of the sparse regression models under the DTC approximation are compared to the SoD and  $\nu$ -SVR technique on training data. Figure 3.16 and 3.17 show the convergence trends in the NRMSE and NMAE under successive inclusion of training points up to a final active set size of  $m = 1\,000$ , respectively. The information gain (IG) criterion and the expensive heuristic from Smola and Bartlett (2001) yield nearly the same NRMSE results on this large

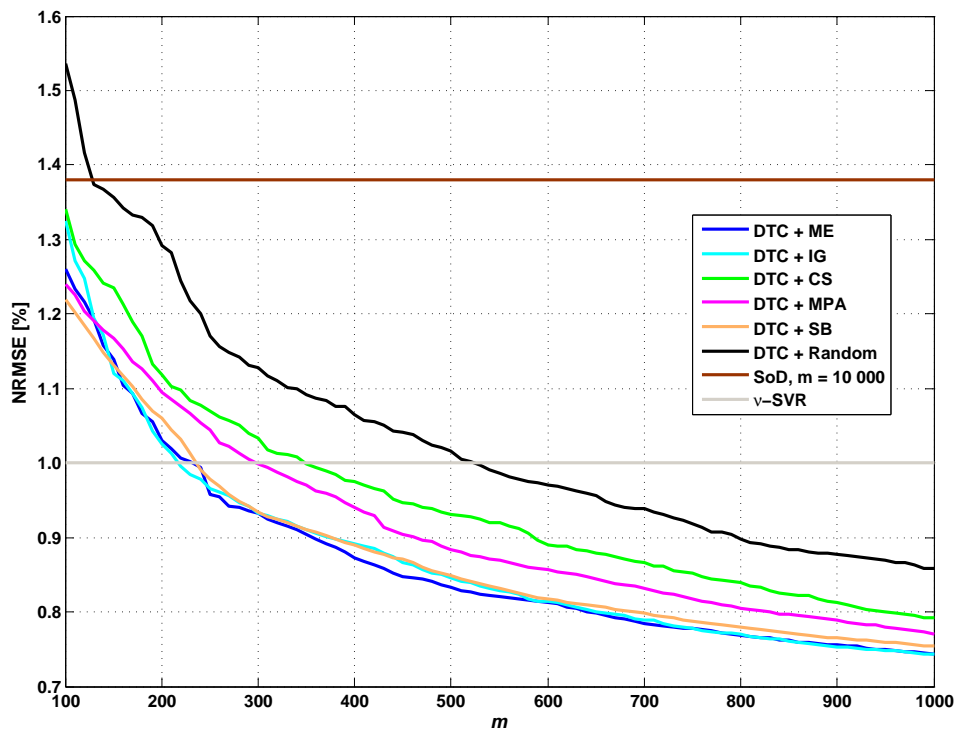


Figure 3.16: The NRMSE convergence trends are shown with respect to the EPS training data set with more than 400 000 points for the different greedy selection criteria of the DTC approximation depending on the number of active training points  $m$ .

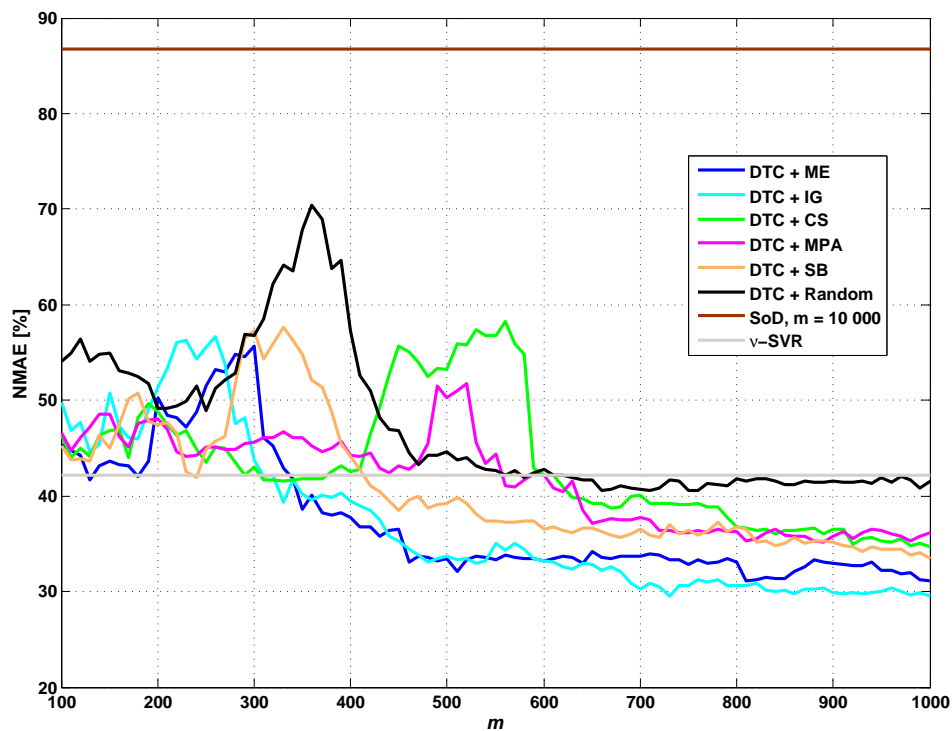


Figure 3.17: The NMAE convergence trends are presented with respect to the EPS training data set for the various greedy selection criteria of the DTC approximation depending on the number of active training points  $m$ .

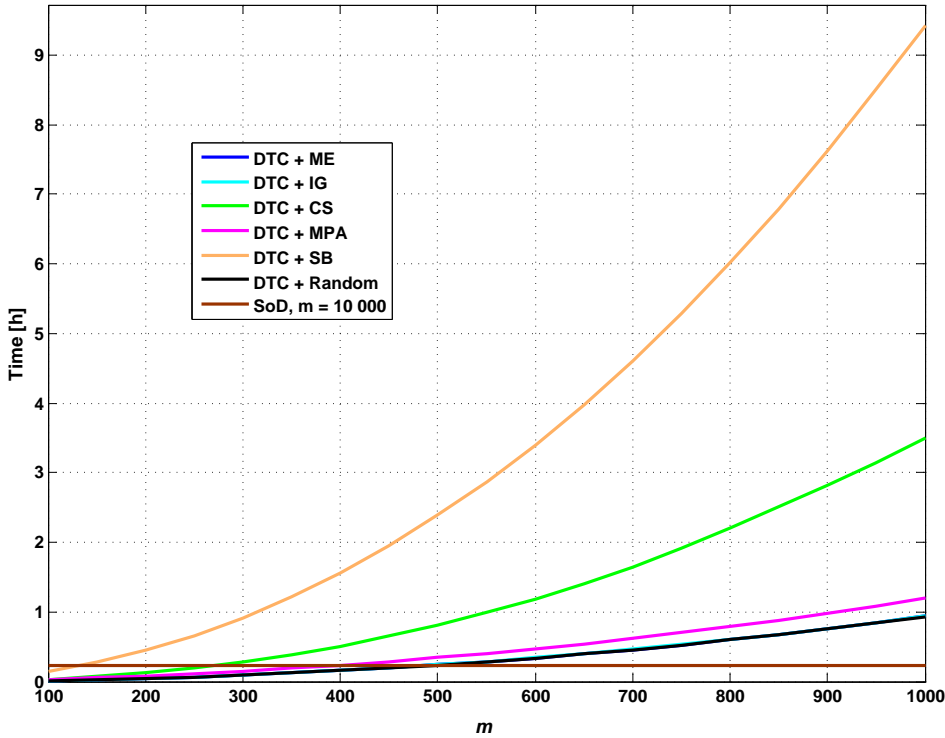


Figure 3.18: Training time in hours for each selection criteria of the DTC approximation and the SoD approach depending on the number of active training points  $m$ . The  $\nu$ -SVR algorithm is not considered in this plot, since the high training time of about 86 hours does not fit into the illustration. Note that the curves of our ME approach, the IG criterion, and the randomized selection nearly correspond to each other in this figure.

training data set, while our maximum error (ME) approach performs slightly better. For small active set sizes  $m$ , the IG criterion requires slightly more active training points before it gets close to the criterion of Smola and Bartlett (2001). These criteria are followed by the matching pursuit approach (MPA) criterion, the heuristic by Csató (2002), the randomized selection, and at last the SoD approach. Despite a ten times higher subset size, the SoD technique does not lead to nearly equivalent results as the sparse GPR approximations. Furthermore, all criteria have a monotonically decreasing convergence trend with respect to the NRMSE. The  $\nu$ -SVR algorithm results in average performance compared to the other methods, where over 5 000 support vectors are needed to reach this outcome. On the training data of sample size larger than 400 000 training points, the convergence trends with respect to the NMAE, cf. Figure 3.17, exhibit strong oscillations for almost all selection heuristics, which may be caused by the complex dynamics of the EPS system. For larger set sizes  $m$ , again our ME approach and the information gain approach by Seeger et al. (2003) perform best, despite showing some problems for small  $m$ . The other greedy selection criteria converge with nearly the same NMAE when the active set grows, but they achieved a 5% higher NMAE than the best one. The randomized selection has a very weak convergence regarding the maximum absolute error. Here, the NMAE is roughly 42%, analogously to the  $\nu$ -SVR, which merely improves for more than 600 active training points. The SoD approach with randomized selection is not suitable for modeling such large and complex regression problems, at least when considering the NMAE. For both error measures, i.e. the NRMSE and the NMAE, the DTC



insertion schemes which do not depend on the unfavorable sub-sampling for criteria evaluations lead to almost better results as the randomized approaches. In Figure 3.18, the training time of the different criteria for the creation of the regression models is visualized depending on the number of active training points  $m$ . Thereby, the hyperparameters are fixed such that the plot illustrates the pure insertion times. The difference in time between the maximum error, the information gain, and the randomized selection is negligible and the three graphs actually coincide in the figure. The selection heuristic by Smola and Bartlett (2001) consumes by far the most time for complete model training. The closely related criterion by Keerthi and Chu (2006) yields much better performance in terms of training time but requires additional memory due to the additionally calculated covariance vectors for criteria evaluation. The computation time needed by Csató’s selection method proves to be unsatisfactory, although it is much faster than the heuristic of Smola and Bartlett.

In Table 3.2 the training and prediction errors are summarized for both test sets on different road conditions, namely dry and wet road, and for the considered DTC selection methods, the SoD approach, and the  $\nu$ -SVR algorithm. The reduced GPR model on a randomly chosen subset of data of size  $m = 10\,000$  is considered as a benchmark. All errors and training times for the SoD method are averaged over ten runs. Finally, the  $\nu$ -SVR algorithm by Chang and Lin (2011) is evaluated on the same training

Method	$m$	Training (dry)		Prediction				Time [h]
		NRMSE [%]	NMAE [%]	NRMSE [%]		NMAE [%]		
				dry	wet	dry	wet	
ME	500	0.83	33.41	0.43	0.40	16.42	11.72	0.240
	1 000	<b>0.74</b>	31.14	<b>0.42</b>	<b>0.39</b>	14.42	13.28	0.939
IG	500	0.85	33.64	<b>0.42</b>	0.40	14.30	10.71	0.241
	1 000	<b>0.74</b>	<b>29.49</b>	<b>0.42</b>	<b>0.39</b>	14.79	15.36	0.941
CS	500	0.93	53.23	0.45	0.42	13.79	11.27	0.812
	1 000	0.79	34.68	<b>0.42</b>	0.40	15.40	13.54	3.505
MPA	500	0.88	50.31	0.43	0.41	14.02	11.88	0.343
	1 000	0.77	36.13	0.43	0.40	14.43	16.38	1.195
SB	500	0.85	39.21	0.44	0.41	15.44	10.66	2.386
	1 000	0.75	33.45	0.43	<b>0.39</b>	14.42	14.86	9.424
Random	500	1.02	44.61	0.47	0.45	13.28	12.35	0.238
	1 000	0.86	41.57	0.47	0.43	18.61	16.40	0.936
SoD	10 000	1.38	86.69	0.51	0.44	20.49	19.25	0.226
$\nu$ -SVR	5 164	1.00	42.21	0.79	0.70	<b>12.70</b>	<b>9.96</b>	86.366

Table 3.2: Training and prediction errors on the EPS data sets. Best results are indicated in bold face.

Here, the first five methods refer to the employed selection strategies for the DTC approximation. Thereby, the sparse DTC approximations with the intelligent insertion strategies perform best in terms of the NRMSE. The  $\nu$ -SVR approach yields the smallest absolute errors but has also the highest computational cost.

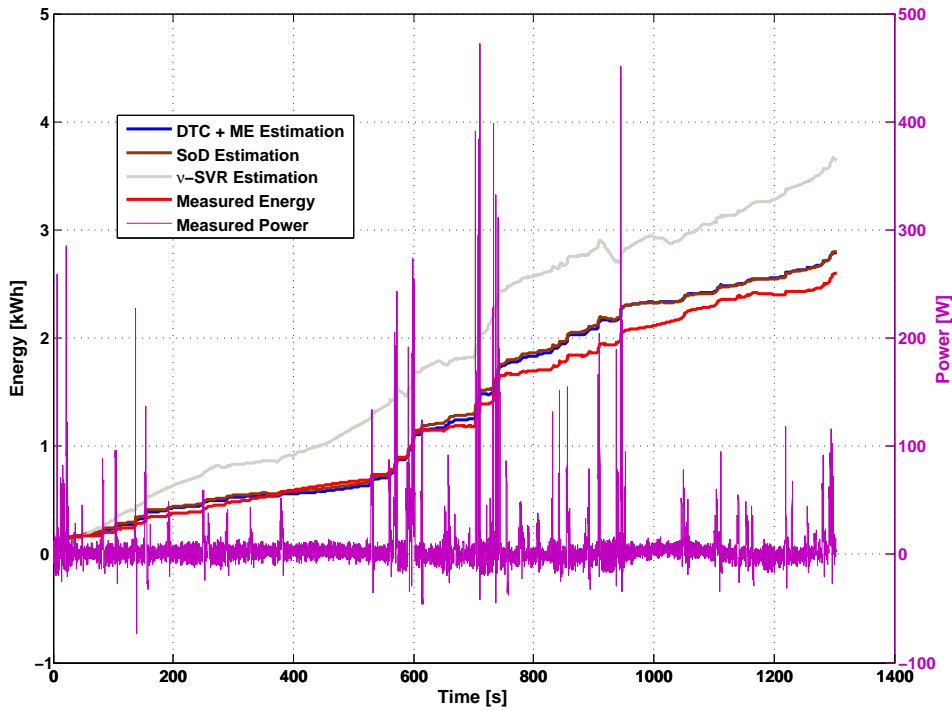


Figure 3.19: Evaluation of the DTC + ME technique, the SoD approach, and the  $\nu$ -SVR algorithm for estimating the energy consumption of the EPS assistance system with respect to the dry test data. The GP technique results in an accurate energy estimation, where the estimation of the  $\nu$ -SVR approach is to inexact and hence useless. The energy estimation on the wet test data with the considered methods yields the same results.

and test sets, where the resulting predictive model is induced by  $m = 5164$  support vectors. Note that training is always done on data from dry roads whereas testing is performed on data acquired under dry and wet conditions. Compared to the training data generated by extreme driving maneuvers, the test data from everyday drives is much easier to model for both, i.e. dry and wet conditions. This is indicated by the significantly lower prediction errors on the test data sets. In the last column of Table 3.2, the training time in hours is listed for each algorithm, where the time for optimizing the hyperparameters and the  $\nu$ -SVR parameters is excluded as in Figure 3.18. As shown in the table, our maximum error approach and the information gain criterion defined by Seeger et al. (2003) perform best, closely followed by the other intelligent selection schemes. The randomized insertion and the SoD approach on a randomly selected subset of all training points achieve poorer results on this large data set. These two methods exhibit also strong fluctuations with respect to the maximum absolute error even if the chosen subsets vary slightly. Regarding the NMAE, the  $\nu$ -SVR algorithm yields the best results but the overall mean prediction is less accurate compared to the Bayesian methods.

The goal of the considered data-based methods is to implement the online prediction of the energy demand for the EPS assistance system. In Figure 3.19, the energy estimation based on the power prediction by means of the DTC approximation with the ME insertion criterion, the SoD approach, and the  $\nu$ -SVR algorithm for the dry test data is illustrated. In this experiment, only the maximum error criterion for the DTC approximation is chosen since it is one of the best selection methods on this large data set and

yields approximately identical results regarding Seeger’s IG approach. The DTC + ME method and the SoD approach offer adequate results given by an estimated energy which lies almost in a 10% confidence bound according to the incurred energy consumption. The  $\nu$ -SVR technique performs badly for the energy estimation of the EPS system due to their large deviation and the unfavorable behavior resulting in strong positive variations shown by an increasing energy demand. Considering the prediction times, the DTC approaches enable fast and efficient calculations for new test cases which can be used for real-time simulations, i.e. with 100 Hz, during the development process to obtain an optimal electronic layout. The SoD approach offers nearly the same accuracy for energy estimation, but leads to the large subset size of  $m = 10\,000$  to a tenfold of prediction times and memory requirements. Due to the  $m = 5\,164$  support vectors for the  $\nu$ -SVR algorithm, the prediction effort results in five times higher computational cost compared to the sparse DTC approaches.

### 3.5 Discussion

This chapter is focused on sparse GP approximation techniques for non-parametric modeling in the regression setting. After an extensive review of the already existing GPR algorithms, the DTC approximation with different state-of-the-art insertion and deletion schemes is presented. At the core of this chapter, our novel maximum error (ME) approach is introduced to speed up the sparse DTC method. Moreover, an efficient way for automatic model selection under the DTC approximation by means of a generalized expectation maximization (EM) scheme is derived. The EM algorithm together with our ME selection criterion leads to a stable and fast technique for learning hyperparameters and active set selection at the same time. Especially the experiments on the SARCOS data sets verify the benefit of the proposed EM scheme with the quality of the obtained sparse GPR models. Furthermore, different greedy insertion and deletion criteria for sparse GPR or, more precisely, for the DTC approximation, were discussed and relationships between them are provided. Our novel selection criterion is based on the maximum error between model and training data which is inspired by an approximation of the computationally intensive but best performing method by Smola and Bartlett (2001), where justification for this approximation is provided. The primary advantage of our maximum error greedy selection is the combination of high accuracy with low computational cost for criterion calculation for all remaining points. Analogously to the criterion by Seeger et al. (2003), which is also efficient calculable for all remaining points, this advantage positively effects the resulting model quality. However, their information gain approach showed higher sensitivity to small active set sizes than the other intelligent insertion criteria, but is besides our strategy and the randomized selection the most efficient greedy method. In contrast, the insertion methods by Smola and Bartlett (2001), Quiñonero-Candela (2004), Keerthi and Chu (2006), or Titsias (2009) have to select a small random subset of remaining points for criterion evaluation. This random restriction can lead to poorer results, especially on harder regression tasks, for example, while modeling the energy consumption of the electronic power steering (EPS) assistance system. Surely, a higher subset size for criteria evaluations will increase modeling performance but also increase the computational effort dramatically. Furthermore, an additional matrix cache for storing specified columns of the full covariance matrix as suggested by Keerthi and Chu (2006) may help to speed up the criteria calculations. Neverthe-

less, this technique strongly increases the already high memory requirements. Apart from that, the only one-dimensional maximization in the derivation of the matching pursuit approach by Keerthi and Chu (2006) reflects in slightly higher errors than the baseline approach given by Smola and Bartlett (2001). Hence, the lower computing time makes it an attractive alternative for medium scale problems. The drawback of high storage requirements remains for both sparse GPR methods. In addition, it is also shown that the approximation technique of Quiñonero-Candela (2004) is nearly equivalent to the method by Smola and Bartlett, which was confirmed in the former evaluations. On the other hand, the variational framework by Titsias (2009) is closely related to the selection criterion by Quiñonero-Candela, where the only difference is an additional regularization term in the variational approach which induces even higher computational cost for criteria evaluations and model selection. The reproducing kernel Hilbert space (RKHS) based selection heuristic by Csató and Opper (2001) increases the complexity of the whole DTC approximation to  $\mathcal{O}(nm^3)$ , where the final results do not justify these computational efforts. Generally speaking, all intelligent insertion criteria provide reliable and more accurate results than a randomized selection scheme. For example, the randomized selection strategy shows strong fluctuations in the maximum absolute error, i.e. even small changes of the final active set can lead to higher generalization errors. Therefore, this strategy is less suitable and not recommendable for the considered large scale modeling problems. The criteria that operate on a randomly selected subset of remaining training points, e.g. the method by Smola and Bartlett (2001) or by Titsias (2009), perform significantly better but extensively increase the computational cost. In contrast, our ME selection strategy provides a desirable behavior, since it yields a favorable compromise between high model accuracy and short learning times. Analogously, for the removal of active points to increase predictive performance, our approach almost reaches the accuracy of Csató and Opper’s deletion method, outperforms the deletion criterion by Smola and Bartlett, and is still nearly as fast as the randomized removal. Note that the randomized deletion variant does not lead to reduced computational effort when evaluating new test instances while simultaneously keeping a comparable high model quality. In this case, our proposed removal strategy gives once more the best trade-off between computation time and accurate modeling excellence. Only considering our maximum error insertion criterion, the sparse DTC approximation consistently outperforms the standard GPR approach on a randomly selected subset of data (SoD) and results in more exact models than the FITC approximation or the  $\nu$ -SVR algorithm. In detail, the FITC approximation by Snelson and Ghahramani (2006a) performs well for small pseudo-input sets, since their optimization together with the hyperparameters works as expected. However, for higher set sizes, the optimization task gets more complex and difficult. Hence, mostly all DTC methods lead to higher prediction accuracy and lower learning times for increasing active set sizes. One reason for that behavior is the incremental approach for learning the induced hyperparameters using our generalized EM scheme. The  $\nu$ -SVR algorithm, which delivers small maximum absolute errors, leads to very hard and extremely costly optimization tasks for optimal parameter determination on large and complex data sets. The other regression methods, e.g. local GPs which are developed for online learning scenarios, are not competitive in such batch function approximation tasks. In fact, many experiments, like compliant and real-time robot control on a PR2 or fast simulation of the energy demand for the EPS assistance system, show that our proposed DTC + ME approach is competitive in learning performance while being fast in computation. Thus, our resulting predictive models are especially suitable for real-time applications in control and simulation tasks. Furthermore, it is necessary to gain as much experience as possible about the underlying systems and sample representative training data to improve the models obtained so far. For example, more investigations are

still needed to ensure the acquired model quality for the EPS system under unusual conditions such as snow and aquaplaning. Finally, the interested reader is referred to Chapter 5 for further conclusions and recommendations for future work.



## 4 Safe Active Learning with Gaussian Process Models

All machine learning methods depend on the available training data to extract knowledge or to provide generalized models of the observed environment. Hence, the data generation process is very important to obtain representative and valuable results. In contrast to passive learning strategies, which only observe the considered environment, active learning approaches act with the environment to improve the resulting data quality. In this way the active learning setting enables the learner to query for new data points which are mostly used for supervised machine learning tasks, namely regression and classification. Thus, active learning deals with the problem of selective and guided generation of labeled data, where the learner guides the data generation process by choosing new informative samples to be labeled based on the knowledge obtained so far. Providing labels for new data points, for example discrete image labels as by Lang and Baum (1992) or continuous measurements of a system output in case of technical environments, like by Hans et al. (2008), can be very costly and tedious. Hence, the overall goal of active learning is to create a representative data set without having to supply more data than necessary, and thus reducing the oracles annotation effort or the measurements on technical systems. A broad overview about various active learning scenarios is given by Settles (2010), where his main focus lies on classification problems. In the early statistical literature, the active learning concept is known as optimal experimental design, cf. Fedorov (1972). Generally, most active learning approaches are model-based, which means that during the sampling process a predictive model is trained on the so far obtained and labeled data, where the query criterion is based on that model. Besides that, most model-free approaches arise in the reinforcement learning context, where the query strategy coincides with the learned policy, cf. Geibel (2001). In this thesis, mainly model-based active learning schemes for regression tasks are considered. Naturally, the former presented GP models are employed as base for the following active learning algorithms. Moreover, the problem of safe exploration is introduced. Thereby, the active learner has additionally to distinguish between safe and unsafe queries when interacting with the environment. Safe exploration is especially important for data-sampling from technical and industrial systems, e.g. combustion engines and gas turbines, where critical and unsafe measurements need to be avoided. Hence, our goal is to actively select a budget of labeled points for learning a GP model from the system, while keeping the probability of measurement failures under given conditions to a minimum at the same time. Depending on the considered environment, our research is subdivided into stationary and transient, i.e. time-dependent, safe active learning scenarios. This helpful distinction allows appropriate exploring strategies for both slightly different cases with respect to the required conditions.

The remainder of this chapter, which summarizes the second part of my PhD studies, is organized as follows. Firstly, a brief overview about some existing work on safe exploration approaches is given. Subsequently, our safe active learning scheme for stationary environments is described, while details about our exploration strategy, the employed safety constraint, a theoretical analysis, and experiments

on toy and real applications are provided. This research is based on the published paper by Schreiter et al. (2015c). The ensuing section about our transient active learning setting has nearly the same structure, but the details differ considerably regarding the former work. Finally, a discussion of the proposed model-based active learning approaches in the last section concludes this chapter.

## 4.1 Related Work

Most existing work on safe exploration in unknown environments arose in the reinforcement learning setting. For example, the active sampling strategy by Moldovan and Abbeel (2012) for safe exploration in finite Markov decision processes (MDPs) relies on the restriction of suitable policies which ensure ergodicity at a user-defined safety level, which means that there exists a policy with high probability to get back to the initial state. An extended approach for risk aversion in MDPs is considered by Moldovan and Abbeel (2013). In the risk-sensitive reinforcement learning approach by Geibel (2001), the ergodic assumption for MDPs is dropped by introducing fatal absorbing states. The risk of a policy is thereby defined over the probability for ending in a fatal state. Therefore, the authors present a model-free reinforcement learning algorithm which is able to find near-optimal policies under bounded risk. The work by Gillula and Tomlin (2011) provides safety guarantees via a reachability analysis, when an autonomous robot explores its environment online. Therein, the robot is observed by an aerial vehicle which prevents the robot from taking unsafe actions in the state space. This hybrid control system assumes bounded actions and disturbances to ensure a safe behavior for all currently observable situations. Instead of bounding the action space, Polo and Rebollo (2011) introduce learning from demonstrations for dynamic control tasks. To safely explore the continuous state space in their reinforcement learning setting, a previously defined safe policy is iteratively adapted from the demonstration with small additive Gaussian noise. This approach ensures a baseline policy behavior which is used for safe exploration of high-risk tasks, namely hovering control of a helicopter. Galichet et al. (2013) consider a multi-armed risk-aware bandit setting to prevent hazards when exploring different tasks, e.g. energy management problems. They introduce a reward-based framework to limit the exploration of dangerous arms with a negative exploration bonus for risky actions. However, their approach is highly dependent on the designed reward function which has significant impact on the probability of damaging the system under consideration. Similarly, Hans et al. (2008) define a reward-based safety function to assess each state of the MDP and assume that there exists a safe return policy to leave critical states with non-fatal actions. Although, this assumption may not hold generally, it allows the usage of dynamic programming to solve an adapted Bellman optimality equation to get a return policy.

Strategies for exploring unknown environments without considering safety issues have also been reflected in the framework of global optimization with GPs. For example, Guestrin et al. (2005) proposed an efficient submodular exploration criterion for near-optimal sensor placements, i.e. they considered discrete input spaces. Zuluaga et al. (2013) presented a GP based approach for multi-objective optimization problems. A framework which yields a compromise between exploration and exploitation through confidence bounds is presented by Auer (2002). Srinivas et al. (2012) showed, that under reasonable assumptions



strong exploration guaranties can be given for Bayesian optimization with GPs. Due to the fact that the exploration tasks may lead to NP-hard problems, cf. Guestrin et al. (2005), the additional introduction of safety will increase the complexity which must be handled by the active learning scheme on a higher level. Furthermore, Sui et al. (2015) introduced an approach to reduce the problem complexity and to enable the derivation of theoretical guaranties. In their setting the safety issue and the explorable system output coincide in one function. That results in an exploration strategy where all inputs are defined as safe if their corresponding system output is larger than a certain threshold. Nevertheless, such a special setting, which is comparable to the reinforcement learning framework with safety encoded mostly in the reward function, is too restrictive for many practical applications.

Most of the above discussed approaches consider safe exploring or optimization strategies for stationary environments, which means that the more difficult transient task is only rarely mentioned in the literature. However, already Thrun (1995) distinguished between stationary and transient, i.e. from his point of view order-dependent, active learning scenarios due to their different requirements. In this thesis, his work is continued since these basic active learning characteristics and assumptions are important for further investigations.

## 4.2 Active Learning for Stationary Environments

The objective in this section is to learn a GPR model from a stationary technical system using a limited budget of labeled points while ensuring that critical regions of the considered systems are avoided during measurements. This GP for system modeling is further referred as exploratory GP, since it generates the base for the exploring strategy of the environment. Thus, active data selection problems are considered for systems with compact input spaces  $\mathbb{X} \subset \mathbb{R}^d$ . This constrained space  $\mathbb{X}$  can have regions, where sampling and measuring is undesirable and can damage the system. When considering technical systems, operation of the system in specific regions can result in exceeding the allowed physical limits such as temperatures and pressures. If the active learner chooses a sample in such a region, it is considered as failure. To anticipate a failure, it is assumed that the active learner observes some feedback from the system for each data point. This feedback indicates the health status of the system. In case of a combustion engine, the feedback can be given by the engine temperature. The safety of an active exploring algorithm is defined over the probability of failure, i.e. an exploration scheme is called safe at the level of  $1 - \delta$ , if the overall failure probability when querying new instances is lower than a certain threshold  $\delta$ . The user can chose  $\delta$  sufficiently small to achieve an acceptable risk of failure. However, reducing this probability of failure comes at the cost of decreased sample efficiency. Thus, more samples will be required, as the active learner will take smaller steps and explore more carefully. Throughout this chapter, the notations  $\mathbb{X}_+$  for safe and  $\mathbb{X}_-$  are used for unsafe regions of the confined input space  $\mathbb{X}$  to distinguish between safe and hazardous input areas of the underlying system. To realize that, a problem specific GP classifier is employed to identify safe and unsafe regions in  $\mathbb{X}$ . For the safe region  $\mathbb{X}_+$  of the input space, it is assumed that  $\mathbb{X}_+$  is compact and connected. The basic idea of this discriminative GP is to learn a decision boundary between the two classes induced by  $\mathbb{X}_+$  and  $\mathbb{X}_-$ , preferably without querying

a point in the unsafe region  $\mathbb{X}_-$ . Therefore, it is assumed that additional information provided by the environment is available when our exploration scheme is getting close to the decision boundary, cf. Hans et al. (2008). For real-world applications as in combustion engine measurement, the engine temperature is an indicator for the proximity to the decision boundary. Thus, it is possible to design a learnable system response, for example with a continuous discriminative function, such that the resulting active learner recognizes if it is getting close to unsafe input locations. Thereby, an exact learning of the unknown decision boundary  $\mathbb{X}_0$  is indented, since for the learner highly informative hazardous samples should be absolutely avoided. Figure 4.1 illustrates the described setting for an input space  $\mathbb{X} \subset \mathbb{R}^2$ . Finally both GPs, namely the exploratory and the discriminative, are incrementally learned when getting new sampled data during exploration of the input space which is explained in the next sections.

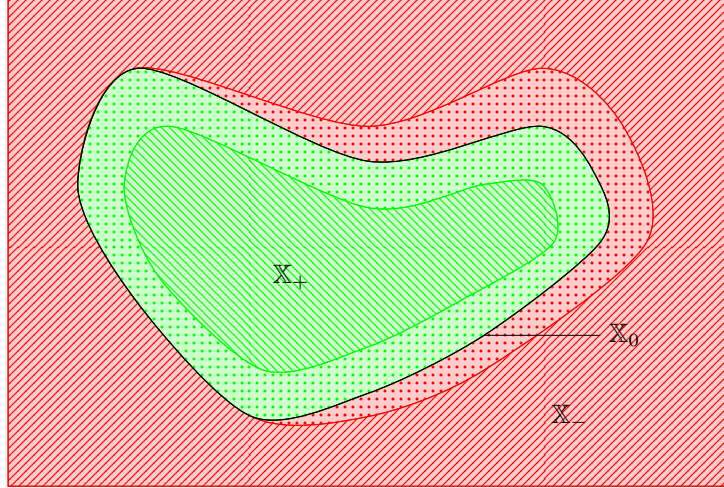


Figure 4.1: Partition of the input space  $\mathbb{X}$  into a safe explorable area  $\mathbb{X}_+$  and an unsafe region  $\mathbb{X}_-$ , separated by the unknown decision boundary  $\mathbb{X}_0$ . Over the dotted area, a discriminative function is learned for recognizing whether the exploration becomes risky.

### 4.2.1 Exploring Strategy

As by Atlas et al. (1990), our exploration strategy is a selective sampling approach based on the posterior entropy of the exploratory GP model for the noisy functional relationship (2.2) of the underlying technical system. The differential entropy criterion has been frequently used in the active learning literature, for example, see Seo et al. (2000) or Krause and Guestrin (2007). The main goal of this uncertainty sampling task, cf. Settles (2010), is to find an optimal set of feasible data points  $\mathbf{X}_{\text{Opt}} \in \mathbb{R}^{n \times d}$  of given size  $n$  and for determined hyperparameters  $\boldsymbol{\theta}$  such that the differential entropy (A.23) of the Gaussian model evidence (2.6) is maximal. Thus, it follows

$$\mathbf{X}_{\text{Opt}} = \arg \max_{\mathbf{X} \subset \mathbb{X}, |\mathbf{X}|=n} (\mathbb{H}[\mathbf{y} | \mathbf{X}, \boldsymbol{\theta}]) . \quad (4.1)$$

According to Equation (A.24), the differential entropy  $\mathbb{H}[\mathbf{y} | \mathbf{X}, \boldsymbol{\theta}]$  of the standard GPR marginal likelihood depends on the determinant of the associated covariance matrix which results in

$$\mathbb{H}[\mathbf{y} | \mathbf{X}, \boldsymbol{\theta}] = \frac{1}{2} \log \left( |2\pi e(\sigma^2 \mathbf{I} + \mathbf{K})| \right) . \quad (4.2)$$

Note that Ko et al. (1995) showed that the problem of finding a finite set  $\mathbf{X}_{\text{Opt}}$  is NP-hard. Nevertheless, the following lemma summarizes some meaningful results about the differential entropy in our GP setting.

**Lemma 4.1.** *Let  $\mathcal{D} = (\mathbf{y}, \mathbf{X})$  be a finite, non-empty data set as employed in Section 2.1.1 and the variance  $\sigma^2 \geq (2\pi e)^{-1}$ . Then, for the stationary covariance functions considered in this thesis and given by the Equations (2.26), (2.27), (2.28), and (2.29) with magnitude  $\sigma_f^2$ , the differential entropy (4.2) of the GP evidence (2.6) is a non-negative, monotonically increasing, and submodular function.*

The detailed proof of the latter lemma uses known results from Nemhauser et al. (1978) and Cover and Thomas (2006) and is given in the Appendix A.3.5. In Lemma 4.1 the lower bound for the noise variance  $\sigma^2$  of the regression model (2.2) can be easily achieved by additionally scaling the target values  $\mathbf{y}$  and the magnitude  $\sigma_f^2$  with some suitable constant, cf. the proof of Lemma 4.1. The properties of the entropy induced by the above lemma guarantee that the greedy selection scheme

$$\mathbf{x}_{n+1} = \arg \max_{\mathbf{x}_* \in \mathbb{X}} (\mathbb{H}[y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}]) \quad (4.3)$$

for our active learning strategy yields a nearly optimal subset of input points, where the predictive distribution of  $y_*$  results from Equation (2.9). More precisely, as shown by Nemhauser et al. (1978), it holds true that the greedy selection scheme (4.3) results in a model entropy which is greater than 63% of the optimal entropy value induced by  $\mathbf{X}_{\text{Opt}}$ . Formally, this statement gives

$$\mathbb{H}[\mathbf{y} | \mathbf{X}_{\text{Greedy}}, \boldsymbol{\theta}] \geq \left(1 - \frac{1}{e}\right) \mathbb{H}[\mathbf{y} | \mathbf{X}_{\text{Opt}}, \boldsymbol{\theta}]$$

for the two sets  $\mathbf{X}_{\text{Greedy}}$  and  $\mathbf{X}_{\text{Opt}}$  of the same size  $n \in \mathbb{N}$ . This guarantee induces an efficient greedy algorithm and, with some foresight to the introduction of safety in Section 4.2.3, points out that it will generally not be possible to design an optimal and fast safe active learning algorithm, cf. Moldovan and Abbeel (2012). Since the entropy from the greedy selection rule (4.3) is a monotonic function in the posterior variance (2.9), the above query strategy reduces to

$$\begin{aligned} \mathbf{x}_{n+1} &= \arg \max_{\mathbf{x}_* \in \mathbb{X}} (\mathbb{H}[y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}]) \\ &= \arg \max_{\mathbf{x}_* \in \mathbb{X}} \left( \frac{1}{2} \log (2\pi e \text{Var}[y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}]) \right) \\ &= \arg \max_{\mathbf{x}_* \in \mathbb{X}} (\text{Var}[y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}]) \\ &= \arg \max_{\mathbf{x}_* \in \mathbb{X}} \left( k_{**} - \|\mathbf{L}^{-1} \mathbf{k}_*\|^2 \right), \end{aligned} \quad (4.4)$$

and thus searches for points where our regression model is maximally uncertain, cf. Settles (2010). This search problem is solved with second-order optimization techniques, where the necessary gradient and Hessian of the posterior variance is given in the Appendix A.3.4. Moreover, the previously presented Lemma 2.1 in Section 2.1.5 provides an informative connection between the GP model entropy and marginal likelihood. Hence, for known and fixed hyperparameters  $\boldsymbol{\theta}_{\text{Opt}}$ , maximum entropy sampling is equivalent to marginal likelihood minimization, which means that the labeled data should be sampled such that the GP model probability is as small as possible. As empirically shown in Ramakrishnan et al. (2005), the described entropy-based sampling strategy conditioned on the requirements for Lemma 4.1 tends to select input locations close to the border and induces a point set  $\mathbf{X}$  which is nearly uniformly distributed in the confined input space  $\mathbb{X}$ . This is a favorable behavior for modeling with GPs, which additionally enables us to slightly extrapolate the borders of the technical system with our GPR model.

To visualize the efficiency of the former introduced active learning strategy with respect to the presented entropy criterion during the data-sampling process, theoretical results by Cover and Thomas (2006) and the bounds

$$\frac{n}{2} \log(2\pi e \sigma^2) \leq H[\mathbf{y} | \mathbf{X}, \boldsymbol{\theta}] \leq \frac{n}{2} \log(2\pi e (\sigma_f^2 + \sigma^2)) \quad (4.5)$$

derived in the proof of Lemma 4.1 are used to normalize the gain in the differential entropy (4.2). Then, after sampling the  $n$ -th query point, the normalized entropy ratio (NER) as defined by

$$\text{NER} = \frac{\frac{2}{n} H[\mathbf{y} | \mathbf{X}, \boldsymbol{\theta}] - \log(2\pi e \sigma^2)}{\log\left(1 + \frac{\sigma_f^2}{\sigma^2}\right)} \quad (4.6)$$

is employed to compare different active sampling settings. If the ratio (4.6) is close to one when the input point set  $\mathbf{X}$  enlarges during the exploration process, nearly the maximal possible entropy is gained for the selected queries. Otherwise, if  $\text{NER} \approx 0$ , the current query  $\mathbf{x}_n$  does not explore the input space very much. The denominator of the NER in Equation (4.6) is slightly related to the signal-to-noise ratio (SNR), where the whole NER is comparable to a standardized version of the entropy rate as defined in Cover and Thomas (2006).

## 4.2.2 Exploring under Uncertainty

One of the most important tasks in active learning scenarios is the determination of hyperparameters, here with respect to our GPR model. The easiest and most common way is to fix the hyperparameters a priori and keep them constant during the learning process, for example, as in Krause and Guestrin (2007) or the previous section. Another reason for keeping the underlying model assumptions fixed is the analytical tractability for the derivation of theoretical results, see Hanneke (2007) and the references therein. Hence, for a different set of hyperparameters, the active learner will not necessarily lead to the same collection of data points. But for the investigation of many technical systems, the GPR hyperparameters, especially for the covariance function, are not explicitly given. Also the former selection of data points from the system, e.g. with a quasi-random design of experiments (DoE) given through a Sobol sequence, cf. Sobol (1976), to estimate the hyperparameters a priori and subsequently active learning of the system dynamics will increase the effort. To handle this problem, a mixture of GP priors is used, as described by Sung (2004), with respect to the unknown hyperparameters for data-based system modeling. Then, the Gaussian prior distribution is discretized in a range of acceptable hyperparameters. That is possible even if the exact hyperparameters are unknown, but their range can be estimated with only a little additional knowledge about the system. This strategy is comparable to the query by committee approach of Burbidge et al. (2007). Finally, the Gaussian mixture prior density is defined by

$$q(\mathbf{f} | \mathbf{X}, \boldsymbol{\theta}) = \sum_{j=1}^m \omega_j p(\mathbf{f} | \mathbf{X}, \boldsymbol{\theta}_j) \quad (4.7)$$

with non-negative mixing probabilities  $\omega_j$  fulfilling  $\sum_{j=1}^m \omega_j = 1$ , a centered GP prior distribution equivalent to Equation (2.3), and all hyperparameters summarized in  $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_m)$ . Analogously to

Equation (2.5), the mixture GP posterior density results in

$$\begin{aligned}
q(\mathbf{f} | \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}) &= \frac{p(\mathbf{y} | \mathbf{f}, \mathbf{X}) q(\mathbf{f} | \mathbf{X}, \boldsymbol{\theta})}{q(\mathbf{y} | \mathbf{X}, \boldsymbol{\theta})} \\
&\propto p(\mathbf{y} | \mathbf{f}, \mathbf{X}) \sum_{j=1}^m \omega_j p(\mathbf{f} | \mathbf{X}, \boldsymbol{\theta}_j) \\
&= \sum_{j=1}^m \omega_j p(\mathbf{y} | \mathbf{f}, \mathbf{X}) p(\mathbf{f} | \mathbf{X}, \boldsymbol{\theta}_j) \\
&\propto \sum_{j=1}^m \omega_j p(\mathbf{f} | \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}_j)
\end{aligned} \tag{4.8}$$

with Bayesian inference and the GPR model likelihood from Equation (2.4). The mixed model evidence is given by

$$\begin{aligned}
q(\mathbf{y} | \mathbf{X}, \boldsymbol{\theta}) &= \int_{\mathbb{R}^n} p(\mathbf{y} | \mathbf{f}, \mathbf{X}) q(\mathbf{f} | \mathbf{X}, \boldsymbol{\theta}) \partial \mathbf{f} \\
&= \int_{\mathbb{R}^n} \sum_{j=1}^m \omega_j p(\mathbf{y} | \mathbf{f}, \mathbf{X}) p(\mathbf{f} | \mathbf{X}, \boldsymbol{\theta}_j) \partial \mathbf{f} \\
&= \sum_{j=1}^m \omega_j \int_{\mathbb{R}^n} p(\mathbf{y} | \mathbf{f}, \mathbf{X}) p(\mathbf{f} | \mathbf{X}, \boldsymbol{\theta}_j) \partial \mathbf{f} \\
&= \sum_{j=1}^m \omega_j p(\mathbf{y} | \mathbf{X}, \boldsymbol{\theta}_j) ,
\end{aligned} \tag{4.9}$$

which is expressed through the convex linear combination of the individual marginal likelihoods (2.6) for every set of hyperparameters  $\boldsymbol{\theta}_j$ . Finally, the mixed predictive density results in

$$\begin{aligned}
q(y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}) &= \sum_{j=1}^m \omega_j \int_{\mathbb{R}^n} p(y_* | \mathbf{x}_*, \mathbf{f}, \mathbf{X}, \boldsymbol{\theta}_j) p(\mathbf{f} | \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}_j) \partial \mathbf{f} \\
&= \sum_{j=1}^m \omega_j p(y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}_j)
\end{aligned} \tag{4.10}$$

analogously to the standard GPR version in Equation (2.7). Here, the expectation value of the mixed predictive GPR model results in the convex linear combination

$$\mathbb{E}_q[y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}] = \sum_{j=1}^m \omega_j \mathbb{E}_p[y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}_j]$$

of the mean values from the single GP components. More interestingly for our exploration framework is the mixed predictive variance

$$\begin{aligned}
\text{Var}_q[y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}] &= \sum_{j=1}^m \omega_j \left( \left( \mathbb{E}_p[y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}_j] \right)^2 + \text{Var}_p[y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}_j] \right) \\
&\quad - \left( \mathbb{E}_q[y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}] \right)^2 ,
\end{aligned} \tag{4.11}$$

see Sung (2004) for more details, since the former query strategy (4.4) yields in this case

$$\mathbf{x}_{n+1} = \arg \max_{\mathbf{x}_* \in \mathbb{X}} \left( \text{Var}_q[y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}] \right) , \tag{4.12}$$

where the mixing probabilities  $\omega_j$  are defined by the marginal likelihood ratio

$$\omega_j = \frac{p(\mathbf{y} | \mathbf{X}, \boldsymbol{\theta}_j)}{\sum_{l=1}^m p(\mathbf{y} | \mathbf{X}, \boldsymbol{\theta}_l)} \quad (4.13)$$

following from the single components, and thus are independent of a new query instance  $\mathbf{x}_* \in \mathbb{R}^d$ . Analogously to the previous query strategy (4.4), the optimization problem (4.12) for selecting high informative data points is solved with a Newton method. Therefore, the gradient and the Hessian of the query criterion need to be determined. For the calculation of the gradient for the mixed predictive variance (4.11) follows

$$\begin{aligned} & \frac{\partial \text{Var}_q [y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}]}{\partial \mathbf{x}_*} \\ &= 2 \sum_{j=1}^m \omega_j \text{E}_p [y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}_j] \frac{\partial \text{E}_p [y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}_j]}{\partial \mathbf{x}_*} + \sum_{j=1}^m \omega_j \frac{\partial \text{Var}_p [y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}_j]}{\partial \mathbf{x}_*} \\ & \quad - 2 \text{E}_q [y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}] \sum_{j=1}^m \omega_j \frac{\partial \text{E}_p [y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}_j]}{\partial \mathbf{x}_*} \\ &= 2 \sum_{j=1}^m \omega_j (\text{E}_p [y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}_j] - \text{E}_q [y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}]) \frac{\partial \text{E}_p [y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}_j]}{\partial \mathbf{x}_*} \\ & \quad + \sum_{j=1}^m \omega_j \frac{\partial \text{Var}_p [y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}_j]}{\partial \mathbf{x}_*}, \end{aligned} \quad (4.14)$$

where the gradients of mean and variance for the standard GP prediction from Equation (2.9) are given in Section A.3.4 of the appendix. Additionally, the Hessian of the predictive variance for the mixture of GP priors results in

$$\begin{aligned} & \frac{\partial^2 \text{Var}_q [y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}]}{\partial \mathbf{x}_* \partial \mathbf{x}_*^T} \\ &= 2 \sum_{j=1}^m \omega_j \left( \frac{\partial \text{E}_p [y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}_j]}{\partial \mathbf{x}_*} \right)^T \frac{\partial \text{E}_p [y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}_j]}{\partial \mathbf{x}_*} \\ & \quad - 2 \left( \sum_{j=1}^m \omega_j \frac{\partial \text{E}_p [y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}_j]}{\partial \mathbf{x}_*} \right)^T \sum_{j=1}^m \omega_j \frac{\partial \text{E}_p [y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}_j]}{\partial \mathbf{x}_*} \\ & \quad + 2 \sum_{j=1}^m \omega_j (\text{E}_p [y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}_j] - \text{E}_q [y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}]) \frac{\partial^2 \text{E}_p [y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}_j]}{\partial \mathbf{x}_* \partial \mathbf{x}_*^T} \\ & \quad + \sum_{j=1}^m \omega_j \frac{\partial^2 \text{Var}_p [y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}_j]}{\partial \mathbf{x}_* \partial \mathbf{x}_*^T}. \end{aligned} \quad (4.15)$$

Compared to the sampling scheme (4.4), the consideration of uncertainty in the hyperparameters increases the computational effort for query optimization. Dependent on the number of different hyperparameter sets  $m$ , which induce the mixture of GP priors, the discretization of the hyperparameters should be as small as possible to keep efficiency. Here, expert knowledge about the explorable system can be included in the active learning approach. A bound for the discretization distance, as presented by Krause and Guestrin (2007), is also helpful, although it depends on the number of queries which would be labeled. Nevertheless, the increase in effort is  $m$  times higher than that of the active explorer based on only one GP with known and fixed hyperparameters.

### 4.2.3 Introducing Safety

To distinguish between safe and unsafe regions in our active learning framework, an extended GP classification approach as presented in Section 2.1.2 is used. Hence, the latent discriminative function  $g : \mathbb{X} \rightarrow \mathbb{R}$  is learned, and subsequently mapped to the unit interval to describe the class probability for each input point. Generally, a positive label means safe and a negative one unsafe, respectively. Our approach to introduce safety is based on additional system information to learn the discriminative function  $g$ , especially when getting close to the decision boundary. In practice, without any feedback from the physical system or some user-defined knowledge, it is not possible to explore the environment safely, see Valiant (1984) for further details. This additional feedback is encoded in a possibly noisy function  $h : \mathbb{X} \rightarrow (-1, +1)$  to train the discriminative function  $g$  around the decision boundary, preferable only by sampling in  $\mathbb{X}_+$ . To define a model likelihood for this heterogeneous data scenario, it is assumed that sampling from the system leads to values  $h_j = h(\mathbf{x}_j)$  or labels  $c_i = c(\mathbf{x}_i)$  which end up in consistent data. The results are, depending on the location of the data point, cf. Figure 4.1, either labels or discriminative function values. For  $\mathbf{c} \in \{-1, +1\}^k$ ,  $\mathbf{h} \in \mathbb{R}^l$ , and  $\mathbf{g} \in \mathbb{R}^n$  with  $k + l = n$  the model likelihood is defined by

$$p(\mathbf{c}, \mathbf{h} | \mathbf{g}, \mathbf{X}) = \prod_{i=1}^k \Phi(c_i g_i) \prod_{j=1}^l \mathcal{N}(h_j | g_j, \eta^2), \quad (4.16)$$

where the GPC model likelihood from Equation (2.12) is combined with a Gaussian regression model for  $\mathbf{h}$  and using a noise variance  $\eta^2$  equivalently to Equation (2.4). Employing the Gaussian prior (2.3) for all  $g_i$  with  $i = 1, \dots, n$  and the Laplace approximation as introduced in Section 2.1.2, a Gaussian posterior approximation  $q(\mathbf{g} | \mathbf{c}, \mathbf{h}, \mathbf{X})$  is obtained with respect to the exact posterior

$$p(\mathbf{g} | \mathbf{c}, \mathbf{h}, \mathbf{X}) \approx q(\mathbf{g} | \mathbf{c}, \mathbf{h}, \mathbf{X}) = \mathcal{N}(\mathbf{g} | \boldsymbol{\mu}, \boldsymbol{\Sigma}), \quad (4.17)$$

where

$$\boldsymbol{\mu} = \arg \max_{\mathbf{g}} \left( \log(p(\mathbf{g} | \mathbf{c}, \mathbf{h}, \mathbf{X})) \right) \in \mathbb{R}^n \quad (4.18)$$

and

$$\boldsymbol{\Sigma} = (\mathbf{W} + \mathbf{K}^{-1})^{-1} \in \mathbb{R}^{n \times n}. \quad (4.19)$$

Analogously to Equation (2.18), the diagonal matrix  $\mathbf{W} \in \mathbb{R}^{n \times n}$  is expressed as

$$\begin{aligned} \mathbf{W} &= - \frac{\partial^2 \log(p(\mathbf{g} | \mathbf{c}, \mathbf{h}, \mathbf{X}))}{\partial \mathbf{g} \partial \mathbf{g}^T} \\ &= \text{diag} \left( \left( \bigoplus_{i=1}^k \left( \frac{\mathcal{N}(c_i g_i)^2}{\Phi(c_i g_i)^2} + \frac{c_i g_i \mathcal{N}(c_i g_i)}{\Phi(c_i g_i)} \right) \right) \oplus \left( \bigoplus_{j=1}^l \left( \frac{1}{\eta^2} \right) \right) \right). \end{aligned} \quad (4.20)$$

As in the standard GPC scheme, Newton iterations as explained in Equation (2.19) are carried out to calculate the approximated posterior moments  $\boldsymbol{\mu}$  and  $\boldsymbol{\Sigma}$  as defined in Equation (4.18) and (4.19), respectively. In the same manner as in Equation (2.20) derived, the predictive distribution of test cases  $\mathbf{x}_*$  results in

$$q(g_* | \mathbf{x}_*, \mathbf{c}, \mathbf{h}, \mathbf{X}) = \mathcal{N} \left( g_* \mid m_* + \mathbf{k}_*^T \boldsymbol{\alpha}, k_{**} - \|\mathbf{L}^{-1} \mathbf{W}^{\frac{1}{2}} \mathbf{k}_*\|^2 \right) \quad (4.21)$$

with the discriminative prediction vector  $\boldsymbol{\alpha} = \mathbf{K}^{-1}(\boldsymbol{\mu} - \mathbf{m}) \in \mathbb{R}^n$ . Note that the Cholesky factorization  $\mathbf{L}\mathbf{L}^T = \mathbf{I} + \mathbf{W}^{\frac{1}{2}}\mathbf{K}\mathbf{W}^{\frac{1}{2}} \in \mathbb{R}^{n \times n}$  as described in Equation (A.1) of the appendix is employed. Now, the following class probability

$$\begin{aligned} \Pr_{\mathbf{q}}[g_* \geq 0 \mid \mathbf{x}_*, \mathbf{c}, \mathbf{h}, \mathbf{X}] &= \int_0^{\infty} \mathbf{q}(g_* \mid \mathbf{x}_*, \mathbf{c}, \mathbf{h}, \mathbf{X}) \partial g_* \\ &= 1 - \Phi\left(\frac{0 - \mathbb{E}_{\mathbf{q}}[g_* \mid \mathbf{x}_*, \mathbf{c}, \mathbf{h}, \mathbf{X}]}{\sqrt{\text{Var}_{\mathbf{q}}[g_* \mid \mathbf{x}_*, \mathbf{c}, \mathbf{h}, \mathbf{X}]}}\right) \\ &= \Phi\left(\frac{m_* + \mathbf{k}_*^T \boldsymbol{\alpha}}{\sqrt{k_{**} - \|\mathbf{L}^{-1}\mathbf{W}^{\frac{1}{2}}\mathbf{k}_*\|^2}}\right), \end{aligned} \quad (4.22)$$

for a new test point is used in our safe active learning framework as indicator when exploration gets dangerous. Thus, the safety constraint given by

$$\Pr_{\mathbf{q}}[g_* \geq 0 \mid \mathbf{x}_*, \mathbf{c}, \mathbf{h}, \mathbf{X}] \geq p \quad (4.23)$$

is included in the query strategy (4.4) or (4.12), respectively. The safety constraint in our active learning scenario should ensure that the probability of making a failure is small, e.g. less than  $1 - p$  for some  $p \in (0, 1)$  in each step. For a successful safe exploration, the function  $h$  must fulfill some conditions, which are discussed in Section 4.2.5. Moreover, if the adaption of the hyperparameters for this special classification approach should be considered with MLM, the approximated marginal likelihood and the gradient with respect to the hyperparameters is given in Section A.3.4 of the appendix.

Another more consistent idea for modeling the safety issue under the same conditions is to set the model likelihood to zero for a negative  $g_i$  with its associated positive class label  $c_i$  and vice versa for the complementary case. This can be realized with a model likelihood defined through truncated Gaussian densities (A.27) which results in

$$p(\mathbf{h} \mid \mathbf{g}, \mathbf{c}, \mathbf{X}) = \prod_{j=1}^l \mathcal{N}(h_j \mid g_j, \eta^2) \prod_{\forall c_i = -1} \mathcal{U}(h_i \mid g_i, \eta^2, 0) \prod_{\forall c_i = +1} \mathcal{L}(h_i \mid g_i, \eta^2, 0), \quad (4.24)$$

such that a system response  $h_i$  for each query point  $\mathbf{x}_i$  is obtained. Hereby, the first term in Equation (4.24) is equal to the term in (4.16) for modeling the discriminative function around the decision boundary. The other products of upper and lower truncated Gaussian densities, cf. Horrace (2005), model the behavior of  $g$  for very safe and unsafe queries indicated through the associated label  $c$ . Note that the factor in the truncated terms will vanish if the value of  $g_i$  is smaller than  $h_i$ , i.e. discriminative function values  $g_i$  smaller than the noisy system response  $h_i$  are not allowed for very safe measurable points  $\mathbf{x}_i$ , and vice versa. This idea leads to a stronger penalization of misclassified points compared to the former presented approach, since the logarithmic posterior will tend to minus infinity in such a case. Nevertheless, this model requires approximate inference techniques to estimate a Gaussian posterior distribution. A Laplace approximation does not seem to be useful due to the discontinuity of the model likelihood (4.24). More favorable would be an expectation propagation (EP) approach, cf. Minka (2001), to induce a Gaussian posterior approximation, or perhaps directly a truncated Gaussian approximation. This approach is still a topic of future research due to the higher cost and implementation effort of the EP algorithm compared to the Laplace approximation.



## 4.2.4 The Algorithm

In this section, our entropy-based active learning framework extended by a constraint inducing safety is presented. As already seen, searching for points with maximal entropy leads to maximizing their predictive variance, see (4.4) and (4.12). Furthermore, it is possible to simplify the safety constraint (4.23) by defining the confidence parameter  $\nu = \Phi^{-1}(p) \in \mathbb{R}$ , since

$$\Pr_{\mathbf{q}} [g_* \geq 0 \mid \mathbf{x}_*, \mathbf{c}, \mathbf{h}, \mathbf{X}, \boldsymbol{\theta}_g] = \Phi\left(\frac{\mu_{g_*}}{\sigma_{g_*}}\right) \geq p$$

results in the lower confidence bound

$$\mu_{g_*} - \nu \sigma_{g_*} \geq 0 \tag{4.25}$$

with the short hand notations  $\mu_{g_*} = \mathbb{E}_{\mathbf{q}} [g_* \mid \mathbf{x}_*, \mathbf{c}, \mathbf{h}, \mathbf{X}, \boldsymbol{\theta}_g]$  and  $\sigma_{g_*}^2 = \text{Var}_{\mathbf{q}} [g_* \mid \mathbf{x}_*, \mathbf{c}, \mathbf{h}, \mathbf{X}, \boldsymbol{\theta}_g]$  following from Equation (4.22). The opposite of the safety constraint (4.25), namely the upper confidence bound (UCB), is often used as query strategy in Bayesian optimization tasks, e.g. as in Kaufmann et al. (2012). In contrast, our safety constraint is employed to restrict the uncertainty-based exploring scheme to non-dangerous input regions. Thus, the following optimization problem

$$\begin{aligned} \mathbf{x}_{i+1} &= \arg \max_{\mathbf{x}_* \in \mathbb{X}} (\text{Var} [y_* \mid \mathbf{x}_*, \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}]) \\ \text{s.t.} &: \quad \mu_{g_*} - \nu \sigma_{g_*} \geq 0 \\ & \quad \mathbf{l} \preceq \mathbf{x}_* \preceq \mathbf{u} \end{aligned} \tag{4.26}$$

is obtained which defines our GP based safe active learning strategy. For this scheme, which is summarized in Algorithm 4.1, for both GPs, i.e. the exploratory and the discriminative, is assumed that the hyperparameters  $\boldsymbol{\theta}$  and  $\boldsymbol{\theta}_g$ , respectively, are previously given. Therefore, recall from Krause and Guestrin (2007) that determining the hyperparameters in advance is similar to defining a grid over the whole input space  $\mathbb{X} \in \mathbb{R}^d$ . Note that  $\boldsymbol{\theta}$  contains all fixed hyperparameters of the exploration model, also if the exploration under uncertainty from Section 4.2.2 is used with a discretized set of hyperparameters. Analogously,  $\boldsymbol{\theta}_g$  summarizes all hyperparameters of the discriminative GP. Additionally, it is assumed that the available input domain is bounded by box constraints given by  $\mathbf{l} \in \mathbb{R}^d$  and  $\mathbf{u} \in \mathbb{R}^d$  as described in (4.26). The query selection problem (4.26) is also solved with a Newton method, where the required

---

### Algorithm 4.1: Stationary Safe Active Learning with GPs

---

**Require:**  $\mathcal{D}_{n_0}, \nu, \boldsymbol{\theta}, \boldsymbol{\theta}_g, \mathbf{l}, \mathbf{u}$

**Ensure:** Safe and near optimal stationary input design  $\mathbf{X}$

- 1:  $i = n_0$
  - 2: Train exploratory and discriminative GP on  $\mathcal{D}_{n_0}$
  - 3: **while**  $i < n$  **do**
  - 4:    $i = i + 1$
  - 5:   Determine and query  $\mathbf{x}_i$  from solving (4.26)
  - 6:   Sample  $y_i, c_i$  or  $h_i$  and add them to  $\mathcal{D}_i$  with query  $\mathbf{x}_i$
  - 7:   Train exploratory and discriminative GP on  $\mathcal{D}_i$
  - 8: **end while**
-

derivatives of the safety constraint are given in Section A.3.4 of the appendix in Equation (A.71) and (A.72). Generally, this optimization problem with respect to the predictive variance possesses a lot of local maxima. To ensure a good query solution, multi-starts on different randomly selected input locations are carried out and the best one is chosen. In addition to the further requirements, it is assumed that at least  $n_0 \geq 1$  safe starting points, which regards to input points with a positive class label  $c_i = +1$ , are given at the beginning of the exploration. For notational simplicity, let  $\mathcal{D}_i$  contain all the data for both GPs in Algorithm 4.1. As stopping criterion for the above problem, the maximal number of queries  $n$  is used. It is also possible to replace the latter by a bound for the current exploratory GP model accuracy. For the determination of the confidence parameter  $\nu$ , which strongly influences the safety and exploration behavior of the active learner, a connection to the overall probability of failure as shown in the next section provides a sufficient and user-friendly choice for this parameter. Also, a bound for the minimal required number of safe starting points for our safe active learning scheme is presented in the following part of this thesis. These results enable a great support for the final user of Algorithm 4.1.

## 4.2.5 Theoretical Analysis

In this section, a theoretical analysis of our proposed safe exploration scheme with respect to the active learning setting is considered. The goal is to investigate and analyze the influences of the parameters, for example the confidence parameter  $\nu$ , of our proposed safe active learning algorithm. In the following analysis, only stationary covariance functions with a magnitude  $\sigma_g^2$  as previously presented in Section 2.1.4 are employed for learning the discriminative GP. It should be noted that the main objective of our exploration strategy is to avoid samples from unsafe regions  $\mathbb{X}_-$  as much as possible. However, a desirable property of an exploration scheme is that it induces a nearly space-filling, i.e. uniform, distribution of the queries in the confined input space  $\mathbb{X} \subset \mathbb{R}^d$ . More precisely, an exploration strategy is called space-filling, if it leads to a low-discrepancy input design  $\mathbf{X}$  of size  $n$  such that the discrepancy

$$D(\mathbf{X}) = \sup_{B \in \mathcal{B}(\mathbb{X})} \left| \frac{1}{n} |\{\mathbf{x}_i \mid \mathbf{x}_i \in B\}| - \mu_d(B) \right| \leq \gamma \frac{\log^d(n)}{n} \quad (4.27)$$

for some positive constant  $\gamma$  independent of  $n$  and the normalized Lebesgue measure  $\mu_d(B)$  for any set  $B$  which is contained in the Borel algebra  $\mathcal{B}(\mathbb{X})$ . An example exploration strategy which yields a low-discrepancy input design is the Sobol sequence, see Sobol (1976). Intuitively, it is clear that a space-filling exploration scheme will cover the input space  $\mathbb{X}$ , and thus quest in dangerous regions  $\mathbb{X}_-$ . This supposition is confirmed by the following theorem, where the proof by contradiction is given in Section A.3.5 of the appendix.

**Theorem 4.1.** *For every space-filling exploration strategy on  $\mathbb{X}$  with a Lebesgue measurable subset  $B \subset \mathbb{X}_- \subset \mathbb{X}$ , such that  $\mu_d(B) > \epsilon$  for some  $\epsilon > 0$ , there exists a query  $\mathbf{x}_n \in B$  for an adequate large  $n$  possibly depending on  $\epsilon$ .*

For the initialization of our safe active learning scheme summarized in Algorithm 4.1, it is assumed that at least one safe starting point with positive label is given. However, depending on the hyperparameter  $\theta_g$  and especially on the confidence parameter  $\nu$ , the optimization problem (4.26) can be empty, i.e. there may not exist any instance  $\mathbf{x}_* \in \mathbb{X}$  fulfilling the safety constraint. This behavior is due to the classification

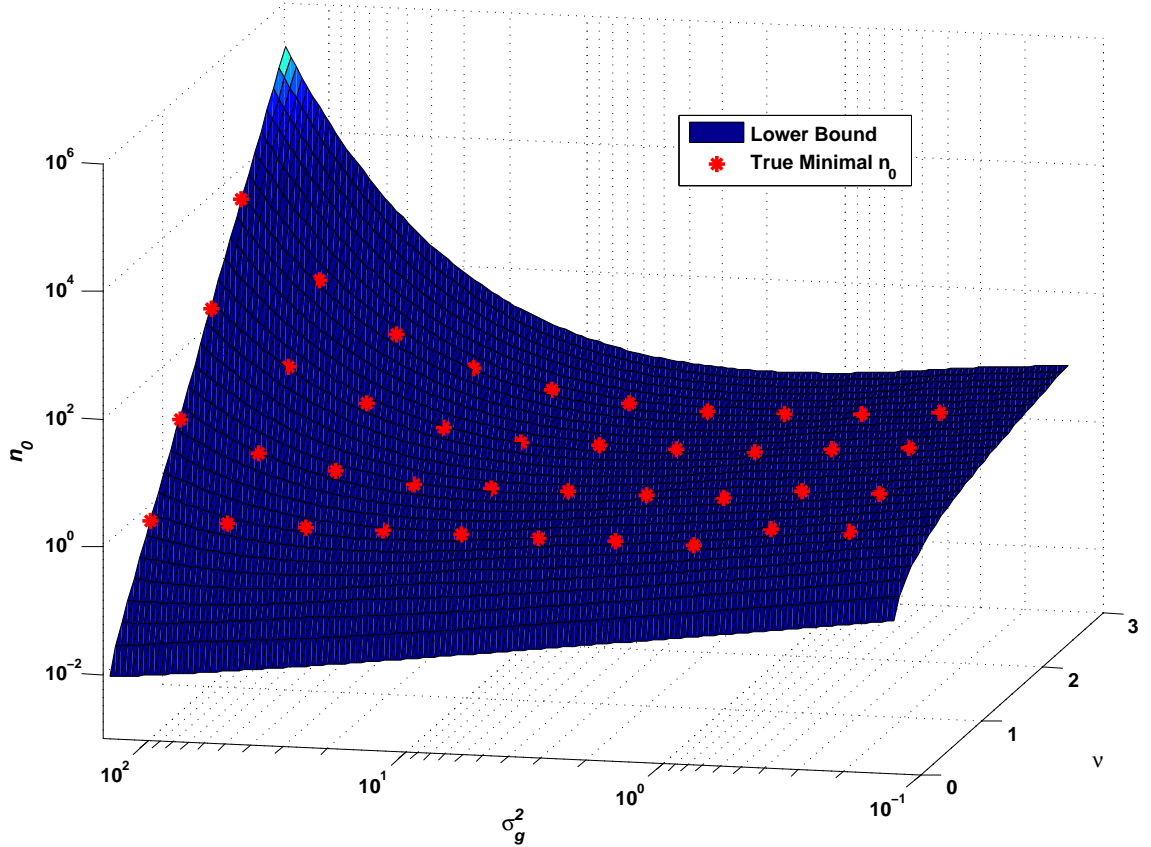


Figure 4.2: Lower bound for the number of necessary initialization points  $n_0$  given by Theorem 4.2 to ensure a non-empty optimization problem (4.26). The red stars indicate some true necessary set sizes, calculated according to the explanations in the proof.

part for learning the discriminative function  $g$ . Namely, this problem occurs if the user wants a very safe exploration which leads to a high confidence parameter  $\nu$ , but the unreliable GP classification model, which consists only of a few data points, is too uncertain to enable exploration under such a strong safety constraint. To solve this problem and to obtain a more confident discrimination model, the user has to define an initialization point set where potentially all points lie close to each other. The next theorem provides a lower bound for the size  $n_0$  of the initial point set. This result is especially relevant when employing the proposed safe active learning algorithm with the employed Laplace approximation for the discriminative GP in practice. The proof of the theorem is shifted to Section A.3.5 of the appendix.

**Theorem 4.2.** *To ensure a non-empty safety constraint in the optimization problem (4.26) for our GP based safe active learning setting under the Laplace approximation of the discriminative GP, at least an initial point set  $\mathcal{D}_{n_0}$  is needed with size*

$$n_0 \geq \left( 2\mathcal{N}\left(\frac{1}{2}(\sqrt{1+4\nu\sigma_g} - 1)\right) \right)^{-1} \min\left(\frac{\nu}{\sqrt{3}\sigma_g}, \sqrt{\frac{\nu}{\sqrt{3}\sigma_g^3}}\right). \quad (4.28)$$

In Figure 4.2, the bound from Theorem 4.2 is illustrated and compared to some true necessary set sizes  $n_0$ . As it is hard to restrict the explicit expressions for  $\mu_{g^*}$  and  $\sigma_{g^*}$  depending on  $n_0$ , the bound of the

theorem is adequately tight. Nevertheless, the magnitude and the asymptotic behavior are captured by the lower bound of the theorem. Note that the safety constraint of the optimization problem (4.26) is always satisfied, if  $\nu$  is non-positive, resulting in  $p \leq 0.5$ , which follows when a centered GP prior for the discriminative function  $g$  is used.

Finally, the probability of failure for our active exploration scheme is bounded to ensure a high level of safety. Having already queried  $i - 1$  points and if our prior GP assumptions are correct, the probability of failure when sampling  $\mathbf{x}_* \in \mathbb{X}$  without considering the safety constraint (4.23) is given by

$$\Pr_{\mathbb{P}} [g_* \leq 0 \mid \mathbf{x}_*, \mathcal{D}_{i-1}, \boldsymbol{\theta}_g] = 1 - \Phi\left(\frac{\mu_{g_*}}{\sigma_{g_*}}\right), \quad (4.29)$$

where the moments of  $g_*$  are explained by the exact posterior distribution (4.17) following from the mixed model likelihood (4.16). Again, let  $\mathcal{D}_i$  contain all of the sampled data according to Algorithm 4.1 up to the  $i$ -th iteration. In other words, Equation (4.29) means that the discriminative function is not positive in the case of a failure. Hence, of interest is an upper bound for the probability of making at least one failure, when querying  $n$  data points with our safe active learning scheme summarized in Algorithm 4.1. Our result is stated in the following theorem, where the sketch of the proof is subsequently given.

**Theorem 4.3.** *Let  $\mathbb{X} \subset \mathbb{R}^d$  be compact and non-empty, pick  $\delta \in (0, 1)$  and define the confidence parameter  $\nu = \Phi^{-1}\left(1 - \frac{\delta}{n - n_0}\right)$ . Ensure that the discriminative prior and posterior GP is exact. Suppose also that at least an initialization point set of size  $n_0 < n$  with respect to Theorem 4.2 is given. For selecting  $n$  possible queries satisfying the safety constraint, our active learning scheme presented in Algorithm 4.1 is unsafe with probability  $\delta$  resulting from*

$$\Pr \left[ \bigvee_{i=n_0+1}^n (g_i \leq 0 \mid \mathbf{x}_i : \mu_{g_i} - \nu \sigma_{g_i} \geq 0, \mathcal{D}_{i-1}, \boldsymbol{\theta}_g) \right] \leq \delta .$$

*Proof of Theorem 4.3.* Firstly, the probability of failure (4.29) is bounded for every possible query  $\mathbf{x}_*$  fulfilling the safety constraint. For an arbitrary but firmly selected input point  $\mathbf{x}_i \in \mathbb{X}$

$$\begin{aligned} & \Pr_{\mathbb{P}} [g_i \leq 0 \mid \mathbf{x}_i : \mu_{g_i} - \nu \sigma_{g_i} \geq 0, \mathcal{D}_{i-1}, \boldsymbol{\theta}_g] \\ & \leq \Pr_{\mathbb{P}} [g_i \leq \mu_{g_i} - \nu \sigma_{g_i} \mid \mathbf{x}_i : \mu_{g_i} - \nu \sigma_{g_i} \geq 0, \mathcal{D}_{i-1}, \boldsymbol{\theta}_g] \\ & = 1 - \Phi(\nu) , \end{aligned}$$

holds true under our condition. That is, the safety constraint (4.23) induces a probability of failure which is less than  $1 - p$  in each iteration, remembering the relationship  $p = \Phi(\nu)$ . Furthermore, the union bound is used to obtain

$$\begin{aligned} & \Pr \left[ \bigvee_{i=n_0+1}^n (g_i \leq 0 \mid \mathbf{x}_i : \mu_{g_i} - \nu \sigma_{g_i} \geq 0, \mathcal{D}_{i-1}, \boldsymbol{\theta}_g) \right] \\ & \leq \sum_{i=n_0+1}^n \Pr_{\mathbb{P}} [g_i \leq 0 \mid \mathbf{x}_i : \mu_{g_i} - \nu \sigma_{g_i} \geq 0, \mathcal{D}_{i-1}, \boldsymbol{\theta}_g] \\ & \leq (n - n_0) (1 - \Phi(\nu)) = \delta . \end{aligned}$$

Note that the  $n_0$  initialization points are feasible with probability 1 under the assumptions of the theorem.  $\square$

The lower bound of Theorem 4.3 provides a safety level of Algorithm 4.1 greater or equal than  $1 - \delta$ . Thus, the user is able to choose a sufficiently small  $\delta$  when carrying out our algorithm. After determining  $\delta$ , the confidence parameter  $\nu$  is calculated and, if necessary,  $p$  as explained in the theorem. It is clear that, for fixed safety level,  $\nu$  has to increase, if  $n$  increases. In this case, the number of necessary initialization points also increases, see Theorem 4.2. To get a more detailed illustration of the bound in Theorem 4.3, it is assumed that each query is selected independently of all others. In contrast to the proof of the safety bound, where no independence was assumed,  $1 - p^{n-n_0} \leq \delta$  is obtained. Intuitively, an upper bound for the expected number of failures is anticipated when sampling  $n$  points. Examples for this bound are shown in the next section, where our safe active learning algorithm is evaluated. Moreover, it is necessary for our presented active learning scheme that the function  $h$  is a lower bound of the true discriminative function of the system. In this case, information can be lost while exploring the system. However, the validity of the safety level compared to Theorem 4.3 is the main requirement. The function  $h$  must also satisfy the conditions for the specified discriminative GP prior like continuity and mean square differentiability.

## 4.2.6 Evaluations

In this section, the presented safe active learning algorithm 4.1 is verified by a one-dimensional toy example, and subsequently evaluated by employing our safe exploration scheme on an inverse pendulum policy search problem. The toy example is described by the cardinal sine function

$$f(x) = 10 \operatorname{sinc}(x - 10) + \varepsilon \quad (4.30)$$

with additive Gaussian noise  $\varepsilon \sim \mathcal{N}(0, 0.0625)$  on the interval  $\mathbb{X} = [-20, 20]$ . Firstly, the approach presented in Section 4.2.2 for active learning with GPR models under uncertainty in the associated hyperparameters is considered. The safety issue is neglected in the first setting, where only the exploration behavior is analyzed. Hence,  $m = 4$  mixture components are employed for the GPR prior, with a zero mean function and the same slightly higher noise variance  $\sigma^2 = 0.1$ . For the used isotropic Gaussian kernel (2.26) the magnitude is set to  $\sigma_{f_j}^2 = 2.5j$  and the length-scales are determined by  $\lambda_j = 5 - j$  for  $j = 1, \dots, 4$ . The initial point is specified by  $x = 0$ . All subsequently included query points are selected over optimization of the query strategy (4.12) according to Algorithm 4.1 without the safety constraint. In Figure 4.3, the first seven selected input points, the mixture of GP priors with its four components, and the true cardinal sine function are illustrated. The mixing probabilities  $\omega_j$  after the seventh inclusion are given by

$$\omega^T = \begin{pmatrix} 0.30 & 0.42 & 0.19 & 0.09 \end{pmatrix}$$

according to Equation (4.13) and lead to the new query point  $x = -4.63$  with highest entropy. Due to the multi-modality of the predictive variance (4.21), as shown in Figure 4.4, a multi-start with ten input points uniformly sampled from the interval  $[-20, 20]$  is employed for adequately solving the optimization task in Equation (4.12). Nevertheless, it can happen that the currently best query is not found by the multiple initialized second-order optimization algorithm, where even different points converge to the same query solution. That is the reason why in Figures 4.3 and 4.4 only six possible new queries can be seen. During the active data-sampling process, a nearly uniform distribution of the queries over the input

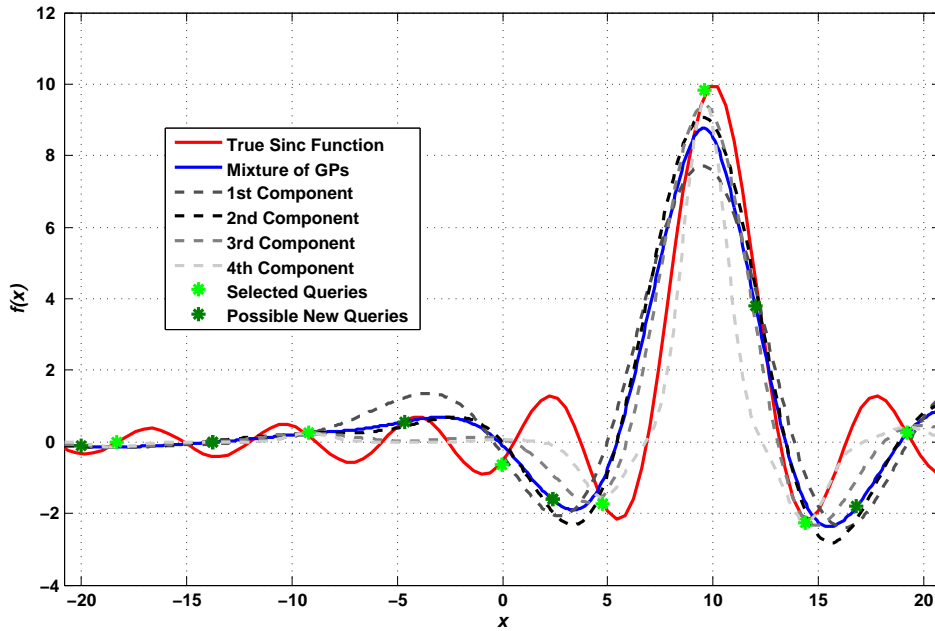


Figure 4.3: Exploration of the cardinal sine function (4.30) with the entropy-based sampling strategy according to the mixture of GP priors. The mean values of each mixture component and the resulting mixture model according to the so far selected seven queries is presented.

interval is observed which yields an accurate approximation of the cardinal sine function with finally  $n = 40$  points. Compared to an active learning scheme based on only one exploratory GPR model with fixed hyperparameters, namely no mixture model, nearly the same solution for this simple one-dimensional

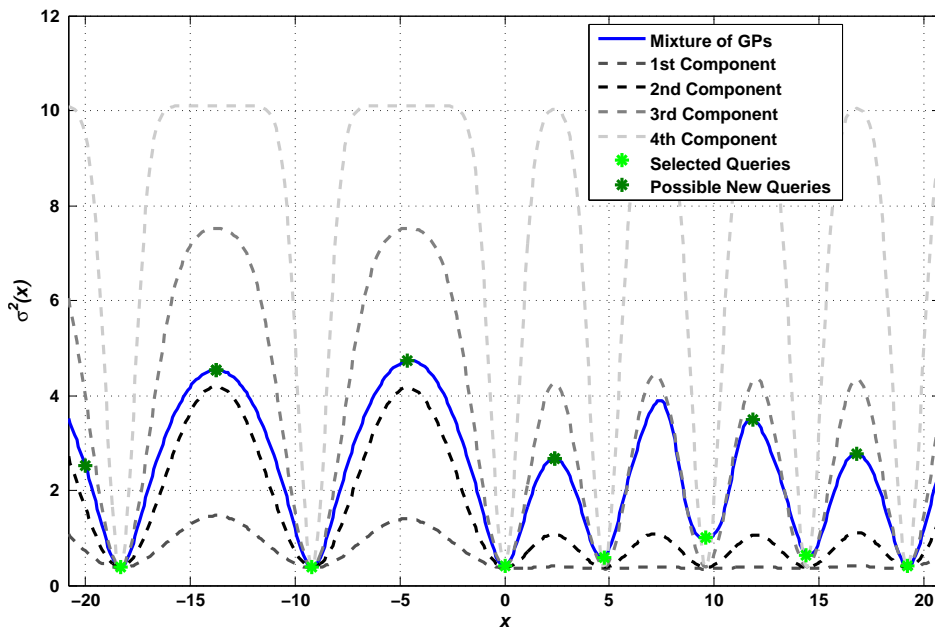


Figure 4.4: Predictive variance of the mixture of GP priors for the one-dimensional toy example. Therein, the multi-modality of the variance is good to recognize, which is the reason for a multiple initialization of the optimization problem (4.12) with uniformly selected start points.

example is obtained. This results from the fact, that the influence of the hyperparameters vanish, because the magnitude and the length-scale parameter do not modify the final query solution, which is almost always in the middle of two former selected points. But in higher dimensions, especially the influence of the length-scale parameters according to the specified covariance functions with ARD is significant. Thus, the higher computational effort for the mixture of GP priors is only justified for higher input dimensions. In this case an extension of this framework is proposed to reduce the computational costs. Namely, after selecting enough but not all queries, the hyperparameters for the mixture component with the highest weight are continuously adapted over likelihood optimization with further increasing included points. The result of this strategy is that all other weights will become zero, if the hyperparameters of the optimized component fit the system well. Hence, only the optimized component will survive and all other components will be neglected for active learning to save memory and computation time. Thus, if the hyperparameters are nearly known, a standard GPR model with careful adaption of the hyperparameters can be used. Hence, only the case of fixed hyperparameters is considered in the next evaluations when introducing the safety issue.

For introducing safety in the toy example of exploring the noisy cardinal sine function (4.30), a safe region  $\mathbb{X}_+ = [-5, 11]$  is defined. Consequently, the unsafe regions are given by  $\mathbb{X}_- = \mathbb{X} \setminus \mathbb{X}_+$ , from which always negative labels are sampled. Otherwise, i.e. within  $[-4, 10]$ , only positive class labels are sampled. To recognize a dangerous exploration within the presented GP based classification framework, the quadratic function

$$h(x) = \begin{cases} \frac{(x+5)^2}{2} & \text{for } -5 \leq x < -4 \\ \frac{(x-11)^2}{2} & \text{for } 10 < x \leq 11 \end{cases} \quad (4.31)$$

is defined around the decision boundary. The Gaussian observation noise of the function  $h$  is set to  $\mathcal{N}(0, 0.01)$ . The hyperparameters of the exploratory and discriminative GP are set to  $\sigma_f^2 = 4$ ,  $\lambda = 3$ , and  $\sigma_g^2 = 1$ ,  $\lambda_g = 10$ , respectively. The  $n_0$  starting points with respect to Theorem 4.2 are uniformly sampled in the range  $[-0.5, 0.5]$ . Finally,  $n = 40$  input points are sampled for different probability levels  $\delta$ . Table 4.1 shows the selected safety levels with the corresponding confidence parameters  $\nu$  and the necessary

$\delta$	$\nu$	$n_0$	H	SEN	SPC	Number of failures	Number of expected failures
0.05	2.99	4	17.39	1.00	1.00	0.0	1.8
0.10	2.77	4	18.54	1.00	0.90	0.2	3.4
0.20	2.54	4	18.73	1.00	0.80	0.6	6.5
0.30	2.40	3	18.69	1.00	0.50	1.1	9.6
0.40	2.30	3	18.88	1.00	0.60	1.6	12.3
0.50	2.21	3	18.87	1.00	0.80	2.3	14.6

Table 4.1: Averaged results over ten runs of the one-dimensional toy example. The differential entropy and the number of failures is slightly decreasing for smaller  $\delta$ . Otherwise, the parameter  $\nu$  and  $n_0$  are growing with decreasing  $\delta$  as explained in the associated theorems. The sensitivity (SEN) of the discriminative GP is always perfect, where the specificity (SPC) varies strongly due to the very unbalanced data and the noise of the function  $h$ . Note that the failures are closely located around the decision boundary.

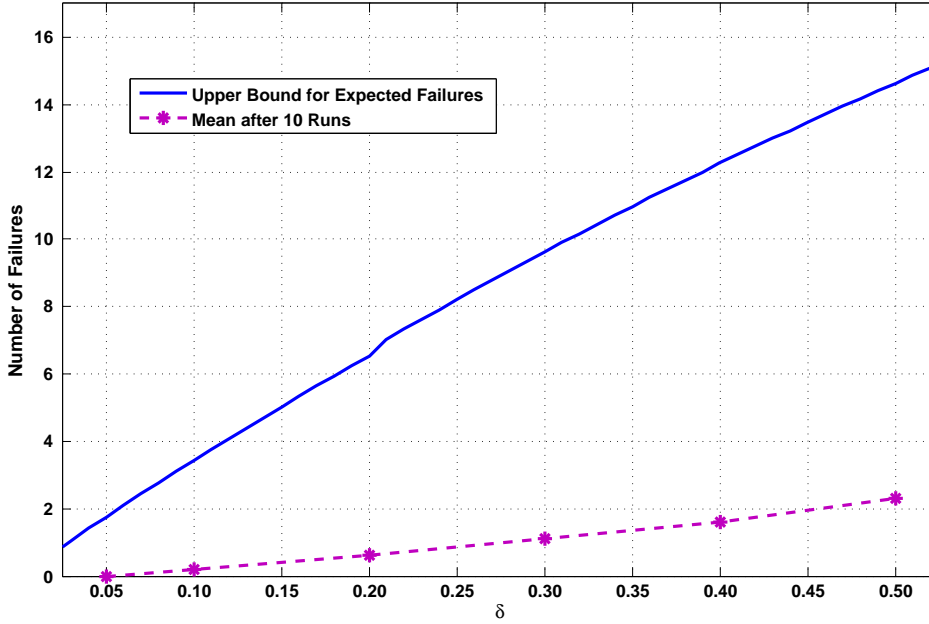


Figure 4.5: Expected number of failures calculated under the independence assumption (upper bound) for the toy example compared with the failures obtained by averaging over ten runs of the safe active learning algorithm 4.1.

number of starting points  $n_0$  derived in the proposed theorems. The results in the presented table, for example the final differential entropy (4.2) and the number of failures, are averaged over ten runs. Besides that, the table provides the sensitivity and specificity of the discriminative GP model, where it is shown, that it is hard to classify especially the unsafe queries correctly. The maximum possible differential entropy value with  $n = 40$  points is 19.50, which is almost reached in all cases. The expected number of failures provides only an upper bound for the true bound, since independence is assumed in its calculation over  $(n - n_0)p$ . These bounds are additionally compared in Figure 4.5. Due to this strong assumption, the upper bound for the expected number of failures will not be very tight. Additionally, the mostly conservative estimation of the discriminative posterior distribution by the Laplace approximation reinforce this behavior. The normalized entropy ratios for all cases averaged over ten runs are illustrated in Figure 4.6, cf. Equation (4.6). Note that for the calculation of the differential entropy  $H$  only the safe queries are taken into account, analogously to Table 4.1. Therefore, the curves for the normalized entropy ratios end before the final number of queries is reached. This figure also shows the effects of the decreasing failure level  $\delta$  for the number of initialization points and the gain in entropy, as presented by the theorems in Section 4.2.5, where a greater input region is explored faster with a higher failure probability  $\delta$ . In Figure 4.7 the final result for one run of the safe active learning Algorithm 4.1 is presented after selecting 40 queries. The cardinal sine function is learned accurately over the safe input region. Furthermore, the figure shows that the small definition regions for the function  $h$  around the decision boundary are sufficient for safe exploration. Also the exploration behavior of the regression model is still good, see Figure 4.6.

As a second test scenario, the exploration of the control parameters for the inverse pendulum hold up problem is considered. Here, the mapping between the parameters of a linear controller and their



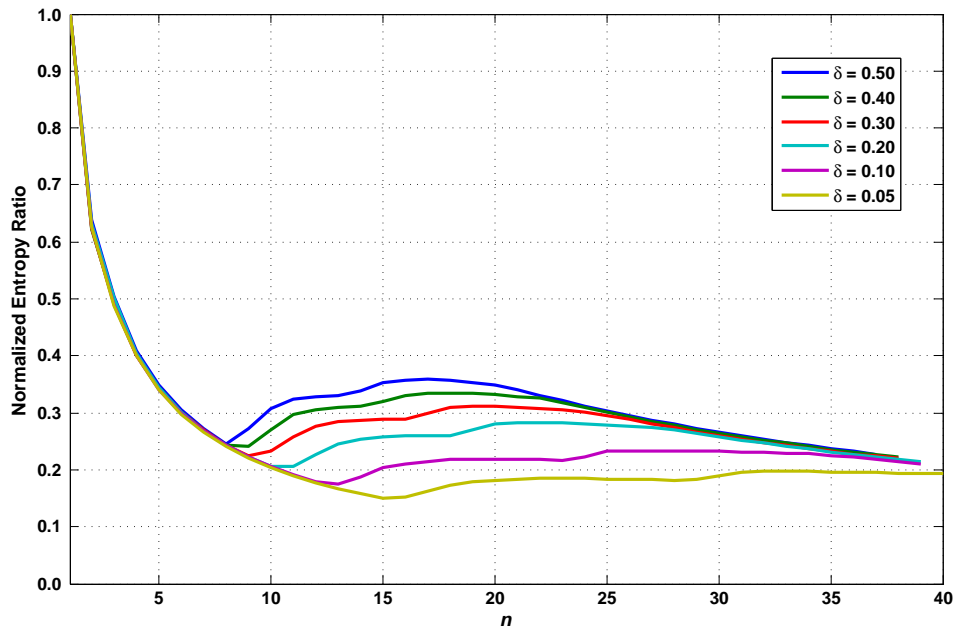


Figure 4.6: Normalized entropy ratios (NERs) for different safety levels, averaged over ten runs. In the beginning phase, the ratios fall until the safety constraint is satisfied, since it is only sampled around zero. After that, the gain in entropy increases until the safe region is explored. This behavior and the value of the gain depends on  $\delta$  and the confidence parameter  $\nu$ , respectively.

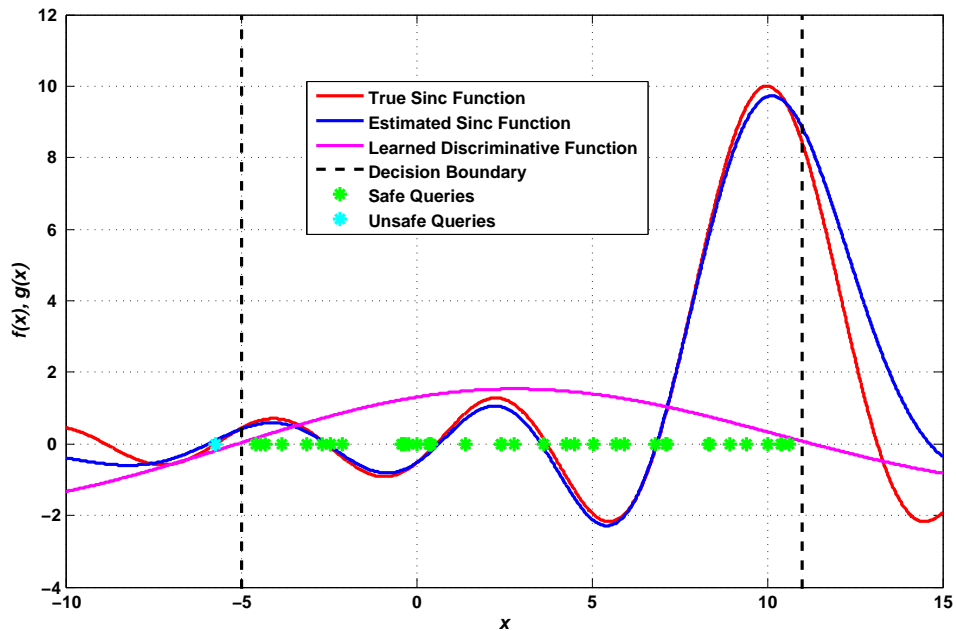


Figure 4.7: Final result for safe active learning of a generalized cardinal sine function in the secure interval  $[-5, 11]$  with 40 queries and  $\delta = 0.30$ . Only one selected query fails, it means that she falls below the lower decision boundary. All other chosen data points cover the safe input space  $\mathbb{X}_+$  well. The final discriminative GP separates the safe and unsafe regions of the whole input space adequately, even if no query above the upper border is selected.

performance on keeping the pole upwards is learned. The goal is an exploring of these parameters while avoiding that the pendulum falls over. The simulated system is an inverse pendulum mounted on a cart as shown in Figure 4.8, cf. Deisenroth et al. (2015a). Hence, the system state  $\mathbf{s}_t \in \mathbb{S}$  is 4-dimensional (cart position  $z_t$  and velocity  $\dot{z}_t$ , pendulum angle  $\vartheta_t$  and angular velocity  $\dot{\vartheta}_t$ ), which results in 5 open parameters, namely  $\mathbf{x}$  and  $x_0$ , of the linear controller  $\pi(\mathbf{s}_t) = \mathbf{x}^T \mathbf{s}_t + x_0$ . The desired target state is defined by a designated cart origin at zero and the pendulum pointing upwards with zero degrees. The controller is applied for 10

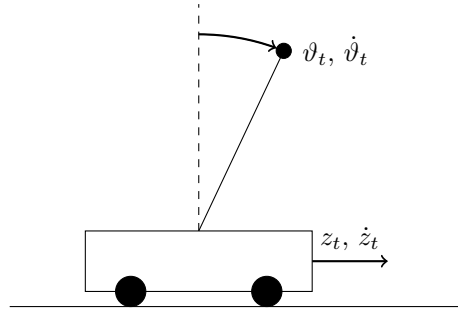


Figure 4.8: Illustration of the inverse pendulum hold up problem.

seconds with a control frequency of  $10\text{ Hz}$  starting from a state sampled around the defined target state. Furthermore, the performance of a given controller is evaluated by first measuring the distance of the current state  $\mathbf{s}_t$  to the target state, formally with  $\|\mathbf{s}_t - \mathbf{0}\|$ , for each time step  $t$ . These distances are used to compute the saturating cost  $1 - \exp(-\|\mathbf{s}_t\|^2)$  during the roll-out. Then, the average over all time steps and ten different starting states is calculated to obtain a meaningful interpretation of the final cost  $y$  which will be explored by our active learning scheme. Note that this cost additionally contains Gaussian noise  $\varepsilon \sim \mathcal{N}(0, 0.001)$ . The hyperparameters of the target and discriminative GP are previously learned over a uniformly distributed point set of size 100 over the entire input space, which concerns the policy parameter space  $\mathbb{X} \subset \mathbb{R}^5$ . The function  $h$  is defined over the deviation from the unstable target state for each time step, namely  $2 - |z_t| \mathbb{I}(|z_t| > 1\text{ cm}) - |\vartheta_t| \mathbb{I}(|\vartheta_t| > 1^\circ)$ , where  $\mathbb{I}(\cdot)$  is the indicator function. As for the final cost  $y$ , these local errors are averaged over all time steps of the roll-out and all starting states to get the value of  $h$ . If  $h \geq 0.95$ , a positive class label is obtained while the label is negative for  $h \leq -1$  or the pendulum falls down. In this case  $n = 1000$  input points are queried for various values of  $\delta$ . Table 4.2 summarizes the resulting values for  $\nu$  and  $n_0$  given by the theorems in Section 4.2.5. The value of the differential entropy  $H$  increases until  $\delta = 0.10$ , where a good tradeoff between a low number of failures and a fast exploration is provided. The quality of the discriminative approach with respect to both considered classification errors differs considerably, where the main reason for this behavior is the

$\delta$	$\nu$	$n_0$	H	SEN	SPC	Number of failures	Number of expected failures
0.01	4.26	8	393.3	1.00	1.00	0	9.9
0.05	3.89	7	397.7	1.00	0.45	6	48.4
0.10	3.72	6	412.8	0.99	0.26	29	94.6
0.20	3.54	6	402.7	1.00	0.41	57	180.2
0.30	3.43	5	395.6	0.99	0.49	80	257.9
0.40	3.35	5	389.3	1.00	0.61	101	328.1
0.50	3.29	5	380.7	0.99	0.72	127	391.6

Table 4.2: Median with respect to the number of failures calculated over ten runs of the policy exploration task from the cart pole. Here, the entropy decreases as  $\delta$  increases, except for the lower values of  $\delta$ , since the number of unsafe queries increases strongly. Analogously to the toy example, it is hard to detect the few failures and obtain a high specificity (SPC). In contrast, the safe queries are mostly right classified due to the high sensitivity (SEN).

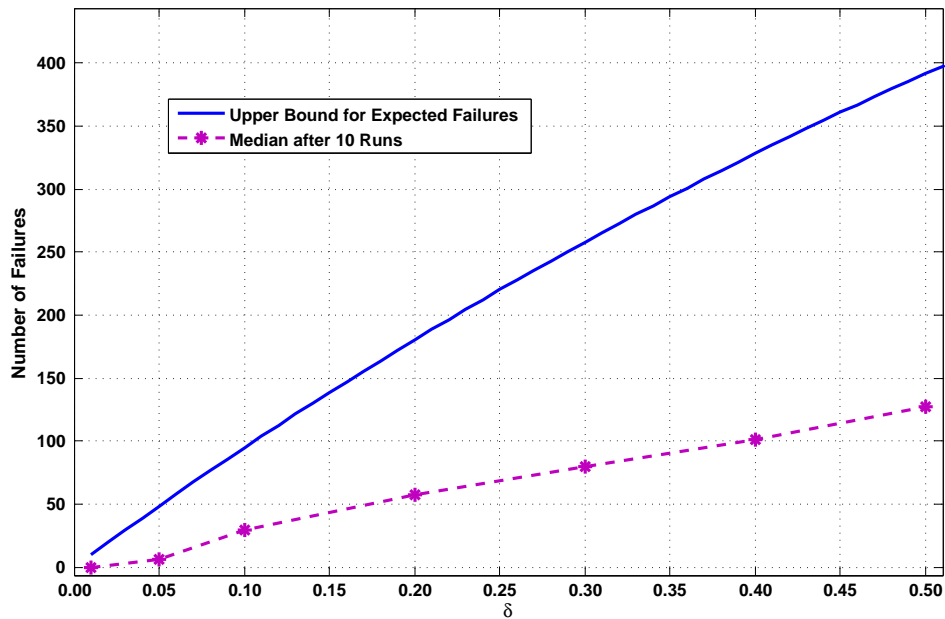


Figure 4.9: Expected number of failures calculated under the independence assumption (upper bound) for the policy search task with the failures obtained by the median of ten runs of the safe active learning algorithm.

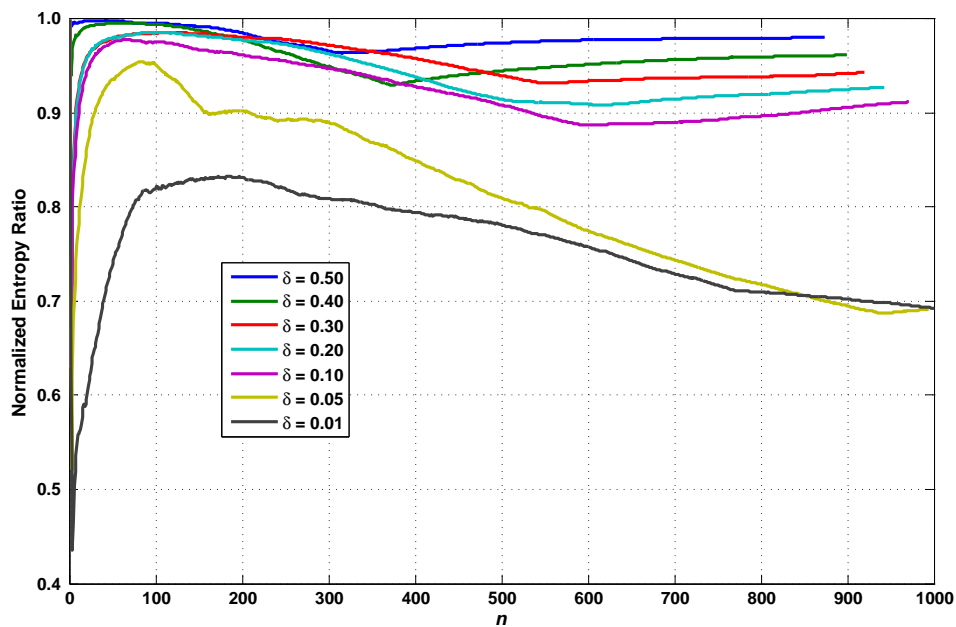


Figure 4.10: Median of normalized entropy ratios (NERs) for different safety levels over ten runs of the policy search task. In the beginning phase, the ratios are nearly maximal, that means close to one, since many very safe queries with positive class labels are sampled which yield a strong exploration behavior. After the beginning phase, the safe region is roughly explored until a slight valley is reached. Then, the most queries are sampled from the inner region.

unbalanced data set. This behavior is further supported by the not perfectly estimated hyperparameters. Regarding the safety issue, a high specificity should be desirable, perhaps with slightly decreasing sensitivity, to provide a little more conservative exploring characteristic. Figure 4.9 shows that the median of the number of failures after ten runs of our safe exploration scheme is much lower than the upper bound of the expected number. This behavior clarifies the effectiveness of the Theorem 4.3 for our safe active learning scheme, analogously to the results of the toy example. Also the nearly heuristic definition of the function  $h$  works very well for this exploring task on five different input dimensions. A reason for that is the almost always favorable modeling accuracy when using GPs. The medians of the normalized entropy ratios after ten runs of the inverse pendulum control task are presented in Figure 4.10. This trends show the effect of the different parameters  $\delta$  after the beginning phase, where more exploration yields a higher curve. The order of the curves results from the chosen safety level  $1 - \delta$ , too.

### 4.3 Active Learning for Transient Environments

In contrast to the former section about static active learning scenarios, now the transient active learning setting is considered. The main difference between these two cases is the additional time-dependency caused by the considered non-stationary system which will be safely explored. Hereby, it is also focused on learning a transient GPR model from the dynamic environment with a limited budget of measuring effort, which means with a previously defined time for interacting with the transient system. The compact input space  $\mathbb{X} \subset \mathbb{R}^d$  of the fully actuated input variables is again subdivided in safe  $\mathbb{X}_+(t)$  and unsafe regions  $\mathbb{X}_-(t)$ . Note that these subspaces depend now on the time  $t$ . For the illustration of this behavior, consider a technical system like a combustion engine or gas turbine, cf. Hans et al. (2008), where after a long full load operation it is not possible to continue with other stressful system stimuli without damaging the environment. Thus, the active learner has to decide depending on the time which trajectory should be measured at next, or in difficult cases, to drive back to the given safe and stationary initial point. Remark, that here are trajectories considered instead of a stepwise point to point planning, since when measuring on physical systems with 100 Hz or more, a stepwise approach seems hard to be realized under real-time conditions. Furthermore, an efficient trajectory planning approach has the advantage to avoid dead ends with respect to safety issues and to provide more global exploration behavior due to the inertia of the environment. Hence, it is necessary to focus on efficient and easy to parametrize trajectory planning strategies. Analogously to the static case, it is assumed that the active learner observes some feedback from the system for each point of the trajectory which indicates the burdening of the considered dynamic system. This feedback is used to learn a transient and discriminative GPR model to determine safe trajectories. Contrary to the stationary setting, the training of the discriminative approach is simplified in this dynamic setting, where the system response is only considered to be consistent and continuous. Equivalently to the former approach, an exploratory GPR model can be employed to design the entropy-based search strategy of the transient active learner. Subsequently, both GPs are incrementally adapted if a complete trajectory with new sampled system outputs and responses is queried. Due to the large amount of data arising in transient learning scenarios, both tasks are modeled with sparse GPR approximations, more precisely, with the DTC approximation and our new introduced maximum error greedy selection

criterion as presented in Section 3.2. Additionally, for accurate transient modeling the NARX approach from Section 2.3 is considered, where the associated structure of features is assumed to be given. Further details for our transient active learning scheme are presented in the next sections.

### 4.3.1 Exploring Strategy

The base for exploration of the transient system is a given stationary and safe input point  $\mathbf{x}_0$  with the corresponding system output  $y_0$ . Moreover, the initial point  $\mathbf{x}_0$  should not lie near to any border of the whole input space  $\mathbb{X}$ . For the effective exploration of transient systems it is not only needed to consider an information-optimal design of  $\mathbb{X}$ , it is furthermore necessary to explore at least the first derivatives of the input variables. This means that each input point  $\mathbf{x}$  should be visited with different gradients  $\dot{\mathbf{x}}$ . Thus, the underlying region for exploration is extended to the phase space  $\mathbb{P} = (\mathbb{X}, \dot{\mathbb{X}}) \subset \mathbb{R}^{2d}$ , where  $\dot{\mathbb{X}} \subset \mathbb{R}^d$  describes the gradient space. Besides the confined input space  $\mathbb{X} = \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{l} \prec \mathbf{x} \prec \mathbf{u}\}$ , the gradient space is also assumed to be restricted, which means that  $\dot{\mathbb{X}} = \{\dot{\mathbf{x}} \in \mathbb{R}^d \mid \dot{\mathbf{l}} \prec \dot{\mathbf{x}} \prec \dot{\mathbf{u}}\}$ , since the actuation of the input variables is mostly bounded regarding technical constraints of the considered system. Hence, the available phase space is also confined. Due to the fact that the safe initial point is stationary, the exploration of the phase space is started in  $\mathbf{x}_0$  with zero gradient. To achieve a good coverage of the phase space, firstly a low-discrepancy design  $\mathbf{P}_{\text{Sobol}} \in \mathbb{R}^{n \times 2d}$  of size  $n$  is defined over the space of interest  $\mathbb{P}$  determined according to Sobol (1976). Figure 4.11 illustrates the initial setting of our transient exploration scheme based on a 4-dimensional phase space with a safe stationary initial point at the coordinate origin. The goal is now to determine an information-optimal path through all points of

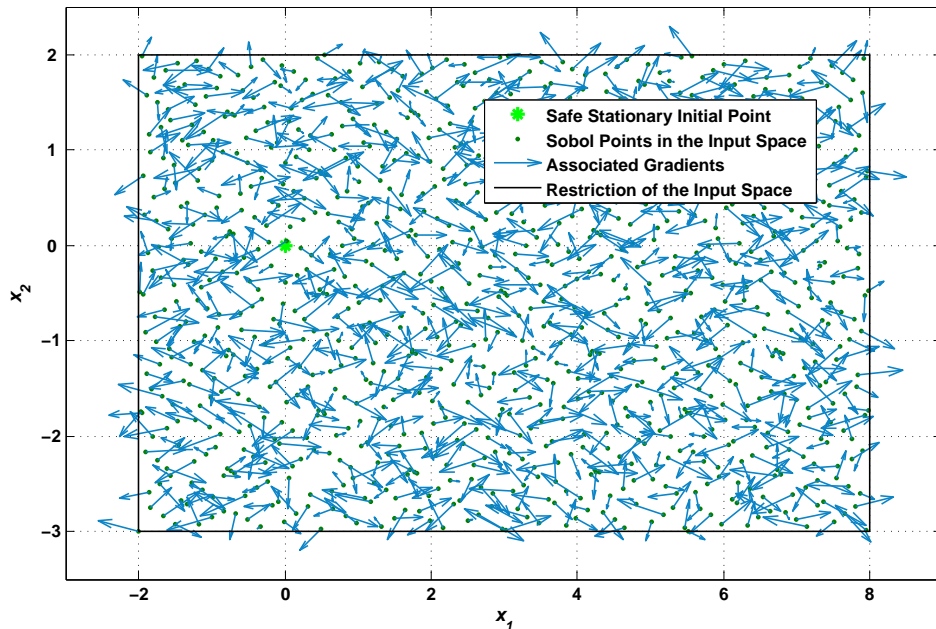


Figure 4.11: Visualization of a Sobol design with 1000 points for a bounded 4-dimensional phase space  $\mathbb{P}$  with a stationary starting point at zero. The transient active learner has now to determine the information-optimal path consisting of connected trajectories between all points in  $\mathbb{P}$ .

the Sobol design in the phase space and end up in the starting point  $\mathbf{x}_0$ . The previous planning of this complete path is too complex, due to its connection to the NP-hard traveling salesman problem, and not necessary, since the safety of this path can not be estimated a priori. Note that this approach can be seen as a pool-based active learning strategy, since all possible paths are defined by the previously selected set of Sobol points. Hence, the generation of this path is realized piecewise, namely a set of trajectories is planned from the current point  $\mathbf{p}_i \in \mathbf{P}_{\text{Sobol}}$  to some other points of the initial Sobol design which have never been visited before. The exploration of the transient environment finishes if the last point in  $\mathbf{P}_{\text{Sobol}}$  is visited and the approach subsequently arrives at the safe initial point  $\mathbf{x}_0$ . It is also possible to cancel the exploration scheme earlier, for example, after a high enough model quality has been achieved. Moreover, a sparse DTC model with the maximum error criterion, cf. Section 3.2.3, is employed on the so far obtained training data from the input space  $\mathbb{X}$  extended to the before determined NARX structure, see Section 2.3. Note that the gradient information is implicitly included in the DTC model with the NARX approach. Then, to decide which point to point trajectories in the phase space are the best, the differential entropy

$$\mathbb{H}[\mathbf{y}_* | \mathbf{X}_*, \mathcal{D}_i, \boldsymbol{\theta}] = \frac{1}{2} \log(|2\pi e \boldsymbol{\Sigma}_*|) \quad (4.32)$$

is calculated according to Equation (A.24) and the predictive distribution (3.8). In this case, multiple-step ahead prediction as described in Figure 2.2 is employed for each input trajectory  $\mathbf{X}_* \in \mathbb{R}^{l \times d}$  consisting of  $l$  data points in the input space. The collected input data, up to observing the  $i$ -th Sobol point in the phase space, is summarized in  $\mathcal{D}_i$ . As in the stationary active learning setting, it is assumed that the hyperparameters  $\boldsymbol{\theta}$  for the exploratory DTC approach are also previously determined. The predictive covariance matrix  $\boldsymbol{\Sigma}_* \in \mathbb{R}^{l \times l}$  is given by

$$\boldsymbol{\Sigma}_* = \mathbf{K}_{**} - \mathbf{K}_{I,*}^T \mathbf{L}^{-T} (\mathbf{I} - \sigma^2 \mathbf{M}^{-1}) \mathbf{L}^{-1} \mathbf{K}_{I,*} + \sigma^2 \mathbf{I}, \quad (4.33)$$

and nearly equivalently to Equation (3.9), since the consideration of the noisy outputs  $\mathbf{y}_*$  induces the additional term  $\sigma^2 \mathbf{I}$ . For the definition of the Cholesky factor  $\mathbf{L}$  and the matrix  $\mathbf{M}$  see Equation (3.6). As required for the calculation of the differential entropy (4.32), the determinant of the symmetric matrix  $\boldsymbol{\Sigma}_*$  can be calculated over the QR algorithm, cf. Lipschutz and Lipson (2013), due to the numerical stability. When applying the QR algorithm, it is possible to approximate the final entropy value with the help of the bounds (4.5) to gain computational efficiency for large trajectories. Since the selection of trajectories between phase space points from  $\mathbf{P}_{\text{Sobol}}$  is realized in a greedy manner, it is proposed to generate a small tree with restricted width and depth of trajectories, compute the differential entropy of the connected trajectories, and choose then the first trajectory which yields the highest entropy value according to the possible part of the future path. In doing so, the Sobol points in the knots of the tree are chosen randomly from the still not visited points in  $\mathbf{P}_{\text{Sobol}}$ . This extension enables a more global exploration character of the transient active learner. Since a path planning approach with continuous and differentiable trajectories is intended, the realizations of the predictive sparse GP are also continuous for the herein considered covariance functions, cf. Section 2.1.4. Hence, a theoretical interesting question is, what happens to the differential entropy (4.32) of a continuous trajectory, i.e. in the limit of a increasingly finer discretization. Intuitively, it is asserted that the differential entropy exists in this case, but a constructive proof of this issue would be hard to obtain and is therefore a topic of future work. Nevertheless, for the discretized trajectories  $\mathbf{X}_*$  it is desired to gain as much information as possible about the physical environment dependent on the current sparse DTC model. If an explicit representation of the trajectory

is given, their entropy is optimized equivalently to the scheme (4.3) which results in

$$\mathbf{X}_{i+1} = \arg \max_{\mathbf{X}_* \subset \mathbb{P}} (\mathbb{H}[\mathbf{y}_* | \mathbf{X}_*, \mathcal{D}_i, \boldsymbol{\theta}]) , \quad (4.34)$$

where  $\mathbf{X}_{i+1}$  describes the input trajectory between the  $i$ -th and randomly sampled  $(i + 1)$ -th design point from the initial Sobol plan  $\mathbf{P}_{\text{Sobol}}$ . This optimization task will be hard to solve for non-parametrized trajectories due to the potentially large number of degrees of freedom and the underlying NARX structure for transient modeling. Note that the box-constraints for the phase space  $\mathbb{P}$  must also be satisfied. Hence, a favorable approach for parametrized trajectory design regarding efficient computation and optimization is presented in the next section.

### 4.3.2 Efficient Trajectory Generation with Bézier Curves

The goal of this section is to provide an efficient and easy to handle parametrized trajectory generation approach for the transient exploration scheme presented in the former. Therefore, the method for determining trajectories has to fulfill some constraints. Firstly, the evaluation for a large discretized set of time stamps should be fast and memory friendly, i.e. approaches with huge look-up tables should be avoided. Another crucial point is the flexible design of the trajectory with only a few parameters to enable effective optimization algorithms for solving problems like (4.34). Furthermore, specifying the gradients at the beginning and the end of the trajectory according to the initial Sobol design in the phase space should be realizable with small effort. This also includes attention when planning the trajectory within the start and end point, since they always have to satisfy the phase space restrictions. Beside many other techniques for trajectory generation, like interpolation with B-splines, an approach based on Bézier curves is considered here due to their favorable properties regarding the requirements above. A broad overview about splines and Bézier curves is given by Salomon (2006), where the following definitions come from. A Bézier curve of order  $m$  is defined as

$$\mathbf{x}(\tau) = \sum_{j=0}^m B_{m,j}(\tau) \mathbf{c}_j \quad (4.35)$$

with  $\tau \in [0, 1]$ , control points  $\mathbf{c}_j \in \mathbb{R}^d$ , and the Bernstein polynomials

$$B_{m,j}(\tau) = \binom{m}{j} \tau^j (1 - \tau)^{m-j} , \quad (4.36)$$

cf. Bernstein (1913), which are expressed over the binomial coefficient

$$\binom{m}{j} = \frac{m!}{j!(m-j)!} .$$

The influence of the control points of the Bézier curve is global, since the Bernstein polynomials provide a basis of the complete unit interval. Compared to a B-spline approach, which is induced by basis functions with bounded support, the Bézier curve is less flexible, but therefore simpler to parametrize with respect to the number of control points. Analogously to B-splines, the Bézier curve passes by the first and last control point, where the number of control points is determined by their order, that means  $m + 1$  points  $\mathbf{c}_j$  are needed to generate a Bézier curve of order  $m$ . In Figure 4.12 a cubic Bézier curve for a two-dimensional input space is presented to illustrate this behavior. Due to the fact that the Bernstein

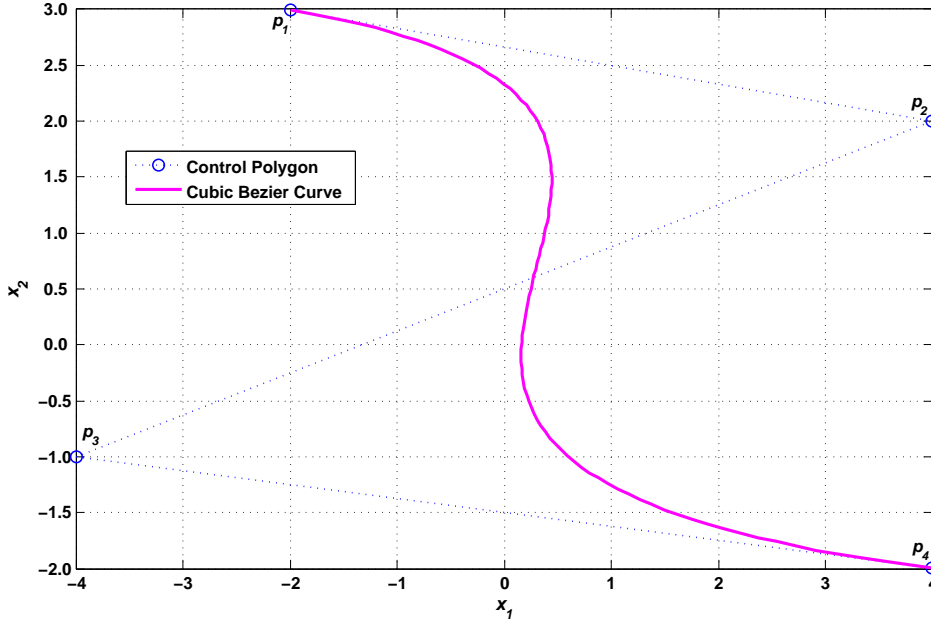


Figure 4.12: A cubic Bézier curve with four control points in a two-dimensional input space. Note that the Bézier curve always lies in the convex hull of his control polygon.

polynomials of any order  $m$  sum to one for  $\tau \in [0, 1]$ , it holds true that the associated Bézier curve lies in the convex hull of the control points. This behavior is exemplary shown in Figure 4.12. This fact can be employed for a preliminary verification that the Bézier curve is inside the input space  $\mathbb{X}$ . Note that the Bézier curve can also be inside  $\mathbb{X}$  when some of the control points are lying outside of  $\mathbb{X}$ . The calculation of points of a Bézier curve is realized after transforming Equation (4.35) to

$$\begin{aligned}
 \mathbf{x}(\tau) &= \sum_{j=0}^m \binom{m}{j} \tau^j (1-\tau)^{m-j} \mathbf{c}_j \\
 &= \sum_{j=0}^m \binom{m}{j} \tau^j \sum_{k=0}^{m-j} \binom{m-j}{k} (-\tau)^k \mathbf{c}_j \\
 &= \sum_{j=0}^m \sum_{k=0}^{m-j} \binom{m}{j} \binom{m-j}{k} (-1)^k \tau^{j+k} \mathbf{c}_j \\
 &= \sum_{j=0}^m \tau^j \sum_{k=0}^j \binom{m}{j-k} \binom{m-j+k}{k} (-1)^k \mathbf{c}_{j-k} \\
 &= \sum_{j=0}^m \tau^j \sum_{k=0}^j \binom{m}{j} \binom{j}{k} (-1)^{j+k} \mathbf{c}_j
 \end{aligned} \tag{4.37}$$

using some index substitutions, properties of the binomial coefficient, and the binomial theorem. To evaluate the Bézier curve on a large discretized point set with respect to the unit interval of the parameter  $\tau$ , firstly the inner sum of coefficient vectors for each  $j$  is calculated and stored, since these are fixed for the whole trajectory. Then, it is continued with computing the outer sum over the Horner scheme, cf. Horner (1819). Compared to De Casteljaun's algorithm, cf. Salomon (2006), less multiplications are needed and the memory effort for storing the coefficient vectors is required only once in the initialization phase. Besides that, the method by De Casteljaun require this memory effort for the calculation of one point



$\mathbf{x}(\tau)$ . To determine the minimal and maximal value of a given Bézier curve for each input dimension, the following derivative

$$\frac{dB_{m,j}(\tau)}{d\tau} = m(B_{m-1,j-1}(\tau) - B_{m-1,j}(\tau)) \quad (4.38)$$

is necessary and yields

$$\frac{d\mathbf{x}(\tau)}{d\tau} = m \sum_{j=0}^{m-1} B_{m-1,j}(\tau) (\mathbf{c}_{j+1} - \mathbf{c}_j) . \quad (4.39)$$

Thus, the derivative of the Bézier curve at  $\tau = 0$

$$\left. \frac{d\mathbf{x}(\tau)}{d\tau} \right|_{\tau=0} = m(\mathbf{c}_1 - \mathbf{c}_0) \quad (4.40)$$

and at  $\tau = 1$

$$\left. \frac{d\mathbf{x}(\tau)}{d\tau} \right|_{\tau=1} = m(\mathbf{c}_m - \mathbf{c}_{m-1}) \quad (4.41)$$

follows immediately. This behavior is also illustrated in Figure 4.12, where the tangential plane of the start and end point of the curve is only described by the first two and last two control points of the Bézier curve, respectively. Due to the definition of the transient exploring strategy for the phase space, with given start and end conditions for each trajectory in the former section, at least a Bézier curve of order  $m \geq 3$  is needed to fulfill these constraints. Hence, from now on only cubic Bézier curves with order  $m = 3$  are considered to keep the parametrization effort low. To include the gradient information of the phase space points from the initial Sobol plan  $\mathbf{P}_{\text{Sobol}}$  in the design of the Bézier curve, the scaling of the parameter  $\tau$  with

$$\tau = \frac{t - t_0}{\Delta t}$$

and the following derivative

$$\frac{d\tau}{dt} = \frac{1}{\Delta t}$$

must be considered with respect to the time  $t \in [t_0, t_{\text{End}}]$  and the length  $\Delta t = t_{\text{End}} - t_0$  of the trajectory. So, for a determined trajectory start point  $\mathbf{p}_i^T = (\mathbf{x}_{t_0}^T, \dot{\mathbf{x}}_{t_0}^T)$  the first control point results in

$$\mathbf{c}_1 = \mathbf{c}_0 + \left. \frac{d\mathbf{x}(\tau)}{3d\tau} \right|_{\tau=0} = \mathbf{x}_{t_0} + \frac{\dot{\mathbf{x}}_{t_0}}{3} \Delta t \quad (4.42)$$

with Equation (4.40) and  $\mathbf{c}_0 = \mathbf{x}_{t_0}$ . Analogously

$$\mathbf{c}_2 = \mathbf{c}_3 - \left. \frac{d\mathbf{x}(\tau)}{3d\tau} \right|_{\tau=1} = \mathbf{x}_{t_{\text{End}}} - \frac{\dot{\mathbf{x}}_{t_{\text{End}}}}{3} \Delta t \quad (4.43)$$

is obtained for the second-last control point with Equation (4.41),  $\mathbf{c}_3 = \mathbf{x}_{t_{\text{End}}}$ , and the end point  $\mathbf{p}_{i+1}^T = (\mathbf{x}_{t_{\text{End}}}^T, \dot{\mathbf{x}}_{t_{\text{End}}}^T)$  of the trajectory in the phase space. The remaining parameter for determining the cubic Bézier curve is the length  $\Delta t$ , which has to be chosen such that the gradient constraints of the phase space are satisfied. Therefore,  $\Delta t$  is defined as a rounded up multiple of the inverse sampling frequency to result in an equidistant discretization of the trajectory according to the time. Intending a fast exploration scheme,  $\Delta t$  is determined according to the following optimization problem

$$\begin{aligned} \Delta t &= \arg \min_{\Delta t \geq 0} (\Delta t) \\ \text{s.t.: } \dot{\mathbf{i}} &\preceq \dot{\mathbf{x}}(\tau, \Delta t) \preceq \dot{\mathbf{u}} \end{aligned} \quad (4.44)$$

with

$$\dot{\mathbf{x}}(\tau, \Delta t) = \frac{d\mathbf{x}(\tau, \Delta t)}{d\tau} \frac{1}{\Delta t} = \frac{6\tau(1-\tau)(\mathbf{x}_{t_{\text{End}}} - \mathbf{x}_{t_0})}{\Delta t} + \tau(3\tau - 2)\dot{\mathbf{x}}_{t_{\text{End}}} + (3\tau^2 - 4\tau + 1)\dot{\mathbf{x}}_{t_0}. \quad (4.45)$$

In contrast to Equation (4.39), the previous derived equation depends additionally on  $\Delta t$  caused by the control points  $\mathbf{c}_1$  and  $\mathbf{c}_2$ . The optimization task (4.44) is solved by increasing  $\Delta t$  iteratively until the gradient constraint is satisfied. The second derivative

$$\frac{d\dot{\mathbf{x}}(\tau, \Delta t)}{d\tau} = \frac{6(1-2\tau)(\mathbf{x}_{t_{\text{End}}} - \mathbf{x}_{t_0})}{\Delta t} + 2(3\tau - 1)\dot{\mathbf{x}}_{t_{\text{End}}} + 2(3\tau - 2)\dot{\mathbf{x}}_{t_0} \quad (4.46)$$

of the Beziér curve is helpful to calculate the values of  $\tau$  according to the points with highest and lowest derivatives for each input dimension. The extremal values of the first derivatives for the complete cubic Beziér curve are summarized in  $\dot{\mathbf{x}}_{\text{Max}}(\Delta t)$  for the maximal values and in  $\dot{\mathbf{x}}_{\text{Min}}(\Delta t)$  for the minimal values, respectively. Then, we can simply adapt

$$\Delta t_{\text{New}} = \max\left(\max(\text{diag}(\dot{\mathbf{i}})^{-1}\dot{\mathbf{x}}_{\text{Min}}(\Delta t_{\text{Old}})), \max(\text{diag}(\dot{\mathbf{u}})^{-1}\dot{\mathbf{x}}_{\text{Max}}(\Delta t_{\text{Old}}))\right)\Delta t_{\text{Old}} \quad (4.47)$$

until the constraint is fulfilled and no smaller  $\Delta t$  can be realized. More generally, if  $\mathbf{p}_i \neq \mathbf{p}_{i+1}$ , it holds true that  $\Delta t$  is strictly larger than zero and can be initialized with the inverse sampling frequency. Thus, for any start and end point of the trajectory in the confined phase space  $\mathbb{P}$  the restriction regarding the derivatives can always be satisfied with a cubic Beziér curve, but the bound for the input space can be violated. To check the input restrictions for a given cubic Beziér curve, i.e. with known control points according to the Equations (4.42) and (4.43) together with the length  $\Delta t$ , the in  $\tau$  quadratic equation

$$\left(\frac{6(\mathbf{x}_{t_0} - \mathbf{x}_{t_{\text{End}}})}{\Delta t} + 3\dot{\mathbf{x}}_{t_{\text{End}}} + 3\dot{\mathbf{x}}_{t_0}\right)\tau^2 + \left(\frac{6(\mathbf{x}_{t_{\text{End}}} - \mathbf{x}_{t_0})}{\Delta t} - 2\dot{\mathbf{x}}_{t_{\text{End}}} - 4\dot{\mathbf{x}}_{t_0}\right)\tau + \dot{\mathbf{x}}_{t_0} \stackrel{!}{=} \mathbf{0} \quad (4.48)$$

following from Equation (4.45) is solved for each input dimension and the resulting extreme values are compared to the given input bounds. If an input bound is violated, the trajectory is skipped and another randomly chosen end point  $\mathbf{p}_{i+1}$  from the initial Sobol design  $\mathbf{P}_{\text{Sobol}}$  in the phase space is considered. Thus, the transient exploring scheme ends if no more feasible trajectory to a Sobol point is available. The last trajectory is then induced by the cubic Beziér curve back to the stationary input point  $\mathbf{x}_0$ . Due to the order  $m = 3$  of the Beziér curve, the resulting trajectories are continuous in the phase space. More detailed, this approach results in continuous partial differentiable trajectories  $\mathbf{X}_{i+1}$  in the input space, which enables a favorable system stimulus caused by the smooth trajectory design. That is also advantageous compared to the often employed linear trajectory design with ramps, cf. Gutjahr (2012). Note that a ramp is a Bézier curve of first order. In Section 4.3.5 a comparison between our and a ramp-based transient exploration strategy is considered.

### 4.3.3 Determining Safe Trajectories

Before the model-based framework to assess the safety of trajectories  $\mathbf{X}_*$  induced by cubic Bézier curves as explained in the former section is introduced, it is necessary to think about the possible cases which can happen during transient exploration. Generally, for each point of the initial Sobol plan  $\mathbf{P}_{\text{Sobol}}$  in the given phase space, a return trajectory which leads the active learner safely back to the stationary starting point  $\mathbf{x}_0$  should exist. Even the existence of such a return trajectory is hard to determine, for example

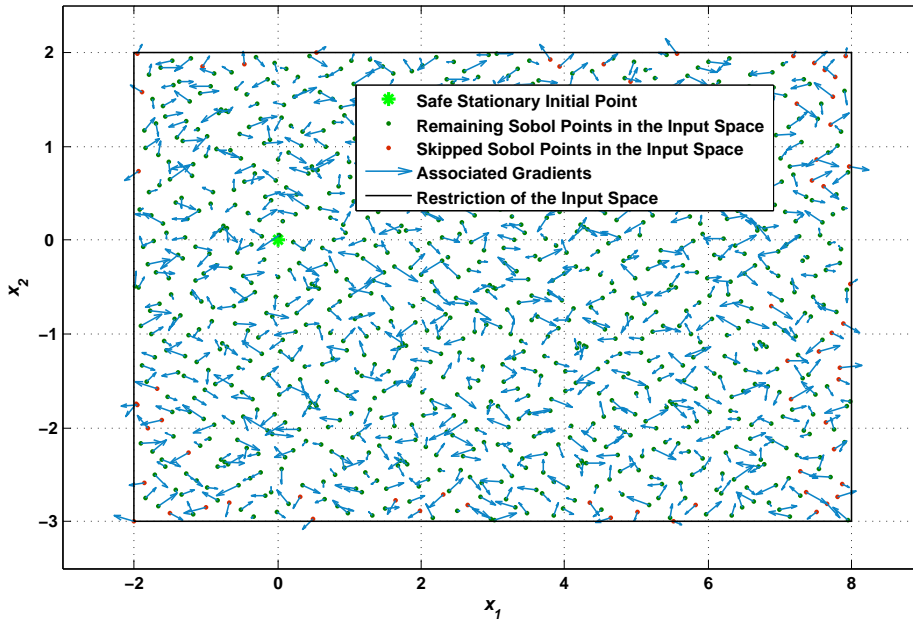


Figure 4.13: Visualization of a Sobol design with 1000 points for a bounded 4-dimensional phase space  $\mathbb{P}$  with a stationary starting point at the coordinate origin. The red Sobol points are previously skipped, since for these points does not exist a feasible and fast trajectory back to the safe initial point  $\mathbf{x}_0$ .

Moldovan and Abbeel (2012) showed for their MDP setting that this decision problem lies in NP. For our framework based on cubic Bézier curves, a return trajectory for each Sobol point  $\mathbf{p}_i$  is calculated before the beginning of the exploration, and if the bounds of the phase space are violated by this curve, the point  $\mathbf{p}_i$  is skipped and never considered again during exploration. Figure 4.13 illustrates this previous step according to the Sobol design from Figure 4.11. As can be seen, the phase space points near to the border do not necessary yield a feasible return trajectory induced by cubic Bézier curves. Besides that, there may exist a feasible return trajectory dependent on another parametrization or designed by an expert of the system, but our approach provides already a conservative exploration behavior. Nevertheless, to avoid unsafe trajectories during active exploration, a sparse DTC model is employed for transient learning of an additional system response transformed to a health function  $h : \mathbb{X} \rightarrow (-1, 1)$ , nearly equivalent to Section 4.2.3, which is an indicator for the current endangering of the physical system. For example, the exhaust temperature  $\vartheta(\mathbf{x})$  of a combustion engine from the interval  $[0^\circ\text{C}, 700^\circ\text{C}]$ , where temperatures over  $500^\circ\text{C}$  can damage the system, is transformed to

$$h(\mathbf{x}) = \text{sgn}(500^\circ\text{C} - \vartheta(\mathbf{x})) \left(1 - \frac{\vartheta(\mathbf{x})}{500^\circ\text{C}}\right)^2$$

induced by a quadratic cost function. Important is thereby the definition of the decision boundary at  $h(\mathbf{x}) = 0$ , which can be softly determined according to hard system constraints. In the transient setting, it is difficult for a human expert of the system to distinguish between very safe and unsafe points, or moreover trajectories through the input space. Thus, no discrete label information is available to design a customized discriminative model as in our stationary active learning framework in the previous Section 4.2.3. Hence, a transient regression model for  $h$  given by the DTC approximation with our efficient maximum error selection criterion is employed, cf. Section 3.2.3, together with the NARX approach

(2.45) to evaluate the safety of trajectories. Analogously to Equation (4.23), and for a test trajectory  $\mathbf{X}_* \in \mathbb{R}^{l \times d}$  the probability

$$\Pr_{\mathbf{q}}[\mathbf{h}_* \succeq \mathbf{0} \mid \mathbf{X}_*, \mathcal{D}_i, \boldsymbol{\theta}_g] = \int_{\mathbf{0}}^{\infty} \mathcal{N}(\mathbf{h}_* \mid \boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*) \partial \mathbf{h}_* \geq p \quad (4.49)$$

following from Equation (3.8) is obtained, where  $\mathcal{D}_i$  collects the sampled data up to the  $i$ -th measured trajectory to restrict the transient active learner to trajectories with a failure probability lower than  $1 - p$  with  $p \in (0, 1)$ . The hyperparameters of the DTC model summarized in  $\boldsymbol{\theta}_g$  and the NARX structure for the transient modeling task are assumed to be a priori known. Equivalent to the entropy in Equation (4.32), the limit of the probability (4.49) for a increasingly finer discretization of the trajectory  $\mathbf{X}_*$  depending on the continuous design with cubic Bézier curves is theoretically interesting. This issue is ignored for now, since the calculation of this probability for large trajectories, for example with  $l \gg 10$ , is already an exhausting task. Note that e.g.  $l = 100$  corresponds to a trajectory projection of only one second into the future for a sampling frequency of 100 Hz.

Methods based on numerical quadrature for solving the multidimensional integral (4.49) yield appropriate results until  $l \approx 25$  due to reasons of stability and efficiency, cf. Genz and Bretz (2002). Generally, a Monte Carlo based approximation of the probability (4.49) by acceptance-rejection sampling, formally

$$\Pr_{\mathbf{q}}[\mathbf{h}_* \succeq \mathbf{0} \mid \mathbf{X}_*, \mathcal{D}_i, \boldsymbol{\theta}_g] \approx \frac{1}{n_l} \sum_{k=1}^{n_l} \mathbb{I}(\mathbf{L}_* \mathbf{r}_k \succeq -\boldsymbol{\mu}_*) \quad (4.50)$$

with independent and identically distributed  $\mathbf{r}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  is always realizable, but mostly requires a high number of sampling steps  $n_l$  to reach an adequate accuracy. In particular, the lower Cholesky factor of the predictive covariance matrix  $\boldsymbol{\Sigma}_*$  is defined as  $\mathbf{L}_* \in \mathbb{R}^{l \times l}$  according to Equation (A.1). Hence, the complexity of the acceptance-rejection sampling approach lies in  $\mathcal{O}(n_l l^2)$ . To avoid the nearly exact and thus extensive calculation of the safety probability for the trajectory  $\mathbf{X}_*$ , it is possible to use the upper bound

$$\begin{aligned} \Pr_{\mathbf{q}}[\mathbf{h}_* \succeq \mathbf{0} \mid \mathbf{X}_*, \mathcal{D}_i, \boldsymbol{\theta}_g] &= \Pr_{\mathbf{q}}[\mathbf{h}_{* \cap J} \succeq \mathbf{0} \mid \mathbf{X}_{* \cap J}, \mathcal{D}_i, \boldsymbol{\theta}_g] \Pr_{\mathbf{q}}[\mathbf{h}_{* \setminus J} \succeq \mathbf{0} \mid \mathbf{h}_{* \cap J}, \mathbf{X}_*, \mathcal{D}_i, \boldsymbol{\theta}_g] \\ &\leq \Pr_{\mathbf{q}}[\mathbf{h}_{* \cap J} \succeq \mathbf{0} \mid \mathbf{X}_{* \cap J}, \mathcal{D}_i, \boldsymbol{\theta}_g] \end{aligned} \quad (4.51)$$

for an index set  $J \subset \{1, \dots, l\}$  to identify and skip unsafe trajectories. The size  $n_j < l$  of the index set  $J$  can be chosen arbitrary small to speed up the computation. However, the selection of the applied subset of trajectory points  $\mathbf{X}_{* \cap J}$  with respect to  $J$  is important to get a tight upper bound. Therefore, the heuristic

$$\begin{aligned} J &= \arg \min_{J \subset \{1, \dots, l\}, |J|=n_j} \left( \prod_{j \in J} \Pr_{\mathbf{q}}[h_{*,j} \geq 0 \mid \mathbf{x}_{*,j}, \mathcal{D}_i, \boldsymbol{\theta}_g] \right) \\ &= \arg \min_{J \subset \{1, \dots, l\}, |J|=n_j} \left( \prod_{j \in J} \Phi \left( \frac{\mu_{*,j}}{\sigma_{*,j}} \right) \right) \end{aligned} \quad (4.52)$$

is employed, because each individual probability provides an upper bound for the full probability (4.49) of the trajectory. Another way for calculating the full safety probability is by the principle of inclusion-exclusion which results in

$$\Pr_{\mathbf{q}}[\mathbf{h}_* \succeq \mathbf{0} \mid \mathbf{X}_*, \mathcal{D}_i, \boldsymbol{\theta}_g] = 1 + \sum_{k=1}^l \left( (-1)^k \sum_{J \subset \{1, \dots, l\}, |J|=k} \Pr_{\mathbf{q}}[\mathbf{h}_{* \cap J} \preceq \mathbf{0} \mid \mathbf{X}_{* \cap J}, \mathcal{D}_i, \boldsymbol{\theta}_g] \right) \quad (4.53)$$

as explained by Jukna (2011). Dependent on a bounded summation with  $n_j < l$ , this principle gives a lower bound if  $n_j$  is odd and an upper bound for even  $n_j$  of the true probability, respectively. Since the number of multidimensional probabilities which must be calculated increase strongly in  $n_l$ , this method will not be practicable for large trajectories. A fast approach for approximating the requested probability can be realized using a principle component analysis (PCA), cf. Bishop (2006), of the predictive covariance matrix that is  $\Sigma_* = \mathbf{U}\mathbf{E}\mathbf{U}^T$  with the diagonal matrix of eigenvalues  $\mathbf{E} = \text{diag}(\mathbf{e})$  sorted in descending order of  $\mathbf{e} \in \mathbb{R}^l$  and the orthogonal matrix of associated basis vectors  $\mathbf{U} \in \mathbb{R}^{l \times l}$ . Employing the representation  $\mathbf{h}_* - \boldsymbol{\mu}_* = \mathbf{U}\mathbf{z}$  gives the transformed integral

$$\Pr_{\text{q}}[\mathbf{h}_* \succeq \mathbf{0} \mid \mathbf{X}_*, \mathcal{D}_i, \boldsymbol{\theta}_g] = \int_{-\mathbf{U}^T \boldsymbol{\mu}_*}^{\infty} \mathcal{N}(\mathbf{z} \mid \mathbf{0}, \mathbf{E}) \partial \mathbf{z} \quad (4.54)$$

according to the principle components  $\mathbf{z} \in \mathbb{R}^l$ . The computation of the complete eigenvalue decomposition of  $\Sigma_*$  is too costly for a fast evaluation of the probability. Note also that the integral (4.54) does not generally factorize over the principle components, since the area of integration is no longer a rectangular area. Nevertheless, for a small subset of principle components according to the largest eigenvalues indexed by  $J$ , e.g. not more than three, it is possible to efficiently approximate the multidimensional integral (4.54), cf. Drezner (1994), which reduces to

$$\Pr_{\text{q}}[\mathbf{h}_* \succeq \mathbf{0} \mid \mathbf{X}_*, \mathcal{D}_i, \boldsymbol{\theta}_g] \approx \int_{-\mathbf{U}_J^T \boldsymbol{\mu}_*}^{\infty} \mathcal{N}(\mathbf{z}_J \mid \mathbf{0}, \mathbf{E}_{J,J}) \partial \mathbf{z}_J. \quad (4.55)$$

Since the largest eigenvalues can be calculated quickly with the power iteration, cf. Lipschutz and Lipson (2013), and solving the linear system of equations which describes the integration boundary is also effectively realizable, such that a mostly adequate estimate of the true probability can be obtained. Nevertheless, a final method for evaluating the probability (4.49) is presented which gives a favorable compromise between accuracy and computational speed. Therefore, it is necessary to rewrite the safety probability of a trajectory as an integral over non-normalized lower truncated Gaussian densities (A.28) resulting in

$$\Pr_{\text{q}}[\mathbf{h}_* \succeq \mathbf{0} \mid \mathbf{X}_*, \mathcal{D}_i, \boldsymbol{\theta}_g] = \int_{\mathbb{R}^l} \mathbb{I}(\mathbf{h}_* \succeq \mathbf{0}) \mathcal{N}(\mathbf{h}_* \mid \boldsymbol{\mu}_*, \Sigma_*) \partial \mathbf{h}_*. \quad (4.56)$$

This probability is equivalent to the marginal likelihood of a GP approach with model likelihood  $\mathbb{I}(\mathbf{h}_* \succeq \mathbf{0})$  and a prior induced by  $\mathcal{N}(\mathbf{h}_* \mid \boldsymbol{\mu}_*, \Sigma_*)$ . Due to the fact that exact inference is not tractable in this truncated GP model, we focus on an expectation propagation (EP) approach, cf. Minka (2001), since the Laplace approximation will not give adequate results caused by the discontinuity of the model likelihood. EP techniques are widely used for Bayesian inference in truncated GP models, for example by Toussaint (2009). In this special case, the algorithm from Herbrich (2005) is employed to determine an approximated marginal likelihood, namely the safety probability of a given trajectory. Hence, the goal is to infer a Gaussian posterior approximation with density

$$\mathcal{N}(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}) = \frac{1}{\hat{z}} \prod_{j=1}^l s_j \exp\left(-\frac{p_j (h_{*,j} - \eta_j)^2}{2}\right) \quad (4.57)$$

according to the moments  $\hat{\boldsymbol{\mu}} \in \mathbb{R}^l$  and  $\hat{\boldsymbol{\Sigma}} \in \mathbb{R}^{l \times l}$  with its lower Cholesky factor  $\hat{\mathbf{L}} \in \mathbb{R}^{l \times l}$ , cf. Equation

**Algorithm 4.2:** Gaussian EP for lower truncated Gaussians**Require:** Moments  $\boldsymbol{\mu}_*$ ,  $\boldsymbol{\Sigma}_*$  and a termination criterion**Ensure:** Approximation  $\hat{z}$  of the multidimensional Gaussian probability (4.49)

- 1:  $\mathbf{s} = \mathbf{1}$ ,  $\boldsymbol{\eta} = \mathbf{0}$ ,  $\mathbf{p} = \mathbf{0}$ ,  $\hat{\boldsymbol{\mu}} = \boldsymbol{\mu}_*$ ,  $\hat{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}_*$
- 2: **repeat**
- 3:   Pick  $j \in \{1, \dots, l\}$
- 4:    $\xi = p_j \hat{\boldsymbol{\Sigma}}_{jj}$
- 5:    $\zeta = (1 - \xi)^{-1}$
- 6:    $\phi = \hat{\mu}_j + \xi \zeta (\hat{\mu}_j - \eta_j)$
- 7:    $\psi = \zeta \hat{\boldsymbol{\Sigma}}_{jj}$
- 8:    $\hat{\phi} = \phi \psi^{-\frac{1}{2}}$
- 9:    $\alpha = \mathcal{N}(\hat{\phi}) \Phi(\hat{\phi})^{-1}$
- 10:    $\beta = \alpha(\alpha + \hat{\phi}) \psi^{-1}$
- 11:    $\alpha = \alpha \psi^{-\frac{1}{2}}$
- 12:    $\hat{\boldsymbol{\mu}} = \hat{\boldsymbol{\mu}} + \zeta (p_j (\hat{\mu}_j - \eta_j) + \alpha) \hat{\boldsymbol{\Sigma}}_j$
- 13:    $\hat{\boldsymbol{\Sigma}} = \hat{\boldsymbol{\Sigma}} + \zeta (p_j - \zeta \beta) \hat{\boldsymbol{\Sigma}}_j \hat{\boldsymbol{\Sigma}}_j^T$
- 14:    $p_j = (\beta^{-1} - \psi)^{-1}$
- 15:    $\eta_j = (\alpha + \phi \psi^{-1})^{-1}$
- 16:    $s_j = \Phi(\hat{\phi}) \exp(2^{-1} \alpha \eta_j) (1 - \psi \beta)^{-\frac{1}{2}}$
- 17:    $\eta_j = \eta_j + \phi$
- 18: **until** Termination criterion is fulfilled
- 19: Determine Cholesky factors  $\mathbf{L}_*$  and  $\hat{\mathbf{L}}$  with respect to Equation (A.1)
- 20: Calculate  $\hat{z}$  according to Equation (4.58)

(A.1), and the parameter vectors  $\mathbf{s} \in \mathbb{R}^l$ ,  $\mathbf{p} \in \mathbb{R}^l$ , and  $\boldsymbol{\eta} \in \mathbb{R}^l$ . The normalization constant

$$\hat{z} = \text{prod}(\mathbf{s}) |\mathbf{L}_*^{-1} \hat{\mathbf{L}}| \exp\left(-\frac{1}{2} \left(\mathbf{p}^T (\boldsymbol{\eta} \circ \boldsymbol{\eta}) + \|\mathbf{L}_*^{-1} \boldsymbol{\mu}_*\|^2 - \|\hat{\mathbf{L}}^{-1} \hat{\boldsymbol{\mu}}\|^2\right)\right) \quad (4.58)$$

is of main interest, since  $\hat{z} \approx \Pr_{\mathbf{q}}[\mathbf{h}_* \succeq \mathbf{0} \mid \mathbf{X}_*, \mathcal{D}_i, \boldsymbol{\theta}_g]$ . The estimation of the parameter vectors to obtain the moments of the approximated Gaussian posterior, and thus the approximated probability  $\hat{z}$  is described in Algorithm 4.2. Compared to Herbrich (2005), the proposed algorithm is slightly adapted to increase numerical stability during the rank one updates. Specifically, several scalar variables like  $\xi$  and  $\zeta$  are introduced as shorthand notations and  $\hat{\boldsymbol{\Sigma}}_j$  describes the  $j$ -th column vector of the covariance matrix  $\hat{\boldsymbol{\Sigma}}$ . Our scheme for approximating the safety probability (4.49) is nearly similar to the approach by Cunningham et al. (2011) in terms of accuracy and computational efficiency. For the ordering of the iteratively selected indices  $j \in \{1, \dots, l\}$  in Algorithm 4.2, the same strategy as in Equation (4.52) is employed to reach a fast convergence of the approximated marginal probability  $\hat{z}$ . As termination criterion, the maximal number of iterations over the whole length  $l$  of the trajectory is used. Additionally, it is possible to introduce a more sophisticated criterion, for example, dependent on the change of the marginal probability to speed up the Gaussian EP algorithm for determining  $\hat{z}$ .

In the former paragraph, only the calculation of the multivariate probability (4.49) with different methods is considered, but it is also possible to optimize a trajectory  $\mathbf{X}_*$  which gives a certain probability level

$p$ . For that, the prediction interval

$$\boldsymbol{\mu}_*^T \boldsymbol{\Sigma}_*^{-1} \boldsymbol{\mu}_* \geq \chi_l^2(p) \quad (4.59)$$

is employed with the quantile of the chi-squared distribution  $\chi_l^2(p)$  according to the  $l$  degrees of freedom determined by the length of the trajectory. This inequality states that the vector  $\mathbf{0} \in \mathbb{R}^l$  should lie outside of the multidimensional ellipsoid described by  $\boldsymbol{\mu}_*^T \boldsymbol{\Sigma}_*^{-1} \boldsymbol{\mu}_*$  with probability  $p$ . Hence, for a given probability level  $p$  and a parametrized and differentiable trajectory  $\mathbf{X}_*$ , e.g. as in the previously section, the gradients with respect to the trajectory parameters can be computed and optimized to fulfill the constraint (4.59). Analogously to the stationary active learning approach in Equation (4.26), the inclusion of the entropy criterion according to Equation (4.32) enables a favorable active learning framework. Nevertheless, the resulting optimization scheme has to deal with some challenges. Firstly, the length  $l$  of a trajectory depends on the parameters of the trajectory, and thus also on the restrictions of the phase space  $\mathbb{P}$ . Remember the time dependency of the planned trajectory considered in the transient case. Another requirement is the efficient online realization of the gradient calculations according to the trajectory parameters to get nearly information-optimal and safe trajectories. Moreover, it is questionable whether the given trajectory parametrization is flexible enough to always reach a predefined safety level, since the existence of safe trajectories is an important issue. Not all of this challenges are considered in this thesis, but the main issues for safe active learning in the transient setting are addressed in the following section presenting our algorithm.

### 4.3.4 The Algorithm

The herein presented active learning algorithm summarizes the main requirements of our safe transient exploration framework as introduced in the former sections. As explained, the given stationary and safe input point  $\mathbf{x}_0 \in \mathbb{X}$  provides the basis for exploration. The bounds of the confined input space  $\mathbb{X} \subset \mathbb{R}^d$  and the respective gradient space  $\dot{\mathbb{X}} \subset \mathbb{R}^d$ , which summarizes the restrictions of the associated phase space  $\mathbb{P}$  as presented in Section 4.3.1, are also assumed to be given. The corresponding phase space  $\mathbb{P}$  allows for a low-discrepancy design  $\mathbf{P}_{\text{Sobol}}$  given by a Sobol sequence of size  $n$  to be generated to cover the whole phase space well, cf. Figure 4.11. For the smooth point to point trajectory planning in the phase space based on  $\mathbf{P}_{\text{Sobol}}$ , the approach with cubic Bézier curves as shown in Section 4.3.2 is employed. Depending on that novel planning approach, the Sobol points from  $\mathbf{P}_{\text{Sobol}}$  which do not lead to a safe backward trajectory are deleted from  $\mathbf{P}_{\text{Sobol}}$  as illustrated in Figure 4.13, Section 4.3.3. For the sake of simplicity and regarding the implementation of our transient active learning scheme the maximization with respect to the differential entropy (4.32) of a trajectory is neglected. Thus, we focus only on the slightly more important safety issue when exploring the transient environment. Nevertheless, the exploration behavior is guaranteed by the favorable properties of the Sobol plan. For a good coverage of the considered phase space  $\mathbb{P}$ , the number of Sobol points should be sufficiently high. Since this number is not easy to obtain, a possible larger  $n$  than necessary is recommended while motivating some other termination criteria of our active learning algorithm. For example, our transient exploring strategy can determine if a time threshold for the whole measurement duration is reached. To evaluate safe trajectories, an additional function  $h$  depending on some system responses that indicates the health status of the dynamic system is required. The transient modeling of  $h$  is realized with the sparse DTC approach under the maximum

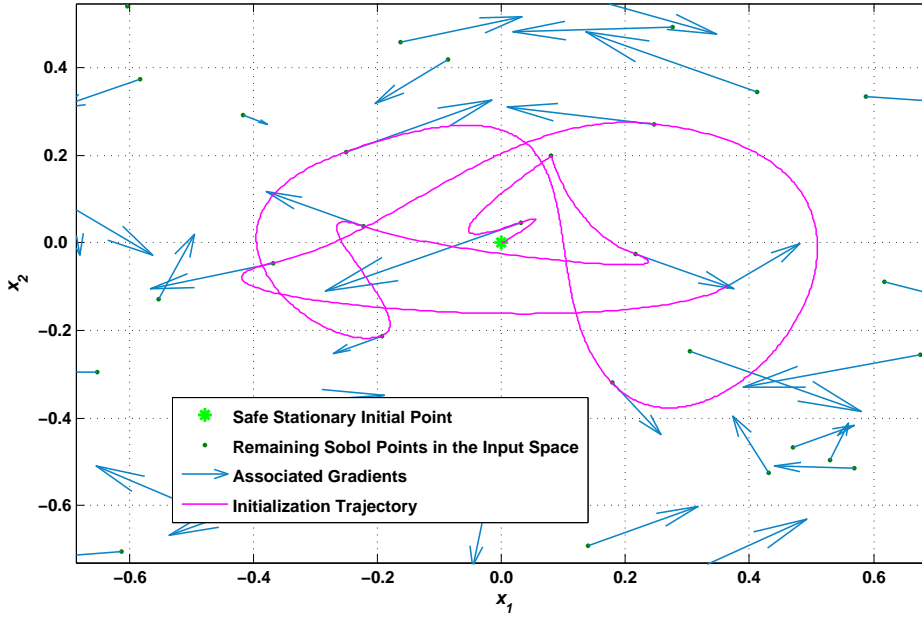


Figure 4.14: First path of trajectories according to the Sobol design from Figure 4.11 to initialize the transient DTC model of our active exploring strategy. Thereby, the nearest Sobol points with respect to the safe initial point in the input space are used.

error insertion criterion as presented in Section 4.3.3 and Chapter 3 for more algorithmic details. Hence, the safety probability of a trajectory is assessed with Equation (4.49) and approximately calculated with the Gaussian EP Algorithm 4.2 based on the transient DTC model predictions (3.8) for a previously determined NARX structure, see Section 2.3. The hyperparameters  $\theta_g$  for the DTC regression approach are also a priori defined.

Our transient exploring strategy is presented in Algorithm 4.3, starting from the safe initial point  $\mathbf{x}_0$  with a zero gradient. A small subset of size  $n_0 < n$  of the phase space points  $\mathbf{P}_{\text{Sobol}}$  is used to initialize the transient DTC model. This subset and their query order is chosen from the Sobol points with minimal Euclidean distance to the safe initial point with respect to the input space  $\mathbb{X}$ . Once this path of trajectories between the points of the chosen subset is determined, the path is extended with the trajectory  $\mathbf{X}_b$  back to the safe initial point and finally queried. Note that the predefined subset size  $n_0$  may depend on the previously determined NARX structure. In Figure 4.14 an example for such an initialization path is shown. This illustration is related to the initial phase space DoE from Figure 4.13. Remember that the phase space points from  $\mathbf{P}_{\text{Sobol}}$  without a feasible backward trajectory to the safe initial point  $\mathbf{x}_0$  were already skipped. With the thereby received data, leading to the transformed system responses  $\mathbf{h}$ , the transient DTC model for safety evaluations of the following trajectories is initialized and trained. After these initialization steps the iteration over the remaining Sobol points is started. To introduce a more global safety character and to avoid dead ends during the iterative path planning, a tree  $T$  of feasible trajectories up to a previously defined depth is generated. This approach compares to the forward-looking path planning idea from Section 4.3.1, where also a tree of trajectories is created to improve the overall exploration behavior. In this thesis, the depth of the binary search tree for trajectory selection according to their safety is set to four, where the last queried Sobol point always builds the root of the binary tree  $T_4$ .



**Algorithm 4.3:** Transient Safe Active Learning with GPs**Require:**  $\mathbf{x}_0, \mathbf{l}, \mathbf{u}, \dot{\mathbf{l}}, \dot{\mathbf{u}}, n, n_0, \boldsymbol{\theta}_g, p$ **Ensure:** Safe transient input design  $\mathbf{X}$  contained in  $\mathcal{D}$ 

- 1: Create initial Sobol design  $\mathbf{P}_{\text{Sobol}} \subset \mathbb{P}$  of size  $n$
- 2: Remove phase space points from  $\mathbf{P}_{\text{Sobol}}$  without feasible backward trajectories
- 3: Generate the trajectories  $\mathbf{X}_1, \dots, \mathbf{X}_{n_0}, \mathbf{X}_b$  from a small subset of  $\mathbf{P}_{\text{Sobol}}$  around  $\mathbf{x}_0$
- 4: Query the system with these trajectories and sample the transformed responses  $\mathbf{h}_1, \dots, \mathbf{h}_{n_0}, \mathbf{h}_b$
- 5: Initialize and train the transient DTC model on  $\mathcal{D} = (\mathbf{h}_1, \dots, \mathbf{h}_0, \mathbf{h}_b, \mathbf{X}_1, \dots, \mathbf{X}_{n_0}, \mathbf{X}_b)$
- 6: **while**  $|\mathbf{P}_{\text{Sobol}}| > 0$  **do**
- 7:     Build a binary search tree  $T_4$  of feasible trajectories with the remaining Sobol points
- 8:     Calculate the DTC model predictions for all paths resulting from  $T_4$
- 9:     Evaluate the safety probability of all paths with Algorithm 4.2
- 10:     Select the first trajectory  $\mathbf{X}_*$  of the path with the highest safety probability
- 11:     Create the associated backward trajectory  $\mathbf{X}_b$  to  $\mathbf{x}_0$  for the selected trajectory  $\mathbf{X}_*$
- 12:     Calculate the DTC model predictions for the combined path consisting of  $\mathbf{X}_*$  and  $\mathbf{X}_b$
- 13:     **if** The with Algorithm 4.2 evaluated safety probability  $p_* > p$  for the combined path **then**
- 14:         Remove the related Sobol point from  $\mathbf{P}_{\text{Sobol}}$
- 15:     **else**
- 16:         **while**  $p_* \leq p$  and  $|\mathbf{X}_*| > 0$  **do**
- 17:             Point-wise shortening of  $\mathbf{X}_*$  and combining with the needed backward trajectory  $\mathbf{X}_b$
- 18:             Calculate the DTC model predictions for the combined path consisting of  $\mathbf{X}_*$  and  $\mathbf{X}_b$
- 19:             Evaluate the safety probability  $p_*$  for the created combined path with Algorithm 4.2
- 20:         **end while**
- 21:         Set the trajectory  $\mathbf{X}_* = \mathbf{X}_* \cup \mathbf{X}_b$  to go back to the safe initial point
- 22:         Move the related Sobol point from  $\mathbf{P}_{\text{Sobol}}$  to the remaining point set  $\mathbf{P}_R$
- 23:     **end if**
- 24:     Query the system with  $\mathbf{X}_*$  and sample the transformed response  $\mathbf{h}_*$  to update  $\mathcal{D}$
- 25:     Train the transient DTC model on the updated data set  $\mathcal{D}$
- 26: **end while**
- 27: **if** Safe initial point  $\mathbf{x}_0$  not already reached **then**
- 28:     Generate a backward trajectory  $\mathbf{X}_b$  to  $\mathbf{x}_0$  and query the system to sample new data
- 29:     Add the queried and transformed data to  $\mathcal{D}$  and train the transient DTC model with  $\mathcal{D}$
- 30: **end if**
- 31: **while**  $|\mathbf{P}_R| > 0$  **do**
- 32:     Create  $\mathbf{X}_*$  to a randomly chosen point from  $\mathbf{P}_R$  together with its backward trajectory  $\mathbf{X}_b$
- 33:     Calculate the DTC model predictions for the combined path consisting of  $\mathbf{X}_*$  and  $\mathbf{X}_b$
- 34:     Evaluate the safety probability  $p_*$  for the created combined path with Algorithm 4.2
- 35:     **if** The combined trajectory is feasible and  $p_* > p$  **then**
- 36:         Query the system with the combined trajectory and sample the new data to update  $\mathcal{D}$
- 37:         Train the transient DTC model on the updated data set  $\mathcal{D}$
- 38:     **end if**
- 39:     Remove the related Sobol point from  $\mathbf{P}_R$
- 40: **end while**

The subsequently selected Sobol points for the knots of the tree were randomly chosen from the remaining ones, but only the Sobol points which lead to feasible trajectories are included. If no more Sobol point is valid with respect to the resulting trajectory during the search tree creation process, the safe initial point  $\mathbf{x}_0$  is allocated to this leave knot where the current path will end. Thus, the final binary search tree  $T_4$  results in maximal sixteen different paths induced by the specified depth equal to four, where one path consists of four connected trajectories. Then, for all these sixteen trajectories and under consideration of the former specified NARX structure the probability according to Equation (4.49) is estimated with Algorithm 4.2 as presented in Section 4.3.3. Choosing from these paths, the path of connected trajectories with the highest probability is selected for further investigations. Due to the high impact of newly queried dynamic data into our sparse GP modeling approach, only the first trajectory of the selected path of trajectories will be queried to provide a nearly up-to-date model for safety evaluations. Theoretically, for a much more accurate safety calculation it would be better to query the determined trajectory point-wise, update with the obtained data our sparse GP model, and recalculate the determined trajectory under the new belief in real-time. But that would not be computational realizable for a given sampling frequency. Hence, a complete trajectory is queried to avoid the computational overhead by the point-wise model updates with the new sampled system outputs and responses. Sure, the sparse model will lose some belief, since the new queried data is not included immediately, but the gain in efficiency is much higher. Additionally, before the final querying run of the dynamic system, the probability of this first trajectory  $\mathbf{X}_*$  extended with the backward trajectory  $\mathbf{X}_b$  to the safe initial point  $\mathbf{x}_0$  is calculated with Algorithm 4.2 according to Equation (4.49). Finally, the trajectory is queried if, and only if the predicted probability is greater or equal than a certain threshold  $p$ . If this condition holds true, the trajectory is queried and the associated Sobol point is removed from the set  $\mathbf{P}_{\text{Sobol}}$ . The reason for this extension is to ensure with a high probability  $p$  that the selected Sobol point is reached and a safe backward trajectory exists. This step is inspired by the work of Moldovan and Abbeel (2012). Now, the crucial point in our exploring strategy is to define what should happen if this probability constraint induced by the threshold value  $p$  is not fulfilled. Due to the random search tree generation it may happen, that the next selected Sobol point will lead to a large trajectory across the whole phase space  $\mathbb{P}$ , for example. In such or a similar case the probability constraint can fail. Thus, to avoid an early determination of the presented algorithm and to explore as much as possible from the system with the already calculated trajectory, this trajectory is point-wise shortened and extended with a backward trajectory to the safe initial point  $\mathbf{x}_0$ . Subsequently, for each shortened and extended trajectory the safety probability is calculated with Algorithm 4.2 as well as the corresponding required predictions. If for one of these extended trajectories the probability constraint depending on  $p$  is fulfilled, the longest trajectory of them is used to query the system. This trajectory leads the algorithm back to  $\mathbf{x}_0$ . Since the Sobol point which induced the trajectory was not reached, the point is moved from  $\mathbf{P}_{\text{Sobol}}$  to an remaining set of Sobol points  $\mathbf{P}_R$  to consider this point later in the Algorithm 4.3 again. Otherwise, i.e. if no shortened and with a backward path extended trajectory fulfills the probability constraint, the associated Sobol point is skipped and also moved to  $\mathbf{P}_R$ . Furthermore, a direct backward trajectory to the safe initial point is calculated and queried. After querying one trajectory, independent if it leads back our exploring scheme to  $\mathbf{x}_0$  or to a new Sobol point, the sampled data is used to update the sparse GP model. Once the model is retrained, the next iteration over  $\mathbf{P}_{\text{Sobol}}$  is started as long as this Sobol point set is not empty. When  $\mathbf{P}_{\text{Sobol}}$  is empty, a backward trajectory to the safe initial point  $\mathbf{x}_0$  is determined and queried, if  $\mathbf{x}_0$  is not already reached. Now, the main part of our transient safe active learning scheme is described.

Furthermore, the set  $\mathbf{P}_R$  may contain some skipped Sobol points due to the special handling of nearly unsafe trajectories filtered out with the probability constraint. Now, it makes a lot of sense to consider these points again since the belief of our sparse GP model could have significantly changed compared to the learning status when the Sobol points were moved to  $\mathbf{P}_R$ . Thus, the last part of our algorithm randomly iterates over the skipped Sobol point set  $\mathbf{P}_R$ . Starting from the safe initial point, a trajectory to the selected skipped Sobol point is generated together with its backward trajectory and checked, whether the merger of both trajectories exceeds the phase space limits. If the phase space limits are exceeded, the trajectory is not queried and the associated Sobol point is removed from  $\mathbf{P}_R$  and never considered again. In case of a valid trajectory the model based predictions for the safety evaluation of the merged trajectory are analogously calculated with Algorithm 4.2 as in the main part of our transient exploring framework. The complete trajectory is queried if the probability constraint holds true for the determined connected trajectory. With the thereby obtained data the sparse GP model is updated and retrained. For the case that the probability constraint fails for this trajectory, the system is not queried and the related Sobol point is directly deleted from  $\mathbf{P}_R$ . Until the skipped Sobol point set  $\mathbf{P}_R$  is not empty, the above steps are subsequently repeated. Note that the safe initial point  $\mathbf{x}_0$  is always the start as well as the end point of this last iteration in our learning algorithm.

The computational complexity of our safe transient exploring scheme presented in Algorithm 4.3 is dominated by the update and training of the sparse DTC model. For this reason the efficient maximum error criterion for the DTC approximation as shown in Chapter 3 is used for the sequential model updates. In this case the number of active training points is slightly increased with the growing data set  $\mathcal{D}$  to provide a sufficient modeling result. An upper bound for the active set size is used to estimate the maximal needed computational effort per iteration, and thus the overall amount of computing resources to enable our presented active learning framework. More details on incremental updating and training of the sparse GP model are given in Section 3.2 and the references therein. Furthermore, it is necessary to address the issue when, despite our safe learning framework with the DTC approximation, the limits of the physical system are exceeded due to the modeling noise or system disturbances. In these cases, it is assumed that the interface to the system consists of a monitoring layer which recognizes such dangerous situations and performs a query intervention to lead the system back to a safe state. This safe state is provided by the safe initial point  $\mathbf{x}_0$ . How the currently queried trajectory is interrupted and the backward trajectory is designed depends on the system and the so far obtained knowledge about it. Usually a simple ramp trajectory is used to bring the system state back to a moderate state as fast as possible and under consideration of the gradient restrictions of the associated phase space. In the next section, an example is shown which captures this and other interesting topics which can appear during active interventions with physical systems.

### 4.3.5 Evaluations

For the evaluation of the above presented active learning scheme, a well known physical system from the Bosch domain is chosen. Mainly, an electromagnetic valve as shown in Figure 4.15 is chosen for the safe exploration of the system behavior to demonstrate the performance of Algorithm 4.3. The one dimensional

input of the electromagnetic valve is given by the voltage of the coil. The safe initial point for the coil voltage is set to  $x_0 = 1 V$  with a zero gradient, where the complete input space  $\mathbb{X}$  is determined by  $l = 0 V$  and  $u = 20 V$ . The associated gradient space  $\dot{\mathbb{X}}$  is restricted by the lower bound  $\dot{l} = -400 kV/s$  and the upper bound  $\dot{u} = 400 kV/s$ . As system response for modeling the health status of the valve its electromagnetic force  $F$  is used, since a force too high can damage the spring between the core and the anchor.

The coil voltage dependent force  $F$  is employed to define a noisy health function for the electromagnetic valve given by  $h(F) = \varepsilon - \exp(F - 5 N)$  with the noisy Gaussian random variable  $\varepsilon \sim N(1, 10^{-4})$  and where  $h(F)$  is related to an exponential cost function. This type of cost function is chosen because of its high gradient around the decision boundary as compared to an quadratic cost function, for example. Hence, this definition of  $h$  should ensure forces below  $5 N$  with a high probability during exploration.

As known from simulation results, the stationary border of the dynamic valve system lies around  $11.7 V$ . That means, that the maximal allowed force of  $5 N$  is reach at this limit with a stationary system stimulation. Overall, a sampling frequency of  $1 MHz$  is needed to capture the system behavior sufficiently.

Thus, this system is not suitable to demonstrate the real-time capability of our exploring scheme, which is not the goal of this evaluation. Hence, it will be shown that our exploring approach provides a good coverage of the related phase space with a high safety induced by the transient GP modeling framework. Due to this fact, all of the evaluations are performed on simulations of the electromagnetic valve as described and provided by Albunni (2010). Specifically, the model order reduction algorithm from Albunni is used for the following experiments with standard parameters for the geometry of the valve as well as for its core parameters such as a resistance parameter of  $2 \Omega$  and 70 windings for the coil. As explained in Algorithm 4.3, the transient modeling of the above defined health function  $h$  is realized with the sparse DTC approach under the maximum error insertion criterion with always 100 active training points. In each employed model training step of Algorithm 4.3 the active subset is completely selected from the beginning. But the simulation results are still obtained in a fast manner due to the computational efficiency of our employed DTC framework. The basis for this sparse DTC model is given by the determined NARX(1,2) structure. This structure is chosen to yield good modeling results under stable model predictions, which is realized with the two recurrences of the input voltage. Additionally, the hyperparameters  $\theta_g$  of the DTC model were previously estimated with the help of the obtained training data from a system stimulation of the considered valve with 20 test trajectories as equivalently used in the experiments by Albunni (2010). For the case that our exploring scheme exceeds the border of the restricted force  $F$ , a backward strategy is defined which leads

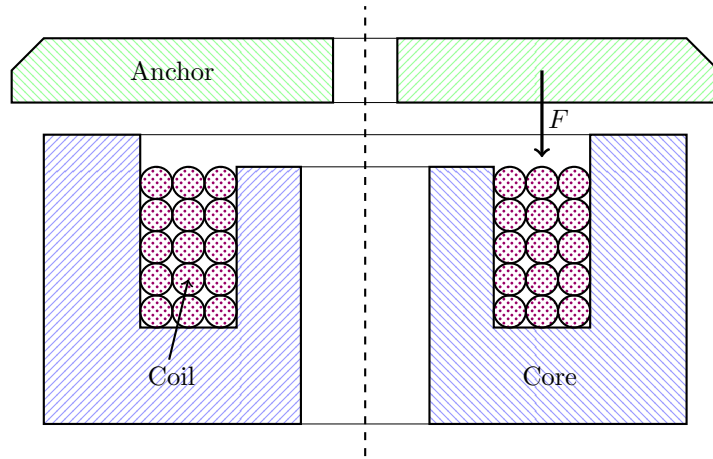


Figure 4.15: Vertical cross section through a simple electromagnetic valve consisting of a coil surrounded by a magnetic core and a movable anchor. Note that the mechanical spring between the core and the anchor is not shown in the rotationally symmetric illustration to provide a better visibility.

the system as fast as possible back to the safe initial point  $x_0 = 1 V$ . Therefore, the currently queried cubic Beziér curve is interrupted and immediately replaced by a ramp trajectory with a maximal gradient to bring the electromagnetic valve system quickly back into a safe state. After that, the transient DTC model is updated with the currently queried data and our Algorithm 4.3 is continued. To reduce the probability of such possible system interruptions during querying, the probability level  $p = 0.50$  is used in all our experiments. Also a Sobol design  $\mathbf{P}_{\text{Sobol}}$  with  $n = 100$  points in the associated phase space  $\mathbb{P}$ , as described in the beginning of this section, is employed. Out of these,  $n_0 = 20$  phase space points around the safe initial point  $x_0$  are selected to initialize our exploring framework. Moreover, since the physical system stimulation requires a starting point at  $0 V$ , the first initial trajectory realizes the step from  $0 V$  to the safe initial point at  $1 V$ . Note that all points from  $\mathbf{P}_{\text{Sobol}}$  are removed in the beginning of Algorithm 4.3 which do not result in a valid backward trajectory.

In the first experiment, our presented active learning Algorithm 4.3 based on the input trajectory planning with cubic Beziér curves is evaluated on the electromagnetic valve system from Figure 4.15 with the settings described above. In Figure 4.16 the stimulation of the voltage is shown. Here, the dashed green line indicates the stationary border of the dynamic valve system. After the short initialization phase, Figure 4.16 illustrates that our safe dynamic system stimulation is able to cover input regions outside of the stationary borders which enables a desirable coverage of the whole input space. The resulting force trajectory of the queried voltage path is presented in Figure 4.17. The red curve in this graphic shows the discriminative function which is learned with the employed sparse Gaussian process model. During the initialization phase in the first  $0.15 ms$ , the sparse DTC model for the discriminative function is adequately learned, such that the exploration of the non-stationary input area is immediately started, cf. Figure 4.16. In the meanwhile, the load on the electromagnetic valve is slightly increasing without

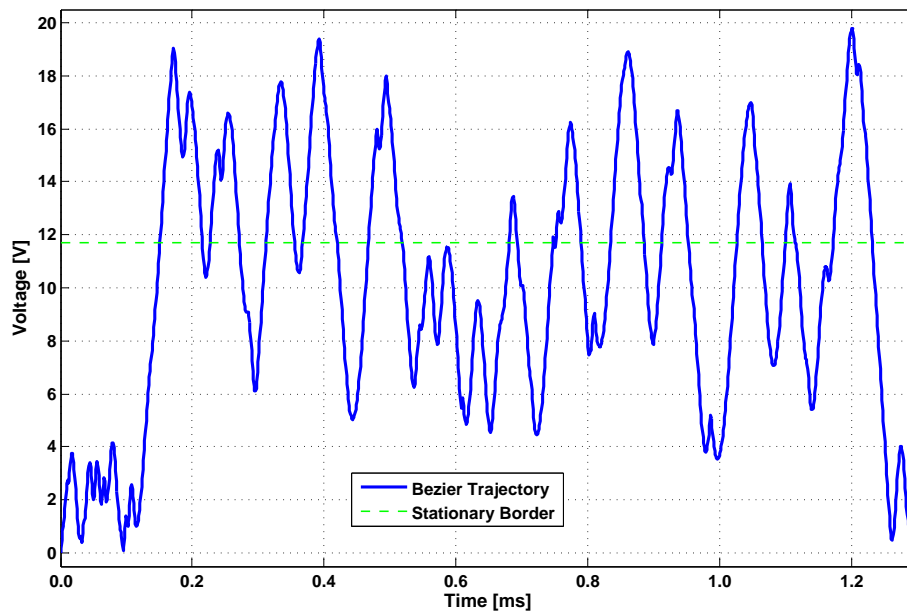


Figure 4.16: Stimulation of the voltage of the simulated electromagnetic valve system as described in Figure 4.15 with our active learning framework based on cubic Beziér curves which is summarized in Algorithm 4.3.

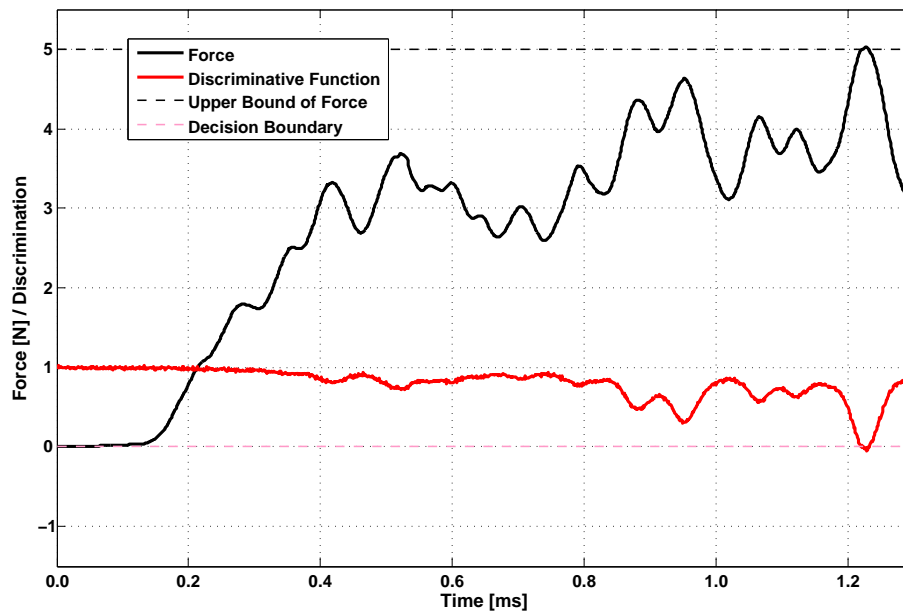


Figure 4.17: Resulting force of the queried voltage trajectory from Figure 4.16 on the simulated electromagnetic valve system as described in Figure 4.15 with our active learning framework based on cubic Beziér curves which is summarized in Algorithm 4.3.

damaging this system which is indicated by the trend of the force and the related discriminative function. However, in the end of the exploration phase after the penultimate available Sobol point was queried, the backward trajectory to the safe initial point, which was the only possible choice of our safe active

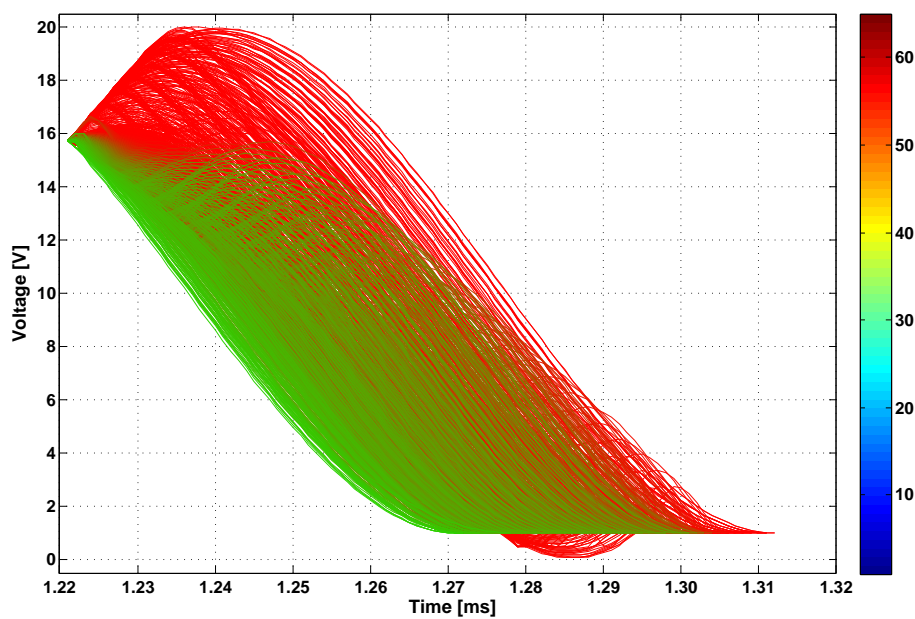


Figure 4.18: Predicted probabilities of failure (in percent) for the cubic Beziér trajectories to all Sobol points of the initial plan, where each of them is extended with its backward trajectory, after exploring the penultimate Sobol point in the electromagnetic valve example.

learner, leads to a small violation of the defined upper bound of the force. This interruption is in the region of the defined noise level on the modeled discriminative function. To understand this issue and our whole exploring framework better, Figure 4.18 shows the predicted probabilities of failure ( $1 - p_*$ ) for the trajectories  $\mathbf{X}_*$  to all Sobol points of the initial plan, where each of them is extended with its backward trajectory. Therein, it is illustrated that all extended trajectories which go as fast as possible towards the safe initial point have a much smaller probability of failure, indicated by the green coloring, than the other ones. Note that the actually queried trajectory including its backward path yields only a probability of failure of around 20% before acting with the system and updating the sparse DTC model with the newly sampled data as used for the calculations in Figure 4.18. This visualization indicates a good accuracy of the transient sparse Gaussian modeling approach for the safety evaluations of the trajectories. Overall, the smoothness of the Beziér curves induce smart trajectories in the associated phase space  $\mathbb{P}$  which enable an adequate stimulation of the considered dynamic system. Additionally, this trajectory design supports perfectly the data-based modeling with the underlying NARX(1, 2) structure. In this exploration nearly all valid points, i.e. 97 out of  $n = 100$  points from the initial Sobol design, were queried. In the end, the exploration finished with only one skipped Sobol point in the remaining set  $\mathbf{P}_R$ . But even the renewed inspection of this Sobol point in the last phase of our Algorithm 4.3 has not led to a querying. The reason for that is the proximity of this point to the upper bound of the input space and its large gradient into an unsafe direction.

In the second scenario our previously employed safe active learning framework is slightly adapted to enable a ramp-based trajectory design. In doing so, the gradient of a selected Sobol point is used to determine the linear ramps. Only the sign of the components of the gradient is determined according to the current and the next Sobol point due to the less flexible ramp trajectories. Note that a ramp trajectory is a Beziér curve of first order. For the initialization phase the same Sobol point set as in the former evaluation is employed. After the initialization phase the safe initial point is set to zero, i.e.  $\mathbf{x}_0 = 0$ . This is done to provide an easier trajectory design since all trajectories are valid due to the ramp definition. In Figure 4.19 the complete voltage trajectory resulting from interaction with the electromagnetic valve simulation is shown. Therein, the dashed green line describes the stationary border as in the similar picture as shown before. The main difference to the previous trajectory design with cubic Beziér curves is the well-visible roughness of the ramp trajectories. Furthermore, the time span needed for querying the same number of Sobol points as in Figure 4.16 increased to 7 ms. Thus, the number of data points is grown by a factor of around five which intensify the modeling effort. This is a disadvantage regarding the coverage of the associated phase space  $\mathbb{P}$  as is shown in this section below. Considering the force trajectory of the ramp-based trajectory design presented in Figure 4.20, it is remarkable that shortly after the initialization phase the upper limit of the force boundary has been exceeded. Generally, the less smooth ramp trajectories induce a jagged discriminative function characteristic which results from the thereby sampled force values of the electromagnetic valve simulation. Furthermore, the discriminative function shows an oscillating behavior due to the alternated querying of new trajectories followed by their backward trajectories. This behavior of the exploration and modeling follows from the ramp-based input design, especially in the end of our active planning strategy where many of the remaining Sobol points are considered again, cf. Figure 4.19. Compared to our previous example with cubic Beziér curves, all Sobol points were queried during the evaluation. Hence, most of the critical Sobol points, i.e. the points outside of the stationary valid area, are queried in the end of our Algorithm 4.3 after they have been

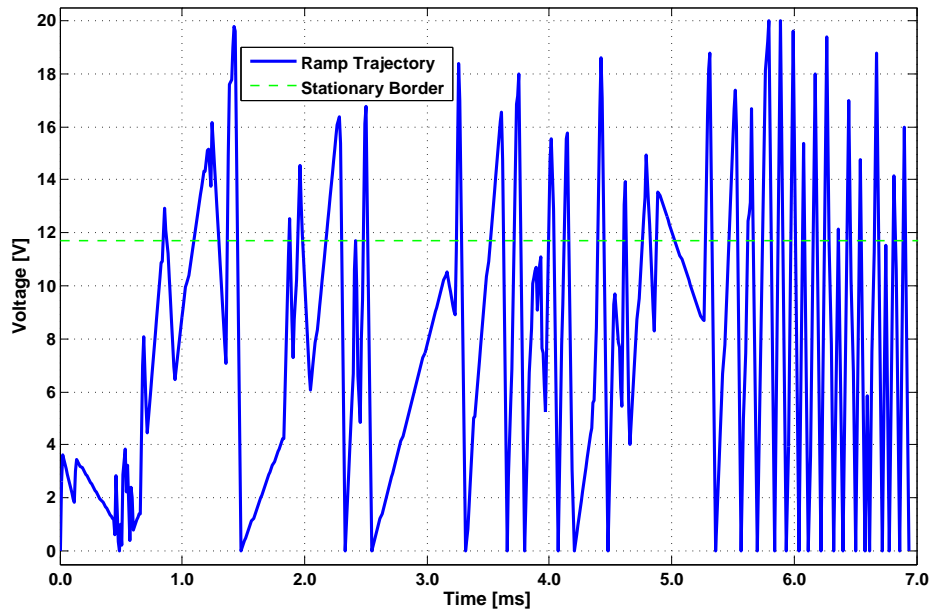


Figure 4.19: Stimulation of the voltage of the simulated electromagnetic valve system as described in Figure 4.15 with our active learning framework which is summarized in Algorithm 4.3 where a ramp-based trajectory design is employed.

moved to the remaining point set  $P_R$ . This follows from the definition of the ramps since all of them are valid by design. Overall, the gain in computational speed using the more resource conserving ramp trajectories is negligible since most of the effort during planning is spent on the safety evaluation.

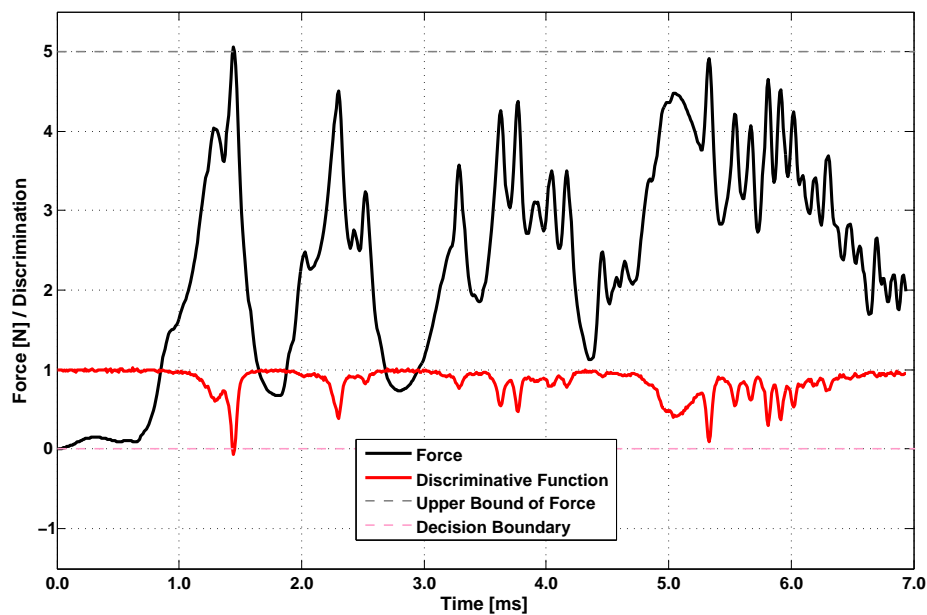


Figure 4.20: Resulting force of the queried voltage trajectory from Figure 4.19 on the simulated electromagnetic valve system as described in Figure 4.15 with our slightly adapted active learning framework based on ramp trajectories which is summarized in Algorithm 4.3.



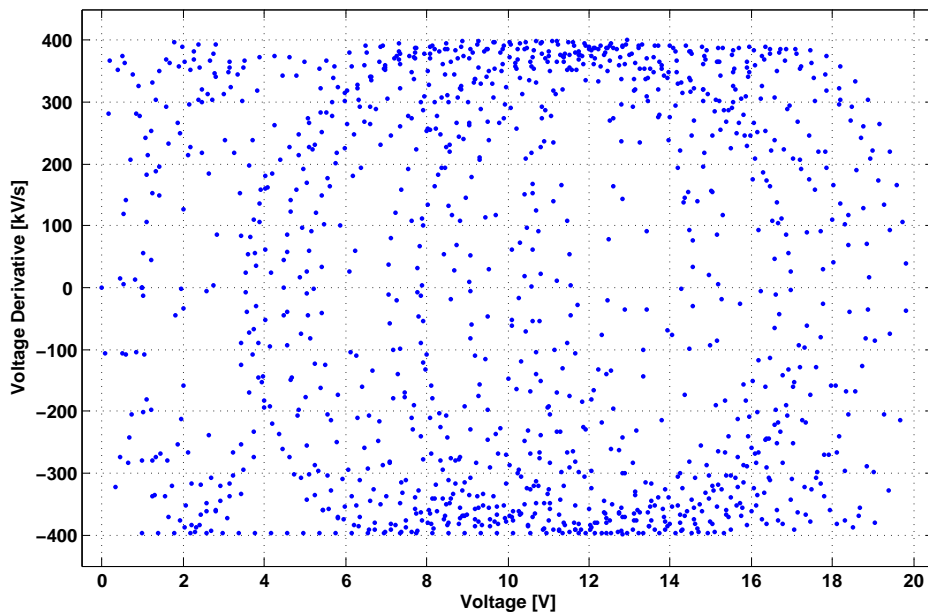


Figure 4.21: Resulting distribution of the queried points associated to the phase space design with cubic Beziér curves on the electromagnetic valve example captured with Algorithm 4.3. This illustration is based on the voltage path from Figure 4.16.

For a better comparison of the two considered trajectory designs on the electromagnetic valve example, the resulting distribution of the queried points in the phase space is analyzed. Figure 4.21 shows the nearly uniformly distributed points in  $\mathbb{P}$  resulting from the safe exploration approach based on cubic Beziér curves, cf. Figure 4.16. Besides that, Figure 4.22 summarizes the distributed phase space points

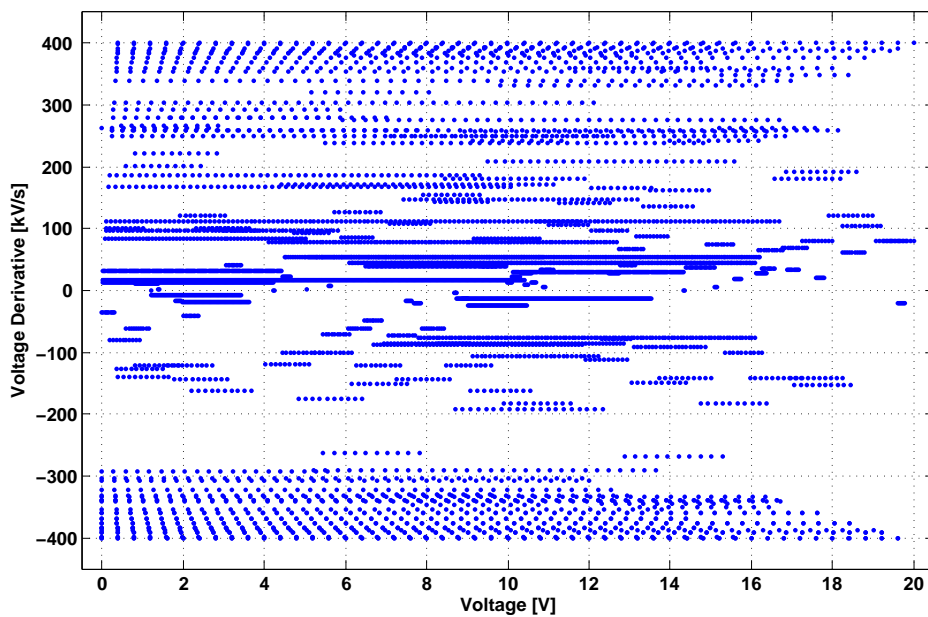


Figure 4.22: Resulting distribution of the queried points associated to the phase space design with ramp trajectories on the electromagnetic valve example captured with Algorithm 4.3. This illustration is based on the voltage path from Figure 4.19.

from the exploration with ramp trajectories based on Figure 4.19. The main difference between the two pictures is the characteristic grid with nearly equidistant voltage steps for constant gradients for the ramp-based trajectory design. Even the five times higher number of queried points in the phase space in the case of ramp trajectories is not enough to reach a coverage of  $\mathbb{P}$  compared to cubic Beziér curves. The coverage of the phase space  $\mathbb{P}$  with ramp trajectories is very dense in some regions due to the larger set of queried points, but it results also in larger holes as compared to the point distribution in Figure 4.21. The nearly equidistant spaced data points captured with the learning framework induced by ramp trajectories can lead to numerical problems during the GP based modeling when employing stationary covariance functions since it may result in a nearly singular Gram matrix. This issue is very unlikely when using our novel exploration scheme as presented first in this section, cf. Figure 4.16. Overall, the smooth trajectory design with cubic Beziér curves yields a good phase space coverage with only a few points so that a circle-like pattern arises.

## 4.4 Discussion

In this section the achieved results of our safe and novel active learning frameworks are reviewed. Remember, the presented safe exploration approaches were divided depending on the considered system dynamics, namely stationary and transient. Thus, after a short excursion through the existing active learning literature our exploration setting for stationary environments was explained firstly. For the safe exploration of stationary systems the overall modeling to solve this task is realized with an exploratory GP and a discriminative GP to evaluate the safety during querying. Both GP models are only connected through the sampled data which consists of input points with associated observations for exploring and safety modeling. This divided approach allows a very flexible and individual modeling of the safety and exploring issue. Besides that, the required computational resources are doubled to enable this flexibility. To reduce the computational effort it is recommended to employ incremental updates on the GP models during the subsequent point selection process. Note that we assume that the hyperparameters for both data-based models are given or previously determined and treated constant during the continuous modeling. This fixed treatment of hyperparameters allows the derivation of some theoretical results and exploration guarantees as shown in Section 4.2.1. In the combined with Schillinger et al. (2016) an active learning approach for stationary systems with online learned hyperparameters is presented. Furthermore, a framework considering uncertainty in the exploratory GP hyperparameters is described in Section 4.2.2. Therein, a mixture of GP priors with a discretized set of hyperparameters is used to model the exploratory part of our stationary active learning framework. The evaluation of this approach under uncertainty on a one-dimensional toy example in Section 4.2.6 shows that the influence of the magnitude and length-scale parameter of the employed kernel nearly vanishes. This behavior is based on the characteristic properties of the used isotropic Gaussian covariance function. The approximately equidistant sampling in the input space is founded on that fact as well. More generally, all stationary covariance functions as introduced in this thesis, see Section 2.1.4, provide such a exploration behavior. Another result from this evaluation is that the effects of not exactly known hyperparameters can be neglected in this exploration setting. Hence, an a priori and approximately estimated set of hyperparameters for a stationary kernel

in our active exploration is good enough to achieve very satisfactory results. Thus, in the following safe learning settings only stationary covariance functions with fixed hyperparameters for the exploratory GP are used and kept constant during the active system querying. Moreover, the used exploration strategy which is based on the differential entropy of the posterior GP has some nice advantages, see Lemma 2.1. The lemma provides properties of the entropy such as submodularity which lead to strong exploration guarantees for the derived greedy selection scheme. Namely, the final model entropy of the exploratory GP is greater than 63% of the optimal reachable entropy with the limited number of queries. This is one of the strongest statements in such generally NP hard exploration scenarios. Moreover, the introduction of safety essentially increases the complexity of the presented active learning framework. Herein, the safety is handled with an extraordinary GP classifier to distinguish between safe and unsafe regions of the input space. To realize a sampling of input points located mainly in the safe region of the stationary system, additional system feedback is required and used to design a health function. This safety function is defined in a way to enable an exact modeling around the decision boundary which ensures a good reasoning when the system status gets unhealthy during exploration. Of course, the definition of this health function requires established knowledge about the system under consideration together with the sampling of additional output, but without such a feedback it will not be possible to realize a safe active learning framework. For the beginning of our safe exploration approach a previously determined safe input point is also needed to start the algorithm. As shown in our theoretical analysis, a set of starting points which are closely located to each other would be needed where its size depends on the model parameters and the confidence parameter  $\nu$  following from the safety constraint for a new query selection. The query optimization of the exploratory model entropy under this constraint is realized with a Newton method. In this case, multi-starts induced by different randomly selected input points are employed to solve the generally multi-model optimization problem. This extravagant extension is necessary to result with a reasonable extrapolation behavior. Note that the configured safety constraint is fulfilled for each optimized query. With all of these settings and prerequisites a safety probability for our whole safe active learning scheme presented in Algorithm 4.1 was derived and proven. That is a remarkable property of our safe exploration framework and provide a configuration of the exploration behavior over the safety threshold induced by  $\nu$ . In the first evaluation of Algorithm 4.1 on a noisy cardinal sine function the tradeoff between fast exploration and a high safety is illustrated. As expected, if the safety of our exploration scheme is increased, the mean number of failed queries decreases to zero as well as the final differential entropy of the exploratory GP model is slightly reduced. The classification accuracy during the successive point selection and model retraining is always on an adequate level. But the classification of the few unsafe input points is very hard due to the sampled unbalanced data set. Nevertheless, the special adapted classification approach with an increased modeling behavior around the decision boundary yields good results on this two-dimensional toy example. In the second stationary active learning scenario the exploration of the configuration parameters for a linear controller on a inverse pendulum hold up problem was evaluated. This considered system required a more complex design of the health function defined with a saturating cost around the target state and averaged during the controller roll-out which is explained in Section 4.2.6 in detail. In this case it was even hard to get an a priori estimate of the safety and exploratory GP hyperparameters due to the large and rugged space of the control parameters. But the hyperparameter definition for the exploration of this control parameter space is not as critical as for the safety evaluations which was previously discussed. As in the previous toy example the unbalanced data obtained by mostly querying safe input points increases the classification complexity. But

despite these difficulties our safe exploration scheme provides meaningful and consistent results based on the assumed and derived theoretical properties of the learning algorithm. Also the favorable modeling accuracy of Gaussian process techniques supports the good and safe system exploration. Not to forget that these respectable results are based on the generated data set with the differential entropy criterion and their favorable properties.

After consideration of the safe active learning framework for stationary environments, the thereby obtained knowledge and experience was used to develop the safe active exploration scheme for transient systems. The main challenge in this transient task was to cope with the additional time-dependency of the dynamic system which increases the exploration algorithm complexity and the performance requirements. Thus, it was necessary to switch from a point to point planning approach as in the previous case to a trajectory based exploration strategy since a stepwise approach would be hard to realize for near real-time planning. Moreover, this planning approach was chosen to enable a forward-looking strategy to avoid dead ends during the safe stimulation of the considered dynamic system. The final querying of complete trajectories gives the possibility to compute and evaluate new trajectories during the time where the current trajectory is queried. To avoid the overhead of point-wise model updates the transient GPR models are updated in a batch manner, i.e. with the sampled data from one complete queried trajectory. Remember that the goal in the transient setting is still the same as in the former discussed stationary framework. Hence, it is also desired to create a representative data set in a safe manner as quickly as possible to reduce the time for interacting with the considered environment. For the sampling of meaningful data a differential entropy-based approach based on a transient exploratory GPR model was presented in Section 4.3.1. Nevertheless, in our obtained Algorithm 4.3 a different exploration approach was used since it is very hard to obtain a fast and stable optimization procedure even for good parametrized trajectories. Additionally, such a trajectory optimization algorithm based on the differential entropy has to fulfill all restrictions of the given phase space. Instead of such a complex trajectory optimization approach a space-filling Sobol design in the restricted phase space lays the foundation for the upcoming trajectory planning approach. Building on the favorable properties of this Sobol point set and a given stationary and safe initial point, an exploration framework with trajectories induced by cubic Beziér curves is started. Thus, the Sobol design takes care of exploring the whole phase space when the point to point trajectories are queried. The cubic Beziér curves for the trajectory creation are chosen because of their flexibility, less parametrization effort and their overall smooth behavior which enables one times continuously differentiable trajectories. Especially, this is true for the Sobol points where the trajectories are connected. For the safety evaluation of the created trajectories a probability based on a transient GPR model is used. To avoid dead ends this probability is not only calculated for the current trajectory, but also for the complete path of the considered trajectories connected with a final backward trajectory to the safe initial point. Additionally, a binary search tree approach was used for the quick creation of a trajectory set where the safest path of that tree was selected for further querying. To keep the learning safe, our transient GPR model gets updated after every queried single trajectory and the whole tree-based planning starts again. That ensures fast and batched GP model updates and gives the chance to prepare the next search tree of feasible trajectories during the current query time. For the safety evaluation of a determined trajectory path a possible high-dimensional multivariate Gaussian probability needs to be calculated. This calculation was realized with an adequate and efficient Gaussian expectation propagation algorithm to yield stable and meaningful results for a forward-looking long-term

---

planning. In the evaluation of this active learning framework for transient environments a simulated electromagnetic valve system is used. The definition of the modeled health function with the sampled system response is a crucial point in our algorithm and requires special attention. It is advisable to include a buffer to the real system limit or use a special adapted class of functions to model the system boundary well. The thereby employed and subsequently updated sparse Gaussian process model induced by the DTC approximation with our novel maximum error greedy selection criteria provides the basis for the performant and secure exploration. Only with the underlying NARX approach for the time-dependent system modeling it was possible to reach such a good model quality. Another reason for the stable and safe exploration of the electromagnetic valve emulation is the smoothness of the developed trajectory planning with cubic Beziér curves. Compared to the additionally evaluated trajectory design with linear ramps the former approach yields a more homogenous system stimulation as well as a better coverage of the associated phase space. Finally, the re-consideration of formally skipped Sobol points at the end of our exploring scheme strives to complete the querying of all created points from the initial set. This step enables a maximum of exploration with the initial design and configuration of the presented active learning algorithm.



## 5 Conclusion

The successful examination of the presented machine learning research topics provides a notable increase in competence on complex system modeling and efficient data generation for the Corporate Sector Research and Advance Engineering of the Robert Bosch GmbH. The investigated topics as well as their underlying challenges were defined together with the University of Stuttgart that supported the writing of the thesis. Furthermore, extensive discussions on the visited conferences and workshops had a notable impact on the delivered content. In addition, most of the practical examples and experimental setups for the computational evaluation of the developed algorithms and frameworks were given or inspired by the two mentioned institutions above. The detailed summary and outlook of the considered research is given in the following sections.

### 5.1 Summary

This thesis investigates two strongly coupled topics from the machine learning research area. Specifically, the data-sampling as well as the data-modeling side of industrial and technical automotive systems. Especially the connection between these two approaches in the considered model-based active learning scenarios enables a high potential for meaningful final results. Before reviewing the active learning frameworks, we summarize our novel sparse Gaussian process approximation for efficient dynamic system modeling. Based on the standard Gaussian process model which is introduced in the background chapter of this thesis, the derivation of the so-called sparse DTC approximation is shown in the first elaborated chapter. After an extensive review of the related work the technical details of this approximation are explained. Additionally, a structured graphical overview of the relations between the standard Gaussian process model and its approximations is given. This review makes it obvious that all of the briefly illustrated sparse GP approximations have more or less drawbacks. A high diversity in computational speed, different model selection approaches, a deteriorated posterior variance or numerical approximation problems are some of the disadvantages. The DTC approach is chosen because of their information-approximation likelihood approximation with an acceptable computational effort and its straightforward model selection as well as hyperparameter optimization technique. The core of this Gaussian process model approximation is based on the selection of an active input set. To obtain a good set of active points many greedy approaches have been devised in the past decades since an exact computation is to costly even for medium-sized training data sets. Besides the presentation of many state-of-the-art greedy insertion and deletion criteria our self developed approaches are introduced. Our maximum error criterion for selecting a nearly optimal active set is intuitively derived from the method by Smola and Bartlett to overcome the high memory requirements and the lack of computational speed of this criterion under a high modeling accuracy. Next to the theoretical foundation of our maximum error criterion, a slightly modified scheme for sparse model selection in the DTC approximation is introduced.

This advantageous expectation maximization strategy combines the hyperparameter optimization with the active set selection to increase the quality of the resulting sparse model. Especially the connection between these two optimization problems enables a stable and performant predictive model creation. Due to the alternating expectation maximization steps, a fast active set selection is essential for an efficient sparse model selection. Thus, our efficient and resource conserving maximum error selection strategy perfectly geared towards this case. The evaluation of this learning process on a training data set from a SARCOS robot arm substantiate the former statements regarding our sparse DTC approach. After the presented experiments for the sparse model selection, our maximum error criterion is compared to all other introduced selection criteria on various data sets from diverse regression problems. In particular, on a training data set with nearly half a million training points our maximum error insertion scheme performs perfectly with respect to mean square errors and the absolute deviation on test data. This data was sampled with real world measurements of the electronic power steering assistance system to simulate the vehicle power while monitoring the cars energy balance. Additionally, the fast training times on this large and transient modeling task confirm the foundation of our novel selection strategy. The basis for the successful modeling in this transient learning scenario is provided by a NARX approach as introduced in our background chapter. The a priori determination of the NARX structure is the only parametric part in our transient modeling framework with sparse Gaussian processes. Furthermore, our developed deletion criterion is as fast as the randomized deletion and applicable to reduce prediction times for test points where the generalization accuracy is only slightly decreasing. Our approach is one of the best performing methods compared to other machine learning methods for regression problems on benchmark data sets. Generally, all Bayesian techniques yield very good results in this offline evaluation on robot data sets. Specifically, the fully independent training conditional Gaussian process approximation produced good results according to the considered normalized mean square errors in this extensive comparison. This method is explained after our novel approach in Chapter 3 and used to extend the proposed experiments with Bayesian methods. This so-called FITC approximation lacks efficiency as well as modeling accuracy on higher dimensional input data due to the increasing optimization problem for model selection. Last but not least, our novel DTC approach was employed to learn an inverse dynamics model for a compliant and real-time tracking control task on a PR2 robot. In this case over half a million training points are sampled to obtain a generic and precise model of the inverse dynamics from the right robot arm. The experiments show that a compliant tracking control scheme based on a learned DTC model is able to realize compliant control under an exact tracking performance which is comparable to a standard control scheme with high gains. Overall, the proposed DTC approximation with our maximum error insertion and deletion criterion is well suited for many regression problems and their practical implementation. Hence, this approach is also considered in our derived model-based active learning scenarios.

The active learning approaches proposed in Chapter 4 for efficient and guided data-sampling from dynamical environments provide the basis for sustainable and generalizable models. Additionally, one of our main requirements is to generate informative as well as less redundant data in a safe manner, i.e. the active learner has to distinguish between safe and unsafe regions of the investigated system during querying. Due to the main characteristics of the chosen systems being explored, we differentiate into stationary and transient active learning frameworks to establish adequately adapted exploration strategies. Beginning with our exploration approach for stationary, i.e. time-independent environments, a distinction between safe and unsafe input regions is realized to enable the definition of safety. This is achieved by



measurable system responses that are used to design a learnable health function of the stationary system, especially close to the decision boundary. To realize an exact modeling around the decision boundary a hybrid learning strategy for the safety classifier is developed. Hence, our approach uses positive class labels for really safe measurable input points and negative ones for points that certainly cause damage on the system. Furthermore, between these edge cases a continuous function is formed from the system responses to enable an adjusted learning of the discriminative function. One requirement on the so-called health function is that each input point must yield either a class label or a continuous function value to ensure consistent training data. Moreover, the health function should indicate the proximity to the decision boundary and not represent the behavior of the system responses. Finally a specially designed GP classifier is used to learn a discriminative model on the obtained data to derive a safety constraint for our exploration framework. Overall, the design of the health function is the key point of our stationary active learning approach and the only way where expert knowledge for introducing safety is included. Moreover, we coincide with the learning theory literature in the point that it is not possible to explore a considered environment safely without any additional system feedback. As shown in the evaluation of this framework using a toy example and a control parameter exploration task for the inverse pendulum hold up problem, it is demonstrated that already a less complex but meaningful representation of the system feedback in the health functions leads to favorable results. For the subsequent exploration of the underlying environments, we use an entropy-based sampling approach. The predicted variance of a continuously trained exploratory GP model forms the basis of this method. This variance based selection approach is chosen because of its favorable properties and exploring guarantees for the employed active learning framework with standard GPs. Besides that, a short excursion about exploring under uncertainty of the hyperparameters for the exploratory GP is investigated. The results of this investigation with mixtures of GP priors show that a priori estimated hyperparameter set is sufficient to achieve a good exploration of the input space. Combining the extraordinary GP classifier for the safety evaluations and the entropy-based exploration scheme with a standard GP on the same input space data gives our stationary safe active learning framework. The selection of new queries in the bounded input space is realized therein with a second-order optimization scheme on the exploratory GP variance, restricted to the safety constraint following from the discriminative GP model. The probability of safety in this restriction is adjustable with a confidence parameter to enable a less risky exploration of the stationary system. Following from the theoretical analysis of our active learning framework, the confidence parameter can be derived from the previously defined failure probability of the whole algorithm if the maximal number of all queries is given. Another requirement for our safe exploration is that at least one safe starting point is available. Depending on the hyperparameters of the discriminative GP and the overall failure probability, more than one initialization point can be necessary to create a sufficient confident safety model. Therefore, we derive a theorem which provides a minimal starting set size to give an intuition about the previously needed number of queries. Moreover, the presented theoretical analysis of our active learning scheme provides a profound extension and further insights into our algorithm. The evaluation of our stationary active learning framework is carried out for a one-dimensional toy example and for a more complex parameter exploration task of a pendulum hold up problem to visualize the safe exploration behavior under the above discussed properties. In both cases it is shown that, depending on the employed failure probability, the number of safe queries increases while the exploration of the environment slows down and vice versa. This behavior of our active learning algorithm represents the most applicable tradeoff between a fast and a safe exploration with respect to the considered stationary

systems. In addition, the obtained entropy trends indicate that the sampled data cover the safe input region in an adequate manner inside the allowed confidence parameter range.

Applying the results from the stationary approach, our framework for transient and safe exploration is derived. In contrast to the previous setting, it is now necessary to consider the complete path through the transient input space during exploration of the time-dependent environment. Thus, the point to point planning is extended to a trajectory based approach where the basis is also given by a safe and stationary initial point. Our exploration scheme starts from this initial point with a trajectory to a point in the phase space selected from a generated Sobol point set of the whole phase space. This Sobol point set of a predefined size is employed to allow for a good exploration of the transient system by avoiding a computational intensive optimization framework for trajectory planning under the near real-time conditions when using high sampling rates. Thus, an iterative approach is introduced to find a safe path successively that may traverse all of the created Sobol points. This gives the possibility to react in an adequate manner if any disturbances occur during exploration where a fast rescheduling is necessary. The trajectory design is implemented with cubic Bézier curves to handle the near real-time conditions and to allow a comfortable as well as flexible parametrization in the restricted phase space. Furthermore, the smooth characteristic of the higher order Bézier curves results in a better modeling behavior for this transient task. This allows for a fast evaluation of the Bézier trajectory on a discretized set of timestamps according to the sampling frequency. The crucial point in our model-based active learning framework is to assess the safety of the generated trajectories. Note that for the safety evaluation, a transient and sparse GP model with a predefined NARX structure and a previously defined hyperparameter set is employed. To yield a forward-looking exploration strategy in each iteration of our framework, a tree of trajectories between not yet visited Sobol points is generated and the safety of each path is calculated with the trained sparse DTC model. To avoid dead ends, each path of the created search tree is extended with a backward trajectory to the safe initial point and a consistent safety probability is assessed. The sparse DTC model is trained on a continuous health function of the considered environment which is build from some simultaneously sampled system responses. In contrast to the stationary case, the transient GP regression model with our new developed maximum error insertion strategy is used for the safety evaluation due to the information-optimal approximation scheme, and thus more real-time capable computational training and prediction performance. Moreover, this modeling framework allows efficient model updates after sampling new data from the environment to ensure an up-to-date safety evaluation. The evaluation of each path from the generated search tree together with its backward trajectory based on the described safety model is realized with a special designed expectation propagation algorithm to yield fast and accurate probabilities even for very long trajectories. This expectation propagation scheme is derived to overcome the disadvantages of some other discussed approaches like the principle of inclusion-exclusion or Monte Carlo based approaches. On a simulated electromagnetic valve system the smooth stimulation and the safe exploration behavior of our transient active learning framework is demonstrated. For this typical example from the Bosch domain it is necessary to provide an exact modeling of the safe phase space region to avoid critical system states where high magnetic forces occur. The magnetic force of the valve is used to derive an exponential health function with a high gradient around the decision boundary to enable an early detection of dangerous trends. Note that this health function can also contain a slight buffer to account for the only approximately known or noisy real system boundary. The sparse DTC approximation generates a reliable prediction and an adequate modeling of

the safe system space with the employed NARX structure. This excellent modeling behavior enables a safe stimulation of the environment over the stationary system border which is possible if the time in the less safe region is short enough. Thus, a more informative coverage with respect to the sampled data of the whole electromagnetic valve system is created. Summarized, our transient active learning scheme leads to a safe system exploration which in turn results in a broad and well distributed data set to enforce a high model quality.

## 5.2 Future Work

An overview of the still existing challenges which could not be considered here together with some motivation for future research topics is given in this last section of the thesis. Herein, another unifying view on insertion and deletion criteria to the already existing width and depth of Gaussian processes and their approximations is given. Nevertheless, a so far not investigated combination of insertion and deletion strategies can reduce the redundancy in the greedily selected active set, and thus lead to a higher approximation accuracy. The employed covariance function has a high significance for the resulting model quality and especially for the presented sparse approximations where all of the selection criteria depend more or less directly on it. Thus, the common practice to use stationary kernels is not the best choice in every modeling task where the underestimated power of non-stationary kernels should not be neglected and be applied when necessary. Even the combination of covariance functions of a different kind or extensions, like dimensional reduction approaches, can help to solve the challenge of a transient modeling. Another promising approach is the multi-output Gaussian process framework, cf. Boyle and Frean (2005), together with its sparse approximations to reduce the increased effort in memory and computations, e.g. as in Alvarez and Lawrence (2009). The advantageous and hardest part is the description of the interdisciplinary covariances to allow an adequate learning of the underlying functional relationship between the outputs to yield a more general model. Furthermore, the adaption of hyperparameters plays a crucial role in all considered Bayesian models. Next to our presented expectation maximization method for learning hyperparameters, it is worth to say that more efficient optimization algorithms and strategies could increase the model training performance. For example, a constrained optimization framework, where the hyperparameters and the active set of training points is adapted together, could be investigated. When dealing with online learning tasks for real-time applications, it is necessary to handle a quick model adaption with the new data and an efficient retraining of the hyperparameters without losing the already learned functional behavior. Partitioning schemes like local Gaussian processes or a generalized Bayesian committee machine, extended with some sparse approaches, can be able to solve the hard problems for streaming based data. With the new computational possibilities and the parallelization of the afore mentioned frameworks, it is feasible to realize good and safe results even with the online adaption of hyperparameters. A sophisticated and well adapted GP modeling approach is essential to reach a high safety as well as an effective exploration with our model-based active learning scenarios. The fundamental design of experiments literature yields already a lot of best practices to construct sampling strategies for the presented GP setting. These ideas may help to derive further exploration criteria like the mutual information, which can result in a more adequate behavior in other

use cases, where the properties of the presented variance sampling approach are insufficient. Furthermore, an extension of the Bayesian modeling setup with a heteroscedastic variance treatment changes the effects of the entropy based querying but raises the complexity of the inference task. The same effects can be achieved with the usage of non-stationary covariance functions which are well suited for the modeling of the considered environment. Besides that, a multi-output active learning setting may be beneficial when one target variable is not enough to realize an adequate space covering design. The worst case of this multi-output approach can lead to a equally distributed set of queries which can also be obtained without the active learning framework. Mostly a fixed hyperparameter set was used in the presented active learning schemes to ensure a constant exploration behavior. However, in some exploration cases more than one good initialization of the hyperparameters is not available which makes an online adaption necessary. Our proposed online adaption method for uncertainty in the hyperparameters provides a good starting point for further investigations on more real world applications. In this case a parallel implementation on the available computational resources is advantageous and speeds up the model-based query building process. Especially the modeling and learning of the safety requires a high attention on our active learning setting. Without any question, a response of the explorable system is necessary to design and learn a safety function. For example, a continuation and implementation of the proposed stationary active learning framework from the end of Section 4.2.3 with truncated Gaussian densities to model the safety would be possible. Besides that, our extended Gaussian process classification approach works very well in the stationary case, but an enhancement with already known unfeasible input points and regions would yield a more confident safety model. The same strategy is also applicable in the transient exploration task, where it is much harder to define dynamically non-reachable input locations. Moreover, the online adaption of the hyperparameters for the safety model is quite hard since the possibility to end up in dangerous system states with incorrectly adjusted hyperparameters can be quite high. Additionally, the a priori estimation of the general failure probability is harder to obtain and currently not done for all model settings like different covariance functions within Gaussian processes. Overall, the real-time requirements for the safety and exploratory modeling part are the biggest challenge in transient active learning scenarios. When bringing together an exploration and safety trajectory optimization scheme the computational complexity increases but can be tackled by a parallelized implemented algorithm. As already mentioned, a more complex and harder to realize task is the online adaption of hyperparameters in a transient active learning framework, for example with respect to an efficient and safe implementation. One strategy to tackle these issues may lie in the creation and evaluation of new trajectories during the current querying of the environment with an already generated one. Nevertheless, both GP models lack some accuracy in this case due to the outdated model training with respect to the newest data. This slight drawback can be accepted since the gained computational time to generate the subsequently following safe trajectories is much more important in the transient case. In addition and as proposed in our setting, an employed backward strategy to handle any unexpected behavior or other disturbances during querying of the system is always recommended. Besides the investigated Bayesian active learning approaches, it is possible to employ novel techniques like generative adversarial networks, cf. Goodfellow et al. (2015), to derive new active learning algorithms with different characteristics which is beyond the scope of this thesis. This review of open questions and further research directions is by far not complete and other researchers have enough opportunities to come up with creative and efficient modeling as well as active learning ideas.

# A Appendix

The appendix of the thesis is subdivided into three sections. The first section of the appendix gives an overview about the used abbreviations and their meanings. An excursion through the employed mathematical notations is presented in Section A.2. Section A.3 contains further mathematical information, explanations, relations, and proofs related to the previous chapters of this thesis which are postponed to the appendix.

## A.1 Abbreviations

The following table summarizes the abbreviations together with their description which are used throughout the thesis. To provide a readable and comprehensible content only the absolutely needed and most common abbreviations are used. These are subsequently summarized in Table A.1.

abbreviation	description / explanation
ANN	artificial neural network
ARD	automatic relevance determination
BCM	Bayesian committee machine
CE	classification error
CG	conjugate gradients
CS	Csató
DoE	design of experiments
DoF	degree of freedom
DTC	deterministic training conditional
ECU	engine control unit
EM	expectation maximization
EP	expectation propagation
EPS	electronic power steering
EHPS	electro-hydraulic power steering
Exp	exponential
FITC	fully independent training conditional
FN	false negative
FP	false positive
GP	Gaussian process
GPC	Gaussian process classification

abbreviation	description / explanation
GPR	Gaussian process regression
IG	information gain
KL	Kullback-Leibler
LGP	local Gaussian process
MAE	maximum absolute error
MDP	Markov decision process
ME	maximum error
MLM	maximum likelihood method
MPA	matching pursuit approach
MSE	mean square error
NARX	non-linear autoregressive exogenous
NER	normalized entropy ratio
NLML	negative logarithmic marginal likelihood
NMAE	normalized maximum absolute error
NMSE	normalized mean square error
NN	neural network
NP	non-deterministic polynomial-time
NRMSE	normalized root mean square error
NX	non-linear exogenous
Opt	optimal
PITC	partially independent training conditional
PCA	principal component analysis
QC	Quiñonero-Candela
RBF	radial basis function
RFRLS	random Fourier regularized least squares
RKHS	reproducing kernel Hilbert space
RMSE	root mean square error
ROC	receiver operating characteristic
RVM	relevance vector machine
SB	Smola and Bartlett
SE	squared exponential
SEN	sensitivity
SNR	signal-to-noise ratio
SoD	subset of data
SoR	subset of regressors
SPC	specificity
SVM	support vector machine
SVR	support vector regression
TN	true negative
TP	true positive
UCB	upper confidence bound

abbreviation	description / explanation
VAR	variational
VDA	German association of the automotive industry, German: Verband der Automobilindustrie e.V.

Table A.1: Most common abbreviations that are employed in this thesis.

## A.2 Notations

This section provides the fundamental mathematical notations and symbols that are used throughout the thesis. To provide a better overview, this section is subdivided into three parts according to the topic where the notation was firstly introduced.

### General

Table A.2 summarizes the standard functions and spaces from the mathematical point of view.

notation	description / explanation
$x$	scalar
$ x $	absolute value
$\text{sgn}(x)$	sign function
$\exp(x)$	exponential function to the base $e$
$\log(x)$	natural logarithm
$\sin(x)$	sine
$\text{sinc}(x)$	cardinal sine, not normalized
$\arcsin(x)$	arcsine
$\mathbb{I}(x)$	indicator function, i.e. one if $x$ is true, otherwise zero
$\Gamma(x)$	gamma function
$\mathcal{N}(x)$	standardized univariate Gaussian density function
$\Phi(x)$	standardized univariate Gaussian probability function
$\Phi^{-1}(x)$	inverse standardized univariate Gaussian probability function
$B_\varrho$	modified Bessel function of second kind with parameter $\varrho$
$B_{m,j}(\tau)$	$j$ -th Bernstein polynomial of order $m$ with $\tau \in [0, 1]$
$\delta_{ij}$	Kronecker delta
$m!$	factorial of a natural number $m$
$\emptyset$	average
$\propto$	proportional to

notation	description / explanation
$\mathbb{N}$	natural numbers $\{1, 2, 3, \dots\}$
$\mathbb{Z}$	integers $\{\dots, -2, -1, 0, 1, 2, \dots\}$
$\mathbb{R}$	real numbers
$\mathbb{C}$	complex numbers
$\mathbb{X}$	input space
$\dot{\mathbb{X}}$	gradient space
$\mathbb{P}$	phase space, outer direct sum of the input and the gradient space

Table A.2: Basic mathematical functions and spaces.

## Matrix Calculus

As usual defined, vectors and matrices are always printed in bold face, cf. Table A.3. For matrix structures, the operators like  $\text{diag}(\cdot)$  or  $\text{trace}(\cdot)$  result in a structure of smaller order, as explained when employed in this thesis.

notation	description / explanations
$\mathbf{0}$	column vector, where all elements are equal to zero
$\mathbf{1}$	column vector, where all elements are equal to one
$\infty$	column vector, where all elements are equal to infinity
$\mathbf{x}$	column vector
$\mathbf{x}^T$	transpose of vector $\mathbf{x}$ , row vector
$\mathbf{x} \circ \mathbf{y}$	Hadamard product
$\mathbf{x} \preceq \mathbf{y}$	element-wise lower equal
$\mathbf{x} \succeq \mathbf{y}$	element-wise greater equal
$\mathbf{x} \oplus \mathbf{y}$	outer direct sum
$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^T \mathbf{y}$	Euclidean scalar product
$\ \mathbf{x}\ $	Euclidean vector norm
$\text{diag}(\mathbf{x})$	diagonal matrix with elements from vector $\mathbf{x}$
$\text{prod}(\mathbf{x})$	product of the elements from vector $\mathbf{x}$
$\phi(\mathbf{x})$	vector function
$\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle_{\mathbb{H}}$	scalar product associated with the Hilbert space $\mathbb{H}$
$\ \phi(\mathbf{x})\ _{\mathbb{H}}$	norm in the Hilbert space $\mathbb{H}$
$\mathbf{A}$	matrix
$\mathbf{A}^T$	transpose of matrix $\mathbf{A}$
$\mathbf{A}^{-1}$	inverse of matrix $\mathbf{A}$
$\mathbf{A}^{-T}$	transpose and inverse of matrix $\mathbf{A}$
$\mathbf{I}$	identity matrix



notation	description / explanations
$ \mathbf{A} $	determinant of matrix $\mathbf{A}$
$\ \mathbf{A}\ $	Euclidean matrix norm
$\text{trace}(\mathbf{A})$	trace of matrix $\mathbf{A}$
$\text{sym}(\mathbf{A})$	symmetric part of matrix $\mathbf{A}$ , i.e. $\text{sym}(\mathbf{A}) = \frac{1}{2}(\mathbf{A} + \mathbf{A}^T)$
$\text{diag}(\mathbf{A})$	column vector with diagonal elements from $\mathbf{A}$

Table A.3: Vector and matrix operators.

## Stochastic

Firstly, we distinguish between probability densities and distributions. Furthermore, an index is introduced for the shorthand notations like  $\text{E}[\cdot]$  or  $\text{H}[\cdot]$  if the underlying distribution is not clear from the context, cf. Table A.4.

notation	description / explanations
$\sim$	distributed according to
$p(\mathbf{x})$	multivariate probability density function of the vector $\mathbf{x}$
$p(\mathbf{y}   \mathbf{x})$	multivariate conditional probability density function of the vector $\mathbf{y}$ given $\mathbf{x}$
$\text{P}(\mathbf{x})$	multivariate probability distribution of the vector $\mathbf{x}$
$\text{P}(\mathbf{y}   \mathbf{x})$	multivariate conditional probability distribution of the vector $\mathbf{y}$ given $\mathbf{x}$
$\text{Ber}(p)$	generalized Bernoulli distribution with parameter $p$
$\chi_n^2(p)$	quantile of the chi-squared distribution with $n$ degrees of freedom according to the probability $p$
$\chi^2(n)$	chi-squared distribution with $n$ degrees of freedom
$\mathcal{R}(x   \mu, \sigma^2, l, u)$	univariate rectified Gaussian density function with mean $\mu$ and variance $\sigma^2$ to the lower and upper bound $l$ and $u$
$\mathcal{L}(x   \mu, \sigma^2, l)$	univariate lower truncated Gaussian density function with mean $\mu$ and variance $\sigma^2$ to the lower bound $l$
$\mathcal{U}(x   \mu, \sigma^2, u)$	univariate upper truncated Gaussian density function with vector $\mu$ and variance $\sigma^2$ to the upper bound $u$
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \mathcal{N}(\mathbf{x}   \boldsymbol{\mu}, \boldsymbol{\Sigma})$	multivariate normal density (of $\mathbf{x}$ ) with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$
$\text{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \text{N}(\mathbf{x}   \boldsymbol{\mu}, \boldsymbol{\Sigma})$	multivariate normal distribution (of $\mathbf{x}$ ) with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$
$\text{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{z}))$	Gaussian process with mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{z})$
$\text{Pr}[\cdot]$	probability
$\text{E}[\cdot]$	expectation

notation	description / explanations
$\text{Var}[\cdot]$	variance
$\text{Cov}[\cdot, \cdot]$	covariance
$\text{H}[\cdot]$	entropy
$\text{KL}[\cdot \parallel \cdot]$	Kullback-Leibler divergence

Table A.4: Statistical and probability theoretical notations.

## A.3 Mathematical Explanations

In this section, fundamental mathematical relationships are explained for a better understanding of the presented work. The main equations for the matrix algebra are related to Lipschutz and Lipson (2013) and Meister (2015). A broad overview is also given by Petersen and Pedersen (2012). The properties and calculation rules for dealing with multivariate normal distributions can be found in Mardia et al. (1980), Press (2005), Herbrich (2005), Cover and Thomas (2006), and Toussaint (2011). The following derivations, differentiations, and proofs are postpone to the appendix to preserve the legibility of the thesis.

### A.3.1 Matrix Calculus

The Cholesky decomposition of a symmetric and positive definite matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is the factorization of  $\mathbf{A}$  in a lower triangular matrix  $\mathbf{L} \in \mathbb{R}^{n \times n}$  and their transpose such that

$$\mathbf{A} = \mathbf{L}\mathbf{L}^T . \quad (\text{A.1})$$

The matrix  $\mathbf{L}$  is named as the lower Cholesky factor of  $\mathbf{A}$ . The calculation of  $\mathbf{L}$  is numerically stable and often used for solving linear systems. Practically, it can be helpful to add a small jitter  $\epsilon$  to the diagonal elements of  $\mathbf{A}$ , i.e. considering  $\mathbf{A} + \epsilon\mathbf{I} = \mathbf{L}\mathbf{L}^T$ , which may numerically ensure the positive definiteness. If the Cholesky factor  $\mathbf{L}$  of a matrix  $\mathbf{A}$  is known, it is easy to compute the determinant

$$|\mathbf{A}| = \prod_{i=1}^n l_{ii}^2 \quad (\text{A.2})$$

or the logarithm of the determinant

$$\log(|\mathbf{A}|) = 2 \sum_{i=1}^n \log(l_{ii}) , \quad (\text{A.3})$$

respectively.

The QR decomposition of a square matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is the factorization of  $\mathbf{A}$  in an orthogonal matrix  $\mathbf{Q} \in \mathbb{R}^{n \times n}$ , thus it holds true that  $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}$ , and an upper triangular matrix  $\mathbf{R} \in \mathbb{R}^{n \times n}$  such that

$$\mathbf{A} = \mathbf{Q}\mathbf{R} . \quad (\text{A.4})$$

Note that the QR decomposition of a matrix always exists and for their absolute determinant follows

$$||\mathbf{A}|| = \left| \prod_{i=1}^n r_{ii} \right|, \quad (\text{A.5})$$

where  $r_{ii}$  refers to the diagonal elements of  $\mathbf{R}$ . The QR decomposition of a given matrix can be numerically stable calculated with Householder reflections or Givens rotations.

Generally, the rules

$$\begin{aligned} |\mathbf{A}^T| &= |\mathbf{A}|, \\ |c\mathbf{A}| &= c^n |\mathbf{A}|, \\ |\mathbf{AB}| &= |\mathbf{A}||\mathbf{B}|, \text{ and} \\ |\mathbf{A}^{-1}| &= |\mathbf{A}|^{-1} \end{aligned}$$

are helpful for calculating the determinant of a square matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  with  $\mathbf{B} \in \mathbb{R}^{n \times n}$  and  $c \in \mathbb{R}$ . The last equation requires the regularity of  $\mathbf{A}$ . Also, the following equations

$$\begin{aligned} \text{trace}(c\mathbf{A} + d\mathbf{B}) &= c \text{trace}(\mathbf{A}) + d \text{trace}(\mathbf{B}) \text{ and} \\ \text{trace}(\mathbf{AB}) &= \text{trace}(\mathbf{BA}) \end{aligned}$$

for the trace of square matrices are frequently used throughout the thesis with the additional constant  $d \in \mathbb{R}$ . For the last case, it is only necessary that  $\mathbf{A}$  and  $\mathbf{B}^T$  have the same dimensions.

Differentiation rules for matrices, where each element is a function of the variable  $t \in \mathbb{R}$ , are often required. Thus, the differentiation of a matrix corresponds to the element-wise differentiation with respect to  $t$ . For a regular quadratic matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  it holds true that

$$\frac{\partial \mathbf{A}^{-1}}{\partial t} = -\mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial t} \mathbf{A}^{-1} \quad (\text{A.6})$$

and

$$\frac{\partial \log(|\mathbf{A}|)}{\partial t} = \text{trace} \left( \mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial t} \right). \quad (\text{A.7})$$

The product rule for two matrices  $\mathbf{A}$  and  $\mathbf{B}$  with appropriate format is given by

$$\frac{\partial (\mathbf{AB})}{\partial t} = \frac{\partial \mathbf{A}}{\partial t} \mathbf{B} + \mathbf{A} \frac{\partial \mathbf{B}}{\partial t}, \quad (\text{A.8})$$

where it should be noted that the commutativity does not generally hold. Then, for two functions  $f$  and  $g$  which depend on a matrix  $\mathbf{A} \in \mathbb{R}^{n \times m}$ , it follows

$$\frac{\partial (f(\mathbf{A})g(\mathbf{A}))}{\partial t} = \frac{\partial f(\mathbf{A})}{\partial t} g(\mathbf{A}) + f(\mathbf{A}) \frac{\partial g(\mathbf{A})}{\partial t}. \quad (\text{A.9})$$

For the differentiation of the trace of the square matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  it is pointed out that

$$\frac{\partial \text{trace}(\mathbf{A})}{\partial t} = \text{trace} \left( \frac{\partial \mathbf{A}}{\partial t} \right). \quad (\text{A.10})$$

Focusing on a quadratic form with respect to a square matrix  $\mathbf{A}$  as already employed above and a vector  $\mathbf{x} \in \mathbb{R}^n$ , it holds true that

$$\frac{\partial (\mathbf{x}^T \mathbf{A} \mathbf{x})}{\partial \mathbf{x}} = \mathbf{x}^T (\mathbf{A} + \mathbf{A}^T) = 2 \mathbf{x}^T \text{sym}(\mathbf{A}) . \quad (\text{A.11})$$

If and only if  $\mathbf{A}$  is symmetric it follows

$$\frac{\partial (\mathbf{x}^T \mathbf{A} \mathbf{x})}{\partial \mathbf{x}} = 2 \mathbf{x}^T \mathbf{A} . \quad (\text{A.12})$$

Throughout the thesis, the gradient of a function depending on  $\mathbf{x}$  is always defined as a row vector. Consequently, the Hessian of the symmetric quadratic form results in

$$\frac{\partial^2 (\mathbf{x}^T \mathbf{A} \mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^T} = 2 \mathbf{A} . \quad (\text{A.13})$$

The matrix inversion lemma, also known as Woodbury formula, points out that

$$(\mathbf{A} \pm \mathbf{U} \mathbf{B} \mathbf{V}^T)^{-1} = \mathbf{A}^{-1} \mp \mathbf{A}^{-1} \mathbf{U} (\mathbf{B}^{-1} \pm \mathbf{V}^T \mathbf{A}^{-1} \mathbf{U})^{-1} \mathbf{V}^T \mathbf{A}^{-1} \quad (\text{A.14})$$

holds true for regular matrices  $\mathbf{A} \in \mathbb{R}^{n \times n}$  and  $\mathbf{B} \in \mathbb{R}^{m \times m}$  with  $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{n \times m}$ . Furthermore, this leads to the following identity for determinants

$$|\mathbf{A} \pm \mathbf{U} \mathbf{B} \mathbf{V}^T| = |\mathbf{A}| |\mathbf{B}| |\mathbf{B}^{-1} \pm \mathbf{V}^T \mathbf{A}^{-1} \mathbf{U}| . \quad (\text{A.15})$$

### A.3.2 Multivariate Normal Distribution

A  $n$ -dimensional random vector  $\mathbf{x}$  is said to be multivariate normal, or Gaussian, distributed, if  $\mathbf{x}$  follows the probability density function

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{|2\pi \boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right) , \quad (\text{A.16})$$

where  $\boldsymbol{\mu} \in \mathbb{R}^n$  is the vector of expectations and  $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n}$  is a symmetric and positive semi-definite covariance matrix. Alternatively, it is a common way to write

$$\mathbf{x} \sim \text{N}(\mathbf{x} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) . \quad (\text{A.17})$$

To sample a Gaussian random vector according to (A.17), it is needed to calculate the lower Cholesky factor  $\mathbf{L}$  of the covariance matrix  $\boldsymbol{\Sigma}$  with respect to (A.1). Thus,

$$\mathbf{x} = \boldsymbol{\mu} + \mathbf{L} \mathbf{z} \quad (\text{A.18})$$

is a Gaussian random vector with the desired distribution. Thereby, the random vector  $\mathbf{z} \in \mathbb{R}^n$  is sampled from the standardized multivariate distribution  $\text{N}(\mathbf{z} | \mathbf{0}, \mathbf{I})$ .

Let  $\mathbf{x} \in \mathbb{R}^n$  be a multivariate Gaussian random vector with covariance matrix  $\mathbf{B} \in \mathbb{R}^{n \times n}$ , where its expectation vector depends linearly on  $\mathbf{z} \in \mathbb{R}^m$  induced by the projection with  $\mathbf{P} \in \mathbb{R}^{n \times m}$ . Then it follows that the product

$$\mathcal{N}(\mathbf{x} | \mathbf{P} \mathbf{z}, \mathbf{B}) \mathcal{N}(\mathbf{z} | \mathbf{a}, \mathbf{A}) \propto \mathcal{N}(\mathbf{z} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad (\text{A.19})$$

of the projected Gaussian likelihood for  $\mathbf{x}$  and a Gaussian prior for  $\mathbf{z}$ , i.e. with mean vector  $\mathbf{a} \in \mathbb{R}^m$  and covariance matrix  $\mathbf{A} \in \mathbb{R}^{m \times m}$ , is proportional to another multivariate Gaussian in  $\mathbf{z}$ , where the moments are given by

$$\boldsymbol{\mu} = \boldsymbol{\Sigma}(\mathbf{A}^{-1}\mathbf{a} + \mathbf{P}^T\mathbf{B}^{-1}\mathbf{x}) \in \mathbb{R}^m$$

and

$$\boldsymbol{\Sigma} = (\mathbf{A}^{-1} + \mathbf{P}^T\mathbf{B}^{-1}\mathbf{P})^{-1} \in \mathbb{R}^{m \times m} .$$

Of course, for a  $n$ -dimensional random vector  $\mathbf{x} \sim \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, \boldsymbol{\Sigma})$  it holds true that

$$\int_{\mathbb{R}^n} \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) \partial \mathbf{x} = 1 . \quad (\text{A.20})$$

The integral over two Gaussian densities, where the expectation vector of the random variable  $\mathbf{x} \in \mathbb{R}^n$  depends again linearly on  $\mathbf{z} \in \mathbb{R}^m$  over the projection matrix  $\mathbf{P} \in \mathbb{R}^{n \times m}$ , results in the multivariate Gaussian density

$$\int_{\mathbb{R}^m} \mathcal{N}(\mathbf{x} | \mathbf{b} + \mathbf{P}\mathbf{z}, \mathbf{B}) \mathcal{N}(\mathbf{z} | \mathbf{a}, \mathbf{A}) \partial \mathbf{z} = \mathcal{N}(\mathbf{x} | \mathbf{b} + \mathbf{P}\mathbf{a}, \mathbf{B} + \mathbf{P}\mathbf{A}\mathbf{P}^T) . \quad (\text{A.21})$$

Hereby, the shift vector  $\mathbf{b} \in \mathbb{R}^n$ , the mean vector  $\mathbf{a} \in \mathbb{R}^m$ , and the covariance matrices  $\mathbf{A} \in \mathbb{R}^{m \times m}$  and  $\mathbf{B} \in \mathbb{R}^{n \times n}$  are previously given.

Furthermore, let  $\mathbf{x} \in \mathbb{R}^n$  and  $\mathbf{y} \in \mathbb{R}^m$  be Gaussian random vectors with the jointly multivariate normal distribution

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \sim \mathcal{N} \left( \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \middle| \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix}, \begin{pmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C}^T & \mathbf{B} \end{pmatrix} \right) ,$$

where the moments described by  $\mathbf{a} \in \mathbb{R}^n$ ,  $\mathbf{b} \in \mathbb{R}^m$ ,  $\mathbf{A} \in \mathbb{R}^{n \times n}$ ,  $\mathbf{B} \in \mathbb{R}^{m \times m}$ , and  $\mathbf{C} \in \mathbb{R}^{n \times m}$ . Then, the conditional Gaussian density of  $\mathcal{P}(\mathbf{y} | \mathbf{x})$  results in

$$\mathcal{p}(\mathbf{y} | \mathbf{x}) = \mathcal{N}(\mathbf{y} | \mathbf{b} + \mathbf{C}^T\mathbf{A}^{-1}(\mathbf{x} - \mathbf{a}), \mathbf{B} - \mathbf{C}^T\mathbf{A}^{-1}\mathbf{C}) . \quad (\text{A.22})$$

The entropy  $\mathbb{H}[\mathbf{x}]$  of a continuous probability distribution  $\mathcal{P}(\mathbf{x})$  with respect to a random vector  $\mathbf{x} \in \mathbb{R}^n$ , also named as differential entropy of  $\mathbf{x}$ , is defined as

$$\mathbb{H}[\mathbf{x}] = - \int_{\mathbb{R}^n} \mathcal{p}(\mathbf{x}) \log(\mathcal{p}(\mathbf{x})) \partial \mathbf{x} . \quad (\text{A.23})$$

If the random vector  $\mathbf{x}$  is distributed according to a multivariate Gaussian, i.e.  $\mathbf{x} \sim \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, \boldsymbol{\Sigma})$ , the differential entropy results in

$$\mathbb{H}[\mathbf{x}] = \frac{1}{2} \log(|2\pi e \boldsymbol{\Sigma}|) . \quad (\text{A.24})$$

The Kullback-Leibler (KL) divergence

$$\text{KL}[\mathbf{P}(\mathbf{x}) \parallel \mathbf{Q}(\mathbf{x})] = \int_{\mathbb{R}^n} p(\mathbf{x}) \log \left( \frac{p(\mathbf{x})}{q(\mathbf{x})} \right) d\mathbf{x}, \quad (\text{A.25})$$

analogously mentioned as relative entropy or information gain, is a non-negative dissimilarity measure between two probability distributions  $\mathbf{P}(\mathbf{x})$  and  $\mathbf{Q}(\mathbf{x})$  of a random vector  $\mathbf{x} \in \mathbb{R}^n$ . It holds true, that  $\text{KL}[\mathbf{P}(\mathbf{x}) \parallel \mathbf{Q}(\mathbf{x})] = 0$ , if and only if  $p(\mathbf{x}) = q(\mathbf{x})$  for all  $\mathbf{x}$ . Moreover, if  $\mathbf{P}(\mathbf{x}) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$  and  $\mathbf{Q}(\mathbf{x}) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)$  are both multivariate Gaussian distributions, the KL divergence between them can be expressed as

$$\text{KL}[\mathbf{P}(\mathbf{x}) \parallel \mathbf{Q}(\mathbf{x})] = \frac{1}{2} \log \left( \left| \boldsymbol{\Sigma}_q \boldsymbol{\Sigma}_p^{-1} \right| \right) + \frac{1}{2} \text{trace} \left( \boldsymbol{\Sigma}_q^{-1} \left( (\boldsymbol{\mu}_q - \boldsymbol{\mu}_p)(\boldsymbol{\mu}_q - \boldsymbol{\mu}_p)^T + \boldsymbol{\Sigma}_p - \boldsymbol{\Sigma}_q \right) \right). \quad (\text{A.26})$$

Additionally, the one-dimensional truncated normal distribution, the so-called rectified Gaussian distribution, is introduced. A random variable  $x$  is said to be truncated normal distributed, if  $x$  underlies the probability density function

$$\mathcal{R}(x \mid \mu, \sigma^2, l, u) = \frac{\mathbb{I}(l < x < u) \mathcal{N}(x \mid \mu, \sigma^2)}{\Phi\left(\frac{u-\mu}{\sigma}\right) - \Phi\left(\frac{l-\mu}{\sigma}\right)} \quad (\text{A.27})$$

with mean  $\mu \in \mathbb{R}$  and variance  $\sigma^2 > 0$  according to the lower and upper bound  $l \in \mathbb{R}$  and  $u \in \mathbb{R}$ . Furthermore, the lower truncated Gaussian density is defined by

$$\lim_{u \rightarrow \infty} \left( \mathcal{R}(x \mid \mu, \sigma^2, l, u) \right) = \mathcal{L}(x \mid \mu, \sigma^2, l) = \frac{\mathbb{I}(l < x) \mathcal{N}(x \mid \mu, \sigma^2)}{1 - \Phi\left(\frac{l-\mu}{\sigma}\right)} \quad (\text{A.28})$$

as well as the upper truncated normal density

$$\lim_{l \rightarrow -\infty} \left( \mathcal{R}(x \mid \mu, \sigma^2, l, u) \right) = \mathcal{U}(x \mid \mu, \sigma^2, u) = \frac{\mathbb{I}(x < u) \mathcal{N}(x \mid \mu, \sigma^2)}{\Phi\left(\frac{u-\mu}{\sigma}\right)}. \quad (\text{A.29})$$

Note that the class of double-sided truncated Gaussian distributions contains the special case of a standard normal distribution since

$$\lim_{\substack{l \rightarrow -\infty \\ u \rightarrow \infty}} \left( \mathcal{R}(x \mid \mu, \sigma^2, l, u) \right) = \mathcal{N}(x \mid \mu, \sigma^2).$$

The expectation value of a rectified Gaussian random variable  $x$  is given by

$$\mathbb{E}[x \mid \mu, \sigma^2, l, u] = \mu + \sigma \frac{\mathcal{N}\left(\frac{l-\mu}{\sigma}\right) - \mathcal{N}\left(\frac{u-\mu}{\sigma}\right)}{\Phi\left(\frac{u-\mu}{\sigma}\right) - \Phi\left(\frac{l-\mu}{\sigma}\right)} \quad (\text{A.30})$$

and the variance results in

$$\text{Var}[x \mid \mu, \sigma^2, l, u] = \sigma^2 \left( 1 + \frac{\frac{l-\mu}{\sigma} \mathcal{N}\left(\frac{l-\mu}{\sigma}\right) - \frac{u-\mu}{\sigma} \mathcal{N}\left(\frac{u-\mu}{\sigma}\right)}{\Phi\left(\frac{u-\mu}{\sigma}\right) - \Phi\left(\frac{l-\mu}{\sigma}\right)} - \left( \frac{\mathcal{N}\left(\frac{l-\mu}{\sigma}\right) - \mathcal{N}\left(\frac{u-\mu}{\sigma}\right)}{\Phi\left(\frac{u-\mu}{\sigma}\right) - \Phi\left(\frac{l-\mu}{\sigma}\right)} \right)^2 \right), \quad (\text{A.31})$$

see Herbrich (2005) for details of the derivation.

### A.3.3 Derivations

This section provides some derivations and technically detailed information about the considered GP models. The section is structured dependent on the underlying GP technique.

## Gaussian Process Regression

The marginal likelihood of the standard GP model (2.6) is described by the density

$$\begin{aligned} p(\mathbf{y} | \mathbf{X}, \boldsymbol{\theta}) &= \mathcal{N}(\mathbf{y} | \mathbf{m}, \sigma^2 \mathbf{I} + \mathbf{K}) \\ &= \frac{1}{\sqrt{|2\pi(\sigma^2 \mathbf{I} + \mathbf{K})|}} \exp\left(-\frac{1}{2}(\mathbf{y} - \mathbf{m})^T (\sigma^2 \mathbf{I} + \mathbf{K})^{-1} (\mathbf{y} - \mathbf{m})\right) \end{aligned}$$

according to the definition (A.16). Thus, it follows the logarithmic marginal likelihood

$$\begin{aligned} \varphi(\boldsymbol{\theta}) &= -\log\left(\sqrt{|2\pi(\sigma^2 \mathbf{I} + \mathbf{K})|}\right) - \frac{1}{2}(\mathbf{y} - \mathbf{m})^T (\sigma^2 \mathbf{I} + \mathbf{K})^{-1} (\mathbf{y} - \mathbf{m}) \\ &= -\frac{n}{2} \log(2\pi) - \frac{1}{2} \log(|\mathbf{L}\mathbf{L}^T|) - \frac{1}{2}(\mathbf{y} - \mathbf{m})^T \boldsymbol{\alpha} \\ &= -\frac{n}{2} \log(2\pi) - \sum_{i=1}^n \log(l_{ii}) - \frac{1}{2}(\mathbf{y} - \mathbf{m})^T \boldsymbol{\alpha} \end{aligned} \quad (\text{A.32})$$

with the definition of  $\boldsymbol{\alpha}$  given in (2.8), the Cholesky decomposition of  $\sigma^2 \mathbf{I} + \mathbf{K}$  according to (A.1) and the further used relationship (A.2).

## Gaussian Process Classification

The approximated logarithmic marginal likelihood of the GP classification model induced by the Laplace approximation from Equation (2.34) is given by

$$\begin{aligned} \psi(\boldsymbol{\theta}) &= \Psi(\boldsymbol{\mu}) + \log\left(\int_{\mathbb{R}^n} \exp\left(-\frac{1}{2}(\mathbf{g} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{g} - \boldsymbol{\mu})\right) \partial \mathbf{g}\right) \\ &= \sum_{i=1}^n \log(\Phi(c_i \mu_i)) - \frac{1}{2} \log(|2\pi \mathbf{K}|) - \frac{1}{2}(\boldsymbol{\mu} - \mathbf{m})^T \mathbf{K}^{-1} (\boldsymbol{\mu} - \mathbf{m}) + \frac{1}{2} \log(|2\pi \boldsymbol{\Sigma}|) \\ &= \sum_{i=1}^n \log(\Phi(c_i \mu_i)) - \frac{1}{2} \log(|\mathbf{K}| |\boldsymbol{\Sigma}^{-1}|) - \frac{1}{2}(\boldsymbol{\mu} - \mathbf{m})^T \boldsymbol{\alpha} \\ &= \sum_{i=1}^n \log(\Phi(c_i \mu_i)) - \sum_{i=1}^n \log(l_{ii}) - \frac{1}{2}(\boldsymbol{\mu} - \mathbf{m})^T \boldsymbol{\alpha} \end{aligned} \quad (\text{A.33})$$

where (A.15) is used to show

$$|\mathbf{K}| |\boldsymbol{\Sigma}^{-1}| = |\mathbf{K}| |\mathbf{W} + \mathbf{K}^{-1}| = |\mathbf{I} + \mathbf{W}^{\frac{1}{2}} \mathbf{K} \mathbf{W}^{\frac{1}{2}}| = |\mathbf{L}|^2$$

with the Cholesky factor  $\mathbf{L}$  from Equation (2.20). In this case the logarithm of the integral can be easily handled by scaling to a multivariate Gaussian which simplifies in

$$\log\left(\int_{\mathbb{R}^n} \exp\left(-\frac{1}{2}(\mathbf{g} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{g} - \boldsymbol{\mu})\right) \partial \mathbf{g}\right) = \log\left(\sqrt{|2\pi \boldsymbol{\Sigma}|} \int_{\mathbb{R}^n} \mathcal{N}(\mathbf{g} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) \partial \mathbf{g}\right) = \frac{1}{2} \log(|2\pi \boldsymbol{\Sigma}|)$$

with the identity (A.20).

## Deterministic Training Conditional Approximation

For the approximated logarithmic marginal likelihood (3.10) of the DTC approximation follows with the approximated marginal likelihood

$$\begin{aligned} q_I(\mathbf{y} | \mathbf{X}, \boldsymbol{\theta}) &= \mathcal{N}(\mathbf{y} | \mathbf{0}, \sigma^2 \mathbf{I} + \mathbf{V}^T \mathbf{V}) \\ &= \frac{1}{\sqrt{|2\pi(\sigma^2 \mathbf{I} + \mathbf{V}^T \mathbf{V})|}} \exp\left(-\frac{1}{2} \mathbf{y}^T (\sigma^2 \mathbf{I} + \mathbf{V}^T \mathbf{V})^{-1} \mathbf{y}\right) \end{aligned}$$

as defined in Equation (3.7), the manipulation of

$$(\sigma^2 \mathbf{I} + \mathbf{V}^T \mathbf{V})^{-1} = \frac{1}{\sigma^2} \mathbf{I} - \frac{1}{\sigma^2} \mathbf{V}^T (\sigma^2 \mathbf{I} + \mathbf{V} \mathbf{V}^T)^{-1} \mathbf{V} = \frac{1}{\sigma^2} (\mathbf{I} - \mathbf{V}^T \mathbf{M}^{-1} \mathbf{V})$$

with the matrix inversion lemma (A.14) equivalently to (3.2), and the transformation of

$$|\sigma^2 \mathbf{I} + \mathbf{V}^T \mathbf{V}| = |\sigma^2 \mathbf{I}| |\sigma^{-2} \mathbf{I}| |\sigma^2 \mathbf{I} + \mathbf{V} \mathbf{V}^T| = (\sigma^2)^{(n-m)} |\mathbf{M}|$$

with (A.15) that finally

$$\begin{aligned} \varphi_I(\boldsymbol{\theta}) &= -\log\left(\sqrt{(2\pi\sigma^2)^{(n-m)} |2\pi\mathbf{M}|}\right) - \frac{\mathbf{y}^T (\mathbf{I} - \mathbf{V}^T \mathbf{M}^{-1} \mathbf{V})^{-1} \mathbf{y}}{2\sigma^2} \\ &= (m-n) \log(\sigma) - \frac{n}{2} \log(2\pi) - \frac{1}{2} \log(|\mathbf{L}_M \mathbf{L}_M^T|) - \frac{\mathbf{y}^T \mathbf{y} - \mathbf{y}^T \mathbf{V}^T \mathbf{L}_M^{-T} \mathbf{L}_M^{-1} \mathbf{V} \mathbf{y}}{2\sigma^2} \\ &= (m-n) \log(\sigma) - \frac{n}{2} \log(2\pi) - \sum_{i=1}^m \log(l_{M,ii}) - \frac{\mathbf{y}^T \mathbf{y} - \boldsymbol{\beta}_I^T \boldsymbol{\beta}_I}{2\sigma^2}, \end{aligned} \quad (\text{A.34})$$

where the definitions of vectors, matrices, and Cholesky factors as introduced in Section 3.2 are extensively used.

For the inclusion of a selected remaining point  $\mathbf{x}_i$  with  $i \in R$  according to the DTC approximation in Section 3.2, it is mainly necessary to update the Cholesky factors  $\mathbf{L}$  and  $\mathbf{L}_M$ . Therefore, let  $I' = I \cup \{i\}$  with  $|I'| = m' = m + 1$  and  $R' = R \setminus \{i\}$  denote the updated variables. Considering

$$\mathbf{K}_{I',I'} = \mathbf{L}' \mathbf{L}'^T = \begin{pmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{l}^T & l_{ii} \end{pmatrix} \begin{pmatrix} \mathbf{L}^T & \mathbf{l} \\ \mathbf{0}^T & l_{ii} \end{pmatrix} = \begin{pmatrix} \mathbf{K}_{I,I} & \mathbf{k}_{I,i} \\ \mathbf{k}_{I,i}^T & k_{ii} \end{pmatrix}$$

which leads to  $\mathbf{l} = \mathbf{L}^{-1} \mathbf{k}_{I,i}$  and  $l_{ii} = \sqrt{k_{ii} - \mathbf{l}^T \mathbf{l}}$  where the vector  $\mathbf{k}_{I,i}$  contains the covariance values between the selected training point and the former active subset  $\mathbf{X}_I$ . Furthermore, from

$$\mathbf{V}' = \begin{pmatrix} \mathbf{V} \\ \mathbf{v}^T \end{pmatrix} = \mathbf{L}'^{-1} \mathbf{K}_{I',N} = \begin{pmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{l}^T & l_{ii} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{K}_{I,N} \\ \mathbf{k}_{N,i}^T \end{pmatrix} = \begin{pmatrix} \mathbf{L}^{-1} & \mathbf{0} \\ -\mathbf{l}_{ii}^{-1} \mathbf{l}^T \mathbf{L}^{-1} & l_{ii}^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{K}_{I,N} \\ \mathbf{k}_{N,i}^T \end{pmatrix} \quad (\text{A.35})$$

follows  $\mathbf{v} = -\mathbf{l}_{ii}^{-1} (\mathbf{k}_{N,i} - \mathbf{V}^T \mathbf{l})$  where  $\mathbf{k}_{N,i}$  is equivalently defined as  $\mathbf{k}_{I,i}$  above. The new Cholesky factor of  $\mathbf{M}'$  results from

$$\mathbf{M}' = \mathbf{L}'_M \mathbf{L}'_M{}^T = \begin{pmatrix} \mathbf{L}_M & \mathbf{0} \\ \mathbf{l}_M^T & l_{M,ii} \end{pmatrix} \begin{pmatrix} \mathbf{L}_M^T & \mathbf{l}_M \\ \mathbf{0}^T & l_{M,ii} \end{pmatrix} = \begin{pmatrix} \mathbf{M} & \mathbf{V} \mathbf{v} \\ \mathbf{v}^T \mathbf{V}^T & \sigma^2 + \mathbf{v}^T \mathbf{v} \end{pmatrix}$$



with  $\mathbf{l}_M = \mathbf{L}_M^{-1} \mathbf{V} \mathbf{v}$  and  $l_{M,ii} = \sqrt{\sigma^2 + \mathbf{v}^T \mathbf{v} - \mathbf{l}_M^T \mathbf{l}_M}$ . Additional, the updated matrix  $\mathbf{W}$  is given by

$$\begin{aligned} \mathbf{W}' &= \begin{pmatrix} \mathbf{W} \\ \mathbf{w}^T \end{pmatrix} = \mathbf{L}_M'^{-1} \mathbf{V}' = \begin{pmatrix} \mathbf{L}_M & \mathbf{0} \\ \mathbf{l}_M^T & l_{M,ii} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{V} \\ \mathbf{v}^T \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{L}_M^{-1} & \mathbf{0} \\ -l_{M,ii}^{-1} \mathbf{l}_M^T \mathbf{L}_M^{-1} & l_{M,ii}^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{V} \\ \mathbf{v}^T \end{pmatrix} = \begin{pmatrix} \mathbf{W} \\ l_{M,ii}^{-1} (\mathbf{v}^T - \mathbf{l}_M^T \mathbf{W}) \end{pmatrix} \end{aligned}$$

and the new vector  $\beta_{I'}$  results in

$$\beta_{I'} = \mathbf{L}_M'^{-1} \mathbf{V}' \mathbf{y} = \mathbf{W}' \mathbf{y} = \begin{pmatrix} \mathbf{W} \mathbf{y} \\ \mathbf{w}^T \mathbf{y} \end{pmatrix} = \begin{pmatrix} \beta_I \\ \beta_{I',i} \end{pmatrix} = \begin{pmatrix} \beta_I \\ l_{M,ii}^{-1} (\mathbf{v}^T \mathbf{y} - \mathbf{l}_M^T \beta_I) \end{pmatrix}. \quad (\text{A.36})$$

Finally, the updated posterior mean is given by

$$\boldsymbol{\mu}_{I'} = \mathbf{W}'^T \beta_{I'} = \begin{pmatrix} \mathbf{W}^T & \mathbf{w} \end{pmatrix} \begin{pmatrix} \beta_I \\ \beta_{I',i} \end{pmatrix} = \mathbf{W}^T \beta_I + \beta_{I',i} \mathbf{w} = \boldsymbol{\mu}_I + \beta_{I,i} \mathbf{w}. \quad (\text{A.37})$$

### Fully Independent Training Conditional Approximation

The approximated logarithmic marginal likelihood (3.34) of the FITC approximation follows with the approximated marginal likelihood

$$\begin{aligned} q_P(\mathbf{y} | \mathbf{X}_P, \mathbf{X}) &= \mathcal{N}(\mathbf{y} | \mathbf{0}, \mathbf{D} + \mathbf{V}^T \mathbf{V}) \\ &= \frac{1}{\sqrt{|2\pi(\mathbf{D} + \mathbf{V}^T \mathbf{V})|}} \exp\left(-\frac{1}{2} \mathbf{y}^T (\mathbf{D} + \mathbf{V}^T \mathbf{V})^{-1} \mathbf{y}\right) \end{aligned}$$

as defined in Equation (3.31), the manipulation of

$$(\mathbf{D} + \mathbf{V}^T \mathbf{V})^{-1} = \mathbf{D}^{-1} - \mathbf{D}^{-1} \mathbf{V}^T (\mathbf{I} + \mathbf{V} \mathbf{D}^{-1} \mathbf{V}^T)^{-1} \mathbf{V} \mathbf{D}^{-1} = \mathbf{D}^{-1} - \mathbf{D}^{-1} \mathbf{V}^T \mathbf{M}^{-1} \mathbf{V} \mathbf{D}^{-1}$$

with the matrix inversion lemma (A.14), and the transformation of

$$|\mathbf{D} + \mathbf{V}^T \mathbf{V}| = |\mathbf{D}| |\mathbf{I}| |\mathbf{I} + \mathbf{V} \mathbf{D}^{-1} \mathbf{V}^T| = |\mathbf{D}| |\mathbf{M}|$$

with (A.15) yields finally

$$\begin{aligned} \varphi_P(\boldsymbol{\theta}) &= -\log\left(\sqrt{(2\pi)^n |\mathbf{D}| |\mathbf{M}|}\right) - \frac{1}{2} \mathbf{y}^T (\mathbf{D}^{-1} - \mathbf{D}^{-1} \mathbf{V}^T \mathbf{M}^{-1} \mathbf{V} \mathbf{D}^{-1})^{-1} \mathbf{y} \\ &= -\frac{n}{2} \log(2\pi) - \frac{1}{2} \log(|\mathbf{D}|) - \frac{1}{2} \log(|\mathbf{L}_M \mathbf{L}_M^T|) - \frac{1}{2} (\mathbf{y}^T \mathbf{D}^{-1} \mathbf{y} - \mathbf{y}^T \mathbf{D}^{-1} \mathbf{V}^T \mathbf{L}_M^{-T} \mathbf{L}_M^{-1} \mathbf{V} \mathbf{D}^{-1} \mathbf{y}) \\ &= -\frac{n}{2} \log(2\pi) - \frac{1}{2} \sum_{i=1}^m \log\left(k_{ii} - \|\mathbf{L}_M^{-1} \mathbf{k}_{P,i}\|^2 + \sigma^2\right) - \sum_{i=1}^m \log(l_{M,ii}) - \frac{1}{2} (\mathbf{y}^T \mathbf{D}^{-1} \mathbf{y} - \beta_P^T \beta_P), \end{aligned} \quad (\text{A.38})$$

where the definitions of vectors, matrices, and Cholesky factors as introduced in Section 3.3 are extensively used.

### A.3.4 Differentiations

Since various differentiations are commonly used throughout the thesis, this section gives more insights into their derivation. This section is subdivided into topics corresponding to their function.

## Gaussian Process Regression

The gradient of the logarithmic marginal likelihood (2.32) is essential for enabling CG optimization algorithms. Firstly, the partial derivative according to the logarithmic noise  $\log(\sigma)$  results in

$$\begin{aligned} \frac{\partial \varphi(\boldsymbol{\theta})}{\partial \log(\sigma)} &= -\frac{1}{2} \frac{\partial}{\partial \log(\sigma)} \left( \log(|\sigma^2 \mathbf{I} + \mathbf{K}|) + \frac{1}{2} (\mathbf{y} - \mathbf{m})^T (\sigma^2 \mathbf{I} + \mathbf{K})^{-1} (\mathbf{y} - \mathbf{m}) \right) \\ &= -\frac{1}{2} \text{trace} \left( \mathbf{L}^{-T} \mathbf{L}^{-1} \frac{\partial (\sigma^2 \mathbf{I} + \mathbf{K})}{\partial \log(\sigma)} \right) \\ &\quad - \frac{1}{2} (\mathbf{y} - \mathbf{m})^T \left( -\mathbf{L}^{-T} \mathbf{L}^{-1} \frac{\partial (\sigma^2 \mathbf{I} + \mathbf{K})}{\partial \log(\sigma)} \mathbf{L}^{-T} \mathbf{L}^{-1} \right) (\mathbf{y} - \mathbf{m}) \\ &= -\sigma^2 \text{trace} \left( \mathbf{L}^{-T} \mathbf{L}^{-1} - \boldsymbol{\alpha} \boldsymbol{\alpha}^T \right), \end{aligned} \quad (\text{A.39})$$

where the rules (A.6) and (A.7), the definition of  $\boldsymbol{\alpha}$  and the Cholesky decomposition given by (2.8) are used. The partial derivative

$$\frac{\partial (\sigma^2 \mathbf{I} + \mathbf{K})}{\partial \log(\sigma)} = 2\sigma^2 \mathbf{I},$$

is used in the derivation above with some basic algebraic manipulations. Furthermore, for the hyperparameters  $\boldsymbol{\theta}_m$  of the associated mean function follows from Equation (A.12)

$$\begin{aligned} \frac{\partial \varphi(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_m} &= -\frac{1}{2} \frac{\partial}{\partial \boldsymbol{\theta}_m} (\mathbf{y} - \mathbf{m})^T (\sigma^2 \mathbf{I} + \mathbf{K})^{-1} (\mathbf{y} - \mathbf{m}) \\ &= \boldsymbol{\alpha}^T \frac{\partial \mathbf{m}}{\partial \boldsymbol{\theta}_m}. \end{aligned} \quad (\text{A.40})$$

Here,  $\frac{\partial \mathbf{m}}{\partial \boldsymbol{\theta}_m} \in \mathbb{R}^{n \times |\boldsymbol{\theta}_m|}$  describes the Jacobian of the mean vector. The partial derivatives of the logarithmic marginal likelihood with respect to the hyperparameters  $\boldsymbol{\theta}_k$  of the specified kernel are summarized in

$$\begin{aligned} \frac{\partial \varphi(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k} &= -\frac{1}{2} \frac{\partial}{\partial \boldsymbol{\theta}_k} \left( \log(|\sigma^2 \mathbf{I} + \mathbf{K}|) + \frac{1}{2} (\mathbf{y} - \mathbf{m})^T (\sigma^2 \mathbf{I} + \mathbf{K})^{-1} (\mathbf{y} - \mathbf{m}) \right) \\ &= -\frac{1}{2} \text{trace} \left( \mathbf{L}^{-T} \mathbf{L}^{-1} \frac{\partial (\sigma^2 \mathbf{I} + \mathbf{K})}{\partial \boldsymbol{\theta}_k} \right) \\ &\quad - \frac{1}{2} (\mathbf{y} - \mathbf{m})^T \left( -\mathbf{L}^{-T} \mathbf{L}^{-1} \frac{\partial (\sigma^2 \mathbf{I} + \mathbf{K})}{\partial \boldsymbol{\theta}_k} \mathbf{L}^{-T} \mathbf{L}^{-1} \right) (\mathbf{y} - \mathbf{m}) \\ &= -\frac{1}{2} \text{trace} \left( \left( \mathbf{L}^{-T} \mathbf{L}^{-1} - \boldsymbol{\alpha} \boldsymbol{\alpha}^T \right) \frac{\partial \mathbf{K}}{\partial \boldsymbol{\theta}_k} \right). \end{aligned} \quad (\text{A.41})$$

Note that  $\frac{\partial \mathbf{K}}{\partial \boldsymbol{\theta}_k} \in \mathbb{R}^{n \times n \times |\boldsymbol{\theta}_k|}$  is a matrix structure of third order and the output of the trace is in this special case a row vector, i.e. the gradient, where the same mathematical definitions and relations as in Equation (A.39) are used.

Furthermore, the first derivatives of the standard GP predictive expectation value results in

$$\frac{\partial \mathbb{E}[y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}]}{\partial \mathbf{x}_*} = \frac{\partial m_*}{\partial \mathbf{x}_*} + \boldsymbol{\alpha}^T \frac{\partial \mathbf{k}_*}{\partial \mathbf{x}_*}. \quad (\text{A.42})$$

Then, the Hessian reads

$$\frac{\partial^2 \mathbb{E}[y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}]}{\partial \mathbf{x}_* \partial \mathbf{x}_*^T} = \frac{\partial^2 m_*}{\partial \mathbf{x}_* \partial \mathbf{x}_*^T} + \boldsymbol{\alpha}^T \frac{\partial^2 \mathbf{k}_*}{\partial \mathbf{x}_* \partial \mathbf{x}_*^T} \quad (\text{A.43})$$

with respect to the test point  $\mathbf{x}_* \in \mathbb{R}^d$ . The gradient of the predictive variance is given by

$$\frac{\partial \text{Var}[y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}]}{\partial \mathbf{x}_*} = \frac{\partial}{\partial \mathbf{x}_*} \left( k_{**} - \mathbf{k}_*^T \mathbf{L}^{-T} \mathbf{L}^{-1} \mathbf{k}_* \right) = \frac{\partial k_{**}}{\partial \mathbf{x}_*} - 2 \mathbf{k}_*^T \mathbf{L}^{-T} \mathbf{L}^{-1} \frac{\partial \mathbf{k}_*}{\partial \mathbf{x}_*} \quad (\text{A.44})$$

using (A.12), where the gradient  $\frac{\partial k_{**}}{\partial \mathbf{x}_*}$  and Jacobian  $\frac{\partial \mathbf{k}_*}{\partial \mathbf{x}_*}$  are provided in the further section about derivatives of the covariance function. Equivalently, the Hessian of the variance results in

$$\frac{\partial^2 \text{Var}[y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{X}]}{\partial \mathbf{x}_* \partial \mathbf{x}_*^T} = \frac{\partial^2 k_{**}}{\partial \mathbf{x}_* \partial \mathbf{x}_*^T} - 2 \left( \mathbf{L}^{-1} \frac{\partial \mathbf{k}_*}{\partial \mathbf{x}_*} \right)^T \mathbf{L}^{-1} \frac{\partial \mathbf{k}_*}{\partial \mathbf{x}_*} - 2 \mathbf{k}_*^T \mathbf{L}^{-T} \mathbf{L}^{-1} \frac{\partial^2 \mathbf{k}_*}{\partial \mathbf{x}_* \partial \mathbf{x}_*^T}. \quad (\text{A.45})$$

## Gaussian Process Classification

To optimize the approximated logarithmic marginal likelihood  $\psi(\boldsymbol{\theta})$  from Equation (2.35) according to the hyperparameters  $\boldsymbol{\theta}$  of the mean and covariance function, the posterior mean  $\boldsymbol{\mu}$  and therefore  $\mathbf{W}$  are implicitly depending on  $\boldsymbol{\theta}$ . Note that the prediction vector  $\boldsymbol{\alpha} = \mathbf{K}^{-1}(\boldsymbol{\mu} - \mathbf{m})$  and the Cholesky decomposition  $\mathbf{L}\mathbf{L}^T = \mathbf{I} + \mathbf{W}^{\frac{1}{2}} \mathbf{K} \mathbf{W}^{\frac{1}{2}}$  are defined according to Equation (2.20) in Section 2.1.2. Hence, the gradient

$$\frac{\partial \psi(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{\partial}{\partial \boldsymbol{\theta}} \left( \sum_{i=1}^n \log(\Phi(c_i \mu_i)) - \frac{1}{2} \log(|\mathbf{I} + \mathbf{W}^{\frac{1}{2}} \mathbf{K} \mathbf{W}^{\frac{1}{2}}|) - \frac{1}{2} (\boldsymbol{\mu} - \mathbf{m})^T \boldsymbol{\alpha} \right) + \frac{\partial \psi(\boldsymbol{\theta})}{\partial \boldsymbol{\mu}} \frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\theta}} \quad (\text{A.46})$$

results in an explicit and implicit part induced by the generalized chain rule. The explicit part with respect to the hyperparameters of the kernel  $\boldsymbol{\theta}_k$  is given by

$$\begin{aligned} & - \frac{1}{2} \text{trace} \left( \mathbf{L}^{-T} \mathbf{L}^{-1} \frac{\partial (\mathbf{I} + \mathbf{W}^{\frac{1}{2}} \mathbf{K} \mathbf{W}^{\frac{1}{2}})}{\partial \boldsymbol{\theta}_k} \right) - \frac{1}{2} (\boldsymbol{\mu} - \mathbf{m})^T \left( - \mathbf{K}^{-1} \frac{\partial \mathbf{K}}{\partial \boldsymbol{\theta}_k} \mathbf{K}^{-1} \right) (\boldsymbol{\mu} - \mathbf{m}) \\ & = - \frac{1}{2} \text{trace} \left( \mathbf{W}^{\frac{1}{2}} \mathbf{L}^{-T} \mathbf{L}^{-1} \mathbf{W}^{\frac{1}{2}} \frac{\partial \mathbf{K}}{\partial \boldsymbol{\theta}_k} \right) + \frac{1}{2} \boldsymbol{\alpha}^T \frac{\partial \mathbf{K}}{\partial \boldsymbol{\theta}_k} \boldsymbol{\alpha} \end{aligned} \quad (\text{A.47})$$

using (A.6) and (A.7). Applying (A.12) yields the explicit part

$$\boldsymbol{\alpha}^T \frac{\partial \mathbf{m}}{\partial \boldsymbol{\theta}_m} \quad (\text{A.48})$$

according to the mean function hyperparameters  $\boldsymbol{\theta}_m$  equivalent to the gradient in (A.40). Firstly, the gradient of  $\psi(\boldsymbol{\theta})$  with respect to  $\boldsymbol{\mu}$  is given by

$$\begin{aligned} \frac{\partial \psi(\boldsymbol{\theta})}{\partial \boldsymbol{\mu}} &= \left( \bigoplus_{i=1}^n \frac{\partial \log(\Phi(c_i \mu_i))}{\partial \mu_i} \right)^T - \boldsymbol{\alpha}^T - \frac{1}{2} \frac{\partial \log(|\mathbf{I} + \mathbf{W}^{\frac{1}{2}} \mathbf{K} \mathbf{W}^{\frac{1}{2}}|)}{\partial \boldsymbol{\mu}} \\ &= \mathbf{0}^T - \frac{1}{2} \text{trace} \left( \mathbf{L}^{-T} \mathbf{L}^{-1} \frac{\partial (\mathbf{I} + \mathbf{W}^{\frac{1}{2}} \mathbf{K} \mathbf{W}^{\frac{1}{2}})}{\partial \boldsymbol{\mu}} \right) \\ &= - \frac{1}{2} \text{trace} \left( \mathbf{L}^{-T} \mathbf{L}^{-1} \frac{\partial (\mathbf{I} + \mathbf{W}^{\frac{1}{2}} \mathbf{K} \mathbf{W}^{\frac{1}{2}})}{\partial \boldsymbol{\mu}} \right) \\ &= - \frac{1}{2} \text{trace} \left( \mathbf{L}^{-T} \mathbf{L}^{-1} \mathbf{K} \frac{\partial \mathbf{W}}{\partial \boldsymbol{\mu}} \right) \\ &= - \frac{1}{2} \text{trace} \left( \boldsymbol{\Sigma} \frac{\partial \mathbf{W}}{\partial \boldsymbol{\mu}} \right) \end{aligned} \quad (\text{A.49})$$

Therein, the first two terms vanish due to the self-consistent Equation (2.17) and

$$\begin{aligned} \frac{\partial \mathbf{W}}{\partial \boldsymbol{\mu}} &= \text{diag} \left( \bigoplus_{i=1}^n \left( \frac{\partial}{\partial \mu_i} \left( \frac{\mathcal{N}(c_i \mu_i)^2}{\Phi(c_i \mu_i)^2} + \frac{c_i \mu_i \mathcal{N}(c_i \mu_i)}{\Phi(c_i \mu_i)} \right) \right) \right) \\ &= \text{diag} \left( \bigoplus_{i=1}^n \left( -\frac{2c_i \mathcal{N}(c_i \mu_i)^3}{\Phi(c_i \mu_i)^3} - \frac{3c_i^2 \mu_i \mathcal{N}(c_i \mu_i)^2}{\Phi(c_i \mu_i)^2} + c_i (1 - \mu_i^2) \frac{\mathcal{N}(c_i \mu_i)}{\Phi(c_i \mu_i)} \right) \right). \end{aligned}$$

Note that  $\frac{\partial \mathbf{W}}{\partial \boldsymbol{\mu}} \in \mathbb{R}^{n \times n \times n}$  is a diagonal matrix structure of third order, i.e. with only non-zero elements on the spatial diagonal, and the trace in (A.49) results in a row vector analogously to (A.41). The partial derivatives of the mode  $\boldsymbol{\mu}$  according to the hyperparameters of the kernel  $\boldsymbol{\theta}_k$  results in

$$\begin{aligned} \frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\theta}_k} &= \frac{\partial}{\partial \boldsymbol{\theta}_k} \left( \mathbf{K} \left( \bigoplus_{i=1}^n \frac{\partial \log(\Phi(c_i \mu_i))}{\partial \mu_i} \right) + \mathbf{m} \right) \\ &= \frac{\partial \mathbf{K}}{\partial \boldsymbol{\theta}_k} \left( \bigoplus_{i=1}^n \frac{c_i \mathcal{N}(c_i \mu_i)}{\Phi(c_i \mu_i)} \right) - \mathbf{K} \mathbf{W} \frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\theta}_k} \\ &= (\mathbf{I} + \mathbf{K} \mathbf{W})^{-1} \frac{\partial \mathbf{K}}{\partial \boldsymbol{\theta}_k} \left( \bigoplus_{i=1}^n \frac{c_i \mathcal{N}(c_i \mu_i)}{\Phi(c_i \mu_i)} \right) \end{aligned} \quad (\text{A.50})$$

which follows from differentiating the self-consistent Equation (2.17) and using

$$\frac{\partial}{\partial \boldsymbol{\mu}} \left( \bigoplus_{i=1}^n \frac{\partial \log(\Phi(c_i \mu_i))}{\partial \mu_i} \right) = \text{diag} \left( \bigoplus_{i=1}^n \frac{\partial^2 \log(\Phi(c_i \mu_i))}{\partial \mu_i^2} \right) = -\mathbf{W}$$

according to the definition of  $\mathbf{W}$  in (2.18). The gradient of  $\boldsymbol{\mu}$  is derived with respect to the mean function hyperparameters  $\boldsymbol{\theta}_m$  in the same manner which gives

$$\frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\theta}_m} = (\mathbf{I} + \mathbf{K} \mathbf{W})^{-1} \frac{\partial \mathbf{m}}{\partial \boldsymbol{\theta}_m}. \quad (\text{A.51})$$

Summarizing the results from (A.47), (A.49), and (A.50) yields the gradient

$$\begin{aligned} \frac{\partial \psi(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k} &= -\frac{1}{2} \text{trace} \left( (\mathbf{W}^{\frac{1}{2}} \mathbf{L}^{-T} \mathbf{L}^{-1} \mathbf{W}^{\frac{1}{2}} - \boldsymbol{\alpha} \boldsymbol{\alpha}^T) \frac{\partial \mathbf{K}}{\partial \boldsymbol{\theta}_k} \right) \\ &\quad - \frac{1}{2} \text{trace} \left( \boldsymbol{\Sigma} \frac{\partial \mathbf{W}}{\partial \boldsymbol{\mu}} \right) (\mathbf{I} + \mathbf{K} \mathbf{W})^{-1} \frac{\partial \mathbf{K}}{\partial \boldsymbol{\theta}_k} \left( \bigoplus_{i=1}^n \frac{c_i \mathcal{N}(c_i \mu_i)}{\Phi(c_i \mu_i)} \right) \end{aligned} \quad (\text{A.52})$$

of the approximated logarithmic marginal likelihood from (2.35) with respect to the kernel hyperparameters. Analogously, the gradient

$$\frac{\partial \psi(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_m} = \left( \boldsymbol{\alpha}^T - \frac{1}{2} \text{trace} \left( \boldsymbol{\Sigma} \frac{\partial \mathbf{W}}{\partial \boldsymbol{\mu}} \right) (\mathbf{I} + \mathbf{K} \mathbf{W})^{-1} \right) \frac{\partial \mathbf{m}}{\partial \boldsymbol{\theta}_m} \quad (\text{A.53})$$

is obtained with respect to  $\boldsymbol{\theta}_m$  by putting together the explicit part (A.48) and the implicit term given by (A.49) and (A.51).

## Deterministic Training Conditional Approximation

The partial derivatives of the approximated logarithmic marginal likelihood (3.10) with respect to the hyperparameters are essential for enabling gradient-based optimization algorithms. The partial derivative

according to the logarithmic noise  $\log(\sigma)$  results in

$$\frac{\partial \varphi_I(\boldsymbol{\theta})}{\partial \log(\sigma)} = m - n - \sigma^2 \text{trace}(\mathbf{M}^{-1}) + \frac{\mathbf{y}^T \mathbf{y} - \boldsymbol{\beta}_I^T \boldsymbol{\beta}_I}{\sigma^2} - \left\| \mathbf{L}_M^{-T} \boldsymbol{\beta}_I \right\|^2 \quad (\text{A.54})$$

with

$$\frac{\partial \mathbf{M}}{\partial \log(\sigma)} = \frac{\partial (\sigma^2 \mathbf{I} + \mathbf{V} \mathbf{V}^T)}{\partial \log(\sigma)} = 2\sigma^2 \mathbf{I}$$

and

$$\frac{\partial (\boldsymbol{\beta}_I^T \boldsymbol{\beta}_I)}{\partial \log(\sigma)} = \mathbf{y}^T \mathbf{V}^T \left( -\mathbf{M}^{-1} \frac{\partial \mathbf{M}}{\partial \log(\sigma)} \mathbf{M}^{-1} \right) \mathbf{V} \mathbf{y} = -2\sigma^2 \left\| \mathbf{L}_M^{-T} \boldsymbol{\beta}_I \right\|^2.$$

For the regularized version of the approximated logarithmic marginal likelihood  $\text{VAR} \varphi_I(\boldsymbol{\theta})$  by Titsias (2009), see Equation (3.11), the same gradient gives

$$\frac{\partial \text{VAR} \varphi_I(\boldsymbol{\theta})}{\partial \log(\sigma)} = \frac{\partial \varphi_I(\boldsymbol{\theta})}{\partial \log(\sigma)} + \frac{1}{\sigma^2} \text{trace}(\mathbf{K} - \mathbf{V}^T \mathbf{V}). \quad (\text{A.55})$$

To evaluate the gradient related to the kernel hyperparameters  $\boldsymbol{\theta}_k$ , the matrix  $\mathbf{A} = \sigma^2 \mathbf{K}_{I,I} + \mathbf{K}_{I,N} \mathbf{K}_{I,N}^T \in \mathbb{R}^{m \times m}$  is defined and yields

$$\frac{\partial \mathbf{A}}{\partial \boldsymbol{\theta}_k} = \sigma^2 \frac{\partial \mathbf{K}_{I,I}}{\partial \boldsymbol{\theta}_k} + 2 \text{sym} \left( \frac{\partial \mathbf{K}_{I,N}}{\partial \boldsymbol{\theta}_k} \mathbf{K}_{I,N}^T \right)$$

Employing  $\log(|\mathbf{M}|) = \log(|\mathbf{L}^{-1} \mathbf{A} \mathbf{L}^{-T}|) = \log(|\mathbf{A}|) - \log(|\mathbf{K}_{I,I}|)$  and (A.7) gives

$$\begin{aligned} \frac{\partial \log(|\mathbf{M}|)}{\partial \boldsymbol{\theta}_k} &= \text{trace} \left( \mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial \boldsymbol{\theta}_k} \right) - \text{trace} \left( \mathbf{K}_{I,I}^{-1} \frac{\partial \mathbf{K}_{I,I}}{\partial \boldsymbol{\theta}_k} \right) \\ &= \text{trace} \left( \left( \sigma^2 \mathbf{A}^{-1} - \mathbf{K}_{I,I}^{-1} \right) \frac{\partial \mathbf{K}_{I,I}}{\partial \boldsymbol{\theta}_k} \right) + 2 \text{trace} \left( \mathbf{K}_{I,N}^T \mathbf{A}^{-1} \frac{\partial \mathbf{K}_{I,N}}{\partial \boldsymbol{\theta}_k} \right). \end{aligned}$$

Furthermore, it holds true that  $\boldsymbol{\beta}_I^T \boldsymbol{\beta}_I = \mathbf{y}^T \mathbf{K}_{I,N}^T \mathbf{A}^{-1} \mathbf{K}_{I,N} \mathbf{y} = \mathbf{y}^T \mathbf{K}_{I,N}^T \mathbf{b}$  with  $\mathbf{b} = \mathbf{A}^{-1} \mathbf{K}_{I,N} \mathbf{y} \in \mathbb{R}^m$  and therefore

$$\frac{\partial \mathbf{b}}{\partial \boldsymbol{\theta}_k} = -\mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial \boldsymbol{\theta}_k} \mathbf{A}^{-1} \mathbf{K}_{I,N} \mathbf{y} + \mathbf{A}^{-1} \frac{\partial \mathbf{K}_{I,N}}{\partial \boldsymbol{\theta}_k} \mathbf{y} = -\mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial \boldsymbol{\theta}_k} \mathbf{b} + \mathbf{A}^{-1} \frac{\partial \mathbf{K}_{I,N}}{\partial \boldsymbol{\theta}_k} \mathbf{y}.$$

Hence,

$$\begin{aligned} \frac{\partial (\boldsymbol{\beta}_I^T \boldsymbol{\beta}_I)}{\partial \boldsymbol{\theta}_k} &= 2 \mathbf{y}^T \frac{\partial \mathbf{K}_{I,N}^T}{\partial \boldsymbol{\theta}_k} \mathbf{b} - \mathbf{b}^T \frac{\partial \mathbf{A}}{\partial \boldsymbol{\theta}_k} \mathbf{b} \\ &= 2 \mathbf{y}^T \frac{\partial \mathbf{K}_{I,N}^T}{\partial \boldsymbol{\theta}_k} \mathbf{b} - \sigma^2 \mathbf{b}^T \frac{\partial \mathbf{K}_{I,I}}{\partial \boldsymbol{\theta}_k} \mathbf{b} - 2 \mathbf{b}^T \text{sym} \left( \frac{\partial \mathbf{K}_{I,N}}{\partial \boldsymbol{\theta}_k} \mathbf{K}_{I,N}^T \right) \mathbf{b} \\ &= 2 \text{trace} \left( \mathbf{y} \mathbf{b}^T \frac{\partial \mathbf{K}_{I,N}}{\partial \boldsymbol{\theta}_k} \right) - \sigma^2 \text{trace} \left( \mathbf{b} \mathbf{b}^T \frac{\partial \mathbf{K}_{I,I}}{\partial \boldsymbol{\theta}_k} \right) - 2 \text{trace} \left( \mathbf{K}_{I,N}^T \mathbf{b} \mathbf{b}^T \frac{\partial \mathbf{K}_{I,N}}{\partial \boldsymbol{\theta}_k} \right). \end{aligned}$$

is obtained. Finally, the gradient of (3.10) results in

$$\begin{aligned} \frac{\partial \varphi_I(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k} &= \frac{\partial}{\partial \boldsymbol{\theta}_k} \left( -\frac{\log(|\mathbf{M}|)}{2} + \frac{\boldsymbol{\beta}_I^T \boldsymbol{\beta}_I}{2\sigma^2} \right) \\ &= \frac{1}{2} \text{trace} \left( \left( \mathbf{K}_{I,I}^{-1} - \sigma^2 \mathbf{A}^{-1} - \mathbf{b} \mathbf{b}^T \right) \frac{\partial \mathbf{K}_{I,I}}{\partial \boldsymbol{\theta}_k} \right) \\ &\quad - \frac{1}{\sigma^2} \text{trace} \left( \left( \sigma^2 \mathbf{K}_{I,N}^T \mathbf{A}^{-1} - \mathbf{y} \mathbf{b}^T + \mathbf{K}_{I,N}^T \mathbf{b} \mathbf{b}^T \right) \frac{\partial \mathbf{K}_{I,N}}{\partial \boldsymbol{\theta}_k} \right). \quad (\text{A.56}) \end{aligned}$$

Here, the Equations (A.6), (A.8), and (A.9) are used extensively as well as the properties for the trace of a matrix. Analogously, the gradient for the regularized approximated logarithmic marginal likelihood of the variational framework results in

$$\frac{\partial \text{VAR} \varphi_I(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k} = \frac{\partial \varphi_I(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k} - \frac{1}{2\sigma^2} \text{trace} \left( \frac{\partial \mathbf{K}}{\partial \boldsymbol{\theta}_k} + \mathbf{L}^{-T} \mathbf{V} \mathbf{V}^T \mathbf{L}^{-1} \frac{\partial \mathbf{K}_{I,I}}{\partial \boldsymbol{\theta}_k} - 2 \mathbf{V}^T \mathbf{L}^{-1} \frac{\partial \mathbf{K}_{I,N}}{\partial \boldsymbol{\theta}_k} \right). \quad (\text{A.57})$$

## Fully Independent Training Conditional Approximation

The partial derivatives of the approximated logarithmic marginal likelihood (3.34) with respect to the hyperparameters and the pseudo-inputs are essential for enabling gradient-based optimization techniques. The partial derivative according to the logarithmic noise  $\log(\sigma)$  results in

$$\begin{aligned} \frac{\partial \varphi_P(\boldsymbol{\theta})}{\partial \log(\sigma)} &= -\sigma^2 \text{trace} \left( \mathbf{D}^{-1} - \mathbf{D}^{-1} \mathbf{V}^T \mathbf{M}^{-1} \mathbf{V} \mathbf{D}^{-1} \right) \\ &\quad + \sigma^2 \mathbf{y}^T \mathbf{D}^{-2} \mathbf{y} - 2\sigma^2 \mathbf{y}^T \mathbf{D}^{-2} \mathbf{V}^T \mathbf{L}_M^{-T} \boldsymbol{\beta}_P - \sigma^2 \left\| \mathbf{D}^{-1} \mathbf{V}^T \mathbf{L}_M^{-T} \boldsymbol{\beta}_P \right\|^2 \end{aligned} \quad (\text{A.58})$$

with

$$\frac{\partial \mathbf{D}}{\partial \log(\sigma)} = \frac{\partial (\boldsymbol{\Gamma} + \sigma^2 \mathbf{I})}{\partial \log(\sigma)} = 2\sigma^2 \mathbf{I}$$

which yields

$$\frac{\partial \mathbf{M}}{\partial \log(\sigma)} = \frac{\partial (\mathbf{I} + \mathbf{V} \mathbf{D}^{-1} \mathbf{V}^T)}{\partial \log(\sigma)} = -2\sigma^2 \mathbf{V} \mathbf{D}^{-1} \mathbf{D}^{-1} \mathbf{V}^T$$

as well as

$$\begin{aligned} \frac{\partial (\boldsymbol{\beta}_P^T \boldsymbol{\beta}_P)}{\partial \log(\sigma)} &= 2 \mathbf{y}^T \left( -\mathbf{D}^{-1} \frac{\partial \mathbf{D}}{\partial \log(\sigma)} \mathbf{D}^{-1} \right) \mathbf{V}^T \mathbf{L}_M^{-T} \boldsymbol{\beta}_P + \mathbf{y}^T \mathbf{D}^{-1} \mathbf{V}^T \left( -\mathbf{M}^{-1} \frac{\partial \mathbf{M}}{\partial \log(\sigma)} \mathbf{M}^{-1} \right) \mathbf{V} \mathbf{D}^{-1} \mathbf{y} \\ &= -4\sigma^2 \mathbf{y}^T \mathbf{D}^{-2} \mathbf{V}^T \mathbf{L}_M^{-T} \boldsymbol{\beta}_P - 2\sigma^2 \left\| \mathbf{D}^{-1} \mathbf{V}^T \mathbf{L}_M^{-T} \boldsymbol{\beta}_P \right\|^2. \end{aligned}$$

The gradient related to the kernel hyperparameters  $\boldsymbol{\theta}_k$  is evaluated with

$$\begin{aligned} \frac{\partial \mathbf{D}}{\partial \boldsymbol{\theta}_k} &= \frac{\partial \boldsymbol{\Gamma}}{\partial \boldsymbol{\theta}_k} = \frac{\partial}{\partial \boldsymbol{\theta}_k} \text{diag} \left( \text{diag} \left( \mathbf{K} - \mathbf{K}_{P,N}^T \mathbf{K}_{P,P}^{-1} \mathbf{K}_{P,N} \right) \right) \\ &= \text{diag} \left( \text{diag} \left( \frac{\partial \mathbf{K}}{\partial \boldsymbol{\theta}_k} - 2 \text{sym} \left( \mathbf{V}^T \mathbf{L}^{-1} \frac{\partial \mathbf{K}_{P,N}}{\partial \boldsymbol{\theta}_k} \right) + \mathbf{V}^T \mathbf{L}^{-1} \frac{\partial \mathbf{K}_{P,P}}{\partial \boldsymbol{\theta}_k} \mathbf{L}^{-T} \mathbf{V} \right) \right) \end{aligned}$$

which is analogously derived to the regularization term in (A.57) and defining the matrix  $\mathbf{A} = \mathbf{K}_{P,P} + \mathbf{K}_{P,N} \mathbf{D}^{-1} \mathbf{K}_{P,N}^T \in \mathbb{R}^{m \times m}$  with

$$\frac{\partial \mathbf{A}}{\partial \boldsymbol{\theta}_k} = \frac{\partial \mathbf{K}_{P,P}}{\partial \boldsymbol{\theta}_k} + 2 \text{sym} \left( \frac{\partial \mathbf{K}_{P,N}}{\partial \boldsymbol{\theta}_k} \mathbf{D}^{-1} \mathbf{K}_{P,N}^T \right) - \mathbf{K}_{P,N} \mathbf{D}^{-1} \frac{\partial \boldsymbol{\Gamma}}{\partial \boldsymbol{\theta}_k} \mathbf{D}^{-1} \mathbf{K}_{P,N}^T$$

yields  $\log(|\mathbf{M}|) = \log(|\mathbf{L}^{-1} \mathbf{A} \mathbf{L}^{-T}|) = \log(|\mathbf{A}|) - \log(|\mathbf{K}_{P,P}|)$ . Combining these results gives

$$\begin{aligned} \frac{\partial \log(|\mathbf{D}| |\mathbf{M}|)}{\partial \boldsymbol{\theta}_k} &= \text{trace} \left( \mathbf{D}^{-1} \frac{\partial \boldsymbol{\Gamma}}{\partial \boldsymbol{\theta}_k} \right) + \text{trace} \left( \mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial \boldsymbol{\theta}_k} \right) - \text{trace} \left( \mathbf{K}_{P,P}^{-1} \frac{\partial \mathbf{K}_{P,P}}{\partial \boldsymbol{\theta}_k} \right) \\ &= \text{trace} \left( \left( \mathbf{D}^{-1} - \mathbf{D}^{-1} \mathbf{K}_{P,N}^T \mathbf{A}^{-1} \mathbf{K}_{P,N} \mathbf{D}^{-1} \right) \frac{\partial \boldsymbol{\Gamma}}{\partial \boldsymbol{\theta}_k} \right) + \text{trace} \left( \left( \mathbf{A}^{-1} - \mathbf{K}_{P,P}^{-1} \right) \frac{\partial \mathbf{K}_{P,P}}{\partial \boldsymbol{\theta}_k} \right) \\ &\quad + 2 \text{trace} \left( \mathbf{D}^{-1} \mathbf{K}_{P,N}^T \mathbf{A}^{-1} \frac{\partial \mathbf{K}_{P,N}}{\partial \boldsymbol{\theta}_k} \right). \end{aligned}$$

Furthermore,

$$\frac{\partial(\mathbf{y}^T \mathbf{D}^{-1} \mathbf{y})}{\partial \theta_k} = \mathbf{y}^T \left( -\mathbf{D}^{-1} \frac{\partial \mathbf{D}}{\partial \theta_k} \mathbf{D}^{-1} \right) \mathbf{y} = -\text{trace} \left( \mathbf{D}^{-1} \mathbf{y} \mathbf{y}^T \mathbf{D}^{-1} \frac{\partial \mathbf{D}}{\partial \theta_k} \right)$$

is obtained with Equation (A.6). Moreover, it holds true that  $\beta_P^T \beta_P = \mathbf{y}^T \mathbf{D}^{-1} \mathbf{K}_{P,N}^T \mathbf{A}^{-1} \mathbf{K}_{P,N} \mathbf{D}^{-1} \mathbf{y} = \mathbf{y}^T \mathbf{D}^{-1} \mathbf{K}_{P,N}^T \mathbf{b}$  with  $\mathbf{b} = \mathbf{A}^{-1} \mathbf{K}_{P,N} \mathbf{D}^{-1} \mathbf{y} \in \mathbb{R}^m$  and therefore

$$\begin{aligned} \frac{\partial \mathbf{b}}{\partial \theta_k} &= -\mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial \theta_k} \mathbf{A}^{-1} \mathbf{K}_{P,N} \mathbf{D}^{-1} \mathbf{y} + \mathbf{A}^{-1} \frac{\partial \mathbf{K}_{P,N}}{\partial \theta_k} \mathbf{D}^{-1} \mathbf{y} + \mathbf{A}^{-1} \mathbf{K}_{P,N} \left( -\mathbf{D}^{-1} \frac{\partial \mathbf{D}}{\partial \theta_k} \mathbf{D}^{-1} \right) \mathbf{y} \\ &= -\mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial \theta_k} \mathbf{b} + \mathbf{A}^{-1} \frac{\partial \mathbf{K}_{P,N}}{\partial \theta_k} \mathbf{D}^{-1} \mathbf{y} - \mathbf{A}^{-1} \mathbf{K}_{P,N} \mathbf{D}^{-1} \frac{\partial \mathbf{D}}{\partial \theta_k} \mathbf{D}^{-1} \mathbf{y}. \end{aligned}$$

Thus,

$$\begin{aligned} \frac{\partial(\beta_P^T \beta_P)}{\partial \theta_k} &= -2 \mathbf{y}^T \mathbf{D}^{-1} \frac{\partial \mathbf{D}}{\partial \theta_k} \mathbf{D}^{-1} \mathbf{K}_{P,N}^T \mathbf{b} + 2 \mathbf{y}^T \mathbf{D}^{-1} \frac{\partial \mathbf{K}_{P,N}^T}{\partial \theta_k} \mathbf{b} - \mathbf{b}^T \frac{\partial \mathbf{A}}{\partial \theta_k} \mathbf{b} \\ &= -2 \mathbf{y}^T \mathbf{D}^{-1} \frac{\partial \mathbf{D}}{\partial \theta_k} \mathbf{D}^{-1} \mathbf{K}_{P,N}^T \mathbf{b} + 2 \mathbf{y}^T \mathbf{D}^{-1} \frac{\partial \mathbf{K}_{P,N}^T}{\partial \theta_k} \mathbf{b} - \mathbf{b}^T \frac{\partial \mathbf{K}_{P,P}}{\partial \theta_k} \mathbf{b} \\ &\quad - 2 \mathbf{b}^T \text{sym} \left( \frac{\partial \mathbf{K}_{P,N}}{\partial \theta_k} \mathbf{D}^{-1} \mathbf{K}_{P,N}^T \right) \mathbf{b} + \mathbf{b}^T \mathbf{K}_{P,N} \mathbf{D}^{-1} \frac{\partial \mathbf{D}}{\partial \theta_k} \mathbf{D}^{-1} \mathbf{K}_{P,N}^T \mathbf{b} \\ &= -2 \text{trace} \left( \mathbf{D}^{-1} \mathbf{K}_{P,N}^T \mathbf{b} \mathbf{y}^T \mathbf{D}^{-1} \frac{\partial \mathbf{D}}{\partial \theta_k} \right) + 2 \text{trace} \left( \mathbf{D}^{-1} \mathbf{y} \mathbf{b}^T \frac{\partial \mathbf{K}_{P,N}}{\partial \theta_k} \right) - \text{trace} \left( \mathbf{b} \mathbf{b}^T \frac{\partial \mathbf{K}_{P,P}}{\partial \theta_k} \right) \\ &\quad - 2 \text{trace} \left( \mathbf{D}^{-1} \mathbf{K}_{P,N}^T \mathbf{b} \mathbf{b}^T \frac{\partial \mathbf{K}_{P,N}}{\partial \theta_k} \right) + \text{trace} \left( \mathbf{D}^{-1} \mathbf{K}_{P,N}^T \mathbf{b} \mathbf{b}^T \mathbf{K}_{P,N} \mathbf{D}^{-1} \frac{\partial \mathbf{D}}{\partial \theta_k} \right) \end{aligned}$$

follows and finally yields the gradient

$$\begin{aligned} \frac{\partial \varphi_P(\boldsymbol{\theta})}{\partial \theta_k} &= \frac{\partial}{\partial \theta_k} \left( -\frac{\log(|\mathbf{D}||\mathbf{M}|)}{2} - \frac{\mathbf{y}^T \mathbf{D}^{-1} \mathbf{y} - \beta_P^T \beta_P}{2} \right) \\ &= -\frac{1}{2} \text{trace} \left( \mathbf{D}^{-1} \left( \mathbf{D} - \mathbf{K}_{P,N}^T \mathbf{A}^{-1} \mathbf{K}_{P,N} - \mathbf{y} \mathbf{y}^T + 2 \mathbf{K}_{P,N}^T \mathbf{b} \mathbf{y}^T - \mathbf{K}_{P,N}^T \mathbf{b} \mathbf{b}^T \mathbf{K}_{P,N} \right) \mathbf{D}^{-1} \frac{\partial \mathbf{D}}{\partial \theta_k} \right) \\ &\quad + \frac{1}{2} \text{trace} \left( \left( \mathbf{K}_{P,P}^{-1} - \mathbf{A}^{-1} - \mathbf{b} \mathbf{b}^T \right) \frac{\partial \mathbf{K}_{P,P}}{\partial \theta_k} \right) \\ &\quad - \text{trace} \left( \mathbf{D}^{-1} \left( \mathbf{K}_{P,N}^T \mathbf{A}^{-1} - \mathbf{y} \mathbf{b}^T + \mathbf{K}_{P,N}^T \mathbf{b} \mathbf{b}^T \right) \frac{\partial \mathbf{K}_{P,N}}{\partial \theta_k} \right) \end{aligned} \quad (\text{A.59})$$

according to the kernel hyperparameters  $\boldsymbol{\theta}_k$ . For this tedious derivation the relations (A.6), (A.8), and (A.9) as well as the properties for the trace of a matrix have been extensively used. Considering the gradient of  $\varphi_P(\boldsymbol{\theta})$  with respect to a pseudo-input  $\mathbf{x}_P$ , the partial derivatives have to be replaced with  $\frac{\partial \mathbf{K}_{P,P}}{\partial \theta_k}$  to  $\frac{\partial \mathbf{K}_{P,P}}{\partial \mathbf{x}_P}$ ,  $\frac{\partial \mathbf{K}_{P,N}}{\partial \theta_k}$  to  $\frac{\partial \mathbf{K}_{P,N}}{\partial \mathbf{x}_P}$ , and  $\frac{\partial \mathbf{D}}{\partial \theta_k}$  simplifies to

$$\frac{\partial \mathbf{D}}{\partial \mathbf{x}_P} = \text{diag} \left( \text{diag} \left( -2 \text{sym} \left( \mathbf{V}^T \mathbf{L}^{-1} \frac{\partial \mathbf{K}_{P,N}}{\partial \mathbf{x}_P} \right) + \mathbf{V}^T \mathbf{L}^{-1} \frac{\partial \mathbf{K}_{P,P}}{\partial \mathbf{x}_P} \mathbf{L}^{-1} \mathbf{V} \right) \right)$$

in Equation (A.59).

## Safe Active Learning

The approximated logarithmic marginal likelihood  $\psi(\boldsymbol{\theta})$  of the discriminative GP for modeling the safety issue is derived in the same manner as in the standard GPC task under the Laplace approximation, cf.

Equation (A.33), and results in

$$\begin{aligned}\psi(\boldsymbol{\theta}) &= \log(\mathfrak{q}(\mathbf{c}, \mathbf{h} | \mathbf{X}, \boldsymbol{\theta})) = \log(\mathfrak{p}(\mathbf{c}, \mathbf{h} | \mathbf{g}, \mathbf{X}))|_{\mathbf{g}=\boldsymbol{\mu}} - \sum_{i=1}^n \log(l_{ii}) - \frac{1}{2}(\boldsymbol{\mu} - \mathbf{m})^T \boldsymbol{\alpha} \\ &= \sum_{i=1}^k \log(\Phi(c_i \mu_i)) - \frac{l}{2} \log(2\pi\eta^2) - \sum_{j=1}^l \frac{(h_j - \mu_j)^2}{2\eta^2} - \sum_{i=1}^n \log(l_{ii}) - \frac{1}{2}(\boldsymbol{\mu} - \mathbf{m})^T \boldsymbol{\alpha}. \quad (\text{A.60})\end{aligned}$$

Note that the prediction vector  $\boldsymbol{\alpha} = \mathbf{K}^{-1}(\boldsymbol{\mu} - \mathbf{m})$  and the Cholesky decomposition  $\mathbf{L}\mathbf{L}^T = \mathbf{I} + \mathbf{W}^{\frac{1}{2}}\mathbf{K}\mathbf{W}^{\frac{1}{2}}$  is defined according to Equation (4.21) in Section 4.2.3. The diagonal matrix  $\mathbf{W}$  is defined in (4.20) and the covariance matrix  $\mathbf{K}$  results from the GP prior. Analogously to (2.19), the Newton step requires the gradient of  $\log(\mathfrak{p}(\mathbf{c}, \mathbf{h} | \mathbf{g}, \mathbf{X}))$  according to  $\mathbf{g}$  which is given by

$$\frac{\partial \log(\mathfrak{p}(\mathbf{c}, \mathbf{h} | \mathbf{g}, \mathbf{X}))}{\partial \mathbf{g}} = \left( \left( \bigoplus_{i=1}^k \left( \frac{c_i \mathcal{N}(c_i g_i)}{\Phi(c_i g_i)} \right) \right) \oplus \left( \bigoplus_{j=1}^l \left( \frac{h_j - \mu_j}{\eta^2} \right) \right) \right)^T. \quad (\text{A.61})$$

To optimize the approximated logarithmic marginal likelihood  $\psi(\boldsymbol{\theta})$  from (A.60) according to the hyperparameters  $\boldsymbol{\theta}$  of the mean and covariance function, it should be recognized that the posterior mean  $\boldsymbol{\mu}$  and therefore  $\mathbf{W}$  are implicitly depending on  $\boldsymbol{\theta}$ , cf. Equation (A.46). Thus, the gradient

$$\frac{\partial \psi(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{\partial}{\partial \boldsymbol{\theta}} \left( \log(\mathfrak{p}(\mathbf{c}, \mathbf{h} | \mathbf{g}, \mathbf{X}))|_{\mathbf{g}=\boldsymbol{\mu}} - \frac{1}{2} \log(|\mathbf{I} + \mathbf{W}^{\frac{1}{2}}\mathbf{K}\mathbf{W}^{\frac{1}{2}}|) - \frac{1}{2}(\boldsymbol{\mu} - \mathbf{m})^T \boldsymbol{\alpha} \right) + \frac{\partial \psi(\boldsymbol{\theta})}{\partial \boldsymbol{\mu}} \frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\theta}} \quad (\text{A.62})$$

is subdivided in an explicit and implicit part induced by the generalized chain rule. The explicit part with respect to the hyperparameters of the kernel  $\boldsymbol{\theta}_k$  is given by

$$\frac{1}{2} \text{trace} \left( \mathbf{W}^{\frac{1}{2}} \mathbf{L}^{-T} \mathbf{L}^{-1} \mathbf{W}^{\frac{1}{2}} \frac{\partial \mathbf{K}}{\partial \boldsymbol{\theta}_k} \right) + \frac{1}{2} \boldsymbol{\alpha}^T \frac{\partial \mathbf{K}}{\partial \boldsymbol{\theta}_k} \boldsymbol{\alpha} \quad (\text{A.63})$$

and with respect to the mean function hyperparameters  $\boldsymbol{\theta}_m$  by

$$\boldsymbol{\alpha}^T \frac{\partial \mathbf{m}}{\partial \boldsymbol{\theta}_m} \quad (\text{A.64})$$

equivalently to (A.47) and (A.48), respectively. Additionally, the partial derivative of the explicit part with respect to the logarithm of the standard deviation  $\log(\eta)$  results in

$$\frac{\partial \psi(\boldsymbol{\theta})}{\partial \log(\eta)} = -l + \sum_{j=1}^l \frac{(h_j - \mu_j)^2}{\eta^2} - \frac{1}{2} \text{trace} \left( \mathbf{L}^{-T} \mathbf{L}^{-1} \mathbf{K} \frac{\partial \mathbf{W}}{\partial \log(\eta)} \right) \quad (\text{A.65})$$

with

$$\frac{\partial \mathbf{W}}{\partial \log(\eta)} = \text{diag} \left( \mathbf{0} \oplus \left( \bigoplus_{j=1}^l \left( \frac{1}{\eta^2} \right) \right) \right).$$

Note that the implicit part according to  $\log(\eta)$  is equal to zero since  $\frac{\partial \boldsymbol{\mu}}{\partial \log(\eta)}$  vanish. To derive the other implicit parts, firstly the gradient of  $\psi(\boldsymbol{\theta})$  with respect to  $\boldsymbol{\mu}$  is given by

$$\frac{\partial \psi(\boldsymbol{\theta})}{\partial \boldsymbol{\mu}} = -\frac{1}{2} \text{trace} \left( \boldsymbol{\Sigma} \frac{\partial \mathbf{W}}{\partial \boldsymbol{\mu}} \right) \quad (\text{A.66})$$

with

$$\frac{\partial \mathbf{W}}{\partial \boldsymbol{\mu}} = \text{diag} \left( \left( \bigoplus_{i=1}^k \left( -\frac{2c_i \mathcal{N}(c_i \mu_i)^3}{\Phi(c_i \mu_i)^3} - \frac{3c_i^2 \mu_i \mathcal{N}(c_i \mu_i)^2}{\Phi(c_i \mu_i)^2} + c_i (1 - \mu_i^2) \frac{\mathcal{N}(c_i \mu_i)}{\Phi(c_i \mu_i)} \right) \right) \oplus \mathbf{0} \right)$$



which is equivalently derived as Equation (A.51). Again, remark that  $\frac{\partial \mathbf{W}}{\partial \boldsymbol{\mu}} \in \mathbb{R}^{n \times n \times n}$  is a diagonal matrix structure of third order, i.e. with only non-zero elements on the spatial diagonal, and the trace in (A.66) results in a row vector analogously to (A.49). The partial derivatives of the mode  $\boldsymbol{\mu}$  according to the hyperparameters of the kernel  $\boldsymbol{\theta}_k$  results in

$$\frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\theta}_k} = (\mathbf{I} + \mathbf{K}\mathbf{W})^{-1} \frac{\partial \mathbf{K}}{\partial \boldsymbol{\theta}_k} \left( \left( \bigoplus_{i=1}^k \left( \frac{c_i \mathcal{N}(c_i g_i)}{\Phi(c_i g_i)} \right) \right) \oplus \left( \bigoplus_{j=1}^l \left( \frac{h_j - \mu_j}{\eta^2} \right) \right) \right) \quad (\text{A.67})$$

which is derived in the same manner as Equation (A.50). The gradient of  $\boldsymbol{\mu}$  with respect to the mean function hyperparameters  $\boldsymbol{\theta}_m$  gives

$$\frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\theta}_m} = (\mathbf{I} + \mathbf{K}\mathbf{W})^{-1} \frac{\partial \mathbf{m}}{\partial \boldsymbol{\theta}_m}. \quad (\text{A.68})$$

identically to (A.51). Finally, plugging the results from (A.63), (A.66), and (A.67) together yields the gradient

$$\begin{aligned} \frac{\partial \psi(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k} &= -\frac{1}{2} \text{trace} \left( (\mathbf{W}^{\frac{1}{2}} \mathbf{L}^{-T} \mathbf{L}^{-1} \mathbf{W}^{\frac{1}{2}} - \boldsymbol{\alpha} \boldsymbol{\alpha}^T) \frac{\partial \mathbf{K}}{\partial \boldsymbol{\theta}_k} \right) \\ &\quad - \frac{1}{2} \text{trace} \left( \boldsymbol{\Sigma} \frac{\partial \mathbf{W}}{\partial \boldsymbol{\mu}} \right) (\mathbf{I} + \mathbf{K}\mathbf{W})^{-1} \frac{\partial \mathbf{K}}{\partial \boldsymbol{\theta}_k} \left( \left( \bigoplus_{i=1}^k \left( \frac{c_i \mathcal{N}(c_i g_i)}{\Phi(c_i g_i)} \right) \right) \oplus \left( \bigoplus_{j=1}^l \left( \frac{h_j - \mu_j}{\eta^2} \right) \right) \right) \end{aligned} \quad (\text{A.69})$$

of the approximated logarithmic marginal likelihood from Equation (A.60) with respect to the kernel hyperparameters. Analogously to (A.53), the gradient

$$\frac{\partial \psi(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_m} = \left( \boldsymbol{\alpha}^T - \frac{1}{2} \text{trace} \left( \boldsymbol{\Sigma} \frac{\partial \mathbf{W}}{\partial \boldsymbol{\mu}} \right) (\mathbf{I} + \mathbf{K}\mathbf{W})^{-1} \right) \frac{\partial \mathbf{m}}{\partial \boldsymbol{\theta}_m} \quad (\text{A.70})$$

with respect to  $\boldsymbol{\theta}_m$  is derived by putting together the explicit part (A.64) and the implicit term given through (A.66) and (A.68).

Furthermore, the first derivatives of the safety constraint from Equation (4.25) results in

$$\begin{aligned} \frac{\partial}{\partial \mathbf{x}_*} \left( \mathbb{E}_{\text{q}} [g_* | \mathbf{x}_*, \mathbf{c}, \mathbf{h}, \mathbf{X}, \boldsymbol{\theta}_g] - \nu \sqrt{\text{Var}_{\text{q}} [g_* | \mathbf{x}_*, \mathbf{c}, \mathbf{h}, \mathbf{X}, \boldsymbol{\theta}_g]} \right) \\ = \frac{\partial m_*}{\partial \mathbf{x}_*} + \boldsymbol{\alpha}^T \frac{\partial \mathbf{k}_*}{\partial \mathbf{x}_*} - \frac{\nu}{2\sigma_{g_*}} \left( \frac{\partial k_{**}}{\partial \mathbf{x}_*} - 2\mathbf{k}_*^T \mathbf{W}^{\frac{1}{2}} \mathbf{L}^{-T} \mathbf{L}^{-1} \mathbf{W}^{\frac{1}{2}} \frac{\partial \mathbf{k}_*}{\partial \mathbf{x}_*} \right) \end{aligned} \quad (\text{A.71})$$

following from the predictive distribution shown in (4.21). Furthermore, the Hessian is given by

$$\begin{aligned} \frac{\partial^2}{\partial \mathbf{x}_* \partial \mathbf{x}_*^T} \left( \mathbb{E}_{\text{q}} [g_* | \mathbf{x}_*, \mathbf{c}, \mathbf{h}, \mathbf{X}, \boldsymbol{\theta}_g] - \nu \sqrt{\text{Var}_{\text{q}} [g_* | \mathbf{x}_*, \mathbf{c}, \mathbf{h}, \mathbf{X}, \boldsymbol{\theta}_g]} \right) \\ = \frac{\partial^2 m_*}{\partial \mathbf{x}_* \partial \mathbf{x}_*^T} + \boldsymbol{\alpha}^T \frac{\partial^2 \mathbf{k}_*}{\partial \mathbf{x}_* \partial \mathbf{x}_*^T} - \frac{\nu}{2\sigma_{g_*}} \left( \frac{\partial^2 k_{**}}{\partial \mathbf{x}_* \partial \mathbf{x}_*^T} - 2 \left( \mathbf{L}^{-1} \mathbf{W}^{\frac{1}{2}} \frac{\partial \mathbf{k}_*}{\partial \mathbf{x}_*} \right)^T \mathbf{L}^{-1} \mathbf{W}^{\frac{1}{2}} \frac{\partial \mathbf{k}_*}{\partial \mathbf{x}_*} \right. \\ \left. - 2\mathbf{k}_*^T \mathbf{W}^{\frac{1}{2}} \mathbf{L}^{-T} \mathbf{L}^{-1} \mathbf{W}^{\frac{1}{2}} \frac{\partial^2 \mathbf{k}_*}{\partial \mathbf{x}_* \partial \mathbf{x}_*^T} \right. \\ \left. - \frac{1}{2\sigma_{g_*}^2} \left( \frac{\partial k_{**}}{\partial \mathbf{x}_*} - 2\mathbf{k}_*^T \mathbf{W}^{\frac{1}{2}} \mathbf{L}^{-T} \mathbf{L}^{-1} \mathbf{W}^{\frac{1}{2}} \frac{\partial \mathbf{k}_*}{\partial \mathbf{x}_*} \right)^T \left( \frac{\partial k_{**}}{\partial \mathbf{x}_*} - 2\mathbf{k}_*^T \mathbf{W}^{\frac{1}{2}} \mathbf{L}^{-T} \mathbf{L}^{-1} \mathbf{W}^{\frac{1}{2}} \frac{\partial \mathbf{k}_*}{\partial \mathbf{x}_*} \right) \right) \end{aligned} \quad (\text{A.72})$$

with respect to the test point  $\mathbf{x}_* \in \mathbb{R}^d$  in nearly the same manner as by (A.43) and (A.45).

## Mean Functions

The differentiation of the constant mean function  $m(\mathbf{x}) = a_0$  given in (2.23) with respect to the hyperparameter  $a_0 \in \mathbb{R}$  is simply given by 1. For a centered GP the differentiation with respect to the hyperparameters of the mean function can be neglected. The linear mean function up to an additive constant as presented in Equation (2.24) yields the partial differentials

$$\frac{\partial m(\mathbf{x})}{\partial a_j} = \begin{cases} 1 & \text{for } j = 0 \\ x_j & \text{for } j \in \{1, \dots, d\} \end{cases} . \quad (\text{A.73})$$

The gradient with respect to the input point  $\mathbf{x}$  is given by

$$\frac{\partial m(\mathbf{x})}{\partial \mathbf{x}} = \mathbf{a}^T \quad (\text{A.74})$$

and also the Hessian by

$$\frac{\partial^2 m(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^T} = \mathbf{0}\mathbf{0}^T . \quad (\text{A.75})$$

## Covariance Functions

For all in this thesis considered covariance functions, i.e. the kernels from Equation (2.26) until (2.30), their partial derivative with respect to  $\log(\sigma_f)$  is given by

$$\frac{\partial k(\mathbf{x}, \mathbf{z})}{\partial \log(\sigma_f)} = 2k(\mathbf{x}, \mathbf{z}) . \quad (\text{A.76})$$

The derivative according to the logarithm of the hyperparameters is considered to enable unconstrained optimization techniques. Hence, for the gradient related to the element-wise logarithm of the length-scales  $\log(\boldsymbol{\Lambda})$  of the squared exponential (SE) covariance function (2.27) with automatic relevance determination (ARD) follows

$$\frac{\partial k_{\text{SEARD}}(\mathbf{x}, \mathbf{z})}{\partial \log(\boldsymbol{\Lambda})} = ((\mathbf{x} - \mathbf{z}) \circ (\mathbf{x} - \mathbf{z}))^T \boldsymbol{\Lambda}^{-2} k_{\text{SEARD}}(\mathbf{x}, \mathbf{z}) \quad (\text{A.77})$$

using the definition of the Hadamard product. The gradient of the same kernel regarding the data point in the first argument results in

$$\frac{\partial k_{\text{SEARD}}(\mathbf{x}, \mathbf{z})}{\partial \mathbf{x}} = -(\mathbf{x} - \mathbf{z})^T \boldsymbol{\Lambda}^{-2} k_{\text{SEARD}}(\mathbf{x}, \mathbf{z}) , \quad (\text{A.78})$$

where the chain rule together with (A.12) was employed. Furthermore, the Hessian is given by

$$\frac{\partial^2 k_{\text{SEARD}}(\mathbf{x}, \mathbf{z})}{\partial \mathbf{x} \partial \mathbf{x}^T} = (\boldsymbol{\Lambda}^{-2} (\mathbf{x} - \mathbf{z}) (\mathbf{x} - \mathbf{z})^T \boldsymbol{\Lambda}^{-2} - \boldsymbol{\Lambda}^{-2}) k_{\text{SEARD}}(\mathbf{x}, \mathbf{z}) . \quad (\text{A.79})$$

If  $\mathbf{z} = \mathbf{x}$ , then it follows

$$\frac{\partial k_{\text{SEARD}}(\mathbf{x}, \mathbf{x})}{\partial \mathbf{x}} = \frac{\partial \sigma_f^2}{\partial \mathbf{x}} = \mathbf{0}^T \quad (\text{A.80})$$

and

$$\frac{\partial^2 k_{\text{SEARD}}(\mathbf{x}, \mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^T} = \mathbf{0} \mathbf{0}^T. \quad (\text{A.81})$$

For the partial derivatives of the neural network covariance function from Equation (2.30)

$$k_{\text{NNARD}}(\mathbf{x}, \mathbf{z}) = \sigma_f^2 \arcsin(\varpi),$$

the following substitutions

$$\varpi = \frac{1 + \mathbf{x}^T \mathbf{\Lambda}^{-2} \mathbf{z}}{\sqrt{\vartheta}}$$

and

$$\vartheta = (2 + \mathbf{x}^T \mathbf{\Lambda}^{-2} \mathbf{x}) (2 + \mathbf{z}^T \mathbf{\Lambda}^{-2} \mathbf{z})$$

are employed to describe the inner gradient

$$\frac{\partial \sqrt{\vartheta}}{\partial \log(\boldsymbol{\lambda})} = - \left( (\mathbf{x} \circ \mathbf{x})^T (2 + \mathbf{z}^T \mathbf{\Lambda}^{-2} \mathbf{z}) + (\mathbf{z} \circ \mathbf{z})^T (2 + \mathbf{x}^T \mathbf{\Lambda}^{-2} \mathbf{x}) \right) \frac{\mathbf{\Lambda}^{-2}}{\sqrt{\vartheta}}$$

with respect to the logarithmic length-scales  $\log(\boldsymbol{\lambda})$  (also element-wise), finally results in

$$\begin{aligned} \frac{\partial k_{\text{NNARD}}(\mathbf{x}, \mathbf{z})}{\partial \log(\boldsymbol{\lambda})} &= \frac{\sigma_f^2 \varpi}{\vartheta \sqrt{1 - \varpi^2}} \left( (\mathbf{x} \circ \mathbf{x})^T (2 + \mathbf{z}^T \mathbf{\Lambda}^{-2} \mathbf{z}) + (\mathbf{z} \circ \mathbf{z})^T (2 + \mathbf{x}^T \mathbf{\Lambda}^{-2} \mathbf{x}) \right) \mathbf{\Lambda}^{-2} \\ &\quad - \frac{2 \sigma_f^2 (\mathbf{x} \circ \mathbf{z})^T \mathbf{\Lambda}^{-2}}{\sqrt{\vartheta} (1 - \varpi^2)}. \end{aligned} \quad (\text{A.82})$$

With the same algebraic manipulations and differentiation rules the gradient

$$\frac{\partial k_{\text{NNARD}}(\mathbf{x}, \mathbf{z})}{\partial \mathbf{x}} = \frac{\sigma_f^2}{\vartheta \sqrt{1 - \varpi^2}} \left( \mathbf{z}^T \mathbf{\Lambda}^{-2} \sqrt{\vartheta} - \mathbf{x}^T \mathbf{\Lambda}^{-2} \varpi (2 + \mathbf{z}^T \mathbf{\Lambda}^{-2} \mathbf{z}) \right) \quad (\text{A.83})$$

and the Hessian

$$\begin{aligned} \frac{\partial^2 k_{\text{NNARD}}(\mathbf{x}, \mathbf{z})}{\partial \mathbf{x} \partial \mathbf{x}^T} &= \frac{\sigma_f^2}{\vartheta \sqrt{1 - \varpi^2}} \left( \varpi (2 + \mathbf{z}^T \mathbf{\Lambda}^{-2} \mathbf{z})^2 \left( \frac{\varpi^2}{1 - \varpi^2} + 2 \right) \frac{\mathbf{\Lambda}^{-2} \mathbf{x} \mathbf{x}^T \mathbf{\Lambda}^{-2}}{\vartheta} - \varpi (2 + \mathbf{z}^T \mathbf{\Lambda}^{-2} \mathbf{z}) \mathbf{\Lambda}^{-2} \right. \\ &\quad \left. - 2 (2 + \mathbf{z}^T \mathbf{\Lambda}^{-2} \mathbf{z}) \left( \frac{\varpi^2}{1 - \varpi^2} + 1 \right) \frac{\text{sym}(\mathbf{\Lambda}^{-2} \mathbf{x} \mathbf{z}^T \mathbf{\Lambda}^{-2})}{\sqrt{\vartheta}} + \frac{\varpi}{1 - \varpi^2} \mathbf{\Lambda}^{-2} \mathbf{z} \mathbf{z}^T \mathbf{\Lambda}^{-2} \right) \end{aligned} \quad (\text{A.84})$$

according to the data point in the first argument is analogously derived as before. Furthermore,

$$\frac{\partial k_{\text{NNARD}}(\mathbf{x}, \mathbf{x})}{\partial \mathbf{x}} = \frac{2 \sigma_f^2 \mathbf{x}^T \mathbf{\Lambda}^{-2}}{\vartheta \sqrt{1 - \varpi^2}} \quad (\text{A.85})$$

and

$$\frac{\partial^2 k_{\text{NNARD}}(\mathbf{x}, \mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^T} = \frac{2 \sigma_f^2}{\vartheta \sqrt{1 - \varpi^2}} \left( \left( \frac{2 \varpi}{\vartheta (1 - \varpi^2)} - \frac{4}{\sqrt{\vartheta}} \right) \mathbf{\Lambda}^{-2} \mathbf{x} \mathbf{x}^T \mathbf{\Lambda}^{-2} + \mathbf{\Lambda}^{-2} \right) \quad (\text{A.86})$$

is equivalently calculated as before for  $\mathbf{z} = \mathbf{x}$ .

### A.3.5 Proofs

To provide a legible thesis with fluently passages the longest and most tedious proofs are skipped to this section. Only the necessary proofs which are helpful for understanding the contributions of our work are shown in the running text.

## Gaussian Process Regression

*Proof of Lemma 2.1.* Beginning with the logarithmic marginal likelihood of the GPR model from Equation (2.32), it holds true that

$$\begin{aligned}\varphi(\boldsymbol{\theta}) &= -\log\left(\sqrt{|2\pi(\sigma^2\mathbf{I} + \mathbf{K})|}\right) - \frac{1}{2}(\mathbf{y} - \mathbf{m})^T (\sigma^2\mathbf{I} + \mathbf{K})^{-1} (\mathbf{y} - \mathbf{m}) \\ &= -\frac{1}{2}\log\left(|2\pi e(\sigma^2\mathbf{I} + \mathbf{K})|\right) - \frac{1}{2}(\mathbf{y} - \mathbf{m})^T \boldsymbol{\alpha} + \frac{n}{2} \\ &= -\mathbb{H}[\mathbf{y} | \mathbf{X}, \boldsymbol{\theta}] - \frac{1}{2}((\mathbf{y} - \mathbf{m})^T \boldsymbol{\alpha} - n)\end{aligned}\tag{A.87}$$

with the definition of the entropy in (A.24) applied on the marginal likelihood (2.6). To proof the lemma, it needs to be shown that the second term in (A.87) equals to zero if  $\boldsymbol{\theta}$  belongs to a (local) optimum. The necessary condition for a maximum of the differentiable logarithmic marginal likelihood is that their gradient becoming zero for  $\boldsymbol{\theta}_{\text{opt}}$ . Hence, it must hold true that

$$\frac{\partial \varphi(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_k} = -\frac{1}{2} \text{trace} \left( \left( \mathbf{L}^{-T} \mathbf{L}^{-1} - \boldsymbol{\alpha} \boldsymbol{\alpha}^T \right) \frac{\partial \mathbf{K}}{\partial \boldsymbol{\theta}_k} \right) \stackrel{!}{=} \mathbf{0}^T$$

for all hyperparameters  $\boldsymbol{\theta}_k$  of the specified covariance function following from Equation (A.41). Especially, for the magnitude  $\sigma_f$  of the considered kernels from Equation (2.26) until (2.30) it follows

$$\begin{aligned}-\frac{1}{2} \text{trace} \left( \left( \mathbf{L}^{-T} \mathbf{L}^{-1} - \boldsymbol{\alpha} \boldsymbol{\alpha}^T \right) \frac{\partial \mathbf{K}}{\partial \log(\sigma_f)} \right) &= -\text{trace} \left( \left( \mathbf{L}^{-T} \mathbf{L}^{-1} - \boldsymbol{\alpha} \boldsymbol{\alpha}^T \right) \mathbf{K} \right) \\ &= -\text{trace} \left( \mathbf{K} (\sigma^2 \mathbf{I} + \mathbf{K})^{-1} (\mathbf{I} - (\mathbf{y} - \mathbf{m}) \boldsymbol{\alpha}^T) \right) \\ &= -\text{trace} \left( \mathbf{I} - (\mathbf{y} - \mathbf{m}) \boldsymbol{\alpha}^T - \sigma^2 \mathbf{L}^{-T} \mathbf{L}^{-1} + \sigma^2 \boldsymbol{\alpha} \boldsymbol{\alpha}^T \right) \\ &= -n + (\mathbf{y} - \mathbf{m})^T \boldsymbol{\alpha} + \sigma^2 \text{trace} \left( \mathbf{L}^{-T} \mathbf{L}^{-1} - \boldsymbol{\alpha} \boldsymbol{\alpha}^T \right)\end{aligned}\tag{A.88}$$

with some mathematical transformations and definitions according to (2.8), and

$$\frac{\partial \mathbf{K}}{\partial \log(\sigma_f)} = 2 \mathbf{K}$$

given by (A.76). Furthermore, exploiting the necessary condition for the partial derivative with respect to the logarithm of the model noise

$$\frac{\partial \varphi(\boldsymbol{\theta})}{\partial \log(\sigma)} = -\sigma^2 \text{trace} \left( \mathbf{L}^{-T} \mathbf{L}^{-1} - \boldsymbol{\alpha} \boldsymbol{\alpha}^T \right) \stackrel{!}{=} 0$$

following from (A.39) yields

$$(\mathbf{y} - \mathbf{m})^T \boldsymbol{\alpha} - n \stackrel{!}{=} 0$$

together with (A.88), and thus to the statement of the lemma for a suspicious optimal set  $\boldsymbol{\theta}_{\text{Opt}}$  after plugging the last result in Equation (A.87).  $\square$

## Safe Active Learning

*Proof of Lemma 4.1.* Firstly, the non-negativity of the differential entropy (4.2) is shown. The covariance matrix  $\mathbf{K}$  is positive semi-definite by definition, and thus has eigenvalues greater than or equal to zero.

Adding  $\sigma^2 \mathbf{I}$  shifts the spectrum of  $\mathbf{K}$  such that each eigenvalue is greater than or equal to  $\sigma^2$ . Together with (A.24) the latter yields

$$\mathbb{H}[\mathbf{y} | \mathbf{X}, \boldsymbol{\theta}_f] = \frac{1}{2} \log \left( |2\pi e (\sigma^2 \mathbf{I} + \mathbf{K})| \right) \geq \frac{n}{2} \log (2\pi e \sigma^2) ,$$

so that the non-negativity assertion immediately follows by  $\sigma^2 \geq (2\pi e)^{-1}$ . As a short remark, the assumption for  $\sigma^2$  can be satisfied by scaling up the regression model (2.2) with a positive constant  $\gamma$ , i.e. considering therefore  $\hat{y}_i = \hat{f}(\mathbf{x}_i) + \hat{\varepsilon}_i$  with  $\hat{y}_i = \gamma y_i$ ,  $\hat{\sigma}_f^2 = \gamma^2 \sigma_f^2$ , and  $\hat{\varepsilon}_i \sim \mathcal{N}(0, \hat{\sigma}^2)$  where  $\hat{\sigma}^2 = \gamma^2 \sigma^2$  fulfilling the constraint. Note that it is assumed that the true hyperparameters  $\boldsymbol{\theta}_f$  are given. To show the monotony, the chain rule for the entropy is employed resulting in

$$\mathbb{H}[y_*, \mathbf{y} | \mathbf{x}_*, \mathbf{X}, \boldsymbol{\theta}_f] = \mathbb{H}[\mathbf{y} | \mathbf{X}, \boldsymbol{\theta}_f] + \mathbb{H}[y_* | \mathbf{x}_*, \mathcal{D}_n, \boldsymbol{\theta}_f] \quad (\text{A.89})$$

for all non-empty data sets  $\mathcal{D}_n = (\mathbf{y}, \mathbf{X})$  with  $n$  data points, cf. Cover and Thomas (2006). Using the assumption  $\sigma^2 \geq (2\pi e)^{-1}$  and the predictive distribution (2.7), it follows

$$\begin{aligned} \mathbb{H}[y_* | \mathbf{x}_*, \mathcal{D}_n, \boldsymbol{\theta}_f] &= \frac{1}{2} \log \left( k_{**} - \mathbf{k}_*^T (\sigma^2 \mathbf{I} + \mathbf{K})^{-1} \mathbf{k}_* + \sigma^2 \right) + \frac{1}{2} \log (2\pi e) \\ &\geq \frac{1}{2} \log (2\pi e \sigma^2) \geq 0 , \end{aligned}$$

where  $k_{**} - \mathbf{k}_*^T (\sigma^2 \mathbf{I} + \mathbf{K})^{-1} \mathbf{k}_*$  is also non-negative since it is the predictive variance of the latent function value  $f(\mathbf{x}_*)$ . Together with Equation (A.89) the monotony assertion follows immediately. For the submodularity, it has to be proven that

$$\mathbb{H}[y_* | \mathbf{x}_*, \mathcal{D}_m, \boldsymbol{\theta}_f] \geq \mathbb{H}[y_* | \mathbf{x}_*, \mathcal{D}_n, \boldsymbol{\theta}_f]$$

for  $1 \leq m \leq n$ . This is simply true due to the diminishing returns property of the differential entropy, cf. Nemhauser et al. (1978), which is analogously used in Guestrin et al. (2005). Finally, note that here the extended definition of submodularity for infinite sets is used.  $\square$

*Proof of Theorem 4.1.* The proof is given by contradiction, which means that it is assumed that  $\nexists n(\epsilon) \in \mathbb{N}$  such that  $\mathbf{x}_{n(\epsilon)} \in B$  for a space-filling exploration strategy. By the definition of active sampling schemes having the space-filling property, the discrepancy of  $\mathbf{X}$  satisfies

$$D(\mathbf{X}) \leq \gamma \frac{\log^d(n)}{n}$$

where  $n$  is the number of input points summarized in  $\mathbf{X}$ . For every  $\epsilon \in (0, 1)$ , there obviously exists an adequately large  $n(\epsilon)$  such that

$$\epsilon \leq \gamma \frac{\log^d(n(\epsilon))}{n(\epsilon)} .$$

On the other hand, the requirements of the theorem induce

$$D(\mathbf{X}) \geq |0 - \mu_d(B)| > \epsilon$$

for the Lebesgue measurable subset  $B$ . Altogether, this yields a contradiction. Moreover, note that it is possible to use a more general measure  $\mu_d$  in further investigations.  $\square$

*Proof of Theorem 4.2.* For the proof of Theorem 4.2, several auxiliary results are presented which yield necessary conditions for the validity of the safety constraint in Equation (4.26). To this end, it is assumed that  $n_0 \in \mathbb{N}$  initial points  $\mathbf{x}_i$ ,  $i \in \{1, \dots, n_0\}$  exist with positive class labels. To prove the theorem, a bound on the necessary number of initializations points  $n_0$  has to be determined such that the safety constraint  $\mu_{g_*} - \nu \sigma_{g_*} \geq 0$  is satisfied for some input point  $\mathbf{x}_* \in \mathbb{X}$ . At first, an explicit representation for  $\mu_{g_*}$  and  $\sigma_{g_*}$  is derived as follows. The first safe initial point  $\mathbf{x}_1$  is given with a positive label. For this and for all following starting points with positive labels, the likelihood (4.16) of the discriminative model consists only of the probit term which factorizes over all  $n_0$  points. Since a stationary covariance function is used, e.g. see (2.27), the mean  $\boldsymbol{\mu}$  of the approximated posterior (4.18) increases as much as possible if  $\mathbf{x} = \mathbf{x}_i$  is always sampled for all  $i \in \{1, \dots, n_0\}$  with label  $c_i = +1$  again and again. Analogously, the variance  $\text{diag}(\boldsymbol{\Sigma})$  decreases maximally. That is why always sampling the input  $\mathbf{x}$  yields a lower bound for the true number of necessary initialization points. Therefore, the mean and the variance of the approximated posterior (4.17) for the discriminative function only depend on  $n_0$  and the hyperparameters  $\boldsymbol{\theta}_g$ . Note that the latter moments denoted by  $\mu_{n_0}$  and  $\sigma_{n_0}^2$  for the sake of notational simplicity. In the subsequent lemma, the representations of  $\mu_{n_0}$  and  $\sigma_{n_0}^2$  are initially specified.

**Lemma A.1.** *Let  $\mathcal{D}_{n_0}$  consist of  $n_0$  times the same initial point  $\mathbf{x}$  with associated positive class labels. For the moments of the discriminative posterior (4.17) calculated with the Laplace approximation on the data set  $\mathcal{D}_{n_0}$ , it holds true that*

$$\mu_{n_0} = n_0 \sigma_g^2 q_{n_0}$$

and

$$\sigma_{n_0}^2 = \frac{\sigma_g^2}{1 + n_0 \sigma_g^2 w_{n_0}},$$

where

$$q_{n_0} = \frac{\mathcal{N}(\mu_{n_0})}{\Phi(\mu_{n_0})}$$

and

$$w_{n_0} = q_{n_0}^2 + q_{n_0} \mu_{n_0}.$$

*Proof of Lemma A.1.* From the Laplace approximation follows that

$$\begin{aligned} \boldsymbol{\mu} &= \mathbf{K} \frac{\partial}{\partial \mathbf{g}} \log(p(\mathbf{c}, \mathbf{h} | \mathbf{g}, \mathbf{X})) \Big|_{\mathbf{g}=\boldsymbol{\mu}} \\ &= \mathbf{K} \mathbf{q} = \sigma_g^2 \mathbf{1} \mathbf{1}^T \mathbf{q} \\ &= n_0 \sigma_g^2 q_{n_0} \mathbf{1}, \end{aligned}$$

where the covariance matrix reduces to  $\mathbf{K} = \sigma_g^2 \mathbf{1} \mathbf{1}^T$ , since  $\mathbf{x}_i = \mathbf{x}_j$  for all  $i, j \in \{1, \dots, n_0\}$ . Besides that, the posterior covariance can be expressed as

$$\boldsymbol{\Sigma} = (\mathbf{W} + \mathbf{K}^{-1})^{-1} = \mathbf{K} - \mathbf{K} (\mathbf{W}^{-1} + \mathbf{K})^{-1} \mathbf{K}$$

with the help of the matrix inversion lemma (A.14). For the variance, we obtain

$$\sigma_{n_0}^2 = \sigma_g^2 - \sigma_g^2 \mathbf{1}^T (w_{n_0}^{-1} \mathbf{I} + \sigma_g^2 \mathbf{1} \mathbf{1}^T)^{-1} \mathbf{1} \sigma_g^2,$$

where  $w_{n_0} = q_{n_0}^2 + q_{n_0} \mu_{n_0}$  follows from Equation (4.20). Since  $q_{n_0}$  has a relationship to the inverse Mill's ratio for Gaussian distributions, using the bounds by Boyd (1959) yields  $0 < w_{n_0} < 1$ . Furthermore, some algebraic basics are needed to simplify the quadratic form in the variance calculation above. The characteristic polynomial of the matrix  $\mathbf{1}\mathbf{1}^T$  results in  $\lambda^{n_0} - n_0 \lambda^{n_0-1}$ , where the representation over the general minors of  $\mathbf{1}\mathbf{1}^T$  are used. Hence,  $w_{n_0}^{-1}\mathbf{I} + \sigma_g^2 \mathbf{1}\mathbf{1}^T$  has the eigenvalues  $w_{n_0}^{-1} + n_0 \sigma_g^2$  and 0 with multiplicity  $n_0 - 1$ . The eigenvalue  $w_{n_0}^{-1} + n_0 \sigma_g^2$  is always positive with the associated eigenvector  $\mathbf{1}$ . Consequently, the inverse of  $w_{n_0}^{-1}\mathbf{I} + \sigma_g^2 \mathbf{1}\mathbf{1}^T$  has then the positive eigenvalue  $(w_{n_0}^{-1} + n_0 \sigma_g^2)^{-1}$  according to the eigenvector  $\mathbf{1}$ . These results are used to write

$$\begin{aligned} \sigma_{n_0}^2 &= \sigma_g^2 - \sigma_g^2 \mathbf{1}^T (w_{n_0}^{-1} + n_0 \sigma_g^2)^{-1} \mathbf{1} \sigma_g^2 \\ &= \sigma_g^2 - \sigma_g^2 \frac{n_0 \sigma_g^2 w_{n_0}}{1 + n_0 \sigma_g^2 w_{n_0}} \\ &= \frac{\sigma_g^2}{1 + n_0 \sigma_g^2 w_{n_0}} , \end{aligned}$$

which induces the expression for the variance of the approximated posterior (4.17) in this framework.  $\square$

Having found explicit representations for  $\mu_{m_0}$  and  $\sigma_{m_0}$  in the special case of deriving a lower bound of  $n_0$ , the safety constraint of our optimization problem can be redefined by

$$\mu_{n_0} - \nu \sigma_{n_0} = n_0 \sigma_g^2 q_{n_0} - \frac{\nu \sigma_g}{\sqrt{1 + n_0 \sigma_g^2 w_{n_0}}} \geq 0 . \quad (\text{A.90})$$

By means of the findings of Lemma A.1, the following lemma yields a lower bound for  $\mu_{n_0}$ , if (A.90) is satisfied.

**Lemma A.2.** *Let  $\mathcal{D}_{n_0}$  consist of  $n_0$  times the same initial point  $\mathbf{x}$  with associated positive class labels. In order for the safety constraint (A.90) to be satisfied, a necessary condition is given by*

$$\mu_{n_0} \geq \frac{1}{2} (\sqrt{1 + 4\nu \sigma_g} - 1) ,$$

where  $\mu_{n_0}$  as well as  $\sigma_g$  are defined as in Lemma A.1 and  $\nu > 0$ .

*Proof of Lemma A.2.* Using the results of Lemma A.1, the safety constraint in (A.90) is rewritten such that

$$\mu_{n_0} \geq \frac{\nu \sigma_g}{\sqrt{1 + n_0 \sigma_g^2 w_{n_0}}} = \frac{\nu \sigma_g}{\sqrt{1 + q_{n_0} \mu_{n_0} + \mu_{n_0}^2}} . \quad (\text{A.91})$$

Next, the polynomial expression in the denominator of (A.91) is bounded by

$$\sqrt{1 + q_{n_0} \mu_{n_0} + \mu_{n_0}^2} \leq \sqrt{1 + 2\mu_{n_0} + \mu_{n_0}^2} \leq 1 + \mu_{n_0} , \quad (\text{A.92})$$

since  $q_{n_0} \leq \sqrt{\frac{2}{\pi}} \leq 2$  and  $\mu_{n_0} > 0$ , where the latter immediately follows by the representation of  $\mu_{n_0}$  given in Lemma A.1. Combining (A.91) and (A.92) induces a polynomial in  $\mu_{n_0}$ , namely,

$$\mu_{n_0}^2 + \mu_{n_0} - \nu \sigma_g \geq 0 .$$

Together with  $\mu_{n_0} > 0$ , the latter finally yields

$$\mu_{n_0} \geq -\frac{1}{2} + \sqrt{\frac{1}{4} + \nu \sigma_g} .$$

□

The previous lemma provides a lower bound for  $\mu_{n_0}$  as a necessary condition for (A.90). Depending on  $\mu_{n_0}$ , a lower bound for the number  $n_0$  of initial points as necessary condition for Equation (A.90) is additionally derived.

**Lemma A.3.** *Let the assumptions of Lemma A.2 hold. To ensure a non-empty safety constraint in the optimization problem (4.26), a necessary condition is given by*

$$n_0 \geq (2\mathcal{N}(\mu_{n_0}))^{-1} \min\left(\frac{\nu}{\sqrt{3}\sigma_g}, \sqrt{\frac{\nu}{\sqrt{3}\sigma_g^3}}\right) .$$

*Proof of Lemma A.3.* At first, the condition (A.90) is considered which yields

$$n_0 \sigma_g \geq \frac{\nu}{\sqrt{q_{n_0}^2 + n_0 \sigma_g^2 q_{n_0}^2 w_{n_0}}} = \frac{\nu}{\sqrt{q_{n_0}^2 + q_{n_0}^3 \mu_{n_0} + q_{n_0}^2 \mu_{n_0}^2}} .$$

Next, our goal is to bound the term under the square root. Note that, due to the findings of Lemma A.1,  $\mu_{n_0}$  is positive, and thus  $q_{n_0} \leq 1$ . To find an adequate bound, the cases  $\mu_{n_0} \in (0, 1]$  and  $\mu_{n_0} > 1$  are separately considered. Let us begin with the first case, i.e.  $\mu_{n_0} \in (0, 1]$ . Then, it follows

$$q_{n_0}^2 + q_{n_0}^3 \mu_{n_0} + q_{n_0}^2 \mu_{n_0}^2 \leq 3q_{n_0}^2 ,$$

as well as

$$n_0 \geq \frac{\nu}{\sqrt{3}\sigma_g} (q_{n_0})^{-1} . \tag{A.93}$$

In the other case, i.e. if  $\mu_{n_0} > 1$ , the bound

$$q_{n_0}^2 + q_{n_0}^3 \mu_{n_0} + q_{n_0}^2 \mu_{n_0}^2 \leq 3q_{n_0}^2 \mu_{n_0}^2 ,$$

is obtained which yields

$$n_0 \sigma_g \geq \frac{\nu}{\sqrt{3} q_{n_0} \mu_{n_0}} .$$

Furthermore, some simple transformations and the choice of  $\mu_{n_0}$  as in Lemma A.1 imply

$$n_0 \geq \sqrt{\frac{\nu}{\sqrt{3}\sigma_g^3}} (q_{n_0})^{-1} . \tag{A.94}$$

Moreover, for all  $\mu_{n_0} \geq 0$ , it holds true that

$$q_{n_0} = \frac{\mathcal{N}(\mu_{n_0})}{\Phi(\mu_{n_0})} \leq 2\mathcal{N}(\mu_{n_0}) ,$$

because  $\mu_{n_0}$  is non-negative and therefore  $\Phi(\mu_{n_0}) \geq \frac{1}{2}$ . The latter result and the minimum of the factors before  $(q_{n_0})^{-1}$  in (A.93) and (A.94) yield the statement of the lemma. □



Finally, all of the previously presented results are summarized to prove the Theorem 4.2. As before, the notations  $\mu_{n_0}$  and  $\sigma_{n_0}^2$  are used for the moments above analogously to Lemma A.1. Considering the explanations in the beginning of the proof for the lower bound, Lemma A.3 yields

$$n_0 \geq (2\mathcal{N}(\mu_{n_0}))^{-1} \min\left(\frac{\nu}{\sqrt{3}\sigma_g}, \sqrt{\frac{\nu}{\sqrt{3}\sigma_g^3}}\right).$$

Together with the necessary condition of Lemma A.2, this induces the assertion of the theorem. Remember that this lower bound is derived by sampling always on the same position  $\mathbf{x}$ . Therefore, the bound holds only true for stationary covariance functions.  $\square$



# List of Figures

2.1	Transient NARX( $p, q$ ) model . . . . .	21
2.2	Multiple-step ahead prediction scheme for the transient NARX( $p, q$ ) model . . . . .	22
3.1	Relation between different approximation techniques for GPR . . . . .	26
3.2	The NMSE convergence trends on the SARCOS test data for all sparse GPR approximations	41
3.3	Complete learning times for the sparse GPR approximations . . . . .	42
3.4	Insertion time with respect to the obtained accuracy level on the SARCOS test data . . . . .	43
3.5	The NMSE convergence trends on the SARCOS test data for all DTC deletion criteria . . . . .	43
3.6	Computing times of the various DTC deletion criteria depending on the current active set size	44
3.7	Deletion time with respect to the obtained NMSE values on the SARCOS test data . . . . .	44
3.8	SARCOS robot arm . . . . .	45
3.9	The NMSE diagrams on the test sets with real SARCOS data and simulated robot data . . . . .	45
3.10	The NMSE diagram on the test set with real robot data from the Barrett WAM arm . . . . .	46
3.11	Feed-forward control scheme with inverse dynamics model for tracking control of the PR2 . . . . .	47
3.12	The torque percentage and tracking errors of the right PR2 arm . . . . .	47
3.13	Tracking performance on the three test trajectories of the right PR2 arm . . . . .	48
3.14	The electronic power steering (EPS) assistance system with paraxially servo unit . . . . .	49
3.15	Power consumption of the EPS assistance system depending on various driving situations . . . . .	50
3.16	The NRMSE convergence trends on the EPS training data for the DTC selection criteria . . . . .	51
3.17	The NMAE convergence trends on the EPS training data for the DTC selection criteria . . . . .	51
3.18	Training times for each selection criteria of the DTC approximation and the SoD approach . . . . .	52
3.19	Estimating the energy consumption of the EPS assistance system with various approaches . . . . .	54
4.1	Partition of the input space into a safe explorable area and an unsafe region . . . . .	62
4.2	Lower bound for the number of necessary initialization points given by Theorem 4.2 . . . . .	71
4.3	Exploration of the cardinal sine function with the mixture of GP priors approach . . . . .	74
4.4	Predictive variance of the mixture of GP priors for the one-dimensional toy example . . . . .	74
4.5	Number of failures calculated under the independence assumption for the toy example . . . . .	76
4.6	Normalized entropy ratios (NERs) for different safety levels on the toy example . . . . .	77
4.7	Final result for safe active learning of a generalized cardinal sine function . . . . .	77
4.8	Illustration of the inverse pendulum hold up problem . . . . .	78
4.9	Number of failures calculated under the independence assumption for the policy search task . . . . .	79
4.10	Normalized entropy ratios (NERs) for different safety levels on the policy search task . . . . .	79
4.11	Visualization of a Sobol design with 1000 points for a bounded 4-dimensional phase space . . . . .	81
4.12	A cubic Bézier curve with four control points in a two-dimensional input space . . . . .	84
4.13	Visualization of the feasible Sobol points for the bounded 4-dimensional phase space . . . . .	87
4.14	First path of trajectories according to the Sobol design from Figure 4.11 . . . . .	92
4.15	Vertical cross section through a simple electromagnetic valve . . . . .	96

4.16 Stimulation of the voltage based on cubic Beziér trajectories . . . . .	97
4.17 Resulting force of the queried voltage trajectory from Figure 4.16 . . . . .	98
4.18 Example for predicted probabilities of failure for the cubic Beziér trajectories . . . . .	98
4.19 Stimulation of the voltage with a ramp-based trajectory design . . . . .	100
4.20 Resulting force of the queried voltage trajectory from Figure 4.19 . . . . .	100
4.21 Resulting distribution of the queried points associated to the design with cubic Beziér curves	101
4.22 Resulting distribution of the queried points associated to the design with ramp trajectories .	101

---

# List of Tables

2.1	Confusion matrix for the binary classification task . . . . .	20
3.1	Sparse model selection for different approximations on the SARCOS training data . . . . .	40
3.2	Training and prediction errors on the EPS data sets . . . . .	53
4.1	Averaged results of our stationary active learning on the one-dimensional toy example . . . . .	75
4.2	Results of the stationary active learning on the policy exploration task from the cart pole . . . . .	78
A.1	Common abbreviations . . . . .	115
A.2	Basic mathematical functions and spaces . . . . .	116
A.3	Vector and matrix operators . . . . .	117
A.4	Statistical and probability theoretical notations . . . . .	118



---

# List of Algorithms

4.1 Stationary safe active learning with GPs . . . . .	69
4.2 Gaussian expectation propagation for lower truncated Gaussians . . . . .	90
4.3 Transient safe active learning with GPs . . . . .	93





# Bibliography

- Abramowitz, M., and Stegun, I. A. (1972). *Handbook of Mathematical Functions* (10th ed.). National Bureau of Standards.
- Albunni, M. N. (2010). *Model Order Reduction of Moving Nonlinear Electromagnetic Devices* (PhD Thesis). Technical University of Munich.
- Alvarez, M., and Lawrence, N. D. (2009). Sparse Convolved Gaussian Processes for Multi-output Regression. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou (Eds.), *Advances in Neural Information Processing Systems* (Vol. 21, pp. 57 – 64).
- Aronszajn, N. (1950). Theory of Reproducing Kernels. *Transactions of the American Mathematical Society*, 68(3), pp. 337 – 404.
- Atlas, L. E., Cohn, D. A., Ladner, R. E., El-Sharkawi, M. A., Marks II, R. J., Aggoune, M. E., and Park, D.-C. (1990). Training Connectionist Networks with Queries and Selective Sampling. In D. S. Touretzky (Ed.), *Advances in Neural Information Processing Systems* (Vol. 2, pp. 566 – 573).
- Auer, P. (2002). Using Confidence Bounds for Exploitation-Exploration Trade-Offs. *Journal of Machine Learning Research*, 3, pp. 397 – 422.
- Bernstein, S. N. (1913). Demonstration of the Theorem of Weierstrass Based on the Calculus of Probabilities. *Communications of the Kharkov Mathematical Society*, 13(2), pp. 1 – 2.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer Verlag.
- Boyd, A. V. (1959). Inequalities for Mill's Ratio. *Reports of Statistical Application Research*, 6, pp. 44 – 46.
- Boyle, P., and Frean, M. (2005). *Multiple Output Gaussian Process Regression* (Technical Report No. CS-TR-05/2). Victoria University of Wellington.

- Băzăvan, E. G., Li, F., and Sminchisescu, C. (2012). Fourier Kernel Learning. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid (Eds.), *Proceedings of the 12th European Conference on Computer Vision*, LNCS (Vol. 7573, pp. 459 – 473).
- Burbidge, R., Rowland, J. J., and King, R. D. (2007). Active Learning for Regression Based on Query by Committee. In H. Yin, P. Tino, E. Corchado, W. Byrne, and X. Yao (Eds.), *Proceedings of the 8th International Conference on Intelligent Data Engineering and Automated Learning*, LNCS (Vol. 4881, pp. 209 – 218).
- Cao, Y., Brubaker, M. A., Fleet, D., and Hertzmann, A. (2013). Efficient Optimization for Sparse Gaussian Process Regression. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems* (Vol. 26, pp. 1097 – 1105).
- Chang, C.-C., and Lin, C.-J. (2011). LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3), pp. 533 – 559.
- Cover, T. M., and Thomas, J. A. (2006). *Elements of Information Theory* (2nd ed.). John Wiley & Sons.
- Csató, L. (2002). *Gaussian Processes – Iterative Sparse Approximations* (PhD Thesis). Aston University Birmingham.
- Csató, L., and Opper, M. (2001). Sparse Representation for Gaussian Process Regression. In T. K. Leen, T. G. Dietterich, and V. Tresp (Eds.), *Advances in Neural Information Processing Systems* (Vol. 13, pp. 444 – 450).
- Cunningham, J. P., Hennig, P., and Lacoste-Julien, S. (2011). *Gaussian Probabilities and Expectation Propagation*. University of Cambridge. Retrieved from <http://arxiv.org/pdf/1111.6832.pdf>
- Damianou, A. C., and Lawrence, N. D. (2013). Deep Gaussian Processes. In C. M. Carvalho, and P. Ravikumar (Eds.), *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics*, JMLR: W&CP (Vol. 31, pp. 207 – 215).
- Deisenroth, M. P., Fox, D., and Rasmussen, C. E. (2015a). Gaussian Processes for Data-

- Efficient Learning in Robotics and Control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2), pp. 408 – 423.
- Deisenroth, M. P., and Ng, J. W. (2015b). Distributed Gaussian Processes. In F. R. Bach, and D. M. Blei (Eds.), *Proceedings of the 32nd International Conference on Machine Learning*, JMLR: W&CP (Vol. 37, pp. 1481 – 1490).
- Drezner, Z. (1994). Computation of the Trivariate Normal Integral. *Mathematics of Computation*, 62(205), pp. 289 – 294.
- Fedorov, V. V. (1972). *Theory of Optimal Experiments*. Academic Press.
- Foster, L., Waagen, A., Aijaz, N., Hurley, M., Luis, A., Rinsky, J., Satyavolu, C., Way, M. J., Gazis, P., and Srivastava, A. (2009). Stable and Efficient Gaussian Process Calculations. *Journal of Machine Learning Research*, 10, pp. 857 – 882.
- Galichet, N., Sebag, M., and Teytaud, O. (2013). Exploration vs Exploitation vs Safety: Risk-Aware Multi-Armed Bandits. In C. S. Ong, and T. B. Ho (Eds.), *Proceedings of the 5th Asian Conference on Machine Learning*, JMLR: W&CP (Vol. 29, pp. 245 – 260).
- Geibel, P. (2001). Reinforcement Learning with Bounded Risk. In C. E. Brodley, and A. P. Danyluk (Eds.), *Proceedings of the 18th International Conference on Machine Learning* (pp. 162 – 169).
- Geiger, C., and Kanzow, C. (1999). *Numerische Verfahren zur Lösung unrestringierter Optimierungsaufgaben* (1st ed.). Springer Verlag.
- Genz, A., and Bretz, F. (2002). Methods for the Computation of Multivariate t-Probabilities. *Journal of Computational and Graphical Statistics*, 11(4), pp. 950 – 971.
- Gijsberts, A. (2011). *Incremental Learning for Robotics with Constant Update Complexity* (PhD Thesis). University of Genoa.
- Gillula, J. H., and Tomlin, C. J. (2011). Guaranteed Safe Online Learning of a Bounded System. In N. M. Amato (Ed.), *Proceedings of the IEEE/RSJ International Con-*

*ference on Intelligent Robots and Systems* (pp. 2979 – 2984).

- Girard, A., Rasmussen, C. E., Quiñonero-Candela, J., and Murray-Smith, R. (2003). Gaussian Process Priors with Uncertain Inputs – Application to Multiple-step Ahead Time Series Forecasting. In S. Becker, S. Thrun, and K. Obermayer (Eds.), *Advances in Neural Information Processing Systems* (Vol. 15, pp. 545 – 552).
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2015). Generative Adversarial Nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems* (Vol. 27, pp. 2672 – 2680).
- Graça, J. V., Ganchev, K., and Taskar, B. (2008). Expectation Maximization and Posterior Constraints. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis (Eds.), *Advances in Neural Information Processing Systems* (Vol. 20, pp. 569 – 576).
- Grimmett, G. R., and Stirzaker, D. R. (2001). *Probability and Random Processes* (3rd ed.). Oxford University Press.
- Guestrin, C., Krause, A., and Singh, A. P. (2005). Near-Optimal Sensor Placements in Gaussian Processes. In L. De Raedt, and S. Wrobel (Eds.), *Proceedings of the 22nd International Conference on Machine Learning*, ACM: ICPS (pp. 265 – 275).
- Gutjahr, T. (2012). *Dynamic System Identification with Gaussian Processes in Model-Based Engine Development* (PhD Thesis). Technical University of Ilmenau.
- Guyon, I., and Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, 3, pp. 1157 – 1182.
- Hanneke, S. (2007). A Bound on the Label Complexity of Agnostic Active Learning. In Z. Ghahramani (Ed.), *Proceedings of the 24th International Conference on Machine Learning*, ACM: ICPS (pp. 353 – 360).
- Hans, A., Schneegaß, D., Schäfer, A. M., and Udluft, S. (2008). Safe Exploration for Reinforcement Learning. In M. Verleysen (Ed.), *Proceedings of the 16th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* (pp. 143 – 148).

- Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer Verlag.
- Haykin, S. O. (2008). *Neural Networks and Learning Machines* (3rd ed.). Prentice Hall.
- Herbrich, R. (2005). *On Gaussian Expectation Propagation*. Microsoft Research. Retrieved from <http://research.microsoft.com/pubs/74554/EP.pdf>
- Hong, X., Mitchell, R. J., Chen, S., Harris, C. J., Li, K., and Irwin, G. W. (2008). Model Selection Approaches for Non-linear System Identification: A Review. *International Journal of Systems Science*, 39(10), pp. 925 – 946.
- Horner, W. G. (1819). A New Method of Solving Numerical Equations of All Orders, by Continuous Approximation. *Philosophical Transactions of the Royal Society of London*, 109, pp. 308 – 335.
- Horrace, W. C. (2005). Some Results on the Multivariate Truncated Normal Distribution. *Journal of Multivariate Analysis*, 94(1), pp. 209 – 221.
- Jaeger, H. (2003). Adaptive Nonlinear System Identification with Echo State Networks. In S. Becker, S. Thrun, and K. Obermayer (Eds.), *Advances in Neural Information Processing Systems* (Vol. 15, pp. 609 – 616).
- Jukna, S. (2011). *Extremal Combinatorics: With Applications in Computer Science* (2nd ed.). Springer Verlag.
- Kaufmann, E., Cappé, O., and Garivier, A. (2012). On Bayesian Upper Confidence Bounds for Bandit Problems. In N. Lawrence, and M. Girolami (Eds.), *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*, JMLR: W&CP (Vol. 22, pp. 592 – 600).
- Keerthi, S. S., and Chu, W. (2006). A Matching Pursuit Approach to Sparse Gaussian Process Regression. In Y. Weiss, B. Schölkopf, and J. C. Platt (Eds.), *Advances in Neural Information Processing Systems* (Vol. 18, pp. 643 – 650).
- Kimeldorf, G. S., and Wahba, G. (1970). A Correspondence Between Bayesian Estimation on Stochastic Processes and Smoothing by Splines. *The Annals of Mathematical*

- Statistics*, 41(2), pp. 495 – 502.
- Ko, C.-W., Lee, J., and Queyranne, M. (1995). An Exact Algorithm for Maximum Entropy Sampling. *Operations Research*, 43(4), pp. 684 – 691.
- Krause, A., and Guestrin, C. (2007). Nonmyopic Active Learning of Gaussian Processes: An Exploration-Exploitation Approach. In Z. Ghahramani (Ed.), *Proceedings of the 24th International Conference on Machine Learning*, ACM: ICPS (pp. 449 – 456).
- Lang, K. J., and Baum, E. B. (1992). Query Learning Can Work Poorly when a Human Oracle is Used. *Proceedings of the International Joint Conference on Neural Networks* (pp. 335 – 340).
- Lawrence, N. D., Seeger, M., and Herbrich, R. (2003). Fast Sparse Gaussian Process Methods: The Informative Vector Machine. In S. Becker, S. Thrun, and K. Obermayer (Eds.), *Advances in Neural Information Processing Systems* (Vol. 15, pp. 625 – 632).
- Lázaro-Gredilla, M., Quiñonero-Candela, J., Rasmussen, C. E., and Figueiras-Vidal, A. R. (2010). Sparse Spectrum Gaussian Process Regression. *Journal of Machine Learning Research*, 11, pp. 1865 – 1881.
- Leontaritis, I. J., and Billings, S. A. (1985). Input-output Parametric Models for Non-linear Systems. *International Journal of Control*, 41(2), pp. 303 – 344.
- Lipschutz, S., and Lipson, M. L. (2013). *Linear Algebra* (5th ed.). McGraw-Hill Professional.
- Mardia, K. V., Kent, J. T., and Bibby, J. M. (1980). *Multivariate Analysis* (1st ed.). Academic Press.
- Markert, H., Schiepe, C., Streichert, F., Meister, U., Diener, R., and Gutjahr, T. (2011). Comparison of Alternative Approaches to Auto-regressive Modelling of Dynamic Systems. In K. Röpke (Ed.), *Proceedings of the 6th Conference on Design of Experiments (DoE) in Engine Development* (pp. 318 – 330).
- Martinetz, T. M., Berkovich, S. G., and Schulten, K. J. (1993). Neural Gas Network for

- Vector Quantization and its Application to Time Series Prediction. *IEEE Transactions on Neural Networks*, 4(4), pp. 558 – 569.
- Matérn, B. (1986). *Spatial Variation* (2nd ed.). Springer Verlag.
- Meister, A. (2015). *Numerik linearer Gleichungssysteme* (2nd ed.). Vieweg Verlag.
- Mercer, J. (1909). Functions of Positive and Negative Type, and their Connection with the Theory of Integral Equations. *Philosophical Transactions of the Royal Society*, 83, pp. 69 – 70.
- Minka, T. P. (2001). Expectation Propagation for Approximate Bayesian Inference. In J. S. Breese, and D. Koller (Eds.), *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence* (pp. 362 – 369).
- Moldovan, T. M., and Abbeel, P. (2012). Safe Exploration in Markov Decision Processes. In J. Langford, and J. Pineau (Eds.), *Proceedings of the 29th International Conference on Machine Learning* (pp. 1711 – 1718).
- Moldovan, T. M., and Abbeel, P. (2013). Risk Aversion in Markov Decision Processes via Near-Optimal Chernoff Bounds. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems* (Vol. 25, pp. 3131 – 3139).
- Moore, E. H. (1935). *General Analysis*. The American Philosophical Society.
- Natarajan, B. K. (1995). Sparse Approximate Solutions to Linear Systems. *SIAM Journal on Computing*, 24(2), pp. 227 – 234.
- Neal, R. M. (1996). *Bayesian Learning for Neural Networks*. Springer Verlag.
- Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. (1978). An Analysis of the Approximations for Maximizing Submodular Set Functions. *Mathematical Programming*, 14(1), pp. 265 – 294.
- Nguyen-Tuong, D., and Peters, J. (2011). Incremental Sparsification for Real-time Online Model Learning. *Neurocomputing*, 74(11), pp. 1859 – 1867.

- Nguyen-Tuong, D., Peters, J., and Seeger, M. (2008). Computed Torque Control with Nonparametric Regression Models. *Proceedings of the American Control Conference* (pp. 212 – 217).
- Nguyen-Tuong, D., Peters, J., and Seeger, M. (2009). Local Gaussian Process Regression for Real Time Online Model Learning and Control. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou (Eds.), *Advances in Neural Information Processing Systems* (Vol. 21, pp. 1193 – 1200).
- Nickisch, H., and Rasmussen, C. E. (2008). Approximations for Binary Gaussian Process Classification. *Journal of Machine Learning Research*, 9, pp. 2035 – 2078.
- Olsen, W. K. (2004). Mean Square Error. In M. Lewis-Beck, A. Bryman, and T. Liao (Eds.), *Encyclopedia of Social Science Research Methods* (pp. 625 – 627).
- Petersen, K. B., and Pedersen, M. S. (2012). *The Matrix Cookbook*. Technical University of Denmark. Retrieved from <http://www2.imm.dtu.dk/pubdb/p.php?3274>
- Polo, F. J. G., and Rebollo, F. F. (2011). Safe Reinforcement Learning in High-Risk Tasks through Policy Improvement. In F. L. Lewis, and D. Vrabie (Eds.), *Proceedings of the IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning* (pp. 76 – 83).
- Press, S. J. (2005). *Applied Multivariate Analysis: Using Bayesian and Frequentist Methods of Inference* (2nd ed.). Dover Publications.
- Quiñonero-Candela, J. (2004). *Learning with Uncertainty – Gaussian Processes and Relevance Vector Machines* (PhD Thesis). Technical University of Denmark.
- Quiñonero-Candela, J., and Rasmussen, C. E. (2005). A Unifying View of Sparse Approximate Gaussian Process Regression. *Journal of Machine Learning Research*, 6, pp. 1939 – 1959.
- Rahimi, A., and Recht, B. (2008). Random Features for Large-scale Kernel Machines. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis (Eds.), *Advances in Neural Information Processing Systems* (Vol. 20, pp. 1177 – 1184).



- Ramakrishnan, N., Bailey-Kellogg, C., Tadepalli, S., and Pandey, V. N. (2005). Gaussian Processes for Active Data Mining of Spatial Aggregates. In H. Kargupta, C. Kamath, J. Srivastava, and A. Goodman (Eds.), *Proceedings of the 5th SIAM International Conference on Data Mining* (pp. 427 – 438).
- Rasmussen, C. E., and Ghahramani, Z. (2001). Occam’s Razor. In T. K. Leen, T. G. Dietterich, and V. Tresp (Eds.), *Advances in Neural Information Processing Systems* (Vol. 13, pp. 294 – 300).
- Rasmussen, C. E., and Quiñonero-Candela, J. (2005). Healing the Relevance Vector Machine by Augmentation. L. De Raedt, and S. Wrobel (Eds.), *Proceedings of the 22nd International Conference on Machine Learning*, ACM: ICPS (pp. 689 – 696).
- Rasmussen, C. E., and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. The MIT Press.
- Salomon, D. (2006). *Curves and Surfaces for Computer Graphics*. Springer Verlag.
- Schillinger, M., Ortelt, B., Hartmann, B., Schreiter, J., Meister, M., Nguyen-Tuong, D., and Nelles, O. (2016). Safe Active Learning of a High Pressure Fuel Supply System. In K. Leiviskä (Ed.), *Proceedings of the 9th EUROSIM Congress on Modelling and Simulation* (pp. 238 – 243).
- Schölkopf, B., Burges, C. J. C., and Smola, A. J. (1998). *Advances in Kernel Methods: Support Vector Learning*. The MIT Press.
- Schölkopf, B., Herbrich, R., and Smola, A. J. (2001). A Generalized Representer Theorem. In D. Helmbold, and B. Williamson (Eds.), *Proceedings of the 14th Annual Conference on Computational Learning Theory*, LNCS (Vol. 2111, pp. 416 – 426).
- Schölkopf, B., and Smola, A. J. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press.
- Schreiter, J., Eberts, M., Nguyen-Tuong, D., and Toussaint, M. (2015d). Safe Exploration for Active Learning with Gaussian Process Models. In J. J. Williams, Y. Abbasi, and F. Doshi-Velez (Eds.), *NIPS Workshop on Machine Learning from and for Adaptive User Technologies: From Active Learning & Experimentation to Optimization &*

*Personalization.*

- Schreiter, J., Englert, P., Nguyen-Tuong, D., and Toussaint, M. (2015b). Sparse Gaussian Process Regression for Compliant, Real-Time Robot Control. In A. Okamura, and S. Hutchinson (Eds.), *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 2586 – 2591).
- Schreiter, J., Markert, H., Hanselmann, M., Nguyen-Tuong, D., and Bohne, C. (2013). Large Scale Transient Data-based Models for the Simulation of Vehicle Power Demand. In K. Röpke (Ed.), *Proceedings of the 7th Conference on Design of Experiments (DoE) in Engine Development* (pp. 176 – 189).
- Schreiter, J., Nguyen-Tuong, D., Eberts, M., Bischoff, B., Markert, H., and Toussaint, M. (2015c). Safe Exploration for Active Learning with Gaussian Processes. In A. Bifet, B. Zadrozny, M. May, F. Bonchi, J. Cardoso, M. Spiliopoulou, R. Gavaldà, and D. Pedreschi (Eds.), *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, LNCS (Vol. 9286, pp. 133 – 149).
- Schreiter, J., Nguyen-Tuong, D., Markert, H., Hanselmann, M., and Toussaint, M. (2015a). Fast Greedy Insertion and Deletion in Sparse Gaussian Process Regression. In M. Verleysen (Ed.), *Proceedings of the 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* (pp. 101 – 106).
- Schreiter, J., Nguyen-Tuong, D., and Toussaint, M. (2016). Efficient Sparsification for Gaussian Process Regression. *Neurocomputing*, 192, pp. 29 – 37.
- Scott, D. W. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization* (1st ed.). John Wiley & Sons.
- Seeger, M. (2003). *Bayesian Gaussian Process Models: PAC-Bayesian Generalisation Error Bounds and Sparse Approximations* (PhD Thesis). University of Edinburgh.
- Seeger, M. (2007). *A Note on Log-Concavity*. Max Planck Institute for Biological Cybernetics Tübingen. Retrieved from <http://infoscience.epfl.ch/record/175484/files/logconcave.pdf>

- Seeger, M., Williams, C. K. I., and Lawrence, N. D. (2003). Fast Forward Selection to Speed up Sparse Gaussian Process Regression. In C. M. Bishop, and B. J. Frey (Eds.), *Proceedings of the 9th International Workshop on Artificial Intelligence and Statistics* (pp. 205 – 212).
- Seo, S., Wallat, M., Graepel, T., and Obermayer, K. (2000). Gaussian Process Regression: Active Data Selection and Test Point Rejection. *Proceedings of the International Joint Conference on Neural Networks* (Vol. 3, pp. 241 – 246).
- Settles, B. (2010). *Active Learning Literature Survey* (Technical Report No. 1648). University of Wisconsin – Madison.
- Shannon, C. E. (1949). Communication in the Presence of Noise. *Proceedings of the IRE*, 37(1), pp. 10 – 21.
- Smola, A. J., and Bartlett, P. L. (2001). Sparse Greedy Gaussian Process Regression. In T. K. Leen, T. G. Dietterich, and V. Tresp (Eds.), *Advances in Neural Information Processing Systems* (Vol. 13, pp. 619 – 625).
- Smola, A. J., and Schölkopf, B. (1998). *A Tutorial on Support Vector Regression* (Technical Report No. NC-TR-98-030). University of London.
- Snelson, E. L. (2007). *Flexible and Efficient Gaussian Process Models for Machine Learning* (PhD Thesis). University of London.
- Snelson, E. L., and Ghahramani, Z. (2006a). Sparse Gaussian Processes using Pseudo-inputs. In Y. Weiss, B. Schölkopf, and J. C. Platt (Eds.), *Advances in Neural Information Processing Systems* (Vol. 18, pp. 1257 – 1264).
- Snelson, E. L., and Ghahramani, Z. (2006b). Variable Noise and Dimensionality Reduction for Sparse Gaussian Processes. In R. Dechter, and T. Richardson (Eds.), *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence* (pp. 461 – 468).
- Sobol, I. M. (1976). Uniformly Distributed Sequences with an Additional Uniform Property. *USSR Computational Mathematics and Mathematical Physics*, 16(5), pp. 236 – 242.

- Spong, M. W., Hutchinson, S., and Vidyasagar, M. (2006). *Robot Modeling and Control* (1st ed.). John Wiley & Sons.
- Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. (2012). Information-Theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting. *IEEE Transactions on Information Theory*, 58(5), pp. 3250 – 3265.
- Stein, M. L. (1999). *Interpolation of Spatial Data: Some Theory for Kriging*. Springer Verlag.
- Sui, Y., Gotovos, A., Burdick, J. W., and Krause, A. (2015). Safe Exploration for Optimization with Gaussian Processes. In F. Bach, and D. Blei (Eds.), *Proceedings of the 32nd International Conference on Machine Learning*, JMLR: W&CP (Vol. 37, pp. 997 – 1005).
- Sung, H. G. (2004). *Gaussian Mixture Regression and Classification* (PhD Thesis). William Marsh Rice University.
- Thrun, S. (1995). Exploration in Active Learning. *The Handbook of Brain Theory and Neural Networks*, 1, pp. 381 – 384.
- Tipping, M. E. (2001). Sparse Bayesian Learning and the Relevance Vector Machine. *Journal of Machine Learning Research*, 1, pp. 211 – 244.
- Titsias, M. K. (2009). Variational Learning of Inducing Variables in Sparse Gaussian Processes. In D. Van Dyk, and M. Welling (Eds.), *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, JMLR: W&CP (Vol. 5, pp. 567 – 574).
- Toussaint, M. (2009). *Pros and Cons of Truncated Gaussian EP in the Context of Approximate Inference Control*. Technical University of Berlin. Proceedings of the NIPS Workshop on Probabilistic Approaches for Robotics and Control. Retrieved from <https://ipvs.informatik.uni-stuttgart.de/mlr/marc/publications/09-toussaint-NIPSws.pdf>
- Toussaint, M. (2011). *Lecture Notes: Gaussian Identities*. University of Stuttgart. Retrieved from <https://ipvs.informatik.uni-stuttgart.de/mlr/marc/notes/>

gaussians.pdf

- Tresp, V. (2000a). A Bayesian Committee Machine. *Neural Computation*, 12(11), pp. 2719 – 2741.
- Tresp, V. (2000b). The Generalized Bayesian Committee Machine. In R. Bayardo (Ed.), *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 130 – 139).
- Uhlenbeck, G. E., and Ornstein, L. S. (1930). On the Theory of Brownian Motion. *Physical Review*, 36, pp. 823 – 841.
- Valiant, L. G. (1984). A Theory of the Learnable. *Communications of the ACM*, 27(11), pp. 1134 – 1142.
- Williams, C. K. I. (1998). Computation with Infinite Neural Networks. *Neural Computation*, 10(5), pp. 1203 – 1216.
- Williams, C. K. I., and Barber, D. (1998). Bayesian Classification with Gaussian Processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12), pp. 1342 – 1351.
- Williams, C. K. I., and Seeger, M. (2001). Using the Nyström Method to Speed up Kernel Machines. In T. K. Leen, T. G. Dietterich, and V. Tresp (Eds.), *Advances in Neural Information Processing Systems* (Vol. 13, pp. 682 – 688).
- Wu, C. F. J. (1983). On the Convergence Properties of the EM Algorithm. *The Annals of Statistics*, 11(1), pp. 95 – 103.
- Zuluaga, M., Krause, A., Sergent, G., and Püschel, M. (2013). Active Learning for Multi-Objective Optimization. In S. Dasgupta, and D. McAllester (Eds.), *Proceedings of the 30th International Conference on Machine Learning*, JMLR: W&CP (Vol. 28, pp. 462 – 470).