Prediction with Gaussian Process Dynamical Models

Thomas Beckers and Sandra Hirche

***Abstract*—The modeling and simulation of dynamical systems is a necessary step for many control approaches. Using classical, parameter-based techniques for modeling of modern systems, e.g., soft robotics or human-robot interaction, is often challenging or even infeasible due to the complexity of the system dynamics. In contrast, data-driven approaches need only a minimum of prior knowledge and scale with the complexity of the system. In particular, Gaussian process dynamical models (GPDMs) provide very promising results for the modeling of complex dynamics. However, the control properties of these GP models are just sparsely researched, which leads to a "blackbox" treatment in modeling and control scenarios. In addition, the sampling of GPDMs for prediction purpose respecting their non-parametric nature results in non-Markovian dynamics making the theoretical analysis challenging. In this article, we present the relation of the sampling procedure and non-Markovian dynamics for two types of GPDMs and analyze the control theoretical properties focusing on the transfer between common sampling approximations. The outcomes are illustrated with numerical examples and discussed with respect to the application in control settings.**

***Index Terms*—Probabilistic models, nonparametric methods, Gaussian processes, stochastic modeling, probabilistic simulation, learning systems, data-based control.**

I. Introduction

MODELING of dynamical systems plays a very import role in the area of control theory. The goal is the derivation of a mathematical model which is based on generated input data and the corresponding output data of the plant. The model is necessary for any model-based control design, such as model predictive control. Besides, a model is required for simulations to evaluate the quality of the control designs and to improve the understanding of the system. To achieve a dynamical model, the output of the model is feedbacked to the model itself. A special class of dynamical models is given by *simulation models*, which do not rely on any data from the plant during the prediciton [1]. Therefore, these models are suitable to perform predictions independent of the plant for not only simulations but also in control scenarios such as model predictive control. Classical system identification deals with parametric models. If the system contains nonlinearities, there exist various identification techniques, which mostly depends on the structure of the nonlinear elements. For these approaches, a suitable model structure must be selected a priori to achieve useful results. However, there exists a large class of systems which can not be accurately described by parametric models. Especially, for complex systems such as human motion dynamics [2], [3], prediction of climate effects [4], [5] or structural dynamics [6], [7], non-parametric techniques appear to be more promising.

Within the past two decades, Gaussian processes (GPs) have been developed as powerful function regressors. A GP connects every point of a continuous input space with a normally distributed random variable. Any finite group of those infinitely many random variables follows a multivariate Gaussian distribution. The result is a powerful tool for nonlinear function regression without the need of much prior knowledge [8]. In contrast to most of the other techniques, GP modeling provides not only a mean function but also a measure for the uncertainty of the prediction. The output is a Gaussian distributed variable which is fully described by the mean and the variance. There are several possibilities to use a Gaussian process for dynamic system modeling. A frequent approach is the state space model which is in general a very efficient model structure. Gaussian process dynamical models (GPDMs) have recently also become a versatile tool in system identification because of their beneficial properties such as the bias variance trade-off and the strong connection to Bayesian mathematics, see [9]. The Gaussian process state space model (GP-SSM) uses GPs for modeling dynamical systems with state space models, see [10], where each state is described by an own GP. The function between the states and the system's outputs are modeled by another GP or a parametric structure. Alternatives are given by nonlinear identification models such as NFIR [11], NARX [10] or nonlinear output error (NOE) models [12]. In comparison to the other models, the NOE has the advantage of being a simulation model such as the GP-SSM. Although the application of Gaussian process dynamical models increases in control theory, e.g., for adaptive control and model predictive control [13], [14], [15], the theoretical properties of these GPDMs are only sparsely researched. However, the theoretical properties are crucial for further investigations in robustness, stability and performance of control approaches based on GPDMs [16], [17].

In many works where GPs are considered as dynamical model, only the mean function of the process is employed, for instance in [18] and [19]. This is mainly because a GPDM is often used for replacing a deterministic model in already existing model-based control approaches. In [20] some basic theoretical properties for deterministic GP-SSMs are derived. However, GPDMs contain a much richer description of the underlying dynamics but also the uncertainty about the model itself when the full probabilistic representation is considered. In [21], [22] control laws are derived which explicitly take the uncertainty of GPDMs into account but without investigation of the control properties of the models. In order to ensure the applicability of GPDMs, classical control theory properties are required,

The authors are with the Chair of Information-oriented Control (ITR), Department of Electrical and Computer Engineering, Technical University of Munich, 80333 Munich, Germany. e-mail: t.beckers@tum.de, hirche@tum.de

see [16] and [17]. Such basic properties of a dynamical system are, among others, the existence of boundedness conditions. In [23] some basic boundedness properties for simplified probabilistic GP-SSMs are presented. However, it turns out that the analysis of GPDMs is challenging. as its usage in simulations requires the sampling of an infinite-dimensional object which is not possible without further simplifications, e.g., discrete sampling with interpolation. To overcome this issue, the authors of [24] propose to marginalize out the transition functions to respect the nonparametric nature of the model using Particle Markov Chain Monte Carlo (PMCMC). However, the approach is limited to autonomous, one-dimensional systems and do not consider the control specific properties of the resulting non-Markovian dynamics.

Contributions: The contribution of this article is the analysis of the control theoretical properties of GP-SSMs and GP-NOE models. We show that the sampling of GPDMs for simulation purposes results in non-Markovian dynamics and present how it is related to approximated GP dynamics. We analyze the boundedness and approximation error of these stochastic models, focusing on the relationship between different approximations and the true dynamics. The remainder of the article is structured as follows. In section II, the background of GPs and the relation to GPDMs are introduced. Section III presents the crux of simulation and a sampling approach including approximations. The boundedness of the presented models is analyzed in section IV, followed by a discussion. Finally, a case study demonstrates the applicability.

Notation: Vectors and vector-valued functions are denoted with bold characters $\boldsymbol{v}$. The notation $[\boldsymbol{a};\boldsymbol{b}]$ is used for $[\boldsymbol{a}^\top,\boldsymbol{b}^\top]^\top$ and $\boldsymbol{x}_{1:n}$ denotes $[\boldsymbol{x}_1,\dots,\boldsymbol{x}_n]$. Capital letters A describes matrices. The matrix I is the identity matrix in appropriate dimension. The expression $\mathcal{N}(\mu,\Sigma)$ describes a normal distribution with mean μ and covariance Σ.

II. Preliminaries and Definitions

In this article, we focus on Gaussian process based dynamic models. Thus, we start with a brief introduction to GPs as they are the central part of the model.

A. Gaussian Process Models

Let $(\Omega,\mathcal{F},P)$ be a probability space with the sample space $\Omega=\mathbb{R}^n, n\in\mathbb{N}$, the corresponding σ-algebra $\mathcal{F}$ and the probability measure P. Consider a vector-valued, unknown function $\boldsymbol{y}=\boldsymbol{f}_u(\boldsymbol{z})$ with $\boldsymbol{f}_u\colon\mathbb{R}^n\to\mathbb{R}^m$ and $\boldsymbol{y}\in\mathbb{R}^m$. The measurement $\tilde{\boldsymbol{y}}\in\mathbb{R}^m$ of the function is corrupted by Gaussian noise $\boldsymbol{\eta}\in\mathbb{R}^m$, i.e.,

$$\tilde{\boldsymbol{y}}=\boldsymbol{f}_u(\boldsymbol{z})+\boldsymbol{\eta},\quad \boldsymbol{\eta}\sim\mathcal{N}(\boldsymbol{0},\Sigma_n) \tag{1}$$

with the positive definite matrix $\Sigma_n=\operatorname{diag}(\sigma_1^2,\dots,\sigma_m^2)$. To generate the training data, the function is evaluated at $n_\mathcal{D}$ input values $\{\boldsymbol{z}^{\{j\}}\}_{j=1}^{n_\mathcal{D}}$. Together with the resulting measurements $\{\tilde{\boldsymbol{y}}^{\{j\}}\}_{j=1}^{n_\mathcal{D}}$, the whole training data set is described by $\mathcal{D}=\{X,Y\}$ with the input training matrix $X=[\boldsymbol{z}^{\{1\}},\boldsymbol{z}^{\{2\}},\dots,\boldsymbol{z}^{\{n_\mathcal{D}\}}]\in\mathbb{R}^{n\times n_\mathcal{D}}$ and the output training matrix $Y=[\tilde{\boldsymbol{y}}^{\{1\}},\tilde{\boldsymbol{y}}^{\{2\}},\dots,\tilde{\boldsymbol{y}}^{\{n_\mathcal{D}\}}]^\top\in\mathbb{R}^{n_\mathcal{D}\times m}$. Now, the objective is to predict the output of the function $\boldsymbol{f}_u(\boldsymbol{z}^*)$ at a test input $\boldsymbol{z}^*\in\mathbb{R}^n$. The underlying assumption of GP modeling is, that the data can be represented as a sample of a multivariate Gaussian distribution. The joint distribution of the i-th component of $\boldsymbol{f}_u(\boldsymbol{z}^*)$ is[1]

$$\begin{bmatrix}Y_{:,i}\\ f_{u\,i}(\boldsymbol{z}^*)\end{bmatrix}\sim\mathcal{N}\left(\begin{bmatrix}\boldsymbol{m}_u(X)\\ m_u(\boldsymbol{z}^*)\end{bmatrix},\begin{bmatrix}K(X,X)+\sigma_i I & \boldsymbol{k}(\boldsymbol{z}^*,X)\\ \boldsymbol{k}(\boldsymbol{z}^*,X)^\top & k(\boldsymbol{z}^*,\boldsymbol{z}^*)\end{bmatrix}\right) \tag{2}$$

with the covariance function $k\colon\mathbb{R}^n\times\mathbb{R}^n\to\mathbb{R}$ as a measure of the correlation of two points $(\boldsymbol{x},\boldsymbol{x}')$. The mean function is given by a continuous function $m_u\colon\mathbb{R}^n\to\mathbb{R}$ and the vector of mean functions $\boldsymbol{m}_u\colon\mathbb{R}^{n\times n_\mathcal{D}}\to\mathbb{R}^{n_\mathcal{D}}$ by $\boldsymbol{m}_u(X)=[m_u(X_{:,1});\dots;m_u(X_{:,n_\mathcal{D}})]$. The covariance function is the central part of the kernel trick, which transforms the data to a higher dimensional feature space Υ, see Fig. 1, without knowing the actual transformation $\phi\colon R^{n_\mathcal{D}}\to\Upsilon$ since $k(x,x')=\langle\phi(x),\phi(x')\rangle$. Then, a linear regression is performed in the feature space and the output is transformed back. The function $K\colon\mathbb{R}^{n\times n_\mathcal{D}}\times\mathbb{R}^{n\times n_\mathcal{D}}\to\mathbb{R}^{n_\mathcal{D}\times n_\mathcal{D}}$ is called

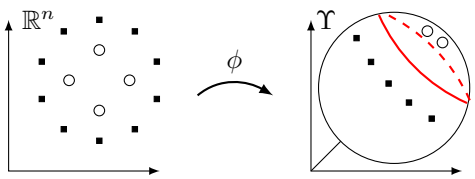


Fig. 1: The kernel trick transforms the data points in a higher dimensional feature space where a linear regression is performed.

the Gram matrix $K_{j,l}=k(X_{:,l},X_{:,j})$ with $j,l\in\{1,\dots,n_\mathcal{D}\}$. Each element of the matrix represents the covariance between two elements of the training data X. The vector-valued function $\boldsymbol{k}\colon\mathbb{R}^n\times\mathbb{R}^{n\times n_\mathcal{D}}\to\mathbb{R}^{n_\mathcal{D}}$ calculates the covariance between the test input $\boldsymbol{z}^*$ and the input training data X

$$\boldsymbol{k}(\boldsymbol{z}^*,X)\text{ with }k_j=k(\boldsymbol{z}^*,X_{:,j}) \tag{3}$$

for all $j\in\{1,\dots,n_\mathcal{D}\}$. The covariance function depends on a set of hyperparameters $\Phi=\{\varphi_1,\dots,\varphi_{n_h}\}$ whose number $n_h\in\mathbb{N}$ and domain of parameters depend on the employed function. A comparison of the characteristics of the different covariance functions can be found in [25]. The prediction of each component of $\boldsymbol{f}_u(\boldsymbol{z}^*)$ is derived from the joint distribution (2) and, therefore, it is a Gaussian distributed variable. The conditional probability distribution for the i-th element of the output is defined by the mean and the variance

$$\begin{aligned}\mu_i(\boldsymbol{f}_u|\boldsymbol{z}^*,\mathcal{D}) =& m_u(\boldsymbol{z}^*)+\boldsymbol{k}(\boldsymbol{z}^*,X)^\top(K+\sigma_i I)^{-1}\\ &(Y_{:,i}-\boldsymbol{m}_u(X))\end{aligned} \tag{4}$$

$$\begin{aligned}\operatorname{var}_i(\boldsymbol{f}_u|\boldsymbol{z}^*,\mathcal{D}) =& k(\boldsymbol{z}^*,\boldsymbol{z}^*)-\boldsymbol{k}(\boldsymbol{z}^*,X)^\top(K+\sigma_i I)^{-1}\\ &\boldsymbol{k}(\boldsymbol{z}^*,X).\end{aligned} \tag{5}$$

Remark 1. *The existence of the inverse Gram matrix is essential for the prediction step. The Gram matrix is invertable if all*

[1]For notational convenience, we simplify $K(X,X)$ to K

vectors in the feature space $\phi(X_{:,1}), \ldots, \phi(X_{:,n_\mathcal{D}})$ are independent. If there exist an $i, j \in \mathbb{N}$ such that $\phi(X_{:,i}) = \phi(X_{:,j})$ or the number of training points exceeds the dimensionality of the feature space, i.e $n_\mathcal{D} > dim(\Upsilon)$, the condition can be violated. In this case, the Moore–Penrose-pseudoinverse is used. For further discussion on regularization methods for GPs, see [26].

The m normally distributed components of $\boldsymbol{f}|\boldsymbol{z}^*, \mathcal{D}$ are combined into a multi-variable Gaussian distribution

$$\begin{aligned} \boldsymbol{f}|\boldsymbol{z}^*, \mathcal{D} &\sim \mathcal{N}(\boldsymbol{\mu}(\cdot), \Sigma(\cdot)) \\ \boldsymbol{\mu}(\boldsymbol{f}|\boldsymbol{z}^*, \mathcal{D}) &= [\mu(f_1|\boldsymbol{z}^*, \mathcal{D}), \ldots, \mu(f_m|\boldsymbol{z}^*, \mathcal{D})]^\top \\ \Sigma(\boldsymbol{f}|\boldsymbol{z}^*, \mathcal{D}) &= \mathrm{diag}(\mathrm{var}(f_1|\boldsymbol{z}^*, \mathcal{D}), \ldots, \mathrm{var}(f_m|\boldsymbol{z}^*, \mathcal{D})), \end{aligned} \tag{6}$$

where $\Phi_i = \{\varphi_1, \ldots, \varphi_{n_h}\}$ is the set of hyperparameters for the i-th output dimension. The hyperparameters are typically optimized by means of the likelihood function, thus by $\varphi_i^{\{j\}} = \arg\max_{\varphi^{\{j\}}} \log P(Y_{:,i}|X, \varphi^{\{j\}})$ for all $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, n_h\}$. A gradient based algorithm is often used to find at least a local maximum of the likelihood function [8].

Remark 2. *Also the correlation between the dimensions of the state variable can be considered, e.g., by placing a separate covariance functions on the GP outputs [27] or by using a multiple-output covariance function [28].*

B. Gaussian Process Dynamical Models

Black-box models of nonlinear systems can be classified in many different ways. One main aspect of GPDMs is to distinguish between recurrent structures and non-recurrent structures. A model is called recurrent if parts of the regression vector depend on the outputs of the model. Even though recurrent models become more complex in terms of their behavior, they allow to model sequences of data, see [29]. If all states are feedbacked from the model itself, we get a simulation model, which is a special case of the recurrent structure. The advantage of such a model is its property to be independent of the real system's states. Thus, it is suitable for simulations, as it allows multi-step ahead predictions. In this article, we focus on two often-used recurrent structures: the Gaussian process state space model (GP-SSM) and the Gaussian process nonlinear error output (GP-NOE) model.

1) Gaussian Process State Space Models: Gaussian process state space models are structured as a discrete-time system. In this case, the states are the regressors, which is visualized in Fig. 2. This approach allows to be more efficient, since the regressors are less restricted in their internal structure. Thus, a very efficient model in terms of number of regressors might be possible. The mapping from the states to the output can often be assumed to be known. The situation, where the output mapping describes a known sensor model, is such an example. It is mentioned in [24] that using too flexible models for both - $\boldsymbol{f}$ and the output mapping - can result in problems of non-identifiability. Therefore, we focus on a known output mapping. The mathematical model of the GP-SSM is thus given by

$$\begin{aligned} \boldsymbol{x}_{t+1} = \boldsymbol{f}(\boldsymbol{\xi}_t) &= \begin{cases} f_1(\boldsymbol{\xi}_t) \sim \mathcal{GP}\left(m^1(\boldsymbol{\xi}_t), k^1(\boldsymbol{\xi}_t, \boldsymbol{\xi}_t')\right) \\ \vdots \qquad \vdots \quad \vdots \\ f_{n_x}(\boldsymbol{\xi}_t) \sim \mathcal{GP}\left(m^{n_x}(\boldsymbol{\xi}_t), k^{n_x}(\boldsymbol{\xi}_t, \boldsymbol{\xi}_t')\right). \end{cases} \\ \boldsymbol{y}_t &\sim p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\gamma}_y), \end{aligned} \tag{7}$$

where $\boldsymbol{\xi}_t \in \mathbb{R}^{n_\xi}, n_\xi = n_x + n_u$ is the concatenation of the state vector $\boldsymbol{x}_t \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$ and the input $\boldsymbol{u}_t \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$ such that $\boldsymbol{\xi}_t = [\boldsymbol{x}_t; \boldsymbol{u}_t]$. The mean function is given by continuous functions $m^1, \ldots, m^{n_x} \colon \mathbb{R}^{n_\xi} \to \mathbb{R}$. The output mapping is parametrized by a known vector $\boldsymbol{\gamma}_y \in \mathbb{R}^{n_\gamma}$ with $n_\gamma \in \mathbb{N}$. The system identification task for the GP-SSM mainly focuses on $\boldsymbol{f}$ in particular. It can be described as finding the state-transition probability conditioned on the observed training data.

Remark 3. *The potentially unknown number of regressors can be determined using established nonlinear identification techniques as presented in [30], or exploiting embedded techniques such as automatic relevance determination [16].*

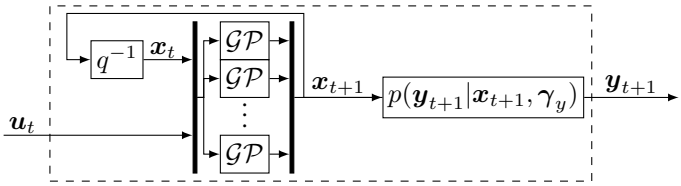


Fig. 2: Structure of a GP-SSM with $\mathfrak{P}$ as backshift operator, such that $\mathfrak{P}^{-1}\boldsymbol{x}_{t+1} = \boldsymbol{x}_t$

2) Gaussian Process Nonlinear Output Error Models: The GP-NOE model uses the past $n_\text{in} \in \mathbb{N}_{>0}$ inputs $\boldsymbol{u}_t \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$ and the past $n_\text{out} \in \mathbb{N}_{>0}$ output values $\boldsymbol{y}_t \in \mathbb{R}^{n_y}$ of the model as the regressors. Figure 3 shows the structure of GP-NOE, where the outputs are feedbacked. In combination with neural networks, this model type is also known as *parallel model*. The mathematical model of the GP-NOE is given by

$$\boldsymbol{y}_{t+1} = \boldsymbol{h}(\boldsymbol{\zeta}_t) = \begin{cases} h_1(\boldsymbol{\zeta}_t) \sim \mathcal{GP}\left(m^1(\boldsymbol{\zeta}_t), k^1(\boldsymbol{\zeta}_t, \boldsymbol{\zeta}_t')\right) \\ \vdots \qquad \vdots \quad \vdots \\ h_{n_y}(\boldsymbol{\zeta}_t) \sim \mathcal{GP}\left(m^{n_y}(\boldsymbol{\zeta}_t), k^{n_y}(\boldsymbol{\zeta}_t, \boldsymbol{\zeta}_t')\right), \end{cases} \tag{8}$$

where the vector $\boldsymbol{\zeta}_t \in \mathbb{R}^{n_\zeta}, n_\zeta = n_\text{out} n_y + n_\text{in} n_u$ is the concatenation of the past output values $\boldsymbol{y}_t$ and input values $\boldsymbol{u}_t$ such that $\boldsymbol{\zeta}_t = [\boldsymbol{y}_{t-n_\text{out}+1}; \ldots; \boldsymbol{y}_t; \boldsymbol{u}_{t-n_\text{in}+1}; \ldots; \boldsymbol{u}_t]$. The mean function is given by continuous functions $m^1, \ldots, m^{n_y} \colon \mathbb{R}^{n_\zeta} \to \mathbb{R}$. In contrast to nonlinear autoregressive exogenous models, that focus on one-step ahead prediction, a NOE model is more suitable for simulations as it considers the multi-step ahead prediciton [1]. However, the drawback is a more complex training procedure that requires a nonlinear optimization scheme due to their recurrent structure [16].

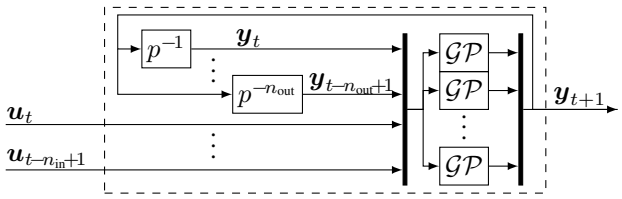


Fig. 3: Structure of a GP-NOE model with $\mathfrak{P}$ as backshift operator, such that $\mathfrak{P}^{-1}\boldsymbol{y}_{t+1} = \boldsymbol{y}_t$

Remark 4. *It is always possible to convert an identified input-output model into a state-space model, see[31]. However, focusing on state-space models only would preclude the development of a large number of useful identification result.*

Remark 5. *As the article focuses on the properties of GPDMs, regardless of the training procedure, we assume for the remainder of the paper that a training set $\mathcal{D}$ is existent and available. For further information on the training procedure, we refer to [10], [12] for GP-NOE models and to [2], [24], [32] for GP-SSMs.*

III. The Crux of Simulation

The prediction with discrete-time GPDMs, needed for simulations and model-based control approaches, is more challenging than GP prediction: The reason is the feedback of the model's output to the input that manifests as correlation between the current and past states defined by the GP model. Therefore, a prediction with the presented GP models in section II-B would require the sampling of the probabilistic mappings $\boldsymbol{f}$ and $\boldsymbol{h}$ given by (7) and (8), respectively. Once sampled, the model could be treated as standard discrete-time system. Unfortunately, these functions are defined on the sets $\mathcal{X} \subseteq \mathbb{R}^{n_x}, \mathcal{U} \subseteq \mathbb{R}^{n_u}$ which contain infinitely many points. Thus, it would be necessary to draw an infinite-dimensional object which represents a sample of the probabilistic mappings. This is not possible without further simplifications, e.g., discrete sampling with interpolation, see [33]. To overcome this issue, we will marginalize out the probabilistic mapping to respect the nonparametric nature of the model. Thus, the result is a joint probability distribution of the states without dependencies on the probabilistic mappings $\boldsymbol{f}, \boldsymbol{h}$. Without lack of generality, the following proposition focuses on an one-dimensional GP-SSM with $\mathcal{X} = \mathcal{U} = \mathbb{R}$. The approach can easily extended to multi-dimensional GP-SSMs due to the fact that the GPs for each state dimension are assumed to be independent (7). The presented approach is analogously applicable for the GP-NOE model.

Remark 6. *For the sake of notational simplicity, we consider GPDMs with identical kernels and noise of the training data for each output dimension. The results can easily be extended to GPDMs with different kernels and noise for each output dimension.*

A. Sampling of GP-SSMs

Proposition 1. *Consider a one-dimensional GP-SSM as in (7) with training set $\mathcal{D} = \{X, Y\}$, where Y is corrupted by Gaussian noise $\mathcal{N}(0, \sigma_n^2)$. For given inputs $\boldsymbol{u}_{0:t} = [u_0, \ldots, u_t] \in \mathbb{R}^{1\times t}$, the joint distribution over the states $\boldsymbol{x}_{0:t} = [x_0, \ldots, x_t] \in \mathbb{R}^{1\times t}$, with $t \in \mathbb{N}$, is given by*

$$p(\boldsymbol{x}_{0:t}|\boldsymbol{u}_{0:t}, \mathcal{D}) = p(x_0)\left|(2\pi)^t \check{K}\right|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\left(\boldsymbol{x}_{1:t} - \check{\boldsymbol{m}}_{0:t-1}\right)\check{K}^{-1}\left(\boldsymbol{x}_{1:t} - \check{\boldsymbol{m}}_{0:t-1}\right)^\top\right), \quad (9)$$

with the Gram matrix

$$\check{K} = K(\boldsymbol{\xi}_{0:t-1}, \boldsymbol{\xi}_{0:t-1}) - K(\boldsymbol{\xi}_{0:t-1}, X)^\top\left(K + \sigma_n^2 I\right)^{-1} K(\boldsymbol{\xi}_{0:t-1}, X), \quad (10)$$

where $\boldsymbol{\xi} = [x; u] \in \mathbb{R}^2$. The elements of the mean vector $\check{\boldsymbol{m}}_{0:t-1} \in \mathbb{R}^{1\times t}$ are given by

$$\check{m}_i = m(\boldsymbol{\xi}_i) + K(\boldsymbol{\xi}_i, X)^\top (K + \sigma_n^2 I)^{-1}(Y - \boldsymbol{m}(X)) \quad (11)$$

for all $i = \{0, \ldots, t-1\}$ with mean vector $\boldsymbol{m}(X) = [m(X_1); \ldots; m(X_{n_\mathcal{D}})]$.

Before we start with the proof, note that even if the probability distribution (9) looks similar to a Gaussian distribution, it is not Gaussian, since the matrix $\check{K}$ depends on the past states $\boldsymbol{x}_{0:t-1}$ by $\boldsymbol{\xi}_{0:t-1}$. In Fig. 4, an example of a joint distribution of a GP-SSM with squared exponential kernel is shown. The initial state x_0 is Gaussian distributed, whereas the joint distribution of $p(x_{0:1})$ is clearly non-Gaussian, since x_0 is passed trough the nonlinear transition function.

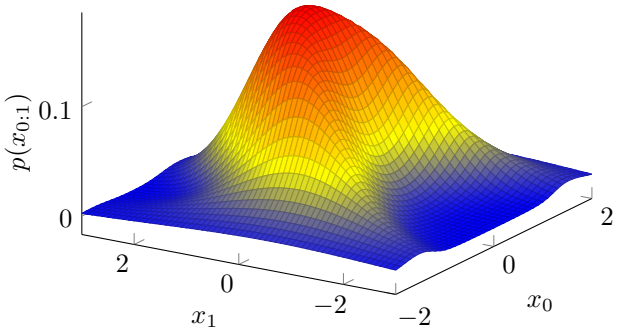


Fig. 4: The joint probability of a GP-SSM is in general non-Gaussian.

Proof. We start with the joint probability distribution of $\boldsymbol{x}_{0:t}$ and the stochastic process $f(\cdot)$ given by

$$p(\boldsymbol{x}_{0:t}, f(\cdot)|\boldsymbol{u}_{0:t}, \mathcal{D}) = p(f(\cdot))p(x_0) \prod_{j=1}^{t} p(x_j|x_{j-1}, u_{t-1}, \mathcal{D}, f(\cdot)). \quad (12)$$

The intuitive goal is to integrate out the stochastic process $f(\cdot)$, such that only the joint probability of the state $\boldsymbol{x}_{0:t}$ remains. Unfortunately, we cannot integrate with respect to a function. For a better understanding, a function can be seen as infinite dimensional vector. Hence, a measure would be needed which is translation invariant and locally finite. The only measure that obeys these two properties is the zero measure, assigning

zero to every input set, and thus, it is not suitable. Instead, we consider the limit

$$p(\boldsymbol{x}_{0:t}|\boldsymbol{u}_{0:t},\mathcal{D}) = \lim_{n_f\to\infty} \int_{\mathbb{R}^{n_f}} p(\boldsymbol{x}_{0:t},\boldsymbol{u}_{0:t},\mathcal{D},f(\cdot))df_1\dots df_{n_f}, \tag{13}$$

where $df_1,\dots,df_{n_f}$ is the function differentiation at different points, following [34, Section 2.3.5]. In probabilistic terms, the relation between x_t and $f(x_{t-1})$ is a Dirac distribution

$$p(x_t|x_{t-1},u_{t-1},\mathcal{D},f(\cdot)) = \delta(x_t - f(x_{t-1})) \tag{14}$$

as for a given state x_{t-1} and a realization $f(x_{t-1})$, there exist only one possible next state x_t. Based on the Dirac distribution, we integrate the stochastic process at every point except $\boldsymbol{x}_{0:t}$, denoted by $\backslash\boldsymbol{x}_{0:t}$, to obtain the joint distribution

$$\begin{aligned} p(\boldsymbol{x}_{0:t}|\boldsymbol{u}_{0:t},\mathcal{D}) = & p(x_0)\lim_{n_f\to\infty}\int_{\mathbb{R}^{n_f}} p(f(\cdot)|\boldsymbol{u}_{0:t},\mathcal{D}) \\ & \prod_{j=1}^{t}\delta(x_j - f(x_{j-1}))df_1\dots df_{n_f\backslash\boldsymbol{x}_{0:t}}. \end{aligned} \tag{15}$$

Given the GP property that f is Gaussian distributed for all x_t, see (2), we obtain

$$\begin{aligned} & p(\boldsymbol{x}_{0:t}|\boldsymbol{u}_{0:t},\mathcal{D}) = p(x_0)\left|(2\pi)^t\check{K}\right|^{-\frac{1}{2}} \\ & \exp\left(-\frac{1}{2}\left(\boldsymbol{x}_{1:t} - \check{\boldsymbol{m}}_{0:t-1}\right)\check{K}^{-1}\left(\boldsymbol{x}_{1:t} - \check{\boldsymbol{m}}_{0:t-1}\right)^\top\right) \end{aligned} \tag{16}$$

with $\check{K}$ and $\check{\boldsymbol{m}}_{0:T-1}$ as in (10) and (11), respectively. □

Remark 7. *In [35], [36], a similar proof is presented but the authors disregard an external input $\boldsymbol{u}_t$ and the conditioning on a training set $\mathcal{D}$. It is our contribution to marginalize out the probabilistic mapping of GP-SSMs with external inputs, such that it is suitable for model-based control approaches.*

The result of Proposition 1, given in (9), suggests how to compute the joint distribution of a GP-SSM without the need of drawing an infinite dimensional sample. For simulation and model-based control scenarios, the conditional probability distribution for the next state ahead $\boldsymbol{x}_{t+1}$ is often required. The next theorem shows that the distribution of $\boldsymbol{x}_{t+1}$ depends on all past states starting with an initial state $\boldsymbol{x}_0 \in \mathbb{R}^n_x$. However, the nature of GP models allows to include the past states as noise-free training data in a way that there exists an analytic closed-form for the prediction.

Theorem 1. *Consider a GP-SSM* (7) *with training set $\mathcal{D} = \{X, Y\}$, where Y is corrupted by Gaussian noise $\mathcal{N}(0,\sigma_n^2 I)$. Then, the conditional distribution of the next state ahead $\boldsymbol{x}_{t+1}\in\mathbb{R}^{n_x}$ and output $\boldsymbol{y}_{t+1}\in\mathbb{R}^{n_y}$ is given by*

$$\begin{aligned} \boldsymbol{x}_{t+1}|\boldsymbol{\xi}_{0:t},\mathcal{D} &\sim \mathcal{N}\big(\boldsymbol{\mu}(\boldsymbol{x}_{t+1}|\boldsymbol{\xi}_{0:t},\mathcal{D}),\Sigma(\boldsymbol{x}_{t+1}|\boldsymbol{\xi}_{0:t},\mathcal{D})\big) \\ \mu_i(\cdot) &= m(\boldsymbol{\xi}_t) + \boldsymbol{k}(\boldsymbol{\xi}_t,\check{X}_t)^\top\check{K}_t^{-1}\big([\check{Y}_t]_{:,i} - \boldsymbol{m}(\check{X}_t)\big) \\ \Sigma_{i,i}(\cdot) &= k(\boldsymbol{\xi}_t,\boldsymbol{\xi}_t) - \boldsymbol{k}(\boldsymbol{\xi}_t,\check{X}_t)^\top\check{K}_t^{-1}\boldsymbol{k}(\boldsymbol{\xi}_t,\check{X}_t) \\ p(\boldsymbol{y}_{t+1}|\boldsymbol{\xi}_{0:t},\boldsymbol{\gamma}_y,\mathcal{D}) &= p(\boldsymbol{y}_{t+1}|\boldsymbol{x}_{t+1},\boldsymbol{\gamma}_y)p(\boldsymbol{x}_{t+1}|\boldsymbol{\xi}_{0:t},\mathcal{D}) \end{aligned} \tag{17}$$

with $\boldsymbol{x}_0\in\mathbb{R}^{n_x}$ for all $t\geq 0$ and extended the data matrices $\check{X}_t\in\mathbb{R}^{n_\xi\times(n_\mathcal{D}+t)}$, $\check{Y}_t\in\mathbb{R}^{(n_\mathcal{D}+t)\times n_x}$

$$\begin{aligned} \check{X}_t &= X, & \check{Y}_t &= Y & \text{if } t=0 \\ \check{X}_t &= [X,\boldsymbol{\xi}_{0:t-1}], & \check{Y}_t &= [Y^\top,\boldsymbol{x}_{1:t}]^\top & \text{otherwise.} \end{aligned} \tag{18}$$

The Gram matrix $\check{K}_t\in\mathbb{R}^{(n_\mathcal{D}+t)\times(n_\mathcal{D}+t)}$ is defined as

$$\check{K}_t = \begin{cases} \begin{bmatrix} K+\sigma_n^2 I & K(\boldsymbol{\xi}_{0:t-1},X) \\ K(\boldsymbol{\xi}_{0:t-1},X)^\top & K(\boldsymbol{\xi}_{0:t-1},\boldsymbol{\xi}_{0:t-1}) \end{bmatrix}, & \text{if } t>0 \\ K+\sigma_n^2 I, & \text{otherwise.} \end{cases} \tag{19}$$

Proof. For the first step, i.e $t=0$, the conditional distribution is identical to the standard GP prediction with predicted mean and variance given by (4). For $t>0$, the current state is feedbacked to the input, as shown in Fig. 2. Following the idea of the previous proof, we condition only on the transitions $\boldsymbol{\xi}_{0:t}$ seen up to that point, instead of conditioning on an infinite-dimensional object. Using (9) with the joint Gaussian distribution property (2) of the GP, we obtain the joint distribution

$$\begin{bmatrix}(\check{Y}_t)_{:,i}\\(x_{t+1})_i\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\boldsymbol{m}(X)\\\boldsymbol{m}(\boldsymbol{\xi}_{0:t})\end{bmatrix},\begin{bmatrix}\check{K}_t & K'_t\\ K'^\top_t & k(\boldsymbol{\xi}_t,\boldsymbol{\xi}_t)\end{bmatrix}\right), \tag{20}$$

where $K'^\top_t = \left[\boldsymbol{k}(\boldsymbol{\xi}_t,X)^\top,\boldsymbol{k}(\boldsymbol{\xi}_t,\boldsymbol{\xi}_{0:t-1})^\top\right]$. Based on (20), the conditional probability distribution of the next state ahead $\boldsymbol{x}_{t+1}$ is computed. □

Remark 8. *The challenge for the subsequent analysis of this approach is that marginalization of $\boldsymbol{f}$ will introduce dependencies across time for the state variable $\boldsymbol{x}_t$ that lead to the loss of the Markovian structure of the state space model.*

Analogously, we derive the prediction for the GP-NOE model.

Theorem 2. *Consider a GP-NOE model* (8) *with training set $\mathcal{D} = \{X, Y\}$, where the output data Y is corrupted by Gaussian noise $\mathcal{N}(0,\sigma_n^2 I)$. Then, the conditional distribution of the next output $\boldsymbol{y}_{t+1}\in\mathbb{R}^{n_y}$ is given by*

$$\begin{aligned} \boldsymbol{y}_{t+1}|\boldsymbol{\zeta}_{0:t},\mathcal{D} &\sim \mathcal{N}\big(\boldsymbol{\mu}(\boldsymbol{y}_{t+1}|\boldsymbol{\zeta}_{0:t},\mathcal{D}),\Sigma(\boldsymbol{y}_{t+1}|\boldsymbol{\zeta}_{0:t},\mathcal{D})\big) \\ \mu_i(\cdot) &= m(\boldsymbol{\zeta}_t) + \boldsymbol{k}(\boldsymbol{\zeta}_t,\check{X}_t)^\top\check{K}_t^{-1}\big([\check{Y}_t]_{:,i} - \boldsymbol{m}(\check{X}_t)\big) \\ \Sigma_{i,i}(\cdot) &= k(\boldsymbol{\zeta}_t,\boldsymbol{\zeta}_t) - \boldsymbol{k}(\boldsymbol{\zeta}_t,\check{X}_t)^\top\check{K}_t^{-1}\boldsymbol{k}(\boldsymbol{\zeta}_t,\check{X}_t) \end{aligned} \tag{21}$$

with $\boldsymbol{\zeta}_0\in\mathbb{R}^{n_\zeta}$ for all $t\geq 0$ and the extended data matrices $\check{X}_t\in\mathbb{R}^{n_\zeta\times(n_\mathcal{D}+t)}$, $\check{Y}_t\in\mathbb{R}^{(n_\mathcal{D}+t)\times n_y}$

$$\begin{aligned} \check{X}_t &= X, & \check{Y}_t &= Y & \text{if } t=0 \\ \check{X}_t &= [X,\boldsymbol{\zeta}_{0:t-1}], & \check{Y}_t &= [Y^\top,\boldsymbol{y}_{1:t}]^\top & \text{otherwise.} \end{aligned} \tag{22}$$

The Gram matrix $\check{K}_t\in\mathbb{R}^{(n_\mathcal{D}+t)\times(n_\mathcal{D}+t)}$ is defined as

$$\check{K}_t = \begin{cases} \begin{bmatrix} K+\sigma_n^2 I & K(\boldsymbol{\zeta}_{0:t-1},X) \\ K(\boldsymbol{\zeta}_{0:t-1},X)^\top & K(\boldsymbol{\zeta}_{0:t-1},\boldsymbol{\zeta}_{0:t-1}) \end{bmatrix}, & \text{if } t>0 \\ K+\sigma_n^2 I, & \text{otherwise.} \end{cases} \tag{23}$$

Proof. The proof is analogous to the proof of Theorem 1. □

B. Approximation Models

The previous section shows that the next step ahead state $\boldsymbol{x}_{t+1}$ of a GP-SSM is a sample drawn from a Gaussian distribution with the posterior mean and variance based on the previous states and inputs. This leads to dependencies between the states such that the dynamical model loses the Markov property, i.e., $\boldsymbol{x}_{t+1}$ depends not only on $\boldsymbol{x}_t$ but on all previous states $\boldsymbol{x}_{0:t}$. Since past states are treated as new training points without noise, the size of the extended training set increases with each time step. This results not only in a fast increasing computing time for the prediction but also an intractable memory problem for long time simulations. To avoid the non-Markovian structure, GPDMs are often approximated for the simulation of dynamical systems [33]. A standard way is to consider only a constant number of past states instead of the full history $\boldsymbol{x}_{0:t}$, see [23]. In the next step, we introduce the formal description of this approximation for GP-SSMs. First, we define the matrix $\Xi_t^m \in \mathbb{R}^{n_\xi \times \underline{m}}$ consisting of past states and inputs as

$$\Xi_t^m := \begin{cases} \emptyset & \text{if } \overline{m} = 0 \vee t = 0 \\ [\boldsymbol{\xi}_{t-1}, \dots, \boldsymbol{\xi}_{\underline{m}}] & \text{otherwise,} \end{cases} \tag{24}$$

which are used for the prediction. The *maximum length of memory* $\overline{m} \in \mathbb{N}$ defines how many past states and inputs are considered for the prediction of the next state. The resulting *actual length of memory* $\underline{m} = \min(t, \overline{m})$ is the number of states and inputs which are actually available. The actual length and the maximum length only differ if the number of past states beginning with $\boldsymbol{x}_0$ is less than $\overline{m}$. The prediction of the next state ahead and the output $\boldsymbol{y}_{t+1} \in \mathbb{R}^{n_y}$ is given by

$$\begin{aligned} \boldsymbol{x}_{t+1}^m &\sim \mathcal{N}\big(\underbrace{\boldsymbol{\mu}(\boldsymbol{x}_{t+1}^m | \boldsymbol{\xi}_t, \Xi_t^m, \mathcal{D})}_{\boldsymbol{f}_t(\boldsymbol{\xi}_t, \Xi_t^m)}, \underbrace{\Sigma(\boldsymbol{x}_{t+1}^m | \boldsymbol{\xi}_t, \Xi_t^m, \mathcal{D})}_{F_t(\boldsymbol{\xi}_t, \Xi_t^m)} \big) \\ \boldsymbol{y}_{t+1} | \boldsymbol{x}_{t+1}^m &\sim p(\boldsymbol{y}_{t+1} | \boldsymbol{x}_{t+1}^m, \boldsymbol{\gamma}_y). \end{aligned} \tag{25}$$

For simplicity in the notation, we introduce helper functions $\boldsymbol{f}_t \colon \mathbb{R}^{n_\xi} \times \mathbb{R}^{n_\xi \times \underline{m}} \to \mathbb{R}^{n_x}$ and $F_t \colon \mathbb{R}^{n_\xi} \times \mathbb{R}^{n_\xi \times \underline{m}} \to \mathbb{R}^{n_x \times n_x}$. The mean and variance of the i-th element of $\boldsymbol{x}_{t+1}^m$ is given by

$$\begin{aligned} [f_t(\boldsymbol{\xi}_t, \Xi_t^m)]_i &= m(\boldsymbol{\xi}_t) + \boldsymbol{k}(\boldsymbol{\xi}_t, \check{X}_t)^\top (\check{K}_t^m)^{-1} \big([\check{Y}_t]_{:,i} - \boldsymbol{m}(\check{X}_t^m) \big) \\ [F_t(\boldsymbol{\xi}_t, \Xi_t^m)]_{i,i} &= k(\boldsymbol{\xi}_t, \boldsymbol{\xi}_t) - \boldsymbol{k}(\boldsymbol{\xi}_t, \check{X}_t)^\top (\check{K}_t^m)^{-1} \boldsymbol{k}(\boldsymbol{\xi}_t, \check{X}_t^m), \end{aligned} \tag{26}$$

respectively. The extended data matrices $\check{X}_t^m \in \mathbb{R}^{n_\xi \times (n_\mathcal{D} + \underline{m})}$ and $\check{Y}_t^m \in \mathbb{R}^{(n_\mathcal{D} + \underline{m}) \times n_y}$ are denoted by

$$\begin{aligned} \check{X}_t^m &= X, & \check{Y}_t^m &= Y & &\text{if } \overline{m} = 0 \vee t = 0 \\ \check{X}_t^m &= [X, \boldsymbol{\xi}_{\underline{t}:t-1}], & \check{Y}_t^m &= [Y^\top, \boldsymbol{x}_{\underline{t}+1:t}]^\top & &\text{otherwise} \end{aligned} \tag{27}$$

with $\underline{t} = \max(0, t - \overline{m})$. The corresponding Gram matrix $\check{K}_t^m \in \mathbb{R}^{(n_\mathcal{D} + \underline{m}) \times (n_\mathcal{D} + \underline{m})}$ is given by

$$\check{K}_t^m = \begin{cases} \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(\boldsymbol{\xi}_{\underline{t}:t-1}, X) \\ K(\boldsymbol{\xi}_{\underline{t}:t-1}, X)^\top & K(\boldsymbol{\xi}_{\underline{t}:t-1}, \boldsymbol{\xi}_{\underline{t}:t-1}) \end{bmatrix} & \text{if } t > 0 \\ & \wedge \overline{m} > 0 \\ K(X,X) + \sigma_n^2 I & \text{otherwise.} \end{cases} \tag{28}$$

Note that the prediction in (25) is based on the past states and inputs back to the time step $\underline{t}$, as indicated in (27). In contrast, the prediction of a GP-SSM is based on the full history of states and inputs, see (27).

Definition 1. *We call* (25) *a* Gaussian process approximated state space model *(GP-ASSM) with maximum memory length* $\overline{m}$*.*

Remark 9. *For* $\overline{m} = \infty$*, the prediction depends on all past states, i.e.,*

$$\boldsymbol{x}_{t+1}^\infty \sim \mathcal{N}\big(\boldsymbol{\mu}(\boldsymbol{x}_{t+1}^\infty | \boldsymbol{\xi}_t, \dots, \boldsymbol{\xi}_0, \mathcal{D}), \Sigma(\boldsymbol{x}_{t+1}^\infty | \boldsymbol{\xi}_t, \dots, \boldsymbol{\xi}_0, \mathcal{D}) \big) \tag{29}$$

and thus, equals the true distribution in (17) *without Markovian property. The most simple approximation is given for maximum memory length* $\overline{m} = 0$

$$\boldsymbol{x}_{t+1}^0 \sim \mathcal{N}\big(\boldsymbol{\mu}(\boldsymbol{x}_{t+1}^0 | \boldsymbol{\xi}_t, \mathcal{D}), \Sigma(\boldsymbol{x}_{t+1}^0 | \boldsymbol{\xi}_t, \mathcal{D}) \big), \tag{30}$$

where the next state ahead is independent of all past states except the current state and input $\boldsymbol{\xi}_t$*. GP-ASSMs with finite maximum length of memory* $\overline{m}$ *are* Markov chains of finite order *as they depend on a finite set of past states and input.*

Figure 5 visualizes the relation between actual length $\underline{m}$, the maximum length $\overline{m}$ and the time step $\underline{t}$ of the last state in the memory.

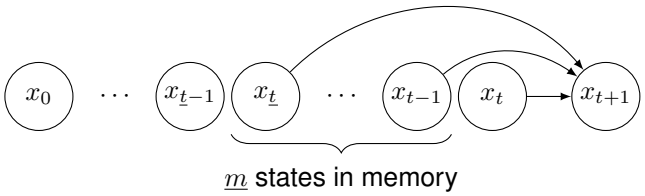


Fig. 5: Time dependencies for the next step ahead state $\boldsymbol{x}_{t+1}$ with the actual length of the memory $\underline{m} = \min(t, m)$ and the last state $\boldsymbol{x}_{\underline{t}}$ in the memory.

Example 1. The idea of the presented approximation is visualized in the top plot of Fig. 6 by a one-dimensional GP-ASSM with maximum length of memory $\overline{m} = 0$. For the sake of simplicity, the external input is set to zero $u_t = 0$ for all $t \in \mathbb{N}$. The distribution of the next state ahead depends only on the current state x_t^0 as it is always sampled from a Gaussian distribution disregarding the history of the past states. Thus, for a given x_0^0, the next state x_1^0 (red circle) is sampled from a Gaussian distribution (green line), where the mean and variance are based on x_0^0, see (30). In the next time step, x_2^0 (red circle) is sampled from a Gaussian distribution (green line), where the mean and variance are solely based on x_1^0. This procedure is continued for the following time steps. As the distribution (green line) of the next state x_{t+1}^0 is independent of the past states $x_{t-1}^0, \dots, x_0^0$, it is always equal to the distribution of the GP (mean and 2-sigma uncertainty) at state x_t^0.

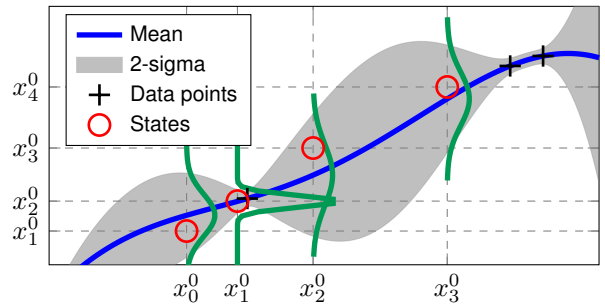


Fig. 6: Sampling of a one-dimensional GP-ASSM with squared exponential kernel.

In contrast, the true sampling ($\overline{m} = \infty$) with a one-dimensional GP-SSM considers all past states $x_t^\infty, \dots, x_0^\infty$, see (29). In Fig. 7, we start again with a given x_0^∞. The next state x_1^∞ (red circle) is sampled from a Gaussian distribution based on the initial state. Then, x_2^∞ is sampled based on x_1^∞ and x_0^∞. For this purpose, the pair (x_0^∞, x_1^∞) is added as noise free training data, see (18). Thus, for any following state where $x_t^\infty = x_0^\infty, t \geq 2$, the next state is given by $x_{t+1}^\infty = x_1^\infty$. Due to the dependency on all past states, the distribution of states, which are not yet added as training data, differ from the predicted mean and variance of the GP. This is visualized at the distribution of x_4^∞ (green line) in contrast to the mean (blue line) and the 2-sigma uncertainty (gray shaded area) of the GP.
This sampling procedure is necessary since the state mapping f, given by (7), can not be drawn directly due to the definition over an infinite set $\mathcal{X} \subseteq \mathbb{R}^{n_x}$. In Fig. 7, the mapping f is illustratively drawn (yellow line) over a finite but large number of states.

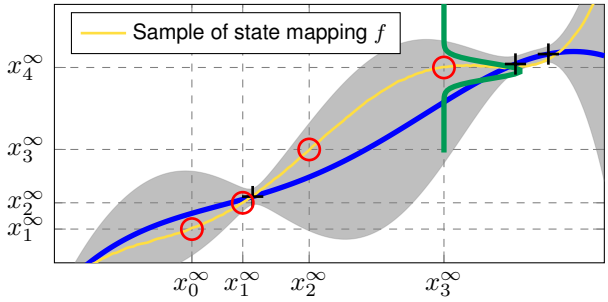


Fig. 7: Sampling of a one-dimensional GP-SSM with squared exponential kernel.

Remark 10. *More advanced strategies for defining the subset of states, that are stored in the memory, are also conceivable. Namely, the same methods as for sparsification of the training data can be exploited. For instance, approaches based on the effective prior [37] or pseudo-inputs [38] have already been successfully applied for sparsification. In this case, the memory Ξ_t^m of the GP-ASSM would not contain the last $\underline{m}$ states but a selected subset of states based on the sparsification algorithm used.*

In the following, we transfer this formal description to GP-NOE models. In comparison to GP-SSMs, the GP-NOE models do not have explicitly defined states. Therefore, we define the matrix of past outputs and inputs as

$$\Lambda_t^m = \begin{cases} \emptyset & \text{if } \overline{m} = 0 \vee k = 0 \\ [\boldsymbol{\zeta}_{t-1}, \dots, \boldsymbol{\zeta}_{\underline{m}}] & \text{otherwise} \end{cases} \tag{31}$$

with $\overline{m} \in \mathbb{N}$ defining the maximum length of memory and $\underline{m} = \min(t, m)$, the actual length of memory. The prediction of the next step ahead output $\boldsymbol{y}_{t+1} \in \mathbb{R}^{n_y}$ is given by

$$\boldsymbol{y}_{t+1}^m \sim \mathcal{N}\big(\underbrace{\boldsymbol{\mu}(\boldsymbol{y}_{t+1}^m | \boldsymbol{\zeta}_t, \Lambda_t^m, \mathcal{D})}_{\boldsymbol{h}_t(\boldsymbol{\zeta}_t, \Lambda_t^m)}, \underbrace{\Sigma(\boldsymbol{y}_{t+1}^m | \boldsymbol{\zeta}_t, \Lambda_t^m, \mathcal{D})}_{H_t(\boldsymbol{\zeta}_t, \Lambda_t^m)} \big). \tag{32}$$

For simplicity in the notation, we introduce the helper functions $\boldsymbol{h}_t \colon \mathbb{R}^{n_\zeta} \times \mathbb{R}^{n_\zeta \times \underline{m}} \to \mathbb{R}^{n_y}$ and $H_t \colon \mathbb{R}^{n_\zeta} \times \mathbb{R}^{n_\zeta \times \underline{m}} \to \mathbb{R}^{n_y \times n_y}$. The mean $[h_t]_i$ and variance $[H_t]_{i,i}$ of the i-th output dimension is given by

$$\begin{aligned} [h_t]_i &= m(\boldsymbol{\zeta}_t) + \boldsymbol{k}(\boldsymbol{\zeta}_t, \check{X}_t)^\top (\check{K}_t^m)^{-1} ([\check{Y}_t]_{:,i} - \boldsymbol{m}(\check{X}_t^m)) \\ [H_t]_{i,i} &= k(\boldsymbol{\zeta}_t, \boldsymbol{\zeta}_t) - \boldsymbol{k}(\boldsymbol{\zeta}_t, \check{X}_t)^\top (\check{K}_t^m)^{-1} \boldsymbol{k}(\boldsymbol{\zeta}_t, \check{X}_t^m). \end{aligned} \tag{33}$$

For GP-NOE models, we define the extended training sets $\check{X}_t^m \in \mathbb{R}^{n_\zeta \times (n_\mathcal{D} + \underline{m})}, \check{Y}_t^m \in \mathbb{R}^{(n_\mathcal{D} + \underline{m}) \times n_y}$ as

$$\begin{aligned} \check{X}_t^m &= X, & \check{Y}_t^m &= Y & &\text{if } \overline{m} = 0 \vee t = 0 \\ \check{X}_t^m &= [X, \boldsymbol{\zeta}_{\underline{t}:t-1}], & \check{Y}_t^m &= [Y^\top, \boldsymbol{y}_{\underline{t}+1:t}]^\top & &\text{otherwise} \end{aligned} \tag{34}$$

with $\underline{t} = \max(0, t - \overline{m})$ and the corresponding Gram matrix $\check{K}_t^m \in \mathbb{R}^{(n_\mathcal{D} + \underline{m}) \times (n_\mathcal{D} + \underline{m})}$ as

$$\check{K}_t^m = \begin{cases} \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(\boldsymbol{\zeta}_{\underline{t}:t-1}, X) \\ K(\boldsymbol{\zeta}_{\underline{t}:t-1}, X)^\top & K(\boldsymbol{\zeta}_{\underline{t}:t-1}, \boldsymbol{\zeta}_{\underline{t}:t-1}) \end{bmatrix} & \text{if } t > 0 \\ & \wedge \overline{m} > 0 \\ K(X,X) + \sigma_n^2 I & \text{otherwise.} \end{cases} \tag{35}$$

Definition 2. *We call* (32) *a* Gaussian process approximated nonlinear output error *(GP-ANOE) model with maximum memory length $\overline{m}$.*

Having introduced the formal description for the approximations of the non-Markovian dynamics, we analyze the approximation error in the following.

C. Approximation Error

In this section, we present the computation of the error between the true state distribution $\boldsymbol{x}_{t+1}$ given by (17) and the approximated distribution $\boldsymbol{x}_{t+1}^m$ based on the maximum length of memory $\overline{m}$. As the Kullback-Leibler (KL) divergence is an important measure of how one probability distribution differs from a second, we start with the KL divergence of the GP-SSM prediction from the GP-ASSM prediction. For the sake of clarity, we define the following notational simplifications

$$F_t^m \coloneqq F_t(\boldsymbol{\xi}_t, \Xi_t^m), \qquad F_t^\infty \coloneqq F_t(\boldsymbol{\xi}_t, \Xi_t^\infty) \tag{36}$$

$$\boldsymbol{f}_t^m \coloneqq \boldsymbol{f}_t(\boldsymbol{\xi}_t, \Xi_t^m), \qquad \boldsymbol{f}_t^\infty \coloneqq \boldsymbol{f}_t(\boldsymbol{\xi}_t, \Xi_t^\infty) \tag{37}$$

for the mean and variance given by (25).

Proposition 2. *Consider a GP-ASSM with maximum length of memory $\overline{m} \in \mathbb{N}$ and data set $\mathcal{D}$ such that*

$$\boldsymbol{x}_{t+1}^m \sim \mathcal{N}\big(\boldsymbol{\mu}(\boldsymbol{x}_{t+1} | \boldsymbol{\xi}_t, \Xi_t^m, \mathcal{D}), \Sigma(\boldsymbol{x}_{t+1} | \boldsymbol{\xi}_t, \Xi_t^m, \mathcal{D}) \big)$$

with $\boldsymbol{x}_0 \in \mathbb{R}^{n_x}$. For given past states and inputs $\boldsymbol{\xi}_{0:t}$, where $\boldsymbol{\xi}_t \neq \boldsymbol{\xi}_0, \dots, \boldsymbol{\xi}_{t-1}$, the KL-divergence of the true distribution $\boldsymbol{x}_{t+1}$ from the approximation $\boldsymbol{x}_{t+1}^m$ is given by

$$d_{\mathrm{KL}}(\boldsymbol{x}_{t+1} \| \boldsymbol{x}_{t+1}^m) = \frac{1}{2} \Delta_t^\top [F_t^m]^{-1} \Delta_t - n_x + \operatorname{tr}\left(F_t^\infty [F_t^m]^{-1}\right) + \ln\left[\operatorname{tr}\left([F_t^\infty]^{-1} F_t^m\right)\right] \tag{38}$$

with $\Delta_t = \boldsymbol{f}_t^m - \boldsymbol{f}_t^\infty$.

Proof. For given past states and inputs $\boldsymbol{\xi}_{0:t}$, the next state $\boldsymbol{x}_{t+1}$ of the GP-SSM and the next state $\boldsymbol{x}_{t+1}^m$ of the GP-ASSM are Gaussian distributed such that the KL-divergence is given by

$$d_{\mathrm{KL}}(\boldsymbol{x}_{t+1} \| \boldsymbol{x}_{t+1}^m) = \frac{1}{2}\Big[\operatorname{tr}\left([F_t^m]^{-1} F_t^\infty\right) + \left(\boldsymbol{f}_t^m - \boldsymbol{f}_t^\infty\right)^\top [F_t^m]^{-1} \left(\boldsymbol{f}_t^m - \boldsymbol{f}_t^\infty\right) - n_x + \ln\left(\frac{|F_t^m|}{|F_t^\infty|}\right)\Big] \tag{39}$$

using the definition of $F_t, \boldsymbol{f}_t$ in (25). As the variance of each element in $\boldsymbol{x}_{t+1}$ and $\boldsymbol{x}_{t+1}^m$ is independent, see (26), the KL-divergence can be rewritten to

$$d_{\mathrm{KL}}(\boldsymbol{x}_{t+1} \| \boldsymbol{x}_{t+1}^m) = \frac{1}{2} \sum_{i=1}^{n_x} \left[\frac{[F_t^\infty]_{i,i} + ([f_t^m]_i - [f_t^\infty]_i)^2}{[F_t^m]_{i,i}} + \ln\left(\frac{[F_t^m]_{i,i}}{[F_t^\infty]_{i,i}}\right) - 1\right]. \tag{40}$$

Finally, simplifying (40) leads to (38). □

Proposition 2 shows that the error is quantified by the drift of mean $\boldsymbol{\mu}(\boldsymbol{x}_{t+1}|\boldsymbol{\xi}_t, \Xi_t^m, \mathcal{D})$ and variance $\Sigma(\boldsymbol{x}_{t+1}|\boldsymbol{\xi}_t, \Xi_t^m, \mathcal{D})$ with respect to the true distribution. Therefore, depending on the maximum length of memory $\overline{m}$, the approximation error is zero at the beginning as the following corollary points out.

Corollary 1. *For all $t \leq \overline{m}$, the approximated distribution $p(\boldsymbol{x}_{t+1}|\boldsymbol{\xi}_t, \Xi_t^m, \mathcal{D})$ given by* (25) *equals the true distribution given by* (7) *with KL-divergence $d_{\mathrm{KL}}(\boldsymbol{x}_{t+1} \| \boldsymbol{x}_{t+1}^m) = 0$.*

Proof. The corollary is a direct consequence of Proposition 2. If the time step t is equal to or less than the maximum length of memory $\overline{m}$, the matrices of past states and inputs of the GP-SSM and the GP-ASSM is identical, i.e., $\Xi_t^m = \Xi_t^\infty$, and thus, the mean and variance of the approximated distribution equals the true distribution. In consequence, the KL-divergence is zero given by (40). □

The restriction of Proposition 2 that the current state must not be part of the past states is necessary as otherwise, the variance $F_t(\boldsymbol{\xi}_t, \Xi_t^\infty)$ or F_t^m would be zero. In Example 1, this case is explained as the past states and inputs are added to the extended data set such that the predicted variance becomes zero. Additionally, the asymmetry of the KL divergence might be obstructive in some applications. Therefore, we introduce a different measure for the approximation error, namely the mean square prediction error (MSPE).

Proposition 3. *Consider a GP-ASSM with maximum memory length $\overline{m} \in \mathbb{N}$ and data set $\mathcal{D}$ such that*

$$\boldsymbol{x}_{t+1}^m \sim \mathcal{N}\big(\boldsymbol{\mu}(\boldsymbol{x}_{t+1}|\boldsymbol{\xi}_t, \Xi_t^m, \mathcal{D}), \Sigma(\boldsymbol{x}_{t+1}|\boldsymbol{\xi}_t, \Xi_t^m, \mathcal{D})\big)$$

with $\boldsymbol{x}_0 \in \mathbb{R}^{n_x}$. For given past states and inputs $\boldsymbol{\xi}_{0:t}$, the MSPE between $\boldsymbol{x}_{t+1}^m$ and $\boldsymbol{x}_{t+1}$ of the GP-SSM is given by

$$\mathrm{E}\left[\left\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_{t+1}^m\right\|^2\right] = \|\boldsymbol{f}_t^\infty - \boldsymbol{f}_t^m\| + \operatorname{tr}\left(F_t^\infty + F_t^m\right). \tag{41}$$

Proof. Since each element of $\boldsymbol{x}_{t+1}$ and $\boldsymbol{x}_{t+1}^m$ with a given history of past states and inputs $\boldsymbol{\xi}_{0:t}$ is Gaussian distributed, the MSPE is defined by

$$\mathrm{E}\left[\left\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_{t+1}^m\right\|^2\right] = \sum_{i=1}^{n_x} \mathrm{E}\left[(x_{t+1,i} - x_{t+1,i}^m)^2\right] \tag{42}$$
$$= \sum_{i=1}^{n_x} \left([f_t^\infty]_i - [f_t^m]_i\right)^2 + [F_t^\infty]_{i,i} + [F_t^m]_{i,i}.$$

Equation (42) is then rewritten to (41). □

With Propositions 2 and 3 the error of the approximation can be computed. Even if the error measures do not decrease in general for increasing maximum length of memory $\overline{m}$, the behavior of the variance can be quantified. The next proposition allows to overestimate the predicted variance based on the maximum length of memory.

Proposition 4. *Consider two GP-ASSMs with states and inputs $\boldsymbol{\xi}_{0:t} \in \mathbb{R}^{n_\xi}$ with $\boldsymbol{\xi}_0 \neq \boldsymbol{\xi}_1 \neq \ldots \neq \boldsymbol{\xi}_t$ such that*

$$\begin{aligned} \boldsymbol{x}_{t+1}^m &\sim \mathcal{N}\big(\boldsymbol{f}_t(\boldsymbol{\xi}_t, \Xi_t^m), F_t(\boldsymbol{\xi}_t, \Xi_t^m)\big) \\ \boldsymbol{x}_{t+1}^{m'} &\sim \mathcal{N}\big(\boldsymbol{f}_t(\boldsymbol{\xi}_t, \Xi_t^{m'}), F_t(\boldsymbol{\xi}_t, \Xi_t^{m'})\big), \end{aligned} \tag{43}$$

where $\overline{m}$ and $\overline{m}'$ are the maximum length of memory, respectively. Then, for $\overline{m}' > \overline{m}$

$$\operatorname{tr}\left(\Sigma(\boldsymbol{x}_{t+1}^{m'}|\boldsymbol{\xi}_t, \Xi_t^{m'}, \mathcal{D})\right) < \operatorname{tr}\left(\Sigma(\boldsymbol{x}_{t+1}^m|\boldsymbol{\xi}_t, \Xi_t^m, \mathcal{D})\right) \tag{44}$$

holds for all $t \in \mathbb{N}$ with $t > \overline{m}$.

Proof. Following (17), the variance for each component of the predicted state of a GP-ASSM is given by

$$\operatorname{var}(x_{t+1,i}^m|\boldsymbol{\xi}_t, \Xi_t^m, \mathcal{D}) = k(\boldsymbol{\xi}_t, \boldsymbol{\xi}_t) - \boldsymbol{k}(\boldsymbol{\xi}_t, \check{X}_t^m)^\top (\check{K}_t^m)^{-1} \boldsymbol{k}(\boldsymbol{\xi}_t, \check{X}_t^m). \tag{45}$$

The Gram matrix $\check{K}_t^m$ is positive definite and from (28) we know, that its dimension is $(n_\mathcal{D} + \underline{m}) \times (n_\mathcal{D} + \underline{m})$. Based on $\check{K}_t^m$, the Gram matrix $\check{K}_t^{m'} \in \mathbb{R}^{(n_\mathcal{D}+\underline{m}')\times(n_\mathcal{D}+\underline{m}')}$ is determined as

$$\check{K}_t^{m'} = \begin{bmatrix} K(\boldsymbol{\xi}_{\underline{t}':\underline{t}-1}, \boldsymbol{\xi}_{\underline{t}':\underline{t}-1}) & K(\boldsymbol{\xi}_{\underline{t}':\underline{t}-1}, X) \\ K(\boldsymbol{\xi}_{\underline{t}':\underline{t}-1}, X)^\top & \check{K}_t^m, \end{bmatrix} \tag{46}$$

where $\underline{t} = \max(0, t - \overline{m})$ and $\underline{t}' = \max(0, t - \overline{m}')$. Since the $\check{K}_t^{m'}$ is also positive definite and $\overline{m}' > \overline{m}$, the inequality

$$\begin{aligned} &k(\boldsymbol{\xi}_t, \boldsymbol{\xi}_t) - \boldsymbol{k}(\boldsymbol{\xi}_t, \check{X}_t^{m'})^\top (\check{K}_t^{m'})^{-1} \boldsymbol{k}(\boldsymbol{\xi}_t, \check{X}_t^{m'}) \\ <&k(\boldsymbol{\xi}_t, \boldsymbol{\xi}_t) - \boldsymbol{k}(\boldsymbol{\xi}_t, \check{X}_t^m)^\top (\check{K}_t^m)^{-1} \boldsymbol{k}(\boldsymbol{\xi}_t, \check{X}_t^m) \\ \Rightarrow &\operatorname{var}(x_{t+1,i}^{m'}|\boldsymbol{\xi}_t, \Xi_t^{m'}, \mathcal{D}) < \operatorname{var}(x_{t+1,i}^m|\boldsymbol{\xi}_t, \Xi_t^m, \mathcal{D}) \end{aligned} \tag{47}$$

holds for all $t \in \mathbb{N}$ with $t > \overline{m}$. Summing up (47) over all elements of $\boldsymbol{x}_{t+1}$ leads to (44). □

Proposition 4 verifies that the variance of the distribution for the next state ahead $\boldsymbol{x}_{t+1}^{m'}$ is less than the variance of $\boldsymbol{x}_{t+1}^m$ with a shorter actual length of memory. This induces that the

variance is the lowest for the true sampling as it is given for $\overline{m} = \infty$. The restriction $t > m$ in Proposition 4 is necessary as otherwise the variances would be equal for $t \leq \overline{m}$ as explained in Corollary 1. The inequality of past states is necessary to ensure that the GP-ASSM with maximum length of memory $\overline{m}'$ contains not only a multiple of the same states which would not decrease the variance. For the sake of completeness, a weaker description for all $t \in \mathbb{N}$ is provided by the following corollary.

Corollary 2. *Consider two GP-ASSMs with states and inputs $\boldsymbol{\xi}_{0:t} \in \mathbb{R}^{n_\xi}$ such that*

$$\begin{aligned} \boldsymbol{x}_{t+1}^{m} &\sim \mathcal{N}\big(\boldsymbol{f}_t(\boldsymbol{\xi}_t, \Xi_t^{m}), F_t(\boldsymbol{\xi}_t, \Xi_t^{m})\big) \\ \boldsymbol{x}_{t+1}^{m'} &\sim \mathcal{N}\big(\boldsymbol{f}_t(\boldsymbol{\xi}_t, \Xi_t^{m'}), F_t(\boldsymbol{\xi}_t, \Xi_t^{m'})\big) \end{aligned} \tag{48}$$

where $\overline{m}$ and $\overline{m}'$ are the maximum length of memory, respectively. Then, for $\overline{m}' > \overline{m}$

$$\operatorname{tr}\left[\Sigma(\boldsymbol{x}_{t+1}^{m'}|\boldsymbol{\xi}_t, \Xi_t^{m'}, \mathcal{D})\right] \leq \operatorname{tr}\left[\Sigma(\boldsymbol{x}_{t+1}^{m}|\boldsymbol{\xi}_t, \Xi_t^{m}, \mathcal{D})\right] \tag{49}$$

holds for all $t \in \mathbb{N}$.

Proof. The corollary is a direct consequence of Proposition 4 since as long as the current time step t is less than the maximum length of memory $\overline{m}$, the variance of $\boldsymbol{x}_{t+1}^{m}$ and $\boldsymbol{x}_{t+1}^{m'}$ is identical as shown in Corollary 1. □

In the next example, a comparison of the presented error measures and the behavior of the variance is presented.

Example 2. In Fig. 8, the distributions (gray shaded) for the next state ahead x_{t+1}^{m} depending on the maximum length of memory $\overline{m}$ for a given trajectory $x_0, \ldots, x_3$ (red circles) is shown. We use here a one-dimensional GP-ASSM with squared exponential function. For sake of simplicity, the input is set to zero, i.e., $u_t = 0$ for all $t \in \mathbb{N}$. With increasing maximum length of memory $\overline{m}$, the variance of the distributions (gray shaded) decreases as stated in Proposition 4. For $\overline{m} = 3$, the distribution is equal to the true distribution as stated in Corollary 1. Table I shows the computed KL-divergence, the MSPE and the variance of x_4^m per maximum length of memory $\overline{m}$.

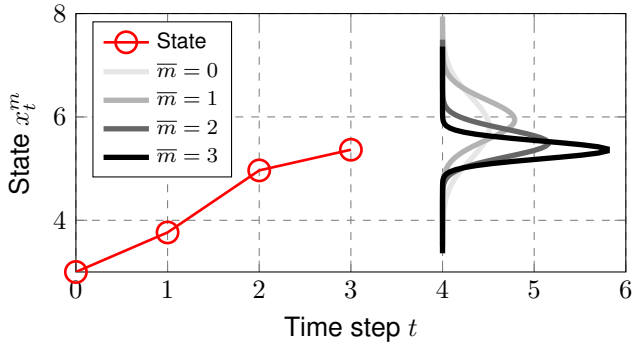


Fig. 8: The distribution for the next state ahead x_t^m depending on the maximum length of memory $\overline{m}$.

	$\overline{m} = 0$	$\overline{m} = 1$	$\overline{m} = 2$	$\overline{m} = 3$
$d_{\mathrm{KL}}(x_4 \| x_4^m)$	2.1131	3.0811	0.5559	0
$\mathrm{MSPE}(x_4, x_4^m)$	0.5720	0.5190	0.1171	0.0575
$\Sigma(x_4^m \| \boldsymbol{\xi}_3, \Xi_3^m, \mathcal{D})$	0.3620	0.1519	0.0706	0.0288

TABLE I: Comparison of the KL-divergence, MSPE and variance Σ for GP-ASSMs with different maximum lengths of memory $\overline{m}$.

So far, we obtain a method for sampling from the non-Markovian GP-SSM and introduce the approximated GP-ASSM which is a Markov chain of finite order. This approximation allows to use GP-ASSMs like parametric dynamical models since the state dependencies across time are removed. The approximation error is analyzed based on different measures and illustrated in Example 2.

Remark 11. *This section focuses on the formal development of GP-ASSMs, but the results are also directly applicable to GP-ANOE models. In this case, the proofs are analogously but with the output $\boldsymbol{y}_t$ as regressor.*

IV. Boundedness of GPDMs

After the introduction of GP-SSMs and GP-ASSMs, the models are analyzed in terms of boundedness. Furthermore, the relation of the boundedness properties between the true and the approximated distribution are investigated.

A. GP State Space Models

We start with the general introduction of the boundedness of GP-ASSMs for bounded mean functions and kernels.

Theorem 3. *Consider a GP-ASSM* (25) *with maximum memory length $\overline{m}$ and bounded mean function and kernel, i.e., $\|\boldsymbol{m}(\boldsymbol{x})\| < \infty, k(\boldsymbol{x}, \boldsymbol{x}') < \infty$, respectively $\forall \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^{n_\xi}$. Then, the expected value of the sequence $\{\boldsymbol{x}_t^m\}, t \in \mathbb{N}$ defined in* (25) *is ultimately* p-bounded *by*

$$\sup_{t \in \mathbb{N}} \mathrm{E}\,\|\boldsymbol{x}_t^m\|^p < \infty \tag{50}$$

for all $\boldsymbol{x}_0 \in \mathbb{R}^{n_x}, \forall p \in \mathbb{N}$.

Proof. We start with the computation of the expected value for a one-dimensional GP-SSM, which equals a GP-ASSM with $\overline{m} = \infty$, as for any other $\overline{m}$ the number of considered past states is reduced. For this purpose, we integrate over the joint probability distribution given by (9). That leads to

$$\sup_{t \in \mathbb{N}} \mathrm{E}[x_t^p] = \sup_{t \in \mathbb{N}} \int_{\mathbb{R}^t} x_t^p p(\boldsymbol{x}_{0:t}|\boldsymbol{u}_{0:t}, \mathcal{D}) d\boldsymbol{x}_{0:t} \tag{51}$$

$$= c_1 \sup_{t \in \mathbb{N}} \int_{\mathbb{R}^t} x_t^p \exp\left(-\frac{1}{2}\, \boldsymbol{\mu}_t^\top\, \check{K}^{-1}\, \boldsymbol{\mu}_t\right) d\boldsymbol{x}_{0:t} \tag{52}$$

with $\boldsymbol{\mu}_t = \boldsymbol{x}_{1:t} - \check{\boldsymbol{m}}_{0:t-1}$, constant $c_1 \in \mathbb{R}$ and $\check{\boldsymbol{m}}_{0:t-1}, \check{K}$ given by (10) and (11). The matrix $\check{K}$ is positive definite, symmetric and the eigenvalues are positive and bounded, i.e., $\underline{\lambda}(\check{K}) > 0, \bar{\lambda}(\check{K}) < \infty$ for all $\boldsymbol{\xi}_{0:t-1} \in \mathbb{R}^{n_\xi \times t}$. The expression $\underline{\lambda}(\cdot)$ and $\bar{\lambda}(\cdot)$ describe the minimum and maximum

eigenvalue of a matrix, respectively. Thus, there exists a positive definite and symmetric matrix $\overline{K} \in \mathbb{R}^{t\times t}$, such that we can upper bound the expected value to

$$\sup_{t\in\mathbb{N}} \mathrm{E}[x_t^p] \leq c_1 \sup_{t\in\mathbb{N}} \int_{\mathbb{R}^t} x_t^p \exp\left(-\frac{1}{2}\,\boldsymbol{\mu}_t^\top\,\overline{K}\,\boldsymbol{\mu}_t\right) d\boldsymbol{x}_{0:t}. \quad (53)$$

As $\check{\boldsymbol{m}}_{0:t-1}$ is given by

$$\begin{aligned}\check{\boldsymbol{m}}_{0:t-1} =& \boldsymbol{m}(\boldsymbol{\xi}_{0:t-1}) + K(\boldsymbol{\xi}_{0:t-1}, X)^\top (K(X,X) + \sigma_n^2 I)^{-1} \\ & (Y - \boldsymbol{m}(X)), \end{aligned} \quad (54)$$

following (11), and the mean function $\boldsymbol{m}$ and kernel k are bounded, it follows that each element of $\boldsymbol{m}_{0:t-1}$ is bounded, i.e., $m_i \leq c_2 \in \mathbb{R}_{>0}, \forall i = 0, \ldots, t-1$. Note, that this bound only depends on the previous time steps and is independent of time step t. Consequently, there exists a vector $\boldsymbol{c}_t \in \mathbb{R}^t$ such that (53) can be rewritten as

$$\begin{aligned}&\int_{\mathbb{R}^t} x_t^p \exp\left(-\frac{1}{2}\,\boldsymbol{\mu}_t^\top\,\overline{K}\,\boldsymbol{\mu}_t\right) d\boldsymbol{x}_{0:t} \\ \leq& \int_{\mathbb{R}^t} x_t^p \exp\left(-\frac{1}{2}\boldsymbol{c}_t^\top \overline{K} \boldsymbol{c}_t\right) d\boldsymbol{x}_{0:t} \\ \leq& c_3 \in \mathbb{R}_{>0} \end{aligned} \quad (55)$$

and, thus, $\sup_{t\in\mathbb{N}} \mathrm{E}[x_t^p] < \infty$. As a multi-dimensional GP-SSM depends on separated GPs and the Gram matrix $\check{K}$ remains bounded, (55) can be extended to $\boldsymbol{x}_t \in \mathbb{R}^{n_x}$. Consequently, $\sup_{t\in\mathbb{N}} \mathrm{E}\,\|\boldsymbol{x}_t\|^p < \infty$ holds. Finally, this remains obviously true for GP-ASSMs with $\overline{m} < \infty$, since only a subset of past states is considered. This concludes the proof. □

Remark 12. *Note, no boundedness of the input $\boldsymbol{u}_t$ is required for Theorem 3.*

Remark 13. *Many commonly used kernels for GPDMs are bounded, for instance, the squared exponential or Matérn kernel.*

Theorem 3 shows the boundedness of GP-ASSMs for bounded mean function and kernel, which holds for the true as well as for the approximated distribution. However, it is also possible that a GP-ASSM with unbounded kernel leads to bounded dynamics. This mainly depends on the training data. In this case, the boundedness property might be lost for a different maximum length of memory, as the following proposition states.

Theorem 4. *Consider two GP-ASSMs with states $\boldsymbol{x}_t^m$ and $\boldsymbol{x}_t^{m'}$, respectively, such that*

$$\begin{aligned}\boldsymbol{x}_{t+1}^m &\sim \mathcal{N}\big(\boldsymbol{f}_t(\boldsymbol{\xi}_t, \Xi_t^m), F_t(\boldsymbol{\xi}_t, \Xi_t^m)\big) \\ \boldsymbol{x}_{t+1}^{m'} &\sim \mathcal{N}\big(\boldsymbol{f}_t(\boldsymbol{\xi}_t', \Xi_t^{m'}), F_t(\boldsymbol{\xi}_t', \Xi_t^{m'})\big), \end{aligned} \quad (56)$$

where $\overline{m}$ and $\overline{m}'$ are the maximum length of memory. Then, for $\overline{m} < \overline{m}'$

$$\sup_{t\in\mathbb{N}} \mathrm{E}\,\|\boldsymbol{x}_t^m\|^p < \infty \not\Rightarrow \sup_{t\in\mathbb{N}} \mathrm{E}\left\|\boldsymbol{x}_t^{m'}\right\|^p < \infty \quad (57)$$

holds for any $p \in \mathbb{N}$ and $\boldsymbol{x}_0^m = \boldsymbol{x}_0^{m'} \in \mathbb{R}^{n_x}$.

Proof. We use a counter example to prove this theorem. Consider a one-dimensional GP-ASSM with $\overline{m} = 0$ and linear kernel $k(\boldsymbol{z}, \boldsymbol{z}') = \boldsymbol{z}^\top \boldsymbol{z}'$, where $\boldsymbol{z}, \boldsymbol{z}' \in \mathbb{R}^n$. We assume two training points at $X_1 = [-1; 0], X_2 = [1; 0]$ and $Y = [Y_1, Y_2] \in \mathbb{R}^2$ with noise $\sigma_n^2 = 1$ and input $u_t = 0$. Using the definition of (25), the mean f_t and variance F_t of next state x_{t+1}^0 is given by

$$f_t(\boldsymbol{\xi}_t, \emptyset) = \frac{1}{3} x_t^0 (Y_2 - Y_1),\; F_t(\boldsymbol{\xi}_t, \emptyset) = \frac{1}{3}(x_t^0)^2. \quad (58)$$

For $|Y_2 - Y_1| \leq 3$, the sequence $\{x_t^0\}, t \in \mathbb{N}$ is p-bounded, since $x^0 = 0$ is stochastically asymptotically stable in the large. Next, in an alternative GP-ASSM, we use the same training points with $m' \geq 1$. Starting at $x_0^{m'} \in \mathbb{R}\backslash 0$, the distribution of $x_1^{m'}$ can be computed using (58). With a Gaussian distributed sampled $x_1^{m'}$, the next step state $x_{t+1}^{m'}$ for $t \geq 1$ are given by

$$f_t\left(\begin{bmatrix} x_t^{m'} \\ 0 \end{bmatrix}, \Xi_t^{m'}\right) = \frac{x_1^{m'}}{x_0^{m'}} x_t^{m'} \quad (59)$$

$$F_t\left(\begin{bmatrix} x_t^{m'} \\ 0 \end{bmatrix}, \Xi_t^{m'}\right) = 0 \quad (60)$$

$$x_{t+1}^{m'} = \frac{x_1^{m'}}{x_0^{m'}} x_t^{m'}. \quad (61)$$

The predicted variance for all states in the future is zero, since the state $x_1^{m'}$ exactly defines a sample of the GP with a linear kernel. The reason is that a linear function is fully defined by one point unequal zero. Based on the Gaussian distribution of $x_1^{m'}$, the probability, that a trajectory of (61) is unbounded, is computed by

$$\begin{aligned}\mathrm{P}\left(\left|\frac{x_1^{m'}}{x_0^{m'}}\right| > 1\right) = 1 &+ \mathrm{cdf}\left(\frac{-3\left|x_0^{m'}\right| + \Delta Y}{[x_0^{m'}]^2}\right) \\ &- \mathrm{cdf}\left(\frac{3\left|x_0^{m'}\right| + \Delta Y}{[x_0^{m'}]^2}\right), \end{aligned} \quad (62)$$

where $\Delta Y = Y_1 - Y_2$ and cdf denotes the standard normal cumulative distribution function. Since the probability (62) is greater than zero, the sequence $\{x_t^{m'}\}, t \in \mathbb{N}$ is not p-bounded. Hence, a different maximum length of memory $\overline{m}$ of a GP-ASSM might lead to a loss boundedness property as stated in Theorem 4. □

Example 3. In Fig. 9, the counter example from the proof of Theorem 4 is visualized. For this purpose, we employ two GP-ASSMs with $\overline{m} = 0$ and $\overline{m} = 10$, respectively, based on a linear kernel $k(\boldsymbol{z}, \boldsymbol{z}') = \boldsymbol{z}^\top \boldsymbol{z}'$. Although the samples of the GP-ASSM with $\overline{m} = 0$ are bounded (top), a GP-ASSM with $\overline{m}' = 10$ (bottom) shows unbounded trajectories, which leads to an unbounded mean and variance.

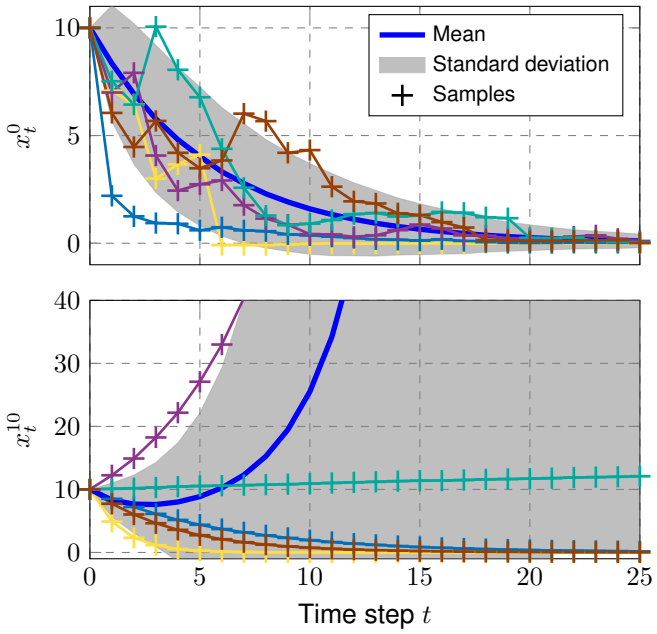


Fig. 9: The GP-ASSM with $\overline{m} = 0$ (top) results in bounded system trajectories whereas a GP-ASSM with $\overline{m}' \geq 1$ (bottom) generates unbounded trajectories. Therefore the boundedness property is lost for different maximum lengths of memory $\overline{m}$

The following theorem shows the relationship between the boundedeness of GP-ASSMs with different length of memory. It states that the approximated dynamics given by a GP-ASSM is bounded if the dynamics of the GP-SSM is bounded. Thus, it allows to use the approximation in control settings without losing the boundedness, which is important for the robustness and stability analysis. Note, that in contrast to Theorem 3, the kernel is not required to be bounded.

Theorem 5. *Considering two GP-ASSMs with states $\boldsymbol{x}_t^m$ and $\boldsymbol{x}_t^{m'}$,respectively, such that*

$$\begin{aligned} \boldsymbol{x}_{t+1}^m &\sim \mathcal{N}\big(\boldsymbol{f}_t(\boldsymbol{\xi}_t, \Xi_t^m), F_t(\boldsymbol{\xi}_t, \Xi_t^m)\big) \\ \boldsymbol{x}_{t+1}^{m'} &\sim \mathcal{N}\big(\boldsymbol{f}_t(\boldsymbol{\xi}_t', \Xi_t^{m'}), F_t(\boldsymbol{\xi}_t', \Xi_t^{m'})\big) \end{aligned} \tag{63}$$

where $\overline{m}$ and $\overline{m}'$ are the maximum length of memory. Then, if $\overline{m} < \overline{m}'$,

$$\sup_{t\in\mathbb{N}, \boldsymbol{x}_0^{m'} \in \mathbb{R}^{n_x}} \mathrm{E}\left\|\boldsymbol{x}_t^{m'}\right\|^p < \infty \Rightarrow \sup_{t\in\mathbb{N}, \boldsymbol{x}_0^{m} \in \mathbb{R}^{n_x}} \mathrm{E}\left\|\boldsymbol{x}_t^{m}\right\|^p < \infty \tag{64}$$

holds for all $p \in \mathbb{N}$.

Remark 14. *Note the swap of $\boldsymbol{x}_t^{m'}$ and $\boldsymbol{x}_t^m$ in* (64) *in contrast to* (57).

Proof. In the following, we split the proof in two parts depending on time step t.
For $t \leq \overline{m}$, the memories Ξ_t^m and $\Xi_t^{m'}$ of both GP-ASSMs are identical and, thus, the expected value is bounded by

$$\sup_{t\in\mathbb{N}, t\leq\overline{m}} \mathrm{E}[(x_t^{m'})^p] = \sup_{t\in\mathbb{N}, t\leq m} \mathrm{E}[(x_t^m)^p] < \infty. \tag{65}$$

For $t > m$, we use the last point in memory $\boldsymbol{x}_{\max(0,t-m'-1)}^{m'}$ as initial point for $\boldsymbol{x}_{t+1}^m$. Thus, we can follow the above argumentation again, which leads to

$$\sup_{t\in\mathbb{N}, t>\overline{m}} \mathrm{E}[(x_t^m)^p] < \infty, \tag{66}$$

such that the boundedness is preserved. □

B. GP Nonlinear Output Error Models

In this section, we transfer our results about boundedness of GP-ASSMs to GP-ANOE models. In GP-ANOE models, the feedback loop is closed by the output $\boldsymbol{y}_t$ instead of the state $\boldsymbol{x}_t$ as in GP-ASSMs. Therefore, we present the following results without further explanation and refer here to section IV-A.

Proposition 5. *Consider a GP-ANOE* (32) *with maximum memory length $\overline{m}$ and bounded mean function and kernel, i.e., $\|\boldsymbol{m}(\boldsymbol{x})\| < \infty, k(\boldsymbol{x}, \boldsymbol{x}') < \infty$, respectively, $\forall \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^{n_\xi}$. Then, the expected value of the sequence $\{\boldsymbol{y}_t^m\}, t \in \mathbb{N}$ by* (32) *is ultimately* p-bounded *by*

$$\sup_{t\in\mathbb{N}} \mathrm{E}\,\|\boldsymbol{y}_t^m\|^p < \infty \tag{67}$$

for all $\boldsymbol{\zeta}_0 \in \mathbb{R}^{n_\zeta}, \forall p \in \mathbb{N}$.

Proof. Analogously to the proof of Theorem 3 with the GP-ANOE model defined by (32). □

Proposition 6. *Consider two GP-ANOEs with outputs $\boldsymbol{y}_t^m$ and $\boldsymbol{y}_t^{m'}$, respectively, such that*

$$\begin{aligned} \boldsymbol{y}_{t+1}^m &\sim \mathcal{N}\big(\boldsymbol{h}_t(\boldsymbol{\zeta}_t^m, \Lambda_t), H_t(\boldsymbol{\zeta}_t^m, \Lambda_t)\big) \\ \boldsymbol{y}_{t+1}^{m'} &\sim \mathcal{N}\big(\boldsymbol{h}_t(\boldsymbol{\zeta}_t^{m'}, \Lambda_t'), H_t(\boldsymbol{\zeta}_t^{m'}, \Lambda_t')\big) \end{aligned} \tag{68}$$

where $\overline{m}$ and $\overline{m}'$ are the maximum length of memory, respectively. Then, for $\overline{m} < \overline{m}'$,

$$\sup_{t\in\mathbb{N}} \mathrm{E}\left\|\boldsymbol{y}_{t+1}^m\right\|^p < \infty \not\Rightarrow \sup_{t\in\mathbb{N}} \mathrm{E}\left\|\boldsymbol{y}_{t+1}^{m'}\right\|^p < \infty \tag{69}$$

holds for $p \in \mathbb{N}$ and $\boldsymbol{\zeta}_0^m = \boldsymbol{\zeta}_0^{m'} \in \mathbb{R}^{n_\zeta}$.

Proof. Analogously to the proof of Theorem 4 with the GP-ANOE model defined by (32). □

Proposition 7. *Consider two GP-ANOE models with outputs $\boldsymbol{y}_t^m$ and $\boldsymbol{y}_t^{m'}$, respectively, such that*

$$\begin{aligned} \boldsymbol{y}_{t+1}^m &\sim \mathcal{N}\big(\boldsymbol{h}_t(\boldsymbol{\zeta}_t^m, \Lambda_t), H_t(\boldsymbol{\zeta}_t^m, \Lambda_t)\big) \\ \boldsymbol{y}_{t+1}^{m'} &\sim \mathcal{N}\big(\boldsymbol{h}_t(\boldsymbol{\zeta}_t^{m'}, \Lambda_t'), H_t(\boldsymbol{\zeta}_t^{m'}, \Lambda_t')\big) \end{aligned} \tag{70}$$

where $\overline{m}$ and $\overline{m}'$ are the maximum length of memory, respectively. Then, if $\overline{m} < \overline{m}'$,

$$\sup_{t\in\mathbb{N}, \boldsymbol{\zeta}_0^{m'} \in \mathbb{R}^{n_y}} \mathrm{E}\left\|\boldsymbol{y}_t^{m'}\right\|^p < \infty \Rightarrow \sup_{t\in\mathbb{N}, \boldsymbol{\zeta}_0^{m} \in \mathbb{R}^{n_x}} \mathrm{E}\,\|\boldsymbol{y}_t^{m}\|^p < \infty \tag{71}$$

holds for all $p \in \mathbb{N}$.

Proof. Analogously to the proof of Theorem 5 with the GP-ANOE model defined by (32). □

V. Case study

In this case study, we demonstrate the modeling of a dynamical system with a GP-SSM and GP-ASSMs with different maximum lengths on memory. As dynamical system to be modeled, we consider the non-autonomous discrete-time predator–prey system introduced in [39]. It is given by

$$\begin{aligned} x^{\mathrm{p}}_{t+1,1} &= x^{\mathrm{p}}_{t,1} \exp\left(1 - 0.4x^{\mathrm{p}}_{t,1} - \frac{(2+1.2u_{t,1})x^{\mathrm{p}}_{t,2}}{1+(x^{\mathrm{p}}_{t,1})^2}\right) \\ x^{\mathrm{p}}_{t+1,2} &= x^{\mathrm{p}}_{t,2} \exp\left(1 + 0.5u_{t,1} - \frac{(1.5-u_{t,2})x^{\mathrm{p}}_{t,2}}{x^{\mathrm{p}}_{t,1}}\right) \end{aligned} \tag{72}$$

$$\boldsymbol{y}^{\mathrm{p}}_t = \boldsymbol{x}^{\mathrm{p}}_t + \boldsymbol{\nu}, \quad \boldsymbol{u}_t = \begin{bmatrix} \cos(0.02\pi t) \\ \sin(0.02\pi t) \end{bmatrix}, \tag{73}$$

with state $\boldsymbol{x}^{\mathrm{p}}_t \in \mathbb{R}^2$, output $\boldsymbol{y}^{\mathrm{p}}_t \in \mathbb{R}^2$, input $\boldsymbol{u}_t \in \mathbb{R}^2$, and Gaussian distributed noise $\boldsymbol{\nu} \in \mathbb{R}^2, \boldsymbol{\nu} \sim \mathcal{N}(\mathbf{0}, 0.05^2 I)$. The states $x^{\mathrm{p}}_{t,1}$ and $x^{\mathrm{p}}_{t,2}$ represent the population size of prays and predators, respectively, but are taken to be continuous. The system dynamics (72) are assumed to be unknown whereas the input and output, given by (73), are assumed to be known. For the modeling with a GP-SSM, 33 training points of a trajectory from the predator–prey system with initial state $\boldsymbol{x}^{\mathrm{p}}_0 = [0.3; 0.8]$ are collected. More detailed, every third state $\boldsymbol{x}^{\mathrm{p}}_t$, input $\boldsymbol{u}^{\mathrm{p}}_t$ and output $\boldsymbol{y}^{\mathrm{p}}_t$ between $t = 1, \dots, 100$ is recorded. Thus, the training set $\mathcal{D} = \{X, Y\}$ consists of

$$\begin{aligned} X &= [\boldsymbol{\xi}_1, \boldsymbol{\xi}_4, \dots, \boldsymbol{\xi}_{97}] \text{ with } \boldsymbol{\xi}_t = [\boldsymbol{x}^{\mathrm{p}}_t; \boldsymbol{u}_t] \\ Y &= [\boldsymbol{y}_1, \boldsymbol{y}_4, \dots, \boldsymbol{y}_{97}]^\top. \end{aligned} \tag{74}$$

Following the structure of GP-SSMs in (7), two GPs are employed to model each element of the state $\boldsymbol{x}^{\mathrm{p}}_t$ separately. The GPs are based on a squared exponential kernel with automatic relevance detection given by

$$k(\boldsymbol{\xi}_t, \boldsymbol{\xi}'_t) = \varphi_1^2 \exp\left(-(\boldsymbol{\xi}_t - \boldsymbol{\xi}'_t)^\top P^{-1} (\boldsymbol{\xi}_t - \boldsymbol{\xi}'_t)\right) \tag{75}$$

with matrix $P = \mathrm{diag}(\varphi_2^2, \dots, \varphi_5^2)$. This kernel is bounded with respect to $\boldsymbol{\xi}_t, \boldsymbol{\xi}'_t \in \mathbb{R}^4$. The hyperparameters $\varphi_1, \dots, \varphi_5$ of each GP are optimized by means of the likelihood function, see [8]. In this study, we model the dynamics (72) with a GP-SSM, a GP-ASSM with maximum length of memory 10 and a GP-ASSM with maximum length of memory 0. For the testing of these models, we select the initial state $\boldsymbol{x}^{\mathrm{p}}_0 = [0.268; 0.400]$. The top plot of Fig. 10 visualizes the trajectory of the predator–prey system (72), considered as the ground-truth. After a transition phase, the numbers of prays (red dashed) and predators (blue solid) converge to a periodic solution. The second plot shows three samples of the GP-SSM drawn by means of Theorem 1. Even though the training set consists only of data up to the time step $t = 97$, see (74), the GP-SSM precisely predicts the trajectory after the transition phase. As the GP-SSM implies $\overline{m} = \infty$, all past state transitions are added to the memory Ξ^∞_t, defined in (24), and used for the next state ahead prediction. Consequently, the shape of each sample is identical in periodic repetitions, as highlighted inside the boxes in the second plot of Fig. 10. However, the drawback of this sampling is the increasing size of the memory Ξ^∞_t and, thus, the increasing Gram matrix $\check{K}^\infty_t$, given by (28). In Fig. 11, the size of the square matrix $\check{K}^\infty_t$ is depicted (solid black) with respect to the time step t. For $t = 0$, the matrix solely contains the covariance between each element of the input training data X. The linear slope is problematic for the computation time as the Gram matrix must be inverted for each prediction step, see (4).

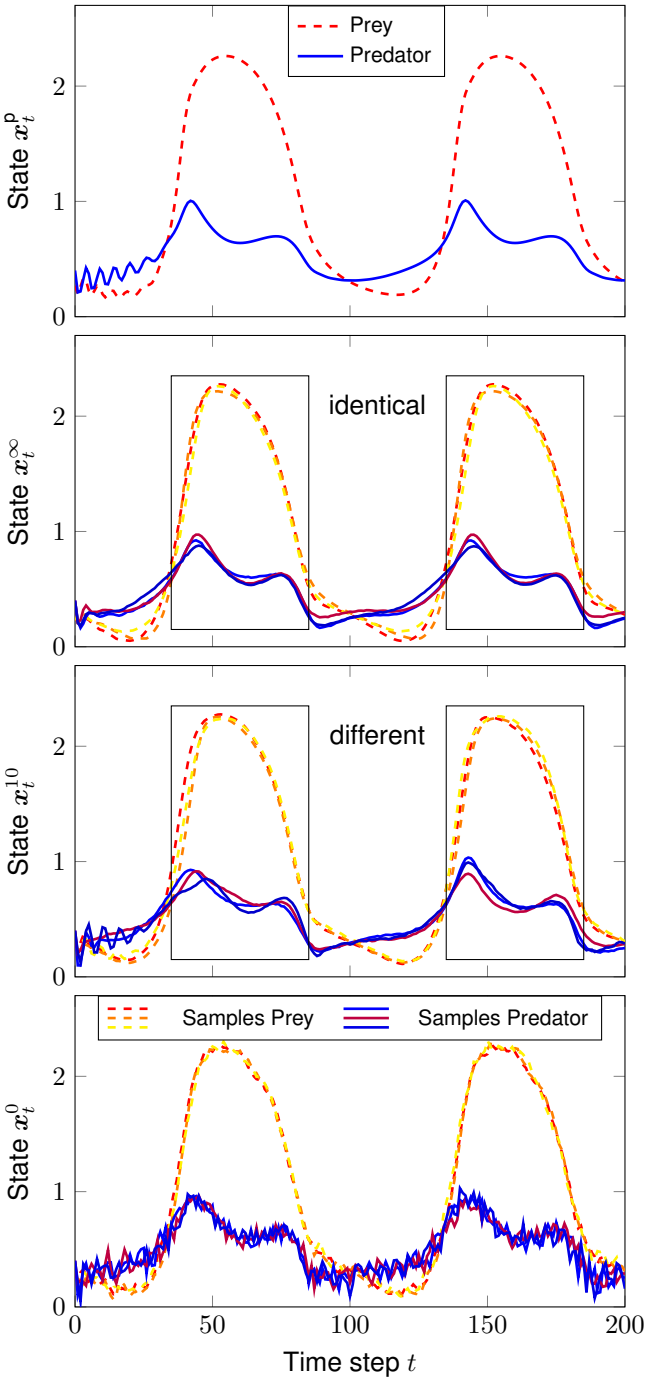


Fig. 10: From top to bottom: Trajectory of predator–prey system, samples of GP-SSM, samples of GP-ASSM with $\overline{m} = 10$, and samples of GP-SSM with $\overline{m} = 0$. For decreasing maximum length of memory of the approximations, the variance is increasing which leads to rougher trajectories.

Three samples of the GP-ASSM with maximum length of

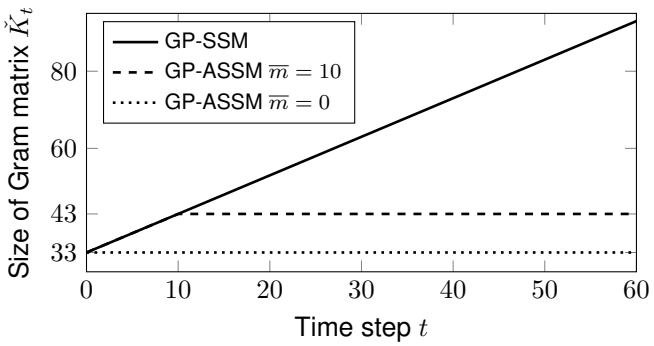


Fig. 11: The number of rows and columns $n_{\mathcal{D}}+\underline{m}$ of the Gram matrix for the GP-SSM and two GP-ASSMs. The GP-ASSM allows to bound the size of the Gram matrix, which must be inverted in each time step.

memory 10, given by the means of (25), are visualized in the third plot of Fig. 10. The samples are similar to the samples of the GP-SSM, since the memory Ξ_t^{10} consists of sufficiently many past states to generate a similar predictive distribution for next step state. However, the shape of the samples differs between the periodic repetitions, as indicated with the two boxes. This variation is due to the reduced memory, which induces that the evolution of the state inside the left box is not considered for the prediction of the corresponding state in the right box. In contrast to the GP-SSM, the maximum length of memory 10 bounds the size (dashed line) of the Gram matrix $\check{K}_t^{10}$ as shown in Fig. 11.

In the bottom plot of Fig. 10, three samples of the GP-ASSM with maximum length of memory 0 are drawn. The variance for each prediction step is significantly higher, as described in Proposition 4, such that the trajectories are rougher. However, the size of the Gram matrix $\check{K}_t^0$ remains constantly low (dotted line) as presented in Fig. 11.

Finally, the GP-SSM and the GP-ASSM with $\overline{m} = 0$ are tested with 50 different initial values, which are drawn from a uniform distribution between $[-5, 5]$ for both states, visualized in Fig. 12. All trajectories are bounded, which supports Theorems 3 and 5.

VI. Discussion

In the previous sections, we show that the sampling of GPDMs, avoiding the impossible sampling of infinite dimensional objects, leads to non-Markovian dynamics. This characteristic is surprising as the representation of the GP-SSM and GP-NOE model, given by (7) and (8) respectively, is based on a Markovian state space structure. However, the covariance term of the GP introduces dependencies across the states that leads to dependencies across time for GPDMs. Thus, the sampling of GP-SSMs and GP-NOE models generates non-Markovian dynamics, which we analyze from a control theoretical point of view. More precisely, a general description for approximations based on the number of included past states/outputs is presented, see Fig. 11, and compared against the true sampling. The approximation error of these models is analyzed with respect to the Kullback-Leibler divergence,

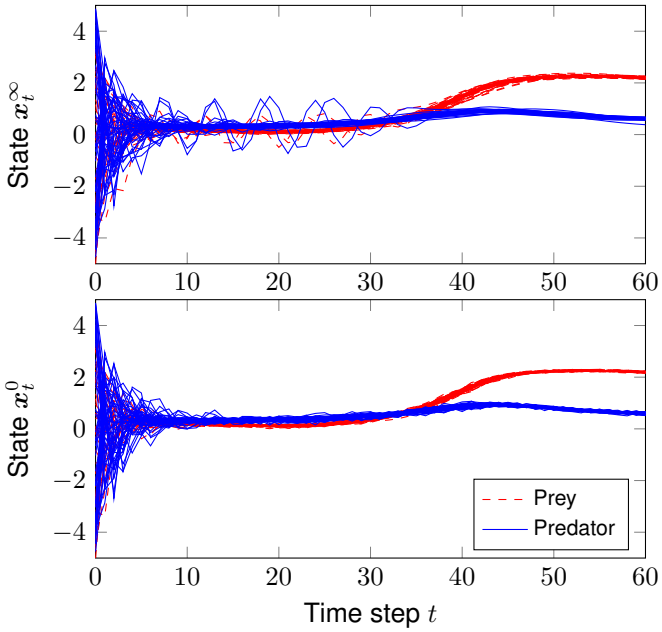


Fig. 12: Trajectories of 50 samples starting from multiple initial points demonstrate the boundedness of the GP-SSM and GP-ASSM.

the mean square prediction error and the variance of the prediction. Furthermore, we prove that the true variance of the next state ahead is always less than the variance of the approximated model as illustrated in Fig. 10. This is relevant for the usage of the approximation in variance based control approaches such as risk-sensitive control approaches, e.g., [40], [41]. Additionally, the boundedness of GPDMs with bounded mean and variance functions, such as the commonly used squared exponential function, is proven and visualized in Fig. 12. The boundedness is an important property for the identification of unknown systems with GPDMs and is likewise exploited for robustness analysis in GPDM based control approaches, see [42]. The introduced characteristics about the relation between the boundedness of the true sampling and the approximations allows a safe usage of the approximation.

Conclusion

In this article, we show that the sampling procedure for Gaussian process dynamical models leads to non-Markovian dynamics. We present a holistic description for approximated models which fulfills the Markov condition. The approximation error of these models is analyzed in respect to the Kullback-Leibler divergence, the mean square prediction error and the variance of the prediction. Furthermore, the boundedness of Gaussian process state space models and nonlinear output error models is analyzed. We proof that the non-Markovian as well as the Markovian approximation is always bounded under specific conditions. Finally, we show the relation between different approximations with respect to the boundedness property of the system. Examples visualize the outcome and highlight the relevance of the results for data-driven based control approaches.

REFERENCES

[1] O. Nelles, *Nonlinear system identification: from classical approaches to neural networks and fuzzy models*. Springer Science & Business Media, 2013.

[2] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models for human motion," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 283–298, 2008.

[3] L. Sigal, M. Isard, H. Haussecker, and M. J. Black, "Loose-limbed people: Estimating 3d human pose and motion using non-parametric belief propagation," *International journal of computer vision*, vol. 98, no. 1, pp. 15–48, 2012.

[4] D. Petelin, A. Grancharova, and J. Kocijan, "Evolving Gaussian process models for prediction of ozone concentration in the air," *Simulation modelling practice and theory*, vol. 33, pp. 68–80, 2013.

[5] I. Hassouneh, T. Serra, B. K. Goodwin, and J. M. Gil, "Non-parametric and parametric modeling of biodiesel, sunflower oil, and crude oil price relationships," *Energy Economics*, vol. 34, no. 5, pp. 1507–1513, 2012.

[6] C. Soize, "A comprehensive overview of a non-parametric probabilistic approach of model uncertainties for predictive models in structural dynamics," *Journal of sound and vibration*, vol. 288, no. 3, pp. 623–652, 2005.

[7] M. Lydia, S. S. Kumar, A. I. Selvakumar, and G. E. P. Kumar, "A comprehensive review on wind turbine power curve modeling techniques," *Renewable and Sustainable Energy Reviews*, vol. 30, pp. 452–460, 2014.

[8] C. E. Rasmussen, *Gaussian processes for machine learning*. Citeseer, 2006.

[9] R. Frigola, Y. Chen, and C. E. Rasmussen, "Variational Gaussian process state-space models," in *Advances in Neural Information Processing Systems*, 2014, pp. 3680–3688.

[10] J. Kocijan, A. Girard, B. Banko, and R. Murray-Smith, "Dynamic systems identification with Gaussian processes," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 11, no. 4, pp. 411–424, 2005.

[11] E. R. Ackermann, J. P. De Villiers, and P. Cilliers, "Nonlinear dynamic systems modeling using Gaussian processes: Predicting ionospheric total electron content over south africa," *Journal of Geophysical Research: Space Physics*, vol. 116, no. A10, 2011.

[12] J. Kocijan and D. Petelin, "Output-error model training for Gaussian process models," in *Proceedings of the Conference on Adaptive and Natural Computing Algorithms*. Springer, 2011, pp. 312–321.

[13] A. Rogers, S. Maleki, S. Ghosh, and J. Nicholas R, "Adaptive home heating control through Gaussian process prediction and mathematical programming," in *Second International Workshop on Agent Technology for Energy Systems (ATES 2011)*, May 2011, pp. 71–78.

[14] J. Kocijan, R. Murray-Smith, C. E. Rasmussen, and A. Girard, "Gaussian process model based predictive control," in *Proceedings of the American control conference (ACC)*, vol. 3. IEEE, 2004, pp. 2214–2219.

[15] L. Hewing, A. Liniger, and M. N. Zeilinger, "Cautious NMPC with Gaussian process dynamics for autonomous miniature race cars," in *Proceedings of the European Control Conference (ECC)*. IEEE, 2018, pp. 1341–1348.

[16] J. Kocijan, *Modelling and Control of Dynamic Systems Using Gaussian Process Models*. Springer, 2016.

[17] K. Ažman and J. Kocijan, "Non-linear model predictive control for models with local information and uncertainties," *Transactions of the Institute of Measurement and Control*, vol. 30, no. 5, pp. 371–396, 2008.

[18] J. Wang, A. Hertzmann, and D. M. Blei, "Gaussian process dynamical models," in *Advances in neural information processing systems*, 2005, pp. 1441–1448.

[19] G. Chowdhary, H. A. Kingravi, J. P. How, and P. A. Vela, "Bayesian nonparametric adaptive control of time-varying systems using Gaussian processes," in *Proceedings of the American Control Conference (ACC)*. IEEE, 2013, pp. 2655–2661.

[20] T. Beckers and S. Hirche, "Stability of Gaussian process state space models," in *Proceedings of the European Control Conference (ECC)*, 2016.

[21] J. R. Medina, T. Lorenz, and S. Hirche, "Synthesizing anticipatory haptic assistance considering human behavior uncertainty," *Transactions on Robotics*, vol. 31, no. 1, pp. 180–190, 2015.

[22] T. Beckers, D. Kulić, and S. Hirche, "Stable Gaussian process based tracking control of Euler-Lagrange systems," *Automatica*, vol. 103, pp. 390–397, 2019.

[23] T. Beckers and S. Hirche, "Equilibrium distributions and stability analysis of Gaussian process state space models," in *Proceedings of the Conference on Decision and Control (CDC)*, 2016.

[24] R. Frigola, F. Lindsten, T. B. Schön, and C. E. Rasmussen, "Bayesian inference and learning in Gaussian process state-space models with particle mcmc," in *Advances in Neural Information Processing Systems*, 2013, pp. 3156–3164.

[25] C. M. Bishop *et al.*, *Pattern recognition and machine learning*. Springer New York, 2006, vol. 4.

[26] H. Mohammadi, R. L. Riche, N. Durrande, E. Touboul, and X. Bay, "An analytic comparison of regularization methods for Gaussian processes," *arXiv preprint arXiv:1602.00853*, 2016.

[27] B. Rakitsch, C. Lippert, K. Borgwardt, and O. Stegle, "It is all in the noise: Efficient multi-task Gaussian process inference with structured residuals," in *Advances in neural information processing systems*, 2013, pp. 1466–1474.

[28] A. Melkumyan and F. Ramos, "Multi-kernel Gaussian processes," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2011.

[29] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.-Y. Glorennec, H. Hjalmarsson, and A. Juditsky, "Nonlinear black-box modeling in system identification: a unified overview," *Automatica*, vol. 31, no. 12, pp. 1691–1724, 1995.

[30] R. H. L. Keviczky, *Nonlinear system identification: input-output modeling approach*. Norwell, MA: Kluwer, 1999.

[31] M. Phan and R. Longman, "Relationship between state-space and input-output models via observer markov parameters," *WIT Transactions on The Built Environment*, vol. 22, 1970.

[32] S. Eleftheriadis, T. Nicholson, M. Deisenroth, and J. Hensman, "Identification of Gaussian process state space models," in *Advances in neural information processing systems*, 2017, pp. 5309–5319.

[33] J. Umlauft, T. Beckers, and S. Hirche, "A scenario-based optimal control approach for Gaussian process state space models," in *Proceedings of the European Control Conference (ECC)*, 2018.

[34] T. L. Gill and W. W. Zachary, *Functional analysis and the Feynman operator calculus*. Springer, 2016.

[35] R. Frigola-Alcalde, "Bayesian time series learning with Gaussian processes," Ph.D. dissertation, University of Cambridge, 2016.

[36] A. G. d. G. Matthews, J. Hensman, R. Turner, and Z. Ghahramani, "On sparse variational methods and the Kullback-Leibler divergence between stochastic processes," *Journal of Machine Learning Research*, vol. 51, pp. 231–239, 2016.

[37] J. Quiñonero-Candela and C. E. Rasmussen, "A unifying view of sparse approximate Gaussian process regression," *Journal of Machine Learning Research*, vol. 6, no. Dec, pp. 1939–1959, 2005.

[38] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," in *Advances in neural information processing systems*, 2006, pp. 1257–1264.

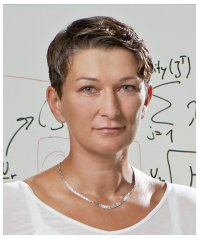
[39] X. Liu, "A note on the existence of periodic solutions in discrete predator–prey models," *Applied Mathematical Modelling*, vol. 34, no. 9, pp. 2477–2483, 2010.

[40] J. M. Hernández, D. Sieber, and S. Hirche, "Risk-sensitive interaction control in uncertain manipulation tasks," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, 2013.

[41] B. Likar and J. Kocijan, "Predictive control of a gas–liquid separation plant based on a gaussian process model," *Computers & chemical engineering*, vol. 31, no. 3, pp. 142–152, 2007.

[42] R. A. Freeman, M. Krstić, and P. Kokotović, "Robustness of adaptive nonlinear control to bounded uncertainties," *Automatica*, vol. 34, no. 10, pp. 1227–1230, 1998.

Thomas Beckers Thomas Beckers received the B.Sc. and M.Sc. degree in electrical engineering in 2010 and 2013, respectively, from the Technische Universität Braunschweig, Germany. He is currently employed at the Chair of Information-Oriented Control at the Technical University of Munich (TUM), Germany, where he is working toward the Ph.D. degree. From November to December 2018, he was a visiting researcher at the University of California, Berkeley. His research interests include data-driven based system identification and control, safe learning, and stability of nonparametric systems.



Sandra Hirche Sandra Hirche received the Diplom-Ingenieur degree in aeronautical engineering from Technical University Berlin, Germany, in 2002 and the Doktor-Ingenieur degree in electrical engineering from Technical University Munich, Germany, in 2005. From 2005 to 2007 she was awarded a Postdoc scholarship from the Japanese Society for the Promotion of Science at the Fujita Laboratory, Tokyo Institute of Technology, Tokyo, Japan. From 2008 to 2012 she has been an associate professor at Technical University Munich. Since 2013 she is TUM Liesel Beckmann Distinguished Professor and has the Chair of Information-oriented Control in the Department of Electrical and Computer Engineering at Technical University Munich. Her main research interests include cooperative and distributed networked control as well as learning control with applications in human-robot interaction, multi-robot systems, and general robotics. She has published more than 150 papers in international journals, books, and refereed conferences.