On Gaussian Process Based Koopman Operators

Yingzhao Lian [*] **Colin N. Jones** [*]

[*] *Automatic Laboratory, Ecole Polytechnique Féderale de Lausanne, Switzerland. (e-mail: {yingzhao.lian, colin.jones}@epfl.ch).*

Abstract: Enabling analysis of non-linear systems in linear form, the Koopman operator has been shown to be a powerful tool for system identification and controller design. However, current data-driven methods cannot provide quantification of model uncertainty given the learnt model. This work proposes a probabilistic Koopman operator model based on Gaussian processes which extends the author's previous results and gives a quantification of model uncertainty. The proposed probabilistic model enables efficient propagation of uncertainty in feature space which allows efficient stochastic/robust controller design. The proposed probabilistic model is tested by learning stable nonlinear dynamics generating hand-written characters and by robust controller design of a bilinear DC motor.

Keywords: Koopman operator, Gaussan Process, Model Predictive Control

1. INTRODUCTION

The Koopman operator captures the behaviours of non-linear systems via linear dynamics, albeit of an infinite dimension. It has been widely used for the analysis of complex dynamics in the fields of molecular physics (e.g. (Wu et al., 2017)) and fluid dynamics (e.g. (Rowley et al., 2009)). In recent years, it has been adopted as a nonlinear controller design method with linear controller design tools (e.g. (Korda and Mezić, 2018a; Surana and Banaszuk, 2016; Peitz and Klus, 2019)).

In the trend of machine learning, the "dynamics mode decomposition" and its extensions play a central role in finding a finite dimensional approximation of a Koopman operator in a data-driven way. These methods produce deterministic models which evolve nonlinear dynamics linearly by mapping states to a feature space with *lifting* functions. In order to accommodate more complex dynamics with a Koopman operator, model capacity is increased either by expanding the feature function space with more basis functions or by approximating lifting functions with more sophisticated structures, such as neural networks ((Takeishi et al., 2017)). However, larger model capacity comes along with a higher risk of overfitting.

This work aims at quantifying model uncertainty while retaining the representative power of a Koopman operator by incorporating Gaussian processes (**GP**) into the model. Unlike the common concept of uncertainty/noise encountered in control science, the model uncertainty in this paper is caused by our limited knowledge of an unknown system and by the use of a finite amount of data, even though the system can be described by deterministic dynamic equations. Problems in the same vein are broadly investigated in Bayesian learning ((Robert, 2007)), and find various applications in control science such as safe learning (e.g. (Akametalu et al., 2014)).

In the following sections, we will first introduce the Gaussian process based Koopman operator in Section 2. Its numerical realization is then elaborated in Section 3, in which the model will be compared against the probabilistic model directly established by the Gaussian process. The proposed probabilistic model is then used for closed-loop robust model predictive control in Section 4. Finally, validation tests are run by modelling hand-written characters and controlling a bilinear DC motor in Section 5.

The major contributions of this work are summarized as follows:

- Establish a probabilistic Koopman operator model and its numerical realization.
- Elaborate the differences and connections between the proposed GP-based Koopman operator model and a direct Gaussian process model.
- The proposed data-driven model allows an efficient robust/stochastic model predictive controller design for nonlinear systems.

2. GAUSSIAN PROCESS BASED KOOPMAN OPERATOR

For the sake of clarity, we will first introduce Koopman operator theory for autonomous systems. Control inputs are incorporated in Section 4.

2.1 Koopman Operator

Given an autonomous, continuous-time dynamical system

$$\dot{x} = F(x), \tag{1}$$

where $F : \mathcal{M} \to \mathcal{M}$ is the system update equation and $\mathcal{M} \subseteq \mathbb{R}^n$ is the state space. Our goal in this section is to demonstrate how the Koopman operator represents the

[⋆] This work has received support from the Swiss National Science Foundation under the RISK project (Risk Aware Data Driven Demand Response, grant number 200021 175627)

state evolution of this system with infinite-dimensional linear dynamics.

Given the function space $\mathcal{F}$ consisting of all functions mapping $\mathcal{M} \to \mathbb{R}$, called 'observables', the Koopman operator ((Koopman, 1931; Koopman and Neumann, 1932)) applied to the observable $f \in \mathcal{F}$ is defined as

$$\mathcal{K}_f^t = f \circ \rho(x,t),$$

where $\rho(x,t)$ denotes the flow at time t starting from x at time 0. The Koopman operator therefore defines a new dynamical system in the function space $\mathcal{F}$ that governs the evolution of the observables.

As the Koopman operator is an operator on a function space, $\mathcal{K}$ is in general infinite-dimensional, but critically it is *linear* even when the dynamics F are non-linear and as such, we call an observable $\phi \in \mathcal{F}$ an eigenfunction associated with the eigenvalue $\lambda \in \mathbb{C}$ if $\mathcal{K}_\phi^t = e^{\lambda t}\phi$, which also means $\frac{d\phi(\cdot)}{dt} = \lambda\phi(\cdot)$. From this we can see that the eigenfunctions (or linear combinations of the eigenfunctions) evolve linearly along the trajectories of our nonlinear system (1)

$$\phi(x(t)) = \phi(\rho(x,t)) = \mathcal{K}_\phi^t(x) = e^{\lambda t}\phi(x) \ . \tag{2}$$

Instead of tracking the state x of our system, we track the evolution of a set of observables f along the state trajectories. Specifically, given an observable in the span of the eigenfunctions $f = \sum v_k(f)\phi_k$, where the weights $v_k(f)$ are called the Koopman modes of f, we notice that the evaluation of the observable at the current $f(x)$ is a *linear* function of the Koopman eigenfunctions evaluated at the current state

$$f(x) = \sum c_k(f)\phi_k(x),$$

and therefore

$$f(x(t)) = \sum c_k(f)\phi_k(x(t)) = \sum c_k(f)e^{\lambda_k t}\phi_k(x) \ .$$

In the rest of this paper, we call evaluation of an observable at x a *lifting* and the space of observables the *feature space.*

Remark 1. We note that the discrete-time Koopman operator can be defined accordingly with a fixed sampling time T_s and $x^+ = F_d(x)$, such that

$$\tilde{\mathcal{K}}_f = f \circ F_d,$$

2.2 Gaussian Process

A Gaussian Process $\mathcal{GP}(\mu,k)$ is an infinite-dimensional distribution over the space of smooth real-valued functions $f : \mathbb{R}^N \to \mathbb{R}$, specialized by a priori mean $\mu : \mathbb{R}^N \to \mathbb{R}$ and covariance functions $k : \mathbb{R}^{N\times N} \to \mathbb{R}$ ((Rasmussen, 2004)), which is also called the kernel function. By definition, function values at a finite set of inputs $[x_1, x_2, x_3 \dots, x_n]$ follows a multi-variate Gaussian distribution $\mathcal{N}(\mu_X, K_{XX})$, where $\mu_X = [\mu(x_1), \dots, \mu(x_n)]^T$ and $K_{XX} = [k(x_i,x_j)]_{i,j=1}^n$. In general, $K_{A,B}$ denotes the cross-covariance between set A and B. If the measurement is contaminated by Gaussian observation noise, $p(y(x)|f(x)) \sim \mathcal{N}(f(x), \sigma^2)$ with σ^2 as measurement noise variance, then the predictive distribution at any point $x^* \in \mathbb{R}^N$ given data $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$ is

$$p(f(x^*)|\mathcal{D}) \sim \mathcal{GP}(\mu_{f|\mathcal{D}}(x^*), k_{f|\mathcal{D}}(x^*,x^*)) \tag{3}$$

$$\mu_{f|\mathcal{D}}(x^*) = \mu(x^*) + K_{x^*X}\hat{K}_{XX}^{-1}y$$

$$k_{f|\mathcal{D}}(x^*,x^*) = K_{x^*x^*} - K_{x^*X}\hat{K}_{XX}^{-1}K_{x^*X}^T$$

where $\hat{K}_{XX} = K_{XX} + \sigma^2 I$ and $y = [y_1, y_2, \dots, y_n]^T$. This defines a scalar-valued regression, for vector-valued regression, various methods have been proposed, such as in (Álvarez and Lawrence, 2011; Bonilla et al., 2008; Micchelli and Pontil, 2005). For simplicity, we will do vector-valued regression via a scalar-valued regression in each dimension.

2.3 Koopman Operator over Gaussian Process

In this section, we will establish the theory in which a Koopman operator evolves a distribution of observables instead of a specific observable. We first make the same assumption as in (Korda and Mezić, 2018b).

Assumption 1. The state space $\mathcal{M} \subset \mathbb{R}^n$ is compact and the Koopman operator is therefore also compact.

Similar to other GP based methods, we further assume that

Assumption 2. The underlying dynamics F is deterministic.

Following the Bayesian learning procedure, we assume an observable has an a priori distribution $f \sim \mathcal{GP}(\mu,k)$.

Proposition 1. If an observable $f \sim \mathcal{GP}(\mu,k)$, then the Koopman operator applied to f is also a Gaussian process such that $\mathcal{K}_f^t = \mathcal{GP}(\mathcal{K}_\mu^t, \mathcal{K}_{k(\cdot,\cdot)}^t)$

Before showing the proof, we mention the following theorem.

Theorem 2. ((Steinwart and Christmann, 2008)) Let $\mathcal{X}$ and $\tilde{\mathcal{X}}$ be sets, and define a map $A : \mathcal{X} \to \tilde{\mathcal{X}}$. Define the kernel k on $\tilde{\mathcal{X}}$. Then the kernel $k(A(x); A(x))$ is a kernel on $\mathcal{X}$.

Proof. (**Proposition 1**)
The Koopman operator is a linear operator. As a Gaussian processes is closed under linear operators ((Rasmussen, 2004; Papoulis and Pillai, 2002)), $\mathcal{K}_f^t$ is a Gaussian process. Since a Gaussian process is fully characterized by its second order statistics ((Bishop, 2006)), its mean and kernel functions can be calculated as follows:

$$\begin{aligned}\mathbb{E}(\mathcal{K}_f^t(x)) &= \mathbb{E}_f(f(\rho(x,t))) \\ &= \int_{-\infty}^{+\infty} f(\rho(x,t))p(f(\rho(x,t)) = \xi))d\xi \\ &= \mu(\rho(x,t)) = \mathcal{K}_\mu^t(x),\end{aligned} \tag{4}$$

$$\begin{aligned}&\mathbb{E}((\mathcal{K}_f^t(x) - \mathbb{E}(\mathcal{K}_f^t(x)))(\mathcal{K}_f^t(y) - \mathbb{E}(\mathcal{K}_f^t(y)))) \\ &= \mathbb{E}_f((f(\rho(x,t)) - \mu(\rho(x,t)))(f(\rho(y,t)) - \mu(\rho(y,t)))) \\ &= k(\rho(x,t), \rho(y,t)) = \mathcal{K}_{k(x,y)}^t,\end{aligned} \tag{5}$$

where $\mathbb{E}_f$ denotes expectation with respect to f. The first equality in Equation 4 and 5 follows Assumption 2. For the sake of completeness, Theorem 2 ensures that $\mathcal{K}_{k(\cdot,\cdot)}^t$ is a kernel function. □

Proposition 1 leads to the following corollary

Corollary 3. If an observable $f \sim \mathcal{GP}(\mu,k)$, then the trajectory of this observable $\{f(\rho(x,t))\}$ is a Gaussian process.

The proof of Corollary 3 is trivial because $\{f(\rho(x,t)\}$ fulfils the definition of a Gaussian process. More specifically, any finite snapshots of a $\mathcal{GP}$ observable trajectory forms a Gaussian random vector. In the rest of this paper, we denote $f(\rho(x,t))$ as $f(t)$ and $\rho(x,t)$ as $x(t)$ for the sake of compactness. We refer to $f \sim \mathcal{GP}(\mu, k)$ as a $\mathcal{GP}$ observable and $\{f(t)\}$ as the trajectory of the $\mathcal{GP}$ observable

3. DATA-DRIVEN MODELLING OF GP BASED KOOPMAN OPERATOR

In this section, we will formulate a numerical method to learn the posterior distribution of the observable $f \sim \mathcal{GP}(\mu, k)$ and its corresponding finite approximation of a Koopman operator given a dataset of measurements. In practice, the data is measured with some sampling time T_s, the dataset is therefore denoted as $\mathcal{D} = \{x(T_i), y(T_i)\}_{i=1}^{n}$, with $y(T_i) = f(x(iT_s)) = f(x(T_i))$. In the rest of this paper, we use capital letters to denote a random variable and their corresponding lower case to denote a realization of this random variable.

3.1 Finite approximation of $\mathcal{GP}$ observable

Assumption 3. The trajectory of a $\mathcal{GP}$ observable is stationary.

Define the Hankel matrix of measurement $\{y(T_i)\}$ matrix $Y_m \in \mathbb{R}^{m \times n-m+1}$ as

$$Y_m = \begin{bmatrix} y(T_1) & y(T_2) & \dots & y(T_{n-m+1}) \\ y(T_2) & y(T_3) & \dots & y(T_{n-m+2}) \\ \vdots & \vdots & \ddots & \vdots \\ y(T_m) & y(T_{m+1}) & \dots & y(T_n) \end{bmatrix}.$$

We first conclude the algorithm with the following lemma.

Lemma 4. Given the singular value decomposition (**SVD**) of $Y_m = U\Lambda V$ with columns of U and rows of V ordered by decreasing singular value, the optimal finite order approximation of the Koopman operator $\mathcal{K}$ of order $j \in [1, m]$ in the mean-square-error sense is [1]

$$K = U[1 : \text{end} -1, 1 : j]^{\dagger} U[2 : \text{end}, 1 : j]$$

The a posteriori distribution of f is

$$f = U[1, 1 : j] \left[z_1 \; z_2 \; \dots \; z_j\right]^T,$$

where $z_i \sim \mathcal{GP}(\mu, k | \{z_i(x(T_p)) = Z_{lift}[i, p]\}_{p=1}^{n-m-1})$ and $Z_{lift} = \Lambda V$

Before showing the proof of Lemma 4, we mention the Karhunen–Loève theorem. It is reasonable to assume that the time interval is bounded, dubbed $t \in [0, T_e]$, then the Karhunen–Loève theorem admits a decomposition of the trajectory of a $\mathcal{GP}$ observable.

Theorem 5. (**Karhunen–Loève theorem** ((Stark and Woods, 1986)))
A centered [2] stochastic process $\{S_t\}_{t \in [0,T_e]}$ admits a decomposition

$$S_t = \sum_{k=0}^{+\infty} Z_k e_k(t),$$

where Z_k are pairwise uncorrelated random variables and $e_k(\cdot)$ are continuous real-valued orthonormal basis functions in $L^2[0, T_e]$. In the case of a zero-mean Gaussian process, Z_k are independent centered Gaussian random variables.

Due to Assumption 1, the system evolves linearly in infinite dimensional feature space within a compact set, zeros and centered limit cycles are therefore the only possible equilibrium points. Hence, a $\mathcal{GP}$ observable is centered and Theorem 5 is applicable to the trajectory of a $\mathcal{GP}$ observable. In the following, we show the proof of Lemma 4.

Proof. **Lemma 4**
According to Corollary 3, we first apply Theorem 5 to a trajectory of a $\mathcal{GP}$ observable

$$Y(t) = \sum_{k=0}^{\infty} Z_k e_k(t). \tag{6}$$

Since X are the states of the system, X is the full statistic of the system dynamics. Hence, Z is $\sigma(X)$-measurable, and there exists a Lebesgue measurable function mapping X to Z ((Jacod and Protter, 2012)). We rewrite Equation (6) as

$$Y(t) = \sum_{k=0}^{\infty} Z_k(X(0)) e_k(t).$$

Following Assumption 3, we get

$$Y(t) = \sum_{k=0}^{\infty} Z_k(X) e_k(t) = f(X(t)) \sim \mathcal{GP}(\mu, k)\ .$$

From Proposition 1, there exists $Z_k(X)$ that is a Gaussian process [3]. Applying the Koopman operator to Z, we get

$$\begin{aligned} Y(t) &= \sum_{k=0}^{\infty} Z_k(X) e_k(t) \\ &= \sum_{k=0}^{\infty} \mathcal{K}^t_{Z_k}(X) e_k(0) \\ &= \sum_{k=0}^{\infty} Z_k(X) \mathcal{K}^{*t}_{e_k(0)}, \end{aligned} \tag{7}$$

the second equality comes with the fact that there exists an adjoint operator of $\mathcal{K}$ due to Assumption 1 ((Eisner et al., 2015)) [4]. Substitute sampling time to Equation 7, we get

$$y(T_i) = \sum_{k=0}^{\infty} z_k(x_i) e_k(T_i) \tag{8}$$

$$= \sum_{k=0}^{\infty} z_k(x_i) \mathcal{K}^{*T_i}_{e_k(0)}. \tag{9}$$

[1] † denotes pseudo inverse and the index follows the MATLAB standard

[2] Meaning that the process is zero-mean.

[3] One can prove that the subprocesses $Z_k(X)$ must be Gaussian processes, however, the proof is non-trivial, please refer to (Feldman and Graczyk, 2000; Skitovitch, 1953) for more details.

[4] The duality is established on the measure of function $\langle \mathcal{K}g(x), g'(x) \rangle = \int g' \circ f'(x) \int (g \circ f(x) p(f, f') d\nu(f)) d\nu(f') = \int g \circ f(x) \int (g' \circ f'(x) p(f, f') d\nu(f')) d\nu(f) = \langle g, \mathcal{K}^* g' \rangle$, with g, g' as mappings from function space $\mathcal{F} \to \mathcal{F}$ and p as transition density function of the system dynamics. In this case, the mapping between Gaussian process must be linear operator, therefore, its conjugate acts on e_k. e_k is considered as the measure of function $Z_k(\cdot)$, hence the adjoint operator evolves the measure/weighting of each Z_k.

For the sake of clarity, we define the power of discrete-time Koopman operator as

$$\tilde{\mathcal{K}}_f^n = f(\underbrace{F_d \circ F_d \cdots \circ F_d}_{n}),$$

The Hankel matrix of measurements can be rewritten as a discrete-time system regarding Assumption 3,

$$
\begin{aligned}
&Y_m = \\
&\begin{bmatrix}
\sum\limits_{k=0}^{+\infty} z_k(x(T_1))e_k(0) & \dots & \sum\limits_{k=0}^{+\infty} z_k(x(T_{n-m+1}))e_k(0) \\
\sum\limits_{k=0}^{+\infty} z_k(x(T_1))e_k(T_s) & \dots & \sum\limits_{k=0}^{+\infty} z_k(x(T_{n-m+1}))e_k(T_s) \\
\vdots & \ddots & \vdots \\
\sum\limits_{k=0}^{+\infty} z_k(x(T_1))e_k(mT_s - T_s) & \dots & \sum\limits_{k=0}^{+\infty} z_k(x(T_{n-m+1}))e_k(mT_s - T_s)
\end{bmatrix} \\
&= \begin{bmatrix}
\sum\limits_{k=0}^{+\infty} z_k(x(T_1))e_k(T_1) & \dots & \sum\limits_{k=0}^{+\infty} z_k(x(T_{n-m+1}))e_k(0) \\
\sum\limits_{k=0}^{+\infty} z_k(x(T_1))\tilde{\mathcal{K}}^*_{e_k(0)} & \dots & \sum\limits_{k=0}^{+\infty} z_k(x(T_{n-m+1}))\tilde{\mathcal{K}}^*_{e_k(0)} \\
\sum\limits_{k=0}^{+\infty} z_k(x(T_1))\tilde{\mathcal{K}}^{*2}_{e_k(0)} & \dots & \sum\limits_{k=1}^{+\infty} z_k(x(T_{n-m+1}))\tilde{\mathcal{K}}^{*2}_{e_k(0)} \\
\vdots & \ddots & \vdots \\
\sum\limits_{k=0}^{+\infty} z_k\tilde{\mathcal{K}}^{*m-1}_{e_k(0)} & \dots & \sum\limits_{k=0}^{+\infty} z_k(x(T_{n-m+1}))\tilde{\mathcal{K}}^{*m-1}_{e_k(0)}
\end{bmatrix} \\
&= \underbrace{\begin{bmatrix}
e_1(0) & e_2(0) & \dots \\
\mathcal{K}^*_{e_1(0)} & \mathcal{K}^*_{e_2(0)} & \dots \\
\vdots & \vdots & \vdots \\
\mathcal{K}^{*m}_{e_1(T-1)} & \mathcal{K}^{*m}_{e_e(T-1)} & \dots
\end{bmatrix}}_{E}
\underbrace{\begin{bmatrix}
z_1(x(T_1)) & \dots & z_1(x(T_{n-m+1})) \\
z_2(x(T_1)) & \dots & z_2(x(T_{n-m+1})) \\
\vdots & \vdots & \vdots
\end{bmatrix}}_{Z_{lift}} .
\end{aligned}
$$

(Gerbrands, 1981) showed that the numerical finite realization of Karhunen–Loève decomposition is equivalent to the singular value decomposition (**SVD**), $Y_m = U\Lambda V$, with $E = U$ and $Z_{lift} = \Lambda V$. Meanwhile, if U and V is ordered by decreasing singular value, then the optimal approximation in the mean-square-error sense up to order j is spanned by the first j dominant singular values and their columns/rows in U/V. Therefore, the corresponding finite order approximation of Koopman operator is $K = U[1 : end - 1, 1 : j]^\dagger U[2 : end, 1 : j]$. From Equation 7, the a posteriori distribution follows Lemma 4. □

To interpret the proposed algorithm in a more intuitive way, we denote the i^{th} column of U as U_i. U_1 captures the most dominant dynamics presented in the column space of Y_m. In control science language, U_1 is the most observable mode in the extended observability matrix. However, due to the limited available trajectories, the lifting Z_1 which maps state X to this mode U_1 cannot be captured completely. An a priori Gaussian process is therefore used to model a distribution of this lifting. In particular, $Z_{lift}[i, j]$ is the realizations of all the i^{th} most dominant $\mathcal{GP}$ observable in the j^{th} $\mathcal{GP}$ observable trajectory. The a priori Gaussian process is then refined to its a posteriori distribution, which is still a Gaussian process. We conclude the algorithm as follows:

Algorithm 1. **$\mathcal{GP}$ Koopman operator Identification**
Inputs: $\mathcal{D} = \{x(T_i), y(T_i)\}_{i=1}^n$, approximation order j;
Ouputs: A posteriori distribution of $\mathcal{GP}$-observable, Koopman operator K;

```
(1) Solve SVD OF $Y_m = U\Lambda V = UZ_{lift}$;
(2) $K = U[1 : \text{end} - 1, 1 : j]^\dagger U[2 : \text{end}, 1 : j]$;
(3) For $i = 1 : j$:
        Calculate a posteriori distribution of $Z_i$ with data
        $\{z_i(x(T_p)) = Z_{lift}[i, p]\}_{p=1}^{n-m-1}$ following Equation
        3;
(4) a posteriori distribution $f = U[1, 1 : j] \, [z_1 \; z_2 \; \dots \; z_j]^T$
```

For an autonomous system, the proposed algorithm has the same form as what we proposed in (Lian and Jones, 2019), which decomposes the modelling of a Koopman operator into a subspace identification problem and a supervised learning problem. The subspace identification problem recovers the Koopman mode and the Koopman operator, while the supervised learning problem learns the lifting from state space to feature space. In an autonomous system, the subspace identification algorithm executes an SVD to find a finite dimensional approximation. If the supervised learning problem is solved with Gaussian process regression, then the algorithm is the same as what we proposed in this paper. Following this heuristic, we suggest that the algorithm proposed in this paper gives a statistical interpretation of the algorithm we proposed before.

Remark 2. Notice that the Karhunen–Loève decomposition does not necessarily admit the eigenfunctions of the Koopman operator.

3.2 Discussion

Gaussian processes are widely used for modelling dynamical systems (e.g. (Klenske et al., 2015; Kocijan et al., 2004)), where system dynamics F or F_d is modelled by a Gaussian process $\mathcal{GP}(\mu_F, k_F)$. In this paper, we call these methods direct GP modelling. Most direct GP models follows the same Assumption 2.

Under Assumption 2, the same state x will evolve into the same x^+ in noiseless case. Therefore, the uncertainty captured by both a direct GP model and a GP based Koopman operator is caused by the information loss. If the state space $\mathcal{M}$ is sampled densely [5] , the model uncertainty will vanish with only measurement noise left. In GP-based Koopman operator, the proposed algorithm finds the most dominant dynamics. However, due to the limited amount of data, the corresponding lifting Z_i has uncertainty, which quantifies the same uncertainty we discussed in Section 1.

In the rest of this section, we will show the difference between the GP based Koopman operator and the direct GP model. If the one-step forward prediction $F_d(x)$ is modelled in general linear estimator form ((Bishop, 2006)) as $x^+ = \sum_i w_i \psi_i(x)$, where ψ_i are some basis functions. As shown in (Rasmussen, 2004), if w_i are i.i.d Gaussian variables, then the linear estimator model produces a direct GP model. Instead, the GP based Koopman operator assumes a distribution over $\psi_i(\cdot)$. In conclusion, the direct GP model assumes a prior knowledge over the dynamics

[5] Meaning that the samples are dense in the state space $\mathcal{M}$

in the feature space with a deterministic initial state while the GP based Koopman operator assumes deterministic dynamics in the feature space with a stochastic initial state in the feature space. Different Bayesian computation methods find the corresponding posterior distribution over the presumed a prior knowledge with respect to the data ((Robert, 2007)). This difference endows the GP-based Koopman operator a computational merit over the direct GP model. As shown in (McHutchon et al., 2015), the distribution of the multi-step forward prediction does not admit a closed form representation, which therefore requires approximation such as moment-matching in (Girard et al., 2003). However, the proposed GP-based Koopman operator evolves linearly in the feature space, therefore, the distribution of multi-step forward prediction is still Gaussian. This benefit allows computationally efficient stochastic/robust controller design, which will be discussed in next section.

4. GP-BASED KOOPMAN ROBUST MODEL PREDICTIVE CONTROL

As discussed in Section 3, this work gives a statistical interpretation of our previous work in (Lian and Jones, 2019). We therefore incorporate the control inputs in an affine form as what we did in (Lian and Jones, 2019), which is a heuristic proven to be effective in (Korda and Mezić, 2018a). For the sake of compactness, please refer to (Lian and Jones, 2019) for more details [6] . The final model is

$$\begin{aligned} z_0 &= \mathcal{GP}(\mu_{f|\mathcal{D}}(x), k_{f|\mathcal{D}}(x)) && \text{state space to observables} \\ z_{k+1} &= Kz_k + Bu_k && \text{observables dynamics} \\ y_k &= Cz_k + Du_k && \text{observables to measurements.} \end{aligned}$$

As z follows a Gaussian distribution, dubbed $z \sim \mathcal{N}(\bar{z}, \sigma_z^2)$, it can be reparametrized as $z = \bar{z} + \sigma_z w$ with $w \sim \mathcal{N}(0, I)$. A robust model predictive control (**MPC**) is formulated as in (Houska and Villanueva, 2019),

$$\begin{aligned} \min_u \max_w \sum_{i=1}^{H} & y_i^T Q y_i + u_{i-1}^T R u_{i-1} \\ \text{s.t.} & \\ & \bar{z}_0 = \mu(x_0|\mathcal{D}), \sigma_z = k(x_0, x_0|\mathcal{D}) \\ & z_0 = \bar{z}_0 + w, w \in [-3\sigma_z, 3\sigma_z] \\ & z_{k+1} = Kz_k + Bu_k \\ & y_k = Cz_k + Du_k \\ & y_i \in \mathcal{Y}, u_i \in \mathcal{U}, i = 0, 1 \ldots H, \end{aligned}$$

where Q and R are penalty matrices for measurements and control inputs, H is the prediction horizon. $\mathcal{U}, \mathcal{Y}$ denotes the feasible sets for control inputs and measurements separately. As w lies in feature space and cannot be measured, we design the following feedback control law to reduce conservatism,

$$\begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{H-1} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 \\ G_{11} & 0 & 0 & \ldots & 0 \\ G_{21} & G_{22} & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ G_{H1} & G_{H2} & G_{H3} & \ldots & 0 \end{bmatrix} \left(\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{H-1} \end{bmatrix} - \begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \\ \vdots \\ \bar{y}_{H-1} \end{bmatrix} \right),$$

where $\bar{y}_i$ represents the nominal trajectory without uncertainty, it is evolved as:

$$\begin{aligned} \bar{z}_0 &= \mu(x_0|\mathcal{D}) \\ \bar{z}_{k+1} &= K\bar{z}_k + Bu_k \\ \bar{y}_k &= C\bar{z}_k + Du_k \end{aligned}$$

This feedback control law only uses the available information up to step i to regulate the control inputs and has a similar form of tube-based robust MPC in (Limon et al., 2010). However, only u_0 is applied and the estimation of z will update once new measurement arrives.

5. VALIDATION

In this section, we will validate the proposed method in both an autonomous system and a system with control inputs, in particular, we will deploy the closed-loop robust MPC formulation shown in Section 4.

5.1 Learning Hand-written Character Dynamics

Learning by demonstration enables robots to imitate human-level control by providing demonstrations (Billard et al. (2008)). A popular method is proposed in Khansari-Zadeh and Billard (2011), which is based on Gaussian mixture models (**GMM**). This method ensures global stability by enforcing a Lyapunov condition. However, this approach is not scalable with respect either to the dimension of the state space or to the amount of data. The proposed method in this paper can also ensure global stability by enforcing stability in the feature space as shown in Lian and Jones (2019), while it has much higher scalability.

In this validation test, the algorithm is applied to learn the dynamics of hand-written characters. We assume that drawing a character is governed by an autonomous, discrete-time dynamical system such that the location of the pen tip x evolves according to $x_{k+1} = F(x_k)$. We apply the proposed method to learn these dynamics. The effectiveness of the algorithm is shown in Figure.1 and Figure.2, whose data comes from Khansari-Zadeh and Billard (2011) with 3 times demonstrations shown in the corresponding figures. All the sample curves start from the same initial point but with different initial states sampling in the feature space. We notice that in both cases, the uncertainty becomes larger when the curves turn which is aligned with our intuition.

5.2 Model and Control of Bilinear Motor

In this section, we apply the proposed method to identify and control a bilinear model of a DC motor Daniel-Berhe and Unbehauen (1998).

$$\begin{aligned} \dot{x}_1 &= -(R_a/L_a)x_1 + (k_m/L_a)x_2 u + u_a/L_a \\ \dot{x}_2 &= -(B/J)x_2 + (k_m/J)x_1 u - \tau_l/J \\ y &= x_2 \end{aligned}$$

where x_1 is the rotor current, x_2 is the angular velocity and the control input u is the stator current. The parameters are $L_a = 0.314$, $R_a = 12.345$, $k_m = 0.253$, $J = 0.00441$, $B = 0.00732$, $\tau_l = 1.47$, and $u_a = 60$. In this experiment,

[6] Technical note on:`http://infoscience.epfl.ch/record/272580?&ln=fr`

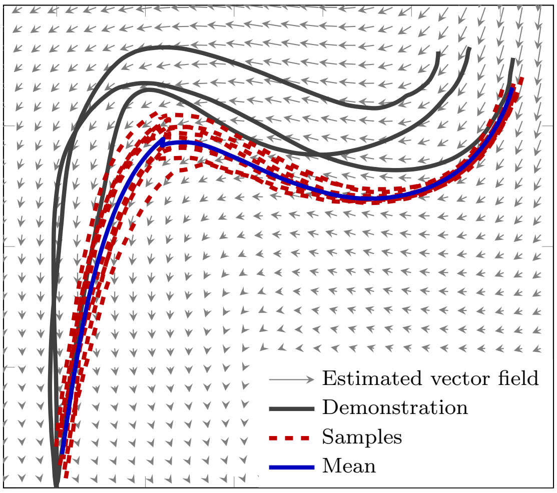

Fig. 1. Learned 18th order dynamics of the character 'r'

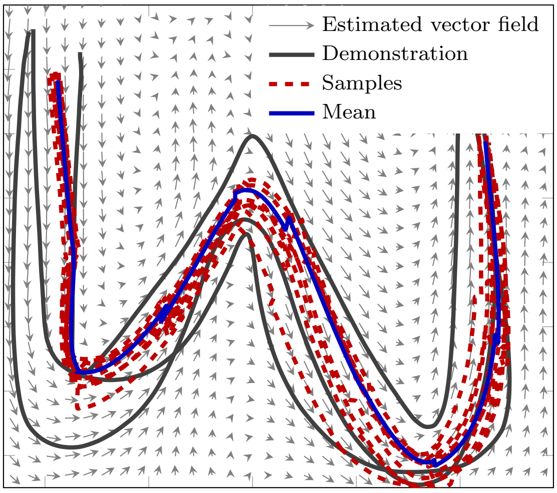

Fig. 2. Learned 22th order dynamics of the character 'w'

only angular velocity x_2 and stator current u is measured and used for modelling.

The effectiveness of the model is shown in Figure.3, in which a local linearized model and an EDMD model from Korda and Mezić (2018a) is shown for performance comparison[7]. It is noted that the mean of the proposed method can properly track the real output while its uncertainty evolves properly with real outputs always included. The closed-loop robust controller proposed in Section 4 is deployed to control the angular velocity x_2. The prediction horizon H is set to 10, Q and R are set to 1 and 0 respectively. The constraints are $u \in [-1, 1]$ and $y \in [-1, 1]$, where y is the angular velocity. The experiments are shown in Figure 4 and 5.

6. CONCLUSION

In this paper, we proposed a probabilistic model of a Koopman operator based on Gaussian process. Based on the proposed framework, we proposed an algorithm to model the system and showed its application in both system identification and controller design. The effectiveness of the proposed model, algorithms and robust MPC controller is validated by two examples.

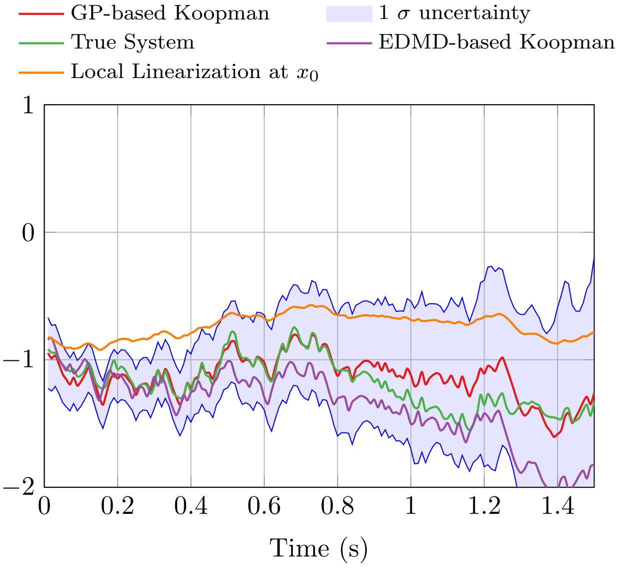

Fig. 3. Comparison of the open-loop prediction given Koopman operator and the real state evolution

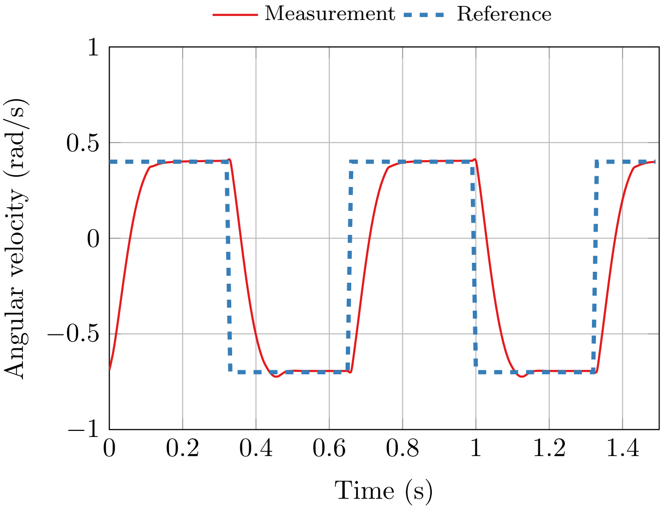

Fig. 4. Angular velocity of the controlled motor

REFERENCES

Akametalu, A.K., Fisac, J.F., Gillula, J.H., Kaynama, S., Zeilinger, M.N., and Tomlin, C.J. (2014). Reachability-based safe learning with gaussian processes. In *53rd IEEE Conference on Decision and Control*, 1424–1431. IEEE.

Álvarez, M.A. and Lawrence, N.D. (2011). Computationally efficient convolved multiple output gaussian processes. *Journal of Machine Learning Research*, 12(May), 1459–1500.

Billard, A., Calinon, S., Dillmann, R., and Schaal, S. (2008). Robot programming by demonstration. *Springer handbook of robotics*, 1371–1394.

[7] For EDMD and the proposed algorithm, we use the same dataset of 2000 datapoints, which is generated by random control inputs

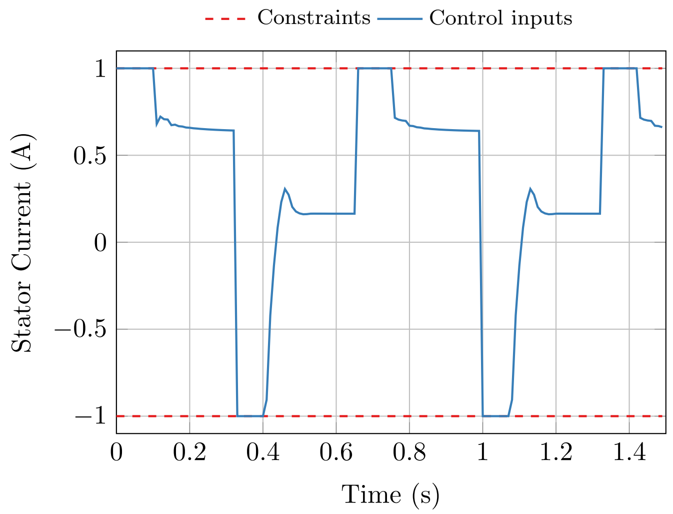

Fig. 5. Control input from GP-based Koopman closed loop robust MPC

Bishop, C.M. (2006). *Pattern recognition and machine learning*. springer.

Bonilla, E.V., Chai, K.M., and Williams, C. (2008). Multi-task gaussian process prediction. In *Advances in neural information processing systems*, 153–160.

Daniel-Berhe, S. and Unbehauen, H. (1998). Experimental physical parameter estimation of a thyristor driven dc-motor using the hmf-method. *Control Engineering Practice*, 6(5), 615–626.

Eisner, T., Farkas, B., Haase, M., and Nagel, R. (2015). *Operator theoretic aspects of ergodic theory*, volume 272. Springer.

Feldman, G. and Graczyk, P. (2000). On the skitovich–darmois theorem for compact abelian groups. *Journal of Theoretical Probability*, 13(3), 859–869.

Gerbrands, J.J. (1981). On the relationships between svd, klt and pca. *Pattern recognition*, 14(1-6), 375–381.

Girard, A., Rasmussen, C.E., Candela, J.Q., and Murray-Smith, R. (2003). Gaussian process priors with uncertain inputs application to multiple-step ahead time series forecasting. In *Advances in neural information processing systems*, 545–552.

Houska, B. and Villanueva, M.E. (2019). Robust optimization for mpc. In *Handbook of Model Predictive Control*, 413–443. Springer.

Jacod, J. and Protter, P. (2012). *Probability essentials*. Springer Science & Business Media.

Khansari-Zadeh, S.M. and Billard, A. (2011). Learning stable nonlinear dynamical systems with gaussian mixture models. *IEEE Transactions on Robotics*, 27(5), 943–957.

Klenske, E.D., Zeilinger, M.N., Schölkopf, B., and Hennig, P. (2015). Gaussian process-based predictive control for periodic error correction. *IEEE Transactions on Control Systems Technology*, 24(1), 110–121.

Kocijan, J., Murray-Smith, R., Rasmussen, C.E., and Girard, A. (2004). Gaussian process model based predictive control. In *Proceedings of the 2004 American control conference*, volume 3, 2214–2219. IEEE.

Koopman, B.O. (1931). Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences of the United States of America*, 17(5), 315.

Koopman, B. and Neumann, J.v. (1932). Dynamical systems of continuous spectra. *Proceedings of the National Academy of Sciences of the United States of America*, 18(3), 255.

Korda, M. and Mezić, I. (2018a). Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93, 149–160.

Korda, M. and Mezić, I. (2018b). On convergence of extended dynamic mode decomposition to the koopman operator. *Journal of Nonlinear Science*, 28(2), 687–710.

Lian, Y. and Jones, C.N. (2019). Learning feature maps of the koopman operator: a subspace viewpoint. In *2019 IEEE Conference on Decision and Control (CDC)*, 4945–4950. IEEE.

Limon, D., Alvarado, I., Alamo, T., and Camacho, E. (2010). Robust tube-based mpc for tracking of constrained linear systems with additive disturbances. *Journal of Process Control*, 20(3), 248–260.

McHutchon, A.J. et al. (2015). *Nonlinear modelling and control using Gaussian processes*. Ph.D. thesis, Citeseer.

Micchelli, C.A. and Pontil, M. (2005). On learning vector-valued functions. *Neural computation*, 17(1), 177–204.

Papoulis, A. and Pillai, S.U. (2002). *Probability, random variables, and stochastic processes*. Tata McGraw-Hill Education.

Peitz, S. and Klus, S. (2019). Koopman operator-based model reduction for switched-system control of pdes. *Automatica*, 106, 184–191.

Rasmussen, C.E. (2004). Gaussian processes in machine learning. In *Advanced lectures on machine learning*, 63–71. Springer.

Robert, C. (2007). *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer Science & Business Media.

Rowley, C.W., Mezić, I., Bagheri, S., Schlatter, P., and Henningson, D.S. (2009). Spectral analysis of nonlinear flows. *Journal of fluid mechanics*, 641, 115–127.

Skitovitch, V. (1953). On a property of the normal distribution. *DAN SSSR*, 89, 217–219.

Stark, H. and Woods, J.W. (1986). *Probability, random processes, and estimation theory for engineers*. Prentice-Hall, Inc.

Steinwart, I. and Christmann, A. (2008). *Support vector machines*. Springer Science & Business Media.

Surana, A. and Banaszuk, A. (2016). Linear observer synthesis for nonlinear systems using koopman operator framework. *IFAC-PapersOnLine*, 49(18), 716–723.

Takeishi, N., Kawahara, Y., and Yairi, T. (2017). Learning koopman invariant subspaces for dynamic mode decomposition. In *Advances in Neural Information Processing Systems*, 1130–1140.

Wu, H., Nüske, F., Paul, F., Klus, S., Koltai, P., and Noé, F. (2017). Variational koopman models: Slow collective variables and molecular kinetics from short off-equilibrium simulations. *The Journal of chemical physics*, 146(15), 154104.