**2024 IEEE 63rd Conference on Decision and Control (CDC)**
**December 16-19, 2024. MiCo, Milan, Italy**

# Efficient Online Inference and Learning in Partially Known Nonlinear State-Space Models by Learning Expressive Degrees of Freedom Offline

Jan-Hendrik Ewering, Björn Volkmann, Simon F. G. Ehlers, Thomas Seel, and Michael Meindl

*Abstract*— **Intelligent real-world systems critically depend on expressive information about their system state and changing operation conditions, e. g., due to variation in temperature, location, wear, or aging. To provide this information, online inference and learning attempts to perform state estimation and (partial) system identification simultaneously. Current works combine tailored estimation schemes with flexible learning-based models but suffer from convergence problems and computational complexity due to many degrees of freedom in the inference problem (i. e., parameters to determine). To resolve these issues, we propose a procedure for data-driven offline conditioning of a highly flexible Gaussian Process (GP) formulation such that *online* learning is restricted to a subspace, spanned by expressive basis functions. Due to the simplicity of the transformed problem, a standard particle filter can be employed for Bayesian inference. In contrast to most existing works, the proposed method enables online learning of target functions that are nested nonlinearly inside a first-principles model. Moreover, we provide a theoretical quantification of the error, introduced by restricting learning to a subspace. A Monte-Carlo simulation study with a nonlinear battery model shows that the proposed approach enables rapid convergence with significantly fewer particles compared to a baseline and a state-of-the-art method.**

## I. INTRODUCTION

Operation under complex and changing conditions is a key challenge in modern control research. The changing conditions can be attributed to intrinsic system behavior (e. g., wear and friction in machines [1], aging of batteries [2]) or environment interaction (e. g., unknown environment map [3], [4], changing tire-road friction [5], [6], [7]), see Fig. 1. In both cases, information on the underlying change is crucial to ensure adaptive and reliable operation. In this light, estimation algorithms fuse assumptions about the system structure with (limited) sensor data to obtain estimates of (latent) system states and varying parameters.

Usually, an approximate system model can be derived from first principles, such as rigid body dynamics. However, knowledge about other aspects influencing the system is often limited, e. g., friction effects, or environment maps.

To address this, offline algorithms for joint inference of latent system states and learning of (partially) unknown models have been proposed [8], [9], [10]. In [8], [9], for instance, learning is facilitated through encoding basic assumptions about smoothness and dimensionality of the underlying true system behavior using a Gaussian Process (GP) prior [11]. For inference and learning, the authors employ particle Markov chain Monte Carlo methods [12]. However, such algorithms are often computationally demanding and rely on offline data to train highly flexible learning-based models, which hinders real-time adaptivity to changing conditions.

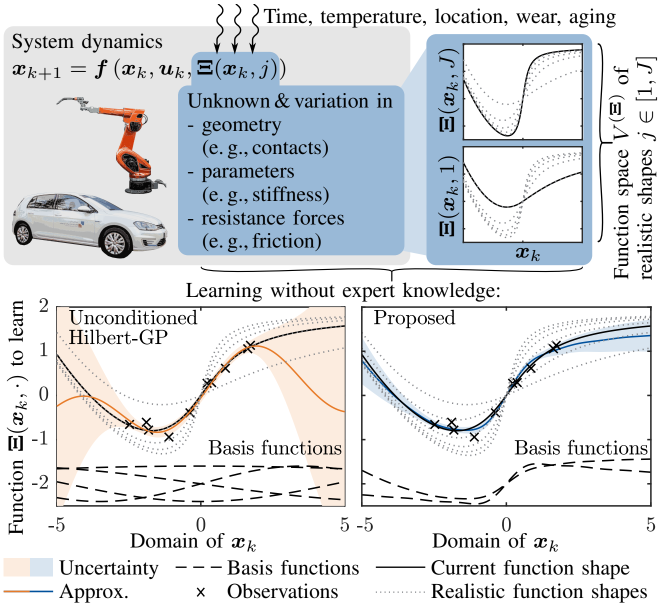
Time, temperature, location, wear, aging
System dynamics
$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\Xi}(\boldsymbol{x}_k, j))$
Unknown & variation in
- geometry
(e. g., contacts)
- parameters
(e. g., stiffness)
- resistance forces
(e. g., friction)
$\boldsymbol{\Xi}(\boldsymbol{x}_k, J)$
$\boldsymbol{\Xi}(\boldsymbol{x}_k, 1)$
$\boldsymbol{x}_k$
Function space $V(\boldsymbol{\Xi})$ of realistic shapes $j \in [1, J]$
Learning without expert knowledge:
Function $\boldsymbol{\Xi}(\boldsymbol{x}_k, \cdot)$ to learn
2
1
0
-1
-2
Unconditioned Hilbert-GP
Proposed
Basis functions
Basis functions
-5
0
5
-5
0
5
Domain of $\boldsymbol{x}_k$
Domain of $\boldsymbol{x}_k$
Uncertainty
Approx.
Basis functions
Observations
Current function shape
Realistic function shapes

Fig. 1. Unknown and varying effects $\boldsymbol{\Xi}$ in real-world systems that complicate operation (top) and approaches for learning of the underlying relationships (bottom). Without expert knowledge, online learning is difficult with state-of-the-art methods, e. g., Hilbert-GP [11], due to many inexpressive degrees of freedom (here: basis functions). In contrast, the proposed method enables efficient online learning by *data-driven* construction of few expressive basis functions.

In contrast, recent estimation algorithms attempt to learn (partially) unknown system behavior online while simultaneously inferring latent states. In [5], Bayesian online learning of GP state-space models is proposed based on [8], [9] and using a carefully designed Sequential Monte-Carlo (SMC) algorithm. However, high-dimensional search spaces can still render the estimation problem infeasible due to (i) convergence issues associated with complex posterior probability densities, and (ii) computational complexity. The associated challenges are commonly termed "curse of dimensionality", especially in the context of SMC methods.

To resolve these issues, a key approach is restricting learning to expressive Degrees of Freedom (DOF). In particular, what we mean with "expressive DOF" is that the model structure should exhibit few adjustable parameters, each of which has a unique and significant impact on the considered target, e. g., the function shape to be learned.

In estimation settings with purely physics-based models, this

The Authors are with the Institute of Mechatronic Systems, Leibniz Universität Hannover, 30167 Hanover, Germany (e-mail: jan-hendrik.ewering@imes.uni-hannover.de).

**979-8-3503-1632-2/24/$31.00 ©2024 IEEE**

would be the derivation of empirical expert models in which only few parameters need to be determined. A common example is to represent the tire-road friction using the magic formula tire model [13] and to employ it for automotive estimation, e. g., using unscented Kalman filters [7].

On the other hand, learning-based models can be constrained to facilitate adaptivity in online settings. In [6], expert knowledge is exploited in the form of symmetry and linear operator constraints to restrict flexibility. While the approaches in [6] are important contributions to simplify the learning problem, they are still limited to specific cases, and expert knowledge may not be available. The authors of [14] solve a high-dimensional inference task by restricting online learning to a region around the current operation point. However, the function shape itself is not constrained, resulting in erroneous target function estimates. A notable contribution for learning of a low-dimensional, yet flexible GP model is presented in [15]. The authors connect a tractable tensor network with a linear-in-the-parameters GP formulation [11] to enable inference in a low-dimensional subspace, but the advantages are not exploited for online inference and learning.

In the vast majority of previous publications, the target function is learned either with overly flexible models (i. e., too many DOF) or employing system-specific expert knowledge. Unlike all prior work, we propose to learn the most significant features of observed function realizations from data offline and to use them as expressive DOF for efficient Bayesian online inference and learning. By doing this, we restrict online learning to a low-dimensional subspace that spans only realistic function shapes, yielding fast convergence and reduced computational burden, while employing a standard Particle Filter (PF). Due to the simplicity of the inference problem and in contrast to most previous work, the proposed methods are capable of learning functions online that are nested inside nonlinear first-principles models, without expert knowledge about the target functions.

The paper is structured as follows. First, the problem is formalized in Sec. II. The proposed method for efficient online inference and learning is presented in Sec. III. Last, we illustrate our findings with a simulation example and draw conclusions in Sec. IV and V, respectively.

*Notation*: For a vector $\boldsymbol{e} \in \mathbb{R}^{n_e}$, $\boldsymbol{e} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes a draw from a multivariate Normal with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ and $e_i$ is its $i$-th element. We use column vectors if not stated explicitly otherwise. A matrix $\boldsymbol{A}$ is written in bold and capital and has elements $a_{ij}$ for row $i$ and column $j$. The identity matrix of dimension $n$ is $\mathbf{I}_n$. We write the conditional density of a state sequence $\boldsymbol{x}_{1:k} := \{\boldsymbol{x}_i\}_{i=1}^{k}$ from time steps 1 to $k$, given the measurements $\boldsymbol{y}_{1:k}$ as $p(\boldsymbol{x}_{1:k}|\boldsymbol{y}_{1:k})$. By writing $V^{(\phi,N)} := \operatorname{span}(\phi_1(\boldsymbol{x}), \ldots, \phi_N(\boldsymbol{x}))$, we refer to the space of functions that can be represented by linear combinations of basis functions $\{\phi_i(\boldsymbol{x})\}_{i=1}^{N}$, defined on the domain $\boldsymbol{x} \in \Omega$. The inverse Wishart distribution with scale matrix $\boldsymbol{\Lambda}$ and DOF $\nu$ is $\mathcal{IW}(\nu, \boldsymbol{\Lambda})$. The multivariate Student-t distribution is $\mathcal{T}(\nu, \boldsymbol{\mu}, \boldsymbol{\Lambda})$. The Dirac delta mass $\delta_{ij} = 1$ for $i = j$ and 0 otherwise. By writing $\|\boldsymbol{x}\|_{\boldsymbol{M}}^2$, we mean $\boldsymbol{x}^\top \boldsymbol{M} \boldsymbol{x}$ and $\|\boldsymbol{x}\|$ denotes the L2-norm of $\boldsymbol{x}$.

## II. PROBLEM FORMULATION

The objective is to learn the nested and changing system behavior *online* from noisy input-output data while using first-principles model knowledge. In a probabilistic discrete-time state-space model

$$\boldsymbol{x}_{k+1} = \overbrace{\boldsymbol{f}\left(\boldsymbol{x}_k, \boldsymbol{u}_k, \Xi(\boldsymbol{x}_k, j)\right)}^{\substack{\text{Dominating first-principles model} \\ \text{\& nested unknown effects to be learned}}} + \boldsymbol{e}_k^x, \tag{1a}$$

$$\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{e}_k^y, \tag{1b}$$

this amounts to learning the function $\Xi : \mathbb{R}^{n_x+1} \to \mathbb{R}^{n_\xi}$ and inferring the latent states $\boldsymbol{x}_k \in \mathbb{R}^{n_x}$ at time step $k$. The variation of $\Xi(\cdot, j)$ in a set of typical shapes, i. e., the function space $V^{(\Xi)}$, is described by a scheduling variable $j \in [1, J]$ (see Fig. 1). Please note that the scheduling variable $j$ is introduced for notational convenience and is not assumed to be known for online inference and learning. The case of $\Xi$ being nested in $\boldsymbol{h}$ is conceptually similar and will not be considered explicitly. In (1), the inputs are $\boldsymbol{u}_k \in \mathbb{R}^{n_u}$, and the measurements $\boldsymbol{y}_k \in \mathbb{R}^{n_y}$. The process noise $\boldsymbol{e}_k^x$ and the measurement noise $\boldsymbol{e}_k^y$ are zero-mean Gaussian random variables with known covariance matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$, i. e., $\boldsymbol{e}_k^x \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q})$ and $\boldsymbol{e}_k^y \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R})$. The state dynamics $\boldsymbol{f} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_\xi} \to \mathbb{R}^{n_x}$ and the measurement function $\boldsymbol{h} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_y}$ are known from first-principles.

For learning of $\Xi$, a generic approximation model (e. g., a GP) is required to impose "artificial structure". To enable efficient inference and learning, the approximation should (i) simplify the estimation problem, and (ii) be computationally efficient. We assume that we do not have system-specific expert knowledge about the function structure of $\Xi$ and its variation in $V^{(\Xi)}$. However, we are given an *offline* data set $\mathcal{D} = \{\boldsymbol{\xi}_{1:K}^j, \boldsymbol{x}_{1:K}^j\}_{j=1}^J$ with noisy observations $\boldsymbol{\xi}_k = \Xi(\boldsymbol{x}_k, j) + \boldsymbol{e}_k^\xi$, $\boldsymbol{e}_k^\xi \sim \mathcal{N}(\boldsymbol{0}, \sigma_\xi \mathbf{I}_{n_\xi})$, that stem from $J$ realizations in the whole range $j \in [1, J]$. Note that measuring quantities of interest *offline* under laboratory conditions and estimating these quantities *online* in operation is common practice in many applications. Further, established methods could be used to infer $\mathcal{D}$, leveraging a system structure that is affine in $\Xi$ [9] or using identification methods for general nonlinear systems [16], [17].

In the online setting, both state estimation and learning of $\Xi$ need to be performed simultaneously in each time step $k$ based on the current inputs $\boldsymbol{u}_{k-1}$ and measurements $\boldsymbol{y}_k$ only. Apart from convergence, the computational complexity of the algorithm should be as small as possible.

## III. PROPOSED METHOD

To enable efficient online inference and learning, the parameters to be learned online should be restricted to expressive DOF while retaining the adaptation flexibility to learn "realistic" (i. e., actually occurring) shapes of $\Xi(\cdot, j)$. The key idea of the proposed method is to capture different realizations $\Xi(\cdot, j)$ of the changing system behavior *offline* with a highly flexible GP approximation [11] and to transform it to a low-dimensional representation. The GP approximation lives in $V^{(\phi,N)}$, spanned by a high-dimensional set

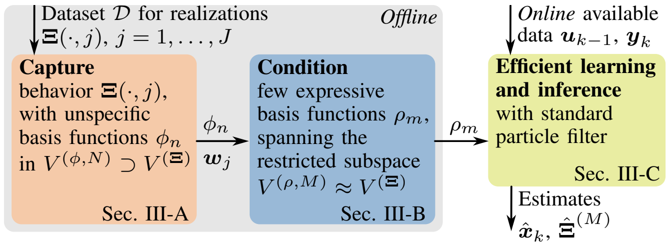
Dataset $\mathcal{D}$ for realizations $\Xi(\cdot, j)$, $j = 1, \dots, J$
Offline
Online available data $\boldsymbol{u}_{k-1}$, $\boldsymbol{y}_k$
**Capture** behavior $\Xi(\cdot, j)$, with unspecific basis functions $\phi_n$ in $V^{(\phi,N)} \supset V^{(\Xi)}$
Sec. III-A
$\phi_n$
$\boldsymbol{w}_j$
**Condition** few expressive basis functions $\rho_m$, spanning the restricted subspace $V^{(\rho,M)} \approx V^{(\Xi)}$
Sec. III-B
$\rho_m$
**Efficient learning and inference** with standard particle filter
Sec. III-C
Estimates $\hat{\boldsymbol{x}}_k$, $\hat{\Xi}^{(M)}$

Fig. 2. Proposed method to enable efficient online inference and learning without expert knowledge. First, "realistic" (i. e., actually occurring) shapes of the target function $\Xi(\cdot, j)$, $j = 1, \dots, J$, are captured using the flexible Hilbert-GP formulation introduced in [11] with unspecific basis functions $\{\phi_n\}_{n=1}^N$. Second, in a *data-driven* conditioning step, a new set of few expressive basis functions $\{\rho_m\}_{m=1}^M$, $M < N$, is constructed from the most significant patterns in the Hilbert-GP coefficients $\boldsymbol{w}_j$ without expert knowledge. Based on the obtained low-dimensional approximation, efficient online inference and learning is accomplished using standard PF [18].

of basis functions $\{\phi_n(\boldsymbol{x}_k)\}_{n=1}^N$. The transformation is done by data-driven extraction of the most significant patterns that account for the change in $\Xi(\cdot, j)$. These patterns are used to condition a new set of few expressive basis functions $\{\rho_m(\boldsymbol{x}_k)\}_{m=1}^M$ along which online learning is performed efficiently in a restricted subspace $V^{(\rho,M)} \subset V^{(\phi,N)}$ with less DOF, i. e., $M < N$. Due to the simplicity of the resulting estimation problem, a standard noise-adaptive PF [18] yields sufficient performance. The methodological steps are illustrated in Fig. 2.

In Sec. III-A and Sec. III-B, we consider single-task regression of the $i$-th target function $\Xi_i$ and refer to it as $\Xi$ to avoid notation clutter. However, methods for multi-task regression follow trivially by using $n_\xi$ single-task regressors in parallel.

### A. Capturing the Function Space using Hilbert-GP

As a generic representation for learning, we model $\hat{\Xi} \sim \mathcal{GP}(0, \kappa(\boldsymbol{x}_k, \boldsymbol{x}'_k))$ which allows to incorporate prior assumptions regarding $\Xi$, e. g., smoothness, intuitively by choosing a kernel $\kappa(\boldsymbol{x}_k, \boldsymbol{x}'_k)$ with corresponding hyperparameters. In particular, we use the GP approximation presented in [11] due to its beneficial orthogonality properties and its integration in existing inference and learning schemes [9], [5], [6]. The formulation relies on a basis function expansion, and we will refer to it as "Hilbert-GP" in the following. The concept is briefly revisited along with the presentation of the proposed method. For a detailed introduction, we refer to [11]. The main idea is to approximate the kernel using $N$ basis functions $\phi_n(\boldsymbol{x}_k)$ according to

$$\kappa(\boldsymbol{x}_k, \boldsymbol{x}'_k) \approx \sum_{n=1}^N S\left(\sqrt{\lambda_n}\right) \phi_n(\boldsymbol{x}_k) \phi_n(\boldsymbol{x}'_k), \qquad (2)$$

where $S\left(\sqrt{\lambda_n}\right)$ is a factor for encoding the GP prior in frequency domain and will be explained later. Using (2), $\Xi(\boldsymbol{x}_k, \cdot)$ is approximated by a finite-dimensional basis function expansion

$$\hat{\Xi}^{(N)}(\boldsymbol{x}_k) = \sum_{n=1}^N w_n \phi_n(\boldsymbol{x}_k) = \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}_k), \qquad (3)$$

with coefficient vectors $\boldsymbol{w}^\top = \begin{bmatrix} w_1 & \dots & w_N \end{bmatrix}$, $\boldsymbol{w} \in \mathbb{R}^N$, resulting in $n_\xi N$ parameters to be found in multi-task regression. The employed basis functions

$$\phi_n(\boldsymbol{x}_k) \triangleq \prod_{i=1}^{n_x} \frac{1}{\sqrt{L_i}} \sin\left(\frac{\pi j_i (x_{k,i} + L_i)}{2L_i}\right), \qquad (4)$$

are eigenfunctions of the Laplace operator and span the function space $V^{(\phi,N)}$ for input features $\boldsymbol{x}_k \in \Omega \subset \mathbb{R}^{n_x}$ on a hypercube domain $\Omega = [-L_1, L_1] \times \dots \times [-L_{n_x}, L_{n_x}]$. In the limit $N, L_1, \dots L_{n_x} \to \infty$, the basis function expansion converges to the actual GP [11]. Please note, the eigenfunctions form an orthonormal basis with respect to the inner product $\langle \phi_{n_1}, \phi_{n_2} \rangle$ and have associated eigenvalues

$$\lambda_n \triangleq \sum_{i=1}^{n_x} \left(\frac{\pi j_i}{2L_i}\right)^2, \qquad (5)$$

in which each basis function features a unique combination of integers $(j_1, \dots, j_{n_x})$ that is chosen to maximize the basis functions expressiveness.

The GP prior is incorporated by finding a set of weights $w_j$ such that the power spectrum of the chosen covariance kernel $\kappa(\boldsymbol{x}_k, \boldsymbol{x}'_k)$ is replicated according to (2). Here, we employ a squared exponential kernel $\kappa_{\text{se}}(\boldsymbol{x}_k, \boldsymbol{x}'_k)$, as it has been employed successfully for similar settings [9], [5]. The corresponding kernel and spectral density $S_{\text{se}}(\omega)$ are

$$\kappa_{\text{se}}(\boldsymbol{x}_k, \boldsymbol{x}'_k) = \sigma^2 \exp\left(-\frac{\|\boldsymbol{x}_k - \boldsymbol{x}'_k\|^2}{2l^2}\right), \qquad (6)$$

$$S_{\text{se}}(\omega) = \sigma^2 \sqrt{2\pi l^2} \exp\left(-\frac{l^2 \omega^2}{2}\right), \qquad (7)$$

with hyperparameters $\sigma^2$, $l$ to be defined by the user. Equipped with the basis functions (4), and the corresponding eigenvalues (5), the varying behavior of target function $\Xi$ is captured by finding suitable coefficient vectors $\boldsymbol{w}_j$ for each realization $\Xi(\cdot, j)$, $j = 1, \dots, J$ in the data set $\mathcal{D}$. This is accomplished by computing the posterior distribution of the coefficients $\boldsymbol{w}_j$ [11], [15] or equivalently solving the regularized least squares problem

$$\boldsymbol{w}_j = \arg\min_{\bar{\boldsymbol{w}}} \sum_{k=1}^K \left(\xi_k^j - \bar{\boldsymbol{w}}^\top \boldsymbol{\phi}(\boldsymbol{x}_k^j)\right)^2 + \sigma_\xi^2 \|\bar{\boldsymbol{w}}\|^2_{\boldsymbol{V}^{-1}}, \qquad (8)$$

where the GP prior is encoded by setting the diagonal regularization matrix $\boldsymbol{V}$ with entries $S_{\text{se}}\left(\sqrt{\lambda_n}\right)$, $n = 1, \dots, N$, following the lines of [11]. The combined hyperparameters $\boldsymbol{\vartheta} = \{\sigma_\xi^2, \sigma^2, l\}$ can be optimized as described in [11].

### B. Data-driven Conditioning

Having captured the shape of $J$ target function realizations in the coefficient vectors $\boldsymbol{w}_j$ of approximation $\hat{\Xi}^{(N)}$, a new set of expressive basis functions is constructed in a *data-driven* fashion. It is worth noting that the target function $\Xi(\cdot, j)$ usually revisits similar shapes $j \in [1, J]$ at different time instants in physical systems. As an example, the tire-road friction characteristic is antisymmetric and can be expressed as combinations of $\arctan$ derivates [13], [6]. Instead of leveraging this by choosing basis functions from expert knowledge, we build a matrix of Hilbert-GP parametrizations for the finite set of realizations $\Xi\left(\cdot, j\right)$, $j = 1, \ldots, J$, as

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_1 & \ldots & \boldsymbol{w}_J \end{bmatrix}^\top = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{Z}^\top, \tag{9}$$

and perform a Singular Value Decomposition (SVD), yielding the unitary matrices $\boldsymbol{U} \in \mathbb{R}^{J\times J}$, $\boldsymbol{Z} \in \mathbb{R}^{N\times N}$, and a matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{J\times N}$ with singular values $\sigma_j$ as diagonal entries in decreasing order.

Now, the $M$ most significant DOF can be extracted by choosing the first $M$ columns $\{\boldsymbol{z}_m\}_{m=1}^{M}$ of $\boldsymbol{Z}$ to define a new set of expressive basis functions $\{\rho_m\left(\boldsymbol{x}_k\right)\}_{m=1}^{M}$ with

$$\rho_m(\boldsymbol{x}_k) = \boldsymbol{z}_m^\top \boldsymbol{\phi}(\boldsymbol{x}_k). \tag{10}$$

The user-defined hyperparameter $M$ is chosen as a trade-off between modeling accuracy and computational complexity of the resulting inference and learning algorithm (which will be described in Sec. III-C). The basis functions span the subspace $V^{(\rho,M)}$, restricted to "realistic" shapes of the target function $\Xi\left(\cdot, j\right)$. Loosely speaking, each of the basis functions $\rho_m$ corresponds to a specific composition of original basis functions $\phi_n$ and enables learning along a distinct DOF. Moreover, the new set of basis functions inherits orthogonality properties from vectors $\boldsymbol{z}_m$ and original basis functions $\phi_n$.

**Lemma 1** The basis functions $\{\rho_m(\boldsymbol{x}_k)\}_{m=1}^{M}$ form an orthonormal set with respect to the inner product $\langle\rho_i, \rho_j\rangle$, i. e.,

$$\int_\Omega \rho_i(\boldsymbol{x}_k)\rho_j(\boldsymbol{x}_k)\mathrm{d}\boldsymbol{x}_k = \delta_{ij}. \tag{11}$$

*Proof:* See appendix, page 8. ■

Using the functions $\{\rho_m(\boldsymbol{x})\}_{m=1}^{M}$, a low-dimensional formulation for modeling $\Xi$ can be constructed according to

$$\hat{\Xi}^{(M)}(\boldsymbol{x}_k) = \sum_{m=1}^{M} v_m \rho_m(\boldsymbol{x}_k) = \boldsymbol{v}^\top \boldsymbol{\rho}(\boldsymbol{x}_k), \tag{12}$$

in which $\boldsymbol{v}^\top = \begin{bmatrix} v_1 & \ldots & v_M \end{bmatrix}$, $\boldsymbol{v} \in \mathbb{R}^M$ is a coefficient vector with $M < N$. The number of parameters to be determined in multi-task regression is $n_\xi M$, independent of the number of original basis functions $N$. Thus, a high-dimensional and accurate Hilbert-GP formulation can be tailored to learn $\Xi(\cdot, j)$ online with few DOF, significantly reducing the complexity of the inference problem.

In Fig. 3, the approximation accuracy of both basis function expansions (3) and (12) in a numerical example is shown for a set of realizations $\Xi(\cdot, j)$, $j = 1, \ldots, J$, depending on the number of parameters to be learned online. The results suggest that significantly fewer DOF are required with the proposed approach. In particular, the accuracy of the original Hilbert-GP increases step-wise with every second DOF, indicating that basis functions with even integers $j_i$ in (4) are inexpressive to represent the chosen target function due to symmetry. In [6], this is exploited to facilitate learning based on expert knowledge. In contrast, the proposed method exploits the DOF effectively in a *data-driven* fashion.

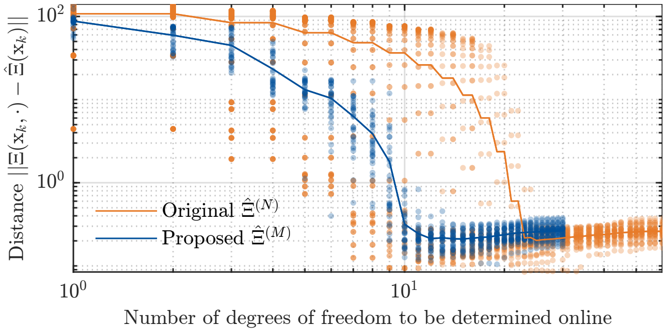

Fig. 3. Error between the true function $\Xi(x_k, j) = 10\mathrm{sinc}(jx_k/100)$ and the basis function expansions $\hat{\Xi}^{(i)}(x_k)$ for $i = N, M$ and for different realizations $j = 1, \ldots, 30$ on the domain $\Omega = [-15, 15]$. Each error result corresponds to a dot, and the mean is drawn as a line. The number of required DOF to achieve a certain approximation performance is significantly reduced using the proposed approach due to the choice of expressive basis functions.

Moreover, the distance between $\hat{\Xi}^{(N)}(\boldsymbol{x}_k)$ and $\hat{\Xi}^{(M)}(\boldsymbol{x}_k)$ can be quantified prior to evaluation.

**Theorem 1** The distance $d(\boldsymbol{x}_k) = \hat{\Xi}^{(N)}(\boldsymbol{x}_k) - \hat{\Xi}^{(M)}(\boldsymbol{x}_k)$ between the basis function expansions $\hat{\Xi}^{(N)}(\boldsymbol{x}_k)$ and $\hat{\Xi}^{(M)}(\boldsymbol{x}_k)$ on the domain $\Omega$ is given by

$$\|d(\boldsymbol{x}_k)\|^2 = \left\|\boldsymbol{w} - \sum_{m=1}^{M} v_m \boldsymbol{z}_m\right\|^2. \tag{13}$$

*Proof:* See appendix, page 8. ■

**Remark 1** If the target functions in multi-task regression $\Xi_i(\cdot, j)$, $i = 1, ..., n_\xi$, are of significantly different shape, the number of basis functions needs to be increased to capture relevant characteristics. Alternatively, separate Hilbert-GPs and/or conditioning steps per target can be employed.

**Remark 2** If we choose $v_m = \sum_{m=1}^{M} u_{jm}\sigma_m$ and $M = \min\left(J, N\right)$ for modeling realization $j$, with $u_{jm}$ and $\sigma_m$ being elements of $\boldsymbol{U}$ and $\boldsymbol{\Sigma}$, the distance $\left\|d(\boldsymbol{x}_k)\right\|^2 = 0$, and the original basis function expansion is recovered.

**Remark 3** In practice, we see that the approximation error of the reduced-order representation with respect to the true function does not decrease asymptotically towards the original basis function expansion error as we add DOF (see Fig. 3). Instead, $\hat{\Xi}^{(M)}$ usually reaches the highest accuracy for $M < J$, i. e., if only the most significant DOF are used.

**Remark 4** As the underlying *dominant features* in the offline data set are extracted, the learnable function space $V^{(\rho,M)}$ is bound to linear combinations of these dominant features. Therefore, $V^{(\rho,M)}$ can contain function shapes (i. e., linear feature combinations) that are not present in the offline data set, with potential implications for generalization beyond the training data distribution.

For online inference and learning of a Hilbert-GP, specifically tailored SMC have been proposed in [5], [6]. In contrast, the low-dimensional basis function expansion $\hat{\Xi}^{(M)}$

simplifies the estimation problem significantly, such that a standard PF [18] can be used here.

### C. Efficient Online Inference and Learning

Having obtained the new set of orthonormal basis functions $\{\rho_m(\boldsymbol{x}_k)\}_{m=1}^{M}$, efficient learning of the target functions $\Xi_i(\cdot, j)$, $i = 1, \ldots, n_\xi$, in the restricted subspace $V^{(\rho,M)}$ is possible using standard Bayesian inference methods. To this end, we formulate a system model that contains the parameters $\boldsymbol{v}_i$ to be learned online following an ad-hoc state augmentation approach, as done in a related setting [14]. The reason for this is two-fold:

(i) Parameter estimation by state augmentation is a baseline approach that, despite its simplicity, yields sufficient estimation quality for a wide range of applications and is thus commonly applied and accepted in practice.

(ii) The target function $\Xi$ is nonlinearly *nested* inside known system dynamics $\boldsymbol{f}$. Due to the nonlinearly nested structure (1) in the present setting, the inverse model $\Xi(\boldsymbol{x}_k, j) = \boldsymbol{f}^{-1}(\boldsymbol{x}_{k+1})\big|_{\boldsymbol{x}_k,\boldsymbol{u}_k}$ is not generally known for updating the posterior of $\boldsymbol{v}_i$, formally obstructing direct application of [9], [5], [6].

In this light, we model the system (1) using the basis function expansion $\hat{\Xi}^{(M)}$ (restricted to the subspace $V^{(\rho,M)}$), by

$$\tilde{\boldsymbol{x}}_{k+1} = \boldsymbol{F}(\tilde{\boldsymbol{x}}_k, \boldsymbol{u}_k) + \tilde{\boldsymbol{e}}_k \tag{14a}$$

$$= \begin{bmatrix} \boldsymbol{x}_{k+1} \\ \boldsymbol{v}_{1,k+1} \\ \vdots \\ \boldsymbol{v}_{n_\xi,k+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k, \hat{\Xi}^{(M)}(\boldsymbol{x}_k)) \\ \boldsymbol{v}_{1,k} \\ \vdots \\ \boldsymbol{v}_{n_\xi,k} \end{bmatrix} + \begin{bmatrix} \boldsymbol{e}_k^{x} \\ \boldsymbol{e}_k^{v_1} \\ \vdots \\ \boldsymbol{e}_k^{v_{n_\xi}} \end{bmatrix},$$

$$\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{e}_k^{y}, \tag{14b}$$

with a random walk assumption on the parameters $\boldsymbol{v}_i$. The parameter noise for target function $\Xi_i$ is shaped according to the significance of the respective DOF (i. e., the basis function $\rho_m(\boldsymbol{x}_k)$), represented by the truncated singular value matrix $\boldsymbol{\Sigma}_i$ from the conditioning step. Hence, the parameter process noise is drawn $\boldsymbol{e}_k^{v_i} \sim \mathcal{N}(\boldsymbol{0}, c\boldsymbol{\Sigma}_i)$, with $c$ being a user-defined design parameter that scales the exploration capability. Hence, the overall process noise has a block-diagonal covariance $\tilde{\boldsymbol{Q}}$ and is drawn $\boldsymbol{e}_k^{\tilde{x}} \sim \mathcal{N}(\boldsymbol{0}, \tilde{\boldsymbol{Q}})$.

Based on the model formulation (14), the noise-adaptive marginalized particle filter proposed in [18] is employed. The motivation for noise adaptation is that, if the true function $\Xi$ changes rapidly, the error between measured evidence and the current estimate of the system state increases. In other words, the current approximation $p(\tilde{\boldsymbol{x}}_k|\tilde{\boldsymbol{x}}_{0:k-1}, \boldsymbol{y}_{0:k-1})$ does not represent new evidence $\boldsymbol{y}_k$ well which would lead to rapid algorithm divergence without noise adaptivity.

In particular, we model the measurement noise covariance $\boldsymbol{R}_k$ with an inverse Wishart distribution $\boldsymbol{R}_k \sim \mathcal{IW}(\nu_k, \boldsymbol{\Lambda}_k)$, which is the conjugate prior for the multivariate normal distribution. In principle, the process noise of the parameters could be adapted as well, but this lead to frequent path degeneracy in simulative case studies. For self-contained presentation, the main steps of the noise-adaptive particle filter in [18] are presented and connected to the problem at hand subsequently. For notational clarity, dependence on the inputs $\boldsymbol{u}_k$ is omitted.

The overall target is the joint probability density function of state trajectory $\tilde{\boldsymbol{x}}_{0:k}$ and current noise parameters $\boldsymbol{\theta}_k = \boldsymbol{R}_k$, given the measurements $\boldsymbol{y}_{0:k}$

$$p(\tilde{\boldsymbol{x}}_{0:k}, \boldsymbol{\theta}_k|\boldsymbol{y}_{0:k}) = \underbrace{p(\boldsymbol{\theta}_k|\tilde{\boldsymbol{x}}_{0:k}, \boldsymbol{y}_{0:k})}_{\text{Posterior (II)}} \underbrace{p(\tilde{\boldsymbol{x}}_{0:k}|\boldsymbol{y}_{0:k})}_{\text{Posterior (I)}}, \tag{15}$$

that is composed of the recursively computed posterior density of the states $p(\tilde{\boldsymbol{x}}_{0:k}|\boldsymbol{y}_{0:k})$ and the posterior density $p(\boldsymbol{\theta}_k|\tilde{\boldsymbol{x}}_{0:k}, \boldsymbol{y}_{0:k})$ of the noise parameters. The posterior (I) is approximated by a set of $N_p$ weighted particles

$$p(\tilde{\boldsymbol{x}}_{0:k}|\boldsymbol{y}_{0:k}) \approx \sum_{i=1}^{N_p} q_k^i \delta_{\tilde{\boldsymbol{x}}_{0:k}^i, \tilde{\boldsymbol{x}}_{0:k}}, \tag{16}$$

that represent different state trajectories. The weights $q_k^i$ capture the probability of the respective trajectory at the current time instant $k$ and are recursively updated as

$$q_k^i \propto q_{k-1}^i \frac{p(\boldsymbol{y}_k|\tilde{\boldsymbol{x}}_{0:k}^i, \boldsymbol{y}_{0:k-1})\, p(\tilde{\boldsymbol{x}}_k^i|\tilde{\boldsymbol{x}}_{0:k-1}^i, \boldsymbol{y}_{0:k-1})}{\pi(\tilde{\boldsymbol{x}}_k^i|\tilde{\boldsymbol{x}}_{0:k-1}^i, \boldsymbol{y}_{0:k})}, \tag{17}$$

when new measurement evidence $\boldsymbol{y}_k$ becomes available. The density $\pi(\tilde{\boldsymbol{x}}_k^i|\tilde{\boldsymbol{x}}_{0:k-1}^i, \boldsymbol{y}_{0:k})$ is a tractable proposal distribution from which the states are drawn [19]. The employed likelihood $p(\boldsymbol{y}_k|\tilde{\boldsymbol{x}}_{0:k}^i, \boldsymbol{y}_{0:k-1})$ is obtained by integrating out the noise parameters particle-based, according to

$$p(\boldsymbol{y}_k|\tilde{\boldsymbol{x}}_{0:k}, \boldsymbol{y}_{0:k-1}) = \int p(\boldsymbol{y}_k|\tilde{\boldsymbol{x}}_k, \boldsymbol{\theta}_{k-1}) \times p(\boldsymbol{\theta}_{k-1}|\tilde{\boldsymbol{x}}_{0:k}, \boldsymbol{y}_{0:k-1})\, \mathrm{d}\boldsymbol{\theta}_{k-1}, \tag{18}$$

which, due to the inverse Wishart prior, is a Student-t ($\mathcal{T}$) distribution

$$p(\boldsymbol{y}_k|\tilde{\boldsymbol{x}}_k^i) = \mathcal{T}\left(\boldsymbol{h}(\boldsymbol{x}_k^i, \boldsymbol{u}_k), \boldsymbol{\Lambda}_k, \nu_k - n_y + 1\right), \tag{19}$$

for each particle, dependent on its current noise statistics. As stated earlier, if the underlying function $\Xi$ changes suddenly, the current approximation $p(\tilde{\boldsymbol{x}}_k^i|\tilde{\boldsymbol{x}}_{0:k-1}^i, \boldsymbol{y}_{0:k-1})$ does not represent new evidence $\boldsymbol{y}_k$ well, leading to rapid algorithm divergence without noise adaptivity. The inverse Wishart prior on $\boldsymbol{R}_k$ accounts for this by adapting the noise covariance, effectively exploring a larger search space.

Given the state estimates $\tilde{\boldsymbol{x}}_k^i$, the posterior density (II) can be evaluated. As the parameter posterior is again an inverse Wishart distribution $\mathcal{IW}(\nu_k, \boldsymbol{\Lambda}_k)$, this amounts to updating the parameter statistics with the "measurement" $\boldsymbol{p}_k = \boldsymbol{y}_k - \boldsymbol{h}(\tilde{\boldsymbol{x}}_k)$ according to

$$\nu_{k|k} = \nu_{k|k-1} + 1, \tag{20a}$$

$$\boldsymbol{\Lambda}_{k|k} = \boldsymbol{\Lambda}_{k|k-1} + \boldsymbol{p}_k \boldsymbol{p}_k^{\top}. \tag{20b}$$

The initial covariance is sampled $\boldsymbol{R}_0 \sim \mathcal{IW}(\nu_0, \boldsymbol{\Lambda}_0)$. If the noise parameters are time-varying, a forgetting factor $\lambda_{\mathrm{f}}$ can be incorporated in the prediction step of the statistics to reduce the impact of old observations and introduce mixing [5], [19]. The resulting method is implemented as a bootstrap PF and summarized in Algorithm 1.

**Algorithm 1** Pseudo-code of the proposed algorithm

**Data-driven conditioning:** Compute expressive basis functions $\{\rho_m(\boldsymbol{x}_k)\}_{m=1}^{M}$ according to (10) and choose initial coefficients $\{\boldsymbol{v}_{j,0}\}_{j=1}^{n_\xi}$.

**Initialize:** Set $\{\tilde{\boldsymbol{x}}_0^i\}_{i=1}^{N_p} \sim p(\tilde{\boldsymbol{x}}_0)$, $\{\boldsymbol{\Lambda}_0^i, \nu_0^i\}_{i=1}^{N_p} = \{\boldsymbol{\Lambda}_0, \nu_0\}$, $\lambda_\mathrm{f} \in (0, 1]$.

```
for $k = 1, \dots$ do
    Read current data $\boldsymbol{u}_{k-1}$, $\boldsymbol{y}_k$.
    for $i = 1, \dots, N_p$ do
        Time update of noise statistics
        $\nu_{k|k-1}^i = \lambda_\mathrm{f} \nu_{k-1|k-1}^i$, $\boldsymbol{\Lambda}_{k|k-1}^i = \lambda_\mathrm{f} \boldsymbol{\Lambda}_{k-1|k-1}^i$.
        Sample $\tilde{\boldsymbol{x}}_k^i \sim \mathcal{N}(\boldsymbol{F}(\tilde{\boldsymbol{x}}_{k-1}^i, \boldsymbol{u}_{k-1}), \tilde{\boldsymbol{Q}})$.
        Compute weight $\bar{q}_k^i = p\big(\boldsymbol{y}_k | \boldsymbol{x}_k^i\big)$ using (19).
        Measurement update of noise statistics (20).
    end for
    Normalize weights $q_k^i = \bar{q}_k^i / \sum_{i=1}^{N_p} \bar{q}_k^i$.
    Compute estimates $\hat{\boldsymbol{x}}_k, \hat{\boldsymbol{v}}_{1,k}, \dots, \hat{\boldsymbol{v}}_{n_\xi,k}$.
    Resample particles and copy the
    corresponding noise statistics.
end for
```

## IV. SIMULATION RESULTS

For evaluation of Algorithm 1, a numerical simulation example inspired by the nonlinear battery model in [2] is used. The continuous-time state-space model

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{z} \\ \dot{V}_1 \\ \dot{T}_\mathrm{c} \end{bmatrix} = \begin{bmatrix} I Q_\mathrm{bat}^{-1} \\ -\alpha\left(z, j\right) V_1 + \beta\left(z\right) I \\ \frac{1}{C_\mathrm{c}} \left( V_1 I + R_0\left(z, I\right) I^2 - \frac{T_\mathrm{c} - T_\mathrm{a}}{R_\mathrm{c}} \right) \end{bmatrix}, \quad (21a)$$

$$\boldsymbol{y} = \begin{bmatrix} z, & V_0\left(z\right) + V_1 + R_0\left(z, I\right) I, & T_\mathrm{c} \end{bmatrix}^\top, \quad (21b)$$

exhibits 3 states and 3 outputs. The “charging current” $I$ is a scalar input. A discrete-time representation is obtained by 4-th order Runge-Kutta integration (RK4) with $0.01$ seconds step width. For simulation, additive process and measurement noise $\boldsymbol{e}_k^x \sim \mathcal{N}(\boldsymbol{0}, 10^{-5} \times \mathbf{I}_3)$ and $\boldsymbol{e}_k^y \sim \mathcal{N}(\boldsymbol{0}, 10^{-2} \times \mathbf{I}_3)$ is considered, respectively. The nested function $\alpha(z_k, j)$ represents a nonlinear and parametric relationship related to an RC circuit that varies over time and is to be learned online. Thus, $\Xi(x_1, j) = \alpha(z_k, j)$ and the true function $\alpha(z_k, j) = 4j - 8j\left(0.5 - z_k\right)^3$ for $j = 1, \dots, 10$. Please note, that $\alpha$ is nested nonlinearly in the discrete-time system dynamics due to RK4 discretization. The remaining quantities are physical parameters and nonlinear functions. For further details, the reader is referred to [2]. In the evaluation scenario, a wrong initialization of $\hat{\Xi}$ is set, and at time step $k = 1,000$, a sudden change from $\alpha(\cdot, 1)$ to $\alpha(\cdot, 10)$ is simulated. In both cases, the true function shape should be found.

To initialize Algorithm 1, we condition $M = 2$ expressive basis functions $\rho_m$ based on coefficients for $N = 50$ original basis functions $\phi_n$, leveraging that the number of DOF to be learned online is independent of $N$. The user-defined parameters $\boldsymbol{\Lambda}_0 = \mathbf{I}_3$, $\nu_0 = 3$, $c = 3 \times 10^{-5}$, and the number of particles $N_p = 100$.

For comparison, we first use, as a baseline approach, a Hilbert-GP combined with the PF described in Sec. III-C. Second, as a state-of-the-art online inference and learning method, we use the tailored marginalized PF for online learning of a Hilbert-GP state-space model proposed in [5]. All Hilbert-GP representations in the comparison study are pre-trained and initialized with coefficient vectors corresponding to the function realization $\alpha(\cdot, 5)$ (i. e., all methods start with the same initial conditions). The hyperparameters $\sigma^2$, $l$ and $\sigma_\xi^2$ are optimized according to [11] using the offline data set $\mathcal{D}$. The remaining design parameters in the comparison methods are hand-tuned.

In Fig. 4, true values and estimates for state $V_1$ and target function $\alpha$ are shown in the top and bottom plots, respectively. In the middle plot, the estimation error regarding the target function is depicted. Despite the use of up to $2,000$ particles, the baseline, and the state-of-the-art comparison both show slower convergence than the proposed approach, which can be attributed to the complex inference task of determining the parameters of a highly flexible, unconditioned Hilbert-GP. In contrast, Algorithm 1 converges rapidly to the true function $\alpha$ both, from a wrong initialization and after a sudden change of the system behavior. The learning effect is visible in the estimates of $V_1$ as well. The overall state estimate accuracy is comparable across the different methods in the present example simulation.

The execution of the unoptimized code took on average $5\,\mathrm{ms}$ (PF & Hilbert-GP, $N_p = 100$), $3\,\mathrm{ms}$ ([5], $N_p = 100$), $57\,\mathrm{ms}$ ([5], $N_p = 2,000$), and $7\,\mathrm{ms}$ (Algorithm 1, $N_p = 100$) per time step, respectively (i5-1235U CPU, 8 GB RAM). The results indicate that Algorithm 1 converges faster to the target function than a state-of-the-art method which uses significantly more particles and computational resources. In this light, Algorithm 1 can be considered computationally efficient thanks to the reduced number of required particles.

## V. CONCLUSIONS

In the current work, a data-driven offline conditioning step for efficient online inference of latent states and learning of an unknown target function is proposed. The key idea is to restrict learning to a low-dimensional subspace spanned by expressive Degrees of Freedom (DOF) without prior expert knowledge about the target function. In operation, online inference and learning is performed efficiently along these DOF. Compared to a baseline method and a state-of-the-art method, the proposed approach yields a significantly simplified estimation problem without relying on expert knowledge about the target function. Moreover, the scheme is capable of learning a nonlinearly *nested* target function inside a first-principles model, which is addressed only in few existing works on online inference and learning.

Thus, we contribute a method that has the potential to facilitate the operation of real-world systems in complex and changing conditions. Specifically, the proposed scheme provides a further step towards intelligent operation under fluctuating resistance forces, changing geometries, and parameters due to varying temperature, location, or time. Relevant real-world effects include wear and aging.

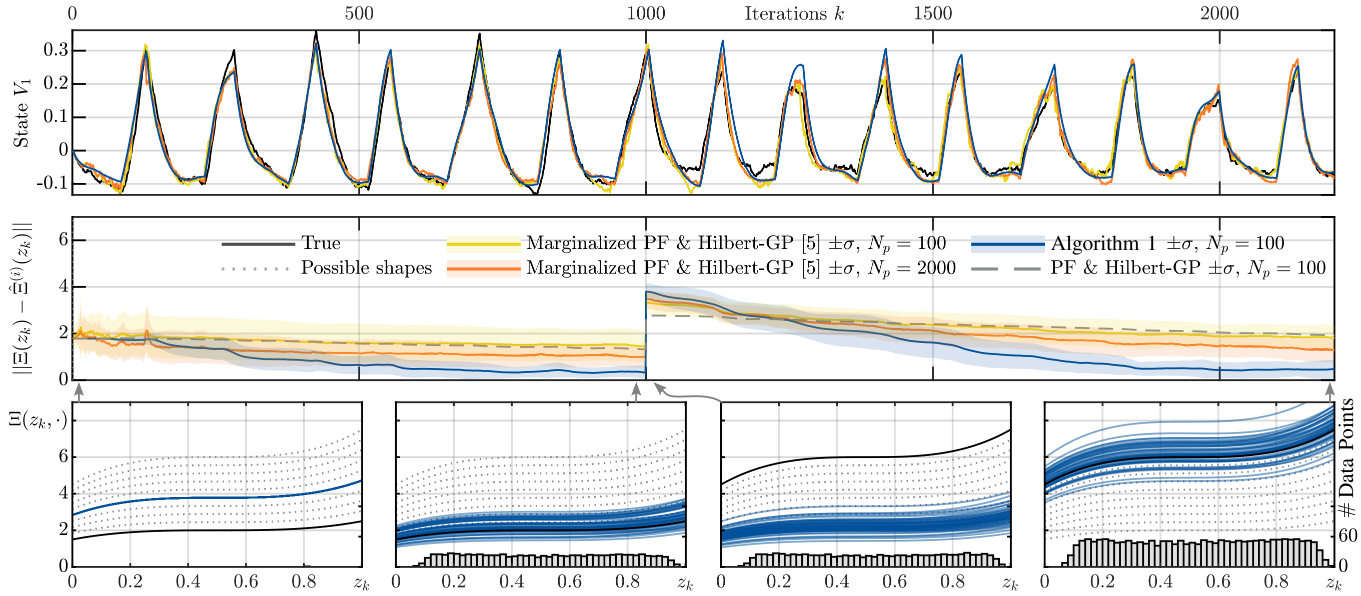

Fig. 4. Online inference and learning in the simulation example (21) using the proposed Algorithm 1 for learning with the expressive basis function expansion $\hat{\Xi}^{(M)}$ and the PF described in Sec. III-C. Algorithm 1 converges rapidly from a wrong initial condition and after a sudden change in the true function $\Xi$ at $k = 1,000$. The shown error is the mean for 50 Monte-Carlo runs (standard deviation $\sigma$ shown semi-transparent), and the estimation results $\hat{\Xi}^{(M)}$ for the respective Monte-Carlo runs are presented in the bottom plots for the indicated iterations.

In future research, the proposed conditioning step to extract expressive basis functions might be integrated into other online inference and learning schemes, e. g., [5], to facilitate learning. Another interesting research direction is to incorporate conditioning directly in offline particle Markov chain Monte Carlo methods to obtain a set of expressive basis functions for online inference and learning.

## ACKNOWLEDGMENT

For implementation of the proposed methods, parts of the program code of [3] have been used and modified. Jan-Hendrik Ewering would like to thank Prof. Manon Kok for making the implementation of [3] publicly available.

## REFERENCES

[1] R. Ishiyama, E. Fukuyama, and B. Enescu, “Estimation of time-variable friction parameters using machine learning,” *Geophys. Journal Int.*, vol. 236, no. 1, pp. 395–412, 2023.

[2] A. Aitio, D. Jöst, D. U. Sauer, and D. A. Howey, “Learning battery model parameter dynamics from data with recursive Gaussian process regression,” preprint, arXiv, 2023.

[3] M. Kok, A. Solin, and T. B. Schön, “Rao-blackwellized particle smoothing for simultaneous localization and mapping,” *Data-Centric Engineering*, vol. 5, p. e15, 2024.

[4] F. Viset, R. Helmons, and M. Kok, “An Extended Kalman Filter for Magnetic Field SLAM Using Gaussian Process Regression,” *Sensors (Basel, Switzerland)*, vol. 22, no. 8, 2022.

[5] K. Berntorp, “Online Bayesian inference and learning of Gaussian-process state–space models,” *Automatica*, vol. 129, p. 109613, 2021.

[6] K. Berntorp and M. Menner, “Online Constrained Bayesian Inference and Learning of Gaussian-Process State-Space Models,” in *American Control Conf.* IEEE, 2022, pp. 940–945.

[7] N. Lampe, Z. Ziaukas, C. Westerkamp, and H.-G. Jacob, “Analysis of the Potential of Onboard Vehicle Sensors for Model-based Maximum Friction Coefficient Estimation,” in *American Control Conf.* IEEE, 2023, pp. 1622–1628.

[8] A. Svensson, A. Solin, S. Särkkä, and T. B. Schön, “Computationally Efficient Bayesian Learning of Gaussian Process State Space Models,” in *Conf. on AI and Statist.*, vol. 51. PMLR, 2016, pp. 213–221.

[9] A. Svensson and T. B. Schön, “A flexible state–space model for learning nonlinear dynamical systems,” *Automatica*, vol. 80, pp. 189–199, 2017.

[10] M. Buisson-Fenet, V. Morgenthaler, S. Trimpe, and F. Di Meglio, “Recognition Models to Learn Dynamics from Partial Observations with Neural ODEs,” *Trans. on Machine Learning Research*, 2023.

[11] A. Solin and S. Särkkä, “Hilbert space methods for reduced-rank Gaussian process regression,” *Statistics and Computing*, vol. 30, no. 2, pp. 419–446, 2020.

[12] C. Andrieu, A. Doucet, and R. Holenstein, “Particle Markov Chain Monte Carlo Methods,” *Journal of the Royal Statistical Society*, vol. 72, no. 3, pp. 269–342, 2010.

[13] H. B. Pacejka and E. Bakker, “The Magic Formula Tyre Model,” *Vehicle System Dynamics*, vol. 21, no. sup001, pp. 1–18, 1992.

[14] A. Kullberg, I. Skog, and G. Hendeby, “Online Joint State Inference and Learning of Partially Unknown State-Space Models,” *IEEE Trans. Signal Process.*, vol. 69, pp. 4149–4161, 2021.

[15] C. Menzen, E. Memmel, K. Batselier, and M. Kok, “Projecting basis functions with tensor networks for Gaussian process regression,” *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 7288–7293, 2023.

[16] T. B. Schön, “Nonlinear System Identification Using Particle Filters,” in *Encyclopedia of Systems and Control*, J. Baillieul and T. Samad, Eds. Cham: Springer International Publishing, 2021, pp. 1483–1492.

[17] A. Wigren, J. Wagberg, F. Lindsten, A. G. Wills, and T. B. Schön, “Nonlinear System Identification: Learning While Respecting Physical Models Using a Sequential Monte Carlo Method,” *IEEE Control Systems*, vol. 42, no. 1, pp. 75–102, 2022.

[18] E. Özkan, V. Šmídl, S. Saha, C. Lundquist, and F. Gustafsson, “Marginalized adaptive particle filtering for nonlinear models with unknown time-varying noise parameters,” *Automatica*, vol. 49, no. 6, pp. 1566–1575, 2013.

[19] S. Särkkä and L. Svensson, *Bayesian filtering and smoothing*, 2nd ed. New York: Cambridge Univ. Press, 2023.

## APPENDIX

### A. Proof of Lemma 1

For notational convenience, we omit the discrete-time index $k$ in this proof. To show the orthogonality of the basis functions $\{\rho_i(\boldsymbol{x})\}_{i=1}^{M}$, we start by inserting the definitions of the basis functions in (11), noting that the functions are real-valued and expanding the products, which yields

$$\begin{aligned}\langle\rho_i,\rho_j\rangle &= \int_\Omega \rho_i(\boldsymbol{x})\rho_j(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\\ &= \int_\Omega \left(\boldsymbol{z}_i^\top\boldsymbol{\phi}(\boldsymbol{x})\right)\left(\boldsymbol{z}_j^\top\boldsymbol{\phi}(\boldsymbol{x})\right)\mathrm{d}\boldsymbol{x}.\end{aligned} \tag{22}$$

Expanding the product into two sums and rearranging gives

$$\begin{aligned}\langle\rho_i,\rho_j\rangle &= \int_\Omega \rho_i(\boldsymbol{x})\rho_j(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\\ &= \int_\Omega \left(\boldsymbol{z}_i^\top\boldsymbol{\phi}(\boldsymbol{x})\right)\left(\boldsymbol{z}_j^\top\boldsymbol{\phi}(\boldsymbol{x})\right)\mathrm{d}\boldsymbol{x}\\ &= \int_\Omega \left(\sum_{n_1=1}^{N} z_{in_1}\phi_{n_1}(\boldsymbol{x})\right)\left(\sum_{n_2=1}^{N} z_{jn_2}\phi_{n_2}(\boldsymbol{x})\right)\mathrm{d}\boldsymbol{x}\\ &= \int_\Omega \sum_{n_1=1}^{N}\sum_{n_2=1}^{N} z_{in_1}z_{jn_2}\phi_{n_1}(\boldsymbol{x})\phi_{n_2}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}.\end{aligned} \tag{23}$$

The order of the integral(s) and the sums can be changed because the sums are uniformly convergent. This can be seen, if we reorder the summands by decreasing order of $z_{in_1}z_{jn_2}$ and extend the finite sums to an infinite series by adding zeros for $n_1n_2 > N^2$. In this case, uniform convergence is provided by Dirichlet's test for uniform convergence.

$$\begin{aligned}\langle\rho_i,\rho_j\rangle &= \sum_{n_1=1}^{N}\sum_{n_2=1}^{N} z_{in_1}z_{jn_2}\int_\Omega \phi_{n_1}(\boldsymbol{x})\phi_{n_2}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\\ &= \sum_{n_1=1}^{N}\sum_{n_2=1}^{N} z_{in_1}z_{jn_2}\delta_{n_1n_2} = \boldsymbol{z}_i^\top\boldsymbol{z}_j = \delta_{ij},\end{aligned} \tag{24}$$

because the vectors $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ are columns of $\boldsymbol{Z}$ and form an orthonormal basis.

### B. Proof of Theorem 1

For notational convenience, we omit the index $k$ in this proof and define

$$f^{(N)} \triangleq \hat{\Xi}^{(N)}(\boldsymbol{x}),\quad f^{(M)} \triangleq \hat{\Xi}^{(M)}(\boldsymbol{x}). \tag{25}$$

The squared L2 norm of the distance function

$$\begin{aligned}\|d(\boldsymbol{x})\|^2 &= \int_\Omega \left(f^{(N)} - f^{(M)}\right)\left(f^{(N)} - f^{(M)}\right)\mathrm{d}\boldsymbol{x}\\ &= \underbrace{\int_\Omega f^{(N)}f^{(N)}\mathrm{d}\boldsymbol{x}}_{\text{Term 1}} - 2\underbrace{\int_\Omega f^{(N)}f^{(M)}\mathrm{d}\boldsymbol{x}}_{\text{Term 2}} + \underbrace{\int_\Omega f^{(M)}f^{(M)}\mathrm{d}\boldsymbol{x}}_{\text{Term 3}}.\end{aligned} \tag{26}$$

The integral is decomposed for the three terms and each is considered separately. Following the same argumentation as in (24), term 1 yields

$$\begin{aligned}&\int_\Omega f^{(N)}f^{(N)}\mathrm{d}\boldsymbol{x}\\ &= \int_\Omega \left(\boldsymbol{w}^\top\boldsymbol{\phi}(\boldsymbol{x})\right)\left(\boldsymbol{w}^\top\boldsymbol{\phi}(\boldsymbol{x})\right)\mathrm{d}\boldsymbol{x}\\ &= \sum_{n_1=1}^{N}\sum_{n_2=1}^{N} w_{n_1}w_{n_2}\int_\Omega \phi_{n_1}(\boldsymbol{x})\phi_{n_2}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\\ &= \sum_{n_1=1}^{N}\sum_{n_2=1}^{N} w_{n_1}w_{n_2}\delta_{n_1n_2} = \boldsymbol{w}^\top\boldsymbol{w}.\end{aligned} \tag{27}$$

Similarly, using the argumentation of the proof of Lemma 1, term 3 gives

$$\int_\Omega f^{(M)}f^{(M)}\mathrm{d}\boldsymbol{x} = \boldsymbol{v}^\top\boldsymbol{v}. \tag{28}$$

Term 2

$$\begin{aligned}&\int_\Omega f^{(N)}f^{(M)}\mathrm{d}\boldsymbol{x}\\ &= \int_\Omega \left(\boldsymbol{w}^\top\boldsymbol{\phi}(\boldsymbol{x})\right)\left(\boldsymbol{v}^\top\boldsymbol{\rho}(\boldsymbol{x})\right)\mathrm{d}\boldsymbol{x}\\ &= \sum_{m=1}^{M}\sum_{n=1}^{N} v_m w_n\int_\Omega \phi_n(\boldsymbol{x})\rho_m(\boldsymbol{x})\mathrm{d}\boldsymbol{x},\end{aligned} \tag{29}$$

with $\rho_m(\boldsymbol{x})$ given by (10). Using the definition, we obtain

$$\begin{aligned}&\int_\Omega f^{(N)}f^{(M)}\mathrm{d}\boldsymbol{x}\\ &= \sum_{m=1}^{M}\sum_{n_1=1}^{N} v_m w_{n_1}\int_\Omega \phi_{n_1}(\boldsymbol{x})\sum_{n_2=1}^{N} z_{n_2m}\phi_{n_2}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\\ &= \sum_{m=1}^{M}\sum_{n_1=1}^{N} v_m w_{n_1}\int_\Omega \sum_{n_2=1}^{N} z_{n_2m}\phi_{n_1}(\boldsymbol{x})\phi_{n_2}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\\ &= \sum_{m=1}^{M}\sum_{n_1=1}^{N} v_m w_{n_1}\sum_{n_2=1}^{N} z_{n_2m}\int_\Omega \phi_{n_1}(\boldsymbol{x})\phi_{n_2}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\\ &= \sum_{m=1}^{M}\sum_{n_1=1}^{N} v_m w_{n_1}\sum_{n_2=1}^{N} z_{n_2m}\delta_{n_1n_2}\\ &= \sum_{m=1}^{M}\sum_{n=1}^{N} v_m w_n z_{nm} = \sum_{m=1}^{M} v_m\boldsymbol{z}_m^\top\boldsymbol{w}\end{aligned} \tag{30}$$

where, in the second step, a similar argumentation as in the proof of Lemma 1 is considered for changing the order of the sum and the integral. Plugging the terms back into (26) gives

$$\begin{aligned}\|d(\boldsymbol{x})\|^2 &= \boldsymbol{w}^\top\boldsymbol{w} - 2\sum_{m=1}^{M} v_m\boldsymbol{z}_m^\top\boldsymbol{w} + \boldsymbol{v}^\top\boldsymbol{v}\\ &= \boldsymbol{w}^\top\boldsymbol{w} - 2\sum_{m=1}^{M} v_m\boldsymbol{z}_m^\top\boldsymbol{w} + \sum_{m=1}^{M} v_m\boldsymbol{z}_m^\top\boldsymbol{z}_m v_m\\ &= \left\|\boldsymbol{w} - \sum_{m=1}^{M} v_m\boldsymbol{z}_m\right\|^2.\end{aligned} \tag{31}$$