Gaussian Process Based Predictive Control for Periodic Error Correction

Edgar D. Klenske, Melanie N. Zeilinger, Bernhard Schölkopf, and Philipp Hennig

***Abstract*—Many controlled systems suffer from unmodeled nonlinear effects that recur periodically over time. Model-free controllers generally cannot compensate these effects, and good physical models for such periodic dynamics are challenging to construct. We investigate nonparametric system identification for periodically recurring nonlinear effects. Within a Gaussian process regression framework, we use a locally periodic covariance function to shape the hypothesis space, which allows for a structured extrapolation that is not possible with more widely used covariance functions. We show that hyperparameter estimation can be performed online by using the maximum a-posteriori point estimate, which provides an accuracy comparable to sampling methods as soon as enough data to cover the periodic structure has been collected. It is also shown how the periodic structure can be exploited in the hyperparameter optimization. The predictions obtained from the Gaussian process model are then used in a model predictive control framework to correct for the external effect. The availability of good continuous predictions allows control at a rate higher than that of the measurements. We show that the proposed approach is particularly beneficial for sampling times that are smaller than, but of the same order of magnitude as, the period length of the external effect. In experiments on a physical system, an electrically actuated telescope mount, this approach achieves a reduction of about 20 % in root mean square tracking error.**

I. Introduction

SCREWS AND GEARS are not the only source of periodically recurring errors in dynamical systems. Every system that is tied to the ubiquitous day- and night cycle (like building control or energy systems) or to recurring movements (like a beating heart or a satellite) suffers from periodic errors. Since these effects are often small relative to the required control precision, they are in practice usually neglected in the controller design. For high precision control systems, however, such errors can be the dominant source of problems. Correcting errors only after they are measured leads to a delay in the error correction. If these errors can be anticipated, the control performance can be significantly improved. While errors arising stochastically cannot be preempted systematically, periodic effects are amenable for prediction: Since their future resembles their past, extrapolation (prediction) is easier and more structured. Based on this idea, we present a framework for identification and control of periodic effects. Our framework continually performs identification at runtime, and is thus applicable to stochastic time varying systems.

The correction of periodic errors has repeatedly been studied. The "Very Large Telescope" uses an internal parametric model for the known error sources, and a Kalman filter (e.g. [1]) as an estimator for the model parameters [2]. High precision tracking of spacecrafts on periodic trajectories was addressed in [3], based on predictive filtering using an extended Kalman filter (e.g. [1]). To predict the beating motion of a human heart, extended Kalman filtering for state estimation was used in [4], allowing the nonlinear model to change over time. Concerning the use of learning based models for control, there is a wide range of literature available in the context of adaptive control. For methods based on model predictive control see e.g. the recent work [5], [6], [7].

In contrast to previous methods for periodic error correction, the approach presented here does not rely on a pre-specified finite-dimensional model class. Instead, we propose a nonparametric framework based on Gaussian process (GP) regression that is frequently used for system identification (see e.g. [8] for a survey). It is closely related to least-squares regression, which is the most commonly employed technique in system identification, but is based on a probabilistic interpretation, which can be used to guide exploration during identification [9]. There is recent work on using GPs for state filtering [10] and on modeling and control of nonlinear systems [8], [11]. The idea of using the learned model in predictive control is conceptually similar to [5], [6], [12], with the key difference that we use a GP to predict time varying effects. The textbook [13] provides a general introduction to Gaussian processes.

A Gaussian process model is parametrized by two objects: mean and covariance function. When used in system identification, in particular the choice of covariance function has a strong effect on performance, and requires consideration of the particular dynamics to be identified. Although the literature knows "universal" covariance functions that can technically approximate any continuous function [14], [15], this notion only applies in the infinite limit, and can be subject to extremely slow (logarithmic) convergence [16], [17]. Hence, the choice of the covariance function is often critical in practice.

In this work, the focus lies on the identification of quasi periodic systems (§II-A) with Gaussian processes (§II-B) by using a specific model class involving periodicity (§III-A). A key element of GP regression is the estimation of the hyperparameters (§III-B), which is performed by exploiting the structure of the problem at hand (§III-C). We focus on the case where the system up to the periodic effect is linear and

E.D. Klenske, M.N. Zeilinger, B. Schölkopf and P. Hennig are with the Max Planck Institute for Intelligent Systems, Spemannstraße 38, 72076 Tübingen, Germany. e-mail: {edgar.klenske, bernhard.schoelkopf, philipp.hennig}@tuebingen.mpg.de

M.N. Zeilinger is with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, CA 94720, USA. e-mail: melanie.zeilinger@eecs.berkeley.edu

The research of M.N. Zeilinger has received funding from the EU FP7 under grant agreement no. PIOF-GA-2011-301436-COGENT.

use model predictive control to achieve optimal closed-loop performance (§IV-A). A reformulation in the form of a tracking problem is proposed, which offers simple implementation and facilitates analysis of the control performance (§IV-B). To show the qualitative properties of this framework, we apply it to a toy problem first (§V-B). As the development of this method was driven by a real problem in astronomy, the method is evaluated on this problem both in simulation (§V-C1, §V-C2) and hardware experiments (§V-C3). While GPs and MPC are well studied techniques, the main contribution of this work is a combination that is tailored to quasi periodic functions, allowing for extrapolation and efficient computation, which is crucial for online identification and control. A practical technique for hyperparameter optimization is proposed and the effectiveness of the approach for compensation of quasi periodic errors is demonstrated on the experiment of a telescope mount.

This paper extends the preliminary conference version on the same topic [18]. We here provide a more extensive treatment of hyperparameter estimation, discuss Markov-Chain-Monte-Carlo (MCMC) versus maximum a-posteriori (MAP) estimation, and propose a custom optimization method for the latter. Additional experiments include an analysis of system behavior under sensor failure.

II. Preliminaries

A. Problem statement

Consider the dynamical system

$$\dot{x}(t) = Ax(t) + Bu(t) + g(t) \tag{1}$$

composed of a linear system with dynamic matrices A, B and a nonlinear time varying function $g : \mathbb{R}_+ \to \mathbb{R}^E$, where $x \in \mathbb{R}^E$ denotes the state and $u \in \mathbb{R}^F$ the input. For simplicity, full state measurement is assumed. The structure of (1) could be chosen more generally – Gaussian process models can also learn nonlinear functions of the state and input. We opt for this linear formulation with nonlinear external reference here to keep the resulting control problem conceptually clear and computationally simple. If needed, the definition can also be adapted to a nonlinear system using a nonlinear model predictive control technique [19].

The function g captures nonlinear time dependent effects, in particular we will focus on systems exhibiting some form of periodic behavior. Systems with such time dependent errors of periodic characteristic appear in different application areas, such as building temperature control, beating-heart surgery or electrical power grids. For strictly periodic functions, there exists a constant period λ, such that $g(t + n\lambda) = g(t)$ for $n \in \mathbb{N}$. However, error sources in real systems are often not perfectly periodic in this sense, they show various forms of phase-shift, deformation and desynchronization. To address this issue, we generalize to consider locally periodic functions. These are functions for which $g(t) \approx g(t + n\lambda)$ for $n\lambda \ll \ell$ and $g(t) \not\approx g(t + n\lambda)$ for $n\lambda \gg \ell$, where ℓ is some measure of temporal locality. Intuitively speaking, local periodicity means that the periodicity is not strict, i.e. variations are allowed. In particular, this covers functions that vary on a slower time-scale, e.g. a decaying oscillation or an oscillation with long-term change in shape.

We consider the case where a linear model (matrices A and B) is available and the goal is to infer the disturbance function $g(t)$ online from measurements. This is motivated by the fact that often a nominal model is derived either from physical considerations or an offline system identification step.

At every measurement time t_k, the system goes through the following process:

1) Measure $x(t_k), \dot{x}(t_k)$
2) Construct observation for $g(t_k)$, update model for $g(t)$
3) Compute control input $u(t_k) = u(x(t_k), g(t_k))$ and apply to system (1)

It should be noted that in practice, measurements of $\dot{x}$ are generally not available and will be approximated numerically, see also the experimental details in §V.

At this point, it should be intuitively clear that the performance gain one can expect from the use of a periodic model for feed-forward compensation depends on the sampling rate of the control system: If the external error is slow compared to the measurement rate, a locally linear model is sufficient. But if the external error is on the same time scale as the measurement, it helps to use feed-forward control based on GP predictions. With the presented approach it is even possible to choose the control interval smaller than the actual measurement interval. See §V-C1 for a more detailed discussion.

B. Gaussian process regression

For notational simplicity, this section only covers the scalar case of Gaussian processes. In the case of a vector-valued function g, one GP is trained for every dimension. For our purposes, a Gaussian process $\mathcal{GP}(g; \mu, k)$ is an infinite-dimensional probability distribution over the space of real-valued *functions* $g : \mathbb{R} \to \mathbb{R}$, such that every finite, D-dimensional linear restriction to function *values* $g(T) \in \mathbb{R}^D$ (measurements) at times $T \in \mathbb{R}^D$ (measurement locations) is a D-variate Gaussian distribution $\mathcal{N}(g(T); \mu(T), k(T, T))$. It is parametrized by a mean function $\mu(T) : \mathbb{R}^D \to \mathbb{R}^D$, and a covariance function $k(T_1, T_2) : \mathbb{R}^{D_1} \times \mathbb{R}^{D_2} \to \mathbb{R}^{D_1 \times D_2}$. The mean has a relatively straightforward role; it simply shifts predictions. The covariance function's responsibility is more intricate. It can be interpreted as a similarity measure over function values in $\mathbb{R}^D$ and controls the shape of the Gaussian process belief in the space of functions. It has to be chosen such that, for any $T_1, T_2 \in \mathbb{R}^D$, the matrix $k(T_1, T_2) \in \mathbb{R}^{D \times D}$, also known as the *kernel*, is positive semidefinite.

If data points z_i are observed at times t_i with Gaussian noise ε

$$z_i = g(t_i) + \varepsilon \qquad \varepsilon \sim \mathcal{N}(0, \sigma^2), \tag{2}$$

then the point-wise posterior distribution under the Gaussian process prior and the Gaussian likelihood encoding this observation is also Gaussian [13, §2.2], with the posterior mean and variance

$$\mu(t) = k_{tT}^\intercal K^{-1} \mathbf{z} \tag{3}$$

$$\Sigma(t) = k_{tt} - k_{tT}^\intercal K^{-1} k_{Tt}, \tag{4}$$

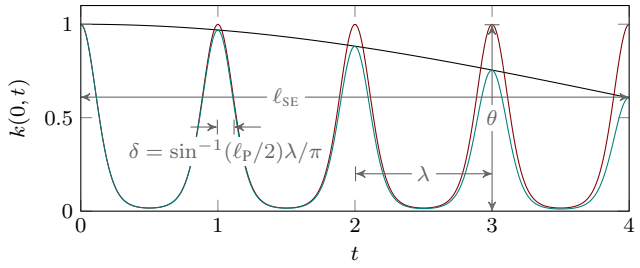
where $K = k(T,T) + \sigma^2 I$ is the kernel Gram matrix with observation noise σ^2, $\mathbf{z}$ is the vector of observations at time points T and t is the prediction time. For notational brevity, we use the short-hand $k_{ab} := k(a,b)$.

GP regression is a general framework for nonlinear regression. As mentioned above, in the context of our particular setup, it may in fact also be used to construct probabilistic models for fully nonlinear systems $g = g(x,u,t)$, without major changes. For the purposes of this paper, however, we focus on the simpler case of $g = g(t)$, allowing for direct incorporation in stochastic predictive control techniques.

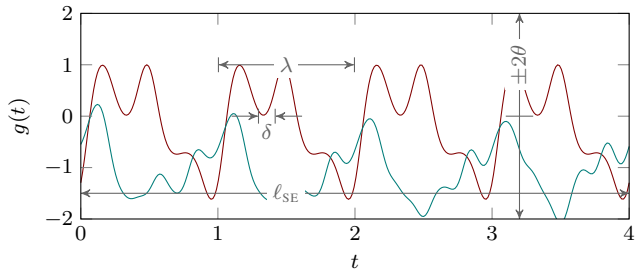
III. GPs for quasi periodic functions

This section will propose a covariance function suitable for the identification of quasi periodic functions and present an efficient technique for hyperparameter optimization.

A. Quasi periodic covariance function



(a) Covariance functions



(b) Samples from GPs with these covariance functions

Fig. 1. (a): The compound kernel k_{C} (green) of (7) is the product of k_{SE} (black, (5)) and k_{P} (red, (6)). (b): Samples drawn from Gaussian process priors using these covariance functions (same colors). Samples using the periodic kernel are perfectly periodic, while samples using the compound kernel are only periodically similar on a scale controlled by the parameter ℓ_{SE} of (7). This local periodicity can be used to increase modeling flexibility.

The way to construct a periodic hypothesis class, and the central idea of this work, is to construct a covariance function that focuses prior probability mass on locally periodic functions. Among the most popular kernels for regression purposes is the square-exponential (a.k.a. Gaussian, radial basis function) kernel

$$k_{\text{SE}}(t,t';\ell_{\text{SE}}) = \exp\left(-\frac{(t-t')^2}{2\ell_{\text{SE}}^2}\right), \tag{5}$$

with length-scale ℓ_{SE}. This kernel gives a stationary model which does not allow for structured extrapolation. MacKay [20] proposed a periodic covariance function, based on a sine-transformation of the input:

$$k_{\text{P}}(t,t';\ell_{\text{P}},\lambda) = \exp\left(-\frac{2\sin^2\left(\frac{\pi}{\lambda}(t-t')\right)}{\ell_{\text{P}}^2}\right), \tag{6}$$

with length-scale ℓ_{P} and period-length λ. Function values $g(t), g(t')$ jointly sampled from Gaussian process priors with this covariance function are perfectly correlated if $|t-t'| = \lambda$, resulting in identical function values for points that are one period length apart. Thus sampled functions are perfectly periodic with period λ. Within this period length, samples vary on a typical regularity length scale of $\delta = \sin^{-1}(\ell_{\text{P}}/2)\lambda/\pi$, over a range with standard deviation $\theta = 1$ (Fig. 1).

For many systems, strict periodicity is too strong an assumption. For example, the weather is stochastic, which has an effect on the periodic temperature in a building. Therefore, modeling external errors with perfect periodicity can lead to severe overfitting and low extrapolation performance. To weaken the perfect correlation, we use the fact that the kernel property is closed under multiplication and addition (i.e. kernels form a semiring, [13, §4.2.4]): Although the product of two Gaussian processes is not a Gaussian process, the product of two kernel functions is also a kernel and therefore a valid covariance function for a Gaussian process.

This property of covariance functions is useful to construct composite covariance functions either automatically from data [21] or manually. Since we have access to physical knowledge about the system, a suitable kernel is constructed in a qualitative manner. Multiplying the periodic kernel with a broad square-exponential gives another kernel whose corresponding Gaussian process condenses mass at functions that change over time:

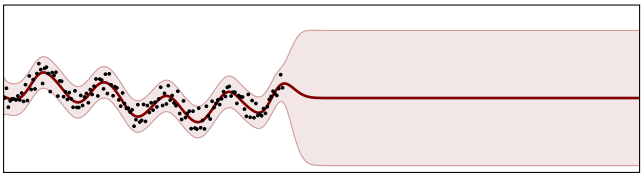
$$k_{\text{C}}(t,t';\theta^2,\ell_{\text{SE}},\ell_{\text{P}},\lambda) = \theta^2 \cdot k_{\text{SE}}(t,t';\ell_{\text{SE}}) \cdot k_{\text{P}}(t,t';\ell_{\text{P}},\lambda), \tag{7}$$

with signal variance θ^2 and the other parameters as stated above. This kernel considers two input times similar if they are similar under both the square-exponential *and* the periodic kernel. If $\ell_{\text{SE}} \gg \lambda$, this allows encoding a decay in the covariance over several oscillations. The different covariance functions are shown in Fig. 1a, exemplary randomly sampled functions from Gaussian processes with those covariance functions are shown in Fig. 1b. The posterior mean of GPs with aperiodic and periodic covariance, trained on periodic data, is shown in Fig. 2. In the region far away from data points the predictions are equal, whereas close predictions show significantly more structure with the (locally) periodic kernel.

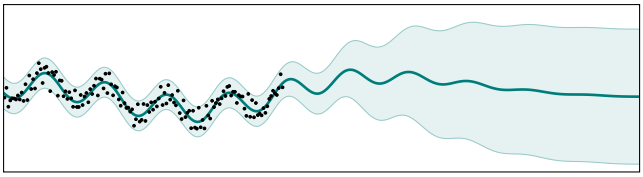
Fig. 2b illustrates the key benefit of this approach: With increasing distance from data, the prediction degrades gracefully back to the zero mean. If a (not perfect) periodicity would be predicted with a purely periodic kernel, prediction and reality could run out of phase over time, leading to bad predictive performance. The proposed locally periodic covariance function circumvents this problem.

B. Hyperparameter estimation

The combined kernel (7) has four *hyperparameters*, which will henceforth be subsumed in the vector $\eta := (\theta^2, \ell_{\text{SE}}, \ell_{\text{P}}, \lambda)$. This includes the period length λ of the periodicity. Inferring



(a) SE-kernel



(b) Locally periodic kernel

Fig. 2. Comparison of Gaussian process posteriors (posterior mean function as thick line, shaded region covers two marginal standard deviations) arising from the same periodic data (black) for the square-exponential kernel (top) and a product of periodic and square-exponential kernel (bottom). The locally periodic kernel provides a richer extrapolation, which can be used for improved predictive control.

good values for η is important for good modeling performance. The fundamental framework for such inference is provided by Bayes' theorem. The likelihood for observations $\mathbf{z}$ at times T, conditioned on the parameters η, can be found by marginalization over the unknown function g, which is feasible because both $p(\mathbf{z}|g)$ and $p(g|\eta)$ are Gaussian distributions:

$$\begin{aligned} p(\mathbf{z}|T,\eta) &= \int p(\mathbf{z}|g)p(g|\eta)dg \\ &= \int \mathcal{N}(\mathbf{z}; g_T, \sigma^2\mathbf{I})\mathcal{GP}(g;0,k(\eta))dg \\ &= \mathcal{N}(\mathbf{z};0,K(\eta)). \end{aligned} \tag{8}$$

In order to make the method more robust, priors on the parameters are introduced. For the strictly positive variables $\theta^2, \ell_{\text{SE}}, \ell_{\text{P}}, \lambda$ that constitute η, Gamma priors are a classic choice (see e.g. [22]):

$$p(\eta|\kappa,\tau) = \prod_i \frac{\eta_i^{\kappa_i - 1}\exp(\frac{\eta_i}{\tau_i})}{\Gamma(\kappa_i)\tau_i^{\kappa_i}}, \tag{9}$$

where κ_i and τ_i are tuning-parameters. The choice of these parameters is however not critical, since the effect of the hyperparameter prior fades quickly once enough data points have been collected. However, even under these simple priors, Bayesian inference is not analytic, because of the complicated shape of the likelihood (8) as a function of η. Ideally, the goal would be to compute the marginal

$$p(g) = \int p(g|\eta)p(\eta)d\eta \tag{10}$$

using a prior density $p(\eta)$ [13]. However, this approach is intractable and can only be performed approximately at high computational expense. One (comparably elaborate and precise) way to approximate the posterior distribution over η and to marginalize over the unknown parameters η is sampling, using a Markov chain Monte Carlo (MCMC) method. We found *shrinking-rank slice sampling* [23] to be particularly well-suited for this task and implemented the method accordingly.

The left column of Fig. 3 shows the resulting marginals on the function g at two different points in the learning process. Not unexpectedly, the posterior uncertainty is high after only a few observations, in particular after less than one full period, but collapses to a highly confident distribution after several periods of observations have been collected.

In the latter case of high certainty, when all samples from the posterior are close to each other, their entirety is represented well by a single point estimate. Such an estimate can be found in a computationally much less demanding process, by optimizing the posterior distribution, or even just the likelihood, to find the maximum a-posteriori (MAP) or maximum likelihood estimate, respectively [13, §5.4.1]. The latter, maximum likelihood approach is also known as *evidence maximization*, or "type-II maximum likelihood", in the literature, to distinguish it from the much more simplistic approach of fitting g itself by a maximum likelihood method.

Using the MAP estimate is one of the most widely studied and best understood strategies in statistics [24, §9.3 – §9.6]. It is not without weaknesses (e.g. the optimization is prone to get stuck in local minima), some of which are often resolved if enough data is available (Fig. 3) or by the use of customized optimization algorithms (III-C). Other approaches, for example integrating the hyperparameters over MAP estimates or cross validation, have been examined in the past and found to perform worse than the above type-II maximum likelihood approach in practice, see e.g. [25], [13, §5.4.2].

The maximization is easier to perform in log domain, in which the effect of the prior is additive, leading to the following optimization problem:

$$\begin{aligned} \eta^* &= \arg\max_\eta p(\eta|\mathbf{z},T) = \arg\max_\eta p(\mathbf{z}|T,\eta)p(\eta) \\ &= \arg\max_\eta \left(\log p(\mathbf{z}|T,\eta) + \log p(\eta)\right), \end{aligned} \tag{11}$$

where the logarithm of the likelihood is given by

$$\begin{aligned} &\log p(\mathbf{z}|T,\eta) = \\ &\quad -\frac{1}{2}\mathbf{z}^\intercal K(\eta)^{-1}\mathbf{z} - \frac{1}{2}\log|K(\eta)| - \frac{D}{2}\log 2\pi. \end{aligned} \tag{12}$$

In (11), the prior effectively turns into a regularizer, simplifying optimization and avoiding degeneracy. The additional computational cost is negligible compared to the matrix inversion needed for (12).

The right column of Fig. 3 shows the Gaussian process point estimates for g resulting from MAP inference. From the figure, it is clear that point estimation leads to a more limited, and generally overly confident extrapolation model, especially in early phases of learning, when the dataset does not yet cover several periods. However, MAP offers two advantages that make it attractive from an applied perspective: The first one is computational cost – MCMC sampling can be orders of magnitude more expensive than optimization for a MAP estimate. The second one is an algebraic one: MCMC estimates are mixtures of Gaussian process models (see Fig. 3). This means the overall dynamical model for the system defined by these models is a very challenging stochastic differential equation which cannot, in general, be

interpreted as a differential equation involving a Wiener process. In our implementation and experiments, we thus rely on the computationally much less taxing MAP inference.

C. Custom parameter optimization

The log-likelihood surface for η is, in general, not convex. Fig. 4 shows a slice through this surface along the periodicity parameter λ. Standard numerical optimizers will thus usually return suboptimal local extrema of this function. An interesting observation in our specific context is that the periodic structure of the covariance function and the data is reflected in this hyperparameter likelihood as well. The reason for this is a harmonic effect: If the data has a true period of λ, then periodic functions whose periodicity is an integer multiple of λ also fit the data well, resulting in low values of (12).

Intuitively, this can be compared to a function with periodicity λ, which could also be considered as a function with period length 2λ. To see this, recall from Fig. 1b that the periodic functions can have an arbitrary recurring pattern in each repetition. By inspecting (12), we can gain intuition and notice that the likelihood (12) is a nonlinear (deterministic) function of terms of the form (6). Since each of these terms is periodic in λ^{-1}, the overall function will show periodicity in that term.

This harmonic structure can be leveraged by means of a heuristic to increase numerical stability of the optimizer. We designed a customization of a numerical optimizer outlined in Alg. 1 that, after convergence to a local minimum $\eta = (\theta^2, \ell_{\text{SE}}, \ell_{\text{P}}, \lambda)$, also checks the function value at $\eta' = (\theta^2, \ell_{\text{SE}}, \ell_{\text{P}}, \frac{1}{2}\lambda)$. If the negative log likelihood at this location is lower, the optimizer is updated and the bisection is repeated. The locations that are iteratively proposed during the loop (line 4 of the algorithm) are shown as vertical lines in Fig. 4. This approach uses the (otherwise problematic) non-convex structure in the hyperparameter optimization to find better optima.

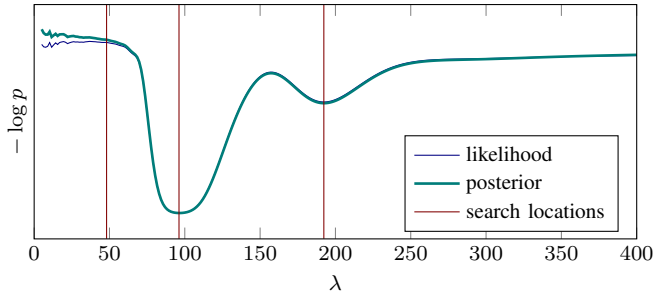


Fig. 4. Slice through the hypothesis space of hyperparameters along the dimension of the hyperparameter λ, defining the period length of g. Logarithm of the type-II (marginal) likelihood in thin blue, logarithm of posterior distribution (sum of log likelihood and vague log prior, not shown) in green. The shape of the log posterior is dominated by the likelihood, indicating that most prior assumptions are dominated by the observed data. The meaning of the vertical lines is explained in §III-C.

IV. GP PREDICTIONS IN MODEL PREDICTIVE CONTROL

Popular control frameworks supporting the incorporation of feed-forward model predictions include linear quadratic regulator (LQR) techniques, see e.g. [26], or model predictive control (MPC), see e.g. [27].

Algorithm 1 Customized optimization

1: $\eta := (\theta^2, \ell_{\text{SE}}, \ell_{\text{P}}, \lambda)$ ▷ initial guess
2: $\eta \leftarrow$ LOCALLY_OPTIMIZE(η) ▷ use std. optimizer
3: **loop**
4: $\quad \eta' \leftarrow (\theta^2, \ell_{\text{SE}}, \ell_{\text{P}}, \frac{1}{2}\lambda)$ ▷ make proposal
5: $\quad$ **if** nll(η') $>$ nll(η) **then** ▷ compare neg. log lik.
6: $\quad\quad$ **break** ▷ reject and leave loop
7: $\quad$ **else**
8: $\quad\quad \eta \leftarrow \eta'$ ▷ accept proposal
9: $\quad$ **end if**
10: **end loop**
11: **return** η

In this work we employ an online MPC framework, computing the optimal control input by solving an optimization problem for each measured state. This allows for direct incorporation and updating of the GP model (potentially also changing with every sampled state) as well as system constraints, such as input constraints.

A. Discrete-time MPC formulation

A discrete-time MPC approach is used based on the discretized model of (1)

$$x_{k+1} = A_d x_k + B_d u_k + a_k, \tag{13}$$

where A_d and B_d are obtained from a zero-order-hold discretization and a_k is a discretization of choice of g at time t_k.

Since the optimal control input is computed at each sampling time based on the current measured state, the model can be updated online. An important aspect and advantage of combining online learning of a continuous time function with MPC is the possibility to decouple the discretization from the sampling time. While in a standard MPC setup, unmodeled effects only become apparent through state measurements and therefore require fast sampling rates, the GP model captures these effects and provides a continuous prediction of their evolution in the future. As a result, the sampling time can be chosen as a multiple of the discretization or prediction interval without sacrificing performance by using the sequence of control inputs in between state measurements. It is clear that an upper bound on the sampling time is imposed by the prediction horizon.

Since the prediction from the Gaussian process model is stochastic and provides a distribution over future function values rather than one particular sequence, stochastic MPC methods offer a natural framework to incorporate the GP model and make use of the posterior model uncertainty. For an overview of recent stochastic MPC methods, see e.g. [28], [29] and the references therein. This is an important advantage over other nonparametric methods like kernel ridge regression or regularized least squares, which do not provide posterior uncertainty.

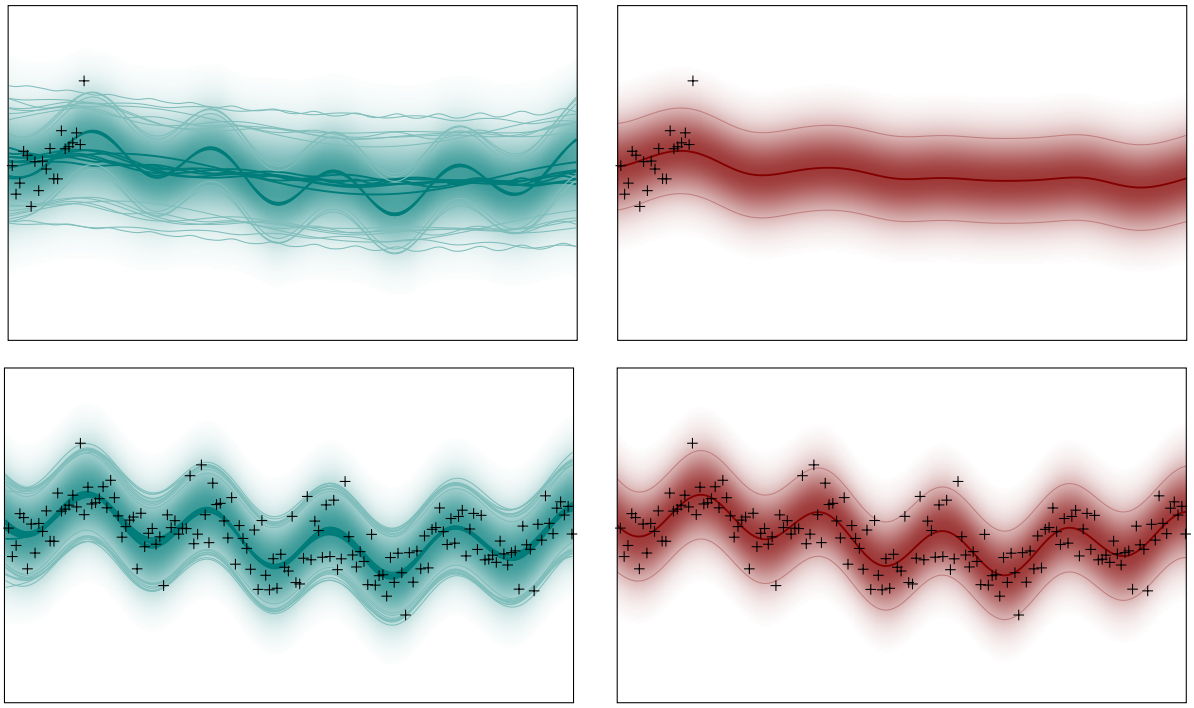


Fig. 3. Comparison of Markov Chain Monte Carlo inference (left column) with maximum a-posteriori point estimation (right column) on hyperparameters of the Gaussian process model. Top: initial phase of learning, after only a few observations. Bottom: convergence after observation of several periods.

A common cost function in stochastic MPC for regulating the system state to the origin is the expected value of the sum of stage costs

$$V(\mathbf{x}(p(g)), \mathbf{u}) := \mathbb{E}\left[\sum_{n=0}^{N-1} l(x_n(p(g)), u_n)\right] \tag{14}$$

where $p(g) = \mathcal{GP}(g(t); \mu, \Sigma)$ denotes the posterior distribution over the function g. If the stage cost l is taken to be quadratic (which is the case for most practical MPC problems) and since the inputs are deterministic and the GP posterior is Gaussian, the expected value is equivalent to using the mean of the state evolution, i.e. the mean GP prediction of $g(t)$ in dynamics (1). The most common stochastic MPC problem hence results in a deterministic formulation by using the GP posterior and reduces to the certainty equivalent (CE) controller.

We consider the case of a quadratic stage cost in the following. As the true function $g(t)$ is unknown, a discrete-time system describing the mean of the state trajectory is approximated by replacing a_k in (13) with $\tilde{a}_k$, obtained by discretizing the mean of the GP prediction $\tilde{g}$. $\tilde{\cdot}$ therefore denotes the mean estimate of the true function inferred from data and emphasizes that these are generally not the same. The resulting discrete-time system can directly be used in a standard MPC formulation.

B. Tracking MPC

We propose a different formulation in the following, transforming the regulation problem into a linear tracking problem. The state of the mean discrete time system at prediction time step $k+n$, starting from state x_k at time step k is given by:

$$\tilde{x}_{k+n} = A_d^n x_k + \sum_{m=1}^{n} A_d^{m-1} B_d u_{k+n-m} + \sum_{m=1}^{n} A_d^{m-1} \tilde{a}_{k+n-m}, \tag{15}$$

where $\tilde{a}_k$ is the discretization of the GP prediction $\tilde{g}$. This is the sum of the linear system and a system driven by $\tilde{a}_k$. It can be easily seen that the MPC problem for regulating system (13) to the origin can be reformulated as a tracking problem for the linear system, tracking a nonlinear reference.

The reference signal is generated from the GP prediction according to (15):

$$\mathbf{x}^{\text{ref}} = \begin{bmatrix} -\tilde{a}_{k+1} & \cdots & -\sum_{m=1}^{n} A_d^{m-1} \tilde{a}_{k+n-m} \end{bmatrix}^\intercal. \tag{16}$$

The resulting MPC problem is given by

$$\begin{aligned} (\mathbf{x}^*, \mathbf{u}^*) = \arg\min_{\mathbf{x},\mathbf{u}} \quad & \sum_{n=0}^{N-1} l(x_n - x_n^{\text{ref}}, u_n) && \text{(17a)} \\ \text{s.t.} \quad & x_0 = x(t_k) && \text{(17b)} \\ & x_{n+1} = A_d x_n + B_d u_n && \text{(17c)} \\ & u \in \mathbb{U} && \text{(17d)} \end{aligned}$$

where $\mathbb{U} \subseteq \mathbb{R}^F$ is a polytopic set defining the input constraints, $\mathbf{x} := [x_0, \ldots, x_N]$ is the state trajectory and similarly for $\mathbf{u}$. The resulting problem is a quadratic program that can be solved efficiently using available optimization software (e.g. [30]) or fast MPC techniques proposed in recent years, such

as code generation (e.g. [31]). Because this is an instance of a basic MPC technique, the standard properties of MPC apply. It also allows for a more principled analysis of the closed-loop properties: Extensions in the field of tracking MPC can be applied to ensure stability, such as reference governors, or the periodic MPC approach in [32]. Applying the modified tracking formulation in [32], convergence can be guaranteed if the model $\tilde{g}(t)$ converges to a periodic function.

Remark 1: The discrete-time model (13) with the predicted nonlinear term $\tilde{a}_k$ can in principle be used in any linear or linearized state estimation or control method based on state prediction. One example is the Kalman filter. The nonlinear prediction from the GP can be incorporated into the state prediction without complicating the Kalman filter equations. The measurement update of the Kalman filter remains unchanged. The GP predictions increase the performance by providing a better state estimate. This leads to smaller correction terms and smaller posterior variance.

Remark 2: Because the posterior of the Gaussian process is a Gaussian distribution, state constraints can be included in the form of soft constraints, penalizing the amount of constraint violation, or chance constraints, ensuring constraint satisfaction with a certain probability [33], [34].

Remark 3: The approach can also be applied to stage cost functions and constraints that do not allow for a deterministic representation using the GP model, e.g. a value-at-risk formulation involving the variance of the cost by using sample-based methods to approximate the stochastic MPC problem [35], [36]. GPs fit well in this framework by being generative models from which sample trajectories can be easily drawn.

Remark 4: Depending on the discretization used, a trade off between the mean and variance of a quadratic cost function can be formulated as a deterministic optimization problem using the posterior GP prediction:

$$\mathbb{E}\left[\frac{1}{2}x_n^\intercal Q x_n\right] = \mu_n^\intercal Q \mu_n + \operatorname{tr}(Q\Sigma_n), \tag{18a}$$

$$\mathbb{V}\left[\frac{1}{2}x_n^\intercal Q x_n\right] = \mu_n^\intercal Q \Sigma_n Q \mu_n + \frac{1}{2}\operatorname{tr}(Q\Sigma_n Q\Sigma_n). \tag{18b}$$

This is the case whenever the distribution $\mathcal{N}(\mu_n, \Sigma_n)$ of a_k can be directly obtained from the distribution of $g(t)$, e.g. in the case of Euler discretization.

V. Experiments

The presented method was implemented and used on different test problems. After providing a toy example that demonstrates how the state evolution is anticipated by the use of the GP prediction, the method is evaluated for a simulated telescope system where the performance under different measurement frequencies and under sensor failure is analyzed. Finally, the proposed method is tested in an experiment on a real telescope system, showing substantial improvements in control performance.

A. Implementation details

Since in practice the state and derivative cannot be measured directly, they have to be approximated from potentially noisy measurements. An observer (like the Kalman filter, e.g. [1]) can be used to estimate the state, which increases the performance, especially when the measurement noise is high.

Since the function to be inferred acts as additional input to the dynamics, observations of the disturbance are constructed from observations of the *change* in state, $\Delta x = x_{k+1} - x_k$. We therefore assume $g(t)$ to be piece-wise constant, here a zero-order-hold linearization is used (chosen to be at $t + \frac{1}{2}\Delta t$)

$$a_k = G_d g(t + \frac{1}{2}\Delta t), \qquad G_d = \int_0^{\Delta t} e^{A\tau} d\tau, \tag{19}$$

where Δt is the sampling time. Note that this is only used to generate the data point, the resulting GP model is still a continuous function. The value $g(t + 1/2\Delta t)$ is obtained as solution of the linear system

$$G_d g(t + \frac{1}{2}\Delta t) \approx (x_{k+1} - A_d x_k - B_d u_k), \tag{20}$$

which is the data point used to train the GP.

In the experiments presented in the following, the discretization of the mean prediction $\tilde{g}(t)$ is obtained from the exact discretization

$$\tilde{a}_k = \int_0^{\Delta t} e^{A\tau} \tilde{g}((k+1)\Delta t - \tau) d\tau, \tag{21}$$

evaluated with a standard ODE-solver [37].

B. Toy problem

As a simple problem for providing intuition, consider the following linear dynamic (double integrator) system with an additive time-periodic component g:

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) + g(t) \tag{22a}$$

$$g(t) = \begin{bmatrix} \sin(t) \\ \cos(1.3t) \end{bmatrix}. \tag{22b}$$

The goal is to control the first state of the system to the origin. We use the quadratic cost

$$l(x_n, u_n) = \frac{1}{2}x_n^\intercal Q x_n + \frac{1}{2}u_n^\intercal R u_n \tag{23}$$

with diagonal state weight $Q = \operatorname{diag}(100, 0)$. The weight on the control input is set to $R = 1$, allowing for aggressive control behavior. The horizon length of the MPC is set to $N = 15$. State and input constraints are omitted for simplicity.

The system was simulated numerically with a sampling rate of 1 Hz. Fig. 5 shows control inputs and resulting state trajectories for a model predictive controller without information about the disturbance g, and a model predictive controller using the posterior mean functions of two periodic Gaussian process regressors as a model for g. One GP is trained for each dimension of the disturbance.

After an identification phase in the first 5 s of the experiment, the GP based controller shows a drastic performance improvement. Omitting this identification phase, the root-mean-square (RMS) error, measured with respect to the origin, drops by 90 %, from 0.94 for the linear model to 0.097 for the GP based

controller. Speaking more qualitatively, Fig. 5 also shows less residual structure in the controlled state x_1. It is visible that control signals are applied earlier when the prediction is used.

While the GP based controller is effective at reducing the periodic structure from the first controlled state, the regression model itself remains able to predict the periodic error correctly into the future, even when trained exclusively on controlled states. This is possible because the regression model is obtained from the controlled dynamics, so it can account for the shift of periodicity from the states to the control input. This feature of the framework is crucial for identifying controlled systems (see next section).

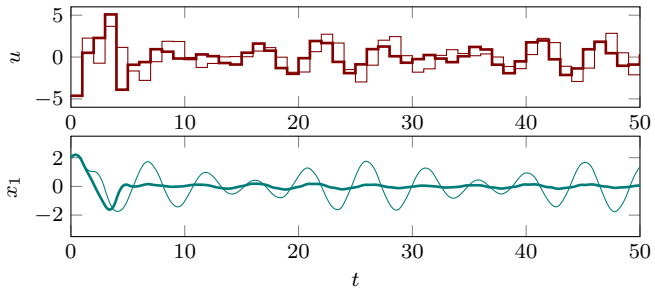


Fig. 5. Closed-loop input and output trajectories under the MPC controller. MPC using a linear model is shown as thin lines, the GP based MPC controller as thick lines.

C. Periodic error correction for telescope mounts

The original motivation for the work presented here is the control of periodic errors in astrophotography systems. Telescope mounts correct for the Earth's rotation relative to the sky by a circular motion at the sidereal velocity. This motion is typically produced by mechanical devices using cogs and worm gears, which gives rise to periodic deviations. Because contemporary telescopes, even those used by amateurs, have high optical resolution, and because images are taken with long exposure times, these mechanical imprecisions are frequently the dominant source of blur on astronomical photographs.

Existing *periodic error correction* systems require careful system identification by the user of the telescope. The corresponding measurements need to be re-aligned after every repositioning of the telescope (that is, multiple times per night), and still regularly lead to unsatisfactory performance.

A problem specific to this astronomical application is that state measurement is performed by taking photographs of the night sky, which requires relatively long exposure times, so that the measurement frequency can reach the order of magnitude of the periodic error. This is precisely the domain in which we expect to see utility from a periodic model.

1) Simulated system: The period length of the periodic error in telescopes is relatively slow. To allow rapid prototyping, we designed a simulated system with dynamics similar to a real telescope. Experiments have shown that the model can be reduced by considering only the angular position as state, measured relative to the desired state. The state can be influenced by an input velocity. The resulting model is

$$\dot{x} = u + g(t), \tag{24}$$

with an unknown function $g(t)$ that is observed to be quasi periodic.

Fig. 6 shows simulation results that empirically confirm the intuition from §II-A that the benefit of periodic prediction in control depends on the sampling rate. Using the numerical simulation, we compare for various sampling rates of the state

- an MPC controller using the linear model (1) with $g(t) = 0 \quad \forall t$
- two MPC controllers, both using a nonparametric, but fully stationary (i.e. not periodic) GP model for g with the square-exponential covariance function (5). One of these models uses a length scale smaller than the periodicity (i.e. it can extrapolate periodic swings locally, but not beyond one period); the other a length scale longer than the periodicity (i.e. it averages over the periodic variations)
- two MPC controllers using instances of the periodic model for g described in this paper; one in which the hyperparameters are fixed to a good value a priori (amounting to the assumption that the period of g is known), the other using the full setup described in §III, in which the periodicity hyperparameter is learned by type-II maximum likelihood during identification

Since we are only interested in the performance in the limit in this experiment, all the controllers were run for an identification phase of 10 period lengths to avoid artifacts from the early identification phase. Fig. 6 then shows RMS error, i.e. deviation of the state from the origin, after this phase as a function of the sampling time. The RMS error is measured over 10 period lengths starting after the identification phase. The discretization time for the MPC is always set to 1/100 of the period length, i.e. to 1 s. Between measurements, the MPC controllers are operated in open-loop mode, i.e. the control inputs are obtained from the sequence of the last MPC optimization.

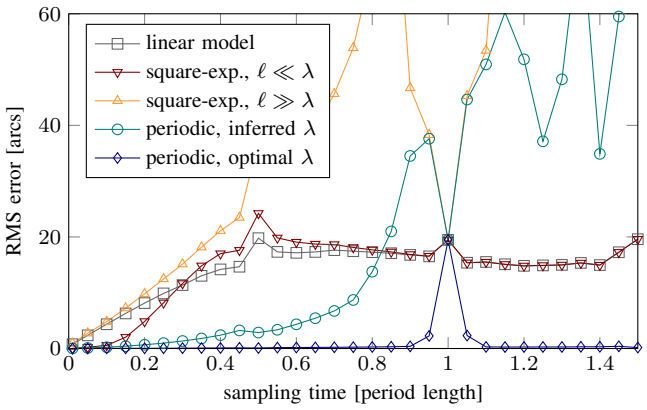


Fig. 6. Comparison of the error at different sampling times (in simulation). MPC control inputs are computed at the indicated sampling times, shown as a fraction of the period length. The control rate is 1 Hz for every plot. The MPC parameters were set to $Q = 10^2$ and $R = 10^1$. The horizon length was chosen such that the horizon covers the time until the next measurement. Between measurements, the MPCs are operated in open-loop mode. $g(t)$ is set to be a sine with fixed period $\lambda = 100\,\mathrm{s}$.

The results demonstrate the intuition: For sampling times much smaller than λ, the dynamics are locally linear, and all models achieve an error close to zero. Their performance difference is only marginal (lower left in Fig. 6). For sampling

times between 10 % and 80 % of λ, the periodic model offers considerable benefits. When the sampling times are close to or larger than the periodicity, the Nyquist rate imposes limits on identifiability of the system, which adversely affects the performance of the periodic nonparametric model. This shows that a broad prior can lead to bad performance if only little data is available (c.f. §III-B). On the other hand, if λ is known precisely, very good control is possible even for sampling rates lower than λ. The green line in Fig. 6 represents the performance of a system almost ignorant of λ in the beginning: the prior is very broad. One can expect prior information about λ of varying vagueness to give performance somewhere in the region between the green and blue curves in Fig. 6.

Of course, the case where sampling rate and λ are equal is special, since then g appears constant in the measurements, and even the informed periodic model can only ever learn the behavior of g at one unique point during the period. A weaker version of this effect is also visible in the plot at a sampling rate of $\lambda/2$. This "selection bias" affects all regression models, including the aperiodic ones.

Summarizing, the proposed approach offers the main benefit for sampling intervals between 10 % and 80 % of the disturbance period length. Whereas for short sampling intervals, all techniques result in low errors, systems with sampling frequency below the Nyquist frequency of the periodic disturbance cannot be expected to improve with a system that has to learn the periodicity.

2) Evaluation of fault tolerance: A similar experiment is conducted to investigate the effect of missing measurements on the performance of the controlled system. Fig. 7 shows the empirical results. The setup is the same as in the experiment described before, but now the sampling time is fixed to a value of 5 s, while the period length is 100 s. The length of the horizon is set to 41 time steps (i.e. 205 s), covering more than two full periods of the periodic effect, which is a realistic setting for a real telescope.

After giving the system enough time to learn under regular output measurements, no new data is given to the learning procedure to update the GP model of $g(t)$ and no new state estimate is available to the controller. This corresponds to a fault in the sensor (or clouds in front of the camera in the telescope setting). The sensor does not recover within the simulation time. Fig. 7 shows the performance of the different controllers, measured in terms of the RMS error and calculated from the beginning of the fault, here at time 0. In all controller setups, the MPC controller cannot be recomputed as there is no state estimate available. For the next time steps (without measurement), the MPC therefore is operated in open-loop mode, i.e. the control inputs are taken from the control sequence computed at the time before the fault.

At the beginning of the failure (lower left of Fig. 7), the performance of all methods is good, since the effect of the periodic function is still small. Over time, we see the simple controller without prediction slowly degrading. Interestingly, the performance of the GP with square exponential kernel with too long a length scale performs even worse than not predicting $g(t)$ at all. This illustrates how critical the choice of the hyperparameters is and that a wrong choice can even degrade the performance. The model with sensible length scale performs significantly better initially, but the extrapolation of the SE kernel degrades quickly (see Fig. 2a) resulting in an overall performance that is only slightly better than for the linear model.

With periodic predictions, in contrast, the controllers perform significantly better during the fault. The periodic GP is clearly better than the SE with short length scale, even if the period length is inferred and not fixed at a good value. The RMS error for the GP with optimal parameter λ is virtually zero and even a controller knowing the true function g would therefore only show marginal improvement. This analysis shows that the proposed combination of a periodic GP model and MPC control is able to compensate temporary sensor failures and maintain high control performance for locally periodic dynamic effects.

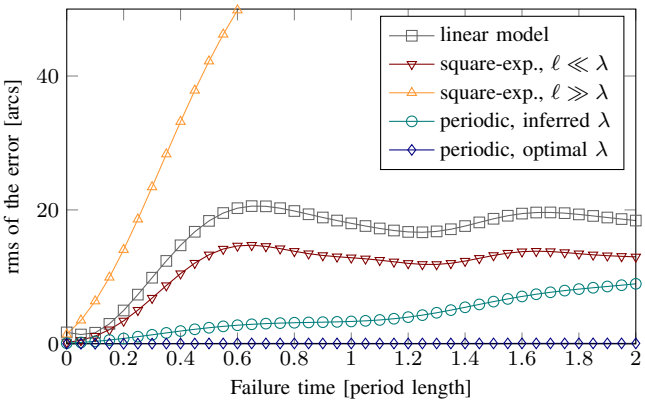


Fig. 7. Comparison of the RMS error after a failure (in simulation). Sampling time and discretization time of the MPC are both 5 s. The MPC parameters are set to $N = 41$, $Q = 10^2$ and $R = 10^1$. After a training period, no new measurements are taken ("dark phase"). The MPC is operated in open-loop mode. $g(t)$ is set to be a sine with fixed period $\lambda = 100$ s.

3) Hardware experiment: We have tested our implementation on a physical system, a commercially available Vixen Sphinx[1] telescope mount (Fig. 8). Without closed-loop control, this mount shows about 7″ of RMS error after correction for static drift. The error arises from the imperfect shape of the cogs in the gear of this mount (Fig. 9). The imperfect shape is not visible to the naked eye, see Fig. 10a for an uncontrolled measurement.

Because outdoor measurements are subject to random, time-varying effects like weather conditions, we constructed a more reproducible experimental setup using a second, high precision gearless ASA DDM60Pro[2] telescope mount equipped with a laser "star" as tracking reference. It has a typical tracking RMS error of about 0.4″. The measurement is done with a Canon EF400DO lens[3] on a "The Imaging Source" DMK 41AU02.AS[4] camera.

[1] http://www.vixenoptics.com/mounts/sphinx.htm
[2] http://www.astrosysteme.at/eng/mount_ddm60.html
[3] http://www.canon-europe.com/For_Home/Product_Finder/Cameras/EF_Lenses/Telephoto/EF_400mm_f4_DO_IS_USM/
[4] http://www.astronomycameras.com/products/usb/dmk41au02as/

Fig. 8. The telescope mount used for the tracking experiments. On the right side is the camera lens used as guiding telescope. A main telescope is not used for the tests.

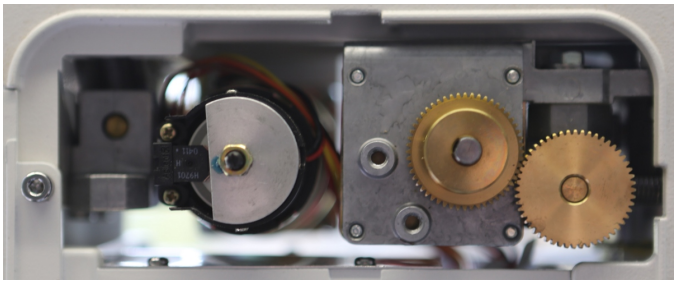


Fig. 9. The gearbox of the telescope mount. One of the motors is visible on the left. The two cogs on the right transmit the motor's rotation to a worm gear (not visible), which sits on the same axis and thus has the same period. These cogs and worm gear are the likely source of the periodic error.

For the hardware interaction, the open source "PhD Guiding"[5] software package is used. In the original implementation this software uses a deadbeat controller. The telescope is connected to the computer with a Shoestring GPUSB[6], a device that sends pulse-width modulated signals to telescopes over a commonly used 6-wire interface.

We altered the software to gain access to the measured displacement of the camera image. The value is sent through a network socket to MATLAB, where the controller developed in this paper calculates the optimal control signal, which is in turn sent back to the guiding software. The software then sends the control signal to the telescope hardware. For plotting and calculation of the RMS error, the measured displacement is converted from pixels into arcseconds ($''$) with an empirically determined conversion factor.

For real-time implementation, algorithmic complexity is relevant. The computational cost of the GP prediction scales cubically in the number of data points. To bound computational cost, we limit the number of used data points to 90 in a moving window fashion. This gives a sufficient coverage of 270 s, or about 3 periods of the short periodic component. Since inference continuously runs in an extrapolation setting (see also Fig. 2b), this is sufficient for precise inference and control.

For the prediction of the dynamics in the model predictive control, an ODE-solver is employed to predict the reference (16) from the mean of the GP prediction. This has manageable computational cost because the inference cost of a Gaussian process is dominated by the initial one-time operation of inverting the Gram matrix K in $\mathcal{O}(D^3)$ time, while subsequent evaluations of the mean function at M times only has cost $\mathcal{O}(MD)$ (see (3)).

The optimization of the hyperparameters is also an expensive part of this algorithm. As the kernel Gram matrix has to be built and inverted at every evaluation of the objective function, the number of evaluations has to be kept small at every sampling time. We use a numerical optimizer based on the BFGS update [38], [39], [40], [41], a quasi-Newton algorithm that updates an estimate of the inverse Hessian in each iteration. In the standard implementation, the estimate of the Hessian obtained at one time step is discarded after each individual call to the optimizer. For the use in the control setting, we altered the algorithm so that the estimate of the inverse Hessian is stored and used to initialize the optimizer's estimate at the next sampling time. This makes it possible to do one iteration per sampling interval (in a sense, threading the optimization algorithm into the learning algorithm). This significantly reduces computation time.

The presented method was tested on two different setups, one with an MPC based on the linear model without g and one with the GP prediction for the periodic error. The test was run 3 times for 25 min each. Both the sampling and the discretization time were set to 3 s. The horizon length was $N = 10$; the state and control weights were set to $Q = 10^2$ and $R = 10^3$. The results of these runs are shown in Table I. The RMS error drops by 22.64 % through the use of GP predictions in this hardware setup. Baseline measurements without movement showed about 0.25 to 0.35$''$ of noise. The noise introduced by the stepper motor could not be quantified with the current measurement system. Overall, the presented method eliminated at least a third of the controllable error, after subtracting baseline noise but without taking the stepper motor into account. That is a good result in this domain, but could probably be improved, which is also visible through the light residual structure visible in Fig. 10c.

	Run 1	Run 2	Run 3	**Mean**
Plain MPC	0.9839	1.0234	0.9353	**0.9809**
GP-MPC	0.7365	0.7792	0.7605	**0.7587**

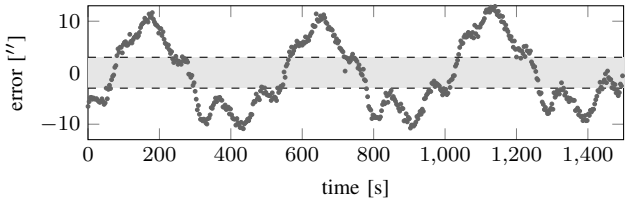
TABLE I
EXPERIMENTAL RESULTS (RMS ERROR, IN $''$)

In order to further investigate this issue and assess the performance of the method, Fig. 11 shows the power spectra of the measurements, obtained from the Fourier transform. It is noticeable that the strong periodic components near 100 s and near 500 s are highly damped with the presented method.

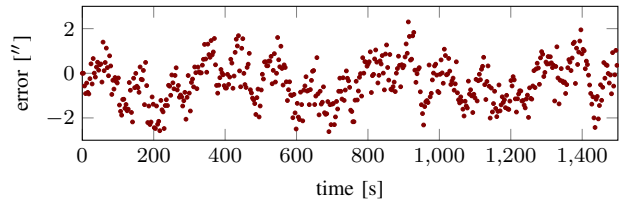
VI. CONCLUSION

High precision control of dynamical systems requires precise models of even minor external error sources. Where analytic

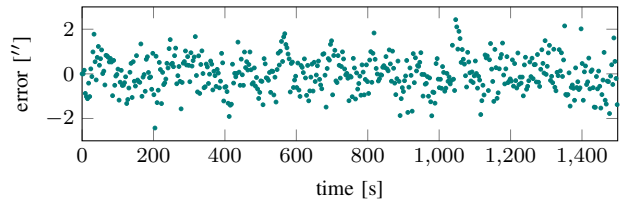
[5]http://www.stark-labs.com/phdguiding.html
[6]http://www.store.shoestringastronomy.com/gpusb.htm



(a) Uncontrolled, RMS(e) = 6.533″

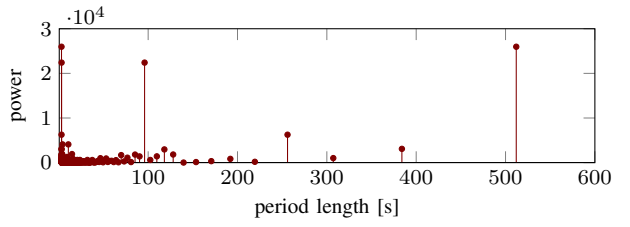


(b) Plain MPC, RMS(e) = 1.023″

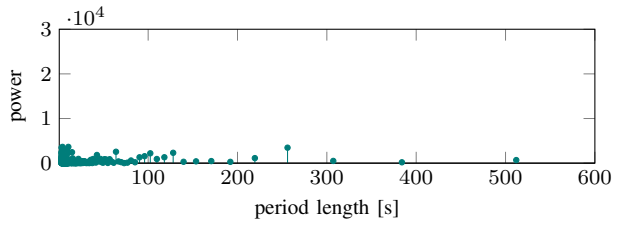


(c) GP-MPC, RMS(e) = 0.7792″

Fig. 10. Measurements of the state for the physical experiments with the telescope: Without controller (a), where the shaded area shows the vertical range of the two other plots; for plain MPC using only a linear model (b); and for the periodic Gaussian process based MPC (c). Results (b) and (c) are from run 2 of our experiments, which resulted in the highest (worst) RMS error for both models (Table I).



(a) Plain MPC



(b) GP-MPC

Fig. 11. Power spectra of the measurements of Fig. 10, plain MPC using only a linear model (a), and periodic Gaussian process model based MPC (b).

models are not available, they can only be constructed numerically from measurements of the system. Periodic error sources are an especially promising domain in this regard, as they can be extrapolated well into the future. We have studied a nonparametric modeling framework based on a carefully crafted Gaussian process prior exhibiting a weak, localized form of periodicity. Because Gaussian regression returns models in the form of stochastic differential equations, they can be combined directly with existing control frameworks. Integration into a model predictive control scheme was investigated, which can leverage the prediction at a desired resolution, even below the sampling time. Numerical and physical experiments confirm the intuitive result that the benefit of periodic models depends on the relative size of state sampling and disturbance frequencies. We showed that, even in cases where the gain of a periodic prediction is only marginal during normal operation, such models are beneficial when sensors fail temporarily. The presented method also shows considerable increases in control performance, confirming the practical utility of this framework.

References

[1] S. Särkkä, *Bayesian filtering and smoothing*. Cambridge University Press, 2013.

[2] T. Erm and S. Sandrock, "Adaptive periodic error correction for the VLT," in *Proceedings of SPIE*, vol. 4837, 2003.

[3] J. L. Crassidis and F. L. Markley, "Predictive filtering for nonlinear systems," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 3, pp. 566–572, 1997.

[4] S. G. Yuen, P. M. Novotny, and R. D. Howe, "Quasiperiodic predictive filtering for robot-assisted beating heart surgery." in *IEEE International Conference on Robotics and Automation*, 2008, pp. 3875–3880.

[5] J. Kocijan, R. Murray-Smith, C. E. Rasmussen, and A. Girard, "Gaussian process model based predictive control," in *Proceedings of the American Control Conference*, 2004.

[6] A. Aswani, H. González, S. S. Sastry, and C. Tomlin, "Provably safe and robust learning-based model predictive control," *Automatica*, vol. 49, no. 5, pp. 1216–1226, 2013.

[7] M. Tanaskovic, L. Fagiano, R. Smith, P. J. Goulart, and M. Morari, "Adaptive model predictive control for constrained linear systems," in *European Control Conference*, 2013, pp. 382–387.

[8] G. Pillonetto, F. Dinuzzo, T. Chen, G. D. Nicolao, and L. Ljung, "Kernel methods in system identification, machine learning and function estimation: A survey," *Automatica*, vol. 50, no. 3, pp. 657–682, 2014.

[9] P. Hennig, "Optimal Reinforcement Learning for Gaussian Systems," in *Advances in Neural Information Processing Systems (NIPS)*, 2011, pp. 325–333.

[10] J. Ko and D. Fox, "GP-BayesFilters: Bayesian filtering using Gaussian process prediction and observation models," *Autonomous Robots*, vol. 27, no. 1, pp. 75–90, 2009.

[11] J. Hall, C. E. Rasmussen, and J. M. Maciejowski, "Modelling and control of nonlinear systems using Gaussian processes with partial model information," in *IEEE Conference on Decision and Control*, 2012, pp. 5266–5271.

[12] J. M. Maciejowski and X. Yang, "Fault tolerant control using gaussian processes and model predictive control," in *Conference on Control and Fault-Tolerant Systems (SysTol)*, 2013, pp. 1–12.

[13] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[14] I. Steinwart, "On the influence of the kernel on the consistency of support vector machines," *Journal of Machine Learning Research*, vol. 2, pp. 67–93, 2002.

[15] C. A. Micchelli, Y. Xu, and H. Zhang, "Universal kernels," *Journal of Machine Learning Research*, vol. 7, pp. 2651–2667, 2006.

[16] A. W. van der Vaart and J. H. van Zanten, "Rates of contraction of posterior distributions based on Gaussian process priors," *The Annals of Statistics*, pp. 1435–1463, 2008.

[17] A. W. van der Vaart and J. van Zanten, "Information Rates of Nonparametric Gaussian Process Methods," *Journal of Machine Learning Research*, vol. 12, pp. 2095–2119, 2011.

[18] E. D. Klenske, M. N. Zeilinger, B. Schölkopf, and P. Hennig, "Nonparametric dynamics estimation for time periodic systems," in *Annual Allerton Conference on Communication, Control, and Computing*, Oct 2013, pp. 486–493.

[19] M. Diehl, H. J. Ferreau, and N. Haverbeke, "Efficient numerical methods for nonlinear MPC and moving horizon estimation," in *Nonlinear Model Predictive Control*. Springer, 2009, vol. 384, pp. 391–417.

[20] D. J. C. MacKay, "Introduction to Gaussian processes," *NATO ASI Series F Computer and Systems Sciences*, vol. 168, pp. 133–166, 1998.

[21] D. Duvenaud, J. R. Lloyd, R. Grosse, J. B. Tenenbaum, and Z. Ghahramani, "Structure discovery in nonparametric regression through compositional kernel search," in *International Conference on Machine Learning (ICML)*, 2013.

[22] D. Barber, *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2011.

[23] M. B. Thompson and R. M. Neal, "Slice sampling with adaptive multivariate steps: The shrinking-rank method," *preprint at arXiv:1011.4722*, 2010.

[24] L. Wasserman, *All of Statistics: A Concise Course in Statistical Inference*. Springer, 2010.

[25] D. MacKay, "Comparison of approximate methods for handling hyperparameters," *Neural Computation*, vol. 11, no. 5, pp. 1035–1068, 1999.

[26] R. R. Bitmead, M. Gevers, and V. Wertz, *Adaptive optimal control: The thinking man's GPC*. Prentice Hall New York, 1990.

[27] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2009.

[28] B. Kouvaritakis and M. Cannon, "Stochastic model predictive control," in *Encyclopedia of Systems and Control*, J. Baillieul and T. Samad, Eds. Springer London, 2014, pp. 1–9.

[29] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[30] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, 2014.

[31] A. Domahidi, "FORCES: Fast optimization for real-time control on embedded systems," http://forces.ethz.ch, 2012.

[32] D. Limon, T. Alamo, D. Muñoz de la Peña, M. N. Zeilinger, C. N. Jones, and M. Pereira, "MPC for tracking periodic reference signals," in *IFAC Nonlinear Model Predictive Control Conference*, 2012, pp. 490–495.

[33] A. T. Schwarm and M. Nikolaou, "Chance-constrained model predictive control," *AIChE Journal*, vol. 45, no. 8, pp. 1743–1752, 1999.

[34] M. Ono and B. C. Williams, "Iterative risk allocation: A new approach to robust model predictive control with a joint chance constraint," in *IEEE Conference on Decision and Control*, 2008, pp. 3427–3432.

[35] M. C. Campi, S. Garatti, and M. Prandini, "The scenario approach for systems and control design," *Annual Reviews in Control*, vol. 33, no. 2, pp. 149–157, 2009.

[36] G. Schildbach, L. Fagiano, C. Frei, and M. Morari, "The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations," *Automatica*, vol. 50, no. 12, pp. 3009–3018, 2014.

[37] J. R. Dormand and P. J. Prince, "A family of embedded Runge-Kutta formulae," *Journal of Computational and Applied Mathematics*, vol. 6, no. 1, pp. 19–26, 1980.

[38] C. G. Broyden, "A new double-rank minimization algorithm," *Notices of the American Mathematical Society*, vol. 16, p. 670, 1969.

[39] R. Fletcher, "A new approach to variable metric algorithms," *The Computer Journal*, vol. 13, no. 3, p. 317, 1970.

[40] D. Goldfarb, "A family of variable metric updates derived by variational means," *Mathematics of Computation*, vol. 24, no. 109, pp. 23–26, 1970.

[41] D. F. Shanno, "Conditioning of quasi-Newton methods for function minimization," *Mathematics of Computation*, vol. 24, no. 111, pp. 647–656, 1970.