Gaussian Process based Feedforward Control for Nonlinear Systems with Flexible Tasks: With Application to a Printer with Friction

Max van Meer [*] **Maurice Poot** [*] **Jim Portegies** [*]
Tom Oomen [*,**]

[*] *Eindhoven University of Technology, 5600 MB Eindhoven, the Netherlands (e-mail: m.v.meer@tue.nl).*
[**] *Delft University of Technology, 2628 CD Delft, the Netherlands.*

Abstract: Feedforward control is essential to achieving good tracking performance in positioning systems. The aim of this paper is to develop an identification strategy for inverse models of systems with nonlinear dynamics of unknown structure using input-output data, which can be used to generate feedforward signals for a-priori unknown tasks. To this end, inverse systems are regarded as noncausal nonlinear finite impulse response (NFIR) systems, and modeled as a Gaussian Process with a stationary kernel function that imposes properties such as smoothness. The approach is validated experimentally on a consumer printer with friction and shown to lead to improved tracking performance with respect to linear feedforward.

Keywords: Feedforward Control, Nonlinear system identification, Grey box modelling, Identification for control, Gaussian Process regression

1. INTRODUCTION

System identification in the presence of nonlinear dynamics of unknown structure is a challenging subject, because of the wide range of possible model descriptions that need to be considered (Sjöberg et al., 1995). For feedforward control, models of inverse systems are of particular interest, since the filtering of some reference by the inverse system yields the required control effort for that task. For a system G, an inverse model G^{-1} can be obtained by either (i) identification of G and subsequent inversion, or (ii) direct identification of G^{-1}. It is shown in Blanken and Oomen (2020) that the latter poses an advantage over the former, since properties such as stability, smoothness and finite preview or history of G^{-1} can be enforced directly on the model.

If the nonlinear structure of the inverse system is known to be representable by, e.g., a set of some polynomial basis functions, the model coefficients can be learned perfectly in an iterative fashion, see Boeren et al. (2015); Van De Wijdeven and Bosgra (2010). Such an approach allows for the generation of feedforward signals for any task, because any reference can be filtered through this inverse model to obtain the required control effort.

Identification methods for inverse systems that rely on user-specified parametrization of the system through a limited number of nonlinear basis functions are not applicable when the nonlinear structure of a system is unknown. In such case, the structural model errors of the feedforward controller can lead to performance degradation (Schoukens and Ljung, 2019).

For a certain class of basis functions, namely, the eigenfunctions of universal kernels, structural model errors of a modeled function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ on a compact subset $\mathcal{X} \subset \mathbb{R}^n$ vanish as the number of basis functions grows to infinity (Micchelli et al., 2006). These kernels impose properties such as smoothness on the model in order to deal with the bias-variance trade-off. In Gaussian Process (GP) regression, these kernels pose a Gaussian prior on f, and predictions are made by extrapolating from measurements using Bayes' rule (Rasmussen and Williams, 2006), see Figure 1.

GPs have been shown in literature to be applicable to modeling dynamic systems with unknown nonlinear dynamics. Such modeling methods are widely available for Euler-Lagrange systems (Beckers et al., 2019; Nguyen-Tuong et al., 2008, 2009), single-input systems that offer full state measurements (Deisenroth and Rasmussen, 2011), and causal systems (Pillonetto et al., 2014).

Although these methods are capable of modeling dynamics with unknown nonlinear structure, the requirement of full state measurements is overly restrictive in some cases, and in the presence of nonlinear dynamics of unknown structure, the inclusion of a state observer is nontrivial. Moreover, for motion systems G, the inverse G^{-1} is always noncausal, and for reasons that will become clear in Section 3.3, nonlinear kernel-based identification methods of G may not be applicable to identification of G^{-1}.

The aim of this paper is to develop a novel technique for grey-box modeling of noncausal nonlinear systems with unknown structure based on input-output data. In particular, inverse systems G^{-1} are viewed as noncausal nonlinear finite impulse response (NFIR) systems and modeled as a Gaussian Process, in order to generate feedforward signals for a range of tasks that may be unknown a-priori. Consequently, the method yields a model of G^{-1} directly, in contrast to stable inversion-based approaches that rely

[⋆] This work is part of the research programme VIDI with project number 15698, which is (partly) financed by the Netherlands Organisation for Scientific Research (NWO). In addition, this research has received funding from the ECSEL Joint Undertaking under grant agreement 101007311 (IMOCO4.E). The Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation programme.

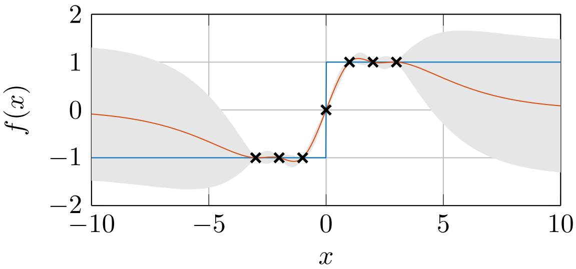

Fig. 1. An example function (—) modeled as a Gaussian Process. The posterior distribution, using a Matèrn$_{3/2}$ kernel, has an accurate mean (—) close to observations (×), and high standard deviation (▨) far from observations.

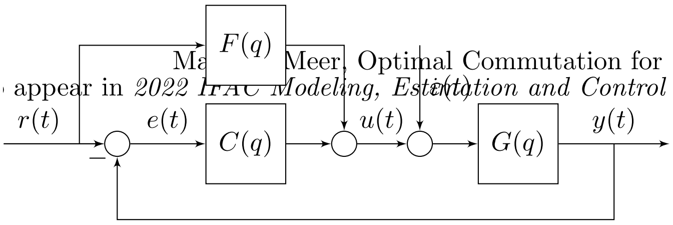

Fig. 2. Motion control architecture.

on inversion of G (Devasia, 1997; Pavlov and Pettersen, 2008). This paper consists of two main contributions:

C1: A procedure for GP-based feedforward control of systems with unknown nonlinear dynamics is proposed and described from a design perspective.

C2: The method is validated experimentally on a printer with friction to show its improved tracking performance with respect to linear feedforward control.

This paper is structured as follows. First, the problem description is given in Section 2. Subsequently, Gaussian Processes regression for NFIR systems is explained and design considerations for kernel selection and experimental design are given in Section 3. Afterwards, in Section 4 the method is applied to an experimental setup. Finally, Section 5 presents the conclusions and proposes some directions for future work.

2. PROBLEM DESCRIPTION

In this section, the problem description is given. First, the control architecture and goal are described. Next, an assumption on the structure of G^{-1} is explained and it is shown how this setting enables task flexibility.

2.1 Setting and goal

Let $G(q)$ denote a discrete-time, nonlinear SISO system, where q is the forward-shift operator such that $q^{\tau}a(t) := a(t+\tau)$, $\tau \in \mathbb{Z}$. The closed-loop motion control architecture in Figure 2 is considered, where $r(t)$ denotes a predefined reference to be tracked at time $t \in \mathbb{Z}$, $y(t)$ is the system output, $e(t)$ is the tracking error and $\varepsilon(t) \sim \mathcal{N}(0, \sigma_n^2)$ is an i.i.d. disturbance.

It is assumed that a stabilizing feedback controller $C(q)$ is available, as well as some prior feedforward controller $F(q)$. Perfect tracking is achieved for $F(q) = G^{-1}(q)$, since then

$$\begin{aligned} e(t) &= r(t) - y(t) \\ &= r(t) - G(q)u(t) \\ &= r(t) - G(q)G^{-1}(q)r(t) \\ &= 0. \end{aligned} \tag{1}$$

The goal in this paper is to obtain $\hat{F}(q) \approx G^{-1}(q)$ from data such that the 2-norm of the tracking error ($\|e\|_2$) is reduced. Moreover, task flexibility must be allowed.

Definition 1. (Task flexibility). Task flexibility refers to the notion that the designed feedforward controller $\hat{F}(q)$ must achieve improved tracking performance for a range of different tasks that may not all be known prior to the generation of $\hat{F}(q)$.

The next section explains an additional assumption on the structure on $G^{-1}(q)$, before the problem formulation is formalized in Section 2.3.

2.2 Noncausal nonlinear finite impulse response systems

In order to find a feedforward controller $\hat{F}(q) \approx G^{-1}(q)$, a structure on $G^{-1}(q)$ is imposed first. Note that *linear* sampled systems $P(z)$ of continuous-time systems $P(s)$ often contain zeros outside the unit disc, depending on the sample time and the relative degree of $P(s)$, see Åström et al. (1984). Consequently, $P^{-1}(z)$ may contain poles outside the unit disc. Whereas systems with poles outside the unit disc are typically viewed as causal and unstable, they may also be viewed as noncausal and stable (Blanken and Oomen, 2020).

Theorem 1. (Non-causal exact inversion). Let system $P(z)$ be given such that $P^{-1}(z) \in \mathcal{RL}_2(\mathbb{T})$, i.e., the set of real, rational, discrete-time systems without poles on the unit disc $\mathbb{T} := \{z \in \mathbb{C} : |z| = 1\}$. Then, there exists a non-causal sequence $\theta \in \ell_1(\mathbb{Z})$ such that, for any signal $r(t) \in \ell_2(\mathbb{Z})$, the signal

$$u(t) = \sum_{\tau=-\infty}^{\infty} \theta_{\tau} r(t-\tau) \in \ell_2(\mathbb{Z}) \tag{2}$$

leads to exact inversion $y(t) = P(q)u(t) = r(t)$.

In the absence of infinite preview or history, a non-causal FIR system parametrization may be used as a finite-dimensional approximation of (2):

$$\begin{aligned} u(t) &\approx \sum_{\tau=-n_{ac}}^{n_c} \theta_{\tau} r(t-\tau) \\ &\approx \theta^{\top}\mathbf{r}_t =: f_{\text{lin}}(\mathbf{r}_t), \end{aligned} \tag{3}$$

with $\mathbf{r}_t := \left[r\left(t+n_{ac}\right), \ldots, r\left(t-n_c\right)\right]^{\top} \in \mathcal{Y}$, where $\mathcal{Y} \subset \mathbb{R}^{n_\theta}$ is compact with $n_\theta = n_c + n_{ac} + 1$, for reasons described in Section 3.2. The values n_c and n_{ac} denote the number of (causal) samples of history, and the number of (noncausal) samples of preview, respectively.

In this paper, this idea is extended to nonlinear systems by assuming that the nonlinear system $G^{-1}(q)$ can be represented with a non-causal nonlinear finite impulse response (NFIR) parametrization, such that the control effort $u(t)$ required to realize output sequence $\mathbf{y}_t := \left[y\left(t+n_{ac}\right), \ldots, y\left(t-n_c\right)\right]^{\top} \in \mathcal{Y}$ can be written as

$$u(t) = f\left(\mathbf{y}_t\right), \tag{4}$$

where $f : \mathcal{Y} \to \mathbb{R}$ is an alternative form of $G^{-1}(q) : \mathbb{R} \to \mathbb{R}$, in which q has been eliminated.

2.3 Problem formulation

In this paper, the aim is to model the function f in (4) describing G^{-1} from a dataset $\mathcal{D} = \{u(t), y(t)\}_{t=1}^{M}$, such that $f(\mathbf{r}_t)$ yields the control effort (or feedforward signal) $u(t)$ that realizes reference sequence $\mathbf{r}_t := [r(t+n_{ac}), \ldots, r(t-n_c)]^\top \in \mathcal{Y}$. Note that by learning f, feedforward samples $u(t)$ can be computed for any $\mathbf{r}_t$, i.e., task flexibility is allowed.

3. NONLINEAR FEEDFORWARD USING GAUSSIAN PROCESS REGRESSION

In this section, it is explained how Gaussian Process regression is employed to generate feedforward signals for nonlinear systems. First, it is explained how NFIR systems can be represented by Gaussian Processes. Second, the choice of kernel function is addressed from a design perspective. Finally, it is shown how a data-set can be obtained that allows for task flexibility.

3.1 Gaussian Process models of NFIR systems

To learn f in (4) from data, it is regarded as a *Gaussian Process* (GP), see Rasmussen and Williams (2006).

Definition 2. (Gaussian Process). A Gaussian Process is defined as an indexed family of random variables $g(\mathbf{x}) \in \mathbb{R}$ with $\mathbf{x} \in \mathcal{X}$, any finite number of which have a joint Gaussian distribution.

Hence, if $f(\mathbf{y}_t)$ is a GP, then there exists a joint distribution between *observations* of $u(t) = f(\mathbf{y}_t) + \varepsilon(t)$ and *unknown feedforward samples* $u(t) = f(\mathbf{r}_t)$. It is shown next how GPs can be used to make predictions of these feedforward samples, based on observations of u and y.

Let the covariance between any two control effort values required for output sequences $\mathbf{y}_t$ be defined by a kernel function

$$k(\mathbf{y}_{t_1}, \mathbf{y}_{t_2}) := \operatorname{cov}(f(\mathbf{y}_{t_1}), f(\mathbf{y}_{t_2})). \tag{5}$$

This kernel function imposes properties on the function space of f, as explained in detail in Section 3.3. Given a dataset $\mathcal{D} = \{Y, \mathbf{u}\}$, with

$$\begin{aligned} Y &= [\mathbf{y}_1, \ldots, \mathbf{y}_M]^\top, \\ \mathbf{u} = \mathbf{f}(Y) + \varepsilon &= [u(1), \ldots, u(M)]^\top, \end{aligned} \tag{6}$$

and assuming a zero-mean prior on f, define the joint distribution

$$\begin{bmatrix} \mathbf{u} \\ \mathbf{f}(R) \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(Y,Y) + \sigma_n^2 I & K(Y,R) \\ K(R,Y) & K(R,R) \end{bmatrix}\right), \tag{7}$$

where $R = [\mathbf{r}_1, \ldots, \mathbf{r}_N]^\top$ is a toeplitz matrix constructed from $r \in \mathbb{R}^N$, with $\mathbf{r}_t := [r(t+n_{ac}), \ldots, r(t-n_c)]^\top$ and σ_n^2 the variance of the measurement noise ϵ. The covariance matrices K are constructed by evaluating k for each element.

To compute the feedforward signal $\mathbf{f}(R)$ for reference R, the joint distribution in (7) is conditioned on the observations using Bayes' rule. The resulting posterior distribution $p(\mathbf{f}(R) \mid Y, \mathbf{u}, R)$ is a Gaussian with mean

$$\mathbb{E}[\mathbf{f}(R)] = K(R,Y)\left[K(Y,Y) + \sigma_n^2 I\right]^{-1} \mathbf{u}. \tag{8}$$

This yields an analytic expression for the expected feedforward signal required for reference r. Note that this entire expression can be computed offline if the reference is available. If the reference is not available beforehand, only the $K(R,Y)$ term needs to be computed online.

3.2 Model structure

Whether the realization $\mathbb{E}[\mathbf{f}(R)]$ is representative of the real function f is dependent on the choice of k in (5). This can be seen easily by rewriting the posterior mean (8) in scalar form as

$$u(t) = \mathbb{E}[f(\mathbf{r}_t)] = \sum_{i=1}^{M} \alpha_i k(\mathbf{y}_i, \mathbf{r}_t), \tag{9}$$

with $\boldsymbol{\alpha} = (K(Y,Y) + \sigma_n^2 I)^{-1}\mathbf{u}$. Naturally, if k were to be chosen inappropriately, e.g., linear in $\mathbf{r}_t$, as done for linear systems in Blanken and Oomen (2020), no nonlinear function f could be represented by the posterior mean. A particularly widely applicable type of kernel is the *universal kernel* (Micchelli et al., 2006).

Definition 3. (Universal kernel). Given any compact subset $\mathcal{Z}$ of $\mathcal{Y}$, let $C(\mathcal{Z})$ be the space of all continuous complex-valued functions from $\mathcal{Z}$ to $\mathbb{C}$ with maximum norm $\|.\|_{\mathcal{Z}}$. More over, let $K(\mathcal{Z})$ denote all functions in $C(\mathcal{Z})$ which are uniform limits of functions of the form (9) where $\{\mathbf{y}_i : i \in \mathbb{N}_n\} \subseteq \mathcal{Z}$. If for any such $\mathcal{Z}$, any positive number ϵ and any function $f \in C(\mathcal{Z})$, there is a function $g \in K(\mathcal{Z})$ such that $\|f - g\|_{\mathcal{Z}} \leq \epsilon$, then K is a universal kernel.

Hence, universal kernels are structurally capable of representing any continuous f through (9), even if the nonlinear structure of f, or equivalently, $G^{-1}(q)$, is unknown.

In practice, some information on f may be available, such as a measure of smoothness. This information can be exploited through *stationary* kernel functions, as explained in more detail in the next section.

Definition 4. (Stationary kernel function). A kernel $k(\mathbf{y}_{t_1}, \mathbf{y}_{t_2})$ is stationary if it can be expressed as a function of the difference between its inputs, i.e., $\mathbf{y}_{t_1} - \mathbf{y}_{t_2}$.

This choice of kernel leads to a data-dependent model (9) in which control effort values $u(t) = f(\mathbf{r}_t)$ are inferred from observations of $u(t)$ corresponding to similar sequences $\mathbf{y}_t \approx \mathbf{r}_t$, i.e., similar paths in $\mathcal{Y}$ require similar control effort values. The next section lists some examples of stationary kernels from a design perspective.

3.3 Kernel selection

This section lists two example kernels that pose a prior on the smoothness of the dynamics. The kernel functions are compared visually in Figure 3.

Smooth dynamics In this paper, dynamics are considered *smooth* if f is infinitely differentiable. A kernel that imposes smoothness on f by being infinitely differentiable is the *squared-exponential kernel*:

$$k(\mathbf{y}_{t_1}, \mathbf{y}_{t_2}) = \sigma_f^2 \exp\left(-\frac{1}{2}\rho\right), \tag{10}$$

where

$$\rho = (\mathbf{y}_{t_1} - \mathbf{y}_{t_2})^\top \Lambda^{-1} (\mathbf{y}_{t_1} - \mathbf{y}_{t_2}), \tag{11}$$

in which hyper-parameters $\sigma_f^2 = \operatorname{Var}(f(\mathbf{y}_t))$ relate to the maximum magnitude of the control effort and $\Lambda = \operatorname{diag}([\ell_1, \ldots, \ell_{n_\theta}])$ contains kernel length-scales ℓ_i. This kernel is universal, see Sriperumbudur et al. (2011).

Non-smooth dynamics Alternatively, f may be finitely differentiable, e.g., as a result of static friction. In this case, the Matèrn$_{3/2}$ kernel function can be used, which is

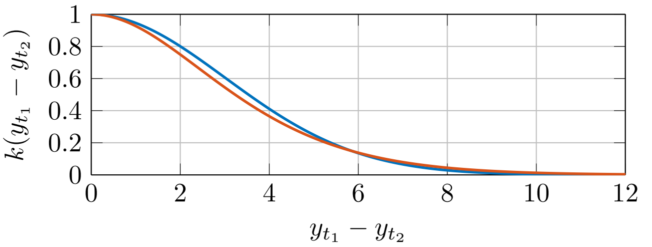

Fig. 3. Comparison of the covariance of a squared-exponential kernel (—), and a Matèrn$_{3/2}$ kernel (—) along one dimension of $\mathbf{y}_{t_1} - \mathbf{y}_{t_2}$. Similar output sequences $\mathbf{y}_t$ require similar control effort values $u(t)$, as can be seen from the high covariance near $y_{t_1} = y_{t_2}$.

able to represent non-smooth functions since it is only 1-time mean square differentiable (Rasmussen and Williams, 2006, Section 4.2):

$$k\left(\mathbf{y}_{t_1}, \mathbf{y}_{t_2}\right) = \sigma_f^2(1 + \sqrt{3}\rho)\exp(-\sqrt{3}\rho). \tag{12}$$

This is a universal kernel as well, see Sriperumbudur et al. (2011).

3.4 Hyper-parameter tuning

The kernel hyper-parameters $\Theta = \{\sigma_n\ \sigma_f, \ \ell_i\}$ can be optimized automatically by maximization of the log-marginal likelihood (Rasmussen and Williams, 2006, Section 5.4). This is the probability of the data given the model, defined as

$$\log p(\mathbf{u} \mid Y, \mathbf{\Theta}) = -\frac{1}{2}\mathbf{u}^\top K_n^{-1}\mathbf{u} - \frac{1}{2}\log|K_n| - \frac{M}{2}\log 2\pi, \tag{13}$$

with $K_n = K + \sigma_n^2 I$ and $|K_n| := \det(K_n)$. A local maximizer $\hat{\Theta}$ of the non-convex reward (13) is obtained through active-set optimization (Papalambros and Wilde, 2017, Section 7.4).

3.5 Convergence

As the density β of the data tends to infinity, expressed in number of observations ($\mathbf{y}_t$,$u(t)$) per unit of $\mathbf{y}_t$-space (e.g., [m$^{n_\theta}$]), the posterior mean of a GP with a universal stationary kernel converges to the true function (Rasmussen and Williams, 2006, Section 7.1). In the setting described in Section 2.1, it not possible to obtain a data-set of observations $\mathbf{y}_t$ distributed uniformly over $\mathcal{Y}$, since for sampled motion systems, each observation exhibits $y(t+1) \approx y(t)$. Still, given that the observations of $\mathbf{y}_t$ are close to $\mathbf{r}_t$ in $\mathbb{R}^{n_\theta}$, it can be derived from Sollich and Williams (2005, Section 1) that

$$\lim_{M\to\infty} \sum_{i=1}^{M} \alpha_i k(\mathbf{y}_i, \mathbf{r}_t) = f(\mathbf{r}_t), \tag{14}$$

if k is a stationary universal kernel with sufficiently small length-scales ℓ_i. The next section describes how observations of $\mathbf{y}_t \approx \mathbf{r}_t$ can be obtained such that the posterior approximately converges according to (14).

3.6 Experiment design for task flexibility

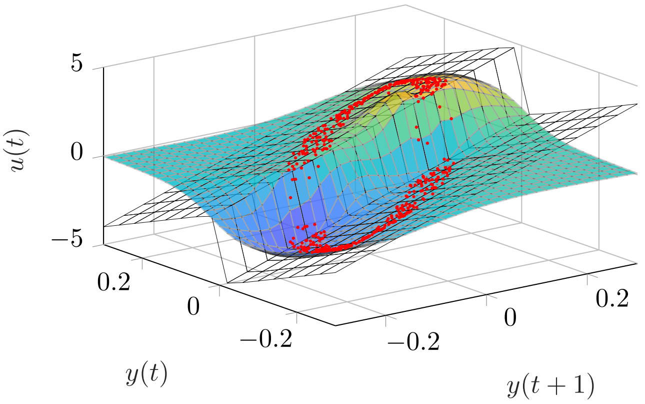

Fig. 4. Nonlinear example system of a mass with friction described by $u(t) = \frac{m}{T_s}(y(t+1) - y(t)) + F_c \operatorname{sign}(y(t))$ (white mesh), modeled as a GP. The posterior mean (coloured mesh) is computed using a dataset containing observations ($\mathbf{y}_t$, $u(t)$) (red dots) extracted from some periodic trajectories. The posterior mean is only accurate near the data, i.e., only feedforward signals for similar references can be predicted accurately.

By virtue of the system parametrization in terms of a stationary covariance function, predictions of $f(\mathbf{r}_t)$ are inferred from observations $f(\mathbf{y}_t)$ of similar output sequences $\mathbf{y}_t \approx \mathbf{r}_t$. Hence, this method allows for task flexibility in so far that feedforward signals for different reference sequences $\mathbf{r}_t$ can be predicted accurately if similar sequences are present in $\mathcal{D}$.

To collect a dataset that allows for the generation of feedforward signals for flexible tasks, it is proposed to design several signals $\tilde{\mathbf{r}}_j \in \tilde{\mathcal{R}}$ to be used in closed-loop experiments to obtain $\mathcal{D}$, see Figures 2 and 4.

Remark 1. In closed-loop, the noise $\varepsilon(t)$ introduces correlation between u and y through feedback, leading to bias in the estimate of G^{-1}. This bias could be reduced by repeating each experiment such as to observe a different realization of the noise.

The proposed method is summarized in Procedure 1. It is stressed that this procedure contains no recursion and hence there can be no issues with stability. Whether the posterior mean $\mathbb{E}[f]$ converges to the true f (see Section 3.5) now solely depends on the chosen $C(q)$, $F(q)$ and $\tilde{\mathcal{R}}$. The next section shows that even with simple choices of $C(q)$, $F(q)$ and $\tilde{\mathcal{R}}$ the procedure can lead to improved tracking performance.

Procedure 1: GP-based feedforward for nonlinear systems

```
Result: Feedforward signal $\mathbf{u}_{\text{ff},i}$ for every reference
        $r_i \in \mathcal{R}$.
Input: references $\mathcal{R}$, signals $\tilde{\mathcal{R}}$, initial feedforward
  controller $F(q)$, feedback controller $C(q)$.
Define a suitable kernel function $k$;
Define $n_c$, $n_{ac}$;
for each signal $\tilde{r}_j$ in $\tilde{\mathcal{R}}$ do
    Conduct a closed-loop experiment with $\tilde{r}_j$ to
      obtain observations $\mathbf{y}_t$ and $u(t)$;
end
Format the data to obtain $\mathcal{D} = \{Y, \mathbf{u}\}$ using (6);
Optimize the hyper-parameters, see Section 3.4;
for each reference $r_i$ in $\mathcal{R}$ do
    Compute the posterior mean $\mathbf{u}_{\text{ff},i} = \mathbb{E}[\mathbf{f}(R)]$ using
      (8).
end
```

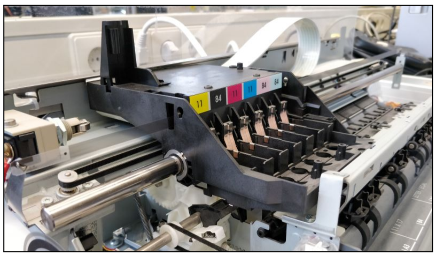

Fig. 5. The desktop printer used for experimental validation. The motor on the bottom left drives the pulley, which is connected to the print-head by the toothed belt.

4. EXPERIMENTAL VALIDATION

The proposed GP-based method for nonlinear feedforward control is applied to a desktop printer with friction, to demonstrate its ability to learn nonlinear dynamics for flexible tasks.

4.1 Setting and goal

The experimental set-up consists of a current-controlled A3 printer subject to static friction, depicted in Figure 5, connected to a computer running Simulink with a sample frequency of 1 kHz. A feedback controller $C(q)$ is available, given by

$$C(q) = \frac{108.6q^3 + 112.9q^2 - 100q - 104.3}{q^3 - 0.6499q^2 - 0.9465q + 0.7035}, \tag{15}$$

as well as an initial feedforward controller $F(q)$ of the form

$$F(q) = 2.8531\frac{q-1}{qT_s} + 0.083\left(\frac{q-1}{qT_s}\right)^2, \tag{16}$$

i.e., a controller with velocity and acceleration as basis. The goal is to track two third-order references r_1 and $r_2 = 1.05r_1$ of length $N = 4501$ samples, depicted in Figure 6, using the closed-loop scheme of Figure 2. To this end, the GP-based approach described in Section 3 is applied to learn a new feedforward controller $\hat{F}(q)$.

4.2 Approach

Following Procedure 1, a kernel function is defined first. The Matèrn$_{3/2}$ kernel (12) is chosen, because it is expected that static friction leads to a non-smooth function $f(\mathbf{y}_t)$. The number of samples history and preview are defined as $n_c = 20$ and $n_{ac} = 40$ respectively.

Subsequently, 11 signals $\tilde{r}_j$ are designed as scaled variations of r_1, such that

$$\tilde{\mathcal{R}} : \tilde{r}_j = a_j r_1, a_j \in [0.90, 0.92, \ldots, 1.10]. \tag{17}$$

Note that this leads to $\tilde{r}_6 = r_1 \in \mathcal{R}$, i.e., the first final reference is used during training. On the other hand, the second final reference r_2 is not used during training.

Each signal $\tilde{r}$ is used for one closed-loop experiment using $C(q)$ and $F(q)$, the result of which is shown in Figure 6. The observations of $u(t)$ and $y(t)$ are used to form $\mathcal{D}$ using (6), assuming $u(\tau) = y(\tau) = 0\ \forall\tau < 0,\ \tau > N$. For computational reasons, the size of the dataset is reduced by only taking every 30 rows of Y and $\mathbf{u}$ into account to form $\mathcal{D}$. It was found that this hardly affected the achieved performance, since many adjacent rows of Y contain similar information. This leads to a dataset of $M = 2970$ observations of $\mathbf{y}_t$ and $u(t)$. The kernel hyper-parameters are optimized by maximization of (13) and the feedforward signals $\mathbf{u}_{\text{ff},1}$, $\mathbf{u}_{\text{ff},2}$ are computed with (8). The hyper-parameter optimization and the matrix inversion in (8) require roughly 90 minutes of computational time on a personal computer.

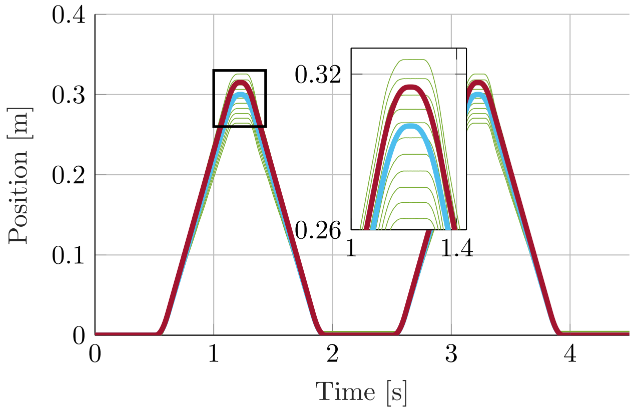

Fig. 6. References r_1 (—) and r_2 (—) and observed outputs y_i (—) from 11 closed-loop experiments using signals $\tilde{r}_i$.

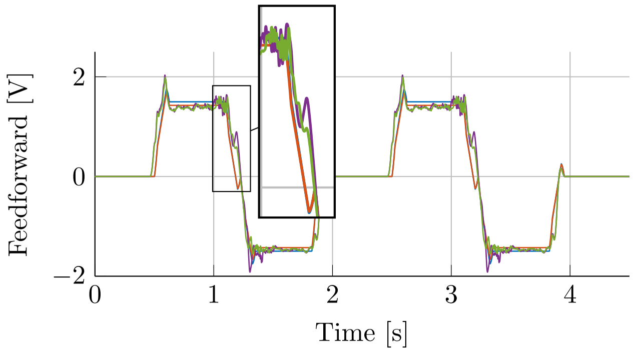

Fig. 7. Feedforward signals for r_1 produced by $F(q)$ (—) and by the GP-based method (—), and for r_2 by $F(q)$ (—) and by the GP-based method (—). The GP leads to a more complex signal, possibly to compensate for position-dependent static friction.

4.3 Results

The GP-based feedforward signals are shown in Figure 7. It can be seen that the GP-based feedforward signals exhibit more complex variations than the signal resulting from $F(q)$, which may indicate that nonlinear dynamics such as position-dependent friction are learned. Moreover, the GP-based feedforward signal acts earlier to a changing reference than $F(q)$, since it has $n_{ac} = 40$ samples preview, whereas $F(q)$ has no preview.

The feedforward signal is applied in closed-loop and the resulting error for r_1 is shown in Figure 8. The results are summarized in Table 1. It can be readily seen that the tracking error is improved significantly. In particular, the linear feedforward controller $F(q)$ leads to a large tracking error around $t = 2$ s since it stops at an incorrect position, as a result of static friction. The nonlinear GP-based feedforward controller $\hat{F}(q)$, on the other hand, appears to have learned these dynamics, since the error at this point in time is reduced by a factor 12.

To demonstrate the ability of the method to deal with task flexibility, Figure 9 shows the achieved performance for a reference r_2 not used during training. The tracking error

Table 1. Tracking errors with different feedforward signals.

		With $r_1 \in \tilde{\mathcal{R}}$	With $r_2 \notin \tilde{\mathcal{R}}$
$\|e\|_2$	$F(q)$	170 [mm]	164 [mm]
	$\hat{F}(q)$	68 [mm]	86 [mm]
$\|e\|_\infty$	$F(q)$	5.6 [mm]	4.2 [mm]
	$\hat{F}(q)$	3.2 [mm]	3.6 [mm]

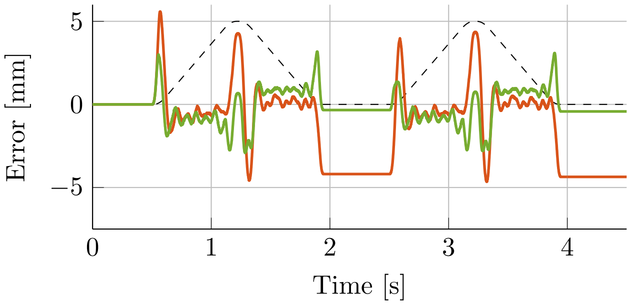

Fig. 8. Error $e_1 = r_1 - y_1$ using $F(q)$ (—) and the GP-based feedforward signal (—), along with the scaled reference (- -).

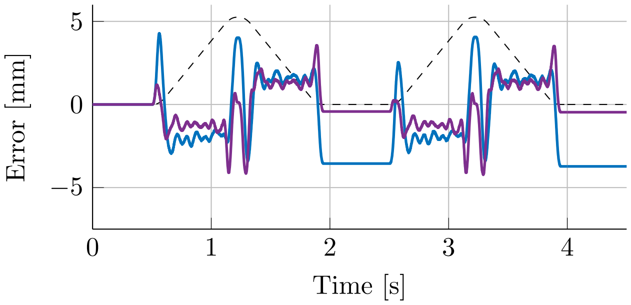

Fig. 9. Error $e_2 = r_2 - y_2$ using $F(q)$ (—) and the GP-based feedforward signal (—), along with the scaled reference (- -). While r_2 was not used during training, the error is still reduced by the GP-based approach.

is reduced to a similar extent as with r_1. This indicates that the method allows for the generation of feedforward signals for flexible tasks, provided that these tasks (or references) are sufficiently similar to the observed outputs in $\mathcal{D}$. This condition appears to be satisfied, as can be seen from Figure 6.

5. CONCLUSIONS AND FUTURE WORK

A method for inverse model control using Gaussian Process regression is proposed that allows for learning nonlinear dynamics of unknown structure. It is shown experimentally that the approach can lead to improved tracking performance with respect to linear feedforward. Since this method requires tasks to be similar to observed outputs in the dataset, it can be particularly useful in situations when many slightly different references need to be tracked. Moreover, since the developed approach makes use of the available controllers $C(q)$ and $F(q)$ to construct an improved $\hat{F}(q)$, it can be seen as an add-on that is applicable without changing the control architecture.

Possible extensions include the application to multi-input multi-output systems, and a formal means to take output noise into account. Finally, the achievable performance can possibly be improved further by applying the method iteratively, i.e., make use of the GP-based feedforward filter to iteratively obtain a better data-set. The convergence and stability properties of such an approach are subject to future research.

REFERENCES

Åström, K.J., Hagander, P., and Sternby, J. (1984). Zeros of sampled systems. *Automatica*, 20(1), 31–38.

Beckers, T., Kulić, D., and Hirche, S. (2019). Stable Gaussian process based tracking control of Euler–Lagrange systems. *Automatica*, 103, 390–397.

Blanken, L. and Oomen, T. (2020). Kernel-based identification of non-causal systems with application to inverse model control. *Automatica*, 114, 108830.

Boeren, F., Oomen, T., and Steinbuch, M. (2015). Iterative motion feedforward tuning: A data-driven approach based on instrumental variable identification. *Control Engineering Practice*, 37, 11–19.

Deisenroth, M. and Rasmussen, C. (2011). PILCO: A Model-Based and Data-Efficient Approach to Policy Search. In *Proceedings of the 28th International Conference on Machine Learning*, 465–472. Bellevue, WA.

Devasia, S. (1997). Stable inversion for nonlinear systems with nonhyperbolic internal dynamics. In *Proceedings of the 36th IEEE Conference on Decision and Control*, volume 3, 2882–2888. IEEE.

Micchelli, C.A., Xu, Y., and Zhang, H. (2006). Universal kernels. *Journal of Machine Learning Research*, 7, 2651–2667.

Nguyen-Tuong, D., Peters, J., Seeger, M., and Scholkopf, B. (2008). Learning inverse dynamics: A comparison. In *ESANN 2008 Proceedings, 16th European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning*, 13–18.

Nguyen-Tuong, D., Seeger, M., and Peters, J. (2009). Model learning with local Gaussian process regression. In *Advanced Robotics*, volume 23, 2015–2034.

Papalambros, P.Y. and Wilde, D.J. (2017). *Principles of Optimal Design*. Cambridge University Press.

Pavlov, A. and Pettersen, K.Y. (2008). A new perspective on stable inversion of non-minimum phase nonlinear systems. *Modeling, Identification and Control*, 29(1), 29–35.

Pillonetto, G., Dinuzzo, F., Chen, T., De Nicolao, G., and Ljung, L. (2014). Kernel methods in system identification, machine learning and function estimation: A survey. *Automatica*, 50(3), 657–682.

Rasmussen, C. and Williams, C. (2006). *Gaussian processes for machine learning*. London, England.

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

Sjöberg, J., Zhang, Q., Ljung, L., Benveniste, A., Delyon, B., Glorennec, P.Y., Hjalmarsson, H., and Juditsky, A. (1995). Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31(12), 1691–1724.

Sollich, P. and Williams, C.K.I. (2005). Understanding Gaussian Process Regression Using the Equivalent Kernel. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 3808 LNCS, 211–228.

Sriperumbudur, B.K., Fukumizu, K., and Lanckriet, G.R. (2011). Universality, characteristic kernels and RKHS embedding of measures. *Journal of Machine Learning Research*, 12, 2389–2410.

Van De Wijdeven, J. and Bosgra, O.H. (2010). Using basis functions in iterative learning control: Analysis and design theory. *International Journal of Control*, 83(4), 661–675.