KERNELIZED LEARNING METHODS IN AUTOMATIC CONTROL

THIS IS A TEMPORARY TITLE PAGE
It will be replaced for the final print by a version
provided by the registrar's office.

Thèse n. 1234 2022
présentée le 14 décembre 2022
à la Faculté des sciences de base
laboratoire SuperScience
programme doctoral en SuperScience
École polytechnique fédérale de Lausanne

pour l'obtention du grade de Docteur ès Sciences
par

Paolino Paperino

acceptée sur proposition du jury :

Prof Name Surname, président du jury
Prof Name Surname, directeur de thèse
Prof Name Surname, rapporteur
Prof Name Surname, rapporteur
Prof Name Surname, rapporteur

Lausanne, EPFL, 2022

Besides this'll be easy with the two of us.
We've got science on our side.
— Bonnibel Bubblegum

Acknowledgment

Pausing for a moment to recognize others' contributions to your personal development makes you realize how life unfolds in intertwined, intricate ways.
Writing this thesis and defending my PhD would not have been possible if it were not for professor Colin Jones. I am sincerely grateful for your having invited me to join LA back in 2018, for your having given me the opportunity to develop my work in such a bright research environment. Your always-positive and light-weight approach to work was much appreciated, along with your talent to constantly brig forth new ideas. Thank you for all, truly.
Here's to all other professors and the defense committee...
LA is a great place to do your PhD at. The group is diverse and amicable, people are intellectually bright, and the overall atmosphere is on point. I am indebted to Mr. Harsh Ambarishkumar Shukla for being an awesome friend, for all the long talks about every possible subject one could imagine, most often accompanied by one drink or two. Throughout this journey, Yingzhao Lian has been my PhD twin, always making provocative research remarks and sharing his wine and food expertise with remarkable excitement. Paul Scharnhorst's contributions were mainly in two forms: being a key collaborator with whom the main theoretical results covered herein were derived, and constituting the Bienne-Nêuchatel alliance that organized quite a number of dinners ending in Qwirkle matches. My time spent at the office would have not been as interesting without Mustafa Turan, for listening to my never-ending semi-philosophical blabbers and for being always ready to talk about the latest and greatest recipe. Cite Pulkit, Alessio, Sohail, Philippe, Clara, J1, J2,
The foremost group is certainly my fiancee and my family. I have an immense respect for my parents who have risen their children ensuring the absorption of important core values, including mutual respect, fairness and empathy. To this day, I look up to you. *Muito obrigado por tudo*, Bruno, Erica, Fabio and Renato.
Since we met,
I am grateful for having met truly special people during my Bachelor studies who encouraged me to go beyond what seemed to be our limits at the time. In particular I wish to thank prof. Ruben Barros Godoy who played a fundamental role in my early academic years.

Acknowledgements

Bienne, December 14, 2022 Emilio Tanowe Maddalena

Abstract

This thesis is situated at the crossroads between machine learning and control engineering. Our contributions are both theoretical, through proposing a new uncertainty quantification methodology in a kernelized context; and experimental, through investigating the suitability of certain machine learning techniques to integrate feedback loops in two challenging real-world control problems.

The first part of this document is dedicated to deterministic kernel methods. First, the formalism is presented along with some widespread techniques to craft surrogates for an unknown ground-truth based on samples. Next, standing assumptions are made on the ground-truth complexity and on the data noise, allowing for a novel robust uncertainty quantification (UQ) theory to be developed. By means of this UQ framework, hard out-of-sample bounds on the ground-truth values can be computed through solving convex optimization problems. Closed-form outer approximations are also presented as a lightweight alternative to solving the mathematical programs. Several examples are given to illustrate how the control community could benefit from using this tool.

In the second part of the thesis, statistical models in the form of Gaussian processes (GPs) are considered. These are used to carry out a building temperature control task of a hospital surgery center during regular use. The engineering aspects of the problem are detailed, followed by data acquisition, the model training procedure, and the developed predictive control formulation. Experimental results over a four-day uninterrupted period are presented and discussed, showing a gain in economical performance while ensuring proper temperature regulation.

Lastly, a specialized neural network architecture is proposed to learn linear model predictive controllers (MPC) from state-input pairs. The network features parametric quadratic programs (pQP) as an implicit non-linearity and is used to reduce the storage footprint and online computational load of MPC. Two examples in the domain of power electronics are given to showcase the effectiveness of the proposed scheme. The second of them consisted in enhancing the start-up response of a real step-down converter, deploying the learned control law on an 80 MHz microcontroller and performing the computations in under 30 microseconds.

Résumé

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Contents

1 Introduction

A non-numbered chapter...

1.1 Outline and Contribution

1.2 Publications

The subsequent chapters of this dissertation were based on the following publications:

- E. T. Maddalena, Y. Lian, and C. N. Jones. “Data-driven methods for building control—A review and promising future directions." Control Engineering Practice 95 (2020): 104211.
- E.T. Maddalena, P. Scharnhorst, and C. N. Jones. “Deterministic error bounds for kernel-based learning techniques under bounded noise." Automatica 134 (2021): 109896.
- P. Scharnhorst, E.T. Maddalena, Y. Jiang, and C. N. Jones. “Robust Uncertainty Bounds in Reproducing Kernel Hilbert Spaces: A Convex Optimization Approach." arXiv.
- E. T. Maddalena, P. Scharnhorst, Y. Jiang, and C. N. Jones. “KPC: Learning-based model predictive control with deterministic guarantees." Learning for Dynamics and Control. PMLR, 2021.
- E. T. Maddalena, S. A. Müller, R. M. dos Santos, C. Salzmann, C. N. Jones. “Experimental Data-Driven Model Predictive Control of a Hospital HVAC System During Regular Use." Energy and Buildings: 112316 (2022).

Works developed during the course of this PhD that are related to the thesis, but not discussed herein include:

- E. T. Maddalena, M. W. F. Specq, V. L. Wisniewski, and C. N. Jones. “Embedded PWM predictive control of DC-DC power converters via piecewise-affine neural networks." IEEE Open Journal of the Industrial Electronics Society (2021): 199-206.
- E.T. Maddalena, and C. N. Jones. “NSM converges to a k-NN regressor under loose Lipschitz estimates." IEEE Control Systems Letters 134 (2020): 880-885.
- E. T. Maddalena, C. G. S. Moraes, G. Waltrich, and C. N. Jones. “A neural network architecture to learn explicit MPC controllers from data." IFAC-PapersOnLine (2020): 11362-11367.
- A. Chakrabarty, E. T. Maddalena, H. Qiao, and C. Laughman. “Scalable Bayesian optimization for model calibration: Case study on coupled building and HVAC dynamics." Energy and Buildings 253, 111460
- E. T. Maddalena, and C. N. Jones. “Learning non-parametric models with guarantees: A smooth Lipschitz regression approach." IFAC-PapersOnLine (2020): 965-970.
- U. Rosolia, Y. Lian, E. T. Maddalena, G. Ferrari-Trecate, and C. N. Jones “On the Optimality and Convergence Properties of the Iterative Learning Model Predictive Controller." IEEE Transactions on Automatic Control.
- L. di Natale, Y. Lian, E. T. Maddalena, J. Shi, and C. N. Jones “Lessons Learned from Data-Driven Building Control Experiments: Contrasting Gaussian Process-based MPC, Bilevel DeePC, and Deep Reinforcement Learning." arXiv.

2 Safely learning with kernels

At its core, learning refers to the process of *gathering information* and using it to *improve one's knowledge* about the subject or phenomenon under study. The standing assumption here is then clearly that a link is in place, tying information and phenomenon together even if such link is partially corrupted. Information typically comes in the form of data, samples, sometimes referred to as examples, and the mathematical formalism often used in modern machine learning to study the link between examples and the underlying phenomena is statistics. This choice is convenient because it can describe the possible non-determinism of outcomes through the concepts of distributions and samples; and because it provides us with plenty of tools to carry out learning, i.e., improve our knowledge about the phenomenon through the samples at hand. In this chapter, we will however adopt a different standpoint to study and tackle the problem of learning, which is, as we will later argue, more aligned with the ways control engineers are taught to see physical systems. This standpoint is the one offered by approximation theory.

In this chapter, we will introduce the problem of learning from data and elucidate what approach will be taken to tackle it. Next, novel uncertainty quantification results will be presented concerning the point-evaluations of an unknown ground-truth. Finally, some examples are given to illustrate the general use of the theory.

Statistical learning and approximation theory are not in opposition. Indeed, we can define the function of interest as the conditional Temlyakov (2008), perhaps talk about Belkin's work linking the two and advocating for using the approximation lenses.

2.1 The formalism of kernels

Our goal in to learn maps of the form $f : \mathcal{X} \to \mathbb{R}$ and, to achieve that end, we will make extensive use of auxiliary functions called kernels.

Definition 1. (Kernel) Given an arbitrary non-empty set $\mathcal{X}$, a kernel k is any symmetric function of the form

$$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R} \tag{2.1}$$

Definition 2. (Kernel matrix) Let $X = \{x_1, \ldots, x_n\} \subset \mathcal{X}$ be a finite set of points. The $n \times n$ matrix K_{XX} with entries $[K_{XX}]_{ij} = k(x_i, x_j)$ is called the kernel matrix of k associated with X.

Definition 3. (Positive-definite kernel) A kernel function k is said to be positive-definite if for any finite subset of points $X \subset \mathcal{X}$, the kernel matrix satisfies $K_{XX} \succeq 0$. If, in particular, $K_{XX} \succ 0$, then the kernel is strictly positive-definite.

Remark 1. Aside from the last definition, there exist broader classes of kernel functions such as the *conditionally positive-definite* one (Schölkopf and Smola, 2002, §2.4)(Wendland, 2004, §8). In addition, one can also generalize the co-domain k to be the field of complex numbers $\mathbb{C}$.

Instances of positive-definite kernel functions with index set $\mathcal{X} = \mathbb{R}^n$ are the linear, the squared-exponential (also known as Gaussian), the exponential (equivalent to the Matern12), the polynominal and the cosine kernels respectively given by

$$k_{\text{lin}}(x, x') = x \cdot x' \tag{2.2}$$

$$k_{\text{se}}(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\ell}\right) \tag{2.3}$$

$$k_{\text{exp}}(x, x') = \exp\left(-\frac{\|x - x'\|}{2\ell}\right) \tag{2.4}$$

$$k_{\text{pol}}(x, x') = \Big(\sigma^2(x \cdot x') + \gamma\Big)^d \tag{2.5}$$

$$k_{\text{cos}}(x, x') = \cos\left(2\pi \sum_i ([x]_i - [x']_i)/\ell\right) \tag{2.6}$$

where $\cdot$ and $\|\cdot\|$ denote the usual inner-product and 2-norm in $\mathbb{R}^n$, respectively; and the constants $\ell \in \mathbb{R}_{>0}$, $\sigma, \gamma \in \mathbb{R}$ are the so-called *hyperparameters*. Plots of these functions are presented in Figure 2.1. A more complete list of PD kernels and their mathematical properties can be found in Appendix A.

In order to construct surrogate models for the unknown f, one could partially evaluate a given k to match their domains and co-domains. In other words, fix

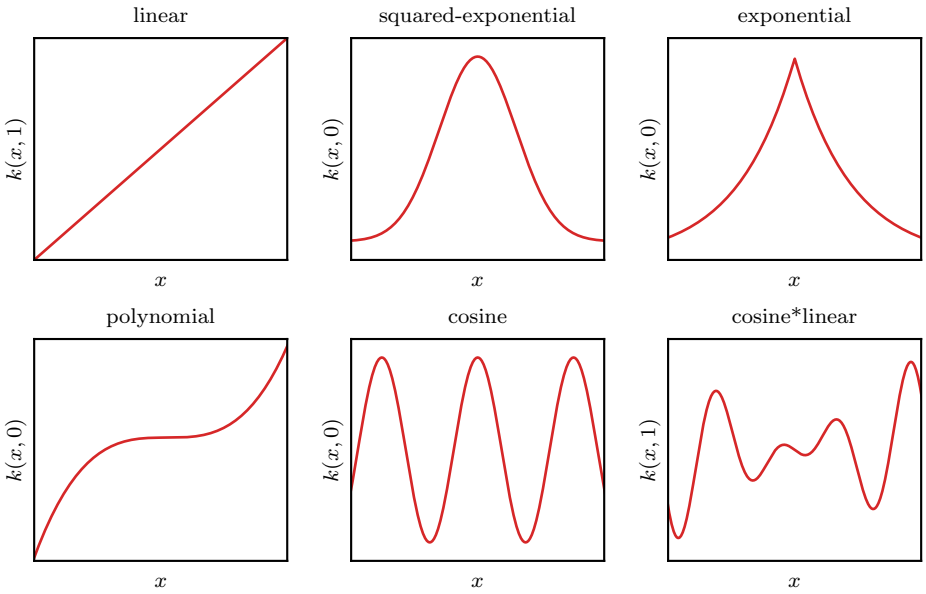


Figure 2.1: Examples of positive-definite kernels.

one of its arguments so that $k(z,\cdot) : \mathcal{X} \to \mathbb{R}, x \mapsto k(z,x)$ for some $z \in \mathcal{X}$. Indeed, this was the approach taken to draw the plots in Figure 2.1. Approximating the unknown f with a single partially evaluated kernel however appears to be overly restrictive. A sensible next step would be to consider linear combinations of such kernel functions. It turns out that every PD kernel has a function space associated with it that contains these linear combinations and is endowed with plenty of useful geometric structure. The following concepts are presented next to set up the stage for defining this special hypothesis space, the *reproducing kernel Hilbert space*.

Feature maps Φ are central in machine learning, allowing one to represent the data x he has in a more suitable format. At the same time, (2.7) unveils another aspect of a kernel evaluation $k(x,x')$: it returns the inner-product value of the transformed inputs $\Phi(x)$, $\Phi(x')$. Educational textbooks often interpret these inner-products as a similarity measure between x and x' (Schölkopf and Smola, 2002).

Proposition 1. (PD kernels have feature maps) Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a positive-definite kernel. Then there exists a Hilbert space $\mathbb{H}$ endowed with an inner product $\langle\cdot,\cdot\rangle_{\mathbb{H}}$ and a mapping $\Phi : \mathcal{X} \to \mathbb{H}$ such that

$$k(x,x') = \langle \Phi(x), \Phi(x') \rangle_{\mathbb{H}} \tag{2.7}$$

holds for any $x, x' \in \mathcal{X}$.

Proof. (Steinwart and Christmann, 2008, Theorem 4.16), modulo the nomencla-

ture difference. □

The mappings Φ above as well as the $\mathbb{H}$ spaces are in general not unique (Steinwart and Christmann, 2008, §4), but there is one such space that enjoys an extra property that rules out some unexpected behavior from its members. This particular $\mathbb{H}$ is moreover not an arbitrary Hilbert space, but a Hilbert space of functions.

Definition 4. (Reproducing kernel Hilbert space) Let $\mathcal{X} \neq \emptyset$ and $\mathbb{R}^{\mathcal{X}}$ the set of functions mapping $\mathcal{X}$ to $\mathbb{R}$. The subset $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ is called a reproducing kernel Hilbert space (RKHS) if it is a Hilbert space and if $\forall x \in \mathcal{X}$ the evaluation functionals

$$L_x : \mathcal{H} \to \mathbb{R},\ L_x(f) \mapsto f(x),\ \forall f \in \mathcal{H} \tag{2.8}$$

are bounded.

In order to see how useful such a property is, consider a sequence $\{f_n\}_{n=1}^{\infty}$ within a certain Hilbert function space $\mathbb{H} \subset \mathbb{R}^{\mathcal{X}}$. Intuitively, one would expect that if $f_n \to f^\star$ in $\mathbb{H}$, then the values $f_n(x)$ attained by the sequence would converge to the values $f^\star(x)$. Yet, this is not always the case (see Example 1 in Appendix A). If, on the other hand, the evaluation functionals are bounded as in Definition 4, then the connection between convergence in the function space and the pointwise convergence of functions is guaranteed. Indeed, if $\{f_n\}_{n=1}^{\infty}$ and $f^\star$ are members of an RKHS $\mathcal{H}$, then $|f_n(x) - f^\star(x)| = |L_x(f_n) - L_x(f^\star)| \leq \|L_x\|\, \|f_n - f^\star\|_{\mathcal{H}}$, where $\|L_x\|$ is the operator norm of L_x that is guaranteed by Definiton 4 to be a finite number. As a result, if $\|f_n - f^\star\|_{\mathcal{H}} \to 0$, the right-hand side of the inequality goes to zero, and so does the pointwise difference $|f_n(x) - f^\star(x)|$. Therefore, function convergence in an RKHS implies pointwise convergence, matching our intuition.

Proposition 2. (Every RKHS has a unique PD reproducing kernel) Let $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ be an RKHS. Then, the map $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, $k(x, x') := \langle L_x, L_{x'} \rangle_{\mathcal{H}}$ is a positive-definite kernel. Furthermore, k is the unique map to satisfy the reproducing property, i.e., for any $x \in \mathcal{X}$, $k(x, \cdot) \in \mathcal{H}$ and

$$\langle f, k(x, \cdot) \rangle_{\mathcal{H}} = L_x(f) = f(x),\ \forall f \in \mathcal{H} \tag{2.9}$$

Proof. (Berlinet and Thomas-Agnan, 2011, Lemma 2) along with (Steinwart and Christmann, 2008, Theorem 4.20). □

Proposition 3. (Every PD kernel has a unique RKHS) Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a PD kernel. If k is the reproducing kernel of an RKHS $\mathcal{H}_A$ and of another RKHS $\mathcal{H}_B$, then $\mathcal{H}_A = \mathcal{H}_B$.

Proof. (Steinwart and Christmann, 2008, Theorem 4.21). □

We understand from Propositions 2 and 3 that a special relationship exists between kernel and its RKHS. Nevertheless, it is still unclear from Definition 4 alone how a given k influences or defines the members of $\mathcal{H}$. To shed light on the matter, it helps to explicitly construct $\mathcal{H}$ starting from a given k. Consider the so-called *pre-Hilbert space*

$$\mathcal{H}_0 := \operatorname{span}\{k(x,\cdot) \mid x \in \mathcal{X}\} \tag{2.10}$$

$$= \left\{ \sum_{i=1}^{n} c_i\, k(x_i,\cdot) \mid n \in \mathbb{N}, c_i \in \mathbb{R}, x_i \in \mathcal{X} \right\} \tag{2.11}$$

equipped with the real-valued map $\langle f, g\rangle_{\mathcal{H}_0} := \sum_{i=1}^{n}\sum_{j=1}^{m} a_i b_i k(x_i, x_j)$ for members $f, g \in \mathcal{H}_0, f = \sum_{i=1}^{n} a_i\, k(x_i,\cdot), g = \sum_{j=1}^{m} b_j\, k(x_j,\cdot)$, which can be shown to be a valid inner-product. This family $\mathcal{H}_0$ of functions is however not guaranteed to be complete, i.e., sequences $\{f_i\}_{i\in\mathbb{N}}$ of members might converge to functions outside $\mathcal{H}_0$. To transform it into a proper Hilbert space, one closes the space

$$\mathcal{H} := \operatorname{clos} \mathcal{H}_0 \tag{2.12}$$

thus encompassing all limit points[1]. Finally, the function space $\mathcal{H}$ defined in (2.12) can then be shown to be a valid RKHS[2] according to Definition 4. In fact, it is the *only* one associated with k. We therefore understand that the members of $\mathcal{H}$ are weighted sums of partially evaluated kernels as per (2.11) along with their limit points.

The questions of how expressive RKHSs can be still lingers on. To better examine the matter, consider the following measure of expressiveness.

Definition 5. (Universal kernel) Let k be a continuous PD kernel and the set $\mathcal{X}$ be a compact metric space. Then k is called universal if its RKHS $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ is dense in the space of real-valued continuous functions $C(\mathcal{X}) \subset \mathbb{R}^{\mathcal{X}}$ with respect to the maximum norm $\|\cdot\|_\infty$.

The above definition guarantees that for any target function $g \in C(\mathcal{X})$ and any tolerable error $\epsilon > 0$, there exists an f in the RKHS of a universal kernel such that their mismatch is bounded $|f(x) - g(x)| \leq \epsilon, \forall x \in \mathcal{X}$. All in all, the universality property is an indication of how rich a hypothesis space is, thus reassuring the

[1] Closing $\mathcal{H}_0$ requires defining an inner-product on the superset $\mathcal{H}$ that is consistent with the one present in the subset $\mathcal{H}_0$. Also, the closure of a set is always closed, which guarantees that sequences within $\mathcal{H}$ cannot converge to functions outside of it.

[2] For a proof of this statement, the reader is referred to (Berlinet and Thomas-Agnan, 2011, §3) or to (Sejdinovic and Gretton, 2012, §4) for a more step-by-step pedagogical exposition.

user that little bias error will be introduced by his choice of model class.

Proposition 4. Let $\mathcal{X}$ be a compact subset of $\mathbb{R}^n$. The squared-exponential kernel with (2.3) with $\ell > 0$ is universal.

Proof. (Steinwart and Christmann, 2008, Corollary 4.58). □

In contrast with Proposition 4, some results on the restrictiveness of RKHSs can also be found in the literature. In Steinwart (2020) for example, the author shows that no RKHS can contain $C(X)$. On a looser note, some authors argue that members of the squared-exponential kernel (**??**) has an $\mathcal{H}$ that is too smooth when compared to alternative hypothesis spaces that are also associated with the same kernel (see the discussion in (Kanagawa et al., 2018, §4)). For a thorough exposition of universal kernels, the reader is referred to Micchelli et al. (2006). Lastly, we underline that other model classes in machine learning also enjoy the same universality properties that kernels do, notable certain architectures of deep neural networks (Kidger and Lyons, 2020).

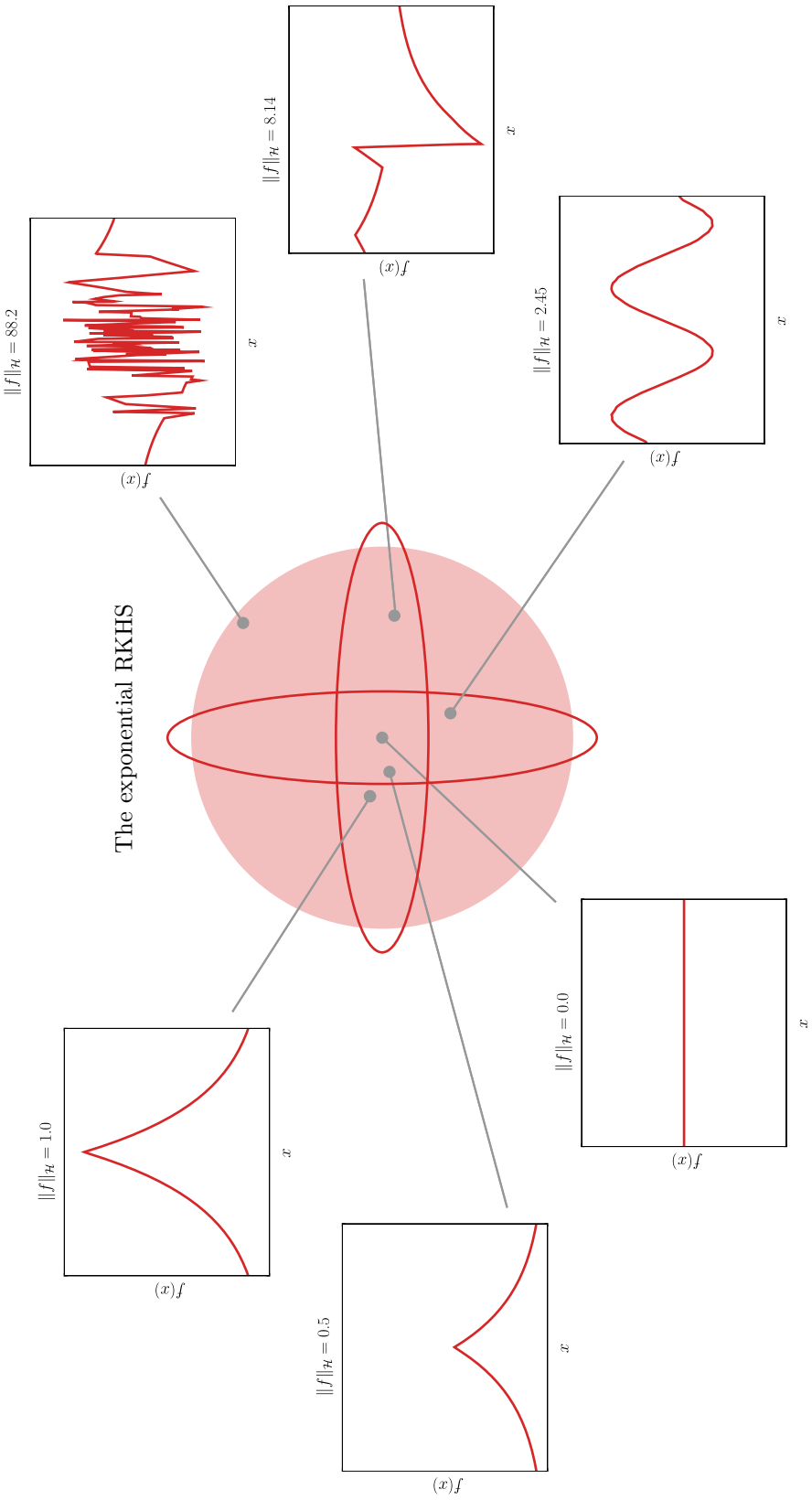


Figure 2.2: Members of the exponential RKHS and their respective norms.

2.2 Crafting models

Suppose a dataset of the form $\{(x_i, y_i)\}_{i=1}^n$ is given. The $x_i \in \mathcal{X}, \forall i$ elements are referred to as *inputs* and the $y_i \in \mathbb{R}, \forall i$ as *outputs*. In this section, we will discuss exclusively the case where $\mathcal{X} \subset \mathbb{R}^m$ is a compact set, and the outputs are real-valued. The values y_i are assumed to deliver information about an underlying ground-truth function $f^\star$ through the measurement model

$$y_i = f^\star(x_i) + \epsilon_i, \quad i = 1, \dots, n \tag{2.13}$$

As detailed in Section 2.1, weighted sums of partially evaluated kernel functions arise naturally in the context of kernel learning. In this section we shall see that, when given $\{(x_i, y_i)\}_{i=1}^n$, maps of the form

$$f(x) = \sum_{i=1}^n \alpha_i k(x_i, x) \tag{2.14}$$

are good candidates for acting as surrogate functions for the ground-truth $f^\star$. Indeed, they are known to solve a number of optimal fitting problems given appropriate weights α as we explain next.

In the absence of measurement noise, i.e., when $\epsilon_i = 0$, the outputs y_i perfectly represent $f^\star$. As a result, data interpolation can be a sensible task to carry out, which could done over $f \in \mathcal{H}$ while minimizing the resulting model norm.

Proposition 5. (Minimum-norm interpolation (MNI)) Let $\{(x_i, y_i)\}_{i=1}^n$ be a collection of points such that $x_i \in \mathbb{R}^m$ and $y_i \in \mathbb{R}$. Let k be a PD kernel and $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ its RKHS. Then, the variational problem

$$\bar{f} \in \arg\inf_{f \in \mathcal{H}} \left\{ \|f\|_{\mathcal{H}}^2 : f(x_i) = y_i,\ i = 1, \dots, n \right\} \tag{2.15}$$

admits the unique solution $\bar{f} \in \mathcal{H}$, $\bar{f}(x) = \sum_{i=1}^n \alpha_i k(x_i, x)$, $\alpha = y^\top K_{XX}^{-1}$.

Proof. (Kanagawa et al., 2018, Theorem 3.5). □

Remark 2. Some might think that models of the form (2.15) would perform poorly in real-world scenarios as overfitting goes against established machine learning guidelines. Yet, some authors have recently advocated for such models, and interpolants in general, stating that they possess strong generalization capabilities (see e.g. Belkin et al. (2018, 2019); Beaglehole et al. (2022)).

To tackle the approximation problem in the presence of measurement noise, a

compromise between fitting the data and rejecting uninformative fluctuations is sometimes desirable. One of the most standard tools used to achieve this balance is kernel ridge regression (KRR), in which the unconstrained problem

Proposition 6. (Kernel ridge regression (KRR)) Let $\{(x_i, y_i)\}_{i=1}^n$ be a collection of points such that $x_i \in \mathcal{X}$ for a compact $\mathcal{X} \subset \mathbb{R}^m$ and $y_i \in \mathbb{R}$. Let k be a SPD kernel and $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ its RKHS. Then, the variational problem

$$\inf_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{H}}^2 \tag{2.16}$$

with $\lambda > 0$ admits a single minimizer, the function $f(x) = \sum_{i=1}^n \alpha_i x_i$ with $\alpha = (K_{XX} + n\lambda I)^{-1} y$.

Original representer theorem Kimeldorf and Wahba (1971).

Theorem 1. (The representer theorem) Let $\{x_i, y_i\}_{i=1}^n$ be a collection of points such that $x_i \in \mathcal{X}$ for an arbitrary $\mathcal{X}$ and $y_i \in \mathbb{R}$. Let k be a PD kernel and $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ its RKHS. Consider an arbitrary function $c : (\mathcal{X} \times R^2)^n \to \mathbb{R} \cup \{\infty\}$ and a strictly monotonic increasing function $\Omega : [0, \infty) \to \mathbb{R}$. Then, if $f \in \mathcal{H}$ is a minimizer of the variational problem

$$\inf_{f \in \mathcal{H}} c((x_1, y_1, f(x_1)), \ldots, (x_n, y_n, f(x_n))) + \Omega(\|f\|_{\mathcal{H}}) \tag{2.17}$$

admits a representation of the form $f(x) = \sum_{i=1}^n \alpha_i x_i$, with $\alpha_i \in \mathbb{R}$.

The proof is given in Schölkopf et al. (2001). □

Specializing (2.17) to a more usual loss function, we arrive at the well-known kernel ridge regression (KRR) problem, which admits a unique, closed-form solution.

2.3 Quantifying uncertainty

Besides being able to craft surrogate functions for our unknown ground-truth, it is also a common desideratum to understand how far away our predictions can be from the real phenomenon. We will start this section by formalizing the problem of bounding the ground-truth values that can be attained even at unseen locations given the information at hand. It is important to highlight that this process will not require a model. Next, alternative bounds are developed, this time around nominal models such as the ones presented in Section 2.2. Rather than limiting ourselves to the theoretical sphere, the discussion will also touch on the computational aspects involved in evaluating the derived expressions.

2.3.1 The setting and problem definition

The theory developed in this section will revolve around a specific class of kernels and input spaces, and be built on the following standing assumptions.

Assumption 1. k is strictly positive-definite and its index set $\mathcal{X} \subset \mathbb{R}^m$ is compact.

Assumption 2. $f^\star$ is contained in the RKHS $\mathcal{H}$ associated with k.

The available data $\{(x_i, \mathsf{y}_i)\}_{i=1}^d$ is such that $x_i \in \mathcal{X}$ and $\mathsf{y}_i \in \mathbb{R}^{n_i}$, where the vector y_i stacks n_i scalar outputs $y_{i,1}, \dots, y_{i,n_i}$ observed at the same input location x_i. The outputs are assumed to carry information about an underlying unknown ground-truth map $f^\star$ according to

$$y_{i,j} = f^\star(x_i) + \delta_{i,j} \tag{2.18}$$

where $\delta_{i,j}$ denotes an additive measurement noise. If only a single output is present at each input location, the observational model (2.18) simplifies to $y_i = f^\star(x_i) + \delta_i$. For brevity, let X denote the set of all inputs $x_1, \dots, x_d$ in the dataset. As for the nature of $\delta_{i,j}$, no specific distributional assumptions are made, but only that its magnitude is uniformly bounded by a known scalar.

Assumption 3. $\delta_{i,j}$ is bounded by a known scalar $\bar{\delta}$, i.e., $|\delta_{i,j}| \leq \bar{\delta}, \forall i, j$.

At this point one could ask himself if any out-of-sample guarantees could be already established on the values attained by $f^\star$. The answer is no. Indeed, for any tentative upper bound $\omega < f^\star(x)$ at $x \notin \{x_1, \dots, x_d\}$ regardless of the number of samples d, there is a member $f \in \mathcal{H}$ capable of reproducing any values at the x_i locations and additionally violating ω by an arbitrary level. Lower bounds could be equally violated as well. What we lack is a complexity bound, which will be posed by restricting $f^\star$ to lie within the Γ-ball of $\mathcal{H}$.

Assumption 4. An upper-bound $\Gamma \geq \|f^\star\|_{\mathcal{H}}$ is known.

Remark 3. The matter of exactly computing RKHS norms from weights and otherwise estimating them from data is discussed in Appendix 2.6.2.

With all assumptions in place, we can formulate the variational problem $\mathbb{P}0$ below, with the query point $x \in \mathcal{X}$ as a parameter

$$\boxed{\mathrm{F}(x) = \sup_{f \in \mathcal{H}} \{f(x) : \|f\|_{\mathcal{H}} \leq \Gamma, \|f_X - \mathsf{y}\|_\infty \leq \bar{\delta}\}} \tag{2.19}$$

where f_X is the vector of evaluations at the input locations, which are repeated whenever multiple outputs are available at a given input (see Appendix 2.6.3). We highlight that the supremum is guaranteed to be a finite numbers. To see this, notice that $|f(x)| = |\langle f, k(x,\cdot)\rangle_{\mathcal{H}}| \leq \|f\|_{\mathcal{H}} \|k(x,\cdot)\|_{\mathcal{H}} \leq \Gamma \sqrt{k(x,x)}$. This last bound is rather loose as it does not exploit any information present in X nor the outputs y, and is moreover uniform for translation-invariant kernels such as the squared-exponential. (2.19) on the other hand makes use the dataset $\{(x_i, \mathsf{y}_i)\}_{i=1}^d$ in its entirety as well as the complexity bound Γ.

2.3.2 The optimal solution

Consider now the convex parametric quadratically-constrained linear program $\mathbb{P}1$

$$\mathrm{C}(x) = \max_{c\in\mathbb{R}^d, c_x\in\mathbb{R}} \quad c_x \tag{2.20}$$

$$\text{subj. to} \quad \begin{bmatrix} c \\ c_x \end{bmatrix}^\top \begin{bmatrix} K_{XX} & K_{Xx} \\ K_{xX} & k(x,x) \end{bmatrix}^{-1} \begin{bmatrix} c \\ c_x \end{bmatrix} \leq \Gamma^2 \tag{2.21}$$

$$\|\Lambda c - \mathsf{y}\|_\infty \leq \bar{\delta} \tag{2.22}$$

for any $x \in \mathcal{X}\backslash X$, and extend its value function $C(x)$ to points $x = x_i \in X$ with the solution of $\mathbb{P}1' : \mathrm{C}(x) = \max_{c\in\mathbb{R}^d}\{c_i \,|\, c^\top K_{XX}^{-1} c \leq \Gamma^2, \|\Lambda\, c - \mathsf{y}\|_\infty \leq \bar{\delta}\}$. The two cases $\mathbb{P}1$ and $\mathbb{P}1'$ are distinguished due to the matrix in (2.21) becoming singular for any $x \in X$, and since it allows for one decision variable to be eliminated. The connection between this optimization problem and (2.19) is unveiled next.

Theorem 2. (Finite-dimensional equivalence): The objective in $\mathbb{P}0$ attains its supremum in $\mathcal{H}$ and $\mathrm{F}(x) = \mathrm{C}(x)$ for any $x \in \mathcal{X}$.

The proof revolves around showing that a solution to $\mathbb{P}0$ necessarily lies in a finite-dimensional subspace of $\mathcal{H}$, in a "representer theorem" spirit (Schölkopf et al., 2001). The attainment of the supremum is shown from topological aspects of the constraints in this subspace; and, finally, the match $\mathrm{F}(x) = \mathrm{C}(x)$ by re-evaluating the constraints in light of the solutions to $\mathbb{P}0$ being finitely representable.

Proof. Let $\mathbb{X} := X \cup \{x\}$ and define the finite-dimensional subspace $\mathcal{H}^\parallel = \{f \in \mathcal{H} : f \in \mathrm{span}(k(x_i,\cdot), x_i \in \mathbb{X})\}$. Furthermore, let $\mathcal{H}^\perp = \{g \in \mathcal{H} : \langle g, f^\parallel\rangle_{\mathcal{H}} = 0, \forall f^\parallel \in \mathcal{H}^\parallel\}$ be the orthogonal complement of $\mathcal{H}^\parallel$. Then, we have $\mathcal{H} = \mathcal{H}^\parallel \oplus \mathcal{H}^\perp$ and for all $f \in \mathcal{H}$, $\exists f^\parallel \in \mathcal{H}^\parallel, f^\perp \in \mathcal{H}^\perp : f = f^\parallel + f^\perp$. By employing the latter decomposition and using the reproducing property, we can reformulate $\mathbb{P}0$ in terms of $\mathcal{H}^\parallel$ and $\mathcal{H}^\perp$ as

$$\sup_{\substack{f^{\|}\in\mathcal{H}^{\|}\\ f^{\perp}\in\mathcal{H}^{\perp}}} \left\{\langle f^{\|}+f^{\perp},k(x,\cdot)\rangle_{\mathcal{H}} : \left\|f^{\|}+f^{\perp}\right\|_{\mathcal{H}}^2 \leq \Gamma^2, \left\|(f^{\|}+f^{\perp})_X-\mathsf{y}\right\|_{\infty}\leq\bar{\delta}\right\} \tag{2.23}$$

$$\overset{(i)}{=} \sup_{\substack{f^{\|}\in\mathcal{H}^{\|}\\ f^{\perp}\in\mathcal{H}^{\perp}}} \left\{f^{\|}(x) : \left\|f^{\|}\right\|_{\mathcal{H}}^2 + \left\|f^{\perp}\right\|_{\mathcal{H}}^2 \leq \Gamma^2, \left\|f^{\|}_X-\mathsf{y}\right\|_{\infty}\leq\bar{\delta}\right\} \tag{2.24}$$

$$\overset{(ii)}{=} \sup_{f^{\|}\in\mathcal{H}^{\|}} \left\{f^{\|}(x) : \left\|f^{\|}\right\|_{\mathcal{H}}^2 \leq \Gamma^2, \left\|f^{\|}_X-\mathsf{y}\right\|_{\infty}\leq\bar{\delta}\right\} \tag{2.25}$$

In (i), the $f^{\perp}$ component vanished from the cost and from the last constraint due to orthogonality w.r.t. $k(x_i,\cdot)\in\mathcal{H}^{\|}$ for any $x_i\in\mathbb{X}$; moreover, the Pythagorean relation $\|f\|_{\mathcal{H}}^2 = \left\|f^{\|}\right\|_{\mathcal{H}}^2 + \left\|f^{\perp}\right\|_{\mathcal{H}}^2$ was also used. To arrive at the second equality (ii), one only has to note that the objective is insensitive to $f^{\perp}$ and that any $f^{\perp}\neq 0_{\mathcal{H}}$ would tighten the first constraint.

The attainment of the supremum is addressed next. Consider (2.25) and denote the members of $\mathcal{H}^{\|}$ simply as f. $\|f\|_{\mathcal{H}}^2\leq\Gamma^2$ is a closed and bounded constraint as it is the sublevel set of a norm. We transform $\|f_X-\mathsf{y}\|_{\infty}\leq\bar{\delta}$ into $|f(x_i)-y_{i,j}|\leq\bar{\delta}$, $i=1,\ldots,d$, $j=1,\ldots,n_i$. Sets of the form $\{a\in\mathbb{R}:|a|\leq b\}$ are clearly closed in $\mathbb{R}$, hence $\{f(x_i)\in\mathbb{R}:|f(x_i)-y_{i,j}|\leq\bar{\delta},\forall i,j\}$ is also closed. For any x_i, the evaluation functional $L_{x_i}(f)=f(x_i)$ is a linear operator and thus pre-images of closed sets are also closed. Consequently, $\{f\in\mathcal{H}^{\|}:|f(x_i)-y_{i,j}|\leq\bar{\delta},\forall i,j\}$ is closed in $\mathcal{H}^{\|}$. The intersection of a finite number of closed sets is necessarily closed, thus all constraint present in (2.25) define a closed feasible set. Since $\mathcal{H}^{\|}$ is finite-dimensional, any closed and bounded subset of it is compact (Heine–Borel); therefore, the continuous objective $L_x(f)=f(x)$ in (2.25) attains a maximum by the Weierstrass extreme value theorem.

Finally, we establish the connection between $\mathbb{P}0$ and $\mathbb{P}1$. From the above arguments, an optimizer for $\mathbb{P}0$ must lie in $\mathcal{H}^{\|}$. The members $f\in\mathcal{H}^{\|}$ have the form $f(z)=\alpha^{\top}K_{\mathbb{X}z}$, being defined by the α weights. Due to the positive-definiteness of k, there exists a bijective map between outputs at the $\mathbb{X}$ locations $f_{\mathbb{X}}=\begin{bmatrix}f(x_1) & \ldots & f(x_d) & f(x)\end{bmatrix}^{\top}$ and the weights α, namely $\alpha=K_{\mathbb{X}\mathbb{X}}^{-1}f_{\mathbb{X}}$. $K_{\mathbb{X}\mathbb{X}}$ denotes the kernel matrix associated with $\mathbb{X}$. Consequently, optimizing over $f\in\mathcal{H}^{\|}$ is equivalent to optimizing over $\begin{bmatrix}f(x_1) & \ldots & f(x_d) & f(x)\end{bmatrix}^{\top} =: \begin{bmatrix}c^{\top} & c_x\end{bmatrix}^{\top}$. The bounded norm condition can be recast as $\|f\|_{\mathcal{H}}^2=\langle f,f\rangle_{\mathcal{H}}=\alpha^{\top}K_{\mathbb{X}\mathbb{X}}\alpha = \begin{bmatrix}c^{\top} & c_x\end{bmatrix}K_{\mathbb{X}\mathbb{X}}^{-1}\begin{bmatrix}c^{\top} & c_x\end{bmatrix}^{\top}$. The last constraint and the objective are straightforward, and this concludes the proof. □

Remark 4. (The optimal lower bound): In a way analogous to (2.19), the problem $\inf_{f\in\mathcal{H}}\{f(x) : \|f\|_{\mathcal{H}} \leq \Gamma, \|f_X - \mathsf{y}\|_\infty \leq \bar{\delta}\}$ could be posed to compute the minimum out-of-sample value that could be attained by the unknown ground-truth. Its finite-dimensional counter-part would be $\mathrm{B}(x) = \min_{c\in\mathbb{R}^d, c_x\in\mathbb{R}}\{c_x : (2.21), (2.22)\}$ for any $x \in \mathcal{X}\backslash X$, and extend it to points $x = x_i \in X$ with $\mathrm{B}(x) = \min_{c\in\mathbb{R}^d}\{c_i \,|\, c^\top K_{XX}^{-1} c \leq \Gamma^2, \|\Lambda\, c - \mathsf{y}\|_\infty \leq \bar{\delta}\}$. As a result, computing the whole "uncertainty envelope" requires solving two problems per query point.

An illustrative example is shown in Figure 2.3, where noisy samples were gathered from an unknown ground-truth (dashed line). The upper and lower bounds $\mathrm{C}(x)$, $\mathrm{B}(x)$ were then computed based on an augmented norm estimate, and are shown in red. Finally, the ground-truth is shown to lie within the uncertainty envelope. We highlight that this procedure does not require defining any nominal model.

Theorem 2 states that quantifying uncertainty in our particular kernelized setting can be done through convex programming involving $d + 1$ decision variables. According to (2.21), $c \in \mathbb{R}^d$ is constrained to be consistent with the already seen outputs y up to the tolerance $\bar{\delta}$; whereas according to (2.22), the ensemble c and c_x must lead to a total complexity not greater than Γ. It turns out that, for any query point x outside the data-set, the complexity bound is always activated since the objective is only sensitive to c_x, which is not constrained by the infinity norm. This is formalized next. Replacing the inequality in (2.21) by an equality would lead to the loss of convexity as is therefore not desirable.

Proposition 7. The inequality constraint (2.21) is always active, i.e., for any $x \in \mathcal{X}\backslash X$ let $(c^\star, c_x^\star)$ be an optimizer of $\mathbb{P}1$, then $\begin{bmatrix} c^\star \\ c_x^\star \end{bmatrix}^\top \begin{bmatrix} K_{XX} & K_{Xx} \\ K_{xX} & k(x,x) \end{bmatrix}^{-1} \begin{bmatrix} c^\star \\ c_x^\star \end{bmatrix} = \Gamma^2$.

Given our knowledge on the noise influence $\bar{\delta}$, it is only natural to ask what the

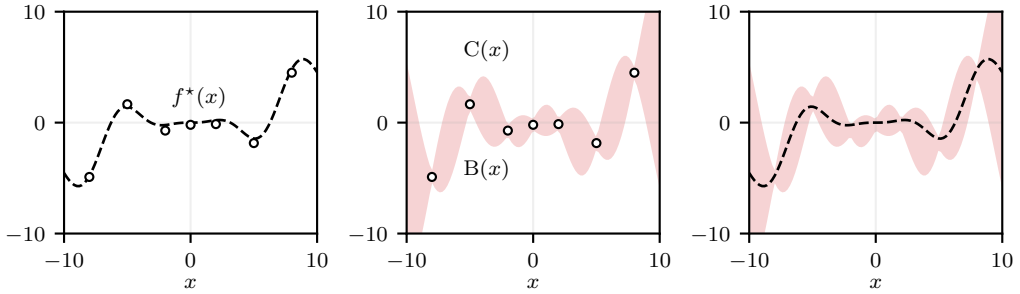


Figure 2.3: Optimal bounds example for the SE kernel (2.3) with $\ell = 2.5$. Left: a member $f \in \mathcal{H}$ with $\|f\|_{\mathcal{H}} = 16.42$ and 7 data-points with $\bar{\delta} = 0.5$. Center: data-points and the optimal bounds $\mathrm{C}(x)$, $\mathrm{B}(x)$ computed with $\Gamma = 1.1\,\|f\|_{\mathcal{H}}$. Right: the ground-truth $f(x)$ and the optimal bounds $\mathrm{C}(x)$, $\mathrm{B}(x)$.

limits of the uncertainty quantification technique considered herein are. More concretely, is the width of the envelope $\mathrm{C}(x)-\mathrm{B}(x)$ restricted to a certain minimum value that cannot be reduced even with the addition of new data? From (2.22), it is clear that at any input location $x_i \in X$, $\mathrm{C}(x_i)$ and $\mathrm{B}(x_i)$ cannot be more than $2\bar{\delta}$ apart. In addition to that, the presence of the complexity constraint (2.21) can bring the two values closer to each other. Depending on how restrictive this latter constraint is for a given $x = x_i$, the corresponding output y_i might lie outside the interval between $\mathrm{C}(x_i)$ and $\mathrm{B}(x_i)$. In this case, the resulting width is considerably reduced as stated next and as illustrated in Figure 2.4.

Proposition 8. (Width smaller than the noise bound): If $\exists \mathsf{y}_i$ such that $y_{i,j} > C(x_i)$ or $y_{i,j} < \mathrm{B}(x_i)$ for some j, then $\mathrm{C}(x_i) - \mathrm{B}(x_i) \leq \bar{\delta}$.

Suppose now one has sampled (x_i, y_i) with $\mathsf{y}_i = \begin{bmatrix} y_{i,1} & y_{i,2} \end{bmatrix}^\top$, $y_{i,1} = f^\star(x_i) + \bar{\delta}$ and $y_{i,2} = f^\star(x_i) - \bar{\delta}$. In this case, there is no uncertainty whatsoever about $f^\star$ at x_i since $f^\star(x_i) = (y_{i,1} + y_{i,2})/2$ is the only possible value attainable by the ground-truth. The possibility of having multiple outputs at the same location therefore allows for the uncertainty interval to shrink past the $\bar{\delta}$ width, and eventually be reduced to a singleton as shown in Figure 2.4. Notwithstanding, the addition of a new datum to an existing dataset, be it in the form of a new output at an already sampled location or a completely new input-output pair, can only reduce the uncertainty as guaranteed by the following proposition.

Proposition 9. (Decreasing uncertainty) Let $C_1(x)$ be the solution of $\mathbb{P}1$ with a dataset $D_1 = \{(x_i, \mathsf{y}_i)\}_{i=1}^d$, and $C_2(x)$ the solution with $D_2 = D_1 \cup \{(x_{d+1}, \mathsf{y}_{d+1})\}$, $\forall x_{d+1} \in \mathcal{X}, \mathsf{y}_{d+1} \in \mathbb{R}^{n_{d+1}}$. Then $C_2(x) \leq C_1(x)$ for any $x \in \Omega$.

Remark 5. An analogous result holds for the lower part of the envelope $\mathrm{B}(x)$.

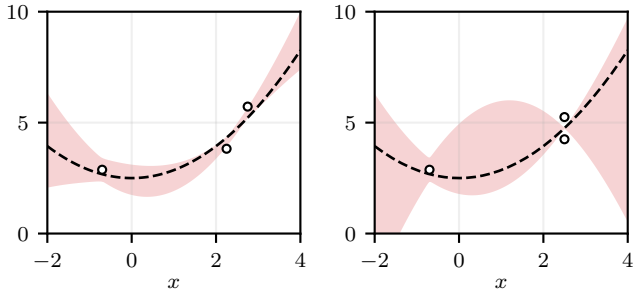


Figure 2.4: Left: samples lying outside of the uncertainty envelope, implying that its width is smaller than $\bar{\delta}$ at those locations. Right: redundant information is used to shrink the uncertainty envelope and recover the exact ground-truth value at $x = 2.5$ as $\mathrm{C}(2.5) = \mathrm{B}(2.5) = f^\star(2.5) = 4.75$.

The practical implications of Proposition 9 are shown in Figure 2.5, where samples are gradually added to the dataset, and the optimal bounds are re-computed. To create this example, the squared-exponential kernel was employed. Notice how new information decreases the uncertainty everywhere in the domain thanks to the global influence of the chosen kernel. This effect can be more clearly observed in the region $-8 \leq x \leq -5$ where the width is progressively reduced despite no new data-points being collected within it.

Remark 6. (On the accuracy of the noise bound): Recovering the ground-truth as shown in Figure 2.4 requires the noise realizations to match $\bar{\delta}$ and $-\bar{\delta}$; it is thus necessary to have *tight* noise bounds at hand for it to happen, which is not always a reality in practical applications. On the other hand, Proposition 9 guarantees the decreasing uncertainty property regardless of how accurate $\bar{\delta}$ is.

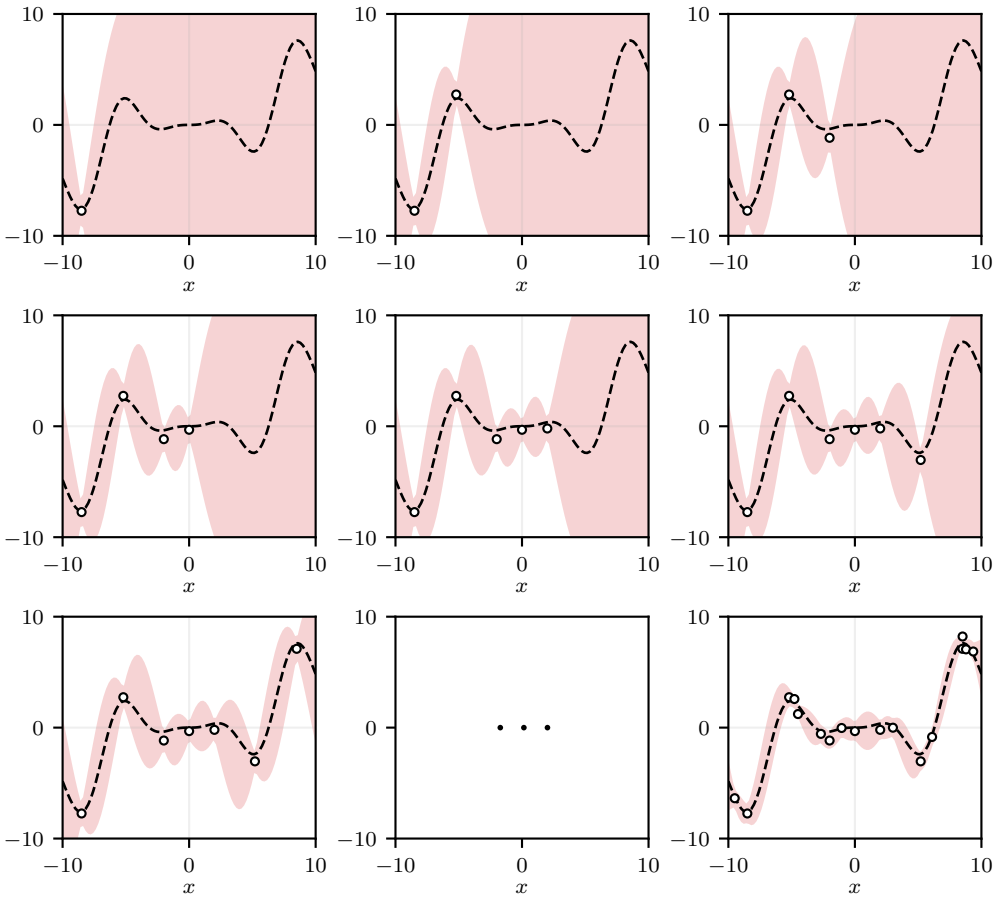


Figure 2.5: Adding new samples to a dataset can only cause uncertainty to be reduced everywhere in the domain. The noise was drawn from a uniform distribution bounded in absolute value by $\bar{\delta} = 0.8$. The last plot depicts the bounds after the collection of 10 new random samples.

One of the fundamental sources of computational complexity in problem (2.20) is the presence of the kernel matrix inverse. Indeed, it is well-known that commonly used algorithms for matrix inversion have cubic time-complexity. Note that the same problem is also faced when trying to scale other kernel-based algorithms (Zhang et al., 2013; Bauer et al., 2016; Lederer et al., 2021). In order to circumvent this obstacle, one could make use of the optimal-bounds dual formulation, which can be shown not to involve the aforementioned matrix.

Proposition 10. The Lagrangian dual of $\mathbb{P}1$ is the convex program $\mathbb{D}1$ given by

$$\min_{\nu\in\mathbb{R}^{\tilde{d}},\,\lambda>0} \frac{1}{4\lambda}\nu^\top \Lambda K_{XX}\Lambda^\top\nu+\left(\mathsf{y}-\frac{1}{2\lambda}\Lambda K_{Xx}\right)^\top\nu+\bar{\delta}\|\nu\|_1+\frac{1}{4\lambda}k(x,x)+\lambda\Gamma^2 \quad (2.26)$$

where $\tilde{d}=\sum_{i=1}^{d} n_i$ is the total number of outputs, that is, the size of y.

The optimization problem above is convex since it is a quadratic-over-linear function with $\Lambda K_{XX}\Lambda^\top \succeq 0$ and λ restricted to the positive reals. The objective can moreover be decomposed into a differentiable part and a single non-differentiable term $\|\nu\|_1$, with ν unconstrained. This class of problems has long been studied and mature numerical algorithms exist to solve them, notably different flavors of splitting methods such as the alternating direction method of multipliers (Boyd et al., 2011, §6). Alternatively, a standard linear reformulation could be employed to replace $\|\nu\|_1$ by $\sum_i \eta_i$, with additional constraints $-\nu\leq\eta$, $\nu\leq\eta$, yielding a differentiable objective, but with extra decision variables and linear constraints.

By definition, weak duality (Bertsekas, 2009, §5) ensures that any feasible solution (ν^*,λ^*) for $\mathbb{D}1$ leads to an objective value greater or equal to the primal problem $\mathbb{P}1$ optimal value. As a result, any feasible solution for $\mathbb{D}1$ returns a valid upper-bound for the ground-truth $f^\star(x)$. When used in real-time applications, users may thus choose not to solve $\mathbb{D}1$ to optimality since early-stopping solvers can always be done with a theoretical guarantee on the returned value. Next, a mild sufficient condition is given to ensure a zero duality gap between the primal and dual problems.

Proposition 11. (Strong duality): If $\bar{\delta}>\delta_{i,j},\forall i,j$ and $\Gamma>\|f^\star\|_{\mathcal{H}}$, then no duality gap exists, i.e., $\max\mathbb{P}1=\min\mathbb{D}1$.

2.3.3 A closed-form sub-optimal solution

The discussion in this subsection assumes that only one sample is present at each input location, i.e., $\mathsf{y}_i=y_i$ for $i=1,\ldots,d$, so that $\mathsf{y}=y$.

In order to alleviate the computational burden of having to solve two optimization

problems at each query point, closed-form equations can be employed instead. These expressions yield sub-optimal bounds around pre-specified kernel models of the form $s(x) = \alpha^\top K_{Xx}$, for some $\alpha \in \mathbb{R}^d$.

Proposition 12. Let Assumptions 1-4 hold and denote by $\{(x_i, y_i)\}_{i=1}^d$ the available data. Let $s(x) = \alpha^\top K_{Xx}$ be a nominal model for some $\alpha \in \mathbb{R}^d$. Then, for any $x \in \mathcal{X}$, $|s(x) - f^\star(x)| \leq S(x)$ with

$$\boxed{S(x) = P_X(x)\sqrt{\Gamma^2 + \tilde{\Delta}} + \bar{\delta}\left\|K_{XX}^{-1}K_{Xx}\right\|_1 + |\tilde{s}(x) - s(x)|} \tag{2.27}$$

where $\tilde{s}(x) = y^\top K_{XX}^{-1}K_{Xx}$, and the constant $\tilde{\Delta}$ is the minimum of the unconstrained convex problem $\min_{\nu\in\mathbb{R}^d}\left\{\frac{1}{4}\nu^\top K_{XX}\nu + \nu^\top y + \bar{\delta}\left\|\nu\right\|_1\right\}$.

Remark 7. We label the bounds provided by Proposition 12 as sub-optimal for $s(s) + S(x) \geq \mathrm{C}(X) \geq f^\star(x)$ and $s(s) - S(x) \leq \mathrm{B}(x) \leq f^\star(x)$ at any $x \in \mathcal{X}$.

The map $\tilde{s}(x) = y^\top K_{XX}^{-1}K_{Xx}$ is an interpolant for the available outputs y. Note also that none of the terms in (2.27) depend on the model weights α with the exception of the last term $|\tilde{s}(x)-s(x)|$. Therefore, the width $S(x)$ will be minimized when $s(x) = \tilde{s}(x) \Leftrightarrow \alpha = y^\top K_{XX}^{-1}$. Such a model choice is however not always desirable and at times a balance between smoothing the data and not diverging too much from $\tilde{s}(x)$ has to be found. This trade-off is illustrated in Figure 2.6 where the optimal bounds are compared against KRR sub-optimal ones built with the same RKHS norm estimate $\Gamma = 1.1\left\|f^\star\right\|_{\mathcal{H}}$ and noise bound $\bar{\delta} = 1$. Three regularization constants were employed when designing the nominal models (2.16). As can be seen from the plots, the sub-optimal bounds are always more conservative than the optimal ones and its conservativeness increases with λ.

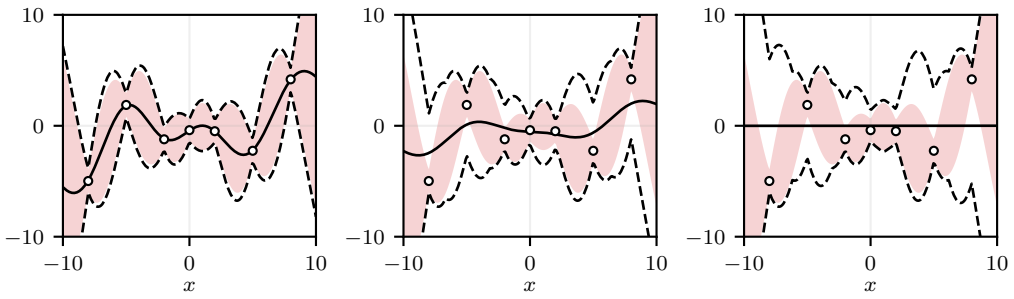


Figure 2.6: A comparison between the optimal bounds (red envelopes) and the closed-form sub-optimal bounds (dashed black lines) around KRR models (solid black lines). Three KRR distinct regularization constants were tested $\lambda = 10^{-3}$ (left), $\lambda = 10^{-1}$ (center) and $\lambda = 10^{2}$ (right).

Let us inspect more closely the source of sub-optimality in Proposition 12. To do so, the optimal bounds problem (2.20) will be reformulated and relaxed, allowing for a closed-form solution to be found.

Begin by employing a change of variables in $\mathbb{P}1$ and optimizing over $(\delta, c_x), \delta := c - y$ rather than over (c, c_x). Next, apply the matrix inversion lemma (A.23) to decompose the quadratic constraint (2.21), carry out the vector-matrix-vector multiplication and solve for c_x. This process leads to

$$c_x \leq P_X(x)\sqrt{\Gamma^2 - \|\tilde{s}\|^2_{\mathcal{H}} - \delta^\top K_{XX}^{-1}\delta + 2y^\top K_{XX}^{-1}\delta} + \tilde{s}(x) + \delta^\top K_{XX}^{-1}K_{Xx} \tag{2.28}$$

where $P_X^2(x) = k(x,x) - K_{xX}K_{XX}^{-1}K_{Xx}$, $\tilde{s}(x) = y^\top K_{XX}^{-1}K_{Xx}$ and $\|\tilde{s}\|^2_{\mathcal{H}} = y^\top K_{XX}^{-1}y$. Since the norm constraint (2.22) is independent of c_x, (2.28) will always be active and we can optimize over the right-hand side of (2.28) instead, thus eliminating one decision variable

$$\max_{\|\delta\|_\infty \leq \bar{\delta}} \; P_X(x)\sqrt{\Gamma^2 - \|\tilde{s}\|_{\mathcal{H}} - \delta^\top K_{XX}^{-1}\delta + 2y^\top K_{XX}^{-1}\delta} + \tilde{s}(x) + \delta^\top K_{XX}^{-1}K_{Xx} \tag{2.29}$$

Now, relax the problem by allowing δ to attain different values inside and outside the square-root

$$\max_{\|\delta_1\|_\infty, \|\delta_1\|_\infty \leq \bar{\delta}} P_X(x)\sqrt{\Gamma^2 - \|\tilde{s}\|^2_{\mathcal{H}} - \delta_1^\top K_{XX}^{-1}\delta_1 + 2y^\top K_{XX}^{-1}\delta_1} + \tilde{s}(x) + \delta_2^\top K_{XX}^{-1}K_{Xx} \tag{2.30}$$

The above objective is separable and the terms associated with δ_2 evaluate to $\max_{\delta_2 \in \mathbb{R}^d}\{\delta_2^\top K_{XX}^{-1}K_{Xx} : \|\delta_2\|_\infty \leq \bar{\delta}\} = \bar{\delta}\,\|K_{XX}^{-1}K_{Xx}\|_1$ since these norms are duals of each other (Boyd and Vandenberghe, 2004, §A.1.6). Notice how what is inside the square-root is independent of the parameter x and, therefore, only needs to be evaluated once for a fixed set of inputs X. Thanks to the strong duality of quadratic programming, we have that $\max_{\delta_1 \in \mathbb{R}^d}\left\{-\delta_1^\top K_{XX}^{-1}\delta_1 + 2y^\top K_{XX}^{-1}\delta_1 - \|\tilde{s}\|^2_{\mathcal{H}} : \|\bar{\delta}_1\|_\infty \leq \bar{\delta}\right\}$ is equal to $\min_{\nu \in \mathbb{R}^d}\left\{\frac{1}{4}\nu^\top K_{XX}\nu + \nu^\top y + \bar{\delta}\,\|\nu\|_1\right\}$, which by definition is $\tilde{\Delta}$. Finally, recall that (2.30) was a (conservative) upper bound for $f^\star(x)$. Given an arbitrary model $s(x)$, the triangle inequality $|f(x) - s(x)| \leq |f(x) - \tilde{s}(x)| + |\tilde{s}(x) - s(x)|$ can be used to bound the distance between its predictions and the ground-truth values, where $|f(x) - \tilde{s}(x)|$ comes from the derivations made in this paragraph. We have thus at the same expressions presented in Proposition 12.

Remark 8. The proof of Proposition 12 presented in Section 2.6.4 follows a different, simpler argumentation. The derivation presented here however better highlights how the relaxed maximizaition (2.30) takes into account the worst-possible inner-product $\delta_2^\top K_{XX}^{-1}K_{Xx}$ and norm term associated with δ_1 jointly. Besides, Proposition 12 is also seen to strongly rely on the interpolant $\tilde{s}(x)$ as shown

by the use of the triangle-inequality. Despite this fact, we have experimentally achieved reasonable results when in moderate noise level scenarios.

Remark 9. The sub-optimal bounds presented in this subsection feature a nominal model at their center, which is desirable in many practical situations. In the optimal scenario, the minimum norm regressor $s^\star(x) = \alpha^{\star\top} K_{Xx}$, $\alpha^\star = \arg\min_{\alpha\in\mathbb{R}^d}\{\alpha^\top K_{XX}\alpha : \|K_{XX}\alpha - y\|_\infty \leq \bar{\delta}\}$ can be used as a nominal model. This choice is guaranteed to lie completely within $\mathrm{C}(x)$ and $\mathrm{B}(x)$|although not necessarily in the middle|since the map $s^\star$ belongs to $\mathcal{H}$ and is a feasible solution for $\mathbb{P}0$.

2.4 Numerical examples

2.5 Conclusions and outlook

2.6 Appendices

2.6.1 Estimating kernel hyperparameters

Kernel functions typically feature a number of internal constants that need to be specified by the user, the so-called *hyperparameters*. Although some theoretical properties remain insensitive to the final choice of hyperparameters[3], the numerical stability and real-world performance of kernel algorithms highly depend on the tuning of these numbers (Fasshauer, 2011).

One popular approach to optimizing kernel hyperparameters is to consider the log marginal likelihood objective of a Gaussian process with Gaussian measurement noise, and apply a gradient-based numerical algorithm to it (Williams and Rasmussen, 2006, §5.4.1). This amounts to solving a smooth, unconstrained non-convex optimization problem to local optimality. Two appealing ... of this approach are the continuous nature of the search and the inherent regularization properties of the objective, known to combat overfitting.

Bayesian optimization (BayesOpt) is another widely adopted methodology to fine-tune hyperparameters against a pre-specified objective function (Snoek et al., 2012; Shahriari et al., 2015). BayesOpt operates on pairs of hyperparameters and their associated objective function values, and constructs a model for their unknown relationship. Next, a so-called *acquisition function* based on the model

[3]For instance, the squared-exponential kernel is universal (Definition 5) regardless of the its lengthscale value. On the other hand, it is known that if k_ℓ is a squared-exponential kernel with lengthscale ℓ, then $\mathcal{H}_{k_{\ell 1}}$... cite Ingo's book

is then employed to tell bad hyperparameters from promising ones (Wilson et al., 2018). A new set of parameters is finally chosen to be experimented with and its performance is measured. This new piece of information is incorporated into the dataset and the process is repeated until some termination criterion is reached. Interestingly, the most popular model class used when reconstructing the aforementioned unknown relationship is Gaussian process, which is itself based on kernels whose own hyperparameters have to be set by the users.

Lastly, we could also mention the different flavors of cross-validation (CV). The procedure builds an estimate of how suitable a set of hyperparameters are by evaluating the resulting model performance on unseen data. More specifically, it consists in shuffling and splitting the available data into batches, say N; making use of $N-1$ batches to define the model and the last one to assess its performance; the process is repeated N times rotating the batches and yielding N performance measures, which could be combined through averaging for instance. By following this algorithm, the most suitable hyperparameter value could be found from a predefined list of possible values. Note that, the log marginal likelihood itself could be a valid performance objective within CV (Williams and Rasmussen, 2006, §5.4.2).

2.6.2 Estimating RKHS norms

Members $f \in \mathcal{H}$ are either finite weighted sums of partially evaluated kernels $k(x, \cdot)$ or limits of sequences of such sums (see Section 2.1). Assume f enjoys a finite expansion of N terms, then

$$\|f\|_{\mathcal{H}}^2 = \langle f, f \rangle_{\mathcal{H}} \tag{2.31}$$

$$= \left\langle \sum_{i=1}^{N} \alpha_i k(x_i, \cdot), \sum_{i=1}^{N} \alpha_i k(x_i, \cdot) \right\rangle_{\mathcal{H}} \tag{2.32}$$

$$= \alpha^\top K_{XX} \alpha \tag{2.33}$$

where (2.31) is the RKHS norm definition, (2.32) is the inner-product definition, and (2.33) follows from the inner-product linearity and from the reproducing property of kernels (see (2.9)). As a result, the norm $\|f\|_{\mathcal{H}}$ can be exactly and easily computed as long as we have at hand the weights α that define f.

In practice, it is hard to imagine a situation where the coefficients α would be known for a given physical system. Suppose however that we have a dataset at hand $\{(x_i, f_{x_i})\}_{i=1}^n$, where $f_{x_i} = f(x_i)$. The inputs x_i need to be pairwise-distinct. Assume that k is SDP and consider the function $s_n(x) := \sum_{i=1}^n \alpha_i k(x_i, x)$,

$\alpha = K_{XX}^{-1} f_X$, which reproduces our dataset at every input, i.e., $s_n(x_i) = f_{x_i}$. According to the well-known optimal recovery property (Iske, 2018, §8.3), s_n not only interpolates the dataset, but also attains a minimum RKHS norm while doing so and, moreover, $\|s_n\|_{\mathcal{H}} \leq \|f\|_{\mathcal{H}}$ for any number of samples n. In view of the quadratic form (2.33), it is evident that for any new pair $(x_{n+1}, f_{x_{n+1}})$ added to the dataset, $\|s_{n+1}\|_{\mathcal{H}} \geq \|s_n\|_{\mathcal{H}}$ holds. One can finally show that if samples are acquired distributed enough to fill the domain[4], then $\|f\|_{\mathcal{H}}$ is the least upper bound of the $\|s_n\|_{\mathcal{H}}$ sequence and, from the monotone convergence theorem, $\|s_n\|_{\mathcal{H}} \to \|f\|_{\mathcal{H}}$ is guaranteed.

The discussion above showed that the RKHS norm $\|f\|_{\mathcal{H}}$ can indeed be estimated from below from samples $\{(x_i, f_{x_i})\}_{i=1}^n$. This is done by simply evaluating the norm of the interpolant for which we know the weights α

$$\|s_n\|_{\mathcal{H}}^2 = \alpha^\top K_{XX} \alpha = f_X^\top K_{XX}^{-1} f_X \tag{2.34}$$

The more samples we have, the closer the quadratic form will be from the target value $\|f\|_{\mathcal{H}}$. Consider the example below for observations on how the estimation process unfolds in a practical scenario.

Let $k : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be the squared-exponential with lengthscale $\ell = 0.5$. A member $f \in \mathcal{H}$ of its RKHS is shown in Figure 2.7 (left), being composed of 100 partially evaluated kernels whose centers and weights were randomly generated. The exact $\|f\|_{\mathcal{H}}$ value was computed through (2.33), yielding $\|f\|_{\mathcal{H}} = 11.24$. Next, the same quantity was estimated through data simply drawn randomly from the domain, following a uniform distribution. The associated $\|s_n\|_{\mathcal{H}}$ from $n = 1$ to 60 is shown in Figure 2.7 (right).

Up to this point, only noiseless samples were consider. Suppose now the data

[4] minimize the fill distance

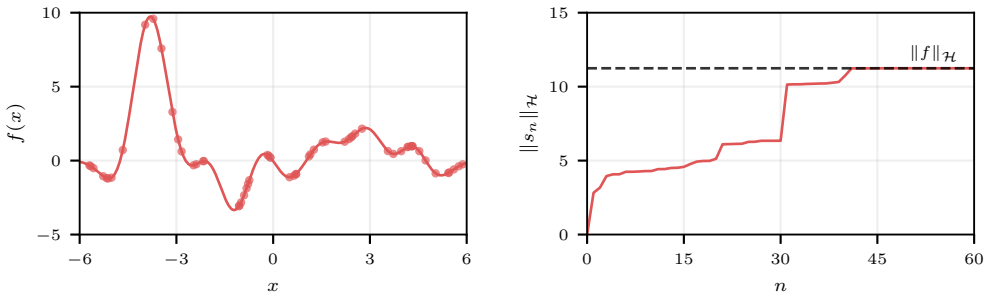


Figure 2.7: Bla bla bla.

come in the form $\{(x_i, y_i)\}_{i=1}^n$ with x_i pairwise distinct and $y_i = f(x_i) + \delta_i$. Let the additive noise be bounded by some known value $\bar{\delta} \geq |\delta_i|$ for all i. Since we do not have access to the evaluations of f anymore, $\|s_n\|_{\mathcal{H}}$ cannot be computed. If we naively interpolate the data with the model $\tilde{s}_n(x) := \sum_{i=1}^n \alpha_i k(x_i, x)$, $\alpha = K_{XX}^{-1} y$, the resulting norm $\|\tilde{s}_n\|_{\mathcal{H}}^2 = y^\top K_{XX}^{-1} y$ can be either larger or smaller than its noise-free counterpart. The mismatch between the two will depend on how each δ_i will disturb the samples, and its maximum effects can be exactly computed.

Proposition 13. Let $\{(x_i, y_i)\}_{i=1}^n$ be such that x_i are pairwise distinct and $y_i = f(x_i) + \delta_i$ with $\bar{\delta} \geq |\delta_i|$ for all i. Let s_n and $\tilde{s}_n$ be respectively the models interpolating the noise-free f_X and the noisy y values of f, i.e., $s_n(x) = \sum_{i=1}^n \alpha_i k(x_i, x)$, $\alpha = K_{XX}^{-1} f_X$ and $\tilde{s}_n(x) = \sum_{i=1}^n \alpha_i k(x_i, x)$, $\alpha = K_{XX}^{-1} y$. It then holds that

$$\nabla \leq \|\tilde{s}_n\|_{\mathcal{H}}^2 - \|s_n\|_{\mathcal{H}}^2 \leq \Delta \tag{2.35}$$

where Δ and ∇ denote respectively the maximum and minimum of $-\delta^\top K_{XX}^{-1} \delta + 2y^\top K_{XX}^{-1} \delta$ over δ, subject to $|\delta| \leq \bar{\delta}$.

Calculating Δ amounts to solving a convex optimization problem since it is the maximum of a strictly concave function, whereas ∇ is not as simple.

Proposition 14. Let all assumption listed in Proposition 13 hold and, additionally, δ be a random vector with $\mathbb{E}(\delta) = \mu$ and $\mathbb{V}(\delta) = \Sigma$, then

$$\mathbb{E}(\|\tilde{s}_n\|_{\mathcal{H}}^2 - \|s_n\|_{\mathcal{H}}^2) = \mu^\top K_{XX}^{-1} \mu + \operatorname{Tr}(K_{XX}^{-1} \Sigma) + 2y^\top K_{XX}^{-1} \mu \tag{2.36}$$

Moreover, if δ follows a uniform distribution $\mathcal{U}(-\bar{\delta}, \bar{\delta})$, then $\mathbb{E}(\|\tilde{s}_n\|_{\mathcal{H}}^2 - \|s_n\|_{\mathcal{H}}^2) = \dots$

2.6.3 Auxiliary definitions

Recall that $n_1, n_2, \dots, n_d$ are the number of outputs available at the input locations $x_1, x_2, \dots, x_d$. Let Λ be the matrix of size $(\sum_i n_i) \times d$ defined as

$$\Lambda := \begin{bmatrix} \mathbf{1}_{n_1} & \mathbf{0}_{n_1} & \mathbf{0}_{n_1} & \cdots & \mathbf{0}_{n_1} \\ \mathbf{0}_{n_2} & \mathbf{1}_{n_2} & \mathbf{0}_{n_2} & \cdots & \mathbf{0}_{n_2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{n_d} & \mathbf{0}_{n_d} & \mathbf{0}_{n_d} & \cdots & \mathbf{1}_{n_d} \end{bmatrix} \tag{2.37}$$

where $\mathbf{1}_{n_i}$ and $\mathbf{0}_{n_i}$ are respectively column vectors of ones and zeros of size n_i. If only a single output is available at every input, Λ simplifies to an identity matrix.

The column vector f_X is made of function evaluations $f(x_i)$, which are repeated

whenever multiple outputs are available at the same input location x_i. More concretely, $f_X := \Lambda \begin{bmatrix} f(x_1) & \dots & f(x_d) \end{bmatrix}^\top$.

2.6.4 Selected proofs

Proof of Proposition 7: It follows from the objective being linear and only sensitive to c_x and from reformulating (2.21). More concretely, we use the matrix inversion lemma and the definition of $P_X(x)$ to re-write the complexity constraint as

$$\begin{bmatrix} c \\ c_x \end{bmatrix}^\top \begin{bmatrix} K_{XX} & K_{Xx} \\ K_{xX} & k(x,x) \end{bmatrix}^{-1} \begin{bmatrix} c \\ c_x \end{bmatrix} \leq \Gamma^2 \tag{2.38a}$$

$$\Leftrightarrow\ c^\top K_{XX}^{-1} c + P_X^{-2}(x) \left(c^\top K_{XX}^{-1} K_{Xx} - c_x \right)^2 \leq \Gamma^2 \tag{2.38b}$$

Note that $P_X^{-2}(x) > 0$ since $P_X(x) > 0$ for any x not in X (Karvonen, 2022). As a result, we see that (2.38b) depends quadratically on c_x. Therefore, for any feasible (c, c_x) such that (2.38a) inactive, there exists $\tilde{c}_x := c_x + \varepsilon, \varepsilon > 0$ such that $(c, \tilde{c}_x)$ attains a higher objective while satisfying the constraints. □

Proof of Proposition 9: Denote by $\mathbb{P}1_1$ the problem solved with D_1 and decision variables $\begin{bmatrix} c & c_x \end{bmatrix}$. Similarly, $\mathbb{P}1_2$ is associated with the dataset D_2 and the decision variables $\begin{bmatrix} c & c_x & c_z \end{bmatrix}$, where c_z are due to the additional input in D_2. Since D_2 contains all members of D_1, the ∞-norm constraint of $\mathbb{P}1_2$ can be recast as that of $\mathbb{P}1_1$ and an additional constraint for c_z and the new outputs. Let $\mathbb{X} := X \cup \{x\}$, $\bar{c} := \begin{bmatrix} c^\top c_x \end{bmatrix}^\top$ and $z := x_{d+1}$ be shorthand variables to ease notation. The complexity constraint of $\mathbb{P}1_2$ is then

$$\begin{bmatrix} \bar{c} \\ c_z \end{bmatrix}^\top \begin{bmatrix} K_{\mathbb{X}\mathbb{X}} & K_{\mathbb{X}z} \\ K_{z\mathbb{X}} & k(z,z) \end{bmatrix}^{-1} \begin{bmatrix} \bar{c} \\ c_z \end{bmatrix} \leq \Gamma^2 \tag{2.39a}$$

$$\overset{(i)}{\Leftrightarrow} \bar{c}^\top K_{\mathbb{X}\mathbb{X}}^{-1} \bar{c} + P_{\mathbb{X}}^{-2}(z) \left\| \begin{bmatrix} K_{\mathbb{X}\mathbb{X}}^{-1} K_{\mathbb{X}z} \\ -1 \end{bmatrix} \begin{bmatrix} \bar{c} \\ c_z \end{bmatrix} \right\|_2^2 \leq \Gamma^2 \tag{2.39b}$$

$$\overset{(ii)}{\Leftrightarrow} \begin{bmatrix} c \\ c_x \end{bmatrix}^\top \begin{bmatrix} K_{XX} & K_{Xx} \\ K_{xX} & k(x,x) \end{bmatrix}^{-1} \begin{bmatrix} c \\ c_x \end{bmatrix} + P_{\mathbb{X}}^{-2}(z) \left(\bar{c}^\top K_{\mathbb{X}\mathbb{X}}^{-1} K_{\mathbb{X}z} - c_z \right)^2 \leq \Gamma^2 \tag{2.39c}$$

where the matrix identity found in Appendix **??** was used in (i) and $P_{\mathbb{X}}^2(z) = k(z,z) - K_{z\mathbb{X}} K_{\mathbb{X}\mathbb{X}}^{-1} K_{\mathbb{X}z}$. In (ii), the definitions of $\bar{c}$ and $\mathbb{X}$ were used. Thanks to $P_{\mathbb{X}}(z) \geq 0, \forall z$ and the quadratic term multiplying it, we conclude that for any choice of the decision variable c_z, (2.39c) is a tightened version of the complexity constraint of $\mathbb{P}1_1$, which is (**??**). As a result, the maximum of $\mathbb{P}1_2$ is lower or

equal than that of $\mathbb{P}1_1$. □

Proof of Proposition 10: Consider the case $x \notin X$. Let $z := \begin{bmatrix} c^\top & c_x \end{bmatrix}^\top$, $a := \begin{bmatrix} \mathbf{0}^\top & 1 \end{bmatrix}^\top$, $A := \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix}$. The Lagrangian of $\mathbb{P}1$ is

$$\mathcal{L}(z, \lambda, \beta, \gamma) = a^\top z - \lambda(z^\top K_{\mathbb{X}\mathbb{X}}^{-1} z - \Gamma^2) - \beta^\top(\Lambda A z - \mathsf{y} - \bar{\delta}\mathbf{1}) - \gamma^\top(\mathsf{y} - \Lambda A z - \bar{\delta}\mathbf{1}) \quad (2.40)$$

where $K_{\mathbb{X}\mathbb{X}}$ denotes the kernel matrix evaluated at $X \cup \{x\}$. Suppose $\lambda > 0$. Computing $\nabla_z \mathcal{L}(z^\star) = 0$ leads to

$$z^\star = -\frac{1}{2\lambda} K_{\mathbb{X}\mathbb{X}} \left(A^\top \Lambda^\top (\beta - \gamma) - a \right).$$

Defining the auxiliary variable $\nu = \beta - \gamma$, and substituting $z^\star$ into (2.40) gives the dual objective

$$\begin{aligned} g(\lambda, \nu) &= \frac{1}{4\lambda} \nu^\top \Lambda A K_{\mathbb{X}\mathbb{X}} A^\top \Lambda^\top \nu + \left(\mathsf{y} - \frac{1}{2\lambda} \Lambda A K_{\mathbb{X}\mathbb{X}} a \right)^\top \nu \\ &\quad + \bar{\delta} \|\nu\|_1 + \frac{1}{4\lambda} a^\top K_{\mathbb{X}\mathbb{X}} a + \lambda \Gamma^2 \qquad (2.41) \\ &= \frac{1}{4\lambda} \nu^\top \Lambda K_{XX} \Lambda^\top \nu + \left(\mathsf{y} - \frac{1}{2\lambda} \Lambda K_{Xx} \right)^\top \nu \\ &\quad + \bar{\delta} \|\nu\|_1 + \frac{1}{4\lambda} k(x, x) + \lambda \Gamma^2 \qquad (2.42) \end{aligned}$$

where in the second equality the matrix $K_{\mathbb{X}\mathbb{X}}$ was expanded and the resulting terms were reorganized. Since $\beta, \gamma \in \mathbb{R}^{\tilde{d}}_{\geq 0}$ and $\nu = \beta - \gamma$ then ν is unconstrained.

Now if $\lambda = 0$, the Lagrangian (2.40) simplifies to $\mathcal{L}(z, \nu) = (a - A^\top \Lambda^\top \nu)^\top z + \nu^\top \mathsf{y} + \bar{\delta} \|\nu\|_1$, which is linear in z. Its supremum w.r.t. z is only finite if $a = A^\top \Lambda^\top \nu$. Recalling the definitions of a, A and Λ, one can see that $\nexists \nu$ that could satisfy the latter condition. Therefore, $\lambda = 0 \implies \sup_z \mathcal{L}(z, \lambda, \nu) = +\infty$, meaning that the dual problem is infeasible. As a conclusion, the Lagrangian dual of $\mathbb{P}1$ in (**??**) is precisely $\mathbb{D}1$ in (2.26).

Next, consider the case $x \in X$, $x = x_i$. The objective of $\mathbb{P}1'$ can be written as $a^\top c$ with $a_i = 1$ and $a_n = 0, n \neq i$. When deriving its Lagrangian, one obtains again (2.40) with the simplifications: $z \leftarrow c$, $K_{\mathbb{X}\mathbb{X}} \leftarrow K_{XX}$ and $A \leftarrow \mathbf{I}$. We proceed by analyzing the two scenarios for λ as before. If $\lambda > 0$, the previous derivations apply, leading to the same the quadratic-over-linear objective (2.42). However, if $\lambda = 0$, the Lagrangian becomes $\mathcal{L}(z, \nu) = (a - \Lambda^\top \nu)^\top z + \nu^\top \mathsf{y} + \bar{\delta} \|\nu\|_1$, whose supremum w.r.t. z is only finite if $a = \Lambda^\top \nu$. In contrast with the previous paragraph, this condition now can be satisfied. It is equivalent to $\nu_{i,1} + \cdots + \nu_{i,n_i} = 1$, where

the variables are all the multipliers associated with the i-th input location x_i. The resulting expression can be minimized analytically, yielding the minimum $\min_j y_{i,j} + \bar{\delta}$, i.e., the smallest output available at x_i augmented by the noise bound. Finally, we conclude that the dual objective for $\mathbb{P}1'$ is

$$g(\lambda, \nu) = \begin{cases} (2.42), & \text{if } \lambda > 0 \\ \min_j y_{i,j} + \bar{\delta}, & \text{if } \lambda = 0 \end{cases} \tag{2.43}$$

As a last observation, a dual problem can also be derived for (**??**), calculating the lower part of the envelope. The formulation is analogous to (2.26), assuming the form

$$\max_{\nu \in \mathbb{R}^{\tilde{d}}, \lambda > 0} \; -\frac{1}{4\lambda} \nu^\top \Lambda K_{XX} \Lambda^\top \nu - \left(\mathsf{y} + \frac{1}{2\lambda} \Lambda K_{Xx} \right)^\top \nu - \bar{\delta} \|\nu\|_1 - \frac{1}{4\lambda} k(x,x) - \lambda \Gamma^2 \tag{2.44}$$

Note that these are distinct objectives, not merely opposites. Therefore, two problems have to be solved to fully quantify the ground-truth uncertainty. □

Proof of Proposition 11: Consider the primal problem $\mathbb{P}1$ and select $c = f^\star_X$ and $c_x = f^\star(x)$. Let $\mathbb{X} := X \cup \{x\}$ and $K_{\mathbb{X}\mathbb{X}}$ denote the kernel matrix associated with $\mathbb{X}$. Thanks to the optimal recovery property (Wendland, 2004, Theorem 13.2), $\begin{bmatrix} c^\top & c_x \end{bmatrix} K_{\mathbb{X}\mathbb{X}} \begin{bmatrix} c^\top & c_x \end{bmatrix}^\top \leq \|f^\star\|^2_{\mathcal{H}}$, which in turn is strictly smaller than Γ^2 by assumption. Also, $\|\Lambda c - \mathsf{y}\|_\infty = \|\Lambda f^\star_X - \mathsf{y}\|_\infty = \left\| \begin{bmatrix} \delta_{1,1} & \dots & \delta_{2,1} & \dots \end{bmatrix}^\top \right\|_\infty < \bar{\delta}$. Therefore, the ground-truth values constitute a feasible solution that lies in the interior of the primal problem feasible set. As a result, Slater's condition is met and, since the primal is convex, there is no duality gap. □

Proof of Proposition 8: Follows from $\mathrm{C}(x_i) \geq \mathrm{B}(x_i)$, $\mathrm{C}(x_i) \leq y_{i,j} + \bar{\delta}$ and $\mathrm{B}(x_i) \geq y_{i,j} - \bar{\delta}$ for any $i = 1, \dots, d$ and any $j = 1, \dots, n_i$. □

Proof of Proposition 13: It follows from expanding $\|s_n\|^2_{\mathcal{H}}$ and $\|\tilde{s}_n\|^2_{\mathcal{H}}$. □

Proof of Proposition 12: For any given $s(x) = \alpha^\top K_{Xx}$, we have

$$\begin{aligned}
|f^\star(x) - s(x)| &= |f^\star(x) - \tilde{s}(x) + \tilde{s}(x) - s(x)| \\
&\leq |f^\star(x) - (f^\star_X + \delta_X) K^{-1}_{XX} K_{Xx}| + |\tilde{s}(x) - s(x)| && (2.45) \\
&\leq |f^\star(x) - \bar{s}(x)| + \bar{\delta} \left\| K^{-1}_{XX} K_{Xx} \right\|_1 + |\tilde{s}(x) - s(x)| && (2.46) \\
&\leq P(x) \sqrt{\Gamma^2 - \|\bar{s}\|^2_{\mathcal{H}}} + \bar{\delta} \left\| K^{-1}_{XX} K_{Xx} \right\|_1 + |\tilde{s}(x) - s(x)| && (2.47) \\
&\leq P(x) \sqrt{\Gamma^2 + \Delta - \|\tilde{s}\|^2_{\mathcal{H}}} + \bar{\delta} \left\| K^{-1}_{XX} K_{Xx} \right\|_1 + |\tilde{s}(x) - s(x)| && (2.48)
\end{aligned}$$

with $f_X^\star$ being the vector of true function values at the sample locations in X and δ_X the vector of additive measurement noise for the samples y. (2.45) follows from the triangle inequality and the additive noise property of y. Using the triangle inequality again, we arrive at (2.46), where $\bar{s}$ denotes the noise-free interpolant of $f_X^\star$. The noise-free interpolation error bound gives the estimation in the first term of (2.47), while (2.48) follows from (**?**, Lemma 1), with $\Delta = \max_{\|\delta\|_\infty \leq \bar{\delta}}(-\delta^\top K_{XX}^{-1}\delta + 2y^\top K_{XX}^{-1}\delta)$. A standard dualization procedure as the one presented in Appendix **??** leads to the dual problem

$$\min_{\nu \in \mathbb{R}^d} \frac{1}{4}\nu^\top K_{XX}\nu + \nu^\top y + \bar{\delta}\,\|\nu\|_1 + y^\top K_{XX}^{-1} y \tag{2.49}$$

for Δ. Notice that the last term in (2.49) is constant and the same as the squared interpolant norm $\|\tilde{s}\|_{\mathcal{H}}^2$. Therefore, these terms cancel in (2.48) and we are left with

$$|f^\star(x) - s(x)| \leq\ P(x)\sqrt{\Gamma^2 + \tilde{\Delta}} + \bar{\delta}\left\|K_{XX}^{-1}K_{Xx}\right\|_1 + |\tilde{s}(x) - s(x)| \tag{2.50}$$

where $\tilde{\Delta}$ represents (2.49) without the constant term. □

3 Experimenting with Gaussian processes

In this chapter, we discuss the problem of learning and elucidate what viewpoint will be taken to tackle it. Next, novel results are presented concerning uncertainty estimation in a kernelized setting. Finally, some examples are given to illustrate the general use of the theory.

3.1 The control problem

3.2 The building and its HVAC system

The building considered in this study was a surgery center situated in the São Julião hospital complex, in the city of Campo Grande, MS, Brazil (Figure **??**, left). The 51 rooms that compose it are in permanent use and, for information purposes, 528 surgical procedures were carried out in it during August 2021. We were concerned with three thermal zones in its ophthalmology section: two operating rooms (ORs) and one waiting room (WR), all located on the West end of the building (Figure **??**, right). Whereas the former rooms are only connected to the waiting room, the latter has a door to the rest of the surgery center. Opaque glass bricks are present in the waiting room as can be seen in the picture, allowing some natural light to enter the space; the operating rooms on the other hand do not feature them, nor do they have any windows. All spaces have exterior walls, but the right-hand side operating room is significantly more affected by direct solar radiation due to the disposition of the nearby trees.

A forced-air HVAC plant is in place to provide the occupants with a suitable indoor climate in accordance with local regulations. A total of seven air-handling units (AHUs) collect outdoor air that is then treated and filtered before being pumped into the several indoor spaces. We had control only over three AHUs, one

for each aforementioned thermal zone. A central chiller connected to an external cooling tower provides chilled water to all AHUs, which in turn feature three-way valves to control the flow of water through their cooling coils. The AHU fans are operated always at constant speed, resulting in a constant volumetric flow through the air-ducts and into the zones. As per the regulations, no air recycling is possible and all return air is directly discharged into the atmosphere. As the temperature in Campo Grande is typically high, the HVAC system was conceived to only cool the space, not having the means to provide positive thermal energy (for more details, see Section 3.2.1).

Two distinct sensor networks were deployed to monitor the HVAC plant and the indoor spaces. Firstly, we will describe the one located in the AHU room. One local controller (LCO)–a National Instruments myRIO–was attached to each air-handling unit, reading all sensors used to monitor the AHUs: supply and return water temperature probes, a water flow meter, an anemometer, as well as an angular position sensor. The LCOs were moreover responsible for running low-level signal processing routines and implementing control actions, i.e., acting on the three-way valve servomotor to change the chilled water flow, hence influencing the supply air temperature. Photos of the AHU room are shown in Figure 3.2. Next, in order to measure the indoor temperatures in a flexible way, a wireless network of Z-wave sensors was set up in the operating rooms and waiting room. These were equipped with external temperature probes (Dallas DS18B20) to guarantee fast and precise readings, reporting their measurements periodically to a local computer (LC) that featured a Z-wave transceiver attached to it.

A 3 GHz, 16 GB RAM, core i7 machine was installed in the waiting room, acting as the main computer platform for the project, i.e., the LC. This computer and the AHU LCOs were all connected to a local area network to exchange information, which was done by using the UDP protocol at a rate of approximately 1 Hz. Lastly, a weather station was deployed on site to measure the outdoor temperature and the solar radiation acting on the building with high accuracy. All signals were sampled with a period of two mins and stored into a local time-series database, InfluxDB. A block-diagram of the complete system is depicted in Figure **??**.

3.2.1 Analysis of the control problem

The control goal is to regulate the indoor temperature within the three zones (T_i, $i = 1, 2, 3$), keeping it always below a pre-specified value $T_{\max}$. Although defining two-level temperature envelopes is common for residences and offices (see e.g. **??**), some employees still make use of the surgery center spaces during nighttime and,

Figure 3.1: Photos of the AHU room depicting the air ducts (top), the supply and return water pipes (top and bottom), and the three-way valve servomotor (bottom).

thus, the indoor temperature has to stay below $T_{\max}$ even then. Furthermore, this is to be done while minimizing the chiller energy consumption as dictated by its coefficient of performance (COP) curve and the building thermal load. The controlled variables are the angular positions of each AHU three-way valve (θ_i, $i = 1, 2, 3$) that regulate the flow of water across their cooling coils. Naturally, these quantities are physically limited between a minimum $\theta_{\min}$ and a maximum value $\theta_{\max}$.

Several disturbances both of internal and external nature act on the system. The measured ones include the outdoor temperature T_{out}, the solar radiation R_{sol}, and the temperature of the water supplied by the chiller to the AHUs T_{sup}. The variables T_{out} and R_{sol} directly affect the indoor climate by heating the external walls. T_{out} and T_{sup} can be regarded as an input disturbance; indeed, these quantities define the HVAC system actuation capabilities along with the valve positions θ_i. The unmeasured disturbances are the internal heat gains generated by occupants and equipment, as well as the eventual opening and closing of doors that lead to air mix among rooms. A summary of the relevant control information described here can be found in Table 3.1.

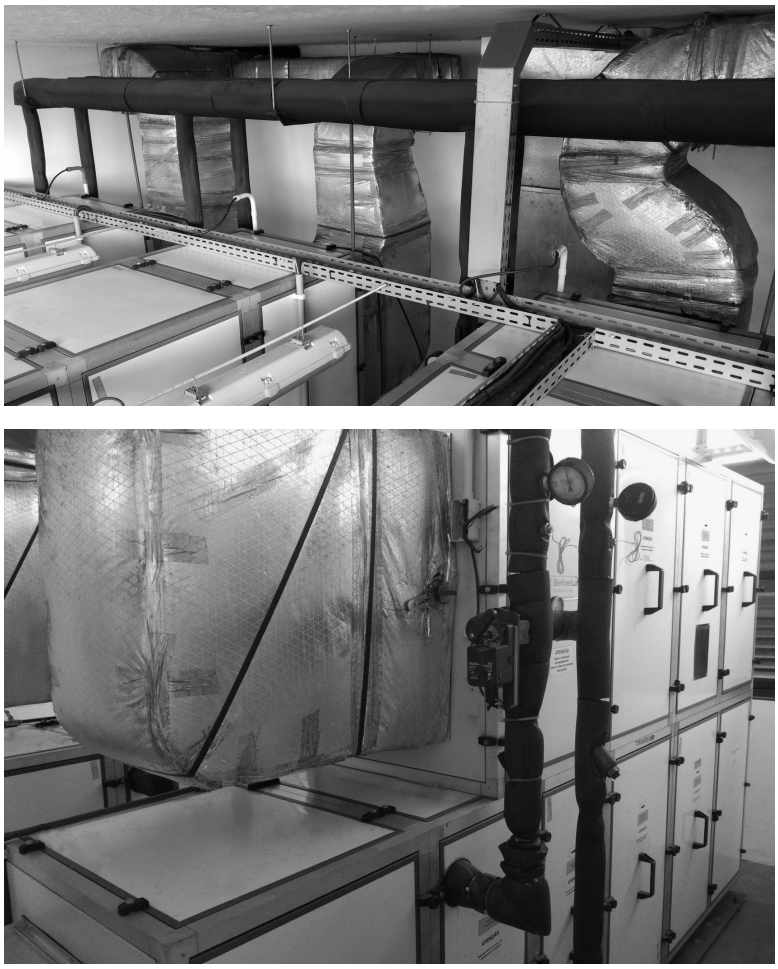


Figure 3.2: Photos of the AHU room depicting the air ducts (top), the supply and return water pipes (top and bottom), and the three-way valve servomotor (bottom).

	Symbols	Description
Inputs	θ_1, θ_2, θ_3	Valve position of AHU 1, AHU 2 and AHU 3
Outputs	T_1, T_2, T_3	Temperatures within zone 1, zone 2 and zone 3
Measured disturbances	T_{sup}, T_{out}, R_{sol}	Chiller supply water temperature, outdoor temperature, solar radiation
Unmeasured disturbances	–	Internal heat gains (e.g. occupants), opening and closing of the doors

Table 3.1: The main physical quantities influencing the HVAC plant and the temperature dynamics inside the rooms.

Creating a reliable model for the system dynamics is crucial to attain high-performance with Model Predictive Control. In the current setting, this task is not trivial as certain disturbances and control variables enter the dynamics non-linearly. For instance, the globe water valves in the AHUs are not linear actuators in that the flow is not directly proportional to the angular position. For this reason, we decided to adopt a flexible class of statistical models to tackle the modeling problem

	T_1	T_2	T_3	θ_1	θ_2	θ_3	T_{sup}	T_{out}
GP1 (OR 1)	2	1	–	1	–	–	1	1
GP2 (WR)	1	2	1	–	1	–	1	1
GP3 (OR 2)	–	1	2	–	–	1	1	1

Table 3.2: The signals that composed the feature vector of each Gaussian process, and their respective delay parameters l. The symbol "$-$" indicates what signals were neglected.

while requiring as little expert knowledge as possible. Their main advantage being that this class not only predicts the expected system behavior, but also quantifies the uncertainty associated with its predictions, hence allowing for a more robust, risk-aware operation.

3.2.2 Crafting Gaussian process dynamical models

Gaussian processes lie in the class of non-parametric[1], non-linear, Bayesian models. For a thorough presentation of the topic, we refer the reader to Williams and Rasmussen (2006); **?**. Their main appeal over other types of statistical modeling paradigms is the analytical tractability, whereby expressions used during training and prediction have closed forms, dispensing with the need of using sample-based approximation techniques (see for example **??**). Assume one wants to model a given phenomenon $f(x)$ through noisy observations of the form $y = f(x) + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$ is a zero-mean Gaussian noise of unknown variance. As per the usual Bayesian approach, we define a *prior* model $f(x) \sim \mathcal{N}(m(x), k(x, x))$ and, after gathering some experimental data $D = \{x_n, y_n\}_{n=1}^N$, it is possible to update our beliefs and form a *posterior* model whose point-wise mean and variance are respectively

$$\mu(x) = m(x) + k_X(x)^\top (K + \sigma_\varepsilon^2 I)^{-1}(y - m_X) \tag{3.1a}$$

$$\text{var}(x) = k(x, x) - k_X(x)^\top (K + \sigma_\varepsilon^2 I)^{-1} k_X(x) \tag{3.1b}$$

where X and y denote the collection of all data features and labels in the dataset D, and $k(x, x')$ is the kernel function. $k_X(x)$ and K are respectively a column vector and a square matrix of kernel evaluations at X and x. Lastly, I represents the identity matrix.

Fully specifying a GP regression model amounts to i) picking a suitable mean function and a suitable non-linearity, i.e., a kernel function $k(x, x)$; and ii) optimizing

[1]Despite the name, non-parametric models still have internal *hyperparameters* to be tuned. Although subtle, this difference has significant practical implications Williams and Rasmussen (2006).

all model hyperparameters. In our case study, a linear mean $m(x) = Ax + b$ was employed. Among the many kernel maps available in the literature **?**, we chose the anisotropic squared-exponential function, a very popular alternative due to its smoothness and expressive power **??**. This kernel has the form

$$k_{\mathrm{SE}}(x, x') = \sigma^2 \exp\left(-\frac{1}{2}\sum_{i=1}^{d}\left(\frac{x_i - x_i'}{\ell_i}\right)^2\right) \tag{3.2}$$

where x_i is the ith component of the feature vector x. In (3.2), σ is the so called vertical scale hyperparameter and ℓ_i are the horizontal scale (a.k.a. lengthscale) hyperparameters. As for optimizing the constants A, b, σ, σ_ε and ℓ_i, we made use of the log-marginal likelihood objective (Williams and Rasmussen, 2006, Chapter 5) and a gradient-based procedure. This is a widely adopted criterion, known to contain a regularization term that combats overfitting.

In order to design dynamic models for the room temperatures, we used an auto-regressive approach, meaning that future predictions of a signal depend on the current and past values of itself as well as on current and past values of other relevant quantities. Since there were three temperatures to predict, three distinct models were trained and, so as to avoid augmenting the Gaussian process with unnecessary features, we made use of domain knowledge. Rooms that are not neighbors do not directly influence each other's temperatures; similarly, changing the valve position of AHU 1 has no effect on any temperature besides T_1. Initially, all exogenous signals T_{sup}, T_{out} and R_{sol} had been included into all models to boost their prediction capabilities. Nevertheless, we later realized that R_{sol} was a significant covariate only for T_3 as discussed in Section 3.2.3. Field tests unveiled a high correlation between the MPC computation times over the day and the solar radiation curve. Since having predictable rather than fluctuating solve times was a project requirement, we decided not to employ R_{sol} as a feature in any GP. The definitive set of employed features is reported in Table 3.2, where the delay parameter l indicates the number of current plus past values used from that particular physical signal. By using the mean functions (3.1a) to evolve the temperature dynamics, we arrive at the final models

$$T_{1,t+1} = \mu_1(T_{1,t}, T_{1,t-1}, T_{2,t}, \theta_{1,t}, T_{\mathrm{sup},t}, T_{\mathrm{out},t}) \tag{3.3a}$$
$$T_{2,t+1} = \mu_2(T_{1,t}, T_{2,t}, T_{2,t-1}, T_{3,t}, \theta_{2,t}, T_{\mathrm{sup},t}, T_{\mathrm{out},t}) \tag{3.3b}$$
$$T_{3,t+1} = \mu_3(T_{2,t}, T_{3,t}, T_{3,t-1}, \theta_{3,t}, T_{\mathrm{sup},t}, T_{\mathrm{out},t}) \tag{3.3c}$$

Concerning the variances (3.1b), we opted for not propagating them forward in time since no closed-form expression exists to accomplish this. Instead, the expression (3.1b) was evaluated in a point-wise fashion to measure uncertainty.

The interested reader is referred to **??** for insightful discussions on the matter.

3.2.3 Model training and testing

Data collection was carried out from August to November 2021. The final batch consisted of 22,455 points sampled at $T_{\text{samp}} = 2$ mins and comprised closed-loop operation with PI and rule-based controllers (RBCs), as well as a variety of open-loop excitation signals such as ramps and uniformly random inputs. After examining the obtained curves, we concluded that a control period of 10 mins would be a good compromise between operating the HVAC system effectively and not oversampling the temperatures – given that our model complexity grows with the size of the dataset, the latter aspect was rather important. The data batch was then downsampled by a factor of 5 times, resulting in 4491 points (748 hours). After that, a meticulous post-processing step was necessary to ensure that unreliable periods were discarded, outliers were detected and filtered, and imputation was performed to fill in certain missing entries. The feature vectors and labels were then created for each of the GPs described in Table 3.2, hence defining their training sets. In our particular case, all variables had similar ranges, thus normalization was not necessary. A critical step was to drop feature vectors that were too close to each other (in a Euclidean norm sense), which not only removed redundant information from the batch, but also improved the numerical stability associated with the kernel matrix Williams and Rasmussen (2006). Finally, the GPs modeling rooms 1, 2 and 3 had respectively 235, 245 and 314 points, which correspond to approximately 38, 51 and 47 hours worth of data. Since these sets did not come from a single experiment, but are an informative subset of the 748 hours initially available to us, they provided enough prediction capabilities to our non-parametric models.

All GPs were defined and trained with the aid of the `GPflow2` **?** and `SciPy` **?** packages. No priors were placed on the hyperparameters and we employed the marginal likelihood criterion along with the limited-memory BFGS optimization algorithm to tune them. Training the models with the aforementioned number of points took consistently less than 5 seconds each on a 2.4 GHz, core i9-9980HK machine. The obtained training and test results can be seen in Figure 3.3. We highlight that the plots show multi-step ahead predictions over a horizon of 2 hours, that is, 12 time steps, correcting for the temperature mismatch only at the orange points. Assessing the prediction quality of the models in this way was necessary as they were to be used within an MPC formulation. It is worth noting that, even though the left plots are labeled as “training results”, the models incorporated only a small fraction of those features due to our dropping of nearby

data-points. The central and right-side plots show the predictions over a period of 50 and 40 hours, but in completely new scenarios, never presented to the model during training.

Inspecting Figure 3.3, we see that the mean predictions mostly follow the underlying ground-truth signal during training. The disparity among the rooms is in their uncertainty bands: whereas model 1 and 2 presented moderate levels of spread, model 3 showed a fairly large one. We believe this uncertainty to stem from room 3 being the most exposed one in terms of direct solar radiation, and from R_{sol} not being a feature of its GP. We remind the reader that R_{sol} was disregarded to accelerate the real-time computations and ensure that the optimization problem was solved within the time allocated to it. By taking this larger uncertainty into account, we were able to avoid violating constraints when closing the loop with the MPC controller. The outcome of the test phase was qualitatively similar to the training results, aside from some additional performance degradation close to the high temperature peaks. Overall, we deemed the results reasonable given the challenging two-hour horizon of the prediction task.

3.2.4 Learning the chiller energy consumption

We tackled the problem of operating the system while minimizing its electrical demand with the two-step approach described next. The first goal was to reconstruct the chiller refrigeration curve from historical data, more specifically, from the volumetric air-flow rates along with the outdoor temperature and the supplied air temperatures. Based on these quantities, the thermal power delivered by the chiller was inferred. Next, we used as features the outdoor temperature T_{out} and the sum of the valve positions $\Theta = \theta_1 + \theta_2 + \theta_3$, the latter correlating with the water flow through the AHU coils (see Section 3.2.1). A representative dataset was gathered over a period of 203 hours, which encompassed both random open-loop excitation and closed-loop operation. During this period, the AHU valves ranged from being completely open to being fully closed, and the ambient temperature varied from 15 to 40 degrees Celsius. The data distribution can be seen in Figure 3.4. We remark that the portion of the domain where the outdoor temperature is high and the total valve openings are low is not populated with samples due to operational constraints of the system, a common issue in HVAC control **?**. The last step was to augment the batch with a grid of T_{out} values paired with $\Theta = 0\,\text{degs}$ and $0\,\text{kW}$ labels to represent the zero water-flow regime.

Polynomial ridge regression was performed to fit the data described above. After experimenting with different model orders, we found that a cubic model provided

a good balance between describing the observed points and not overfitting them. The results are presented in Figure 3.4, where the obtained model analytical expression is

$$\begin{aligned} \mathrm{Q}(T_{\text{out}}, \Theta) = &-3.15\, T_{\text{out}} - 3.03\, \mathsf{e}^{-2}\, \Theta + 1.73\, \mathsf{e}^{-1}\, T_{\text{out}}^2 \\ &- 1.56\, \mathsf{e}^{-3} T_{\text{out}} \Theta + 3.09\, \mathsf{e}^{-4} \Theta^2 - 2.75\, \mathsf{e}^{-3} T_{\text{out}}^3 \\ &+ 4.90\, \mathsf{e}^{-4} T_{\text{out}}^2 \Theta - 6.86\, \mathsf{e}^{-5} T_{\text{out}} \Theta^2 + 2.56\, \mathsf{e}^{-6} \Theta^3 + 20.22 \end{aligned} \tag{3.4}$$

with e^n being a shorthand for $\times 10^n$. The model attained a mean absolute error of 2.88 and a mean squared error of 12.97 kW. As a second step, we fit a concave coefficient of performance (COP) curve, which is typical for variable-speed compressor chillers and dictates how efficient they are in converting electrical to thermal energy. The final utilized COP curve was

$$\begin{aligned} \mathrm{COP}(Q) = 3.30\, \mathsf{e}^{-7} Q^4 - 2.69\, \mathsf{e}^{-5} Q^3 - 2.67\, \mathsf{e}^{-3} Q^2 \\ +2.34\, \mathsf{e}^{-1} Q^1 - 4.45\, \mathsf{e}^{-4} \end{aligned} \tag{3.5}$$

which, given the thermal range displayed in Figure 3.4, implies in a performance coefficient varying approximately between 1.5 and 4.5.

With the thermal model (3.4) and the COP curve (3.5) at hand, the electrical power could be calculated according to $E = Q(T_{\text{out}}, \Theta)/\mathrm{COP}(Q(T_{\text{out}}, \Theta))$, measured in kW. Several slices of the thermal power surface, and their associated electrical power counterparts are presented in Figure 3.4. The plots illustrate how the curves change depending on the outdoor temperature, and how strongly the electrical power profile is affected by this external factor. In particular, one notices that when the outside temperature is high, it is more economic to open the valves and increase the chilled water flow rather than keeping them partially closed. Clearly though, the real-time optimal position for them will depend on the system dynamics, the desired temperature envelope and the external disturbances.

3.3 MPC formulation and numerical computations

Given the learned GP models μ_i, $i = 1, 2, 3$ described in (3.3), a given maximum temperature $T_{\max}$, and our reconstructed electrical power surface, we formulate the following optimization problem to control the valves θ_i while reducing the

chiller energy consumption E_t

$$\min \quad \sum_{t=0}^{N-1} (E_t + \rho\Delta_t) + \rho_N \Delta_N \tag{3.6a}$$

$$\text{s.t.} \quad T_{t+1} = \mu(T_t, \theta_t, T_{\text{sup}}, T_{\text{out}}) \tag{3.6b}$$

$$T_t + \beta \, \text{var}^{1/2}(T_t, \theta_t, T_{\text{sup}}, T_{\text{out}}) \leq T_{\max} + \delta_t \tag{3.6c}$$

$$E_t = Q(T_{\text{out}}, \Theta_t) / \, \text{COP}(Q(T_{\text{out}}, \Theta_t)) \tag{3.6d}$$

$$\theta_{\min} \leq \theta_t \leq \theta_{\max} \tag{3.6e}$$

$$\delta_t \geq 0 \tag{3.6f}$$

where $\Theta_t = \sum_{i=1}^{3} \theta_{i,t}$ is the sum of all valve positions. The variables δ_t in (3.6c) are positive slacks introduced to avoid infeasibility. If needed, these can relax the temperature constraint so that the solver can return a viable control plan. Of course, their use is heavily penalized in the objective, where $\Delta_t = \sum_{i=1}^{3} \delta_{i,t}^2$ and ρ, ρ_N are large constants, which in our case were respectively set to 100 and 200. The temperature constraint (3.6c) also accounts for prediction uncertainty as it includes the standard deviation $\text{var}^{1/2}$. Its use confers on the formulation a risk-aware quality and robustifies the closed-loop operation. The degree of conservativeness is controlled by the constant β, chosen to be 2 as in Figure 3.3. The prediction horizon was set to $N = 12$ steps, which translates to 2 hours. As suggested by our notation, T_{out} and T_{sup} were kept constant throughout all prediction steps–but updated from one sampling period to the next. Finally, our maximum temperature value was $T_{\max} = 21$ degrees Celsius.

The optimization problem (3.6) was written in `Python` with the aid of CasADi **?**, an automatic-differentiation package that provides gradient information for numerical solvers–in our case, the interior-point method IPOPT. As is customary in predictive control, (3.6) was recursively solved on-line with the most recently available system information, with only the first optimal control action being transmitted to the valves. We underline that the main source of complexity in (3.6) is the presence of the constraints (3.6b) and (3.6c), which are highly non-linear due the GP mean and variance. Since convexity is absent, multiple local optima might exist, a fact that was indeed verified in practice. By intelligently providing solvers with high-quality initial guesses, this problem can be mostly overcome. Our particular case study relied on initializing the numerical solver with control, temperature, slack and energy trajectories obtained with a virtual PI controller. The intuition was to allow the MPC loop to build on such an initial guess and further optimize operation. For a detailed study on solve times and how the number of GP data-points impacted them, see Appendix A.

3.4 Experimental results

The previously described Gaussian process-based MPC formulation was deployed on the local computer and used to operate the HVAC system during multiple days in the months of October and November 2021. We report in Figure 3.5 a four-day uninterrupted experiment carried out from November 10 to November 13 that is rather representative of the local internal and external conditions. The plots show the room temperatures and the "immediate" uncertainty associated with the GP predictions: $T_{\text{unc}} = \beta \text{var}^{1/2}$ as employed in the formulation (3.6c), and evaluated for the next time-step. Both outdoor signals, the temperature and the solar radiation, are also given. The reader is reminded that, although the latter contributes with additional heat gains, it is completely unknown to the controller as explained in Section 3.2.3. We highlight that the curves displayed in the figure were not filtered in any way; the sole manipulation performed with the data was the imputation of the missing temperature entries using linear interpolation. These points, however, accounted for only 43 out of the 1728 indoor temperature values gathered during the four-day experiment.

Consider first the day November 10 and note the relatively high internal room temperatures when the experiment started, which were the consequence of a harsh previous day. The MPC controller used some control authority to bring the temperatures below the 21-degree line and then partially closed the valves. After the morning shift started (7 am), even though θ_2 and θ_3 were fully open, T_2 and T_3 violated the constraints and were only brought below 21 degrees late that evening. High initial conditions along with a peak outdoor temperature of 35 degrees overloaded the cooling system, causing violations of the indoor temperature constraint in two rooms.

The two days that followed (November 12 and 13) were less warm and, as a result, the MPC controller successfully modulated the valves so as to guarantee constraint satisfaction. It is evident how θ_1, θ_2 and θ_3 assume lower values when T_{out} is low, and tend to saturate at their maximum during working hours, which matches our intuition.

Lastly, we focus on the data from November 13, where one can readily see a sudden peak in the indoor temperatures, being also present in T_{sup}. This was caused by a momentary halt in the water pumps responsible for the chilled water circuit–an event that could be regarded as a fault from a control system perspective. During this period, as there was no water circulation through the AHU cooling coils, there was also no refrigeration and the indoor spaces received warm air since the fans were kept on. As soon as the pumps were again activated, the

chiller immediately decreased the supply water temperature and the operation was normalized. During daytime, the indoor climate was kept within the desired limits despite the valves staying saturated at their low values, even at noon. The fact that almost no additional actuation was needed is due to that day being a Saturday, when no operations are scheduled and the three doors present in the environment are minimally opened and closed. This demonstrates how strong the internal heat gains and unmeasured disturbances normally are.

To assess the efficiency gains as well as the thermal performance of the deployed strategy, MPC was compared to alternative algorithms, all subject to exactly the same environmental conditions by means of simulations. We underline that this simulation model was calibrated on data that was *not* included in the GPs training set, thus putting to test the MPC prediction capabilities. The disturbance signals T_{out}, T_{sup} and R_{sol} from November 10 were employed, and the indoor temperatures of the three rooms were uniformly initialized at values ranging from 17 to 21 degrees Celsius. The outdoor temperature profile was processed to yield three different weather scenarios: hot weather, which was exactly the same T_{out} curve seen in Figure 3.5; warm weather, a -2°C shifted version of it, peaking at 33°C around noon; and mild weather, a -5°C shifted version of it, peaking at peaking at 30°C. Besides the MPC algorithm (3.6), the following were also tested:

- An MPC controller (herein referred to as REF) with perfect prediction capabilities, perfect disturbance information (T_{out} and T_{sup}) and a long prediction horizon of five hours.
- PI controllers featuring anti-windup schemes and feedforward components to enhance their performance.
- Rule-based ON/OFF controllers that set the valves respectively to $\theta_{\min}$ and $\theta_{\max}$ if the indoor temperatures were below or above the set-point.
- An average $(\theta_{\max} - \theta_{\min})/2$ controller (AVG) whose instantaneous values are selected by sampling the interval $\theta_{\min}$ to $\theta_{\max}$ using a uniform distribution.

In order to gauge the energy saving potential of the HVAC plant, we executed the REF strategy described above. This algorithm can reach significant efficiency gains while guaranteeing thermal comfort since it exploits a perfect internal model as well as perfect forecasts of the outdoor and supply water temperatures.

The obtained normalized energy consumption results and average room thermal comfort violations are shown in Figure 3.6. Since the non-linear chiller curves described in Section 3.2.4 were employed to measure the energy consumption, the

reader is reminded that there is a non-trivial relationship between the weather conditions and indoor temperatures, and the final consumed energy. This aspect is due to the nature of the chiller and not the use of any particular control technique. As a last note, one specific energy normalization factor was used for each weather scenario shown in Figure 3.6 to enhance clarity.

Glancing at the mild and warm weather plots, one notices how the REF and MPC data tended to be close together, and relatively far from the PI, ON/OFF and AVG clusters. Moreover, the REF and MPC points were also mostly to the left side and vertically below the other data given the same indoor temperature conditions–thus confirming their superior performance in terms of energy efficiency and indoor climate regulation. The ON/OFF and PI controllers yielded overall similar numerical results and, surprisingly, were outperformed by the AVG scheme under mild weather and starting indoor temperatures of 17 °C and 19 °C. AVG nevertheless performed poorly under warm weather and 21 °C, and hot weather in general. All in all, the predictive control strategies MPC and REF yielded the best results in the mild and warm weather cases, whereas the separation among them and the other techniques became less evident under hot weather, indicating a less important advantage over classical control.

By analyzing the horizontal scales and contrasting REF to PI, ON/OFF and AVG, one concludes that this particular HVAC plant could have its efficiency boosted by approximately 2.5%, 4% and 5% respectively in the mild, warm and hot weather scenarios. Notice how, as opposed to studies such as **?**, these numbers refer to the electrical energy associated with a chiller, and not to a cumulative thermal energy. Moreover, as the hospital was not subject to time-varying electricity prices, the contrast among control strategies was not as broad as for instance the one reported in **?**. The proposed MPC strategy (3.6) attained results close to the aforementioned maximum percentages. Quantitatively, MPC lead to a maximum energy efficiency improvement of 2.29%, 3.13% and 4.76% respectively in mild, warm and hot weather, when compared to the PI and ON/OFF counterparts.

3.5 Time complexity

In order to shed light on how the number of training points affects the solve times of the non-convex optimization problem (3.6), the following study was conducted. Five sets of GP models were trained on distinct datasets with cardinalities $N = 352$, 794 (precisely the one used in the experiments), 1280, 1910 and 2643. In all scenarios, the total number of points present in each of the GPs was approximately a third of the total number N, so that the models were balanced. We then

generated random initial conditions, uniformly sampled from sensible intervals: $16 \leq T_{1,2,3} \leq 23$, $9 \leq T_{\text{sup}} \leq 13$, and $15 \leq T_{\text{out}} \leq 35$. Finally, we solved the warm-started non-convex MPC (3.6) on a 2.4 GHz, i9 machine 50 times per scenario and recorded their run times.

The results are presented in Figure 3.7, where the vertical scale is logarithmic. The median values of the box plots rose from 4.19 to 18.69, 85.42, 121.12 and 255.68 seconds respectively from the smallest to the largest dataset. Although using $N = 1910$ points does not seem unreasonable at first, challenging initial conditions such as ones close to violating constraints can easily increase the problem solve time: the highest point obtained for the $N = 1910$ scenario was 429 seconds. Ideally, and specially when occupants experience thermal discomfort, the control action has to be computed in negligible time to be applied as soon as possible to the system. Keeping the solve times below a low percentile of the total sampling period is thus a common desideratum. In our application, we regarded $N = 794$ to be an adequate choice.

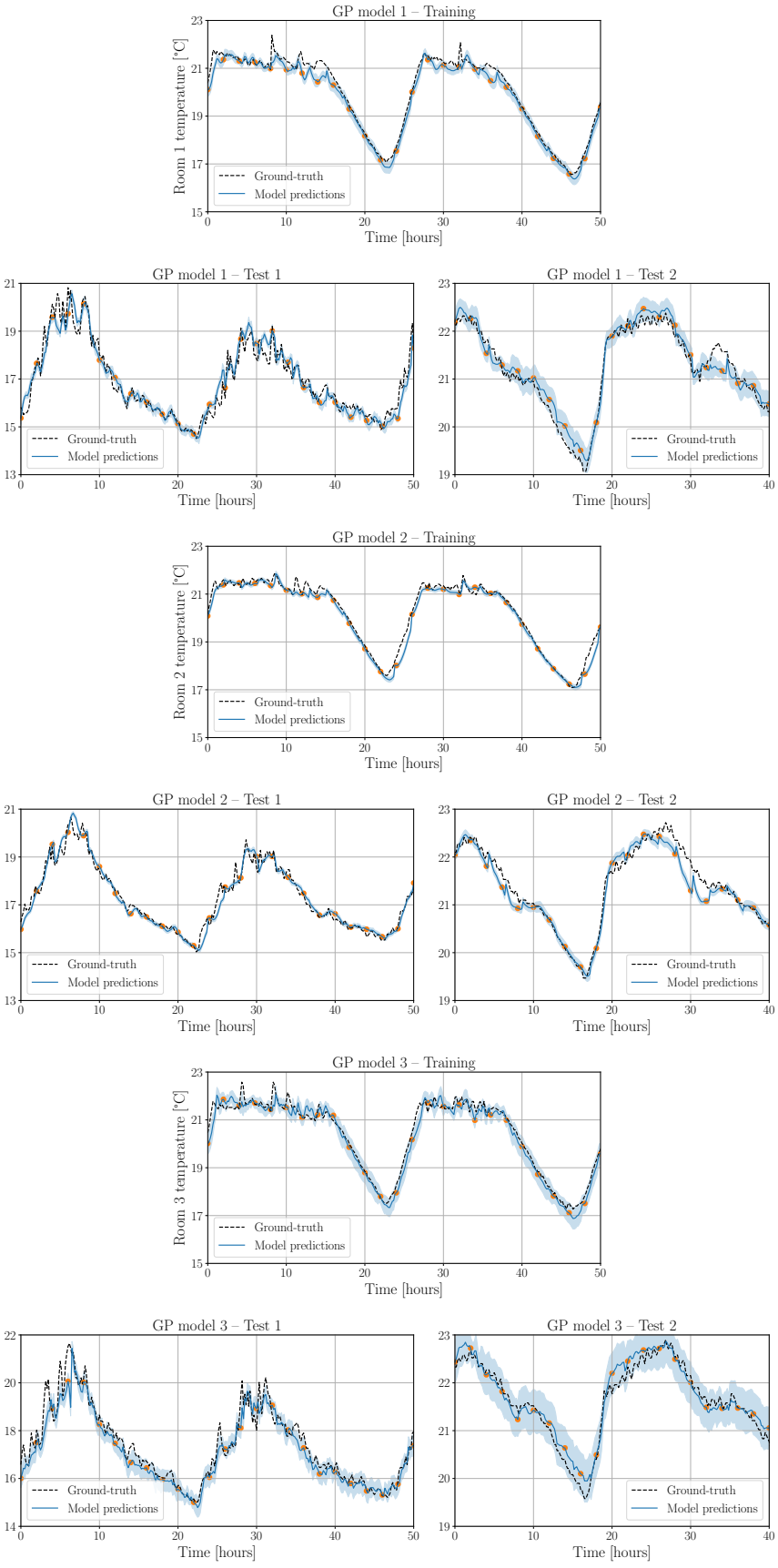


Figure 3.3: Training (left) and test results (center, right) of the Gaussian process models based on real experimental data. The mean predictions are shown in solid blue, whereas the uncertainty envelope of two standard deviations, in light blue. The sampling period is 10 minutes. The mean predictions were obtained by simulating the GP auto-regressive models forward in time for 2 hours and only feeding back new temperature information at the orange dots. The training plots only show a portion of the training set.

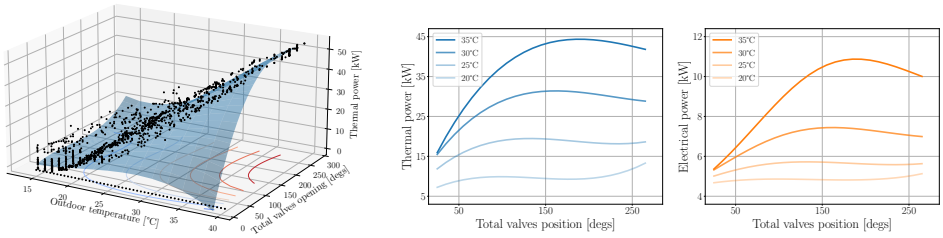


Figure 3.4: (Left) Reconstruction of the chiller refrigeration surface. Data were collected during a period of 203 hours, including both of open-loop excitation as well as closed-loop operation. (Center) Thermal power $Q(T_{\text{out}}, \Theta)$ and (right) electrical power $E(T_{\text{out}}, \Theta)$ curves of the chiller as a function of the valves openings Θ. The plots consider typical outdoor temperature values $T_{\text{out}} = 20, 25, 30$ and 35°C.

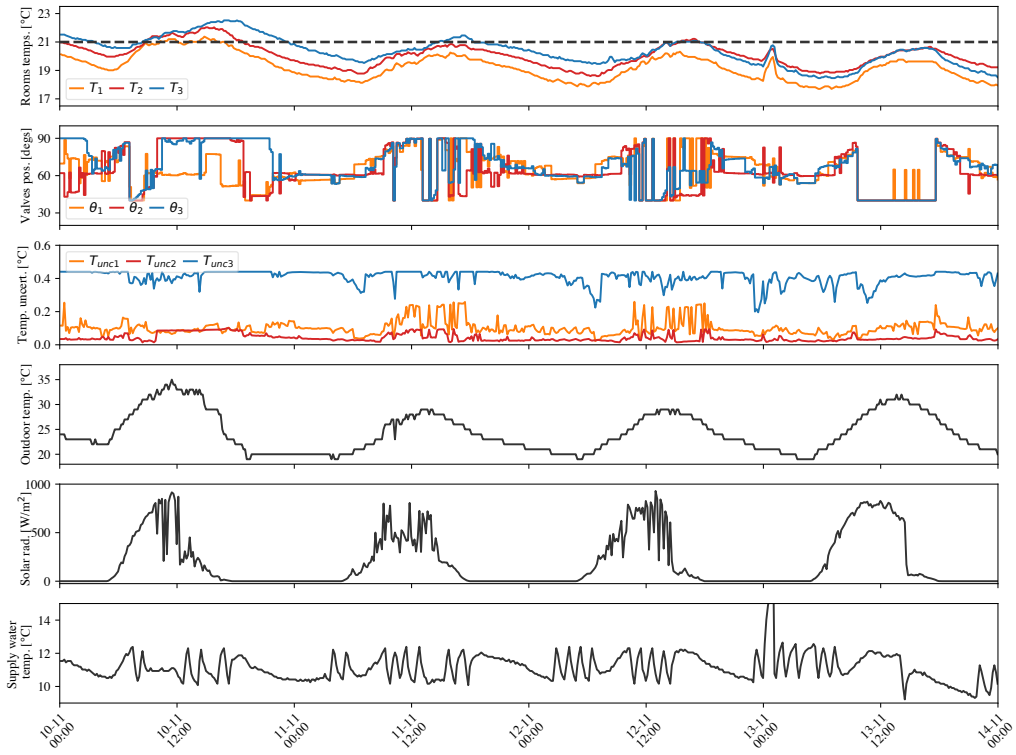


Figure 3.5: MPC experimental results over four days: indoor temperature, valve position and uncertainty estimate associated with each room (top three plots); outdoor temperature, solar radiation and AHUs supply water temperature (bottom three plots). The system was sampled and controlled with a periodicity of 10 minutes.

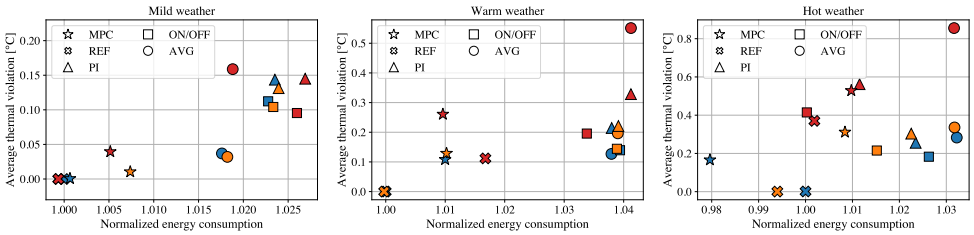


Figure 3.6: Simulation results of the normalized energy consumption and thermal performance (average temperature bound violation) of different control strategies. Three weather profiles were considered: mild, warm and hot. The indoor temperatures were initialized at different values according to the color scheme:...

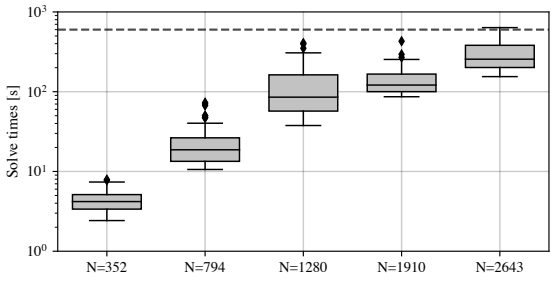


Figure 3.7: Box and whisker plots of the MPC solve times considering different dataset sizes. The dashed gray line marks our sampling period of 10 mins. Each boxplot is based on 50 time samples, obtained using randomized initial conditions.

4 Learning MPC controllers with pQP neural networks

In this chapter, we ...

4.1 pQP neural networks

4.2 Learning linear MPC controllers with pQP neural networks

Consider the following standard MPC formulation for linear dynamical systems

$$\mathbb{P}1 : \min_{X,U} \quad \sum_{k=0}^{H-1} \left(x_k' Q x_k + u_k' R u_k \right) + x_H' P x_H \tag{4.1a}$$

$$\text{s.t.} \quad \forall k = 0, \dots, H-1 \tag{4.1b}$$

$$x_{k+1} = A x_k + B u_k \tag{4.1c}$$

$$x_k \in \mathbb{X} \tag{4.1d}$$

$$u_k \in \mathbb{U} \tag{4.1e}$$

$$x_H \in \mathbb{X}_H \tag{4.1f}$$

$$x_0 = x(0) \tag{4.1g}$$

where $X := \{x_1, \dots, x_H\}$, $U := \{u_0, \dots, u_{H-1}\}$, $Q \succeq 0$, $P \succeq 0$, $R \succ 0$, and the constraints are all described by affine equalities and inequalities. Denote by $\pi : \mathcal{X} \to \mathcal{U}$ the optimal solution of (4.23) in its parametric form with respect to the initial conditions $x(0)$, where $\mathcal{X} \subseteq \mathbb{R}^n$ is the feasible state space of $\mathbb{P}1$ and $\mathcal{U} \subseteq \mathbb{R}^m$ is the control space. We assume a set of N samples can be acquired from

the original control law[1]

$$D = \{\mathrm{x}_i, \mathrm{u}_i\}_{i=1}^{N} \tag{4.2a}$$

$$\mathrm{u}_i = \pi(\mathrm{x}_i),\ i = 1, \ldots, N \tag{4.2b}$$

The process of acquiring the training dataset does not have to follow a particular distribution, nor do the samples have to be independent.

4.2.1 The proposed architecture

The key idea behind the proposed approximator is the use of a parametric quadratic program layer as part of the Neural Network, and optimizing over its parameters in order to fit the available dataset. This layer is implicitly described by the quadratic program

$$z^\star = \arg \min_{z \geq 0} \ ||Lz + y_1(x)||^2 + \epsilon||z||^2 \tag{4.3}$$

which is *always feasible* and *bounded from below*. The size of this mathematical program, i.e. the dimension of z, can be tuned to attain approximations with different complexity. Moreover, as notation suggests, the parameter y_1 depends on a previous affine layer that maps the system states into the z space, $y_1 := Fx + f$. Let $y_2 := z^\star$, then another affine layer maps the optimal solution to the input space $y_3 := Gy_2 + g$, and a projection onto the feasible input set produces the final control action $\hat{u} := \mathrm{Proj}_{\mathbb{U}}(y_2)$. This last step is necessary to guarantee feasibility of the control moves (see e.g. **?**). An illustration of the proposed architecture is presented in Fig. **??**, where the projection layer was particularized to a familiar element-wise saturation operation $\mathrm{sat}(\cdot)$, valid for box input constraints.

Let the chosen number of decision variables in the pQP be $z \in \mathbb{R}^{n_z}$. We choose $L \in \mathbb{R}^{n_z \times n_z}$ to be square, and therefore $F \in \mathbb{R}^{n_z \times n}$, $f \in \mathbb{R}^{n_z}$. Moreover, $G \in \mathbb{R}^{m \times n_z}$ and $g \in \mathbb{R}^m$. If $n_z \geq n$, the first layer lifts the input data into a higher dimensional space before it is passed through the optimization layer. A last affine function then projects it onto the control space. These facts will be later employed to analyze the representative power of the network.

The parameters to be trained are therefore F, f, L, G and g. This process can be carried out via a stochastic gradient descent algorithm applied to an appropriate loss function. Differentiability of all layers is trivial with the exception of the pQP one (**??**). Regarding the latter, note that the objective in (4.3) can be rewritten

[1] The dataset can also be directly obtained from the implicit controller; depending on the size of the problem at hand, computing the explicit solution might be intractable.

as

$$V(z) := z'(\epsilon I + L'L)z + (2L'y_1(x))'z + y_1(x)'y_1(x) \tag{4.4}$$

whose Lagrangian is simply

$$\mathcal{L}(z, \lambda) = V(z) - \lambda' z \tag{4.5}$$

The Karush-Kuhn-Tucker (KKT) conditions for primal and dual feasibility, complementary slackness, and stationarity then read

$$z^\star \geq 0 \tag{4.6a}$$
$$\lambda^\star \geq 0 \tag{4.6b}$$
$$\lambda_i^\star z_i^\star = 0, \ \forall i = 1, \ldots, n_z \tag{4.6c}$$
$$2(\epsilon I + L'L)z^\star + (2L'y_1(x)) - \lambda^\star = 0 \tag{4.6d}$$

where λ_i and z_i denote the components of the Lagrange multipliers and decision variables vectors. The following proposition presents the differentiability properties of the pQP layer, and holds since (4.3) is a particular instance of the `OptNet` layer (**?**) with strictly convex objective function.

Proposition 15. Let $\theta := (L, y_1)$. The parametric solution $z^\star(\theta)$ of (4.3) is subdifferentiable everywhere in its domain, i.e., $\partial z^\star(\theta) \neq \{\,\}$, and $\partial z^\star(\theta)$ has a unique element (the jacobian) everywhere but in a set of measure zero.

As shown in **?**, the relevant gradients with respect to the parameters to be trained can be obtained from the KKT set of equations (4.6). Hence, backward passes are possible and backpropagation can be performed to optimize all of the NN parameters.

4.2.2 Properties of the approximator

The authors of **?** showed that any continuous PWA function can be obtained as the solution of a particular parametric linear program (pLP) transformed by a linear map. Even though this view could be adopted herein, we instead prove a different result that is enough in the context of linear eMPC.

Theorem 3. (The proposed NN architecture can learn any linear quadratic MPC controller) Let $\hat{\pi} : \mathcal{X} \to \mathcal{U}$ be the map defined by the composition of all four layers, i.e., $\hat{\pi}(x) := y_4 \circ y_3 \circ y_2 \circ y_1(x)$. Set $\epsilon = 0$, then $\exists F$, f, L, G and g with appropriate dimensions such that $\forall x \in \mathcal{X}, \ \hat{\pi}(x) = \pi(x)$.

Proof: Start by condensing the MPC problem $\mathbb{P}1$, i.e., using the equality constraints

to eliminate all state decision variables except for the initial state $x(0)$. This leads to the following parametric problem

$$\mathbb{P}2 : \min_{U} \quad U' \Lambda U + x(0)' \Gamma U \tag{4.7a}$$

$$\text{s.t.} \quad \Phi U \leq \Omega x(0) + \omega \tag{4.7b}$$

The step by step procedure can be found in **?**. We have that $\Lambda \succ 0$. Problems $\mathbb{P}2$ and $\mathbb{P}1$ are then equivalent in the sense that the solution $U^\star$ of $\mathbb{P}2$ and $\{X^\star, U^\star\}$ of $\mathbb{P}1$ share the same $U^\star$ component. Next, calculate the dual problem of $\mathbb{P}2$, which is

$$\mathbb{D}2 : \min_{\lambda \geq 0} \ \frac{1}{4}\left[\lambda' \Phi \Lambda^{-1} \Phi' \lambda + (4x(0)'\Omega' + 2x(0)'\Gamma\Lambda^{-1}\Phi' + 4\omega')\lambda + x(0)'\Gamma\Lambda^{-1}\Gamma' x(0)\right] \tag{4.8}$$

It is possible to recover the primal optimal solution $U^\star$ from the dual optimal solution $\lambda^\star$ through the stationarity optimality condition of $\mathbb{P}2$

$$U^\star = -0.5\,\Lambda^{-1}\Phi'\lambda^\star - 0.5\,\Lambda^{-1}\Gamma' x(0) \tag{4.9}$$

The above equation is learned by the second linear layer in Figure **??**. Nevertheless, from (4.9) we see that it requires the value of $x(0)$, which is the NN input. It is shown next that with a pQP layer of appropriate size and parameters, it is possible not only to learn (4.8), but also let the value of $x(0)$ 'pass through' the NN and arrive to the second linear layer as needed to retrieve the primal optimal solution.

Let the auxiliary variable $\tilde{L}$ and function $\tilde{y}_1(x) := \tilde{F}x + \tilde{f}$ be the solution to (compare (4.4) and (4.8))

$$\tilde{L}'\tilde{L} + \epsilon I = 0.25\,\Phi\Lambda^{-1}\Phi' \tag{4.10a}$$

$$2\tilde{L}'\tilde{y}_1(x) = \Omega\, x(0) + 0.5\,\Phi\Lambda^{-1}x(0) + \omega \tag{4.10b}$$

which leads to $\epsilon = 0$, $\tilde{L} = 0.5\,(\Phi\tilde{\Lambda})'$, where $\tilde{\Lambda}$ is the unique square root of Λ^{-1}, guaranteed to exist as $\Lambda^{-1} \succ 0$. Then, $\tilde{y}_1(x) = (\Phi\tilde{\Lambda})^{-1}(\Omega\, x(0) + 0.5\,\Phi\Lambda^{-1}x(0) + \omega) \implies \tilde{F} = (\Phi\tilde{\Lambda})^{-1}(\Omega + 0.5\,\Phi\Lambda^{-1}),\ \tilde{f} = (\Phi\tilde{\Lambda})^{-1}\omega$.

Set the first layer weights to $F = [-I\ I\ \tilde{F}]'$ and $f = [\mathbf{0}\ \mathbf{0}\ \tilde{f}]'$ so that $y_1(x) = [-x;\ x;\ \tilde{F}x + \tilde{f}]$. Set the weights of the pQP layer (4.3) to $\epsilon = 0$ and $L = [I\ \mathbf{0}\ \mathbf{0};\ \mathbf{0}\ I\ \mathbf{0};\ \mathbf{0}\ \mathbf{0}\ \tilde{L}]$. If we partition the decision variable as $z = [\tilde{z}\ x^p\ x^n]'$, this results in

$$\min_{\tilde{z}, x^p, x^n \geq 0} \ ||x^p - x(0)||^2 + ||x^n + x(0)||^2 + ||\tilde{L}\tilde{z} + \tilde{y}_1(x)||^2 \tag{4.11}$$

which is a separable objective in $\tilde{z}$, $\tilde{x}^p$ and $\tilde{x}^n$. Due to the choice of $\tilde{L}$ and $\tilde{y}_1(x)$ in (4.10), the last term of the pQP matches the dual $\mathbb{D}2$ with the exception of its constant term – not relevant for determining the optimal solution. Therefore, we have that $\tilde{z}^\star$ in (4.11) matches $\lambda^\star$ in $\mathbb{D}2$. Regarding $x^{p\star}$, the n optimizer components will satisfy $\forall i = 1, \ldots, n$, $x_i^{p\star} = x_i(0)$ if $x_i(0) \geq 0$, else $x_i^{p\star} = 0$. Similarly, $x_i^{n\star} = -x_i(0)$ if $x_i(0) \leq 0$, else $x_i^{n\star} = 0$. Therefore, $x^{p\star} - x^{n\star} = x(0)$, and the output of the pQP layer (4.11) has the dual optimizer $\lambda^\star$ and the initial condition $x(0)$ encoded in it.

Next set the weights of the second linear layer y_3 to $G = [-0.5\Lambda^{-1}\Phi' \;\; -0.5\Lambda^{-1}\Gamma' \;\; 0.5\Lambda^{-1}\Gamma']$ and $g = \mathbf{0}$. Therefore, $y_3 = G\,[\tilde{z}^\star \; x^{p\star} \; x^{n\star}]' = G\,[\lambda^\star \; x^{p\star} \; x^{n\star}]' = U^\star$, where equality (4.9) was used in the last step. Finally, note that the last layer $y_4 = \text{Proj}_{\mathbb{U}}(y_3)$ will simply evaluate to y_3 since y_3 is the optimal primal solution $U^\star$, which satisfies the constraints (4.7b) and necessarily belongs to $\mathbb{U}$. The theorem then follows from the fact that $x(0)$ in the above calculations can be taken to be any point x in $\mathcal{X}$. □

Exactly matching the original MPC controller would require L to have the same size of $\Phi\tilde{\Lambda}$ and $\epsilon = 0$ as shown. Nevertheless, we are interested precisely in reducing the complexity of the resulting controller through employing less parameters. In this process, choosing a regularizer $\epsilon > 0$ is beneficial since it ensures that the QP is bounded during the training phase for all possible parameters.

Comment: Stability certification

4.3 Simulation results

4.3.1 Analysis and controller design

Parallelism is a key concept to increase the efficiency and power levels of electronic converters. Still, this design choice has to be followed by proper current/voltage balancing techniques to ensure that no stage is subjected to a higher electrical stress when compared to the others.

A schematic representation of a multicell step-down converter is shown in Fig. **??**, and its parameters can be found in Table **??**. The topology features three arms that are connected to a coupled inductor, and an L-C output filter. All self inductances are assumed equal $L_1 = L_2 = L_3 = L_s$, and all mutual inductances have value L_m. The switches of each arm operate in a complementary fashion at a fixed frequency $f = 15$ kHz, and with variable but constrained duty cycle $0 \leq d_i \leq 0.9$, $i = 1, 2, 3$. Let the average voltage applied by the arms over one

switching period be denoted by $v_i := d_i V_{in}, i = 1, 2, 3$. In order to ease the analysis, we apply the Lunze transform Ψ to all variables, decomposing the phase voltages and currents into differential and common mode components

$$\begin{bmatrix} i_{dm1} & i_{dm2} & i_{cm} \end{bmatrix}' := \Psi \begin{bmatrix} i_1 & i_2 & i_3 \end{bmatrix}' \tag{4.12}$$

$$\begin{bmatrix} v_{dm1} & v_{dm2} & v_{cm} \end{bmatrix}' := \Psi \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix}' \tag{4.13}$$

where $\Psi = (1/3)\,[2\ -1\ -1;\ -1\ \ 2\ -1;\ 1\ \ 1\ \ 1]$.

The control input is defined as $u := [v_{dm1}\ v_{dm2}\ v_{cm}]'$ and the continuous-time state vector, by appending the output voltage to the transformed currents $x := [i_{dm1}\ i_{dm2}\ i_{cm}\ v_{out}]'$. By using Kircchoff's circuit laws, a linear model of the form $\dot{x} = A_{ct}x + B_{ct}u$ can be derived with

$$A_{ct} = \begin{bmatrix} \frac{-R}{L_s - L_m} & 0 & 0 & 0 \\ 0 & \frac{-R}{L_s - L_m} & 0 & 0 \\ 0 & 0 & \frac{-R}{L_s + 2L_m + 3L_f} & \frac{-1}{L_s + 2L_m + 3L_f} \\ 0 & 0 & \frac{3}{C_o} & \frac{-1}{R_o C_o} \end{bmatrix} \tag{4.14}$$

$$B_{ct} = \begin{bmatrix} \frac{1}{L_s - L_m} & 0 & 0 \\ 0 & \frac{1}{L_s - L_m} & 0 \\ 0 & 0 & \frac{1}{L_s + 2L_m + 3L_f} \\ 0 & 0 & 0 \end{bmatrix} \tag{4.15}$$

Finally, discretization at frequency f is carried out using the zero-order hold method, yielding $x_{k+1} = Ax_k + Bu_k$.

The control goal is to regulate the output voltage v_{out} to 300 V while maintaining the phase currents balanced at all times, which translates to driving the differential currents to zero. More specifically we have the following fixed reference $x_{eq} = [0\ 0\ 16\ 300]'$ with $u_{eq} = B^{\dagger}(I - A)x_{eq}$, where $B^{\dagger}$ is the pseudo-inverse of B. Moreover, the controller approximation procedure must not incur a steady-state error larger than 200 mA for i_{dm1} and i_{dm2}, and 5% for the common mode component i_{cm} and output voltage v_{out}. The chosen MPC cost function was

$$J = \sum_{k=0}^{H-1} (||x_k - x_{eq}||_Q^2 + ||u_k - u_{eq}||_R^2) \ + \ ||x_N - x_{eq}||_P^2 \tag{4.16}$$

where $Q = \mathrm{diag}(10,\ 10,\ 0.1,\ 0.1)$, $R = 0.1\,I$, P is the solution to the associated the discrete-time algebraic Riccati equation, and $H = 10$. For all time instants, box state constraints were imposed $\begin{bmatrix} -5 & -5 & -10 & -20 \end{bmatrix}' \leq x_k \leq \begin{bmatrix} 5 & 5 & 30 & 400 \end{bmatrix}'$ and polyhedron constraints on the controls $H_u\,u_k \leq h_u$ that simply mapped the duty cycle saturation to the Lunze domain. Due to the polytopic input constraints,

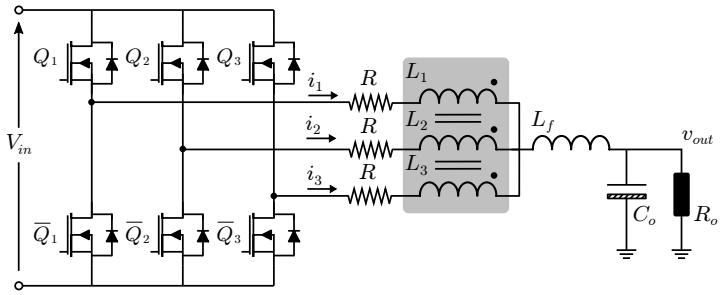


Figure 4.1: Photos of the AHU room depicting the air ducts (top), the supply and return water pipes (top and bottom), and the three-way valve servomotor (bottom).

the system cannot be decomposed into three decoupled parts as the structure of matrices A_{ct} and B_{ct} suggest. Furthermore, the standard terminal set constraint was imposed on x_H, defined as the invariant set associated to the unconstrained infinite-time problem formulation.

4.3.2 Learning the optimal controller

With the aid of the Multi-Parametric Toolbox (MPT) (**?**), the optimal eMPC solution $\boldsymbol{\pi}(x)$ was calculated and consisted of 2'337 critical regions. By counting the number of parameters necessary to describe each halfspace and control gain, the memory requirement of this PWA function was found to be 518 kB. In the previous calculation, a 4 byte representation was considered for both integers and floating point numbers.

Next, 5'000 samples were randomly acquired from the eMPC controller using a uniform distribution. The first and second components of the sampled control moves had considerably smaller amplitudes compared to the third due to the structure of the Lunze transform Ψ. The dataset labels $\{\mathrm{u}_i\}_{i=1}^{5000}$ had therefore to be scaled to ensure a similar learning of all control components. Moreover, instead of $\mathrm{Proj}_{\mathbb{U}}(y_3)$, the last NN layer was simplified to $y_4 = \Psi\,\mathrm{sat}(y_3)$ with saturation limits 0 and $0.9\,V_{in}$. This clearly guarantees control feasibility without the need of a second quadratic program. NN approximators were trained using `PyTorch` and the `OptNet` framework (**?**). A mean squared error loss function was minimized by

Table 4.1: DC-DC converter parameters

V_{in}	L_s	L_m	R	L_f	C_o	R_o
350 V	4 mH	-2 mH	10 mΩ	270 μH	20 μF	6.25 Ω

employing the Adam algorithm, mini-batch stochastic gradient descent with batch size 50, and 150 epochs. The size n_z of the pQP layer was varied from 1 to 7 and a total of 10 models were trained for each size; the lowest obtained losses are shown in Figure **??**. On average, training a model required 42 minutes on a 3.1 GHz Intel Core i7 machine without GPU acceleration, and 23 minutes with a single NVIDIA Tesla T4 graphics card. The learned parameters were then exported to MATLAB in order to calculate the PWA solution of the pQP layer.

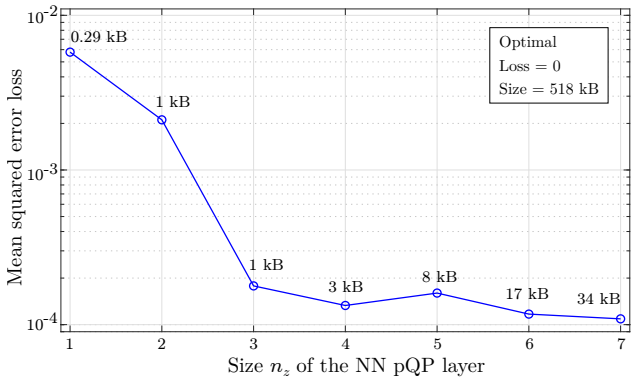


Figure 4.2: Neural network training loss as a function of the pQP layer size, and storage requirements associated with their PWA representations.

An increase in the n_z size clearly expands the representation capabilities of the neural network. This however does not always translate to a decrease in the final loss since the training process is affected by the weights initialization among other factors. Although not monotonic, we see a decrease of the overall training loss in Figure **??** as n_z grows. In order to validate the models, the start-up response of the converter was analyzed under all 7 different approximate controllers, and only the two largest ones ($n_z = 6$ and $n_z = 7$) met the target specifications given in Section 4.3.1. We refer to these two solutions as the *viable learned controllers*. Slices of their control surfaces are shown in Figure 4.3, and a phase portrait of the closed-loop system evolution over 50 steps starting from four initial conditions is depicted in Figure **??**. A summary of the two viable learned controller features is presented in Table **??**, including the number of polytopic regions, the storage requirements, the worst-case computation time[2] and the steady state (SS) error for i_{cm} and v_{out} – both clearly always equal. Even though four initial states were given, the systems always converged to the same points and, hence, only one SS error is reported. Plus, the storage numbers also take into account all

[2]Before proceeding to implementation, a further speed up would be possible through the methods listed in the Introduction.

the remaining layers parameters. Analyzing the obtained results we see that the approximations drastically reduced the storage requirements by 93.4% and 96.7% and sped up the average evaluation time by 83.7% and 88.4%, respectively for the $n_z = 7$ and $n_z = 6$ cases. The closed-loop trajectories with the proposed $\hat{\boldsymbol{\pi}}(x)$ remained reasonably close to the scenario with the optimal $\boldsymbol{\pi}(x)$, converging to nearby equilibrium points. In practice, steady-state errors could be completely removed by using the tools mentioned in Section **??**.

4.4 Experimental results

A schematic representation of the buck converter considered in this work is shown in Figure 4.4 and its parameters are found in Table **??**. V_{IN}, V_D, L, and C refer respectively to the input voltage, the diode forward drop, the inductance and the capacitance; whereas R_{ON}, R_L, R_C and R_O refer to the switch on-resistance, the inductor parasitic resistance, the capacitor parasitic resistance, and the output load.

We choose as state variables the inductor current and the output voltage

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} i_L \\ v_O \end{bmatrix} \tag{4.17}$$

The power switch is operated at a constant frequency f_{sw} and variable duty cycle, which is taken to be the control variable $u = \delta$. Following the classical time-averaging technique, Kircchoff's circuit laws are used to derive differential equations for both when the switch is open, and when it is closed. The expressions can be found in Appendix **??**. Averaging these equations with δ as a weight yields

Table 4.2: Parameters of the DC-DC converter

V_{IN}	V_{OUT}	V_D	L	C	R_{ON}	R_L	R_C	R_O	f_{sw}
15 V	5 V	0.1 V	10 mH	56 μF	5 mΩ	2 Ω	330 mΩ	100 Ω	20 kHz

$$\dot{x}_1 = -\frac{R_L}{L}x_1 - \frac{1}{L}x_2 + \frac{V_{IN}+V_D}{L}u - \frac{R_{ON}}{L}x_1 u - \frac{V_D}{L} \tag{4.18a}$$

$$\dot{x}_2 = -\frac{R_C R_O R_L C + R_O L}{(R_C+R_O)LC}x_1 - \frac{R_C R_O C + L}{(R_C+R_O)LC}x_2 + \frac{R_C R_O (V_{IN}+V_D)}{(R_C+R_O)L}u - \frac{R_C R_O R_{ON}}{(R_C+R_O)L}x_1 u - \frac{R_C R_O V_D}{(R_C+R_O)L} \tag{4.18b}$$

The expressions above are not linear since the inductor current and the duty cycle multiply each other. As the goal is to design a linear MPC controller, linearization is needed. We first fix the output voltage to the desired value x_{2eq} and solve for the current and duty cycle steady-state values

$$x_{1eq} = \frac{x_{2eq}}{R_O} \tag{4.19}$$

$$u_{eq} = \frac{R_O V_D + (R_L + R_O)x_{2eq}}{R_O(V_{IN}+V_D) - R_{ON}x_{2eq}} \tag{4.20}$$

Finally, (4.18a) and (4.18b) are expanded around (x_{1eq}, u_{eq}) and the linear terms are kept, leading to the familiar state-space equations $\dot{x} = A_{ct}x + B_{ct}u$ where

$$A_{ct} = \begin{bmatrix} -\frac{R_L + R_{ON}u_{eq}}{L} & -\frac{1}{L} \\ -\frac{R_C R_O (R_L C - R_{ON} C u_{eq}) + R_O L}{(R_C+R_O)LC} & -\frac{R_C R_O C + L}{(R_C+R_O)LC} \end{bmatrix} \tag{4.21}$$

$$B_{ct} = \begin{bmatrix} \frac{V_{IN}+V_D-R_{ON}x_{1eq}}{L} \\ \frac{R_C R_O (V_{IN}+V_D-R_{ON}x_{1eq})}{(R_C+R_O)L} \end{bmatrix} \tag{4.22}$$

As a last step, a discrete-time model $x_{t+1} = Ax_t + Bu_t$ is obtained by integrating the continuous-time dynamics using the standard zero-order hold method. The chosen discretization frequency was $f_{\text{samp}} = 10\,\text{kHz}$, which is also the predictive controller frequency.

The goal is to attain a fast start-up response with as little overshoot as possible and regulate the output voltage v_O to $v_{\text{eq}} = 5\,\text{V}$. Furthermore, an inductor current constraint of 200 mA and voltage constraint of 7 V must be respected at all times. The prediction horizon has to be long enough to yield a large feasible set **?** and we chose $N = 10$ steps. A standard quadratic objective was employed[3], penalizing the deviation of the states and control variable from the reference values

[3] In the MPC objective function, the squared weighted norms read as in $\|x_t - x_{\text{ref}}\|_Q^2 = (x_t - x_{\text{ref}})^\top Q\,(x_t - x_{\text{ref}})$.

$x_{\text{eq}} = \begin{bmatrix} 0.05 & 5 \end{bmatrix}^\top$, $u_{\text{eq}} = 0.3379$. The final optimal-control formulation was

$$\min_{X,U} \sum_{t=0}^{N-1} \left(\|x_t - x_{\text{eq}}\|_Q^2 + \|u_t - u_{\text{eq}}\|_R^2 \right) + \|x_N - x_{\text{eq}}\|_P^2 \tag{4.23a}$$

$$\text{s.t.} \quad \forall t = 0, \ldots, N-1$$

$$x_{t+1} = Ax_t + Bu_t \tag{4.23b}$$

$$\begin{bmatrix} i_L^{\min} \\ v_O^{\min} \end{bmatrix} \leq x_t \leq \begin{bmatrix} i_L^{\max} \\ v_O^{\max} \end{bmatrix} \tag{4.23c}$$

$$u^{\min} \leq u_t \leq u^{\max} \tag{4.23d}$$

$$x_N \in \mathcal{X}_N \tag{4.23e}$$

$$x_0 = x(0) \tag{4.23f}$$

with state and control constraints

$$x^{\min} = \begin{bmatrix} i_L^{\min} \\ v_O^{\min} \end{bmatrix} = \begin{bmatrix} 0\,\text{mA} \\ 0\,\text{V} \end{bmatrix} \tag{4.24}$$

$$x^{\max} = \begin{bmatrix} i_L^{\max} \\ v_O^{\max} \end{bmatrix} = \begin{bmatrix} 200\,\text{mA} \\ 7\,\text{V} \end{bmatrix} \tag{4.25}$$

$$u^{\min} = 0, \quad u^{\max} = 1 \tag{4.26}$$

The matrix weights were $Q = \text{diag}(90, 1)$, $R = 1$, and P was the solution of the associated discrete-time algebraic Ricatti equation. $\mathcal{X}_N$ was chosen to be the the system's maximal invariant set under the corresponding LQR policy. Both the terminal ingredients (P and $\mathcal{X}_N$) can be easily calculated with the aid of the Multi-Parametric Toolbox (MPT) **?** for MATLAB, and are employed to ensure recursive feasibility and closed-loop stability **?**.

As a final step, the MPC controller (4.23) was solved off-line using MPT, which yielded a piecewise-affine (PWA) function $\pi(x)$ that maps states directly to optimal control inputs. As well known in the area of explicit model predictive control **?**, this function partitions the space of feasible states $\mathcal{X}$ into regions described by sets of linear inequalities. Then, applying the predictive controller on-line boils down to implementing the look-up table of feedback gains

$$u = \pi(x) = \begin{cases} F_1 x + g_1, & \text{if } x \in \text{region } 1 \\ \ldots & \ldots \\ F_M x + g_M, & \text{if } x \in \text{region } M \end{cases} \tag{4.27}$$

As provided by MPT, the computed control policy $\pi(x)$ had $M = 70$ regions, a number too large to be embedded into the target MCU due to the large storage and computational demands (more details are given in Section 4.5). These implementation issues motivate the use of our PWA-NN complexity reduction scheme.

4.5 Learning a faithful still simpler representation of the controller

Explicit MPC controllers are the exact parametric solution of their optimization counterparts. The geometric landscape depicted by the PWA function $\pi(x)$ is composed of numerous linear pieces patched together. At times, neighboring regions share the same control law and, depending on their arrangement, they could be merged into an equivalent single one. Moreover, the overall surface usually presents two scales of complexity: a general shape and, inspecting it more closely, intricate small details. Based on these observations, it is reasonable to try to reproduce the rough shape of $\pi(x)$ without necessarily replicating its small wiggles.

4.5.1 The general architecture

The architecture of the piecewise-affine neural network used to learn $\pi(x)$ is shown in Figure **??**. It has two affine layers (L1 and L3), an optimization problem as the activation layer **?** (L2) and one projection layer (L4) that in this specific case is simply a saturation function. The latter is needed to ensure that the final control values produced by the NN are within the control bounds $0 \leq u \leq 1$. As discussed in **?**, the motivation behind the structure is that of learning the dual MPC problem: L1 maps the state x to the dual space, where L2 represents the dual optimization problem that is solved, L3 then maps the solution back to the primal space, and finally L4 guarantees it respect the control constraints.

As opposed to other approaches to learning MPC controllers with NN **??**, the one explored here can be translated to a *closed-form* piecewise-affine function. More specifically, the parametric quadratic program in layer L2 can be solved off-line after training (e.g. by using MPT), yielding a PWA map of the same form as (4.27). The complexity of such function in terms of the number of regions can be adjusted by choosing the size of matrix $H \in \mathbb{R}^{n_z \times n_z}$ inside L2. Fixing n_z also defines the sizes of all remaining trainable parameters highlighted in orange in Figure **??**. The result presented next assures the designer that this PWA-NN

structure is suitable for any possible predictive controller.

The optimization problem associated with training this NN—in fact, almost any NN architecture—is non-convex. As a consequence, even though there might exist a combination of parameters and weights capable of exactly representing the desired function, reaching them is not an easy task. Since local minima exist, the training process has to be performed multiple times with different initializations. Nevertheless, it is reasonably accepted in the machine learning community that these loss functions possess many high quality local minima, and pursuing a global optimum is irrelevant in this context (see for instance the influential work **?**).

Theorem 1 establishes that the size of the NN could be chosen to exactly replicate $\pi(x)$, but that would defeat its purpose since the goal is to learn a faithful but *simpler* version of the MPC controller. For this reason, we gradually increased the size n_z during the training process until a desirable approximation quality was attained. From a machine learning perspective, the problem could be interpreted as an approximation one, where the ground-truth is known.

The explicit control law $\pi(x)$ was sampled in order to collect a set of state-control pairs

$$\{(x_d, u_d) \,|\, d = 1, \ldots, D\} \tag{4.28}$$

where x_d can be regarded as features and u_d as labels. A total of $D = 5000$ points were gathered randomly using a uniform distribution over the set of feasible states. We highlight that the samples could have been acquired directly from (4.23) as well. Next, the data-points were used to train the internal parameters of the layers shown in Figure **??**.

A standard backpropagation approach can be used to iteratively update the NN parameters since, as shown in **?**, optimization layers of this type are differentiable (except on sets of measure zero, where subgradients can be used). The `PyTorch` and `OptNet` packages for Python were employed to code the NN and mini-batch stochastic gradient descent was used to train it. The batch size was chosen to be 50, and the whole dataset was presented to the algorithm a total of 150 times, i.e., 150 epochs. In order to achieve a balanced learning throughout the domain, the currents and voltages values that formed the input locations x_d were normalized to a range of $[0, 1]$. Furthermore, all trainable weights were initialized randomly. The code was run on a 3.1 GHz Intel Core i7 laptop with 16 GB 2133 MHz of memory. As previously explained, we gradually increased the size n_z of the PWA-NN. Training the network once took approximately 35 mins without any GPU acceleration. With $n_z = 3$, after only 5 initializations, the network presented a very low mean squared error training loss: 1.66×10^{-7}. As for the testing phase,

we calculated the true outputs $u = \pi(x)$ and the predicted values $\hat{u} = \hat{\pi}(x)$ on a grid of points; the latter were capable of closely reproducing the original controller as shown in the top plots of Figure **??**.

In order to assess the complexity of the learned controller, its L2 layer was converted into a PWA function using the MPT toolbox. As can be seen from lower plots in Figure **??**, the number of region was greatly reduced: from 70 in the original partition to 6 in the simplified one, a reduction of 91%. The total memory required to store the control law parameters was reduced from 9.25 kB to 528 B. The latter quantities were calculated by counting the total number of constants needed to describe all the inequalities that compose the polytopes and the remaining NN layers, and assuming that each of them occupies 1 *word* of space.

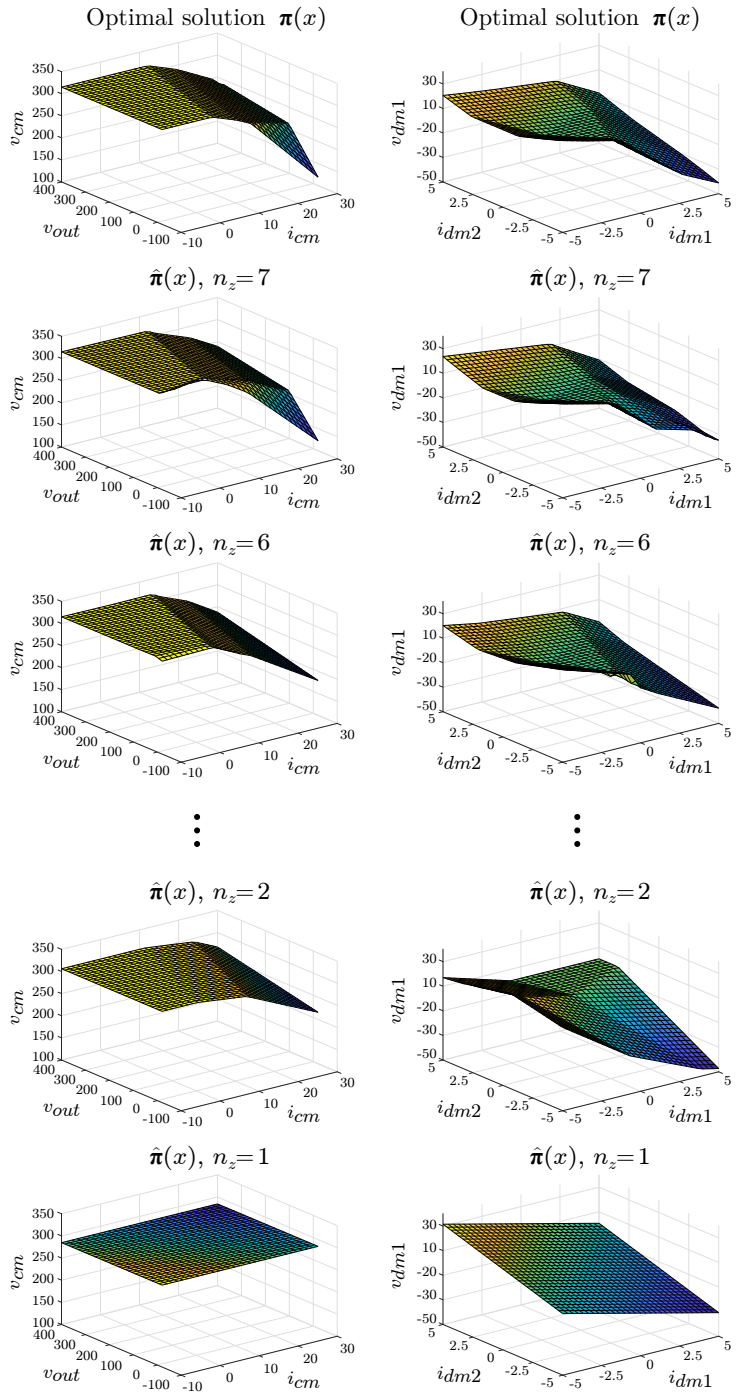


Figure 4.3: Slices of the optimal eMPC controller $\boldsymbol{\pi}(x)$ and several PWA NN approximations $\hat{\boldsymbol{\pi}}(x)$. The left plots are associated to v_{cm} and the right plots, to v_{dm1}.

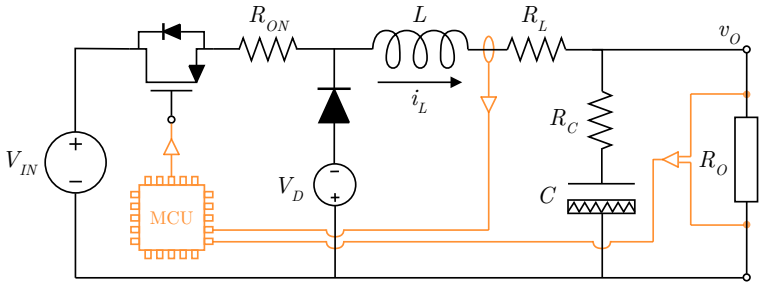


Figure 4.4: A circuit diagram of the buck converter including its parasitic resistances and the diode forward voltage drop. The feedback loop is closed by the MCU, which implements our proposed PWA-NN controller.

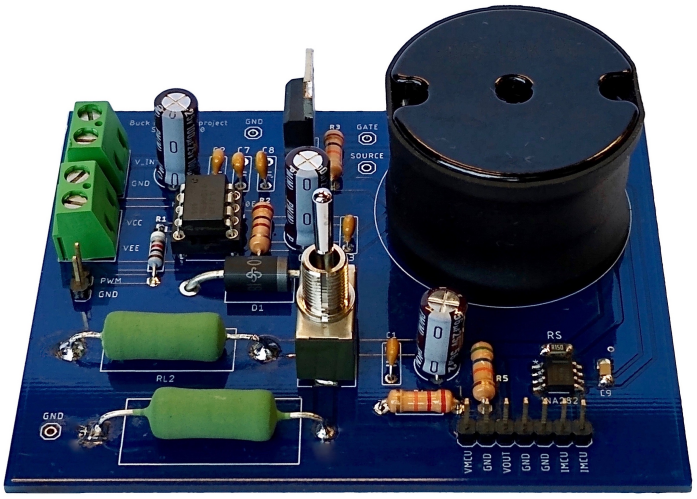


Figure 4.5: Photos of the AHU room depicting the air ducts (top), the supply and return water pipes (top and bottom), and the three-way valve servomotor (bottom).

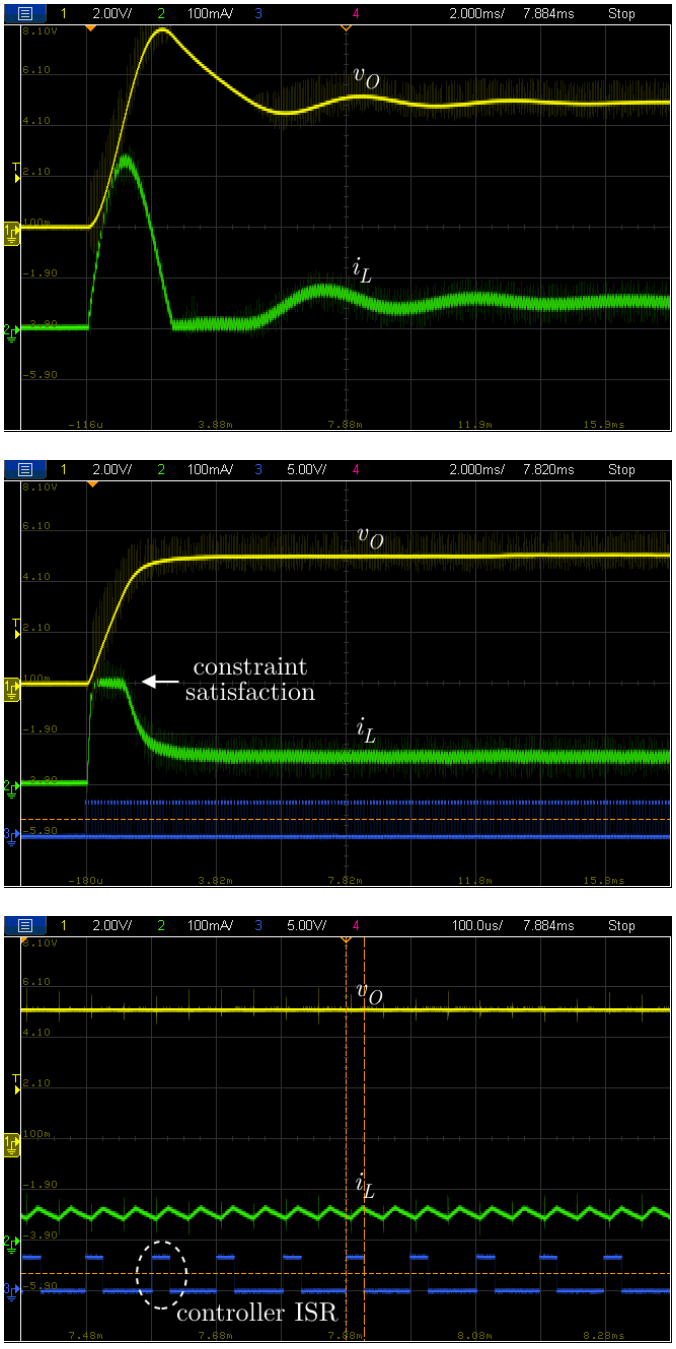


Figure 4.6: Open-loop start-up response (top), closed-loop start-up response (middle), and a close-up view of the closed-loop start-up response highlighting individual switching cycles and the controller interrupt service routine (ISR) execution time, approximately 27 μs.

A Elements of analysis and algebra

For a comprehensive presentation of the concepts, the reader os referred to Searcóid and Searcoid (2002); Pugh (2002).

All vector spaces herein are defined over the field of real numbers $\mathbb{R}$.

Definition 6. (Metric space) A metric space is a vector space $(V, +, \times)$ equipped with a map $d(\cdot, \cdot) : V \times V \to \mathbb{R}$ called a *metric* satisfying

$$(i)\, d(v, v) \geq 0 \tag{A.1}$$
$$(ii)\, d(v, w) = 0 \Leftrightarrow v = w \tag{A.2}$$
$$(iii)\, d(v, w) = d(w, v) \tag{A.3}$$
$$(iv)\, d(v, w) \leq d(v, z) + d(z, w) \tag{A.4}$$

for any $v, w, z \in V$.

For simplicity, we write (V, d) instead of $(V, +, \times, d)$.

Definition 7. (Convergent sequence) Given a metric space (X, d), a sequence $\{x_n\}_{n\in\mathbb{N}}$ in X is said to *converge* to an element $x \in X$ if

$$\forall \epsilon > 0 \, : \, \exists N \in \mathbb{N} \, : \, \forall n \geq N \, : \, d(x_n, x) < \epsilon \tag{A.5}$$

Convergent sequences are usually written $\lim_{n\to\infty} x_n = x$ or more simply $x_n \to x$. Moreover, sequences cannot converge to two or more points.

Definition 8. (Cauchy sequence) Given a metric space (X, d), a sequence $\{x_n\}_{n\in\mathbb{N}}$ in X is said to be *Cauchy* if

$$\forall \epsilon > 0 \, : \, \exists N \in \mathbb{N} \, : \, \forall n, m \geq N \, : \, d(x_n, x_m) < \epsilon \tag{A.6}$$

Cauchy sequences are a superset of convergent sequences.

Definition 9. (Complete space) A metric space is $(V, +, \times, d)$ is said to be complete if every Cauchy sequence $\{x_n\}_{n\in\mathbb{N}}$ converges to an element $x \in X$.

Definition 10. (Normed space) A normed space is a vector space $(V, +, \times)$ equipped with a map $\|\cdot\| : V \to \mathbb{R}$ called a *norm* satisfying

$$(i)\ \|v\| \geq 0 \tag{A.7}$$
$$(ii)\ \|v\| = 0 \Leftrightarrow v = 0 \tag{A.8}$$
$$(iii)\ \|\alpha v\| = |\alpha|\,\|v\| \tag{A.9}$$
$$(iv)\ \|v + w\| \leq \|v\| + \|w\| \tag{A.10}$$

for any $v, w \in V$ and any $\alpha \in \mathbb{R}$.

Metrics can be defined via norms through $d(x, y) := \|x - y\|$. As a result, every normed space is a metric space.

Definition 11. (Banach space) A normed space $(X, \|\cdot\|)$ is called a Banach space if it is complete.

Definition 12. (Inner-product space) An inner-product space is a vector space $(X, +, \times)$ equipped with a map $\langle\cdot,\cdot\rangle : X \times X \to \mathbb{R}$ called an *inner-product* satisfying

$$(i)\ \langle x, y\rangle = \langle y, x\rangle \tag{A.11}$$
$$(ii)\ \langle \alpha x + \beta y, z\rangle = \alpha\langle x, z\rangle + \beta\langle y, z\rangle \tag{A.12}$$
$$(iii)\ \langle x, x\rangle \geq 0 \tag{A.13}$$
$$(iv)\ \langle x, x\rangle = 0 \Leftrightarrow x = 0 \tag{A.14}$$

for any $v, w \in V$ and any $\alpha \in \mathbb{R}$.

Norms can be defined via inner-products through $\|x\| := \sqrt{\langle x, x\rangle}$. As a result, every inner-product space is also a normed space.

Definition 13. (Hilbert space) An inner-product space $(X, \langle\cdot,\cdot\rangle)$ is called a Hilbert space if it is complete.

Definition 14. (Bounded linear operator) Let $(V, \|\cdot\|_V)$ and $(W, \|\cdot\|_W)$ be two Banach spaces. A map $A : V \mapsto W$ is said to be a bounded linear operator if

$$\sup_{v\in V\setminus\{0\}} \frac{\|Av\|_W}{\|v\|_V} < \infty \tag{A.15}$$

Definition 15. (Operator norm) Let $A : V \mapsto W$ be a bounded linear operator. The operator norm is defined as

$$\|A\| := \sup_{v \in V \setminus \{0\}} \frac{\|Av\|_W}{\|v\|_V} \tag{A.16}$$

Definition 16. (Pointwise convergence) Let X, Y be two metric spaces and $\{f_n\}_{n \in \mathbb{N}}$ be a sequence of functions where $f_n : X \to Y$ for all n. The sequence is said to converge to a function $f : X \to Y$ if for every $x \in X$

$$\lim_{n \to \infty} f_n(x) = f(x) \tag{A.17}$$

The example below, taken from (Berlinet and Thomas-Agnan, 2011, §1), highlights an issue one has to pay attention to when working with spaces of functions.

Example 1. (Convergence does not imply pointwise convergence) Let P be the vector space of all polynomials over $[0, 1]$ and endow it with the norm

$$\|f\|_P = \left(\int_0^1 |f(x)|^2 \mathrm{d}x \right)^{1/2} \tag{A.18}$$

The sequence $\{p_n\}_{n \in \mathbb{N}}$, $p_n(x) = x^n$ converges to the zero function since

$$\lim_{n \to \infty} \|p_n - 0\|_P = \lim_{n \to \infty} \left(\int_0^1 x^{2n} \mathrm{d}x \right)^{1/2} \tag{A.19}$$

$$= \lim_{n \to \infty} \frac{1}{\sqrt{2n+1}} \tag{A.20}$$

$$= 0 \tag{A.21}$$

and yet $p_n(1) = 1, \forall n$, i.e., $|p_n(x) - 0(x)| \nrightarrow 0$.

Definition 17. (Span) Let X be a vector space and $B \subseteq X$ be a subset of it. The *span* of B is defined as the set

$$\operatorname{span} B = \left\{ \sum_{i=1}^{n} \lambda_i b_i \mid \lambda_i \in \mathbb{R}, b_i \in B, n \in \mathbb{N} \right\} \tag{A.22}$$

Definition 18. (Linear independence) Let X be a vector space and $B \subseteq X$ be a subset of it. B is said to be linearly independent if for every finite subset $\{b\}_{i=1}^n \subseteq B$, $\sum_{i=1}^n \lambda_i b_i = 0 \iff \lambda_1 = \cdots = \lambda_n = 0$.

Definition 19. (Hamel basis) Let X be a vector space and $B \subseteq X$. B is called a Hamel basis for X if B is linearly independent and $\operatorname{span} B = X$.

Proposition 16. Every vector space has a Hamel basis.

Proposition 17. All Hamel bases of a vector space have the same cardinality.

The concept of a Hamel basis is aligned with the more specific concept of a "basis" in finite-dimensional vector spaces.

Definition 20. (Dimension of a vector space) The dimension of a vector space denoted $\dim X$ is the cardinality of any Hamel basis B of X. If any B is not finite, X is said to be infinite-dimensional.

Proposition 18. Let $A \in \mathbb{R}^{d\times d}$ be an invertible matrix, $B \in \mathbb{R}^d$ and $c \in \mathbb{R}$. The following identity holds

$$\begin{bmatrix} A & B \\ B^\top & c \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} + \frac{1}{d}A^{-1}BB^\top A^{-1} & -\frac{1}{d}A^{-1}B \\ -\frac{1}{d}B^\top A^{-1} & \frac{1}{d} \end{bmatrix} \tag{A.23}$$

where $d = c - B^\top A^{-1}B$.

Perhaps explain the difference between SUP and MAX, INF and MIN.

B Properties of kernels

Let k, k_1 and k_2 be PD kernels (Definition 3) defined on $\Omega \times \Omega$, $\Omega \subseteq \mathbb{R}^n$. Let $\alpha \geq 0$ be an arbitrary positive scalar. We have that the new function $k^\star$ as defined by any of the following constructions is also a PD kernel

$$k^\star(x, x') := \alpha k(x, x') \tag{B.1}$$

$$k^\star(x, x') := k_1(x, x') + k_2(x, x') \tag{B.2}$$

$$k^\star(x, x') := k_1(x, x') k_2(x, x') \tag{B.3}$$

For the first three cases, see (Steinwart and Christmann, 2008, §4). □

Bibliography

Bauer, M., van der Wilk, M., and Rasmussen, C. E. (2016). Understanding probabilistic sparse gaussian process approximations. *Advances in neural information processing systems*, 29.

Beaglehole, D., Belkin, M., and Pandit, P. (2022). Kernel ridgeless regression is inconsistent for low dimensions. *arXiv preprint arXiv:2205.13525*.

Belkin, M., Hsu, D., Ma, S., and Mandal, S. (2019). Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854.

Belkin, M., Ma, S., and Mandal, S. (2018). To understand deep learning we need to understand kernel learning. In *International Conference on Machine Learning*, pages 541–549. PMLR.

Berlinet, A. and Thomas-Agnan, C. (2011). *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media.

Bertsekas, D. (2009). *Convex optimization theory*, volume 1. Athena Scientific.

Boyd, S., Parikh, N., and Chu, E. (2011). *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc.

Boyd, S. P. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Fasshauer, G. E. (2011). Positive definite kernels: past, present and future. *Dolomites Research Notes on Approximation*, 4:21–63.

Iske, A. (2018). *Approximation theory and algorithms for data analysis*. Springer.

Kanagawa, M., Hennig, P., Sejdinovic, D., and Sriperumbudur, B. K. (2018). Gaussian processes and kernel methods: A review on connections and equivalences. *arXiv preprint arXiv:1807.02582*.

Karvonen, T. (2022). Error bounds and the asymptotic setting in kernel-based approximation. *Dolomites Research Notes on Approximation*, 15(3).

Kidger, P. and Lyons, T. (2020). Universal approximation with deep narrow networks. In *Conference on learning theory*, pages 2306–2327. PMLR.

Kimeldorf, G. and Wahba, G. (1971). Some results on tchebycheffian spline functions. *Journal of mathematical analysis and applications*, 33(1):82–95.

Lederer, A., Conejo, A. J. O., Maier, K. A., Xiao, W., Umlauft, J., and Hirche, S. (2021). Gaussian process-based real-time learning for safety critical applications. In *International Conference on Machine Learning*, pages 6055–6064. PMLR.

Micchelli, C. A., Xu, Y., and Zhang, H. (2006). Universal kernels. *Journal of Machine Learning Research*, 7(12).

Pugh, C. C. (2002). *Real mathematical analysis*. Springer.

Schölkopf, B., Herbrich, R., and Smola, A. J. (2001). A generalized representer theorem. In *International conference on computational learning theory*, pages 416–426. Springer.

Schölkopf, B. and Smola, A. J. (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Searcóid, M. Ó. and Searcoid, M. (2002). *Elements of abstract analysis*. Springer.

Sejdinovic, D. and Gretton, A. (2012). What is an RKHS? *Lecture Notes*.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and De Freitas, N. (2015). Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25.

Steinwart, I. (2020). Reproducing kernel hilbert spaces cannot contain all continuous functions on a compact metric space. *arXiv preprint arXiv:2002.03171*.

Steinwart, I. and Christmann, A. (2008). *Support vector machines*. Springer Science & Business Media.

Temlyakov, V. (2008). Approximation in learning theory. *Constructive Approximation*, 27(1):33–74.

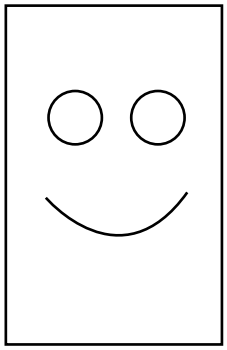
Wendland, H. (2004). *Scattered data approximation*. Cambridge university press.

Williams, C. K. and Rasmussen, C. E. (2006). *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA.

Wilson, J., Hutter, F., and Deisenroth, M. (2018). Maximizing acquisition functions for bayesian optimization. *Advances in neural information processing systems*, 31.

Zhang, Y., Duchi, J., and Wainwright, M. (2013). Divide and conquer kernel ridge regression. In *Conference on Learning Theory (COLT)*, pages 592–617. PMLR.

Personal details:



Name :	Mr. Sample CV
Address :	Samplestreet 70 6005 Luzern Switzerland
Date of Birth :	2nd of October 1981
Nationality :	Swiss
Legally work :	legally work in EU
Marital status :	with partner
Children :	none
Languages :	Chinese/Mandarin, English, French, German
Education level :	Bachelors degree
Hospitality work experience :	3-5 years
Special experience :	Europe work experience
Date of availability :	September 2009
Current location :	Africa
Travelling Status :	will be travelling single status
Telephone :	0041 41 370 6759
Email address :	jeff@h-g-r.com
Position(s) sought :	Permanent position for graduates
Department(s) sought :	Food & Beverage Bar/Sommelier

Personal profile:

As a Bachelor of Business Administration and after obtaining first relevant international work experience within the hospitality industry, I am now ready to take on new responsibilities to further my professional career.
My key strengths include strong analytical and logical skills, an eye for detail, communication and interpersonal skills.
I enjoy working in a team and help others progress. At the same time I work well independently.
As a highly motivated and driven individual I strive on taking up challenges.

Interests:

Travelling
Foreign Cultures
Photography
Sports

Educational qualifications:

Oct 99 - Feb 02	Higher Diploma (Hotel Management) Swiss Hotelmanagement School, SHL

Employment history:

Mar 04 - Ongoing	Assistant Manager (Rooms Division/Food & Beverage)
	Hotel Atlantic Kempinski Hamburg www.kempinski.com 5 star business hotel, part of Leading Hotels of the World 412 guest rooms, large function facilities, 3 food & beverage outlets
	Optimization of bar procedures, reinforcing SOPs
	Developing & implementing promotions
	Responsible for day-to-day operations
	Optimization and streamlining of housekeeping and laundry procedures
	Implementation of new SOPs
	Analyzing monthly reports for rooms division performance and sub departments
Mar 03 - Mar 04	Management Trainee
	Hospitality Graduate Recruitment www.h-g-r.com Leading company for placements within the Hospitality industry.
	Traineeship covering all aspects of an online recruitment agency.
Mar 02 - Mar 03	Management Trainee (Rooms Division)
	Hyatt Regency Xian, China www.hyatt.com 5 star business hotel 404 guest rooms, 4 food & beverage outlets
	Traineeship covering all rooms division departments on operational as well as supervisory level.

Training courses attended:

Mar 02 - Ongoing	OpenOffice - IT Courses
May 01 - Jan 03	Language Course - Chinese

References:

Hyatt Regency Xian
Patrick Sawiri, Phone: 86 22 2330 7654

Hospitality Graduate Recruitment
Jeff Ross, Phone: 41 41 370 99 88