Eindhoven University of Technology

MASTER

Gaussian Process Modeling for Non-Linear Feedforward Control with Task Flexibility

van Meer, M.

Award date:
2021

Eindhoven University of Technology
Department of Mechanical Engineering
Control Systems Technology

Gaussian Process Modeling for Non-Linear Feedforward Control with Task Flexibility

Master Thesis
CST2021.010

Max van Meer
0951669

Supervisors:
dr. ir. Tom Oomen
dr. Jim Portegies
ir. Maurice Poot

Eindhoven, February 2021

Abstract

Feedforward control is essential to achieve good tracking performance for positioning systems with flexible tasks. This thesis aims to obtain a model of the inverse system using solely input-output data of a system containing non-linear dynamics of unknown structure, in order to produce feedforward signals. To this end, the Kernel-based Inverse Model Control of Non-linear systems (KIMCON) framework is developed to parametrize the inverse system as a non-causal non-linear finite impulse response (NFIR) system, modeled as a Gaussian Process. Prior knowledge of the system can be taken into account by imposing a prior on the smoothness of the non-linear inverse dynamics using some stationary kernel function. It is shown numerically and experimentally that in comparison to linear methods, this framework leads to increased tracking performance for a range of non-linear systems, e.g., Wiener systems or systems subject to friction.

Contents

Chapter 1

Introduction

Motion systems that have to perform flexible positioning tasks require feedforward control to achieve the desired performance. Ideally, the references to be tracked are filtered by a model of the inverse system to achieve perfect tracking [18]. For a system H, the model of H^{-1} can be obtained by (i) identification of H and subsequent inversion to arrive at approximation $(\hat{H})^{-1}$ [28], or (ii) direct identification of the inverse of H, leading to approximation $\widehat{H^{-1}}$ [9]. It was shown in [5] that the latter poses an advantage over the former, since desired properties such as stability, smoothness, and finite preview or delay of $\widehat{H^{-1}}$ can be enforced directly on the model.

Obtaining $\widehat{H^{-1}}$ from physical modeling can be challenging in the presence of unknown dynamics. Instead, $\widehat{H^{-1}}$ can be obtained from input-output data [5]. This allows for the synthesis of control signals to perform flexible tasks without extensive modeling of the system to be controlled.

However, the direct identification of the non-causal system $\widehat{H^{-1}}$ in existing work (e.g., [5]) is typically limited to linear systems. Many systems exhibit non-linear behavior [23]. Therefore, some techniques, such as Iterative Learning Control with basis functions (BFILC) [27], parametrize $\widehat{H^{-1}}$ in terms of a sum of possibly non-linear *basis functions* and learn the corresponding weights from input-output data. However, the selection of these basis functions from first principles is hard, and when chosen incorrectly, it may lead to an inadequate representation of the true system and thus reduced tracking performance.

Existing black-box and grey-box modeling approaches for non-linear systems (such as [25, 10]) are unsuited for inverse model control using input-output data. Namely, these methods (a) are geared towards causal systems only [20], which is inadequate to model the non-causal system H^{-1}, (b) offer little interpretation of the model structure [31], or (c) are applicable only to systems that offer full state measurements [6].

This thesis aims to model H^{-1} as a non-causal non-linear finite impulse response (NFIR) system [23] for the synthesis of feedforward signals for flexible tasks. The reasoning behind this is as follows. In [5] it is shown that the inverse of non-minimum phase (NMP) linear systems H can be identified directly from data by imposing a non-causal finite impulse response (FIR) structure on H^{-1}. On the other hand, the grey-box NFIR modeling techniques in [23] are shown to be capable of modeling non-linear systems with *fading memory*, however, the considered systems are exclusively causal. Hence, these techniques cannot produce feedforward signals with preview for physical systems with possible delays and NMP dynamics.

Therefore, this thesis proposes a novel framework of feedforward control for systems with unknown non-linear dynamics by modeling *non-causal* NFIR systems. In this framework, named Kernel-based Inverse Model Control of Non-linear systems (KIMCON), H^{-1} is modeled as a *Gaussian Process* [22], which allows for the specification of prior knowledge on the *smoothness* of H^{-1}. It is shown numerically and experimentally that this technique can improve motion control performance for flexible tasks by modeling unknown non-linear dynamics from data.

This thesis is structured as follows. In Chapter 2, the control problem is formulated and the assumptions on the setting are described. It is explained why existing methods are insufficiently capable of solving the control problem under the given setting, and the main contributions of this thesis are listed. In Chapter 3, the proposed KIMCON approach is explained. Subsequently, KIMCON is applied to simulation examples in Chapter 4. Next, the approach is applied to a desktop printer in Chapter 5. Finally, in Chapter 6 the conclusions are given, as well as some recommendations for future work.

Chapter 2

Problem formulation

In this chapter, the control objective is formulated. The ability of some existing methods from the field of learning control to solve this control problem is evaluated to motivate for the proposed Gaussian Process-based solution. Finally, the contributions of this thesis are listed.

2.1 Setting and goal

Let H denote a discrete-time causal non-linear single-input single-output (SISO) system such that

$$H : y(t) = g(\mathbf{y}_t, \mathbf{u}_t). \tag{2.1}$$

Here, $y(t) \in \mathbb{R}$ denotes the system output and $u(t) \in \mathbb{R}$ the control effort at time $t \in \mathbb{N}$. The sequences $\mathbf{y}_t$ and $\mathbf{u}_t$ are defined as $\mathbf{y}_t = [y(t-1), y(t-2), \dots, y(t-m)]^\top$, $\mathbf{u}_t = [u(t-1), u(t-2), \dots, u(t-m)]^\top$. Following [5], H^{-1} is assumed to be a non-causal non-linear finite impulse response (NFIR) system.

Definition 2.1.1 (NFIR)**.** A causal single-input single-output non-linear finite impulse response (NFIR) system G_1 with input $a(t) \in \mathbb{R}$ and output $b(t) \in \mathbb{R}$ is defined as

$$G_1 : b(t) = f_1\left(a(t), a(t-1), \dots, a(t-n_c)\right), \tag{2.2}$$

with $n_c \in \mathbb{N}$ and f_1 a non-linear function. More generally, a *non-causal* NFIR system G_2 with the same inputs and outputs is defined as

$$G_2 : b(t) = f_2\left(a(t+n_{ac}), a(t+n_{ac}-1), \dots, a(t-n_c)\right), \tag{2.3}$$

with $n_{ac} \in \mathbb{N}$ and f_2 a non-linear function. ■

This leads to the following non-causal NFIR description of the inverse system H^{-1}:

$$H^{-1} : u(t) = f(\mathbf{x}_t) + \varepsilon \in \mathbb{R}, \tag{2.4}$$

such that $u(t)$ denotes the control effort required to generate the sequence $\mathbf{x}_t = [y(t+n_{ac}), \dots, y(t-n_c)]^\top \in \mathbb{R}^{n_\theta}$, $n_\theta = n_{ac} + n_c + 1$. The scalars n_{ac} and n_c represent the number of (anti-causal) preview samples and (causal) delayed samples respectively. The control effort is assumed to be affected by noise $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$, see Figure 2.1.

Remark 1. If g and f are linear functions, H is a FIR system and the setting just described is equivalent to the setting in [5]. A value of n_c greater than zero then allows (2.4) to be a representation of the inverse of non-minimum phase system H. ■

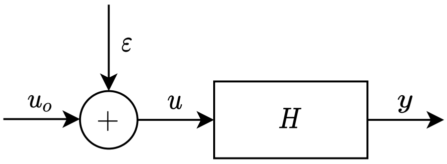


Figure 2.1: An open-loop setting with noise acting on the control effort.

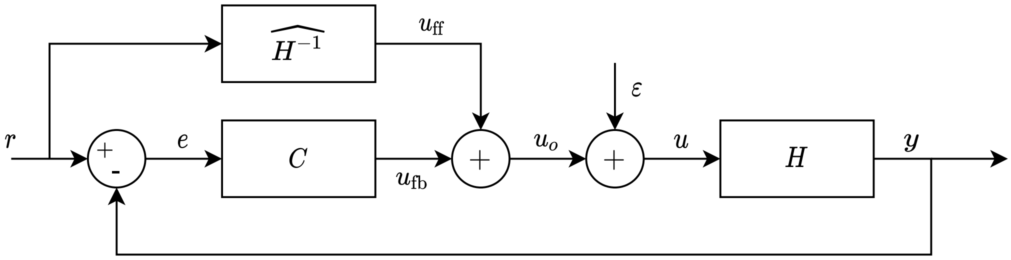


Figure 2.2: Feedback loop with parallel feedforward.

The goal is for H to track a number of known reference paths $\mathcal{R} := \{\mathbf{r}_1, \dots, \mathbf{r}_{n_R}\}$ with $\mathbf{r}_i = [r(0), \dots, r(T_i - 1)]^\top \forall i \in [1, \dots, n_R]$, where T_i denotes the number of samples of task i. To this end, any path $\mathbf{r}_i$ in $\mathcal{R}$ is split into overlapping sequences $\mathbf{x}_{*,t} = [r(t + n_{ac}), \dots, r(t - n_c)]^\top$, $\forall\ t \in [0, \dots, T_i - 1]$ such that $f(\mathbf{x}_{*,t})$ denotes the control effort $u(t)$ to produce reference sequence $\mathbf{x}_{*,t} \in \mathcal{R}$. In this setting, the goal is to obtain a (non-linear) model $\widehat{H^{-1}}$ of $f(\mathbf{x})$ such that a *feedforward signal* (or control effort) $\mathbf{u}_{\text{ff}} \in \mathbb{R}^{T_i}$ can be computed for any $\mathbf{r}_i \in \mathcal{R}$. This model is to be used to minimize the tracking errors $\mathbf{e}_i \in \mathbb{R}^{T_i}$ using the control scheme depicted in Figure 2.2.

It is assumed that sufficient prior knowledge of H is available for a data-set $\mathcal{D} = \{u(t), y(t)\}_{t=1}^{N}$ to be obtained that contains observations of sequences $\mathbf{x}_t = [y(t + n_{ac}), \dots, y(t - n_c)]^\top$ that are *similar* to reference sequences $\mathbf{x}_{*,t} = [r(t + n_{ac}), \dots, r(t - n_c)]^\top \in \mathcal{R}$, where similarity is defined by Euclidian distance in $\mathbb{R}^{n_\theta}$. Note that N denotes the total number of samples $u(t)$ matching with observed sequences $\mathbf{x}_t$. This condition is explained in detail in Section 3.4. Effectively, it poses the requirement that some initial approximate model $\widehat{H^{-1}}_{\text{prior}}$ or feedback controller C is available.

2.2 Existing methods and their shortcomings

Numerous approaches exist that aim to achieve the goal of letting system H produce reference outputs $\mathcal{R}$. A number of these will be covered briefly and compared by their ability to satisfy the following requirements:

R1: The approach must be able to model non-causal systems, such that it is applicable to inverse model control.

R2: For a SISO system H, the approach must use output data y, as opposed to relying on full state measurements.

R3: The approach must be able to produce feedforward signals for a variety of tasks, extrapolating from the data-set.

R4: The approach must be able to model systems with non-linear dynamics of unknown structure.

These requirements are assessed for various control techniques in Table 2.1. In the remainder of this section, the ability of these techniques to satisfy the requirements is elaborated.

2.2.1 Motion control for systems with prescribed non-linear dynamics

One technique to let a system H track $\mathcal{R}$ is Iterative Learning Control (ILC) [11], satisfying R1 and R2, as well as R4 under certain conditions. This approach updates the feedforward signal $\mathbf{u}_{\text{ff},j+1}$ based on the error $\mathbf{e}_j$ at trial j. Although ILC can in many cases reduce the error down to reproducible behavior, it converges to work for only a single reference $\mathcal{R} = \{\mathbf{r}_1\}$, i.e., it does not satisfy R3.

For multiple references in $\mathcal{R}$ (i.e., task flexibility), as posed in R3, ILC with basis functions [27] can be used, as explained in Appendix B. By approximating H^{-1} with the parametrization

$$\widehat{H^{-1}} : \mathbf{u}_{\text{ff}} = \Psi(\mathbf{r})v, \tag{2.5}$$

where $\Psi(\mathbf{r}) \in \mathbb{R}^{T \times n_v}$ is a basis containing polynomial functions of $\mathbf{r}$, and iteratively learning the parameters $v \in \mathbb{R}^{n_v}$, multiple references in $\mathcal{R}$ can be tracked after converging for only one. However, this technique requires explicit knowledge of the (non-linear) dynamics $\Psi(\mathbf{r})$, which may not be (completely) available, thus it does not satisfy R4. Additionally, similar to ILC it requires explicit model-knowledge of the process sensitivity for convergence.

Alternatively, kernel-based regularization (KBR) [5] can be used to identify a non-causal model $\widehat{H^{-1}}$ from data, to be used for feedforward. While KBR allows for task flexibility, it is aimed exclusively at linear systems, and hence R4 is not satisfied by this approach either.

2.2.2 Motion control for systems with unknown non-linear dynamics

Some methods learn the non-linear structure of systems from data, instead of expert knowledge, to satisfy R4. One such method is Probabilistic Inference for Learning Control (PILCO) [6]. By modeling H as a *Gaussian Process* (see Section 3.2) and conducting internal simulations, PILCO uses gradient-based optimization to learn a policy

$$u(t) = \pi(\tilde{\mathbf{x}}_t, \tilde{\theta}), \tag{2.6}$$

to minimize some cost $J(\tilde{\mathbf{x}}_f)$ of the final state $\tilde{\mathbf{x}}_f$. Here, $\tilde{\mathbf{x}}_t$ is a vector of measured system states and $\tilde{\theta}$ a vector of learned policy parameters. However, similar to ILC, PILCO converges to learn a policy that works for only a single reference in $\mathcal{R}$ and thus it does not satisfy R3. By relying on state measurements, it fails to satisfy R2 as well.

Moreover, in [20], the non-linear structure of H is learned by modeling (2.1) as a Gaussian Process. Although this structure contains NFIR models as a special case (when $\mathbf{y}_t$ is replaced with the empty set), this is a *causal* model of H, since $n_{ac} = 0$. The case where the system is non-causal, i.e., when modeling H^{-1}, is not covered in [20] and thus R1 is not satisfied. The application of this method to feedforward control is more involved than just the inclusion of $n_{ac} \neq 0$, since Gaussian Process modeling of inverse systems introduces unique difficulties on the collection of data, as will be shown in Chapter 3.

An alternative approach to learn the non-linear structure of systems from data is used in [3, 17, 16]. Here, H^{-1} is assumed to be of the form

$$u(t) = f_p(\mathbf{p}), \tag{2.7}$$

with $\boldsymbol{p} = \left[\ddot{\boldsymbol{q}}^\top, \dot{\boldsymbol{q}}^\top, \boldsymbol{q}^\top\right]^\top$ and $\mathbf{q} \in \mathbb{R}^n$ a vector of *generalized coordinates*. This function is then modeled as a Gaussian Process, in order to learn the non-linear structure of f_p from data. By modeling f_p, this approach is capable of computing feedforward signals for multiple references in $\mathcal{R}$, such that both R3 and R4 are satisfied. However, the assumption that $\mathbf{p}$ is measured means that the position, velocity and acceleration of the generalized coordinates are required, which is in contrast with R2. This setting is more restrictive than the setting assumed in this thesis, in which only input-output measurements are required, e.g., position measurements only. Moreover,

[3] requires an initial inverse model $\widehat{H^{-1}}_{\text{prior}}$. As will be shown in Chapter 3, for the KIMCON method proposed in this thesis it can be sufficient to have only a prior feedback controller.

2.3 Contributions

The requirements and knowledge gap described in the previous section has led to the following research question:

How to improve the reference tracking performance of single-input single-output systems with unknown non-linear dynamics using input-output data while allowing for task flexibility?

The answer to this question that this thesis provides is summarized by the following contributions:

C1: This thesis proposes Kernel-based Inverse Model Control for Non-linear systems (KIMCON) as a solution to the control problem, in which inverse systems are viewed as non-causal NFIR systems and modeled as a Gaussian Process. This approach satisfies all requirements R1-R4 defined in the previous section, see Table 2.1.

C2: The effectiveness of the framework is demonstrated using simulations and experiments of a range of non-linear systems.

C3: Moreover, the relation between the framework and the kernel-based identification approach in [5] is studied.

Table 2.1: Several control techniques compared.

	R1: Applicable to non-causal systems	R2: Uses SISO input-output data	R3: Allows for task flexibility	R4: Models unknown non-linear dynamics
ILC [11]	✓	✓		
KBR [5]	✓	✓	✓	
BFILC [27]	✓	✓	✓	
PILCO [6]				✓
Pillonetto [20]		✓	✓	✓
CTC-GPR [3]	✓		✓	✓
KIMCON	✓	✓	✓	✓

Chapter 3

Kernel-based Inverse Model Control of Non-linear systems

This chapter proposes Kernel-based Inverse Model Control of Non-linear systems (KIMCON), constituting contribution C1. The inverse system H^{-1} is viewed as a non-causal non-linear finite impulse response (NFIR) system, modeled as a Gaussian Process, in order to produce feedforward signals for flexible tasks.

First, the identification problem of H^{-1} is framed as an optimization problem in Section 3.1, and it is shown how this relates to Gaussian Process regression in Section 3.2. Section 3.3 shows how the non-linear dynamics of unknown structure are represented by stationary covariance functions. Subsequently, Section 3.4 describes how an appropriate data-set can be obtained, using any prior knowledge available. The kernel hyper-parameters are then optimized based on the data, as explained in Section 3.5. The practical implementation of KIMCON is described in Section 3.6.

The chapter concludes with a discussion about the applicability of the framework and some possible extensions.

3.1 Definition of the optimization problem

The aim is to obtain feedforward signals $\mathbf{u}_{\text{ff,i}}$ to track references $\mathbf{r}_i \in \mathcal{R}$. As shown in Section 2.1, the inverse system H^{-1} is assumed to be a non-causal NFIR system in the form of (2.4). The goal is then to find a representation of non-linear function f, such that the computation of $f(\mathbf{x}_{*,t})$ yields the control effort $u(t) \in \mathbb{R}$ required to produce reference sequence $\mathbf{x}_{*,t} = [r(t+n_{ac}), \ldots, r(t-n_c)]^\top$.

To this end, f is parametrized in terms of observed output sequences $\mathbf{x}_i = [y(i+n_{ac}), \ldots, y(i-n_c)]^\top \in \mathcal{D}\ \forall i \in [1, \ldots, N]$ as

$$u(t) =: f(\boldsymbol{\alpha} \mid \mathbf{x}_t) = \sum_{i=1}^{N} \alpha_i k\left(\mathbf{x}_t, \mathbf{x}_i\right), \tag{3.1}$$

with coefficients $\boldsymbol{\alpha} := [\alpha_1, \ldots, \alpha_N]^\top,\ \alpha_i \in \mathbb{R}$ and *kernel function* $k\left(\mathbf{x}_i, \mathbf{x}_j\right) = \left\langle k\left(\cdot, \mathbf{x}_i\right), k\left(\cdot, \mathbf{x}_j\right)\right\rangle_{\mathcal{K}}$ defining the reproducing kernel Hilbert space (RKHS) $\mathcal{K}$, defined as:

Definition 3.1.1 (Reproducing kernel Hilbert space [22])**.** Let $\mathcal{K}$ be a Hilbert space of real functions f defined on an index set $\mathcal{X}$. Then $\mathcal{K}$ is called a reproducing kernel Hilbert space endowed with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{K}}$ (and norm $\|f\|_{\mathcal{K}} = \sqrt{\langle f, f \rangle_{\mathcal{K}}}$) if there exists a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ with the following properties:

1. for every $\mathbf{x}, k\left(\mathbf{x}, \mathbf{x}'\right)$ as a function of $\mathbf{x}'$ belongs to $\mathcal{K}$, and

2. k has the reproducing property $\langle f(\cdot), k(\cdot,\mathbf{x})\rangle_{\mathcal{K}} = f(\mathbf{x})$. ■

To find the coefficients $\boldsymbol{\alpha}$ that best explain the data in $\mathcal{D}$, we first define *training matrix* X and *training targets* $\mathbf{u}$ as

$$\begin{aligned} X &:= \begin{bmatrix}\mathbf{x}_1 & \dots & \mathbf{x}_N\end{bmatrix}^\top \in \mathcal{D} \\ &:= \begin{bmatrix} y(n_c+n_{ac}) & y(n_c+n_{ac}-1) & \dots & y(0) \\ y(n_c+n_{ac}+1) & y(n_c+n_{ac}) & \dots & y(1) \\ \vdots & \vdots & & \vdots \\ y(N-1+n_c+n_{ac}) & y(N-2+n_c+n_{ac}) & \dots & y(N-1) \end{bmatrix}, \\ \mathbf{u} &:= \begin{bmatrix}u(1) & \dots & u(N)\end{bmatrix}^\top \in \mathcal{D}. \end{aligned} \tag{3.2}$$

From here on, the data-set is referred to as $\mathcal{D} = \{X, \mathbf{u}\}$. The following regularized least squares problem is then proposed:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} J[f] &= \|\mathbf{u} - \mathbf{f}\left(\boldsymbol{\alpha} \mid X\right)\|_2^2 + \sigma_n^2 \|\mathbf{f}\left(\boldsymbol{\alpha} \mid X\right)\|_{\mathcal{K}}^2 \\ &= \|\mathbf{u} - K\boldsymbol{\alpha}\|_2^2 + \sigma_n^2\|K\boldsymbol{\alpha}\|_{\mathcal{K}}^2 \\ &= \|\mathbf{u} - K\boldsymbol{\alpha}\|_2^2 + \sigma_n^2\boldsymbol{\alpha}^\top K^\top K^{-1} K\boldsymbol{\alpha} \\ &= \|\mathbf{u} - K\boldsymbol{\alpha}\|_2^2 + \sigma_n^2\boldsymbol{\alpha}^\top K\boldsymbol{\alpha}, \end{aligned} \tag{3.3}$$

with $\mathbf{f}(\boldsymbol{\alpha}) := [f(\mathbf{x}_1), \dots, f(\mathbf{x}_N)]^\top$ and K shorthand for the Gramian $K(X,X) = K(X,X)^\top$, with elements $K_{ij} = k(\mathbf{x}_i, \mathbf{x}'_j)$. It is shown in [22, Section 6.2.2] that the minimizer of (3.3) is obtained for

$$\hat{\boldsymbol{\alpha}} = \left(K + \sigma_n^2 I\right)^{-1}\mathbf{u}. \tag{3.4}$$

Substitution of this minimizer in (3.1) yields the predicted control effort $u(t)$ required to track a reference sequence $\mathbf{x}_{*,t}$ at time t:

$$\hat{f}(\mathbf{x}_{*,t}) = \mathbf{k}(\mathbf{x}_{*,t})^\top (K + \sigma_n^2 I)^{-1}\mathbf{u}, \tag{3.5}$$

with $\mathbf{k}(\mathbf{x}_{*,t}) = [k(\mathbf{x}_1, \mathbf{x}_{*,t}), \dots, k(\mathbf{x}_N, \mathbf{x}_{*,t})]^\top$. It is shown next how this result relates to Gaussian Process regression.

3.2 Non-causal systems as a Gaussian Process

Gaussian Process regression is explained first, after which the application to identification problem (3.3) is given.

3.2.1 Gaussian Process regression

Gaussian Processes are defined as follows.

Definition 3.2.1 (Gaussian Process [22])**.** A Gaussian Process is defined as an indexed family of random variables $f(\mathbf{x}) \in \mathbb{R}$ with $\mathbf{x} \in \mathbb{R}^{n_\theta}$, any finite number of which have a joint Gaussian distribution. ■

The Gaussian Process is fully specified by its mean function $m(\mathbf{x}) := \mathbb{E}[f(\mathbf{x})] \in \mathbb{R}$ and covariance function

$$k\left(\mathbf{x}, \mathbf{x}'\right) := \operatorname{cov}(f(\mathbf{x}) - m(\mathbf{x}), f(\mathbf{x}') - m(\mathbf{x}')) \in \mathbb{R}, \tag{3.6}$$

where $\mathbf{x}$ and $\mathbf{x}'$ denote any two points in $\mathbb{R}^{n_\theta}$. The kernel function $k\left(\mathbf{x}, \mathbf{x}'\right)$ is symmetric and positive definite, i.e., $k\left(\mathbf{x}, \mathbf{x}'\right) = k\left(\mathbf{x}', \mathbf{x}\right)$ and $k\left(\mathbf{x}, \mathbf{x}'\right) \geq 0 \ \forall \ \mathbf{x}, \mathbf{x}'$. From here on, the choice $m(\mathbf{x}) = 0$ is made, such that a Gaussian Process of $f(\mathbf{x})$ is denoted as

$$f(\mathbf{x}) \sim \mathcal{GP}\left(0, k(\mathbf{x}, \mathbf{x}')\right) : \mathbb{R}^{n_\theta} \to \mathbb{R}. \tag{3.7}$$

By definition of a GP, there exists a joint distribution $(\mathbf{u}, \mathbf{f}_*)$ expressed in terms of $k(\mathbf{x}, \mathbf{x}')$, where

$$\mathbf{u} = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_N)]^\top + \varepsilon \tag{3.8}$$

is a vector of noisy observations in $\mathcal{D}$, see (3.2), and

$$\begin{aligned} \mathbf{f}_* &= \Big[\mathbb{E}[u_{\text{ff}}(0)], \dots, \mathbb{E}[u_{\text{ff}}(T-1)]\Big]^\top \\ &= \Big[\mathbb{E}[f(\mathbf{x}_{*,0})], \dots, \mathbb{E}[f(\mathbf{x}_{*,T-1})]\Big]^\top \end{aligned} \tag{3.9}$$

is a vector of control effort values required to produce reference sequences $\mathbf{x}_{*,j},\ j \in [0, \dots, T-1]$ of reference $\mathbf{r} \in \mathcal{R}$ of length T, grouped as

$$\begin{aligned} X_* &:= \begin{bmatrix} \mathbf{x}_{*,0} & \dots & \mathbf{x}_{*,T_i-1} \end{bmatrix}^\top \\ &:= \begin{bmatrix} r(n_c + n_{ac}) & r(n_c + n_{ac} - 1) & \dots & r(0) \\ r(n_c + n_{ac} + 1) & r(n_c + n_{ac}) & \dots & r(1) \\ \vdots & \vdots & & \vdots \\ r(T - 1 + n_c + n_{ac}) & r(T - 2 + n_c + n_{ac}) & \dots & r(T-1) \end{bmatrix}. \end{aligned} \tag{3.10}$$

By conditioning this joint distribution $(\mathbf{u}, \mathbf{f}_*)$ on observations X and $\mathbf{u}$, we obtain the posterior $p(\mathbf{f}_*|X, \mathbf{u}, X_*)$, i.e., the probability distribution of the feedforward signal $\mathbf{f}_*$ required to produce reference sequences $X_* \in \mathcal{R}$ given the data. This step is called *Gaussian Process regression* and the resulting *posterior distribution* is a Gaussian with mean and variance [22, Section 2.2]

$$\begin{aligned} \mathbb{E}[\mathbf{f}_*] &= K(X_*, X) \left[K(X, X) + \sigma_n^2 I\right]^{-1} \mathbf{u}, \\ \operatorname{cov}(\mathbf{f}_*) &= K(X_*, X_*) - K(X_*, X) \left[K(X, X) + \sigma_n^2 I\right]^{-1} K(X, X_*), \end{aligned} \tag{3.11}$$

where the elements of $K(X_*, X) = K(X, X_*)^\top \in \mathbb{R}^{M \times N}$, $K(X, X) \in \mathbb{R}^{N \times N}$ and $K(X_*, X_*) \in \mathbb{R}^{M \times M}$ are computed by evaluating $k(\mathbf{x}, \mathbf{x}')$ for the corresponding training inputs and test inputs.

3.2.2 System identification with a GP

When comparing solution (3.5) of optimization problem (3.3) to the posterior mean (3.11) of GP (3.7), it occurs that these are equivalent. This is explained as follows. It is shown in [22, Section 6.2.3] that (3.5) is the *maximum a posteriori* (MAP) estimate of the posterior $p(f(\mathbf{x}_*) \mid \mathbf{x}_*, X, \mathbf{u})$. Since the posterior of a GP is Gaussian, the MAP equals the mean and hence (3.11) is equivalent to (3.5).

As these results are equivalent, from here on H^{-1} in the non-causal NFIR form of (2.4) is viewed as Gaussian Process (3.7), knowing that by computation of the posterior mean, the solution to the identification problem (3.3) is obtained. The minimal example below shows NFIR systems can be modeled as a GP.

Example 1. Let $H : y(t) = 2u(t)$ denote the transfer function of a linear system and define $\mathbf{x}_t := [y(t)]$, i.e. $n_c = n_{ac} = 0,\ n_\theta = 1$, such that H contains only one direct-feedthrough term. We can then write H^{-1} as

$$u(t) = f(\mathbf{x}_t) = \frac{y(t)}{2} + \varepsilon, \quad \varepsilon \sim \mathcal{N}\left(0, 10^{-6}\right). \tag{3.12}$$

Three noisy observations of f at locations $\mathbf{x}_i$ are available, such that the training matrix is

$$X = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 \end{bmatrix}^\top = \begin{bmatrix} -3 & 0 & 3 \end{bmatrix}^\top, \tag{3.13}$$

with training targets

$$\mathbf{u} = \begin{bmatrix} -1.496 & 0.003 & 1.499 \end{bmatrix}^\top. \tag{3.14}$$

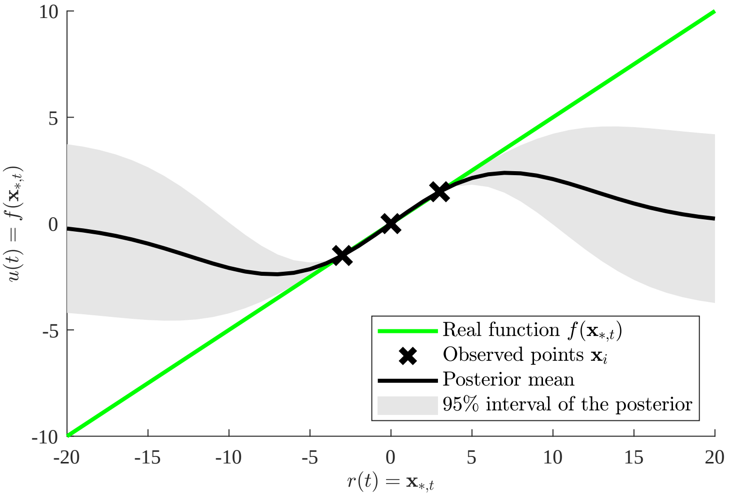


Figure 3.1: Posterior distribution of the Gaussian Process from Example 1. The posterior mean is close to the real function in the neighborhood of the observations.

The GP is defined with a *squared exponential* kernel function $k(\mathbf{x}, \mathbf{x}')$, see (3.16). The GP is used to make predictions of $\mathbf{f}_*$ at locations $X_* = [-20, \ldots, 20]^\top$ by computation of (3.11). The resulting *posterior distribution* is shown for each $\mathbf{x}_* \in X_*$ in Figure 3.1. The posterior mean is close to the real function in the neighborhood of the data. This is the result of the choice of the kernel function, which is discussed in more detail in Section 3.3.1. Example 6 in Appendix D.1 shows the posterior distribution in case a linear kernel function is chosen. ■

3.3 Parametrization of the system: kernel selection

The approximate system $\widehat{H^{-1}}$ is purely parametrized by the covariance function $k(\mathbf{x}, \mathbf{x}')$ in (3.1) and hence the selection of the kernel is critical. In this section, it is explained how the presented KIMCON approach parametrizes $\widehat{H^{-1}}$ using *stationary* covariance functions, in order to model systems with non-linear dynamics of unknown structure. Moreover, it is shown how some other control approaches which model systems with dynamics of *known* structure are retrieved by selecting other kernels.

3.3.1 Modeling systems with unknown structure

This section commences by defining stationary covariance functions, which can model systems with unknown non-linear dynamics. Subsequently, some examples of stationary covariance functions are given, representing either smooth, non-smooth, or periodic functions $f(\mathbf{x}_t)$.

Stationary covariance functions are defined as follows.

Definition 3.3.1 (Stationary covariance function [22]). A stationary covariance function is a function of $\mathbf{x} - \mathbf{x}'$, i.e., $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x} - \mathbf{x}')$. ■

If system (3.1) is parametrized in terms of a stationary covariance function, it follows that

$$u(t) =: f(\boldsymbol{\alpha} \mid \mathbf{x}_t) = \sum_{i=1}^{N} \alpha_i k\left(\mathbf{x}_t - \mathbf{x}_i\right), \tag{3.15}$$

i.e., $H^{-1} : f(\boldsymbol{\alpha} \mid \mathbf{x}_t)$ is parametrized purely by difference from $\mathbf{x}_t$ to observed sequences $\mathbf{x}_i \in \mathcal{D}$. In other words, such a model of H^{-1} is *data-dependent.* A value $u(t)$ required to produce sequence $\mathbf{x}_{*,t}$ is obtained by inference from measurements in $\mathbb{R}^{n_\theta}$, i.e. 'similar' reference sequences require 'similar' inputs $u(t)$.

Since stationary covariance functions lead to a data-dependent model, the collection of an appropriate data-set is critical. This is the topic of Section 3.4. First, the concept of smoothness in dynamic systems is explained and three examples of stationary covariance functions are given.

Smoothness of dynamic systems

A system H^{-1} is considered *smooth* in this thesis if the function f in (2.4) has infinitely many continuous derivatives with respect to $\mathbf{x}_t$. It is stressed that smoothness of H^{-1} relates to *dynamics*, and says nothing about the smoothness of some reference signal $\mathbf{r}$ over time. For example, the linear system $H^{-1} : u(t) = f(\mathbf{x}_t) = \theta^\top \mathbf{x}_t$ is smooth, since f has infinitely many continuous derivatives w.r.t. $\mathbf{x}_t$, regardless of the sequence $\mathbf{x}_t = [y(t+n_{ac}), \dots, y(t-n_c)]^\top$ extracted from $\mathbf{y}$, which might display sharp variations when plotted over time.

The smoothness of the prior posed on H^{-1} by a kernel $k(\mathbf{x}, \mathbf{x}')$ is different for each kernel. For smooth systems, the squared exponential (SE) is proposed, and for non-smooth systems, the Matèrn$_{3/2}$ kernel is proposed. The relation of both of these kernels to the smoothness of H^{-1} is the topic of Appendix C. Therein, the ability of both SE and Matèrn$_{3/2}$ kernels to represent non-smooth systems is visualized, and it is explained that Matèrn$_{3/2}$ kernels are better able to represent non-smooth functions since they are *finitely mean squared differentiable.* In the remainder of this section, the SE and Matèrn$_{3/2}$ kernels are defined, along with a kernel for periodic systems.

Smooth function parametrizations

The squared exponential (SE) kernel represents *smooth* non-linear functions, parametrizing $f(\mathbf{x}_t)$ in terms of standardized Euclidian distance:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2} (\mathbf{x} - \mathbf{x}')^\top \Lambda^{-1} (\mathbf{x} - \mathbf{x}') \right) \tag{3.16}$$

with hyper-parameters $\sigma_f^2 = \mathrm{Var}(f(\mathbf{x}))$ and $\Lambda = \mathrm{diag}([\ell_1, \dots, \ell_{n_\theta}])$. This kernel is able to represent smooth systems because it is infinitely MS-differentiable, see Appendix C. Note the following limits:

$$\begin{aligned} \lim_{\ell_\tau \to \infty} \frac{\partial k(\mathbf{x}, \mathbf{x}')}{\partial x'_\tau} &= \sigma_f^2, \\ \lim_{\ell_\tau \to 0} \frac{\partial k(\mathbf{x}, \mathbf{x}')}{\partial x'_\tau} &= 0, \end{aligned} \tag{3.17}$$

in other words, the contribution of dimension $y(t+\tau)$, $\tau \in [-n_c, \dots, n_{ac}]$ to $k(\mathbf{x}, \mathbf{x}')$ is constant or non-existent respectively at these limits.

Non-smooth function parametrizations

A covariance function capable of representing *non-smooth* non-linear functions is the Matèrn$_{3/2}$ kernel, given by

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \left(1 + \sqrt{3} r\right) \exp\left(-\sqrt{3} r\right), \tag{3.18}$$

with $r = \sqrt{(\mathbf{x} - \mathbf{x}')^\top \Lambda^{-1} (\mathbf{x} - \mathbf{x}')}$ and $\Lambda = \mathrm{diag}([\ell_1^2, \dots, \ell_{n_\theta}^2])$. Note that the lengthscales in Λ are squared in this kernel, contrary to the lengthscales in kernel (3.16). The limits in (3.17) hold for this kernel as well. Appendix C visualizes the prior defined by this kernel and the SE kernel, and explains why Matèrn$_{3/2}$ kernels are able to represent non-smooth non-linear systems. We continue with an example of a non-smooth non-linear system H^{-1}, modeled by a GP with a Matèrn$_{3/2}$ kernel.

Example 2. Consider a mass with $m = 1$ subject to Coulomb friction with friction coefficient $F_c = 1$. The system is subject to a force $u(t)$ and its velocity y is measured, such that the Euler-discretized inverse dynamics, obtained by the transformation $\frac{d}{dt} \to \frac{z-1}{T}$ with shift operator z and sample time $T_s = 0.1$, become

$$u(t) = f(\mathbf{x}_t) = \frac{1}{T_s}(y(t+1) - y(t)) + \text{sign}(y(t)), \tag{3.19}$$

with $\mathbf{x}_t = [y(t+1), y(t)]^\top$, such that $n_{ac} = 1$ and $n_c = 0$. An open-loop experiment (see Figure 3.2) of length $N = 500$ samples is done with $u(t) \sim \mathcal{N}\left(0, \frac{1}{4}\right)$ to form data-set $\mathcal{D}$. Gaussian Process (3.7) is trained with kernel function (3.18), and the posterior distribution for a range of test values $\mathbf{x}_{*,t} = [r(t+1), r(t)]^\top$ is computed. The real function $f(\mathbf{x}_t)$ for these test values and the posterior mean are shown in Figure 3.3. The control effort $u(t)$ is a function of samples $y(t)$ and $y(t+1)$. The non-linear sign-structure of (3.19) is seen from the ridge in the true model on the left.

By virtue of the stationary parametrization in terms of $\mathbf{x} - \mathbf{x}'$, the posterior mean $\mathbb{E}[f(\mathbf{x}_*)]$ is only accurate close to the observations in $\mathcal{D}$. In other words, the GP model of H^{-1} has been rendered data-dependent by the stationary covariance function. This is explained further in Section 3.4. ■

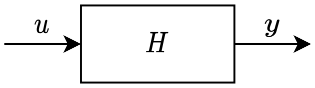
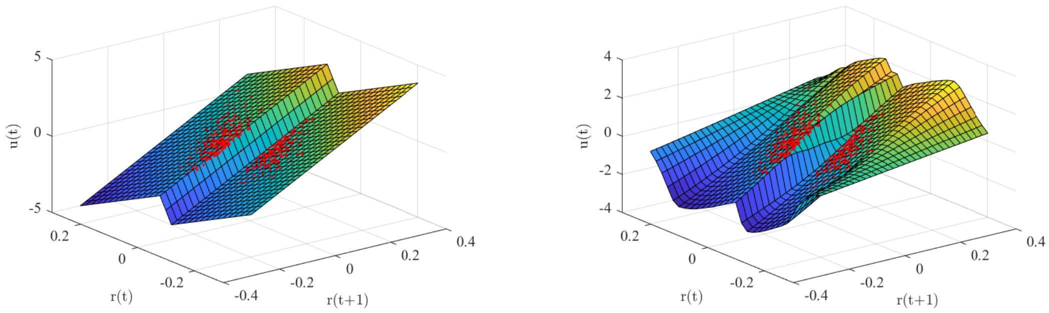


Figure 3.2: A noiseless open-loop setting.



(a) The real function $f(\mathbf{x}_*)$. Note the non-smooth non-linearity in the surface due to the non-linear sign term.

(b) The posterior mean $\mathbb{E}[f(\mathbf{x}_*)]$ of the GP with a Matèrn$_{3/2}$ kernel. The stationary kernel leads to a model which is consistent with the observations and is only able to extrapolate well close to the observations.

Figure 3.3: The inverse system H^{-1} of Example 2 visualized, with observed samples shown in red.

Periodic function parametrizations

Systems $f(\mathbf{x}_t)$ that are periodic in dimension i of $\mathbf{x}_t$ with period r_i can be modeled [13] by the stationary covariance function

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left[-\frac{1}{2}\sum_{i=1}^{n_\theta}\left(\frac{\sin\left(\frac{\pi}{r_i}(x_i - x_i')\right)}{\ell_i}\right)^2\right]. \tag{3.20}$$

What follows is an example of a function $f(\mathbf{x}_t)$ for which this kernel may be useful.

Example 3. Let H be a pendulum such that H^{-1} is described by the differential equation

$$u = m\ddot{y} + b\sin y, \tag{3.21}$$

with $m,\ b \in \mathbb{R}^+$. The discretized system becomes

$$u(t) = f(\mathbf{x}_t) = \frac{m}{T^2}\left(y(t+2) - 2y(t+1) + y(t)\right) + b\sin y(t), \tag{3.22}$$

with $\mathbf{x}_t = [y(t+2), y(t+1), y(t)]^\top$, i.e., $n_{ac} = 2$ and $n_c = 0$. The periodic term in (3.21) can be modeled by kernel (3.20) with $\ell_{1,2} = 0$ and $r_3 = 1$. The non-periodic terms can be modeled by adding a different kernel, see Appendix D.3. ■

3.3.2 Other parametrizations

In the proposed KIMCON approach, stationary covariance functions are used to parametrize non-causal NFIR systems. However, to place the approach in context, Appendix D describes how other approaches are retrieved by selecting alternative (non-stationary) covariance functions.

In Appendix D.1, it is shown how non-causal kernel-based regularization [5], described in Appendix A, is retrieved by choosing a linear kernel function, constituting contribution C3. Moreover, Appendix D.2 shows how Volterra systems can be represented by a particular kernel function. Furthermore, Appendix D.3 explains how some approaches that model H as a GP combine kernels to simultaneously identify linear and non-linear dynamics. It is explained in Appendix D.4 that such a parametrization is not directly applicable to KIMCON, in which H^{-1} is modeled as a GP.

The next section explains how an appropriate data-set $\mathcal{D}$ can be obtained.

3.4 Data collection

It was shown in the previous section that GP (3.7) with a stationary kernel requires observations of output sequences $\mathbf{x}_t = [y(t+n_{ac}), \ldots, y(t-n_c)]^\top$ to be *close* to reference sequences $\mathbf{x}_{*,t} = [r(t+n_{ac}), \ldots, r(t-n_c)]^\top$ in $\mathbb{R}^{n_\theta}$. However, since y is the *output* of system H and one can only choose the control effort u, this poses a challenge for data collection. In this section, the requirements on $\mathcal{D}$ are defined formally, and a procedure to achieve such a data-set is proposed.

3.4.1 Requirements on the data-set

Ideally, for inference from observations $\mathbf{x} \in \mathcal{D}$ to reference sequences $\mathbf{x}_* \in \mathcal{R}$ with a stationary kernel, the data-set $\mathcal{D}$ contains observations $\mathbf{x}_i$ *as close as possible around all reference sequences* $\mathbf{x}_*$ in $\mathbb{R}^{n_\theta}$. The following condition on data-set $\mathcal{D}$ for tasks $\mathcal{R}$ is proposed:

$$\begin{aligned} \forall \mathbf{x}_* \in \mathcal{R}\ \exists\ \mathcal{B} = \{\mathbf{x}_1, \ldots, \mathbf{x}_b\} \subset \mathcal{D}, \beta \in \mathbb{R}^b \geq \mathbf{0} \mid \beta_1\mathbf{x}_1 + \ldots + \beta_b\mathbf{x}_b = \mathbf{x}_*,& \\ \sum_{i=1}^{b}\beta_i = 1,& \\ \|\mathbf{x}_i - \mathbf{x}_*\|_2 \leq \epsilon\ \forall i \in [1, \ldots, b],& \end{aligned} \tag{3.23}$$

i.e., each references sequence $\mathbf{x}_* \in \mathcal{R}$ is a combination $\beta_1\mathbf{x}_1 + \ldots + \beta_b\mathbf{x}_b = \mathbf{x}_*$ of observations $\mathbf{x}_i \in \mathcal{B} \subset \mathcal{D}$. With $\sum_{i=1}^{b} \beta_i = 1$, this means that any $\mathbf{x}_*$ is *enclosed by* b observations $\mathbf{x}_i$ in $\mathbb{R}^{n_\theta}$. In other words, any $\mathbf{x}_*$ is a *convex combination* of observations $\mathbf{x}_i \in \mathcal{B}$. The observations $\mathbf{x}_i$ must have sufficiently short Euclidian distance to the reference sequences $\mathbf{x}_*$ in $\mathbb{R}^{n_\theta}$, dictated by some small number ϵ.

For $\epsilon \to 0$, each reference sequence $\mathbf{x}_*$ is present in the data-set as an observation $\mathbf{x} = \mathbf{x}_*$. If the output sequence $\mathbf{x} = [y(t+n_{ac}), \ldots, y(t-n_c)]^\top$ was observed, then also its corresponding control effort $u(t)$ was observed. In practice, one would be unable to obtain such a perfect data-set, since it requires perfect knowledge of $H^{-1} : u(t) = f(\mathbf{x}_t)$.

Moreover, since one only influences the control effort $u(t)$, and observations $\mathbf{x}_t$ follow from the system dynamics, it is near impossible in practice to obtain observations $\mathbf{x}$ such that Condition (3.23) is fully satisfied if ϵ is small. Therefore, the aim of data collection is to approximate Condition (3.23) as well as possible for a value of ϵ which is as small as possible. The next section describes how this is done in practice. First, an example is given of a data-set satisfying Condition (3.23).

Example 4. Let $r(t) = 1$ be the reference to be tracked by the non-linear system H and suppose $n_c = 0$, $n_{ac} = 1$, such that $n_\theta = 2$. For this simple example reference, any reference sequence is $\mathbf{x}_{*,t} = [r(t+1), r(t)]^\top = [1,1]^\top$. Suppose four observations $\mathbf{x}_t = [y(t+1), y(t)]^\top$ are available, namely, $\mathcal{B} = \{[0,0]^\top, [0,a]^\top, [a,0]^\top, [a,a]^\top\}$, with $a \in \mathbb{R} > 1$. Note that these output sequences need not be observed in any particular order.

In this case, the goal would be to design experiments that yield these four sequences, preferably with $a \to 1$. For $a = 1$, the reference $\mathbf{x}_{*,t}$ is observed exactly and no inference is required. Alternatively, if $a < 1$, the GP would need to extrapolate from the data, which yields satisfactory results solely close to the data when using a stationary covariance function, see Figures 3.1 and 3.3.

An example output sequence containing the aforementioned observations $\mathbf{x}_i$, $i \in [1, \ldots, 4]$ with $a = 2$ is

$$\mathbf{y} = [\overbrace{0 \quad \underbrace{0}_{}}^{\mathbf{x}_1^\top} \quad \overbrace{2 \quad 2}^{\mathbf{x}_3^\top} \quad 0]^\top, \quad \text{with } \underbrace{0\;2}_{\mathbf{x}_2^\top},\ \underbrace{2\;0}_{\mathbf{x}_4^\top} \tag{3.24}$$

Indeed, for $\beta_i = \frac{1}{4}\ \forall i \in [1, \ldots, 4]$, Condition (3.23) holds with $\epsilon = \sqrt{2}$. ■

How such data-sets can be obtained is explained next.

Remark 2. It is stressed that Condition (3.23) holds exclusively for stationary covariance functions. Linear covariance functions, while capable only of representing linear systems, allow for extrapolation in $\mathbb{R}^{n_\theta}$ far from observations, see Appendix D.1. Moreover, for periodic covariance functions, Condition (3.23) can be adapted to include the periodicity in f in order not to be too conservative. ■

3.4.2 Procedure

To design experiments that lead to observations $\mathbf{y}$ which are *similar* to $\mathbf{r}$ in $\mathbb{R}^{n_\theta}$, the closed-loop setting in Figure 2.2 is used. A summary of the required steps is given first, before elaborating each point in the sections below.

The procedure to obtain $\mathcal{D}$ is summarized as follows:

1. Obtain a stabilizing feedback controller $C(z)$ for H.

2. Obtain an initial approximate model $\widehat{H^{-1}}_{\text{prior}}$ using any technique available.

3. Design *excitation references* $\tilde{\mathcal{R}} := \{\tilde{\mathbf{r}}_1, \dots, \tilde{\mathbf{r}}_{n_{\tilde{R}}}\}$ with $\tilde{\mathbf{r}}_i = [\tilde{r}(0), \dots, \tilde{r}(\tilde{T}_i{-}1)]^\top \forall i \in [1, \dots, n_{\tilde{R}}]$, where $\tilde{T}_i$ denotes the number of samples in excitation reference $\tilde{\mathbf{r}}_i$. When these excitation references are applied to the closed-loop system, sequences $\mathbf{x}_i \in X$, $\forall i \in [1, \dots, N]$ (see (3.2)) should be observed that approximate Condition (3.23), i.e., enclosing each $\mathbf{x}_*$ in $\mathcal{R}$.
4. Format the data to obtain $\mathcal{D} = \{X, \mathbf{u}\}$.

The following sections further elaborate on these steps.

Feedback design

Feedback can be used to obtain outputs $\mathbf{y}$ similar to $\mathbf{r}$ in $\mathbb{R}^{n_\theta}$. Any feedback design technique can be used to design $C(z)$. Since H may be non-linear, special attention should be paid to assure that the output $\mathbf{y}$ remains bounded.

In closed-loop, measurement noise ε affecting u leads to u and y being correlated by feedback. This introduces a bias when estimating $\widehat{H^{-1}}$ from u and y. To reduce this bias, one can repeat each experiment an additional time with the same control effort. By including the results from both experiments, an average is obtained.

Inclusion of an approximate model

The initial model $\widehat{H^{-1}}_{\text{prior}}$ should be as accurate as possible, since a more accurate model causes observations $\mathbf{y}$ to be closer to the excitation references in $\tilde{\mathcal{R}}$. The user can then focus on the design of $\tilde{\mathcal{R}}$ such that when applied to the closed-loop system, the discrepancy between $\tilde{\mathbf{r}}_i$ and $\mathbf{y}_i$ is a result of only *unknown dynamics* not modeled by $\widehat{H^{-1}}_{\text{prior}}$, e.g., non-linear dynamics or higher order linear dynamics.

Excitation reference design

The excitation references leading to observations $\mathbf{y}$ allowing for inference to $\mathbf{r} \in \mathcal{R}$ in $\mathbb{R}^{n_\theta}$ depend heavily on $\mathcal{R}$. Since H^{-1} is unknown, this process may involve trial-and-error. Some examples of $\tilde{\mathcal{R}}$ include:

1. Periodic references with varying frequencies.
2. Scaled variations of the references $\mathcal{R}$.

Practical examples of excitation references are shown in Chapters 4 and 5.

Construction of the data-set

The designed excitation references $\tilde{\mathbf{r}}_i$ are used in $n_{\tilde{R}}$ closed-loop experiments in the setting of Figure 2.2, yielding outputs $\mathbf{y}_i = [y(0), \dots, y(\tilde{T}_i - 1)]^\top$ and control efforts $\mathbf{u}_i = [u(0), \dots, u(\tilde{T}_i - 1)]^\top$, $i \in [1, \dots, n_{\tilde{R}}]$. The resulting $n_{\tilde{R}}$ vectors of data are stacked as

$$\begin{aligned} \mathbf{y} &= [\mathbf{y}_1^\top, \dots, \mathbf{y}_{n_{\tilde{R}}}^\top]^\top, \\ \mathbf{u} &= [\mathbf{u}_1^\top, \dots, \mathbf{u}_{n_{\tilde{R}}}^\top]^\top. \end{aligned} \tag{3.25}$$

The training matrix X and training targets $\mathbf{u}$ are constructed by substituting (3.25) in (3.2), such that the total number of samples is $N = \sum_{i=1}^{n_{\tilde{R}}} T_i$ (see Example 4).

Remark 3. Note that the ordering of the rows in X and $\mathbf{u}$ is irrelevant, as long as each training input $\mathbf{x}_t = [y(t + n_{ac}), \dots, y(t), \dots, y(t - n_c)]^\top \in X$ is aligned with the corresponding training target $u(t) \in \mathbf{u}$. This allows for the stacking of data in (3.25). ■

As shown in the next section, the training time of the GP heavily relies on the number of samples N. Optionally, to reduce the size of the data-set, rows can be removed from X and $\mathbf{u}$. Indeed, physical systems with high sample-rates exhibit $y(t+1) \approx y(t)$. Consequently, $\mathbf{x}_{t+1} \approx \mathbf{x}_t$ and thus one of these samples adds little extra information. Therefore, to save computational time it may be desirable to take into account only every $n_s \in \mathbb{N}$ rows of X and $\mathbf{u}$ to form $\mathcal{D}$ and discard all other rows.

3.5 Optimization of hyper-parameters

The kernels mentioned in Section 3.3 all contain a number of hyper-parameters $\mathbf{\Theta}$, such as characteristic lengthscales ℓ_i, σ_n and σ_f. These can be chosen from expert knowledge or found by maximization of the *marginal likelihood*, sometimes referred to as *evidence maximization*. The marginal likelihood can be interpreted as the probability of the observed data given the model. For numerical reasons, the log of the marginal likelihood is considered [22, Section 5.4], defined as

$$\log p(\mathbf{u} \mid X, \mathbf{\Theta}) = -\frac{1}{2}\mathbf{u}^\top K_n^{-1}\mathbf{u} - \frac{1}{2}\log|K_n| - \frac{n}{2}\log 2\pi, \tag{3.26}$$

with $K_n = K + \sigma_n^2 I$ and $|K_n| := \det(K_n)$. To find maximizer $\hat{\mathbf{\Theta}}$ of the non-convex reward (3.26), active-set optimization [19, Section 7.4] is used. This requires the computation of the gradient

$$\begin{aligned}\frac{\partial}{\partial \Theta_j}\log p(\mathbf{u} \mid X, \mathbf{\Theta}) &= \frac{1}{2}\mathbf{u}^\top K_n^{-1}\frac{\partial K_n}{\partial \Theta_j}K_n^{-1}\mathbf{u} - \frac{1}{2}\operatorname{tr}\left(K_n^{-1}\frac{\partial K_n}{\partial \Theta_j}\right)\\ &= \frac{1}{2}\operatorname{tr}\left(\left(\boldsymbol{\alpha}\boldsymbol{\alpha}^\top - K_n^{-1}\right)\frac{\partial K_n}{\partial \Theta_j}\right),\end{aligned} \tag{3.27}$$

where $\boldsymbol{\alpha} = K_n^{-1}\mathbf{u}$. This gradient needs to be computed with respect to each hyper-parameter Θ_j.

Remark 4. Since the stationary covariance functions used in KIMCON (see Section 3.3.1) contain a hyper-parameter ℓ_i for each $n_\theta = n_{ac} + 1 + n_c$ dimensions in $\mathbf{x}$, the required number of computations for hyper-parameter optimization grows with n_θ. The computational complexity of the inversion of K_n is $\mathcal{O}(N^3)$. The computation of the gradients (3.27) involves another $\mathcal{O}(N^2)$ computations *per hyper-parameter* [22, Section 5.4]. For, e.g., the SE kernel (3.16), the total complexity for the computation of the marginal likelihood and gradients thus is $\mathcal{O}(\overline{N}^3 + \overline{N}^2(n_\theta + 2))$, where the extra 2 hyper-parameters are σ_n and σ_f. Here, $\overline{N}$ denotes the length of the active set, i.e. a subset of X and $\mathbf{u}$ used for optimization. Since active-set optimization requires the computation of these gradients at every iteration, optimization of hyper-parameters can be time consuming for large data-sets. For example, the experiments of Chapters 4 and 5 require up to 36 hours of training[1]. ■

3.6 Implementation

The presented KIMCON approach is summarized in Algorithm 1. While any suitable software package may be employed to implement the approach, in this thesis the MATLAB Statistics and Machine Learning Toolbox [15] is used. Several implementation aspects of KIMCON will be clarified next. A full example of the implementation of KIMCON is given in Appendix F, where the MATLAB code of the example in Section 4.1 is included.

3.6.1 Gaussian Process regression in MATLAB

To train the Gaussian Process, Line 8 of Algorithm 1 is executed using the `fitrgp` function from the aforementioned MATLAB toolbox. To compute the posterior mean in Line 11, the `predict` function of the same toolbox is used.

[1] All computations in this thesis are carried out on a desktop PC with a quad-core Intel i7 3.4 GHz processor and 16GB of RAM.

Remark 5. If using the `fitrgp` function in MATLAB R2020a, the user should manually disable the use of basis functions and choose a kernel with the `ard` prefix to ensure that each dimension i of $\mathbf{x}$ admits a separate lengthscale ℓ_i. Counter-intuitively, when using the `ardsquaredexponential` kernel (3.16) or `ardmatern32` kernel (3.16), hyper-parameters ℓ_i, σ_n and σ_f are optimized even if the `OptimizeHyperparameters` parameter is disabled. ■

3.6.2 Assessment of the data-set

Verifying whether each $\mathbf{x}_{*,t} \in \mathcal{R}$ is enclosed by observations $\mathbf{x}_i \in \mathcal{D}$ in $\mathbb{R}^{n_\theta}$ in Condition (3.23) can be done in several ways. The direct assessment of Condition (3.23) could be done by applying a k-nearest neighbors algorithm [24] to each $\mathbf{x}_{*,t} \in \mathcal{R}$ to find $\mathcal{B}$ and numerically verify the condition for some user-specified value of ϵ.

However, such a direct assessment of Condition (3.23) is computationally expensive. Moreover, the condition needs only to be *approximately* met for inference to $\mathcal{R}$ from $\mathcal{D}$. For smooth references, such as the ones used in Chapters 4 and 5, a quantitative assessment of Condition (3.23) may not be required. Rather, from practical experience it appears that in many cases it suffices to visually assess the similarity of signals $\mathbf{y}_i,\ i \in [1, \dots, n_{\tilde{R}}]$ and $\mathbf{r}_j,\ j \in [1, \dots, n_R]$. The user should be aware that similarity in n_θ is key, i.e., when n_θ grows large, it becomes increasingly difficult to visually assess the quality of the data-set.

Moreover, it is important to note that sequences $\mathbf{x}_t = [y(t + n_{ac}), \dots, y(t - n_c)]^\top$ extracted from observations $\mathbf{y}_i,\ i \in [1, \dots, n_{\tilde{R}}]$ are time-invariant: the t in these terms serves as nothing more than a label to match the observation $\mathbf{x}_t$ with control effort $u(t)$, see (2.4). This should be taken into consideration when assessing similarity between $\mathbf{y}_i$ and $\mathbf{r}_j$ over time. See also Figure 4.4, where observations $\mathbf{y}_i$ with negative slope around $t = 60$ [s] yield sequences $\mathbf{x}_t$ that contribute to the model when computing the posterior mean for the reference $\mathbf{r}$ around $t = 30$ [s].

Algorithm 1: Kernel-based Inverse Model Control of Non-linear systems (KIMCON)

```
Result: A feedforward signal $\mathbf{u}_{\text{ff},i} \in \mathbb{R}^{T_i}$ for each reference $\mathbf{r}_i \in \mathcal{R}$.
Input: References $\mathcal{R}$, initial model $\widehat{H^{-1}}_{\text{prior}}$, feedback controller $C$.
Choose a stationary kernel $k(\mathbf{x}, \mathbf{x}')$ based on the smoothness and periodicity of $H^{-1}$, see
  Section 3.3;
while Condition (3.23) not approximately satisfied do
    Design $n_{\tilde{R}}$ excitation references $\tilde{\mathcal{R}}$, see Section 3.4;
    Perform $n_{\tilde{R}}$ closed-loop experiments with $\tilde{\mathcal{R}}$ using $\widehat{H^{-1}}_{\text{prior}}$ and $C$, resulting in $N$
      observations of $\mathbf{x}_t$ and $u(t)$;
    Assess Condition (3.23);
end
Format the data-set to obtain $\mathcal{D} = \{X, \mathbf{u}\}$;
Maximize the log marginal likelihood (3.26) to obtain kernel hyper-parameters $\mathbf{\Theta}$;
for each reference $\mathbf{r}_i \in \mathcal{R}$ do
    Format $\mathbf{r}_i$ as $X_*$, see (3.10);
    Compute the posterior mean $\mathbf{u}_{\text{ff,i}} = \mathbb{E}[\mathbf{f}_*]$ with (3.11);
end
```

3.7 Discussion

This section describes the applicability of KIMCON and gives some opportunities for improvement.

3.7.1 Applicability

KIMCON was developed to produce feedforward signals for systems with non-linear dynamics of unknown structure and task flexibility. Within this class of problems, the technique works better for some subclasses than for others. Deciding factors are the types of references to be tracked and the nature of the non-linear dynamics.

Shape of the reference

The approach is easiest to implement with *smooth* references $\mathcal{R}$, as will be shown in Chapters 4 and 5. With only limited prior knowledge (e.g., only a feedback controller), a data-set can be obtained that allows for inference of observations to the smooth reference using the procedure described in Section 3.4. However, when the shape of the reference is more complex, it becomes increasingly difficult to obtain such a data-set.

Task flexibility

While KIMCON does allow for task flexibility, it is imperative that the reference sequences $\mathbf{x}_{*,t}$ are close to the observed sequences $\mathbf{x}_i$, because the model $\widehat{H^{-1}}$ is data-dependent. As a result, task flexibility is limited to reference paths similar to paths observed during training. This is the price paid for the fact that no non-linear basis functions are required. In contrast, techniques such as BFILC can extrapolate to arbitrary references because the coefficients of user-specified non-linear basis functions are learned. When a good representation of the non-linear basis functions is available, BFILC may be preferred.

Fading memory

Furthermore, stationary kernels described in Section 3.3 represent a large range of non-linear functions, however, by parametrizing H^{-1} as a non-causal NFIR system, only systems with *fading memory* can be modeled [23]. Such systems rely on only a *finite* number of preview samples and delayed samples to determine the next system state. In contrast, inverse systems H^{-1} containing bifurcations, jumps, or moving resonances would require an infinite history (or memory) to determine the next system state. Such systems would require a non-linear infinite impulse response (NIIR) parametrization and are outside the scope of KIMCON.

3.7.2 Extensions

The following extensions of KIMCON may further increase its practical applicability in the future.

Iterative data-collection

As the performance of KIMCON is largely dependent on the distance between observed sequences $\mathbf{x}$ and reference sequences $\mathbf{x}_*$, any prior knowledge $\widehat{H^{-1}}_{\text{prior}}$ that can be used during data collection can be of use. Since KIMCON yields an estimate $\widehat{H^{-1}}$ given $\widehat{H^{-1}}_{\text{prior}}$, it may be worth investigating if an iterative procedure may increase its performance. For example, by setting $\mathcal{D}_{j+1} = \mathcal{D}_j \cup \widehat{H^{-1}}_j\tilde{\mathcal{R}}$ after each iteration j and training a new GP. Initial experiments of this kind have yielded no satisfactory results, but it may deserve more attention in the future.

Computational speed

The computational complexity of training a Gaussian Process is significant, as shown in Section 3.5. The results in Chapters 4 and 5 require up to several dozen hours of training, which can hamper development speed in practical situations. Therefore, it is worth researching whether the sparse Gaussian Process methods in [12, 26] are applicable to KIMCON.

Noise on the output

A Gaussian Process is typically defined for the case when noisy observations of $f(\mathbf{x})$ are available. Since KIMCON models H^{-1} as a GP, this introduces the assumption of noise $\varepsilon(t)$ on $u(t)$, as shown in Figure 2.2. Many physical systems, however, also exhibit noise on $y(t)$. To deal with this noise, the approaches proposed in [22, Section 9.5] might be employed.

Chapter 4

Numerical validation

In this chapter, two numerical examples are presented to demonstrate the presented KIMCON approach, constituting contribution C2. First, to show how the non-linear structure of a system can be completely inferred from data using a GP, a Wiener system is considered. Moreover, the approach is applied to a simulation of a wire-bonder, to demonstrate how a small number of simple experiments per task can lead to a noticeable reduction in the tracking error.

4.1 Feedforward control of a Wiener system

This section describes the application of the presented KIMCON approach to a Wiener system, defined as:

Definition 4.1.1 (Wiener system)**.** A Wiener system is defined as the cascade of a dynamic linear system followed by a static non-linear system. ■

See Figure 4.1 for a visual representation. The inverse of a Wiener system is a Hammerstein system:

Definition 4.1.2 (Hammerstein system)**.** A Hammerstein system is defined as the cascade of a dynamic linear system preceded by a static non-linear system. ■

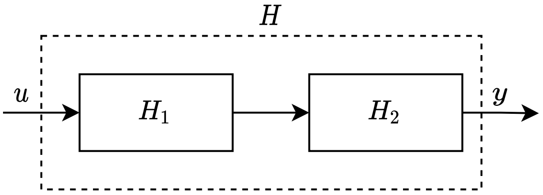


Figure 4.1: A cascade of systems H_1 and H_2. If H_1 is a dynamic linear system and H_2 is static non-linear system, H is a Wiener system. Conversely, if H_1 is a static non-linear system and H_2 is a dynamic linear system, H is a Hammerstein system.

By assuming merely sufficient knowledge of the *linear* component of the Wiener system to design a feedback controller, the procedure is shown to be able to model the unknown non-linear dynamics well around a given reference.

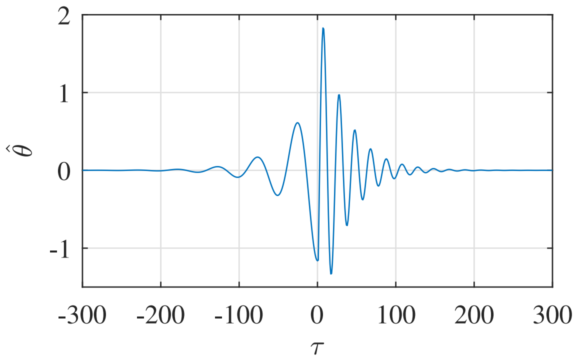


Figure 4.2: Impulse response coefficients of the linear system $P^{-1}(z)$, with $n_c = n_{ac} = 300$. It it clearly visible that $P^{-1}(z)$ is non-causal, since the impulse at time $\tau = 0$ leads to a response at $\tau < 0$.

Setting and goal

Consider the linear example system from [5]:

$$P(z) := \frac{\tilde{y}(z)}{u(z)} = \frac{1.550\left(z^2 - 2.035z + 1.052\right)\left(z^2 - 1.844z + 0.9391\right)}{z^2(z - 0.9514)(z - 0.9511)} \tag{4.1}$$

which has two NMP zeros at $1.018 \pm 0.126\text{i}$ and two minimum-phase zeros at $0.922 \pm 0.298\text{i}$. In [5], the linear system $P^{-1}(z)$ is approximated as a non-causal FIR system with $n_c = n_{ac} = 300$, such that the function

$$f_P(\mathbf{x}_t) = \hat{\theta}^\top \mathbf{x}_t, \tag{4.2}$$

with $\mathbf{x}_t = [y(t + n_{ac}), \ldots, y(t - n_c)]^\top$, defines the FIR approximation $\widehat{P^{-1}}$. Here, $\hat{\theta} \in \mathbb{R}^{n_\theta}$ are the impulse response coefficients, see also Appendix A. These impulse response coefficients are depicted in Figure 4.2.

In this example, the output of $P(z)$ is fed through a static non-linear function such that $y(t) = \sqrt{|\tilde{y}(t)|}$, see Figure 4.3. Consequently, the system H to be controlled is a Wiener system, described by

$$y(t) = \sqrt{|P(q)\left(u(t) - \varepsilon(t)\right)|}, \tag{4.3}$$

with noise signal $\varepsilon(t) \sim \mathcal{N}(0, \sigma_n^2)$, $\sigma_n = 10^{-3}$ acting on the input of H and q the forward shift operator such that $q^\tau a(t) := a(t + \tau), a \in \mathbb{R}$. The inverse system H^{-1} is then defined as the following Hammerstein system:

$$u(t) =: f(\mathbf{x}_t) + \varepsilon(t) = P^{-1}(q)\left(y(t)\right)^2 + \varepsilon(t). \tag{4.4}$$

Since the non-linear square term in $f(\mathbf{x}_t)$ is *static*, i.e., not a function of q, it follows that $f(\mathbf{x}_t)$ can be rewritten as a combination of the impulse response coefficients $\hat{\theta}$ of $\widehat{P^{-1}}$:

$$u(t) =: f(\mathbf{x}_t) = \hat{\theta}^\top\left(\mathbf{x}_t \odot \mathbf{x}_t\right) + \varepsilon(t), \tag{4.5}$$

where $\odot$ denotes element-wise multiplication. The assumption that the NFIR function $f(\mathbf{x}_t)$ in (4.5) is a valid parametrization of H^{-1} then follows naturally from the assumption that the FIR function $f_P(\mathbf{x}_t)$ in (4.2) is a valid parametrization of P^{-1}, as shown in Appendix A.

The goal is to track a single reference:

$$r(t) = \begin{cases} 0.5\cos(\frac{2\pi}{40}t) + 1 & 20 < t < 60 \\ 1.5 & \text{otherwise,} \end{cases} \tag{4.6}$$

with discrete-time representation $\mathbf{r} = [r(0), \ldots, r(T-1)]^\top \in \mathbb{R}^T$, where $T = 1500$, with sample time $T_s = \frac{1}{20}$ [s]. This reference is to be tracked using the closed-loop scheme of Figure 4.3.

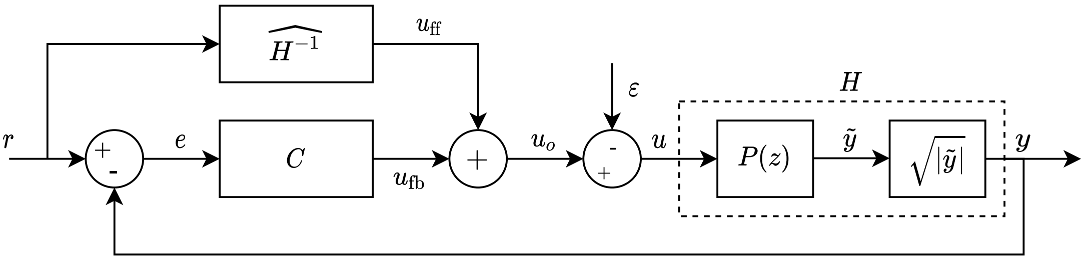


Figure 4.3: Control scheme of the Wiener example.

Approach

Following the approach described in Algorithm 1, a system parametrization is selected first. Assuming that the non-linear dynamics of H^{-1} are smooth (as defined in Section 3.3.1), the SE kernel (3.16) is chosen.

A dataset $\mathcal{D}$ is collected by performing $n_{\tilde{R}} = 20$ closed-loop experiments with linear controller

$$C(z) = \frac{0.09688z^2 + 0.001974z - 0.0929}{z^3 - 2.844z^2 + 2.783z - 0.9391}, \tag{4.7}$$

which was tuned with knowledge of the linear system $P(z)$. The excitation references $\tilde{\mathcal{R}}$ are chosen as

$$\tilde{r}_i(t) = 1 - 0.5\cos(2\pi\frac{3b_i}{100}t - \pi) \in \tilde{\mathcal{R}},\ b_i \in [0.3, \dots, 1.5], \tag{4.8}$$

where $i \in [1, \dots, n_{\tilde{R}}]$ and each experiment yields $T_i = 1200$ samples. During these experiments, no approximate model $\widehat{H^{-1}}_{\text{prior}}$ was used. These references $\tilde{\mathcal{R}}$ are shaped as to produce closed-loop outputs $\mathbf{y}_i = [y_i(0), \dots, y_i(T_i - 1)]^\top$ resembling $\mathbf{r}$, see Figure 4.4. They are *similar* in the sense that the sequences $\mathbf{x}_j$ extracted from $\mathbf{y}_i(t)$ have short Euclidian distance to $\mathbf{x}_{*,j}$ in $\mathbb{R}^{n_\theta}$, see Sections 3.4.1 and 3.6.2, with $n_c = n_{ac} = 300$ such that $n_\theta = 601$.

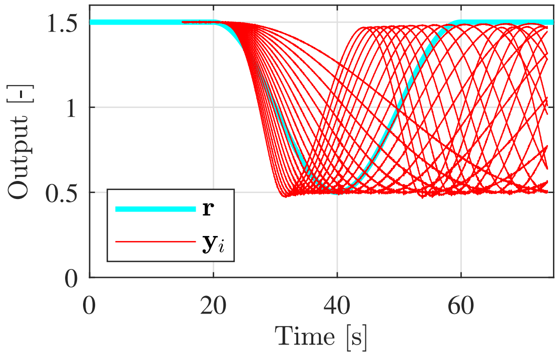


Figure 4.4: Reference to be tracked, compared to observed outputs. Note that the outputs are shifted in time in the figure to show that indeed, these may allow for inference to the reference in $\mathbb{R}^{n_\theta}$.

The measurements are formatted using (3.2) to obtain X and $\mathbf{u}$. To reduce the size of the dataset, only every one in $n_s = 40$ rows of X and $\mathbf{u}$ is taken into account. The resulting data-set $\mathcal{D} = \{X, \mathbf{u}\}$ contains $N = 300$ observations (or rows).

Gaussian Process (3.7) is then trained with $\mathcal{D}$ and SE kernel (3.16), and hyper-parameters σ_f, σ_n and ℓ_i in $\Lambda = \text{diag}([\ell_1, \dots, \ell_{n_\theta}])$ are found by maximizing the marginal likelihood, as described in Section 3.5. This required roughly 6 minutes of training. The computational time required for the synthesis of the feedforward signal with (3.11) is in the order of seconds, since the matrix $(K + \sigma_n^2 I)^{-1}$ is stored after training, i.e., no matrix inversion is required for predictions.

Appendix F includes the MATLAB code of this example.

Results

The resulting feedforward signal is shown in Figure 4.5. The feedforward signal produced by KIMCON resembles the perfect feedforward signal obtained by stable inversion of $P(z)$. These feedforward signals are applied in closed-loop with reference $\mathbf{r}$ and the resulting error is shown in Figure 4.6, magnified in Figure 4.7. The feedforward signal produced by KIMCON leads to a similar tracking error as the perfect inverse model feedforward signal in green. This shows that with minimal expert knowledge, i.e., only a feedback controller tuned with knowledge of the linear system, KIMCON learned the non-linear structure of H^{-1} from data.

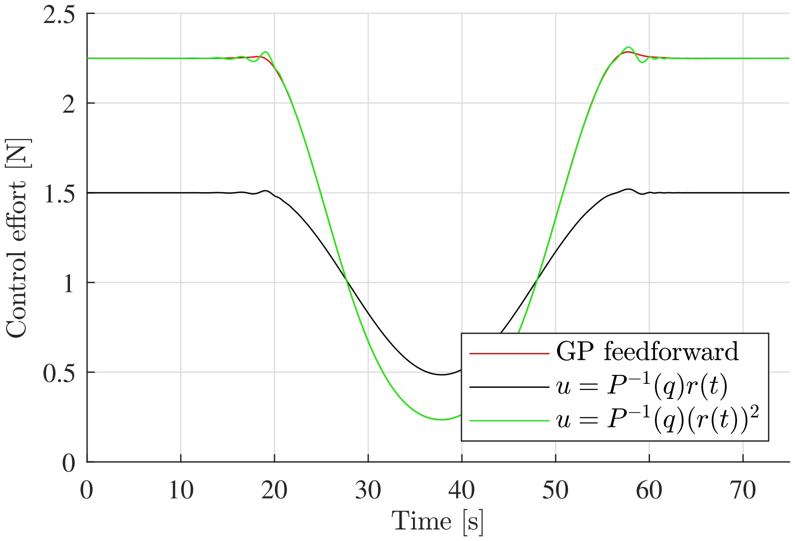


Figure 4.5: Feedforward signals generated by different techniques. For the black and green lines, stable inversion of the true system was used. The red feedforward signal produced by KIMCON closely resembles the true inverse system in green.

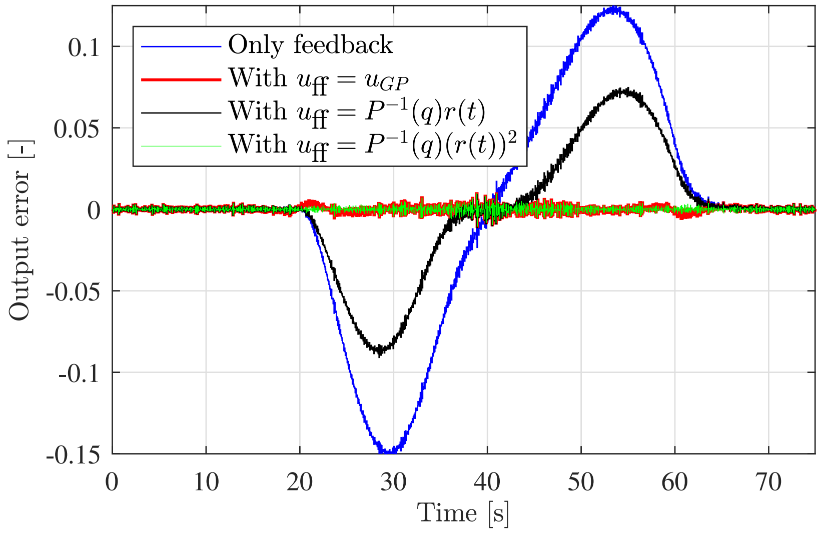


Figure 4.6: Closed-loop tracking error for different feedforward signals. The red error profile resulting from KIMCON is almost as small as the green error profile resulting from stable inversion of the real non-linear system.

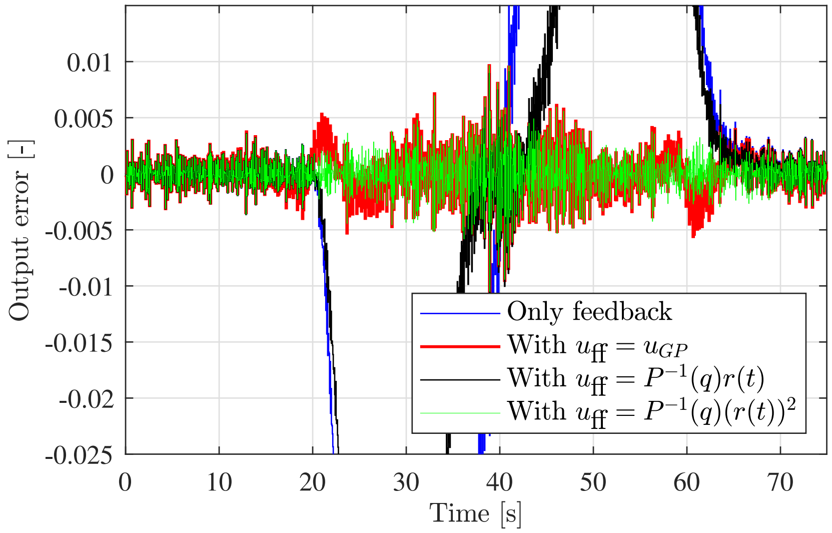


Figure 4.7: Closed-loop tracking error for different feedforward signals, zoomed in. The red error profile resulting from KIMCON is almost as small as the green error profile resulting from stable inversion of the real non-linear system.

4.2 Feedforward control of a non-linear wire-bonder system

In this section, the presented KIMCON approach is applied to a simulation of a wire-bonder system from ASM-PT [2]. A setting is assumed in which a feedback controller $C(z)$ and a linear approximation $\widehat{H^{-1}}_{\text{prior}}$ are available. The non-linear dynamics of H^{-1} are learned using KIMCON.

Setting and goal

The system H to be controlled is a 6-DOF mass system with reaction forces and high order dynamics, simulated by a proprietary Simscape Multibody [14] model. Details of the dynamics are omitted for confidentiality reasons, however, these are not necessary to understand the performance increase that KIMCON accomplishes.

The only knowledge of H^{-1} assumed in this section is that the dynamics are non-linear and smooth. The available simulation models of H, $\widehat{H^{-1}}_{\text{prior}}$ and C are treated as black-box models to show that it is sufficient for KIMCON to have some prior control scheme, even if no details of the control scheme are available.

The goal is to let H track some reference $\mathbf{r}$, depicted in Figure 4.8. With the existing model $\widehat{H^{-1}}_{\text{prior}}$, a feedforward signal $\mathbf{u}_{\text{ff}}$ is constructed, leading to some closed-loop error $\|\mathbf{e}\|_{2,\text{nominal}}$[1]. The goal is to reduce the norm of this error as much as possible with KIMCON.

Approach

With an initial model $\widehat{H^{-1}}_{\text{prior}}$ available, as well as a feedback controller, the focus is on the choice of kernel function and excitation reference design.

The kernel function is chosen as the squared-exponential kernel (3.16) since $f(\mathbf{x}_t)$ is assumed to be smooth (i.e., it does not contain non-smooth terms such as a sign-function). For the data collection, $n_{\tilde{R}} = 7$ excitation references are defined:

$$\tilde{\mathbf{r}}_i = a_i \mathbf{r}, \quad a_i \in [0.985, 0.990, \ldots, 1.015],\ i \in [1, \ldots, n_{\tilde{R}}]. \tag{4.9}$$

The resulting observations are shown in Figure 4.8. Gaussian Process (3.7) is trained with the data-set $\mathcal{D}$ resulting from these 7 experiments as described in Section 3.4, with $n_{ac} = 200$ and $n_c = 15$. Only every second sample in $\mathcal{D}$ was taken into to reduce the size of the data-set (i.e. $n_s = 2$), resulting in $N = 4851$. The kernel hyper-parameters σ_n, σ_f and ℓ_j, $j \in [1, \ldots, n_\theta]$ are obtained by active-set optimization of the marginal likelihood with $\overline{N} = 2000$ the size of the active set, see Section 3.5. This required roughly 36 hours of computational time.

Results

The posterior mean (3.11) of the feedforward signal is compared to the original feedforward signal in Figure 4.9. The oscillations of the feedforward signal produced by KIMCON, visible in the magnified box, span exactly the preview window of $n_{ac} = 200$ samples, functioning as pre-actuation. When applied to the closed-loop system, the effect of this pre-actuation is clearly visible, see Figure 4.10. Although the pre-actuation signal leads to a nonzero error before the reference starts (at $t = 300$), the error is reduced significantly until $t = 420$, when the reference has reached its peak value. The same effect is observed when the system moves back to its initial position.

The non-zero error between $t = 0$ and $t = 120$ results from the fact that the GP has not inferred from the data that standing still requires zero feedforward. This could be resolved by including more observations around $\mathbf{x}_* = \mathbf{0}$ in $\mathcal{D}$, e.g., by applying white noise to the system when it is at $y = 0$.

[1] Exact numbers have been redacted for confidentiality reasons, all values in this section are normalized.

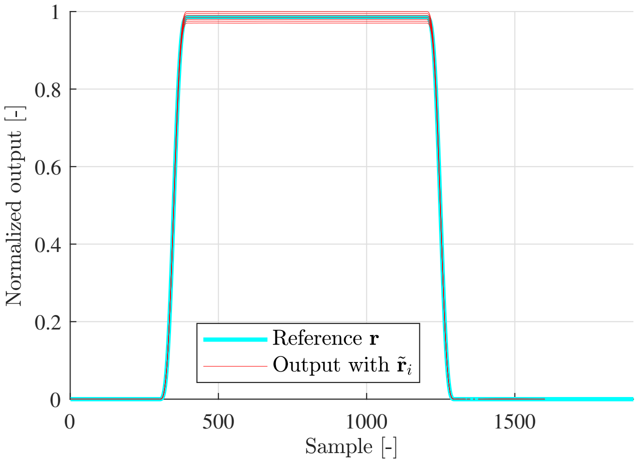


Figure 4.8: Original reference $\mathbf{r}$ and observed closed-loop output of the wire-bonder using excitation references $\tilde{\mathbf{r}}_i$.

With the GP-based feedforward signal, the norm of the closed-loop error $\|e\|_{2,\text{GP}}$ is 9% lower than $\|e\|_{2,\text{nominal}}$. Given that only seven simple experiments were carried out for data collection and a general-purpose SE kernel was used without the assumption of any additional expert knowledge, this is a considerable improvement.

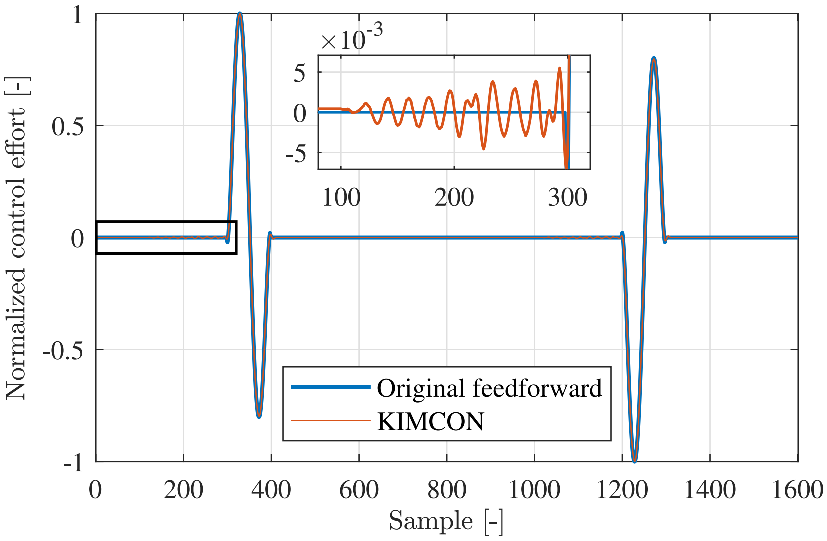


Figure 4.9: Posterior mean of the feedforward signal compared to the original feedforward signal. The black box on the left marks the zoomed in region on top, showing the pre-actuation behavior of KIMCON.

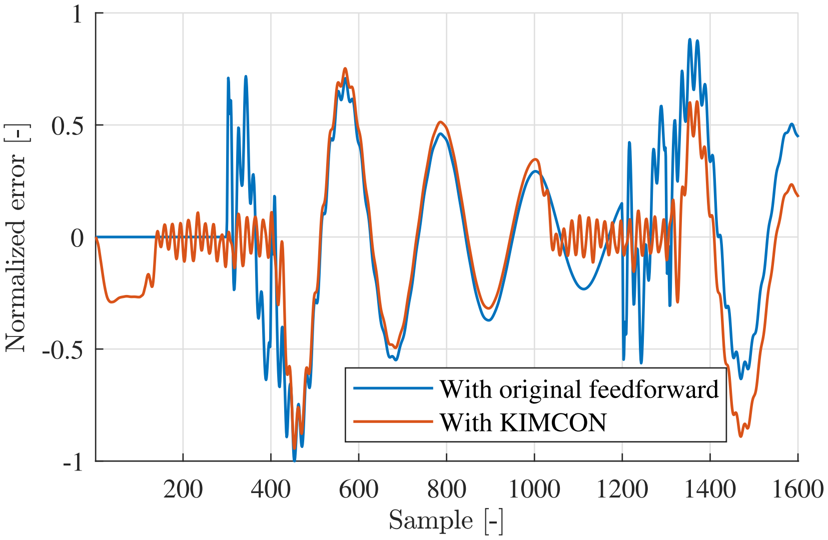


Figure 4.10: Closed-loop tracking error using the original feedforward signal and the KIMCON feedforward signal. The latter leads to an 9% reduction in the 2-norm of the error, as a consequence of the pre-actuation behaviour learned by KIMCON.

Chapter 5

Experimental results

In this chapter, KIMCON is applied to a commercial desktop A3 printer subject to non-linear friction, constituting contribution C2. It is shown that if the non-linear dynamics are unknown, the proposed approach leads to better tracking performance for flexible tasks than linear methods.

5.1 Setting and goal

An A3 printer (see Figure 5.1) is set up with a real-time platform such that control effort $u(t)$ in [V] can be applied and position $y(t)$ in [m] can be read at a sample rate of 1000 Hz. First, the control objective is formulated. Subsequently, the prior knowledge is listed and a non-linear model for the purpose of validation is given.

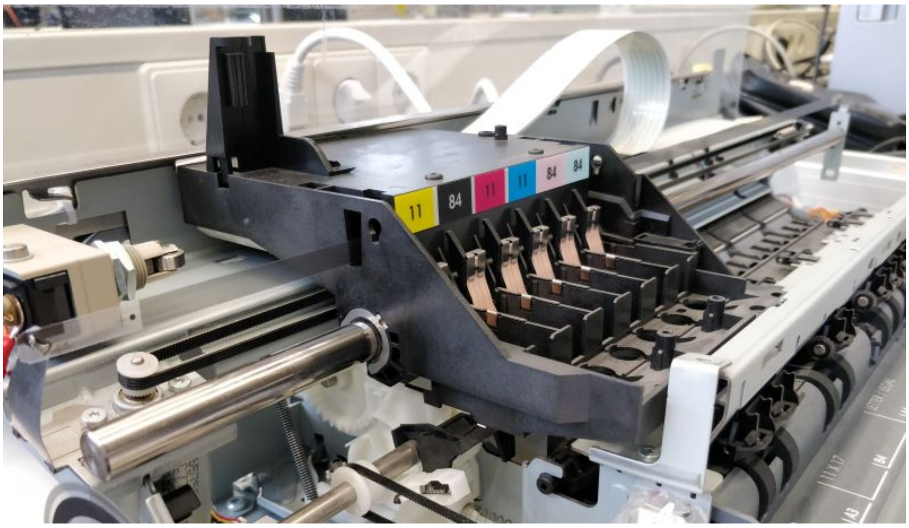


Figure 5.1: The A3 printer used in this experiment.

Goal

The aim is to track two third-order references $\mathcal{R} = \{\mathbf{r}_1, \mathbf{r}_2\}$, depicted in Figure 5.3, using the closed-loop control scheme in Figure 2.2. More specifically, the goal is to reduce the tracking error $\|\mathbf{e}_i\|_2,\ i \in [1, 2]$ of $\mathbf{r}_1$ and $\mathbf{r}_2$ as much as possible using KIMCON.

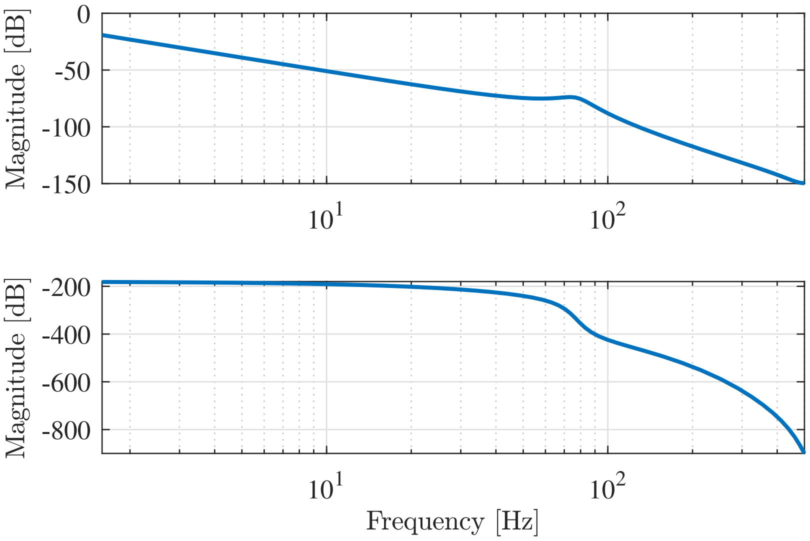


Figure 5.2: Bode plot of $\hat{H}(z)$.

Prior knowledge

The available prior knowledge is as follows. An approximate linear model of the printer H is available:

$$\hat{H}(z) = 10^{-8}\frac{3.399z^4 + 106.2z^3 + 142.2z^2 - 8.806z - 1.882}{z^7 - 3.666z^6 + 5.218z^5 - 3.438z^4 + 0.8857z^3}, \tag{5.1}$$

shown visually in Figure 5.2. A feedback controller $C(z)$ is available as well, given by

$$C(z) = \frac{108.6z^3 + 112.9z^2 - 100z - 104.3}{z^3 - 0.6499z^2 - 0.9465z + 0.7035}. \tag{5.2}$$

Moreover, a prior model $\widehat{H^{-1}}_{\mathrm{a,v}}$ is given:

$$\begin{aligned}\widehat{H^{-1}}_{\mathrm{a,v}} : \mathbf{u}_{\mathrm{ff}} &= \Psi_1(\mathbf{r})v \\ &= [\ddot{\mathbf{r}}, \dot{\mathbf{r}}]\begin{bmatrix} v_1 \\ v_2 \end{bmatrix}.\end{aligned} \tag{5.3}$$

Coefficients $v_1 = 0.0830$ and $v_2 = 2.8531$ corresponding to the basis functions in Ψ_1 are learned by 10 trials of BFILC using reference $\mathbf{r}_1$ and weights $W_e = I,\ W_f = 0$ and $W_{\Delta f} = 0$, see Appendix B. Note that this prior model includes no information about non-linear dynamics.

Validation

In reality, the print-head is subject a significant amount of (location-dependent) static friction. These non-linear dynamics are considered to be *unknown* during training. For the sole purpose of validation, however, a non-linear model $\widehat{H^{-1}}_{\mathrm{a,v,sign}}$ that approximates these non-linear dynamics is taken into account:

$$\begin{aligned}\widehat{H^{-1}}_{\mathrm{a,v,sign}} : \mathbf{u}_{\mathrm{ff}} &= \Psi_2(\mathbf{r})\overline{v} \\ &= [\ddot{\mathbf{r}}, \dot{\mathbf{r}}, \mathrm{sign}(\dot{\mathbf{r}})]\begin{bmatrix} \overline{v}_1 \\ \overline{v}_2 \\ \overline{v}_3 \end{bmatrix},\end{aligned} \tag{5.4}$$

where coefficients $\overline{v}_1 = 0.0955$, $\overline{v}_2 = 0.5643$ and $\overline{v}_3 = 1.0574$ are learned using BFILC with the same settings as before. Note that $\overline{v}_1 \neq v_1, \overline{v}_2 \neq v_2$ because the basis functions in Ψ_2 are not orthogonal, see Appendix B. Indeed, as will be shown in Section 5.3, the inclusion of this non-linear basis function, if known to the user, would lead to a factor 7 reduction in $\|\mathbf{e}_1\|_2$ with respect to the linear model $\widehat{H^{-1}}_{\text{a,v}}$. This confirms that H^{-1} is non-linear to a significant extent.

5.2 Approach

Following the approach in Chapter 3, a system parametrization is specified first. The model $\widehat{H^{-1}}$ is parametrized in terms of the data using a stationary covariance function. In particular, the Matèrn$_{3/2}$ kernel (3.18) is chosen, for its ability to represent *non-smooth* functions, which suits the static friction observed in the system.

Next, a data-set is to be collected using the prior knowledge available. To represent a situation in which no information of the non-linear dynamcs is available, the linear inverse model (5.3) is used for training, with feedback controller $C(z)$ in the setting of Figure 2.2. The excitation references are simply chosen as

$$\tilde{\mathcal{R}} : \tilde{\mathbf{r}}_i = a_i \mathbf{r_1}, \;\; i \in [0.90, 0.92, \ldots, 1.10]. \tag{5.5}$$

Note that this leads to $n_{\tilde{\mathcal{R}}} = 11$ experiments, one of which with $\tilde{\mathbf{r}}_6 = \mathbf{r}_1 \in \mathcal{R}$, i.e., the reference $\mathbf{r}_1$ to be tracked is used during training. However, $\mathbf{r}_2 \in \mathcal{R}$ is not used during training and thus it can be used to assess KIMCON's ability to deal with task flexibility. The references $\mathbf{r}_1$ and $\mathbf{r}_2$ are compared with the observed closed-loop output in Figure 5.3.

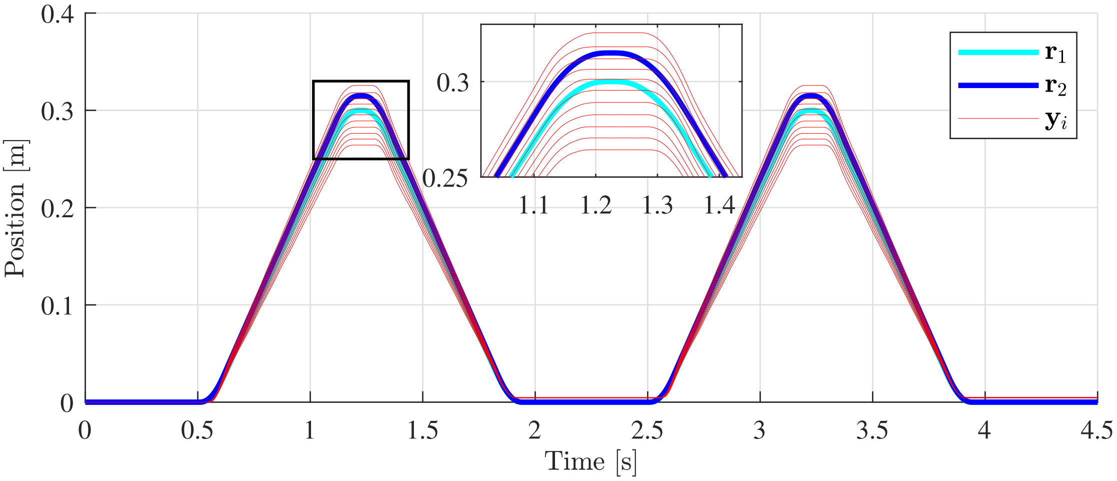


Figure 5.3: References $\mathbf{r}_1, \mathbf{r}_2 \in \mathcal{R}$ and observed outputs $\mathbf{y}_i \in \mathcal{D}$ resulting from $n_{\tilde{\mathcal{R}}} = 11$ closed-loop experiments with excitation references $\tilde{\mathcal{R}}$, see (5.5).

With these observations of $\mathbf{y}$ and $\mathbf{u}$, data-set $\mathcal{D} = \{X, \mathbf{u}\}$ is formed with $n_{ac} = 40$ and $n_c = 20$ as described in Section 3.4. To reduce the size of the data-set, only every one in $n_s = 30$ rows of X and $\mathbf{u}$ is taken into account, leaving $N = 2970$ samples (or rows). Subsequently, Gaussian Process (3.7) is trained with $\mathcal{D}$ and kernel (3.18), after which the kernel hyper-parameters are optimized by active-set maximization of the marginal likelihood (see Section 3.5) with active-set size $\overline{N} = 2000$. This required roughly 90 minutes of computational time.

5.3 Tracking performance

The feedforward signals produced by the three inverse models are compared in Figure 5.4. The feedforward signal generated by KIMCON is considerably different from $\widehat{H^{-1}}_{\text{a,v}}\mathbf{r}_1$, even though

KIMCON used $\widehat{H^{-1}}_{\text{a,v}}$ for data-collection. This indicates that KIMCON has at least learned dynamics not present in $\widehat{H^{-1}}_{\text{a,v}}$, possibly non-linear dynamics.

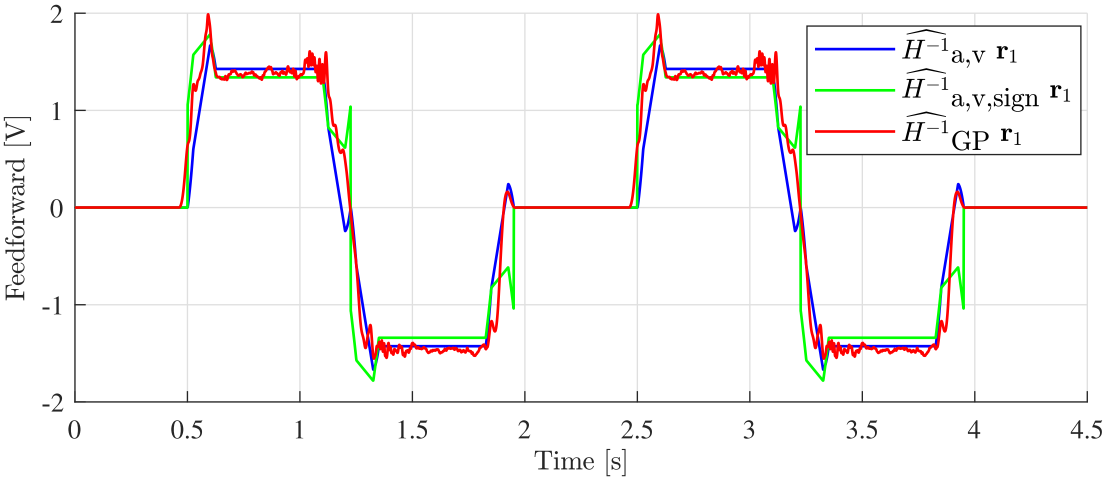


Figure 5.4: Feedforward signals generated by BFILC witout a sign-term ($\widehat{H^{-1}}_{\text{a,v}}$), with a sign term ($\widehat{H^{-1}}_{\text{a,v,sign}}$) and by KIMCON ($\widehat{H^{-1}}_{\text{GP}}$), for reference $\mathbf{r}_1 \in \tilde{\mathcal{R}}$, i.e., a reference that is used during training.

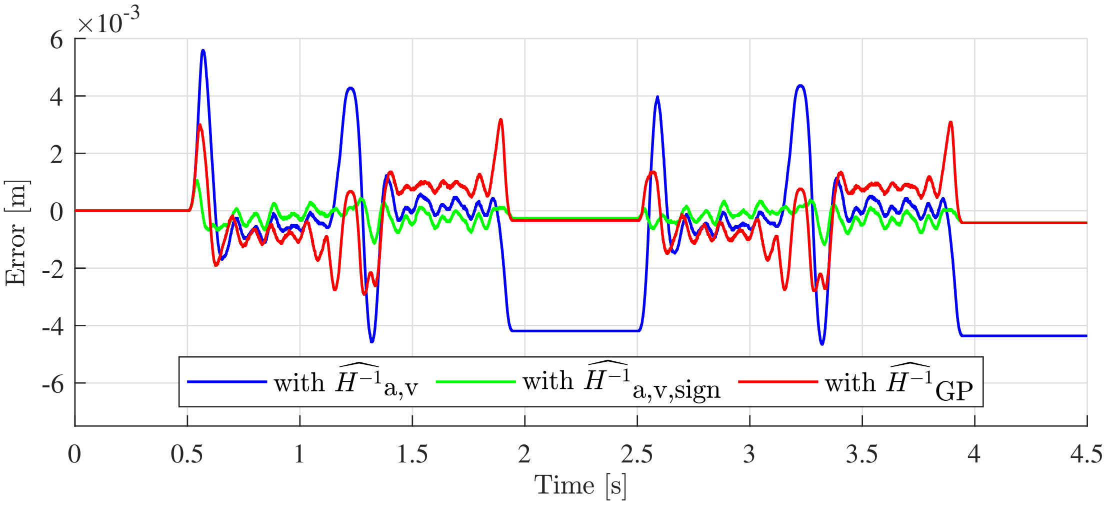


Figure 5.5: Closed-loop error $\mathbf{e}_1 = \mathbf{r}_1 - \mathbf{y}_1$ using feedforward from BFILC witout a sign-term ($\widehat{H^{-1}}_{\text{a,v}}$), with a sign term ($\widehat{H^{-1}}_{\text{a,v,sign}}$) and from KIMCON ($\widehat{H^{-1}}_{\text{GP}}$). While expert knowledge of the sign term leads to the smallest error, KIMCON is superior if this prior is unavailable, with a factor 2.5 reduction in $\|\mathbf{e}_1\|_2$ compared to $\widehat{H^{-1}}_{\text{a,v}}$.

The resulting closed-loop error for $\mathbf{r}_1$ is shown in Figure 5.5. The effect of the feedforward signal produced by KIMCON is most visible at its resting position at $t = 2$ [s]. Whereas with $\widehat{H^{-1}}_{\text{a,v}}$ the print-head stops 4 [mm] off from its desired position since $C(z)$ contains no integrator, both KIMCON and BFILC with the sign-term stop close to zero. Since the only difference between the blue and the green plot in Figure 5.4 is the sign-term, the correct resting position is presumably the result of the spike in control effort just before $t = 2$ [s]. The need for such a spike in force, although applied slightly earlier (around $t = 1.9$ [s]), is learned from input-output data by KIMCON.

Table 5.1 displays the 2-norm and peak of the tracking errors of the three approaches. While the best performance is achieved with knowledge of the non-linear sign term, KIMCON far outperforms BFILC if the non-linear basis function is unknown by modeling the non-linear dynamics from data. With only $n_{\tilde{R}} = 11$ simple experiments and a general-purpose kernel, KIMCON leads to a factor 2.5 reduction in the 2-norm of the error.

5.4 Tracking performance with flexible tasks

Lastly, the ability of KIMCON to deal with task flexibility is demonstrated. To this end, the same GP, trained using $\tilde{\mathcal{R}}$, is used to compute a feedforward signal for $\mathbf{r}_2 = 1.05\mathbf{r}_1 \notin \tilde{\mathcal{R}}$, see Figure 5.6. The resulting error profile is shown in Figure 5.7, where similar results are seen as before: while knowledge of the sign-term leads to the best performance, KIMCON outperforms BFILC when the non-linear component of H^{-1} is unknown, even when extrapolating to a reference not used in training. The 2-norm and peak of the tracking errors are shown in Table 5.1. With a factor 1.9 reduction in the 2-norm compared to BFILC without knowledge of the non-linear sign-term, this confirms that KIMCON can be well suited to produce feedforward signals for flexible tasks.

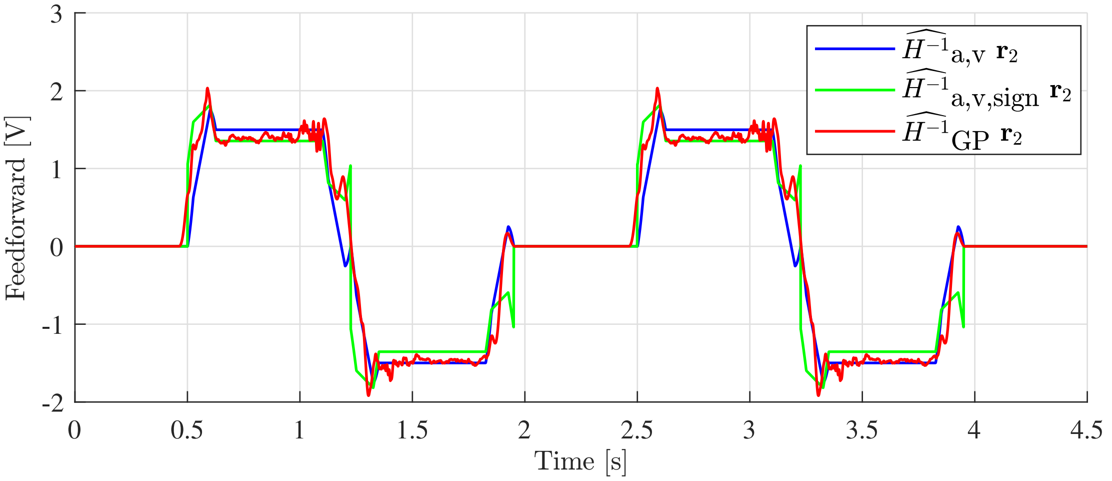


Figure 5.6: Feedforward signals generated by BFILC witout a sign-term ($\widehat{H^{-1}}_{\mathrm{a,v}}$), with a sign term ($\widehat{H^{-1}}_{\mathrm{a,v,sign}}$) and by KIMCON ($\widehat{H^{-1}}_{\mathrm{GP}}$), for reference $\mathbf{r}_2 \notin \tilde{\mathcal{R}}$, i.e., $\mathbf{r}_2$ is not used during training.

Table 5.1: Closed-loop tracking errors using different feedforward signals.

	With reference $\mathbf{r}_1 \in \tilde{\mathcal{R}}$	With flexible reference $\mathbf{r}_2 \notin \tilde{\mathcal{R}}$
BFILC: $\|\mathbf{e}_{\mathrm{a,v}}\|_2$	170 [mm]	164 [mm]
BFILC: $\|\mathbf{e}_{\mathrm{a,v,sign}}\|_2$	23 [mm]	46 [mm]
KIMCON: $\|\mathbf{e}_{\mathrm{GP}}\|_2$	68 [mm]	86 [mm]
BFILC: $\max(\lvert\mathbf{e}_{\mathrm{a,v}}\rvert)$	5.6 [mm]	4.2 [mm]
BFILC: $\max(\lvert\mathbf{e}_{\mathrm{a,v,sign}}\rvert)$	1.1 [mm]	1.2 [mm]
KIMCON: $\max(\lvert\mathbf{e}_{\mathrm{GP}}\rvert)$	3.2 [mm]	3.6 [mm]

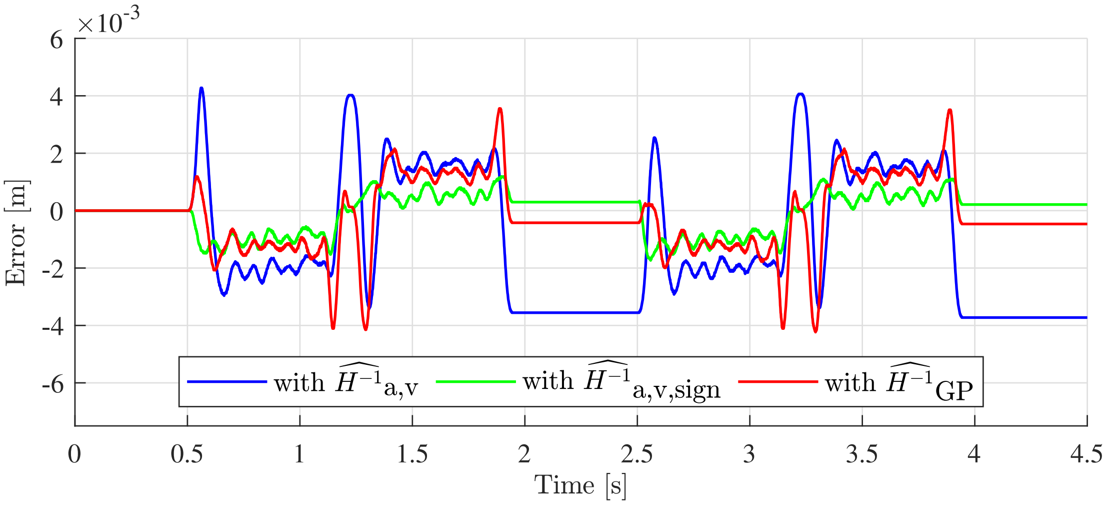


Figure 5.7: Closed-loop error $\mathbf{e}_2 = \mathbf{r}_2 - \mathbf{y}_2$ using feedforward from BFILC witout a sign-term ($\widehat{H^{-1}}_{\text{a,v}}$), with a sign term ($\widehat{H^{-1}}_{\text{a,v,sign}}$) and from KIMCON ($\widehat{H^{-1}}_{\text{GP}}$), for a reference not in the training data. While expert knowledge of the sign term leads to the smallest error, KIMCON is superior if this prior is unavailable, with a factor 1.9 reduction in $\|\mathbf{e}_2\|_2$. Evidently, KIMCON leads to relatively small tracking errors even when extrapolating from the training data.

Chapter 6

Conclusions

Conclusions

With the developed Kernel-based Inverse Model Control for Non-linear systems (KIMCON) approach, it has become possible to produce feedforward signals for systems with non-linear dynamics of unknown structure while allowing for task flexibility. By modeling the inverse system as a Gaussian Process, parametrized as a non-causal non-linear finite impulse response (NFIR) system, non-linear dynamics of unknown structure are learned from input-output data. It was shown numerically and experimentally that by selection of some suitable stationary kernel function and experimental design, feedforward signals are obtained that significantly improve the performance of systems.

Since the technique relies on inference of observed output sequences to reference sequences, the main applications for this technique are systems with non-linear dynamics of unknown structure for which at least some prior control strategy is available, e.g., a feedback controller or a decent feedforward controller. This renders it feasible to design the necessary experiments that lead to observations allowing for inference to different references. Examples covered in this thesis are Wiener systems of which only the linear part is known, non-linear systems of which only a simplified linear model is available, and systems experiencing location-dependent friction. In each of these case studies, KIMCON led to a considerable performance increase, with up to a factor 2.5 reduction in the 2-norm of the error on an experimental setup compared to linear methods.

Future work

Several directions for future work are proposed. First, it would be interesting to see how the performance could be improved if the Gaussian Process is not trained with the raw output y, but with filtered outputs. For example, for systems where the non-linearity presents itself in the nth derivative of y, the approximate nth derivative of r and measurements y could be used for training. Other filters used for this purpose are Laguerre and Kautz filters [29, 30], the application of which to KIMCON could be worth researching.

Moreover, the presented KIMCON approach could be extended to feedforward control of multi-input multi-output (MIMO) systems. This is done for causal state-transition models in [6] by viewing each output dimension as a conditionally independent MISO system and modeling each of these as a Gaussian Process. A similar approach could be taken to extend KIMCON to MIMO systems.

Bibliography

[1] N. Aronszajn. Theory of Reproducing Kernels. *Transactions of the American Mathematical Society*, 68(3):337, May 1950. 49

[2] ASM-PT. ASM Pacific Technology website. 26

[3] Thomas Beckers, Dana Kulić, and Sandra Hirche. Stable Gaussian process based tracking control of Euler–Lagrange systems. *Automatica*, 103:390–397, May 2019. 5, 6

[4] Georgios Birpoutsoukis. *Volterra Series Estimation in the Presence of Prior Knowledge.* Number January. Uitgeverij University Press, Zelzate, Oost-Vlaanderen, 2018. 49

[5] Lennart Blanken and Tom Oomen. Kernel-based identification of non-causal systems with application to inverse model control. *Automatica*, 114:108830, April 2020. 1, 3, 5, 6, 13, 22, 39, 40, 47, 48

[6] Marc Deisenroth and Carl Rasmussen. PILCO: A Model-Based and Data-Efficient Approach to Policy Search. In *Proceedings of the 28th International Conference on Machine Learning*, pages 465–472, Bellevue, WA, 2011. 1, 5, 6, 35

[7] Matthias O. Franz and Bernhard Schölkopf. A Unifying View of Wiener and Volterra Theory and Polynomial Kernel Regression. *Neural Computation*, 18(12):3097–3118, December 2006. 49

[8] Peter S.C. Heuberger, Paul M.J. Van den Hof, and Bo Wahlberg. *Modelling and Identification with Rational Orthogonal Basis Functions.* Springer London, London, 2005. 41

[9] Ylva Jung and Martin Enqvist. Estimating models of inverse systems. In *52nd IEEE Conference on Decision and Control*, pages 7143–7148. IEEE, December 2013. 1

[10] Juš Kocijan. *Modelling and control of dynamic systems using gaussian process models.* Advances in Industrial Control. Springer International Publishing, Cham, 2016. 1

[11] Ren Jun Li and Zheng Zhi Han. Survey of iterative learning control. *Kongzhi yu Juece/Control and Decision*, 20(9):961–966, September 2005. 5, 6

[12] Haitao Liu, Yew-Soon Ong, Xiaobo Shen, and Jianfei Cai. When Gaussian Process Meets Big Data: A Review of Scalable GPs. July 2018. 19

[13] David J C Mackay. Information Theory, Inference, and Learning Algorithms. *IEEE Transactions on Information Theory*, 50(10):2544–2545, October 2004. 13

[14] Mathworks. Simscape Multibody, 2020. 26

[15] Mathworks. Statistics and Machine Learning Toolbox, 2020. 16

[16] Duy Nguyen-Tuong, Jan Peters, Matthias Seeger, and Bernhard Sc̈holkopf. Learning inverse dynamics: A comparison. In *ESANN 2008 Proceedings, 16th European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning*, pages 13–18, 2008. 5

[17] Duy Nguyen-Tuong, Matthias Seeger, and Jan Peters. Model learning with local Gaussian process regression. In *Advanced Robotics*, volume 23, pages 2015–2034, 2009. 5

[18] Tom Oomen. Advanced motion control for precision mechatronics: Control, identification, and learning of complex systems. *IEEJ Journal of Industry Applications*, 7(2):127–140, 2018. 1

[19] Panos Y. Papalambros and Douglass J Wilde. *Principles of Optimal Design.* Cambridge University Press, January 2017. 16

[20] G. Pillonetto, M. H. Quang, and A. Chiuso. A new kernel-based approach for nonlinear system identification. *IEEE Transactions on Automatic Control*, 56(12):2825–2840, 2011. 1, 5, 6, 49

[21] Gianluigi Pillonetto, Francesco Dinuzzo, Tianshi Chen, Giuseppe De Nicolao, and Lennart Ljung. Kernel methods in system identification, machine learning and function estimation: A survey. *Automatica*, 50(3):657–682, 2014. 40

[22] C.E. Rasmussen and C.K.I. Williams. *Gaussian processes for machine learning.* London, England, 2006. 1, 7, 8, 9, 10, 16, 19, 44, 46, 49, 52

[23] J. Schoukens and L. Ljung. Nonlinear system identification: A user-oriented road map. *IEEE Control Systems Magazine*, 39(6):28–99, 2019. 1, 18

[24] Gregory Shakhnarovich, Trevor Darrel, and Piotr Indyk. *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice (Neural Information Processing).* The MIT Press, Cambridge, MA, 2006. 17

[25] Jonas Sjöberg, Qinghua Zhang, Lennart Ljung, Albert Benveniste, Bernard Delyon, Pierre Yves Glorennec, Håkan Hjalmarsson, and Anatoli Juditsky. Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31(12):1691–1724, 1995. 1

[26] Edward Snelson and Zoubin Ghahramani. Sparse Gaussian Processes using Pseudo-inputs. In Y Weiss, B Schölkopf, and J Platt, editors, *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2006. 19

[27] J. Van De Wijdeven and O. H. Bosgra. Using basis functions in iterative learning control: Analysis and design theory. *International Journal of Control*, 83(4):661–675, April 2010. 1, 5, 6, 41, 42

[28] Jurgen van Zundert and Tom Oomen. On inversion-based approaches for feedforward and ILC. *Mechatronics*, 50:282–291, April 2018. 1

[29] Bo Wahlberg. System Identification Using Laguerre Models. *IEEE Transactions on Automatic Control*, 36(5):551–562, 1991. 35

[30] Bo Wahlberg. System Identification Using Kautz Models. *IEEE Transactions on Automatic Control*, 39(6):1276–1282, 1994. 35

[31] Weidong Zhang, Fanming Zeng, Guojun Cheng, and Shengguang Gong. Feedforward-feedback combined control system based on neural network. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3174:110–116, 2004. 1

Appendix A

Kernel-based regularization

The KIMCON approach presented in Chapter 3 is closely related to *kernel-based identification of non-causal systems* [5]. This technique will be explained next.

A.1 System parametrization

Consider the linear discrete-time single-input single-output (SISO) non-minimum phase plant $H(z)$. The inverse system $H^{-1}(z)$ can be exactly expressed as the infinite two-sided formal Laurent series

$$H^{-1}(z) = \sum_{\tau=-\infty}^{\infty} \theta_\tau^o z^{-\tau}, \tag{A.1}$$

which is of non-causal infinite impulse response (IIR) structure. It was shown in [5] that the signal

$$u(t) = \sum_{\tau=-\infty}^{\infty} \theta_\tau^o r_{t-\tau} \tag{A.2}$$

leads to exact inversion $y(t) = H(z)u(t) = r(t)$, i.e., open-loop inverse model feedforward. Alternatively, consider the finite-dimensional truncation of (A.1):

$$u(t) = \sum_{\tau=-n_{ac}}^{n_c} \hat{\theta}_\tau r_{t-\tau}. \tag{A.3}$$

Here, n_{ac} represents the number of preview samples, and n_c the number of delayed samples. This parametrization of H^{-1} is a finite impulse response (FIR) structure. The next section explains how to obtain $\hat{\theta}$.

A.2 System identification of non-causal systems

In [5], $\hat{\theta}$ is obtained from data as follows. Given a data-set of input-output data $\mathcal{D} = \{u_t, y_t\}_{t=1}^N$, the following optimization problem is considered:

$$\hat{\theta} = \arg\min_{\theta\in\mathcal{H}} \sum_{t=1}^{N} \left(u_t - \sum_{\tau=-\infty}^{\infty} \theta_\tau y_{t-\tau} \right)^2 + \sigma_n^2 \|\theta\|_{\mathcal{H}}^2. \tag{A.4}$$

Here, $\theta \in \mathcal{H}$ indicates that the impulse response coefficients are restricted to a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$, see Definition 3.1.1. This restriction allows for a finite-dimensional reformulation of (A.4) [21, Section 10.1], by imposing the FIR structure (A.3) on H^{-1}. The optimization problem then becomes

$$\begin{aligned}\hat{\theta} &= \arg\min_{\theta} \|\mathbf{u} - \Phi_N \theta\|^2 + \sigma_n^2 \theta^T \Pi^{-1} \theta \\ &= \Pi \Phi_N^T \left(\Phi_N \Pi \Phi_N^T + \sigma_n^2 I\right)^{-1} \mathbf{u},\end{aligned} \tag{A.5}$$

where $\theta \in \mathbb{R}^{n_\theta}$ are the coefficients of the function $\mathbf{u} = \Phi_N \theta + E,\ E_i \sim \mathcal{N}\left(0, \sigma_n^2\right)$. The regressor matrix $\Phi_N \in \mathbb{R}^{N \times n_\theta}$ is defined as

$$\Phi_N = \begin{bmatrix} y(n_{ac}) & y(n_{ac}-1) & \dots & y(-n_c) \\ y(n_{ac}+1) & y(n_{ac}) & \dots & y(-n_c+1) \\ \vdots & \vdots & & \vdots \\ y(N-1+n_{ac}) & y(N-2+n_{ac}) & \dots & y(N-n_c-1) \end{bmatrix}, \tag{A.6}$$

where it is assumed that $y(N+i) = y(-i) = 0\ \forall\ i \in \mathbb{N}$. The Gramian $\Pi \in \mathbb{R}^{n_\theta \times n_\theta}$ with $n_\theta = n_{ac} + 1 + n_c$ is obtained from $\Pi_{ij} = k(t_i, t_j')$, where $k(t_i, t_j')$ specifies the covariance between two impulse response coefficients, i.e., a non-causal prior on θ. See [5] for a number of non-causal kernels.

Appendix B

ILC with basis functions

In this appendix, Iterative Learning Control with basis functions (BFILC) [27] is explained, for the purpose of understanding how models of non-linear inverse systems can be obtained if the non-linear structure is known. An application of BFILC is given in Chapter 5.

Setting and goal

Consider the closed-loop setting in Figure 2.2, where H is a non-linear system. All signals in the figure are defined in the same fashion as in Section 2.1. It is assumed that a linear approximation of the process sensitivity is available, defined by

$$J(z) := \frac{y(z)}{\varepsilon(z)} = \frac{H(z)}{1+H(z)C(z)}, \tag{B.1}$$

as well as a linear approximation of the sensitivity:

$$S(z) := \frac{e(z)}{r(z)} = \frac{1}{1+H(z)C(z)}. \tag{B.2}$$

Remark 6. Since H may be non-linear, the linear term $H(z)$ in (B.1) and (B.2) is a linear approximation of H. ■

The inverse system H^{-1} is parametrized by

$$\widehat{H^{-1}} : \mathbf{u}_{\text{ff}} = \Psi(\mathbf{r})v, \tag{B.3}$$

where $\Psi(\mathbf{r}) \in \mathbb{R}^{T\times n_v}$ is a basis containing polynomial functions of the reference $\mathbf{r} \in \mathbb{R}^T$ and possibly non-linear terms if known, and $v \in \mathbb{R}^{n_v}$ are the corresponding coefficients. The aim is to learn coefficients v from data using the model $J(z)$.

Remark 7. If the basis functions $\psi_i(\mathbf{r}) \in \Psi(\mathbf{r})$ are *orthogonal* [8, Section 1.2.5], there is no overlap in $\mathbb{R}^{n_v}$ and consequently, adding a new basis function $\psi_{n_v+1}(\mathbf{r})$ leaves parameters $v_1, \dots, v_{n_v}$ unchanged. ■

Before BFILC is explained, the notation used for the filtering of signals is given. For a dynamic system $A(z) \in \mathbb{R}$ and a signal $\mathbf{b} \in \mathbb{R}^T$, the notation $A\mathbf{b} = \mathbf{c} : \mathbb{R}^T \to \mathbb{R}^T$ denotes the filtering of $\mathbf{b}$ by dynamic system $A(z)$. Here, $A \in \mathbb{R}^{T\times T}$ is the (Toeplitz) impulse response matrix of $A(z)$.

Method

The coefficients v are learned by performing N_B closed-loop experiments, or trials, with reference $\mathbf{r} \in \mathbb{R}^T$. After each trial j, a feedforward signal is computed for trial $j+1$. The method commences by performing a trial at $j = 0$ without feedforward to obtain error $\mathbf{e}_0(v_0) \in \mathbb{R}^T$ with initial parameters v_0, after which it proceeds as follows.

The approximated error at the next trial can be written as

$$\mathbf{e}_{j+1}(v_{j+1}) = S\mathbf{r} - J\Psi(\mathbf{r})v_j. \tag{B.4}$$

The next coefficients v_{j+1} are then obtained by solving the optimization problem

$$v_{j+1} = \arg\min_{v_{j+1}} \mathcal{J}(v_{j+1}), \tag{B.5}$$

with cost

$$\begin{aligned}\mathcal{J}(v_{j+1}) &= \|\mathbf{e}_{j+1}(v_{j+1})\|^2_{W_e} + \|\mathbf{u}_{\text{ff},j+1}\|^2_{W_f} + \|\mathbf{u}_{\text{ff},j+1} - \mathbf{u}_{\text{ff},j}\|^2_{W_{\Delta f}} \\ &= \|\mathbf{e}_j - J\Psi(\mathbf{r})(v_{j+1} - v_j)\|^2_{W_e} + \|\Psi(\mathbf{r})v_{j+1}\|^2_{W_f} + \|\Psi(\mathbf{r})(v_{j+1} - v_j)\|^2_{W_{\Delta f}},\end{aligned} \tag{B.6}$$

with $\|x\|^2_W = x^\top W x$, and using the assumption that $\mathbf{r}_{j+1} = \mathbf{r}_j = \mathbf{r}$, i.e., only one reference is used during learning. The positive semi-definite weighting matrices W_f and $W_{\Delta f}$ are used to punish large control effort values and large changes in feedforward signals over trials respectively. The positive definite weighting matrix W_e assigns a relative cost to the expected error at the next trial.

Since this cost is quadratic in v_{j+1}, a closed-form solution for v_{j+1} can be found, resulting in the update law

$$v_{j+1} = Qv_j + Le_j, \tag{B.7}$$

with

$$\begin{aligned}Q &= \left(\Psi(\mathbf{r})^\top \left(J^\top W_e J + W_f + W_{\Delta f}\right)\Psi(\mathbf{r})\right)^{-1} \Psi(\mathbf{r})^\top \left(J^\top W_e J + W_{\Delta f}\right)\Psi(\mathbf{r}) \\ L &= \left(\Psi(\mathbf{r})^\top \left(J^\top W_e J + W_f + W_{\Delta f}\right)\Psi(\mathbf{r})\right)^{-1} \Psi(\mathbf{r})^\top J^\top W_e.\end{aligned} \tag{B.8}$$

Conditions for convergence and optimal performance are given in [27].

In conclusion, BFILC enables the iterative computation of parameters v in (B.3) describing $\widehat{H^{-1}}$, using update law (B.7).

Appendix C

Smoothness of Gaussian Processes with Matèrn and SE kernels

This Appendix shows the difference between squared exponential (SE) kernels (3.16) and Matèrn$_{3/2}$ kernels (3.18). In particular, it is shown how Matèrn$_{3/2}$ kernels allow for the representation of *non-smooth* functions via (3.15). We commence with an example demonstrating the difference between both kernels visually. Thereafter, it is explained that Matèrn$_{3/2}$ kernels are able to represent non-smooth functions $f(\mathbf{x})$ since they are *finitely mean squared differentiable*, as opposed to SE kernels which are infinitely mean squared differentiable.

C.1 Example: SE kernels vs Matèrn kernels for non-smooth systems

The following example explores the way in which both SE kernels and Matèrn$_{3/2}$ kernels model a non-smooth system H^{-1}, and investigates the influence of the lengthscale ℓ on the variability of the prior.

Example 5. Consider the system

$$H^{-1} : u(t) := f(\mathbf{x}_t) = \operatorname{sign}(y(t)), \tag{C.1}$$

where $\mathbf{x}_t = [y(t)]$, i.e., $n_c = n_{ac} = 0$. The kernel hyper-parameters are chosen as $\sigma_f = 1$ for both kernels (3.16) and (3.18). The lengthscales are chosen as $\ell_{\text{SE}} = \ell_{\text{Matèrn}} = 3$. The covariance of both kernels as a function of distance $\mathbf{x} - \mathbf{x}'$ is shown in Figure C.1. Although the covariance looks similar, it will be shown next that the type of functions they represent differ considerably.

To show this, five random realizations $\mathbf{f}$ are drawn from the prior. Recall from Section 3.2 that the prior of a GP is uniquely defined by the (Gramian) covariance matrix $K(X, X')$ of $k(\mathbf{x}, \mathbf{x}')$:

$$\mathbf{f}_{\text{prior}} \sim \mathcal{N}(0, K(X, X')). \tag{C.2}$$

Random functions are drawn from this prior for a range of values $X = [0, \dots, 20]^\top \in \mathbb{R}^n$ with $n = 4001$ as follows. First, a number of random Gaussian noise sequences are defined as $\boldsymbol{\omega}_i \in \mathbb{R}^n \sim \mathcal{N}(0, 1),\ i \in [1, \dots, 5]$. Then, realizations $\mathbf{f} \in \mathbb{R}^n$ are constructed to have covariance matrix $K + \sigma_n^2$ using

$$\begin{aligned} \mathbf{f}_{\text{prior,SE},i} &= \boldsymbol{\omega}_i^\top \ \operatorname{chol}(K_{\text{SE}} + \sigma_n^2 I), \\ \mathbf{f}_{\text{prior,Matèrn},i} &= \boldsymbol{\omega}_i^\top \ \operatorname{chol}(K_{\text{Matèrn}} + \sigma_n^2 I), \end{aligned} \tag{C.3}$$

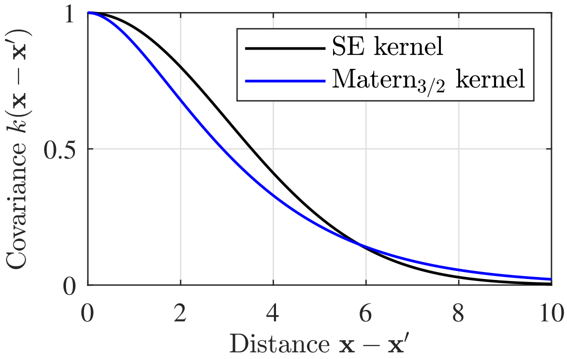
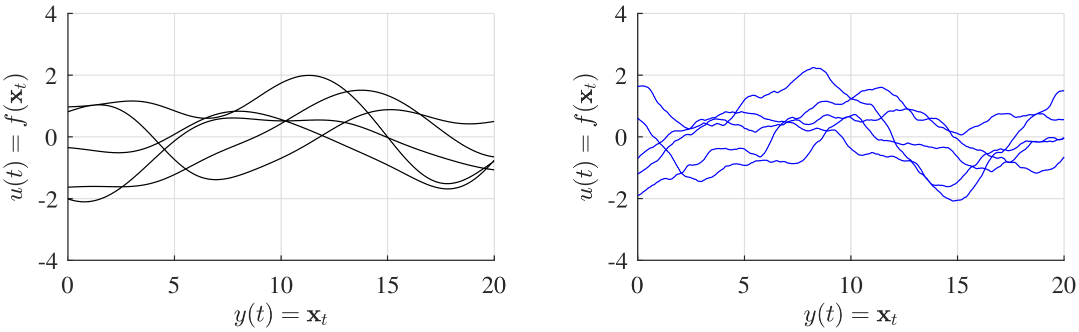


Figure C.1: Covariance of the SE kernel and Matèrn$_{3/2}$ kernel, with $\ell_{\text{SE}} = \ell_{\text{Matèrn}} = 3$.



(a) Using the SE kernel with $\ell_{\text{SE}} = 3$

(b) Using the Matèrn$_{3/2}$ kernel with $\ell_{\text{Matèrn}} = 3$

Figure C.2: Random functions drawn from the prior of both GPs, using the same noise realizations. It is clearly visible that the prior defined by the Matèrn$_{3/2}$ kernel admits much sharper variations in $u(t) = f(\mathbf{x}_t)$ and thus is better able to represent non-smooth systems H^{-1}.

with $\sigma_n = 10^{-3}$ and chol(K_n) the Cholesky factorization L of K_n such that $LL^\top = K_n$. The $\sigma_n^2 I$ term is added to the covariance matrix for numerical stability, as explained in [22, Appendix A.3]. These realizations $\mathbf{f}$ drawn from the prior are shown in Figure C.2. Clearly, the Matèrn$_{3/2}$ kernel allows for 'rougher' variations in $u(t) = f(\mathbf{x}_t)$ as a function of $\mathbf{x}_t$.

It is stressed that the realizations $\mathbf{f}$ of the *prior*, as computed in (C.3), are not based on any observations of $\mathbf{x}_t$. Realizations $\mathbf{f}$ of the prior thus are unrelated to the posterior in (3.11) or (3.15). It will be shown next how the different prior distributions lead to different posterior distributions.

Five noisy observations are available at locations $X = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\top$,

$$X = [-7.5, -1, 0, 1, 7.5]^\top + \varepsilon, \tag{C.4}$$

with $\varepsilon \in \mathbb{R}^N \sim \mathcal{N}(0, \sigma_n^2)$. The posterior mean (3.11) of both GPs with these observations is shown in Figure C.3. The real function (C.1) is highly non-smooth because of the sign-term, i.e., it contains sudden variations in $f(\mathbf{x})$. It was observed before that the Matèrn$_{3/2}$ kernel leads to priors with sharper variations in $f(\mathbf{x})$ and indeed, from this posterior mean it becomes clear that the posterior mean with the Matèrn$_{3/2}$ kernel closely resembles the real function near the data. The SE kernel, on the other hand, leads to a posterior mean that is too smooth to represent the non-smooth system (3.16).

One may wonder if the SE kernel would allow for rougher variations in $f(\mathbf{x})$ if it is defined with a shorter lengthscale. With lengthscale $\ell_{\text{SE}} = 2$, it approximately matches the Matèrn$_{3/2}$ covariance for short distances, see Figure C.4.

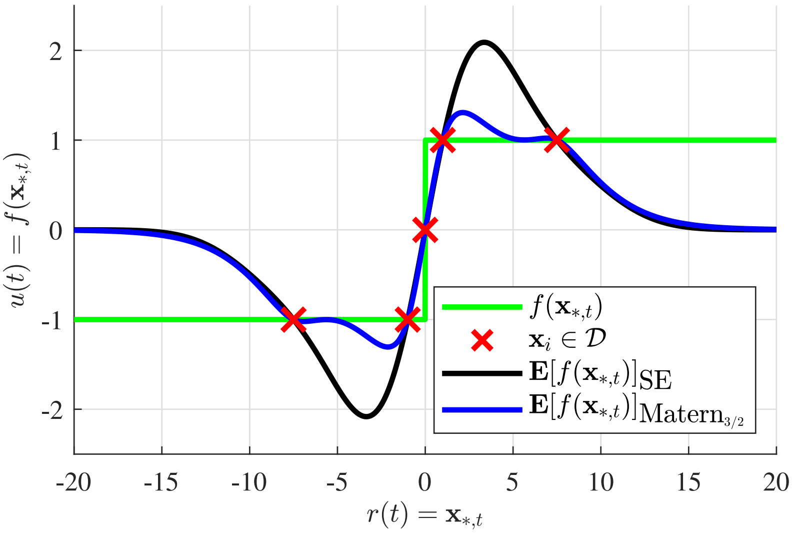


Figure C.3: Posterior mean of two GPs, defined with an SE kernel and a Matèrn$_{3/2}$ kernel with $\ell_{\mathrm{SE}} = \ell_{\mathrm{Matèrn}} = 3$, along with the real function in green and observations marked by crosses. The Matèrn$_{3/2}$ kernel is better able to represent the non-smooth real function.

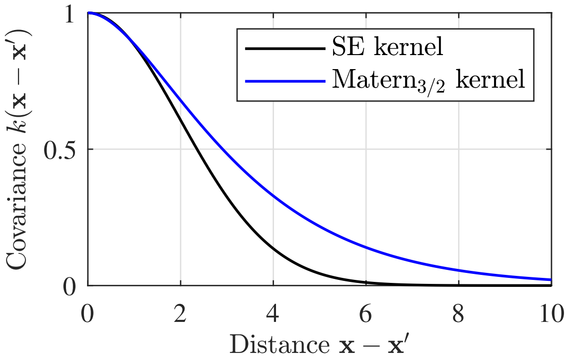


Figure C.4: Covariance of the SE kernel and Matèrn$_{3/2}$ compared, with $\ell_{\mathrm{SE}} = 2$, $\ell_{\mathrm{Matèrn}} = 3$. Now the covariance of the SE kernel is similar to that of the Matèrn$_{3/2}$ kernel for short distances.

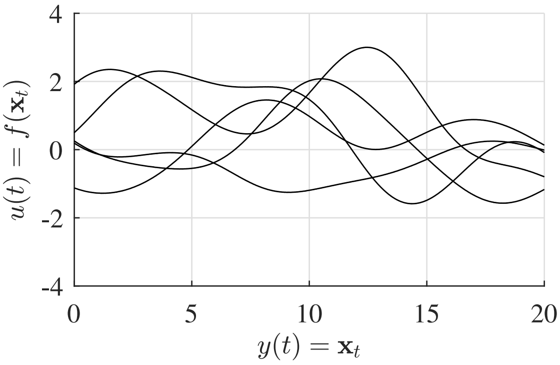


Figure C.5: Five random functions drawn from the prior of a GP with an SE kernel with $\ell_{\mathrm{SE}} = 2$. Even with a narrower covariance than the Matèrn$_{3/2}$ kernel with $\ell_{\mathrm{Matèrn}} = 3$, the priors with the SE kernel admit less variation in $f(\mathbf{x})$ than the rough priors observed using the Matèrn$_{3/2}$ kernel.

However, the prior with $\ell_{\mathrm{SE}} = 2$, depicted in Figure C.5, still exhibits significantly less variation than that of the Matèrn$_{3/2}$ kernel with $\ell_{\mathrm{Matèrn}} = 3$ in Figure C.2. This implies that the width of the SE kernel, as seen in Figures C.1 and C.4 and defined by ℓ, is not the cause of the SE kernel being worse than the Matèrn$_{3/2}$ kernel at representing non-smooth systems H^{-1}. Instead, this is caused by the structure of the SE kernel, as will be shown next. ■

C.2 Derivatives of Gaussian Processes

When discussing smoothness of functions $H^{-1} : f(\mathbf{x})$, the concept of derivatives arises naturally. In fact, since differentation is a linear operator, the derivative of a GP of $f(\mathbf{x})$ is another GP [22, Section 9.4]. The *mean square* (MS) *derivative* of a Gaussian process $f(\mathbf{x})$ in direction i is defined as

$$\frac{\partial f(\mathbf{x})}{\partial x_i} = \underset{h\to 0}{\mathrm{l.i.m}} \frac{f\left(\mathbf{x} + h\mathbf{e}_i\right) - f(\mathbf{x})}{h}, \tag{C.5}$$

where l.i.m refers to the limit in the mean square and $\mathbf{e}_i$ is the unit vector in direction i [22, Section 4.1.1]. This new Gaussian Process $\partial f(\mathbf{x})/\partial x_i$ is defined by a covariance function that can be expressed in terms of $k(\mathbf{x}, \mathbf{x}')$, such that the GP is denoted as

$$\frac{\partial f(\mathbf{x})}{\partial x_i} \sim \mathcal{GP}\left(0, \frac{\partial^2 k(\mathbf{x}, \mathbf{x}')}{\partial x_i \partial x_i'}\right) : \mathbb{R}^{n_\theta} \to \mathbb{R}, \tag{C.6}$$

given that $f(\mathbf{x})$ is defined with a zero-mean function as in (3.7). Thus, for a Gaussian Process $f(\mathbf{x})$ to be k-times MS-differentiable (i.e. the kth order partial derivative derivative $\partial^k f(\mathbf{x})/\partial x_{i_1} \dots x_{i_k}$ exists for all $\mathbf{x} \in \mathbb{R}^{n_\theta}$), it must hold that the $2k$th order partial derivative $\partial^{2k} k(\mathbf{x})/\partial^2 x_{i_1} \dots \partial^2 x_{i_k}$ exists and is finite at $\mathbf{x} = \mathbf{0}$. These definitions can be extended to derivatives of higher order [22, Section 4.1.1].

Whereas a GP of $f(\mathbf{x})$ with an SE kernel is infinitely MS-differentiable, a GP of $f(\mathbf{x})$ with a Matèrn$_{3/2}$ kernel is only 1-time MS-differentiable [22, Section 4.2]. It then becomes clear immediately from (C.6) why the Matèrn$_{3/2}$ kernel constitutes a prior that is better able at representing non-smooth functions $f(\mathbf{x})$ than the SE kernel, as observed in the previous section.

Appendix D

Alternative system parametrizations

In this appendix, different system parametrizations are explained. First, a linear parametrization is given and it is shown how this relates to the kernel-based regularization estimate [5]. Subsequently, a kernel for Volterra parametrizations is given.

Moreover, it is shown how kernels can be combined to represent systems consisting of a sum or product of different structures. Lastly, it is explained that while this enables the simultaneous identification of linear and non-linear dynamics in causal models, such an approach is not directly applicable to the non-causal models created by KIMCON.

D.1 Linear systems

Consider the following kernel function:

$$k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top P \mathbf{x}', \tag{D.1}$$

with $P \in \mathbb{R}^{n_\theta \times n_\theta}$ a positive semi-definite matrix.

Proposition D.1.1 (Equivalence of KBR and GP regression)**.** *The posterior mean* (3.11) *of Gaussian Process* (3.7) *with covariance function* (D.1)*, $X = \Phi_N$ and $P = \Pi$ is equivalent to the kernel-based regularization solution* (A.5) *with $\Phi = \Phi_N$ in* (A.6)*, regardless of the data.*

Proof. See Appendix E. □

Substitution of this kernel function in (3.1) leads to the following parametrization of $\widehat{H^{-1}}$:

$$u(t) =: f(\boldsymbol{\alpha} \mid \mathbf{x}_t) = \sum_{i=1}^{N} \alpha_i \mathbf{x}_t^\top P \mathbf{x}_i, \tag{D.2}$$

such that the control effort $u(t)$ is a linear combination of output sequence $\mathbf{x}_t$ and observations $\mathbf{x}_i \in \mathcal{D}$.

An example of a Gaussian Process with a linear kernel is given next.

Example 6. Consider the same setting as Example 1, where $H : y(t) = 2u(t)$ denotes the transfer function of a linear system and the choice $\mathbf{x}_t := [y(t)]$ is made, i.e. $n_c = n_{ac} = 0,\ n_\theta = 1$.

The inverse dynamics are

$$u(t) = f(\mathbf{x}_t) = \frac{y(t)}{2} + \varepsilon, \quad \varepsilon \sim \mathcal{N}\left(0, 10^{-6}\right). \tag{D.3}$$

Three noisy observations of f at locations $\mathbf{x}_i$ are available, such that the training matrix is

$$X = \begin{bmatrix}\mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3\end{bmatrix}^\top = \begin{bmatrix}-3 & 0 & 3\end{bmatrix}^\top, \tag{D.4}$$

with training targets

$$\mathbf{u} = \begin{bmatrix}-1.496 & 0.003 & 1.499\end{bmatrix}^\top. \tag{D.5}$$

In contrast to Example 1, which uses a stationary SE kernel, the GP is defined here with the linear kernel function (D.1) with $P = 1$. The GP is used to make predictions of $\mathbf{f}_*$ at locations $X_* = [-20, \dots, 20]^\top$ by computation of (3.11). The resulting *posterior distribution* is shown for each $\mathbf{x}_* \in X_*$ in Figure D.1. As opposed to the GP with a stationary kernel in Example 1, here the posterior mean is close to the real function even far from the observations, even if Condition (3.23) is not satisfied. This is because the system parametrization (3.1) with linear kernel (D.1) matches the linear structure of H^{-1}. This shows that *for linear systems*, this parametrization by (D.1) is a reasonable choice of [5] indeed. However, in KIMCON, non-linear systems are considered and therefore stationary covariance functions are preferred, see Section 3.3.1.

If the noise ε were larger, the slope of the posterior in Figure D.1 may have been more visibly incorrect. The matrix P functions as regularization to mitigate this effect, see Section A.2 (where $P = \Pi$). ■

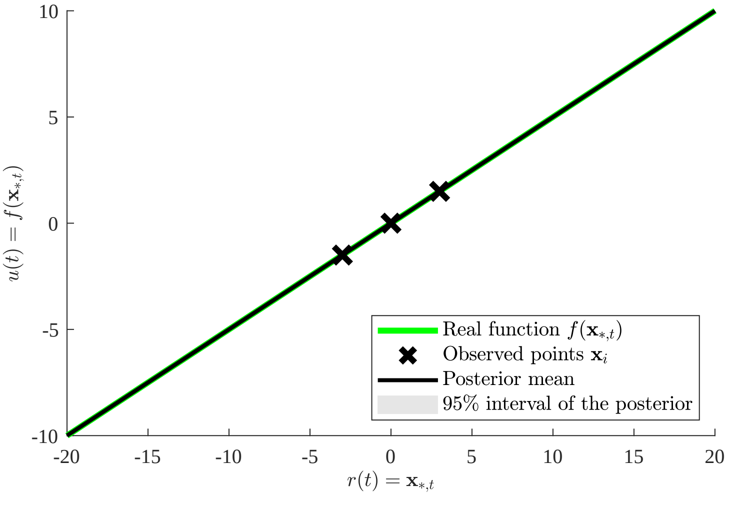


Figure D.1: Posterior distribution of the Gaussian Process from Example 6, with a linear kernel. The posterior mean matches the real function well, because the real function is linear.

D.2 Volterra systems

Suppose H^{-1} is a pth order Volterra system of the form

$$u(t) = f(\mathbf{x}) = \sum_{n=0}^{p} H_n(\mathbf{x}) = \sum_{n=0}^{p} \sum_{i_1=1}^{n_\theta} \dots \sum_{i_n=1}^{n_\theta} h^{(n)}_{i_1 \dots i_n} x_{i_1} \dots x_{i_n}. \tag{D.6}$$

The homogeneous operator H_n can be written as

$$H_n(\mathbf{x}) = \eta_n^\top \phi_n(\mathbf{x}), \tag{D.7}$$

with $\phi_n(\mathbf{x}) = \left[x_1^n, x_1^{n-1}x_2, \ldots, x_1x_2^{n-1}, x_2^n, \ldots, x_m^n\right]$ and $\eta_n = \left(h_{1,1,\ldots,1}^{(n)}, h_{1,2,\ldots,1}^{(n)}, h_{1,3,\ldots,1}^{(n)}, \ldots\right)^\top$. It is shown in [7] that since the monomials in (D.7) constitute a RKHS, (D.6) can be parametrized by (3.1). In particular, the following kernel function is shown to represent $f(\mathbf{x})$:

$$k(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^\top \mathbf{x}')^p. \tag{D.8}$$

Remark 8. Since the covariance of the prior grows rapidly for $|\mathbf{x}| > 1$, it may be desirable to normalize $\mathbf{x}$ to $[-1, 1]$, see [22, Section 4.2.2]. ■

In [4], an alternative kernel is proposed that exploits prior knowledge of the Volterra operator H_n in (D.6), but only causal systems are considered.

D.3 Combining kernels

The sum, product and convolution of two kernels $k_1(\mathbf{x}, \mathbf{x}')$ and $k_2(\mathbf{x}, \mathbf{x}')$ is a valid (positive-semidefinite) kernel [22, Section 4.2.4]. The application for a sum of kernels is explained next, but the same reasoning can be used for the product and convolution of kernels [1].

Let $f(\mathbf{x}) = f_1(\mathbf{x}) + f_2(\mathbf{x})$. It is proven in [1] that if $f_1 \in \mathcal{H}_1$ and $f_2 \in \mathcal{H}_2$, with reproducing kernel Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$ defined by $k_1(\mathbf{x}, \mathbf{x}')$ and $k_2(\mathbf{x}, \mathbf{x}')$ respectively, then $k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}')$ defines $\mathcal{H} = \mathcal{H}_1 + \mathcal{H}_2$ and thus $k(\mathbf{x}, \mathbf{x}')$ has the reproducing property $\langle f(\cdot), k(\cdot, \mathbf{x})\rangle_{\mathcal{H}} = f(\mathbf{x})$. In conclusion, functions $f(\mathbf{x})$ consisting of a sum of different structures can be modeled using a sum of relevant kernel functions.

Example 7. Consider again system (3.22). Since kernel (D.1) represents the class of linear functions and the periodic term is represented by kernel (3.20), $f(\mathbf{x})$ can be represented by

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \left(\exp\left[-\frac{1}{2} \sum_i \left(\frac{\sin\left(\frac{\pi}{r_i}(x_i - x_i')\right)}{\ell_i} \right)^2 \right] + \lambda \mathbf{x} P \mathbf{x}' \right), \tag{D.9}$$

with $\lambda \in \mathbb{R}$. If the non-linear term in (3.22) were not known to be periodic, the sum of kernels (D.1) and (3.16) or (3.18) can be used. ■

D.4 Simultaneous identification of linear and non-linear dynamics

As described in Appendix D.3, a GP with a sum of kernels can represent a function with a sum of different (possibly non-linear) structures, because the kernel function represents the system parametrization, see (3.1).

In [20], the system parametrization is the sum of SE kernel (3.16) and linear kernel $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top P \mathbf{x}'$, $P \in \mathbb{R}^{n_\theta \times n_\theta}$. While this allows for simultaneous identification of linear and non-linear dynamics, it has been a conscious choice not to include this linear kernel in KIMCON. The reason for this is as follows.

The approach of [20] models causal systems H as a GP, in the form of (2.1). In this case, the data-collection challenge described in Section 3.4 is non-existent since the input to the GP then is $u(t)$, which can be chosen freely. However, as described in said section, when modeling H^{-1} one has to try to obtain observations close to the reference. Such a data-set is not persistently exciting, which is a requirement of linear system identification where the model needs to be data-independent, in contrast to the parametrizations in Section 3.3.1.

For this reason, parametrizing the system in such a way would require two different data-sets. While this is possible, it has been decided to assume that the linear model $\widehat{H^{-1}}_{\text{prior}}$ is available a priori, such that it can be used as a feedforward controller in the data-collection procedure described in Section 3.4.

Appendix E

Kernel-based regularization as a special case of Gaussian Process regression

This appendix contains the proof of Proposition proposition D.1.1:

Proposition D.1.1 (Equivalence of KBR and GP regression)**.** *The posterior mean* (3.11) *of Gaussian Process* (3.7) *with covariance function* (D.1)*, $X = \Phi_N$ and $P = \Pi$ is equivalent to the kernel-based regularization solution* (A.5) *with $\Phi = \Phi_N$ in* (A.6)*, regardless of the data.*

Proof. Let GP (3.7) be defined by the following covariance function:

$$k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top P \mathbf{x}'. \tag{E.1}$$

This covariance function is commonly referred to as a *linear regularized covariance function* because it leads to a linear parametrization when substituted in (3.1), and P can be interpreted as a regularization matrix.

Moreover, assume that $y(N+i) = y(-i) = 0 \ \forall \ i \in \mathbb{N}$. Under this assumption, the matrix of training inputs $X = \begin{bmatrix} \mathbf{x}_1 & \dots & \mathbf{x}_N \end{bmatrix}^\top \in \mathcal{D}$ equals the regressor matrix Φ_N from (A.6).

Substitution of the covariance function in the system parametrization (3.1) then leads to the form

$$\begin{aligned} u(t) =: f(\mathbf{x}_t) &= \sum_{i=1}^{N} \alpha_i \mathbf{x}_t^\top P \mathbf{x}_i \\ &= \mathbf{x}_t^\top P \sum_{i=1}^{N} \alpha_i \mathbf{x}_i \\ &= \mathbf{x}_t^\top P \Phi_N^\top \boldsymbol{\alpha}. \end{aligned} \tag{E.2}$$

To denote the control effort sequence $\mathbf{f}$ that led to observed output sequences $X = \begin{bmatrix} \mathbf{x}_1 & \dots & \mathbf{x}_N \end{bmatrix}^\top$, we expand (E.2) vertically to obtain

$$\mathbf{f}(\boldsymbol{\alpha} \mid X) = X P \Phi_N^\top \boldsymbol{\alpha}. \tag{E.3}$$

If coefficients $\boldsymbol{\alpha}$ are known, X can be replaced by X_* to obtain the control effort required for arbitrary reference sequences $X_* \in \tilde{\mathcal{R}}$. It is shown next that the KBR problem (A.5) can be retrieved from the optimization problem (3.3) characterizing a Gaussian Process (as explained in Section 3.1).

First, note that the Gramian K, shorthand for $K(X,X)$ with $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ of covariance function (E.1), is given by the definition of Gramian matrices:

$$K = XPX^\top. \tag{E.4}$$

With $X = \Phi_N$, as assumed in this proposition, this becomes

$$K(X,X) = \Phi_N P \Phi_N^\top, \tag{E.5}$$

such that under parametrization (E.2), the control effort sequence $\mathbf{f}$ explaining the data Φ_N can be written as

$$\mathbf{f}(\boldsymbol{\alpha} \mid X = \Phi_N) = K\boldsymbol{\alpha}. \tag{E.6}$$

This result allows us to rewrite optimization problem (3.3) characterizing the GP as

$$\begin{aligned} \min_f J[f] &= \|u_N - \mathbf{f}(\boldsymbol{\alpha} \mid X = \Phi_N)\|_2^2 + \gamma \|f(\boldsymbol{\alpha} \mid X = \Phi_N)\|_{\mathcal{K}}^2 \\ &= \|\mathbf{u} - K\boldsymbol{\alpha}\|_2^2 + \gamma \|K\boldsymbol{\alpha}\|_{\mathcal{K}}^2 \\ &= \|\mathbf{u} - K\boldsymbol{\alpha}\|_2^2 + \gamma \boldsymbol{\alpha}^\top K \boldsymbol{\alpha}, \end{aligned} \tag{E.7}$$

where the last step results from the definition of the $\mathcal{K}$-norm, see [22, Section 6.2.2].

Next, the optimization problem will be altered to arrive at the KBR problem. In particular, $f(\boldsymbol{\alpha} \mid X = \Phi_N)$ will be replaced by $f(\boldsymbol{\alpha} \mid X = X_* = I_N)$, to represent the case in which $\mathbf{f}$ yields a feedforward signal θ required to produce output $X_* = I_N$, i.e., a non-causal impulse on reference r for output y. The choice $X_* = I_N$ representing a non-causal impulse on r follows from the ordering of $\mathbf{x}_* \in X_*$, assuming $r(N+i) = r(-i) = 0 \ \forall \ i \in \mathbb{N}$:

$$X_* = \begin{bmatrix} r(n_{ac}) & r(n_{ac}-1) & \dots & r(-n_c) \\ r(n_{ac}+1) & r(n_{ac}) & \dots & r(-n_c+1) \\ \vdots & \vdots & & \vdots \\ r(N-1+n_{ac}) & r(N-2+n_{ac}) & \dots & r(N-n_c-1) \end{bmatrix}. \tag{E.8}$$

See Example 8 at the end of this Appendix for more details. Using (E.3) we can then relate $\mathbf{f}$ to impulse response coefficients θ (see (A.3)) as

$$\mathbf{f}(\boldsymbol{\alpha} \mid X_* = I_N) = I_N P \Phi_N^\top \boldsymbol{\alpha} \equiv \theta, \tag{E.9}$$

such that coefficients $\boldsymbol{\alpha}$ can be expressed in terms of impulse response coefficients θ as

$$\boldsymbol{\alpha} \equiv (\Phi_N^\top)^\dagger P^{-1} \theta. \tag{E.10}$$

Here, $(\Phi_N^\top)^\dagger$ denotes the *right inverse*, i.e., $\Phi_N^\top (\Phi_N^\top)^\dagger = I_N$. Substitution of this expression in (E.7) with (E.4) yields the problem

$$\begin{aligned} \min_f J[f] &= \|\mathbf{u} - K(\Phi_N^\top)^\dagger P^{-1}\theta\|_2^2 + \gamma((\Phi_N^\top)^\dagger P^{-1}\theta)^\top K (\Phi_N^\top)^\dagger P^{-1}\theta \\ &= \|\mathbf{u} - \Phi_N P \Phi_N^\top (\Phi_N^\top)^\dagger P^{-1}\theta\|_2^2 + \gamma((\Phi_N^\top)^\dagger P^{-1}\theta)^\top \Phi_N P \Phi_N^\top (\Phi_N^\top)^\dagger P^{-1}\theta \\ &= \|\mathbf{u} - \Phi_N\theta\|_2^2 + \gamma((\Phi_N^\top)^\dagger P^{-1}\theta)^\top \Phi_N \theta \\ &= \|\mathbf{u} - \Phi_N\theta\|_2^2 + \gamma\theta^\top (P^{-1})^\top \theta \\ &= \|\mathbf{u} - \Phi_N\theta\|_2^2 + \gamma\theta^\top P^{-1} \theta, \end{aligned} \tag{E.11}$$

since P is symmetric by definition of a kernel matrix. This optimization problem is identical to the kernel-based regularization problem (A.5) for $P = \Pi$. □

In the next example, it is shown how a matrix $X_* = I$ represents a non-causal impulse response.

Example 8. Suppose H^{-1} is represented by the non-causal NFIR parametrization (2.4) with $n_c = 1$, $n_{ac} = 2$, such that control effort $u(t) = f(\mathbf{x}_{*,t})$ produces reference sequence $\mathbf{x}_{*,t}$ when applied to H, defined as

$$\mathbf{x}_{*,t} := [r(t+2), r(t+1), r(t), r(t-1)]^\top. \tag{E.12}$$

The reference describing an impulse is denoted as

$$\begin{aligned} \mathbf{r} &= [r(0), r(1), r(2), r(3)]^\top \\ &= [0, 0, 1, 0]^\top, \end{aligned} \tag{E.13}$$

such that $N = 4 = n_\theta$. Under the assumption that $r(N+i) = r(-i) = 0 \; \forall \; i \in \mathbb{N}$, this reference may be extended as

$$\begin{aligned} \mathbf{r} &= [r(-1), r(0), r(1), r(2), r(3), r(4), r(5)]^\top \\ &= [0, 0, 0, 1, 0, 0, 0]^\top. \end{aligned} \tag{E.14}$$

The overlapping test sequences $\mathbf{x}_{*,t} \forall t \in [0, \ldots, N-1]$ can then be constructed from the reference by definition of (E.12) as

$$\begin{aligned} \mathbf{x}_{*,t=0} &= [r(0+2), r(0+1), r(0), r(0-1)]^\top = [1, 0, 0, 0]^\top \\ \mathbf{x}_{*,t=1} &= [r(1+2), r(1+1), r(1), r(1-1)]^\top = [0, 1, 0, 0]^\top \\ \mathbf{x}_{*,t=2} &= [r(2+2), r(2+1), r(2), r(2-1)]^\top = [0, 0, 1, 0]^\top \\ \mathbf{x}_{*,t=3} &= [r(3+2), r(3+1), r(3), r(3-1)]^\top = [0, 0, 0, 1]^\top. \end{aligned} \tag{E.15}$$

By definition of the matrix $X_* := [\mathbf{x}_{*,t=0}, \ldots, \mathbf{x}_{*,t=N-1}]^\top$, the impulse reference (E.13) then leads to $X_* = I$. ■

Appendix F

MATLAB code of the Wiener example

This chapter contains the MATLAB code of the Wiener example in Section 4.1. The accompanying controller file and Simulink file (see Figure F.1) can be found here: http://maxvanmeer.nl/wienerGP.zip

For simplicity, the code only includes the results yielded by KIMCON, i.e., the comparison with other methods is omitted. The code relies on the Statistics and Machine Learning Toolbox and the DSP System Toolbox of MATLAB.

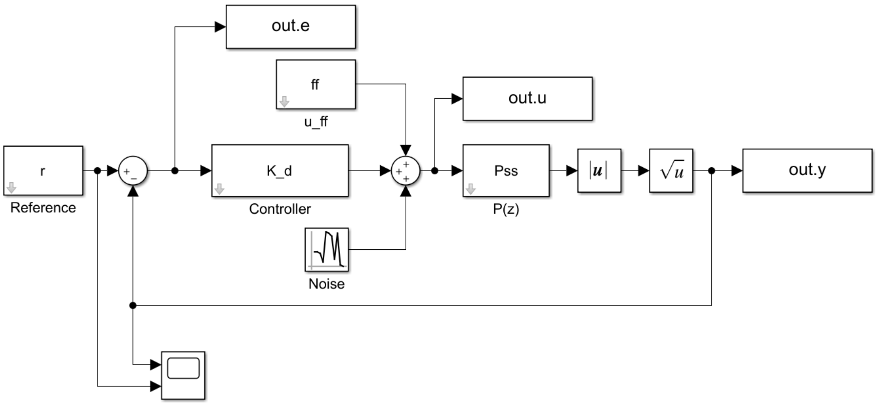


Figure F.1: Simulink file corresponding to the Wiener example.

```
clc
clear
close all

%% Define parameters
n_ac = 300;
n_c = 300;
n_th = n_c + n_ac + 1;
std_noise = 1e-3;
tau = 1/20;

%% Define P and C
w1 = 1*2*pi; % natural eigenfrequency [rad/s]
```

```
zeta1 = 0.1; % damping ratio
w2 = -0.4*2*pi; % natural eigenfrequency [rad/s]
zeta2 = 0.2; % damping ratio
s=tf('s');
Pinv_c = (tf([w1^2],[1 2*zeta1*w1 w1^2]) * tf([w2^2],...
    [1 2*zeta2*w2 w2^2])) * (s+1)^2;

% Discretize the system
Pinv = c2d(Pinv_c,tau);
Pinv =ss(Pinv);
[zz,pp,kk]=zpkdata(Pinv);
zz=zz{1};
zz_rem = zz(1);%[zz(3),zz(4)];
zz=[zz(2),zz(3)];
k_scale = sum(1+abs(zz_rem));
Pinv=zpk(zz,pp,kk*k_scale,tau);
P_tmp = inv(Pinv);
z = tf('z',tau);
P = P_tmp / z^2;
[num,den] = tfdata(P);
Pss = ss(P);

% Load the controller
load Controller
K_d = c2d(shapeit_data.C_tf,tau);
%% Define reference
T = 800; % Length of the reference
t_final = 0:tau:T*tau-tau;

% Start with 20 seconds of a constant reference to get rid of transients:
n_pre = ceil(20/tau);
r_final = cos(2*pi*t_final / T / tau)' * 0.5 + 1;
r_final = [r_final(1) * ones(n_pre,1); r_final; r_final(end)*ones(n_ac,1)];
t_final = 0:tau:length(r_final)*tau-tau;
T = length(t_final);

%% Design excitation references and perform simulations
T_tilde = 20/tau + T;
t_tilde = 0:tau:T_tilde*tau-tau;
freqs = linspace(0.3,1.5,20);

X = []; u = [];
c = 1;
n_skip = 40;
options = simset('SrcWorkspace','current');
for i = 1:length(freqs)
    r_tmp{i} = 1 - 0.5*cos(2*pi*t_tilde * 0.6/4/5 * freqs(i)-pi)';

    r_tmp{i} = [r_tmp{i}(1)*ones(n_pre,1);  r_tmp{i}(1:end-n_pre)];
    y0 = r_tmp{i}(1);
    ff = zeros(T,1);
    r = r_tmp{i};
    t = t_tilde;
    out_tmp = sim('Wiener_gp_sim',[],options);
    nstart = 300; % We don't care about the first 300 samples (transients)
    idxs = nstart:nstart+1200-20; % Indices we want to extract for D
    y_train = out_tmp.y(idxs);
    u_train = out_tmp.u(idxs);
    for j = 1+n_c:n_skip:length(y_train)-n_ac
        X(c,:) = y_train(j+n_ac:-1:j-n_c);
        u(c,1) = u_train(j);
        c = c + 1;
    end
    all_y{i} = y_train;
    all_u{i} = u_train;
end

```

```
figure
plot(t_final,r_final,'c','LineWidth',2);
hold on
for i = 1:length(all_y)
    plot(t_tilde(idxs),all_y{i},'r')
end
grid
legend('$\mathbf{r}$','${\mathbf{y}}_i$','Location','southwest',...
    'Interpreter','Latex')
xlabel('Time [s]');
ylabel('Output [-]');
%% Define the GP and optimize hyper-parameters
% Initial values for fitrgp
l = 1; % kernel width
stdf = 15; % variance of u(t)
gp_Mdl_ref = fitrgp(X,u,...
    'Kernelfunction','ardsquaredexponential',...
    'KernelParameters',[l*ones(n_th,1);stdf],...
    'Sigma',std_noise,...
    'BasisFunction','none');
%% Compute the feedforward signal
% Again, insert a constant reference first to get rid of transients
pre2 = 3000;
r_final_long = [r_final(1)*ones(pre2,1); r_final];
t_final_long = [0:tau:pre2*tau-tau, pre2*tau+t_final];
f_gp = zeros(T,1);
L2 = length(r_final_long);
for i = 1:L2
    idx_vals = min(i+n_c,L2):-1:max(i-n_ac,1);
    idx = [L2 * ones(1,i+n_c-L2), idx_vals, 1 * ones(1,1 - (i-n_ac))];
    f_gp(i) = predict(gp_Mdl_ref,r_final_long(idx)');
end
cut_idx = pre2+1:L2;
f_gp_cut = f_gp(cut_idx);

figure
plot(t_final,f_gp_cut,'r')
grid
xlabel('Time [s]');
ylabel('Control effort [N]');
%% Closed-loop simulation
y0 = r_final(1);
r = r_final_long;
t = t_final_long;
ff = f_gp;
out_gp = sim('Wiener_gp_sim',[],options);

figure
stairs(t_final,out_gp.e(cut_idx),'r','LineWidth',1.1)
xlabel('Time [s]');
ylabel('Output error [-]');
grid
axis([0 max(t_final) -0.015 0.015]);
```

Appendix G

Declaration of Scientific Conduct

The next page includes a signed declaration of scientific conduct.

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Declaration concerning the TU/e Code of Scientific Conduct for the Master's thesis

I have read the TU/e Code of Scientific Conduct[i].

I hereby declare that my Master's thesis has been carried out in accordance with the rules of the TU/e Code of Scientific Conduct

Date

13-2-2021
………………………………………………………..…………..

Name

Max van Meer
………………………………………………………..…………..

ID-number

0951669
………………………………………………………..…………..

Signature

………………………………………………………..…………..

Submit the signed declaration to the student administration of your department.

[i] See: https://www.tue.nl/en/our-university/about-the-university/organization/integrity/scientific-integrity/
The Netherlands Code of Conduct for Scientific Integrity, endorsed by 6 umbrella organizations, including the VSNU, can be found here also. More information about scientific integrity is published on the websites of TU/e and VSNU