Nonlinear Dyn (2020) 102:241–267
https://doi.org/10.1007/s11071-020-05925-8

ORIGINAL PAPER

Bayesian system ID: optimal management of parameter, model, and measurement uncertainty

Nicholas Galioto · **Alex Arkady Gorodetsky**

Received: 13 April 2020 / Accepted: 28 August 2020 / Published online: 7 September 2020

Abstract System identification of dynamical systems is often posed as a least squares minimization problem. The aim of these optimization problems is typically to learn either propagators or the underlying vector fields from trajectories of data. In this paper, we study a first principles derivation of appropriate objective formulations for system identification based on probabilistic principles. We compare the resulting inference objective to those used by emerging data-driven methods based on dynamic mode decomposition (DMD) and system identification of nonlinear dynamics (SINDy). We show that these and related least squares formulations are specific cases of a more general objective function. We also show that the more general objective function yields more robust and reliable recovery in the presence of sparse data and noisy measurements. We attribute this success to an explicit accounting of imperfect model forms, parameter uncertainty, and measurement uncertainty. We study the computational complexity of an approximate marginal Markov Chain Monte Carlo method to solve the resulting inference problem and numerically compare our results on a number of canonical systems: linear pendulum, nonlinear pendulum, the Van der Pol oscillator, the Lorenz system, and a reaction–diffusion system. The results of these comparisons show that in cases where DMD and SINDy excel, the Bayesian approach performs equally well, and in cases where DMD and SINDy fail to produce reasonable results, the Bayesian approach remains robust and can still deliver reliable results.

Keywords System ID · Approximate marginal MCMC · UKF-MCMC · Bayesian inference · DMD · SINDy

N. Galioto (✉)· A. A. Gorodetsky
Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, USA
e-mail: ngalioto@umich.edu

A. A. Gorodetsky
e-mail: goroda@umich.edu

1 Introduction

Recovering nonlinear models of dynamical systems from data is quickly becoming a primary enabling technology for analysis and decision making in fields spanning science and engineering where first principles models are often incomplete or simply unavailable. Examples range from forecasting the weather and climate [1–3], predicting fluid flows [4–6], and enabling adaptive control [7–10]. All of these fields have a long history of developing estimation and system identification techniques such as advanced Kalman filtering in forecasting [11,12], decomposition methods for computational fluid dynamics [13–15], and a wide ranging set of schemes in adaptive control [16–18]. In this paper, we compare the implicit and explicit optimization formulations posed by several representative approaches, and we demonstrate that algorithms that appropriately manage parameter, model, and measurement uncertainty in a cohesive manner are often

more robust than more standard least squares-based approaches.

For any system identification approach, there are two primary challenges: (1) parameterizing a model space over which to search and (2) posing an optimization problem whose minimum yields an optimal model. A great majority of recent work has focused on addressing the first challenge, primarily due to the rapid availability of machine learning software. These recent works seek to learn neural network representations of problems because of their representation capacity [19–21]. These works are partly motivated by the belief that modern systems are complicated and existing linear or linear-subspace models are no longer capable of representing the systems we seek to model.

In this paper, we explore the second challenge—that of posing an optimization problem, or, more generally, specifying a goal whose minimum will yield a system with predictive power. We argue that this problem is equally, if not more, important than appropriately parameterizing a model space. We support this assertion by showing that many currently used optimization objective specifications fail to recover models *even when the correct model class is known*. Specifically, these specifications cause system identification techniques to break down in the presence of sparse measurements and/or noisy data.

We advocate a probabilistic approach to modeling system dynamics that explicitly provides for the representation and incorporation of three uncertainties: parameter uncertainty, model uncertainty, and measurement uncertainty. This probabilistic setting, given in Sect. 3, poses the problem as a hidden Markov model and is well known in the estimation and filtering literature across disciplines [22–24]. Despite being well known, this setting has not been thoroughly compared to predominant system identification approaches in the context of model learning rather than filtering/smoothing.

The probabilistic learning formulation uses probabilistic first principles to produce a rigorously, rather than heuristically, defined objective function that can either be minimized or solved using Bayesian machinery. The Bayesian framework described in this paper allows us to derive and optimize learning objectives that can be used for the identification of general continuous-time dynamical systems. It turns out that these objectives are also consistent with existing least squares system ID objectives under additional assumptions. As such, they provide a fresh viewpoint on a large class of system ID formulations.

The solution to this Bayesian formulation is a posterior distribution of the model parameters given the observed data. As a result, predictions and forecasting become probabilistic—weighting future outcomes by their relative probabilities. This posterior distribution must be computed using computational inference approaches such as Markov Chain Monte Carlo or variational inference. Given a posterior distribution, goal-oriented estimators can be extracted based on a specified loss and risk metric [25]. For instance, it is well known that the posterior mean is the optimal estimator for Bayes risk with squared loss, and the posterior median is optimal for L_1 loss.

1.1 The challenge and significance of optimal uncertainty management

The two main challenges in optimally managing the uncertainty in a problem revolve around specifying prior knowledge in an accurate way, and managing the computational expense of the algorithm. The first challenge requires converting prior information into probability distributions on the uncertain parameters, process noise, and measurement data. Following this specification, the rules of probability inform how data are sequentially assimilated. The second challenge arises due to the nonlinearity, and therefore non-Gaussianity, of inference in systems where both parameters, states, and state transitions are uncertain.

One way to simplify these challenges is to *ignore* certain sources of uncertainty; in Sect. 4, we show that common least squares system ID approaches take this path. However, such assumptions cause the procedures to break down under sparse and noisy data, as we show in Sect. 6. Indeed, a *full* accounting and management of uncertainty are necessary to achieve reliability under difficult data settings.

Accounting for all these sources of uncertainty results in an increased computational expense associated with characterizing a non-Gaussian posterior distribution. When accounting for each source of uncertainty, the likelihood becomes a function not only of the parameters but also of the time series of the states of the system, drastically increasing the dimensionality of our problem. To overcome this challenge, we reduce the dimension of the likelihood to the dimension of

the parameter vector by integrating out the states from the likelihood. This integration in and of itself can be computationally expensive, but we mitigate this cost by using a recursive solution powered by Kalman filtering [22].

1.2 Contributions

To this end, our contributions involve proving that several existing and popular approaches for system identification, sparse identification of nonlinear dynamics (SINDy) [26] and dynamic mode decomposition (DMD) [15], are realizations of the probabilistic framework under some limiting assumptions (they assume no measurement uncertainty). We choose these two approaches because they are representative of many (nonlinear) least squares type approaches that are used. We then empirically demonstrate that we yield improved predictions compared to these approaches on wide varying problems. Concretely, our contributions are the following:

1. A complexity analysis of the unscented Kalman filter MCMC (UKF-MCMC) algorithm developed in [27], which enables an approximate marginal Markov Chain Monte Carlo algorithm to sample from the marginal posterior of the model parameters;
2. Theorems 4 and 5 proving that DMD and SINDy can be viewed as specific cases of the presented probabilistic approach with additional assumptions of zero measurement noise; and
3. A wide ranging set of numerical simulation results demonstrating the robustness and improved prediction quality of our approach in all cases, including sparse and noisy data.

The UKF-MCMC approach mentioned in the first contribution refers to a computational algorithm that targets the marginal posterior distribution of the model parameters to avoid performing inference over the joint parameter-state space. It can be viewed as an approximation of the marginal likelihood that is traditionally very difficult to compute for dynamical systems [28]. The UKF-MCMC algorithm is one of a number of algorithms that have been recently developed that draw on Gaussian-based filtering to approximate the marginal likelihood [29–31]. These algorithms trade off the approximation quality for some additional computational efficiency compared with the seminal particle-marginal approach of Andrieu [32], which is able to reconstruct the exact posterior. Nevertheless, our results indicate that the posterior approximated by the UKF-MCMC algorithm is still able to reconstruct systems with good accuracy.

We apply the UKF-MCMC algorithm to the hierarchical Bayesian setting where we explicitly learn the process and measurement covariance of the dynamical system. Furthermore, we use standardized uninformative priors for the model parameters and standard half-normal priors for the unknown covariances [33]. As a result, our algorithm requires no additional parameters, besides number of MCMC samples, compared to competing single-point estimators (DMD and SINDy). Furthermore, we provide a computational complexity analysis showing that the expense of our approach compared to these existing approaches grows linearly with the number of data points. However, our accuracy gains are shown to sufficiently offset this expense.

The second contribution aims to uncover, or at least interpret, some of the underlying assumptions that have led to observed poor performance of the methods to which we compare. Many data-driven methods claim a certain degree of objectivity (as compared to, for instance, the Bayesian approach we propose here) because they avoid placing strong assumptions (priors) on the system model that may influence the method's estimate. In reality, however, "analyses that have the appearance of objectivity virtually always contain hidden, and often quite extreme, subjective assumptions" [25]. It will be shown that the estimators DMD and SINDy hold the hidden assumption that uncertainty enters only through process noise and that the measurements are noiseless. Conversely, techniques such as parameter optimization of deterministic ODEs account only for noise in the measurements and not in the process. The Bayesian estimator presented here will be shown to outperform these common approaches as soon as their underlying assumptions are violated, even in the modifications of these algorithms that incorporate denoising [34,35].

1.3 Paper structure

The rest of this paper is structured as follows. In Sect. 2, we explain the central problem this paper hopes to address: how common least squares-based system

ID approaches create objective functions with certain undesirable features. We illustrate this problem by providing the contours of two common objective functions for a simple two-dimensional problem and show how the Bayesian approach incorporates the advantageous features of both without including the problematic features. In Sect. 3, we detail the probabilistic framework of a system ID problem, including the problem setup and primary goals. Then, in Sect. 4, we provide an analysis of four common approaches to system ID with a focus on providing theory that unveils the underlying assumptions used by these approaches. Section 5 outlines the algorithms used to implement the Bayesian approach and provides a comparison of their computational complexity to that of DMD and sparse regression. Finally, Sect. 6 applies the Bayesian algorithm to five different dynamical systems including linear, nonlinear, chaotic, and PDE systems. Comparisons of the Bayesian algorithm to DMD and SINDy are given, and it is shown that not only is the Bayesian approach just as effective for systems for which these common approaches display exemplary performance, but the Bayesian algorithm also remains robust in certain regimes where DMD and SINDy fail.

2 Representative challenges in common least squares approaches to system ID

In this section, we highlight the geometry of the objective functions of several representative optimization formulations for system identification that we explore in this paper. Specifically, we consider three objectives: one that considers measurement uncertainty but no process/model uncertainty; one that assumes process uncertainty but no measurement uncertainty; and finally our proposed approach that considers both process and measurement uncertainty. The first two approaches are the most commonly used, but we show that they suffer from multiple minima and poor data sensitivity, respectively. Furthermore, while variations of these approaches are used on complex systems, we highlight their limitations in an extremely simple setting of recovering a linear pendulum.

To motivate the results, consider a simple setting where the true model is a linear oscillator with a frequency of 2.00rad/s. Suppose that the learning objective is to identify the frequency. One might intuitively believe that the following least squares objective (in the time-domain) would appropriately penalize incorrect frequencies ω

$$J(\omega, T) = \int_0^T \Big(\cos(2.00t) - \cos(\omega t)\Big)^2 dt, \tag{1}$$

where $J(\omega, T)$ measures the "error" of estimating some parameter ω. An optimization scheme would then try to find the parameter ω to minimize J. This objective is not derived from any arguments, rather it is intuitively specified and here we attempt to see whether this specification makes sense.[1]

Prior to considering the full system identification, we consider a property of the least squares objective. We compare the cost of two parameters $\omega = 2.01$ rad/s and $\omega = 4.00$ rad/s at two different times $T = 10$ and $T = 1000$. In the case where we obtain noise-free data for 10 s, we obtain $J(2.01, 10) = 0.02$ and $J(4.00, 10) = 9.63$—as we desire, the cost of estimating $\omega = 2.01$rad/s is more than 100 times lower than estimating $\omega = 4.00$rad/s. However, suppose that we obtain data for 100 times longer. Then, we obtain $J(2.01, 1000) = 1053.96$ and $J(4.00, 1000) = 999.58$—the relative difference between the two objectives has shrunk tremendously.

In this example, even small perturbations from the true parameters of a system yield large errors given enough time, and, in this case, greatly reduce the relative benefit of $\omega = 2.01$ over $\omega = 4.00$. In simpler terms, this example demonstrates that as the number of data points increases, the relative difference between $\omega = 2.01$ and $\omega = 4.00$ decays! The practical implication is that optimization formulations may have significantly more difficulty in distinguishing between correct and incorrect parameters. The issue here is that the least squares objective does not seem to behave as intuition would expect, nor does it match the behavior we are aiming to achieve. Specifically, we seek an objective function that exaggerates the difference between parameters with small errors and those with large errors as more data are obtained.

[1] For this linear problem, it is more appropriate to consider frequency-domain system ID, which would not encounter the problems described here. However, these types of time-domain system ID procedures using least squares-based regression/machine learning approaches are increasingly being used for complex nonlinear systems [36–38], and we seek to show that they can be limited in an extremely simple setting.

In this paper, we show that an approach that introduces (and then seeks to reduce) the uncertainty in parameters, models, and measurements leads to objective functions that are far better behaved. To this end, we have found that existing approaches can be characterized by their consideration of three sources of uncertainty: model structure, model parameters, and measurement. The presence of model structure uncertainty refers to our lack of knowledge of the perfect underlying model structure. Typically, this uncertainty arises from two sources: (1) the model representation is not sufficiently expressive and/or (2) the numerical integration of the model introduces deviations from the true system. Parameter uncertainty refers to lack of knowledge in the parameters that describe the chosen model structure to learn. Finally, measurement uncertainty refers to the fact that sensors that collect data on the system are not free of error—we cannot definitively say that the state of the system is identical to the measurements. Instead, we have some uncertainty as to what the state of the system truly is when we receive a measurement. The noisier the data, the more measurement uncertainty that is present.

While parameter and measurement noise can be readily modeled, representing model structure uncertainty is highly nontrivial. In this paper, we use the simple characterization of including process noise. While the process noise is not the model error, it does encapsulate the fact that the predicted motion is incorrect [39]. In fact, we empirically show that it should be included even when the model class spanned by the parameters encapsulates the truth to yield more robust results.

More concretely, we consider identification problems that arise from solving the following three optimization objectives.

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{n} \|(y_i - x(t_i))\|_2^2 \quad \text{s.t.} \quad \frac{\mathrm{d}x}{dt} = f(t, x; \theta) \tag{2}$$

$$\theta^* = \arg\min_{\theta} \sum_{i=2}^{n} \|y_i - \Psi(y_{i-1}; \theta)\|_2^2 \tag{3}$$

$$\theta^* = \arg\max_{\theta} \log(p(y_1, \ldots, y_n \mid \theta)) \tag{4}$$

where θ are model parameters, f are continuous dynamics representing the time derivatives of a problem, and Ψ are discrete propagators. The first objective [Eq. (2)] assumes deterministic dynamics and performs least squares regression to match the trajectory of a differential equation to the data. The least squares objective here implicitly accounts for measurement noise, and is widely used in the literature [40,41]. The second objective [Eq. (3)] assumes there is no measurement noise, only process noise/model uncertainty, and instead builds a propagator between observations. This objective is representative of DMD [15] and similar least squares approaches [42,43]. The final objective [Eq. (4)] is the log marginal likelihood arising from Theorem 1 that we advocate for in this paper. Note that standard L_2 and sparsity-enhancing L_1 regularizations/priors can also be included to each of these objectives, but they do not change the qualitative conclusions.

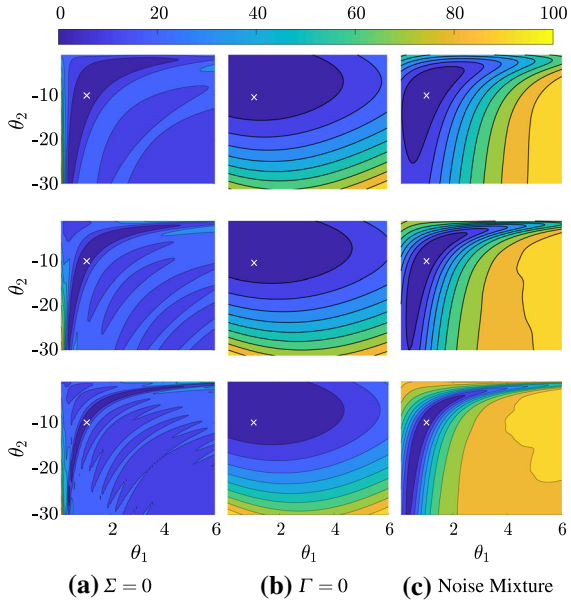


Fig. 1 Comparison of three optimization objectives for the identification of a linear pendulum. The rows correspond to the objective functions obtained after 20, 40, and 80 data points are taken at 0.1 second intervals from top to bottom. White crosses indicate true parameters. Neglecting process noise in the left column results in many local minima. Neglecting measurement noise in the middle column results in an objective insensitive to the number of data. The Bayesian approach in the right column results in the ideal scenario where the objective becomes steeper in the direction of the minimum as the amount of data increases

Figure 1 shows these objective functions for the case of learning a continuous-time linear pendulum

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & \theta_1 \\ \theta_2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \tag{5}$$

where the true parameters are $\theta_1 = 1$ and $\theta_2 = -g/L$. Here, g is the acceleration due to gravity and L is the pendulum length. Data are obtained in 0.1 second increments with noise standard deviation of 0.1. Each column corresponds to a different objective. The first column corresponds to the objective of Eq. (2), the second corresponds to Eq. (3), and the third to Eq. (4).

The rows of this figure correspond to 20, 40, and 80 data points collected in 0.1 second increments, respectively. In the left panel, we see that the assumption of a deterministic system (no process noise) results in many local minima, each of which represents a system that matches the data closely at some points and at other points may be completely opposite. In the middle column, we see that excluding the measurement noise has smoothed over certain features of the deterministic system and the objective becomes insensitive to the number of data points. This panel corresponds to the shape of the objective used by DMD. In this case, finding the minimum of the objective is fast, but the reconstructed system may lack some of the key features of the true trajectory and is not tremendously affected by increasing data. Lastly, the third column represents the objective arising from our probabilistic approach that considers all types of uncertainty. Only in this approach, do we see that increasing data have a beneficial effect on the objective function. Multiple local minima do not exist, the characteristic shape seen in the left column remains, and the objective becomes steeper in the region of the minimum.

3 General problem setting

In this section, we describe the probabilistic framework for system identification, the problem statement, and a description of our high-level solution approach.

3.1 Notation

Let $\mathbb{R}$ denote the set of reals and $\mathbb{Z}_+$ denote the set of positive integers. Let us define the norm of a vector as $\|a\|_C^2 = a^T C^{-1} a$. Let $(\Omega, \mathscr{F}, \mathbb{P})$ denote a probability space, $d \in \mathbb{Z}_+$ denote the dimension of a state space, $m \in \mathbb{Z}_+$ denote the dimension of an observation space, $p \in \mathbb{Z}_+$ denote the dimension of a parameter space, $n \in \mathbb{Z}_+$ denote the number of observations, and $k \in \mathbb{Z}_+$ denote a time index corresponding to a time $t_k \geq 0$. Sequential time indices will typically occur with a constant interval Δ so that $t_k = t_{k-1} + \Delta$. We model the state $X_k \in \mathbb{R}^d$ and the measurement $Y_k \in \mathbb{R}^m$, each at time t_k, as random variables. The lowercase form of these random variables, x_k and y_k, is used to denote a realization of the random variable. A sequence of random variables is denoted with calligraphic font and a subscript corresponding to the time index of the final instance: $\mathscr{Y}_k = Y_1, Y_2, \ldots, Y_k$ and $\mathscr{X}_k = X_0, X_1, \ldots X_k$.

3.2 Probabilistic formulation

In this section, we describe the probabilistic inference problem. We consider discrete-time dynamical systems for the evolution of the unobserved system state $X_k \in \mathbb{R}^d$. The system is observed through a noisy measurement operator providing us data $y_k \in \mathbb{R}^m$. These data can be viewed as realizations of another observed stochastic process Y_k that is dependent on the hidden states.

The dynamics and measurement operators are uncertain and the parameters $\theta \in \mathbb{R}^p$ for $p \in \mathbb{Z}_+$ define a search space over which we will seek to learn the system. We partition the parameters $\theta = (\theta_\Psi, \theta_h, \theta_\Sigma, \theta_\Gamma)$ into different aspects of the problem including the dynamics model parameters θ_Ψ, observation model parameters θ_h, process noise parameters θ_Σ, and observation noise parameters θ_Γ. Together these states, observations, and parameters are related through a hidden Markov model describing a discrete-time stochastic process [22]

$$\begin{aligned} X_k &= \Psi(X_{k-1}, \theta_\Psi) + \xi_k; \quad & \xi_k &\sim \mathscr{N}(0, \Sigma(\theta_\Sigma)) \\ Y_k &= h(X_k, \theta_h) + \eta_k; & \eta_k &\sim \mathscr{N}(0, \Gamma(\theta_\Gamma)), \end{aligned} \tag{6}$$

for $k = 1, \ldots, n$ where $\Psi : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}^d$ is the dynamics operator, $\mathscr{N}$ denotes the normal distribution, ξ_k is the process noise with uncertain covariance $\Sigma(\theta_\Sigma)$, $h : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}^m$ is the observation/measurement operator, Y_k is the predictive stochastic process for the observable, and η_k is the observation noise with uncertain covariance $\Gamma(\theta_\Gamma)$. Finally, we have an additional source of uncertainty corresponding to the initial condition of the states X_0. A visual representation, in the form of a Bayesian network of this model, is provided in Fig. 2.

Examples of Ψ could include physics-inspired PDE operators [42], empirical linear models (a matrix), or nonlinear models such as neural networks [44–46]. The

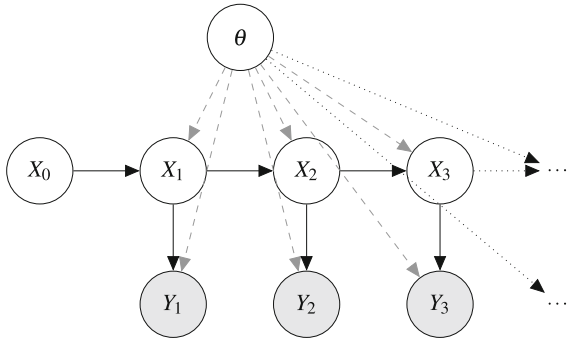


Fig. 2 Bayesian network representation of the system identification problem. The data are realizations of the observed process Y_k

observation operator h is typically some known sensor model that may or may not have uncertain calibration parameters θ_h. We include the parameters for the observation model to maintain generality.

System (6) implicitly defines several probability distributions that completely describe our state of knowledge. The first distribution reflects the Markovian propagation dynamics

$$p(X_k \mid X_{k-1}, \theta_\Psi, \theta_\Sigma) = \frac{\exp\left(-\frac{1}{2}\|X_k - \Psi(X_{k-1}, \theta_\Psi)\|^2_{\Sigma(\theta_\Sigma)}\right)}{\sqrt{2\pi}^d |\Sigma(\theta_\Sigma)|^{\frac{1}{2}}} \tag{7}$$

where the norm inside the exponential represents the *misfit*, or model error, of the dynamics under a fixed set of parameters.

The next distribution reflects the noisy measurement models

$$p(Y_k \mid X_k, \theta_h, \theta_\Gamma) = \frac{\exp\left(-\frac{1}{2}\|Y_k - h(X_k, \theta_h)\|^2_{\Gamma(\theta_\Gamma)}\right)}{\sqrt{2\pi}^m |\Gamma(\theta_\Gamma)|^{\frac{1}{2}}} \tag{8}$$

where the norm inside the exponential represents the *residual* between the states and the observed measurements. Together, along with a prior, these distributions will enable us to concretely form the learning problem, which we establish in the next section.

3.3 Goals

In this section, we describe our two objectives: system identification (learning) and prediction/forecasting.

Our learning objective is to determine a dynamical model Ψ. Specifically, this objective requires representing our knowledge about the parameters θ (or θ_Ψ) after data are obtained. This knowledge is represented via a conditional distribution over θ given the observed data. This distribution is given by Bayes' rule

$$p(\theta \mid \mathscr{Y}_n) = p(\theta)\frac{\mathscr{L}(\theta; \mathscr{Y}_n)}{p(\mathscr{Y}_n)}, \quad \text{where } \mathscr{Y}_n = (y_1, \ldots, y_n), \tag{9}$$

where the prior is denoted by $p(\theta)$ and the marginal likelihood is a function of the unknown parameter

$$\mathscr{L}(\theta; \mathscr{Y}_n) \equiv p(Y_1 = y_1, \ldots, Y_n = y_n \mid \theta). \tag{10}$$

This conditional/posterior distribution captures all the relevant information about our parameters contained in the data. It will be useful to leverage the sequential/Markovian nature of the process to factorize this likelihood as

$$\begin{aligned}\mathscr{L}(\theta; \mathscr{Y}_n) &= p(Y_1 = y_1 \mid \theta)\prod_{k=2}^{n} p\left(Y_k = y_k \mid \theta, \mathscr{Y}_{k-1}\right) \\ &= \mathscr{L}_1(\theta; \mathscr{Y}_1)\prod_{k=2}^{n} \mathscr{L}_k\left(\theta; \mathscr{Y}_k\right)\end{aligned} \tag{11}$$

where we have set $\mathscr{L}_1(\theta; \mathscr{Y}_1) \equiv p(Y_1 = y_1 \mid \theta)$ and $\mathscr{L}_k(\theta; \mathscr{Y}_k) \equiv p(Y_k = y_k \mid \theta, \mathscr{Y}_{k-1})$ for $k = 2, \ldots, n$.

The challenge is computing the marginal likelihood when the states are also uncertain due to process noise. When both parameters and states are uncertain the joint posterior becomes

$$p(\theta, \mathscr{X}_n | \mathscr{Y}_n) = p(\theta)\frac{\mathscr{L}(\theta; \mathscr{X}_n, \mathscr{Y}_n)}{p(\mathscr{Y}_n)}, \tag{12}$$

where $\mathscr{L}(\theta; \mathscr{X}_n, \mathscr{Y}_n)$ is the joint likelihood. Similar to the marginal likelihood, the joint likelihood is once again a function of θ, but is now defined as

$$\mathscr{L}(\theta; \mathscr{X}_n, \mathscr{Y}_n) \equiv p(\mathscr{Y}_n | \mathscr{X}_n, \theta) p(\mathscr{X}_n | \theta). \tag{13}$$

Using Eqs. 7 and 8, the joint likelihood becomes

$$\begin{aligned}\mathscr{L}(\theta; \mathscr{X}_n, \mathscr{Y}_n) = \prod_{k=1}^{n} &\left(\frac{\exp\left(-\frac{1}{2}\|Y_k - h(X_k, \theta_h)\|^2_{\Gamma(\theta_\Gamma)}\right)}{\sqrt{2\pi}^m |\Gamma(\theta_\Gamma)|^{\frac{1}{2}}}\right) \\ &\times \left(\frac{\exp\left(-\frac{1}{2}\|X_k - \Psi(X_{k-1}, \theta_\Psi)\|^2_{\Sigma(\theta_\Sigma)}\right)}{\sqrt{2\pi}^d |\Sigma(\theta_\Sigma)|^{\frac{1}{2}}}\right).\end{aligned} \tag{14}$$

We now see that our target marginal likelihood (11) is related to this joint likelihood through a marginalization procedure (integration)

$$\mathscr{L}(\theta; \mathscr{Y}_n) = \int \mathscr{L}(\theta; \mathscr{X}_n, \mathscr{Y}_n) d\mathscr{X}_n. \tag{15}$$

Evaluating an integral such as this is in general very computationally expensive, but in the following section, a recursive method of evaluating this integral will be given that significantly reduces the cost.

Our second goal is to predict, or forecast, the system state at some future time t_k. This prediction could either be the full posterior predictive distribution $p(X_k \mid \mathscr{Y}_n)$ or some "best estimate" X_k^* that can be derived from the posterior to satisfy some optimality conditions [25]. Furthermore, these two goals (system identification and prediction) are related in that the prediction is obtained by averaging over all possible system parameters, weighted according to the posterior distribution,

$$p(X_k \mid \mathscr{Y}_n) = \int p(X_k \mid \theta) p(\theta \mid \mathscr{Y}_n) d\theta. \tag{16}$$

3.4 Marginal likelihood computation

In this section, we review the formulas for computing the marginal likelihood used in the target distribution (9). We first present the general case [22] provided by the result of Theorem 1.

Theorem 1 (Marginal likelihood (Th. 12.1 [22])) *Let $\mathscr{Y}_k \equiv \{y_i; i \leq k\}$ denote the set of all observations up to time k. Let the initial condition be uncertain with distribution $p(X_0 \mid \theta)$. Then, the marginal likelihood (11) is defined recursively in three stages: prediction*

$$p(X_{k+1} \mid \theta, \mathscr{Y}_k) = \int \frac{\exp\left(-\frac{1}{2}\|X_k - \Psi(X_{k-1}, \theta_\Psi)\|^2_{\Sigma(\theta_\Sigma)}\right)}{\sqrt{2\pi}^d |\Sigma(\theta_\Sigma)|^{\frac{1}{2}}} \times p(X_k \mid \theta, \mathscr{Y}_k) dX_k \tag{17}$$

update,

$$p(X_{k+1} \mid \theta, \mathscr{Y}_{k+1}) = p(X_{k+1} \mid \theta, \mathscr{Y}_k) \times \frac{\exp\left(-\frac{1}{2}\|Y_k - h(X_k, \theta_h)\|^2_{\Gamma(\theta_\Gamma)}\right)}{\sqrt{2\pi}^m |\Gamma(\theta_\Gamma)|^{\frac{1}{2}} p(Y_{k+1} \mid \theta, \mathscr{Y}_k)} \tag{18}$$

and marginalization,

$$\mathscr{L}_{k+1}(\theta \mid \mathscr{Y}_{k+1}) = \int p(X_{k+1} \mid \theta, \mathscr{Y}_k) \times \frac{\exp\left(-\frac{1}{2}\|Y_k - h(X_k, \theta_h)\|^2_{\Gamma(\theta_\Gamma)}\right)}{\sqrt{2\pi}^m |\Gamma(\theta_\Gamma)|^{\frac{1}{2}}} \mathrm{d}X_{k+1} \tag{19}$$

for $k = 1, 2, \ldots$.

This theorem provides a recursive algorithm for evaluating the marginal likelihood. This recursion requires not only maintaining a standard Bayesian filter for the prediction and update steps, but also keeping track of the marginalized distribution after every observation. Extensions to situations where data are not obtained at every time step is trivial—for times when no data are obtained, the update step is skipped.

4 Analysis of common assumptions and existing approaches

In this section, we specialize the marginal likelihood from the general case to several special cases. The special cases reviewed here are (1) zero process noise (model error is ignored) and (2) noiseless, invertible measurements (measurement error is ignored). We then provide the additional assumptions that are necessary to arrive at the DMD and SINDy algorithms. Figure 3 is a flowchart showing how we begin with the joint log likelihood and arrive at a number of special cases of the log likelihood.

4.1 Zero process noise

Objective (2) uses a deterministic model, discarding process noise. In this setting, the distribution (7) reduces to a Dirac delta function

$$p(X_k \mid X_{k-1}, \theta_\Psi, \theta_\Sigma) = \delta_{X_k}(\Psi(X_{k-1}, \theta_\Psi)). \tag{20}$$

The assumption of zero process noise leads to the marginal likelihood given in Theorem 2.

Theorem 2 (Marginal likelihood—zero process noise) *Let the dynamics model be deterministic. Then, the marginal likelihood (11) is defined recursively as*

$$\mathscr{L}_k(\theta; \mathscr{Y}_k) = \frac{\exp\left(-\frac{1}{2}\|y_k - h(\Psi^k(x_0, \theta_\Psi), \theta_h)\|^2_{\Gamma(\theta_\Gamma)}\right)}{\sqrt{2\pi}^m |\Gamma(\theta_\Gamma)|^{\frac{1}{2}}} \tag{21}$$

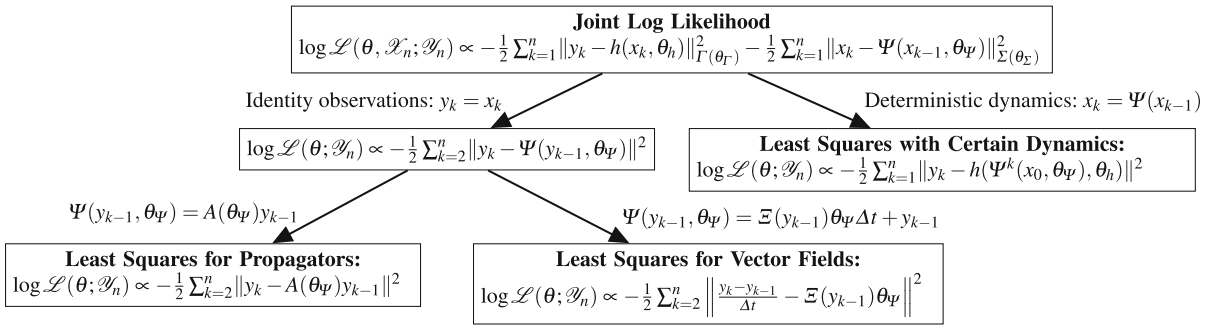


Fig. 3 Flowchart of the logic in Sects. 3.4 and 4

for $k = 1, \ldots, n$ where Ψ^k denotes k applications of the dynamics model. Moreover, the log marginal likelihood becomes

$$\log \mathscr{L}(\theta; \mathscr{Y}_n) = \sum_{k=1}^{n} \left(-\frac{1}{2} \|y_k - h(\Psi^k(x_0, \theta_\Psi), \theta_h)\|^2_{\Gamma(\theta_\Gamma)} \right) - \frac{nm}{2} \log 2\pi - \frac{n}{2} \log |\Gamma(\theta_\Gamma)|. \tag{22}$$

Proof The proof follows from the fact that a deterministic system must follow a fixed trajectory defined entirely by the parameters. In other words, we have $p(\mathscr{X}_n \mid \theta_\Psi, \mathscr{Y}_n) = p(\mathscr{X}_n \mid \theta_\Psi) = \delta_{\Psi(x_0,\theta_\Psi),\ldots,\Psi^n(x_0,\theta_\Psi)}(\mathscr{X}_n)$. As a result, the second term of the joint likelihood 14 drops out, and we are left with

$$\mathscr{L}(\theta; \mathscr{X}_n, \mathscr{Y}_n) = \prod_{k=1}^{n} \frac{\exp\left(-\frac{1}{2}\|y_k - h(x_k, \theta_h)\|^2_{\Gamma(\theta_\Gamma)}\right)}{\sqrt{2\pi}^m |\Gamma(\theta_\Gamma)|^{\frac{1}{2}}}.$$

Because the dynamics are deterministic, we have $x_k = \Psi^k(x_0, \theta_\Psi)$. Thus, the likelihood no longer depends on the states other than the known initial state, and what remains is the marginal likelihood as stated

$$\mathscr{L}(\theta; \mathscr{Y}_n) = \prod_{k=1}^{n} \frac{\exp\left(-\frac{1}{2}\|y_k - h(\Psi^k(x_0, \theta_\Psi), \theta_h)\|^2_{\Gamma(\theta_\Gamma)}\right)}{\sqrt{2\pi}^m |\Gamma(\theta_\Gamma)|^{\frac{1}{2}}}.$$

Taking the log of this expression completes the proof. □

4.2 Noiseless and invertible measurements

In this section, we consider the ramifications on the posterior of assuming no measurement noise. In the next section, we will show that several least squares optimization approaches correspond to this case.

Consider an invertible observation operator so that the states are uniquely determined $X_k = h^{-1}(Y_k)$. Using this assumption in System (6) leads to a Markovian system for the system observables

$$Y_{k+1} = h\left(\Psi\left(h^{-1}(Y_k), \theta_\Psi\right) + \xi_k, \theta_h\right) \tag{23}$$

for $k = 1, \ldots, n-1$ where $\xi_k \sim \mathscr{N}(0, \Sigma(\theta_\Sigma))$.

This assumption yields the marginal likelihood given in Theorem 3.

Theorem 3 (Marginal likelihood—noiseless, invertible observations) *Let h be an invertible operator and the measurements be noiseless. Then, the marginal likelihood* (11) *is defined recursively as*

$$\mathscr{L}_k(\theta; \mathscr{Y}_k) = |\nabla h^{-1}(y_k)| \frac{\exp\left(-\frac{1}{2}\|h^{-1}(y_k) - \Psi\left(h^{-1}(y_{k-1}), \theta_\Psi\right)\|^2_{\Sigma(\theta_\Sigma)}\right)}{\sqrt{2\pi}^d |\Sigma(\theta_\Sigma)|^{\frac{1}{2}}} \tag{24}$$

for $k = 2, \ldots, n$ and

$$\log \mathscr{L}_1(\theta; \mathscr{Y}_1) = \log \int \exp\left(\|h^{-1}(y_1) - \Psi(X_0; \theta_\Psi)\|^2_{\Sigma(\theta_\Sigma)}\right) p(X_0 \mid \theta) dX_0 - \frac{d}{2} \log 2\pi - \frac{1}{2} \log |\Sigma(\theta_\Sigma)|. \tag{25}$$

Together, the log marginal likelihood becomes

$$\log \mathscr{L}(\theta; \mathscr{Y}_n) = \sum_{k=2}^{n} \left(\log |\nabla h^{-1}(y_k)| - \frac{1}{2}\|h^{-1}(y_k) - \Psi\left(h^{-1}(y_{k-1}), \theta_\Psi\right)\|^2_{\Sigma(\theta_\Sigma)} \right) - \frac{nd}{2} \log 2\pi - \frac{n}{2} \log |\Sigma(\theta_\Sigma)| + \log \mathscr{L}_1(\theta; \mathscr{Y}_1). \tag{26}$$

Proof Noiseless observations $y_k = h(x_k, \theta_h)$ imply that $\|Y_k - h(X_k, \theta_h)\|^2_{\Gamma(\theta_\Gamma)} = 0$ such that the first term of the joint likelihood (14) drops out. Second, we notice that we can rewrite the second term in terms of Y_k rather than X_k by using the change of variables formula

$$\mathscr{L}(\theta; \mathscr{Y}_n) = \prod_{k=1}^{n} |\nabla h^{-1}(Y_k)| \frac{\exp\left(-\frac{1}{2}\|h^{-1}(Y_k) - \Psi(h^{-1}(Y_{k-1}), \theta_\Psi)\|^2_{\Sigma(\theta_\Sigma)}\right)}{\sqrt{2\pi}^d |\Sigma(\theta_\Sigma)|^{\frac{1}{2}}}.$$

The likelihood no longer depends on the states and thus the given result is the marginal likelihood. □

As we intuitively expected, there is no marginalization over states under this assumption because the learning problem effectively "resets" after every data point. After the reset, the states are at their true value, and optimization progresses to ensure that the residual of propagation between true values is small. This is exactly the same methodology that inspires the least squares regression-based approaches such as DMD and SINDy. In fact, we will show that special assumptions on h and Ψ recover these least squares approaches.

Remark 1 (Data on initial condition) If the initial condition is treated as beginning when the data are obtained, then the log likelihood for the first data point becomes independent of the parameters and we can set it to an arbitrary constant.

4.3 Dynamic mode decomposition (DMD)

Dynamic mode decomposition (DMD) is a data-driven method for system identification that is used to identify the "dynamic modes" of a dynamical system [15]. These modes reveal characteristics such as unstable growth modes, resonance, and spectral properties [47]. DMD is favorable when the system at hand is high dimensional but has some hidden low-dimensional structure, as is the case in many fluids problems. DMD first organizes a series of measurements at regular time intervals into two matrices

$$Y = \begin{bmatrix} y_1 & y_2 & \dots & y_{n-1} \end{bmatrix}; \quad Y' = \begin{bmatrix} y_2 & y_3 & \dots & y_n \end{bmatrix}, \tag{27}$$

and then seeks a linear operator A which maps the observables from one time step to the next, i.e., $Y' = AY$. To find A, one simply minimizes the Frobenius norm of $AY - Y'$ by solving the least squares problem

$$A = \arg\min_{\tilde{A}} \sum_{k=2}^{n} \|y_k - \tilde{A} y_{k-1}\|^2. \tag{28}$$

The solution is given by $A = Y'Y^\dagger$, where $\dagger$ denotes the pseudo-inverse.

The method given above may at first appear only applicable to linear systems, but [48] showed that in the nonlinear case, the approximated operator A and its corresponding modes are approximations to the linear but infinite-dimensional Koopman operator and Koopman modes, respectively, thus revealing its applicability to nonlinear systems.

Next, we show the least squares procedure for DMD can also be *derived* directly from the general probabilistic system (6) under certain assumptions.

Theorem 4 (DMD as a maximum likelihood of system (6)) *Assume a linear model $\Psi(X_k, \theta_\Psi) = \theta_\Psi X_k$; identity observation operator $h = I$; noiseless measurements $\Gamma(\theta_\Gamma) = 0$; and identity process noise $\Sigma(\theta_\Sigma) = I$. Then, the maximum marginal likelihood estimator corresponding to System (6) is equivalent to the least squares objective of the DMD problem (28).*

Proof This result uses a straightforward application of Theorem 3. Without loss of generality, we use the fact that the first measurement is of the initial condition, and therefore we can ignore $\mathscr{L}_1$. Here, we have an identity observation operator, and therefore the inverse and Jacobian are also the identity. The dynamics are linear and unknown so we can write $A \equiv \theta_\Psi$. With these substitutions, the log marginal likelihood (26) becomes

$$\log \mathscr{L}(\theta; \mathscr{Y}_n) = \sum_{k=2}^{n} \left(\log|I_d| - \frac{1}{2}\|y_k - A y_{k-1}\|^2_{I_d} \right) - \frac{nd}{2}\log 2\pi - \frac{n}{2}\log|I_d|. \tag{29}$$

After evaluating $\log|I_d| = 0$, we arrive at our stated result

$$\log \mathscr{L}(\theta; \mathscr{Y}_n) = -\sum_{k=2}^{n} \frac{1}{2}\|y_k - A y_{k-1}\|^2 - \frac{nd}{2}\log 2\pi. \tag{30}$$

Clearly, the maximizer of this function is equivalent to the minimizer of (28). □

While the invertible measurement operator is not a restrictive assumption because all DMD cares about is mapping observables and not underlying states, Theorem 4 shows why DMD may not be appropriate for cases where the observations are noisy. This fact has been recognized in the literature and several procedures for rectifying this issue have been proposed. For instance, [34] showed that total least squares is a more appropriate algorithm to identify A when measurement noise is present, a method known as *total DMD* (TDMD). For a full analysis of the total least squares problem, see [49,50]. We will empirically compare TDMD to our approach in Sect. 6, where we see that it also performs worse than the posterior predictive mean. Future work will attempt to determine the assumptions that TDMD makes in the context of System (6).

In [51], another connection between the Bayesian approach to DMD was developed that infers the Koopman modes and eigenfunctions of the Koopman operator directly, rather than learning the dynamical operator itself. That work showed that when the measurements are noiseless, the maximum likelihood estimate of their Bayesian model, TDMD, and DMD all provide the same estimate. In contrast, here we have provided our result in terms of the underlying hidden state dynamics rather than explicitly assuming observation dynamics.

One benefit of the analysis in our context is that our use of an underlying state-space model makes the framework valid even when the observations cannot be written using a Markovian (zero-lag) model as in Eq. (23), which was required for the approach developed in [51]. In fact, this result can be interpreted to indicate that zero-lag DMD is most effective if the observation operator is invertible.

4.4 Regularized regression for nonlinear models

Least squares optimization can also be used for identifying nonlinear systems by searching in a linear subspace. In these cases, it is often advantageous to add regularization to seek parsimonious solutions. One such approach that uses a sparsity enhancing regularization is the method of sparse regression or sparse identification of nonlinear dynamics (SINDy) [26].

These approaches organize a library of candidate functions (linear and nonlinear) into a matrix. They then aim to approximate the time derivative, or vector field, in the span of this library. For instance,

$$\dot{x} = f(x) \approx \begin{bmatrix} 1 & x & x^2 & \dots & x^p \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_p \end{bmatrix}. \tag{31}$$

This example uses monomial candidate functions, but any basis (wavelets, orthogonal polynomials, empirical bases) can be used.

Suppose that the general dictionary of terms is given by $\Xi : \mathbb{R}^d \to \mathbb{R}^d$ so that the deterministic portion of some continuous-time autonomous dynamics can be written as a linear system with respect to the parameters/coefficients of the functions in the dictionary $\dot{x} = \Xi(x)\theta_\Psi$. If direct data were available on the states and derivatives, one might then try to solve a (regularized) linear least squares problem for the parameters

$$\theta_\Psi = \arg\min_{\tilde{\theta}} \sum_{k=1}^{n} \|\dot{x}_k - \Xi(x_k)\tilde{\theta}\|_2^2 + \lambda\|\tilde{\theta}\|, \tag{32}$$

where λ is a regularization weight and the norm can be chosen by the user. If the L_1 norm is chosen, this becomes a sparse regression problem.

Practical applications, however, do not have data on the derivative of each state $\dot{x}_i$. As a result, various numerical approximations can be made, and this is the approach taken by the SINDy algorithm. Here, we will consider one type of numerical approximation to the derivative, but our analysis can be extended to others. If a forward-difference approximation to the time derivative is taken, then the SINDy objective function is

$$\theta_\Psi = \arg\min_{\tilde{\theta}} \sum_{k=2}^{n} \left\| \frac{y_k - y_{k-1}}{\Delta t} - \Xi(y_{k-1})\tilde{\theta} \right\|_2^2 + \lambda\|\tilde{\theta}\|_1. \tag{33}$$

Notice that this approach requires direct observation of the states. Next, we show that it is also equivalent to the maximum *a posteriori* (MAP) of our target conditional distribution under more strict assumptions.

Theorem 5 (SINDy as a maximum *a posteriori* estimate of system (6)) *Let $\Xi(x) : \mathbb{R}^d \to \mathbb{R}^d$ denote a library of candidate functions for continuous-time drift*

dynamics. Let $\Psi(x;\theta_\Psi)$ denote the resulting discrete-time operator that uses a forward-Euler integration scheme

$$\Psi(X,\theta_\Psi) = X + \Delta t\,\Xi(X)\theta_\Psi. \tag{34}$$

Furthermore, assume an identity observation operator $h = I$; noiseless measurements $\Gamma(\theta_\Gamma) = 0$; identity process noise $\Sigma(\theta_\Sigma) = I$; and a Laplace prior $p(\theta_\Psi) \propto \exp\left(-\tilde{\lambda}|\theta_\Psi|\right)$. Then, the MAP estimate of the conditional distribution given in Eq. (9) is equivalent to the SINDy estimator obtained by minimizing (33).

Proof This proof is again a straightforward application of Theorem 3. Recall that the data are taken on the initial condition, and note that we have $Y_k = X_k$. The log marginal likelihood (26) is then

$$\log \mathscr{L}(\theta;\mathscr{Y}_n) = \sum_{k=2}^{n}\Big(-\frac{1}{2}\|y_k - (y_{k-1} + \Delta t\,\Xi(y_{k-1})\theta_\Psi)\|_{I_d}^2 + \log|I_d|\Big) - \frac{nd}{2}\log 2\pi - \frac{n}{2}\log|I_d| \tag{35}$$

$$= -\frac{\Delta t}{2}\sum_{k=2}^{n}\left\|\frac{y_k - y_{k-1}}{\Delta t} - \Xi(y_{k-1})\theta_\Psi\right\|_2^2 - \frac{nd}{2}\log 2\pi. \tag{36}$$

Then, we can drop the parameter-independent term and add the log prior to obtain a posterior that is proportional to

$$\log p(\theta;\mathscr{Y}_n) \propto -\frac{\Delta t}{2}\sum_{k=2}^{n}\left\|\frac{y_k - y_{k-1}}{\Delta t} - \Xi(y_{k-1})\theta_\Psi\right\|_2^2 - \tilde{\lambda}|\theta_\Psi| \tag{37}$$

$$= -\frac{\Delta t}{2}\left(\sum_{k=2}^{n}\left\|\frac{y_k - y_{k-1}}{\Delta t} - \Xi(y_{k-1})\theta_\Psi\right\|_2^2 + \frac{2\tilde{\lambda}}{\Delta t}|\theta_\Psi|\right). \tag{38}$$

Maximizing the posterior is equivalent to minimizing the term in the parentheses. By setting $\lambda \equiv \frac{2\tilde{\lambda}}{\Delta t}$, we see that this is the exact form of the SINDy objective (33).□

5 Algorithm and computational complexity

In this section, we describe an approximate marginal MCMC approach that has recently been introduced and analyzed in parallel by several different fields [27,29–31]. This approach is fundamentally based on approximately evaluating the marginal likelihood described in Theorem 1.

5.1 Algorithm

Theorem 1 provides a recursive approach to evaluate the marginal likelihood that avoids computation of a high-dimensional integral, but this theorem still requires the evaluation of lower-dimensional integrals. In the linear case, the solution to these recursive integrals can be found using the Kalman filter; however, no solution is available for general nonlinear systems.

When no closed-form solution exists for these integrals, nonlinear filtering techniques can be introduced. These can include ensemble Kalman filtering [52], Gaussian filtering (including cubature Kalman filter [53] and unscented Kalman filter [54]), and particle filtering [55]. Of these filters, only the particle filter has been proven to enable an exact pseudo-marginal MCMC scheme [56]. The other schemes approximate the prediction, update, and marginalization equations—yielding a (generally) biased estimate of the posterior. Nevertheless, they are often more computationally tractable and have empirically shown good performance.

These algorithms embed these filters within the accept–reject step of Metropolis–Hastings MCMC scheme, as shown in Algorithm 1. We slightly modify the UKF-MCMC scheme of [27] by using delayed-rejection adaptive Metropolis MCMC [57] instead of the standard Metropolis–Hastings MCMC. Specifically, the log posterior enters these schemes during the computation of the likelihood portion of the posterior

$$\alpha = \min\left(1, \frac{\hat{\mathscr{L}}(\theta^*;\mathscr{Y}_n)p(\theta^*)}{\hat{\mathscr{L}}(\theta^{(k-1)};\mathscr{Y}_n)p(\theta^{(k-1)})}\frac{\pi(\theta^{(k-1)})}{\pi(\theta^*)}\right), \tag{39}$$

where $\pi(\theta)$ is the proposal distribution and $\hat{\mathscr{L}}(\theta;\mathscr{Y}_n)$ is the likelihood estimator. As we mentioned above, in the linear case we use a Kalman filter to exactly evaluate the marginal likelihood ($\hat{\mathscr{L}}(\theta;\mathscr{Y}_n) \equiv \mathscr{L}(\theta;\mathscr{Y}_n)$). This algorithm is shown in Algorithm 2. In the nonlinear case, we approximate each distribution to be Gaussian and approximate the marginal posterior

using an unscented Kalman filter (UKF) as shown in Algorithm 3. Algorithms 2 and 3 are given in the "Appendix".

Algorithm 1 Approximate marginal MCMC for Bayesian inference

Input: Prior distribution $p(\theta)$
UKF-based likelihood estimator $\hat{\mathscr{L}}(\theta; \mathscr{Y}_n)$
Proposal distribution $\pi(\theta)$
Initial sample $\theta^{(0)}$
Output: Samples from stationary distribution $p(\theta \mid \mathscr{Y}_n)$
1: Compute $\hat{z}^{(0)} = \hat{\mathscr{L}}(\theta^{(0)}; \mathscr{Y}_n)$
2: **for** $k = 1$ to N **do**
3: $\theta^* \sim \pi$ Sample from proposal
4: $z^* = \hat{\mathscr{L}}(\theta^*; \mathscr{Y}_n)$ Compute estimated likelihood
5: Compute acceptance probability

$$\alpha = \min\left(1, \frac{z^* p(\theta^*)}{z^{(k-1)} p(\theta^{(k-1)})} \frac{\pi(\theta^{(k-1)})}{\pi(\theta^*)}\right) \tag{40}$$

6: Accept $\theta^{(k)} = \theta^*$ and $z^{(k)} = z^*$ with probability α; otherwise $\theta^{(k)} = \theta^{(k-1)}$ and $z^{(k)} = z^{(k-1)}$
7: **end for**

5.2 Computational complexity

We will show in Sect. 6 that this approach yields more robust estimators than competing system ID approaches by accounting for measurement noise; however, this robustness will be at the cost of slightly increased computational complexity. In this section, we assess the cost of the algorithm both in the linear case where the Kalman filter is used and the nonlinear case where the UKF is used by counting the number of floating-point operations (flops) required by each algorithm.

For this analysis, addition, subtraction, multiplication, and division of two floating point numbers and the logarithm of one floating point number all count as one flop. The multiplication of an $m \times n$ matrix by an $n \times p$ matrix then counts as $mp(2n-1)$ flops because each of the mp entries of the product matrix requires n multiplications and $n-1$ additions.[2] Similarly, the multiplication of an $m \times n$ matrix by an $n \times 1$ vector requires $n(2n-1)$ flops. Additionally, we approximate the cost of a Cholesky decomposition, matrix inversion, and determinant performed on an $n \times n$ matrix all to be $n^3/3$ flops. Furthermore, the complexity of these algorithms strongly depends on the complexity of the dynamical and measurement models used, which will vary from problem to problem. For the sake of generality, we define the computational complexity of the dynamical model Ψ and measurement model h to be denoted as F and H, respectively. Clearly in the linear case, these variables will not be needed as the dynamical and measurement models are matrices, and the number of flops can be calculated without loss of generality. The number of flops for each algorithm will be given in terms of the problem dimensions, so recall the following notation: d the dimension of the state, m the dimension of the measurements, p the number of parameters, and n the total number of measurements available.

Our analysis focuses entirely on the computation of the marginal likelihood, which is the dominant cost of the MCMC algorithm. The complexity of the rest of the algorithm will depend on the complexity of the MCMC algorithm and prior selected by the user, but is typically orders of magnitude lower than the likelihood computation. In the following analysis, we provide results for the Kalman filtering algorithm, the unscented Kalman filtering algorithm, their prediction and update subcomponents, DMD, and sparse regression. Table 1 shows the number of different types of operations and flops required by each algorithm where the computation of the regularization term in sparse regression is excluded. Note that although the mean and covariance of the marginal likelihood are computed in the update step of the Bayesian algorithms, the computation of the log of this distribution is excluded from this step, and is instead included only in the total. Also, the 18 flops outside the parentheses in the UKF total count comes from the formation of the weights, which is required only once at the beginning of the algorithm.

In determining the number of flops used in DMD, we counted the number of flops needed to solve the normal equation $A = Y'Y^T(YY^T)^{-1}$ where $Y, Y' \in \mathbb{R}^{m \times n-1}$. Similarly, sparse regression was considered to be the computation $\Theta = (\Xi^T \Xi)^{-1} \Xi^T \dot{X}$, where $\Xi \in \mathbb{R}^{n-1 \times p/m}$ and $\dot{X} \in \mathbb{R}^{n-1 \times m}$. In practice, this computation is performed multiple times with an increasingly small Ξ matrix, but for this analysis, only one iteration of the optimization procedure is considered. To execute TDMD, a singular value decomposition (SVD)

[2] We only consider the naive matrix-multiplication scheme, not the asymptotically more optimal approaches such as Strassen's algorithm.

Table 1 Tally of matrix and vector operations and flop count of Algorithms 2, 3, DMD, and SINDy. VEW and MEW are element-wise vector and matrix operations, respectively, such as addition, subtraction, and element-wise multiplication and division. MV is a matrix–vector or vector–vector multiplication, and MM is matrix–matrix multiplication. Inv is a matrix inversion, Det a determinant, and Chol a Cholesky decomposition

Algorithm	VEW	MEW	MV	MM	Inv	Det	Chol	Flop count
KF prediction	0	1	1	2	0	0	0	$4d^3 + d^2 - d$
KF update	2	2	3	6	1	0	0	$2d^3 + \frac{1}{3}m^3 + 6d^2m + 4dm^2 + -d^2 - m^2 + 3dm - 1$
KF total	$4n$	$3n$	$6n$	$8n$	$2n$	n	0	$n(6d^3 + m^3 + 6d^2m + 4dm^2 + m^2 + 3dm - d + 3m + 8)$
UKF prediction	$4d$	8	0	1	0	0	1	$\frac{13}{3}d^3 + 17d^2 + 4d + 2 + (2d+1)F$
UKF update	$4d+2$	14	1	5	1	0	1	$\frac{1}{3}d^3 + \frac{1}{3}m^3 + 6d^2m + 8dm^2 + 9d^2 + 4m^2 + 13dm + 2d + 6m + 2 + (2d+1)H$
UKF Total	$(8d+4)n$	$22n$	$3n$	$6n$	$2n$	n	$2n$	$n\left(\frac{14}{3}d^3 + m^3 + 6d^2m + 8dm^2 + 26d^2 + 6m^2 + 13dm + 6d + 9m + 13 + (2d+1)(F+H)\right) + 18$
DMD	0	0	0	3	1	0	0	$\frac{7}{3}m^3 + 4m^2n - 7m^2$
Sparse Regression	0	0	0	3	1	0	0	$\frac{1}{3}\frac{p^3}{m^3} + 4\frac{p^2n}{m^2} - 5\frac{p^2}{m^2} - \frac{pn}{m} + 2pn + \frac{p}{m} - 3p$

of the concatenated matrix $\begin{bmatrix} Y^T & Y'^T \end{bmatrix} \in \mathbb{R}^{n-1\times 2m}$ is first performed, which has computational complexity on the order of $\mathcal{O}(m^2n + n^2m + m^3)$. The solution of the total least squares problem is then given by $A = -V_1V_2^T(V_2V_2^T)^{-1}$. Let r be the rank of matrix $\begin{bmatrix} Y^T & Y'^T \end{bmatrix}$. Then, $V_1 \in \mathbb{R}^{m\times 2m-r}$ is a matrix composed of the first m rows of the last $2m - r$ right singular vectors, and $V_2 \in \mathbb{R}^{m\times 2m-r}$ is a matrix composed of the last m rows of the last $2m - r$ right singular vectors. The computational complexity of this least squares problem is then $\frac{19}{3}m^3 - 2m^2r - 2m^2$. Since $m = d$ in the case of DMD the total computational complexity is on the order $\mathcal{O}(d^3 + d^2n + n^2d)$. Thus, the added cost of including measurement noise is on the order $\mathcal{O}(n^2d)$.

The computational costs of the Bayesian algorithms are on the order $\mathcal{O}(n(d^3 + m^3))$. Typically, the dimension m of the observations is small, so this algorithm is primarily limited by the dimension d of the state vector. Furthermore, the dimension p of the parameter vector only affects the evaluation of the prior, which is usually chosen so as to be easy to compute. Therefore, this algorithm is most efficient for problems where the state dimension is low and the parameter dimension is high, such as in nonlinear regression problems.

6 Numerical experiments

In this section, we provide a set of empirical results that demonstrate a lack of robustness among methods that do not account for all three sources of uncertainty. We then show that our proposed approach is able to perform well under a greater variety of experimental conditions. The conditions of each experiment are designed to highlight and exaggerate the specified limitation of some specific methods. We will show that in many cases only small changes to the setting, for instance a slightly larger noise or slower sampling frequency, can yield significant difference in learning with these existing methods—demonstrating their lack of robustness.

Our evaluations of the methodology examine two quantities: reconstruction errors and prediction/forecasting errors. Reconstruction error compares how well the learned parameter is able to match the trajectory from which the data were generated. This is essentially training error, and used more to verify that the algorithms are working properly. Prediction/forecasting compares our estimate to some trajectory that is not contained in the data. These trajectories could be a continuation of the system into the future from the last point at which data were taken, or it could be starting the estimated dynamics at a different initial condition. This comparison is of greater interest

because it tests the extrapolatory power of the learned dynamics.

6.1 Evaluation and prediction

Whereas traditional system ID and ML approaches define the problem through an optimization objective, the Bayesian approach separates learning and decision making. In effect, it provides a way of generating new optimization objectives and interpreting existing ones. Here, we briefly comment on the fact that this separation comes in the form of a two-step procedure: (1) computing the posterior and (2) extracting a goal-oriented estimator through the specification of a loss function. For detailed discussion of these topics, we refer the reader to [25].

First note that we have considered θ to contain all uncertain parameters in the problem. For prediction, however, it is standard to make predictions into the future using deterministic models based on Ψ. As a result, we can partition the parameters $\theta = (\theta_\Psi, \theta_h, \theta_\Sigma, \theta_\Gamma)$ into those that correspond to the dynamics, observations, process noise, and measurement noise, respectively. Next, we define the *posterior predictive* distribution of the states as an average over all possible values of the dynamics parameters conditioned on the observations

$$p(X_k \mid \mathscr{Y}_n) = \int p(X_k \mid \theta_\Psi) p(\theta_\Psi \mid \mathscr{Y}_n) d\theta_\Psi, \tag{41}$$

where we will use a deterministic prediction that discards the process noise

$$p(X_k \mid \theta_\Psi) = \delta_{\Psi^k(x_0,\theta_\Psi)}(X_k). \tag{42}$$

This restriction is not explicitly necessary, but it is representative of how learned models are used in practice.

Finally, we extract an estimator to use as the "point estimate" from the posterior. In this paper, unless otherwise specified, we use the mean estimator, which corresponds to the optimal estimator for the squared loss [25],

$$X_k^{\text{avg}} = \mathbb{E}_{\theta_\Psi|\mathscr{Y}_n}\left[p(X_k \mid \mathscr{Y}_n)\right]. \tag{43}$$

Additionally, we require a point estimate as a starting point for our sampling. In MCMC sampling, it is good practice to start the sampler at a high probability point to reduce the convergence time. For this reason, we select the MAP estimate of the parameter posterior

$$\theta^{\text{map}} = \arg\max_\theta p(\theta|\mathscr{Y}_n). \tag{44}$$

6.2 Algorithmic settings

To perform the following experiments, MATLAB 2019b was used. For our MCMC algorithm, we selected the delayed-rejection adaptive Metropolis (DRAM) algorithm [57]. The tuning parameters of this algorithm are n_0 the number of samples to draw before beginning the AM algorithm, and γ the scaling factor used by DR to scale the second-tier proposal covariance. In this paper, we used $n_0 = 200$ and $\gamma = 0.01$ for each experiment. Also throughout the algorithm, whenever a covariance matrix was calculated, a nugget εI was added where $\varepsilon = 10^{-10}$ to help ensure positive definiteness. Furthermore, the algorithm requires selection of a starting sample and initial proposal covariance; we used the MAP point $\theta^{\theta\text{-map}}$ as our initial sample $\theta^{(0)}$, and the inverse Hessian of the negative log posterior evaluated at $\theta^{\theta\text{-map}}$ to be the initial covariance of our proposal distribution:

$$\pi_0(\theta) = \mathscr{N}\left(\theta^{\text{map}}, \left(-\frac{\partial^2 \log p(\theta^{\text{map}}; \theta|\mathscr{Y}_n)}{\partial \theta^2}\right)^{-1}\right). \tag{45}$$

Both of these values were found using MATLAB's `fminunc` function. For nonlinear systems, we must additionally select parameters α, κ, and β for the UKF. In this paper, we followed a common choice of parameter selection where $\alpha = 10^{-3}$, $\kappa = 0$, and $\beta = 1$.

Unless otherwise specified, an improper uniform prior is used for all dynamics model parameters $\theta \in \theta_\Psi$

$$p(\theta) = \mathscr{U}(-\infty, \infty), \tag{46}$$

and half-normal priors are specified on the variance parameters $\theta \in \theta_\Sigma$ and $\theta \in \theta_\Gamma$ as suggested in [33]

$$p(\theta) = \text{half-}\mathscr{N}(0, 1). \tag{47}$$

The code used to implement the Bayesian algorithms can be found on the author's GitHub https://github.com/ngalioto/BayesID. To execute DMD, MATLAB's right matrix division operator "/" was used, which returns the least squares solution. TDMD was performed using a script taken from MATLAB file exchange [58] that solves the total least squares problem. Lastly, SINDy was run using code from [26],

which utilizes code from [35] to compute the total variation regularized derivatives.

6.3 Linear pendulum, linear model

In this section, we consider learning a linear model under an identity observation operator $h = I$ when the truth model is also linear. We show that that the proposed probabilistic approach is more robust to sparse observations and measurement noise than the least squares-based DMD and TDMD.

Consider the linear model (5) for which the exact propagator is

$$x_k = \exp\left(\begin{bmatrix} 0 & 1 \\ -\frac{g}{L} & 0 \end{bmatrix} \Delta t\right) x_{k-1}, \quad x_0 = \begin{bmatrix} 0.1 \\ -0.5 \end{bmatrix} \tag{48}$$

where $g = 9.81$ is the acceleration due to gravity and $L = 1$ is the length of the pendulum.

We are learning an unknown linear model $A(\theta_\Psi)$ and assume that the process noise and measurement noise is also uncertain. Under this setting, System (6) becomes

$$\begin{aligned} x_k &= A(\theta_\Psi)x_{k-1} + \xi_k, \quad \xi_k \sim \mathcal{N}(0, \Sigma(\theta_\Sigma)) \\ y_k &= x_k + \eta_k, \quad \eta_k \sim \mathcal{N}(0, \Gamma(\theta_\Gamma)), \end{aligned} \tag{49}$$

for $k = 1, \ldots, n$ where

$$A(\theta_\Psi) = \begin{bmatrix} \theta_1 & \theta_2 \\ \theta_3 & \theta_4 \end{bmatrix}, \ \Sigma(\theta_\Sigma) = \theta_5 I_{2\times 2}, \ \Gamma(\theta_\Gamma) = \theta_6 I_{2\times 2}. \tag{50}$$

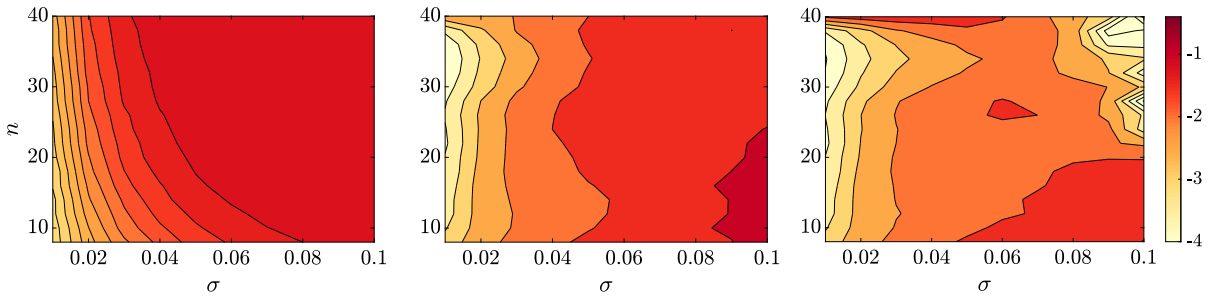
Because this setup is precisely the one corresponding to DMD, we seek to compare the performance of our approach to DMD and TDMD. Our comparison takes the form of average performance over 500 different realizations of the data sets for different combinations of training data sizes n and true measurement noise standard deviation σ. The data points are spread out over a simulation period of 4 s, so increasing n indicates increasing density of data per time.

The results, shown in Fig. 4, provide (log base 10) ratios of the expected error of the posterior predictive mean (computed with 1000 posterior samples) to the (T)DMD estimators. The squared errors were calculated only at the times of observations, and the largest MSE from each data set for each algorithm was discarded to prevent biasing from outliers. We see that the biggest gains in using the probabilistic Bayesian approach come in the low-noise regime. At first, this seems surprising, but in the low-noise regime, this is likely the result of the scale of the errors being so small. As the noise increases, we see the ratio increasing even though we'd expect DMD to break down much more quickly than the Bayesian approach. The reason this occurs is because DMD predictions decay to zero after a certain level of noise (shown in Fig. 4a), effectively placing an upper bound on the MSE of the algorithm. Regardless, the contour plots show that the Bayesian algorithm outperforms both DMD and TDMD at every measurement frequency and noise pair considered.

Next, we provide a detailed look at two specific points on these contour plots to demonstrate the mechanism by which DMD/TDMD decline. The first case is a low-noise/sparse-data case of $\sigma = 10^{-2}$ and $n = 8$, and the second case is for a higher-noise case $\sigma = 10^{-1}$ with more data $n = 40$.

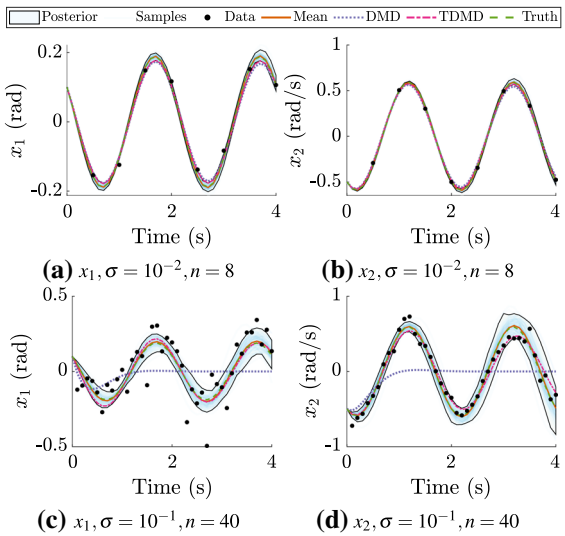
The reconstruction results for each state are compared in Fig. 5. The prediction (forecasting) results for just the second state are shown in Fig. 6. The mean here refers to the posterior predictive mean given in (43). The shaded area represents the region between the 97.5th and 2.5th quantiles of the Bayesian posterior. In the low-noise case, we see that all three algorithms perform essentially equally—though the DMD-based approaches slightly underestimate the amplitude. In other words, even in the case for which DMD was designed to perform well, the Bayesian approach performs slightly better. In the high-noise case, we see that the TDMD predictions become completely out of phase with increasingly small amplitude, and the DMD estimator smooths out the data too much and rapidly converges to zero. Not only does the Bayesian approach provide the most accurate estimate, but it also gives a quantification of the certainty of its estimate in the form of its posterior, which (T)DMD is unable to provide.

Figure 7 shows the estimated eigenvalues of the system by the Bayesian and (T)DMD algorithms. In the low-noise case, Fig. 7a shows that the Bayesian approach is slightly more accurate than the (T)DMD approaches, though they all perform well. For the high-noise case, Fig. 7b shows that DMD is unable to provide a reasonable estimate of the eigenvalues. TDMD gives a close estimate, but the estimated eigenvalues are too far in the left-hand plane, causing the gradual



(a) DMD Prediction MSE **(b)** Bayes/DMD Prediction **(c)** Bayes/TDMD Prediction

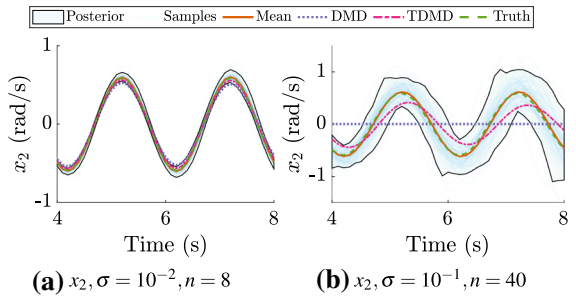
Fig. 4 Log base 10 ratio of the MSE obtained by the proposed Bayesian approach to that obtained by (T)DMD for the linear pendulum model. In all cases, this value is less than zero signifying that our proposed approach outperforms (T)DMD in all cases considered. Also observe in the high-noise regime, TDMD can begin to lose stability



(a) $x_1, \sigma = 10^{-2}, n = 8$ **(b)** $x_2, \sigma = 10^{-2}, n = 8$

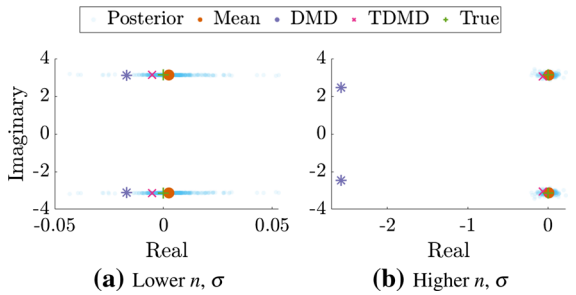
(c) $x_1, \sigma = 10^{-1}, n = 40$ **(d)** $x_2, \sigma = 10^{-1}, n = 40$

Fig. 5 Comparison of reconstruction error among the Bayesian and (T)DMD algorithms for the linear pendulum truth model. Top row corresponds to a low-noise/sparse-data case and the bottom row corresponds to a high-noise/dense-data case. Left column corresponds to the first state (angular position) and right column corresponds to the second state (angular velocity). For low-noise, the algorithms perform similarly; however, the (T)DMD approaches underestimate the amplitude. For the high-noise case, DMD fails and TDMD misfits the amplitude. The Bayesian approach is able to recognize greater uncertainty for the high-noise case



(a) $x_2, \sigma = 10^{-2}, n = 8$ **(b)** $x_2, \sigma = 10^{-1}, n = 40$

Fig. 6 Comparison of prediction error among the Bayesian and (T)DMD algorithms for the linear pendulum truth model. Left panel corresponds to a low-noise/sparse-data case and the right panel corresponds to a high-noise/dense-data case. Both panels show the angular velocity of the pendulum. For low-noise, the algorithms perform similarly. For the high-noise case, DMD fails and TDMD can be seen to be out of phase and have a smaller amplitude. The Bayesian approach is able to recognize greater uncertainty for the high-noise case

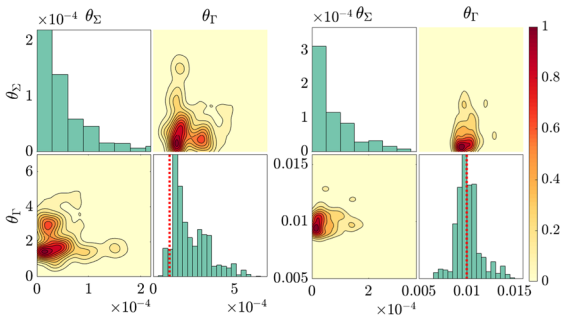


(a) Lower n, σ **(b)** Higher n, σ

Fig. 7 Eigenvalue distributions for the estimators of the linear pendulum. The mean value here represents the mean of the eigenvalues. All three algorithms come very close to learning the true eigenvalues in the low-noise case, but Bayes is able to outperform the other two in both the high- and low-noise cases. DMD achieves significant error when the data are noisy

decay seen in Fig. 6. The Bayesian estimate lies almost exactly on top of the truth.

Finally, Fig. 8 shows the marginal and joint distributions of the process and measurement noise variances for these two cases. The process noise is very close to zero because we are using a linear model for a linear



(a) Lower noise/measurement frequency **(b)** Higher noise/measurement frequency

Fig. 8 Marginal and joint posterior distributions of the process and measurement noise variance parameters during the recovery of the linear pendulum. In the left panel, 8 measurements are not enough for the Bayesian estimator to unambiguously determine the measurement noise, but its best guess (the mode) aligns with the truth. On the right, we see that 40 measurements are enough to define a distinct mode within the joint distribution, which also aligns with the truth

system, and thus the system learns that the dynamics can be captured exactly. These plots also indicate that we have learned the measurement noise, as the mode aligns closely with the true value shown in red. Note also that the joint distribution in this figure shows that the two noise variances are negatively correlated, conveying the fact that the estimator does not yet have enough data to determine if the model is off and the measurements are accurate, or if the model is accurate and the measurements are noisy. As more data come in, however, one of these scenarios can usually be ruled out and the distribution becomes unimodal.

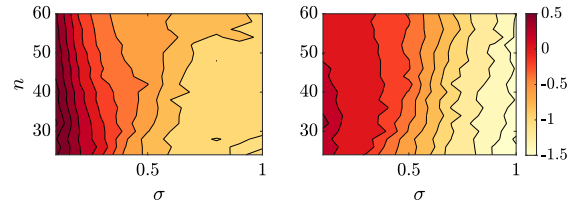
6.4 Nonlinear pendulum, linear model

Next, we consider a problem where the model class within which we are learning does not encompass the true underlying dynamical system. This is the most realistic situation that would be encountered in practice, and avoids the so-called inverse crime [59,60].

Consider a nonlinear pendulum

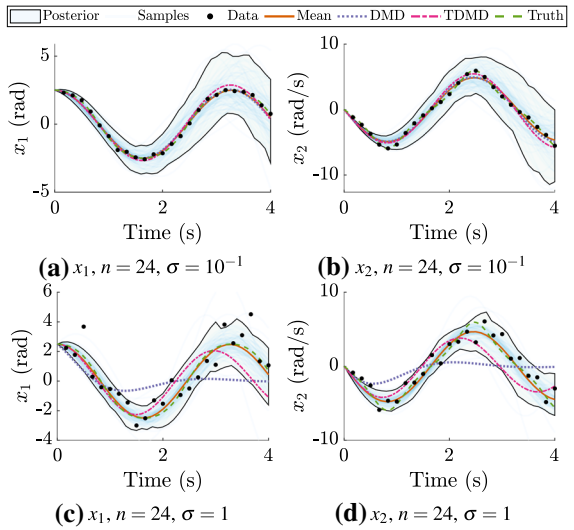
$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix} x_2 \\ -\frac{g}{L}\sin(x_1)\end{bmatrix}, \quad x_0 = \begin{bmatrix}2.5\\ 0\end{bmatrix} \tag{51}$$

to be the truth model. We have changed the initial condition to ensure that we are operating in the nonlinear regime.



(a) Bayes/DMD Prediction Ratio **(b)** Bayes/TDMD Prediction Ratio

Fig. 9 Contours of the ratios from the nonlinear pendulum experiments. The experiment is the same as in Fig. 4. A detailed explanation for the low-noise regime where it appears (T)DMD outperforms Bayes is given in Sect. 6.4.1



(a) x_1, $n = 24$, $\sigma = 10^{-1}$ **(b)** x_2, $n = 24$, $\sigma = 10^{-1}$

(c) x_1, $n = 24$, $\sigma = 1$ **(d)** x_2, $n = 24$, $\sigma = 1$

Fig. 10 Reconstruction performance for low-noise (top row) and high-noise (bottom row) data sets for the nonlinear pendulum using a linear model. All three estimates capture the truth closely in the low-noise case, but only the Bayesian algorithm performs well (it is in phase and approximately the correct amplitude) for the high-noise case

The learning setup is identical to that provided in Sect. 6.3; we learn a linear model, and the same validation experiments are performed. These experimental results are shown in Fig. 9. We are able to clearly see here that, although the three algorithms are comparable in the low-noise regime, the strength of the Bayesian approach increases with the measurement noise. A discussion on why (T)DMD may outperform the mean estimator from the Bayesian approach in the low-noise regime is provided later in Sect. 6.4.1.

We again present more detailed results for two representative cases. Both cases have $n = 24$ data points, but the first case is a low-noise case of $\sigma = 10^{-1}$ and the second case is a higher-noise case of $\sigma = 1$.

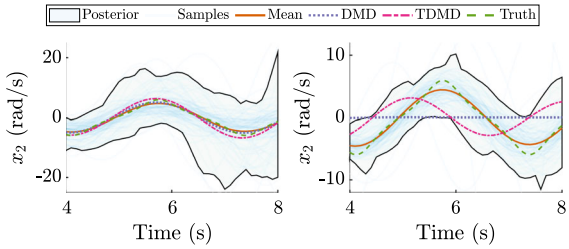


Fig. 11 Comparison shown here is the same as in Fig. 6, but this time for a nonlinear pendulum truth model. In the low-noise case, the estimates are all visually aligned with the truth. In the high-noise case, DMD fails and TDMD falls out of phase, but the Bayesian algorithm remains robust and produces an accurate estimate

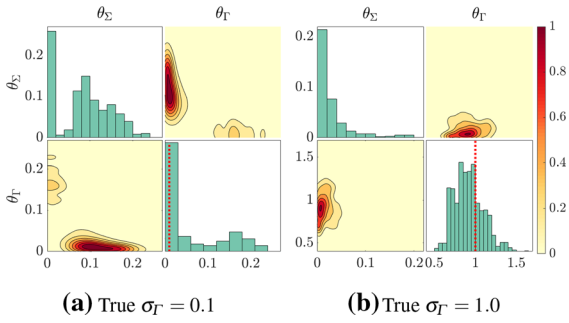


Fig. 12 Marginal and joint posterior distributions of the process and measurement noise variance parameters during the recovery of the nonlinear pendulum. In the left panel, the joint distribution is bimodal, offering two possible models with the true case being strongly preferred. In the right panel, all of the distributions are unimodal and in alignment with the truth

The resulting reconstructions are shown in Fig. 10, and the predictions are given in Fig. 11. Note that the variances of the posterior distributions in both cases grow much more quickly than in either of the linear pendulum examples as a consequence of increased model uncertainty (process noise). The posterior distribution can therefore be used to qualitatively assess not only how informative the data are, but also how appropriate the chosen model is for the system at hand. In the low-noise case, the performances of the three estimates are virtually indistinguishable, once again demonstrating that even in systems that are ideal for (T)DMD, there is no loss of performance when using the Bayesian estimator. In the high-noise case, DMD struggles with noisy measurements and settles on quickly decaying to zero, similar to what we observed in the linear case. TDMD, on the other hand, comes closer but is noticeably out of phase with the truth. The Bayesian approach is able to reconstruct the signal very closely, at least within the constraints imposed by using a linear model.

Next, we investigate what the Bayesian approach learns for the process and measurement noise in the case where there is a model error. The marginal and joint posterior distributions for both measurement noise cases are shown in Fig. 12. We observe that in the low-noise case 12a, the joint distribution is bimodal. The smaller mode corresponds to a model with low process noise and high measurement noise, and the larger mode corresponds to a model with high process noise and low measurement noise. The Bayesian algorithm has effectively uncovered that the data can be explained in one of two ways: either the model fits the true system well, but the data are very noisy, or the measurement noise is low and the model is not capable of properly capturing the dynamics. In this case, the latter is true and is also the option that the Bayesian algorithm found to be much more likely. For the high-noise case 12b, the joint distribution is unimodal, conveying the possibility of only one process-measurement noise pairing. Once again, the modes of both the measurement noise marginal distributions align closely with the truth shown in red. Finally, we see that the process noise magnitudes in both cases are much larger than those seen in the linear pendulum examples (Fig. 8) as a consequence of trying to capture nonlinear dynamics with a linear model.

6.4.1 Discussion on diagnostics

One of the strengths of the Bayesian approach is that it separates the learning stage from the decision-making stage, so if the initial decision rule yields an unsatisfactory estimate, one can go back and analyze the posterior distribution to devise an improved decision rule. It was noted earlier in Fig. 9 that the average MSE of (T)DMD is lower than that of the average MSE of the Bayesian estimator over 500 data sets when the measurement noise is low. This observation likely implies that there is a better decision rule that can be used to achieve performance at least as strong as DMD. To understand how to best select a point from the posterior to be our estimate, we first look at the posterior over the states. Figure 13 shows samples from the posterior predictive distribution for a single data set containing $n = 26$ measurements with noise standard deviation of $\sigma = 0.1$. The mean deviates from the truth near the

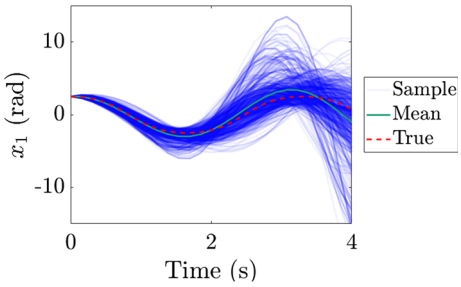


Fig. 13 Posterior samples from the data set with $n = 26$, $\sigma = 0.1$ that produced the worst mean estimate out of the 500 with respect to MSE. This figure illustrates that the mean deviates from the truth at the extrema of the curve where samples are skewed toward larger magnitudes. Using a decision rule that selects the mode here would give a much better estimate

peaks and valleys of the trajectory between about 2.5 and 4 s. This is the same location in which the posterior appears to be significantly spread in possible predictions. This presence of significant outliers is a result of the bimodal noise distribution previously discussed. Furthermore, it is clear that the mean is not a good estimator in the case of bimodal distributions; however, we see that there exists a mode in alignment with the truth. Upon this realization, we can then craft a decision rule that selects this mode rather than the mean for improved performance. In this case, the mode-based rule would result in the Bayesian approach being 1.3 times better than the TDMD estimator. Moreover, this entire analysis can be done *a posteriori*, and therefore uses no additional assumptions or requirements on our approach.

We also note that the effect this has on the MSE ratio appears more strongly in this nonlinear case for two reasons. The first reason is that the higher process noise due to the model error and low measurement noise can create a bimodal distribution because of the alternate possibility of a good model with noisy data as shown earlier. The second reason is that the ratio of process noise to measurement noise is higher than that in the linear case. As we have shown in Theorem 4, the (T)DMD approaches effectively assume the existence of process noise but no measurement noise. In cases where the linear and nonlinear models are mismatched, this becomes a better assumption.

In summary, for cases where the model error can be significant, a non-mean estimator should be extracted from the Bayesian posterior. This estimator should be chosen by considering the bimodality of the learned process/measurement noise estimator, and can often be the peak of one of the modes. If this is done (it is an *a posteriori* procedure), we have seen that it yields improved performance compared to (T)DMD.

6.5 Optimal estimators and the Van der Pol oscillator

Next, we consider learning a sparse representation of a nonlinear system so that we can compare the Bayesian algorithm directly to SINDy. Here, it will once again be shown that factoring the process and measurement noise into our estimator will allow it to be robust even for noisy measurements.

Consider the nonlinear Van der Pol oscillator

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}x_2\\ \mu(1-x_1^2)x_2 - x_1\end{bmatrix}, \quad x_0 = \begin{bmatrix}0\\ 2\end{bmatrix}, \tag{52}$$

where $\mu = 3$. In this case, we use the SINDy algorithm rather than DMD to account for the nonlinear dynamics. For both the Bayesian and SINDy algorithms we therefore consider a subspace of right hand sides that is spanned by a set of candidate functions. We choose monomial candidates up to third degree and their interacting terms. As a result, each algorithm seeks to learn 20 dynamics parameters (10 for each state). The Bayesian algorithm is additionally tasked with learning the covariance matrices parameterized as follows:

$$\Sigma(\theta_\Sigma) = \begin{bmatrix}\theta_{21} & 0\\ 0 & \theta_{22}\end{bmatrix}, \quad \Gamma(\theta_\Gamma) = \theta_{23} I_{2\times 2}. \tag{53}$$

The priors on the dynamics parameters are Laplace distributions with zero mean and on the variance parameters are once again half-normal distributions.

We consider two cases: one where SINDy shows strong performance, and one in which SINDy struggles, and we show that the Bayesian algorithm yields an accurate estimate in both cases. The case in which SINDy excels is frequent and low-noise data. Here, $n = 2000$ measurements were taken over the course of 20 s with measurement noise standard deviation of $\sigma = 10^{-3}$. In the opposite case, we collect only $n = 200$ measurements over 20 s with measurement noise standard deviation of $\sigma = 2.5 \times 10^{-1}$.

The reconstructions from these experiments are shown in Fig. 14, predictions are given in Fig. 15, and the phase plots over 200 s are given in Fig. 16.

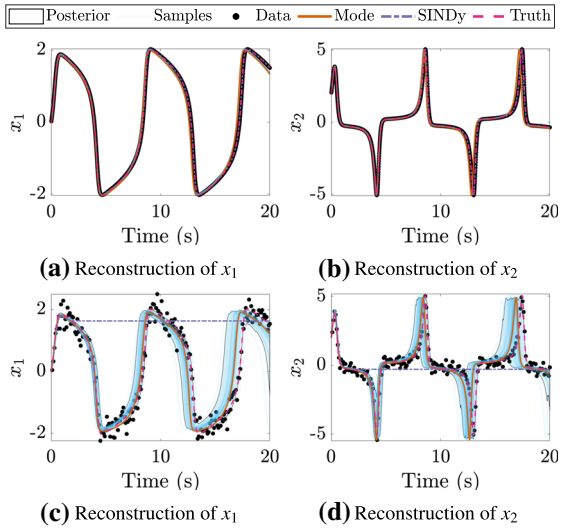


Fig. 14 Comparison of reconstruction error among the Bayesian and SINDy algorithms for the Van der Pol system. Top row corresponds to a low-noise/dense-data case, and the bottom row corresponds to a high-noise/sparse-data case. Left column corresponds to the first state (position), and right column corresponds to the second state (velocity). The Bayesian estimator is able to accurately reconstruct the dynamics, even in the presence of high noise

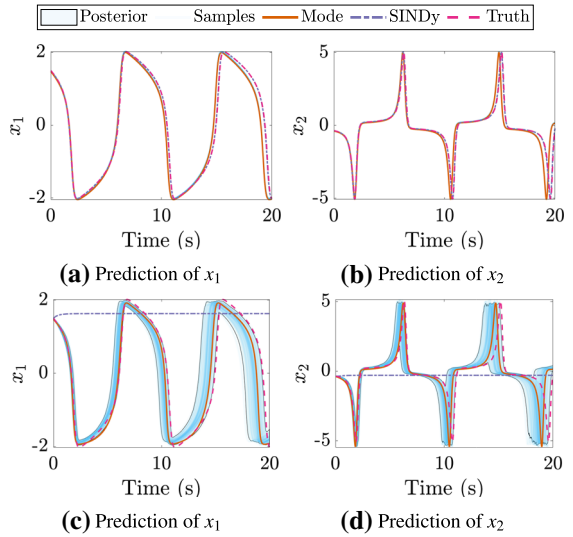


Fig. 15 Comparison of prediction error among the Bayesian and SINDy algorithms for the Van der Pol system. The meaning of the figures is the same as described in Fig. 14. The model learned by the Bayesian estimator is still accurate at a different initial condition

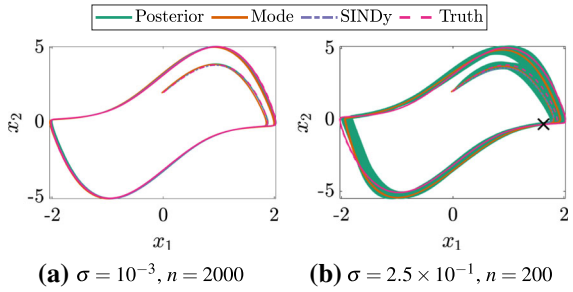


Fig. 16 Phase-diagram reconstruction for the Van der Pol oscillator under the two indicated data conditions. In the low-noise and frequent data domain, both the Bayesian and SINDy estimates lie directly on the truth. In the high-noise case, the Bayesian posterior is wider, but is still visually aligned with the truth. The SINDy estimate is unable to recover the limit cycle, and the large "x" marks the equilibrium point to which SINDy converges, as shown in Fig. 15

Here, the mode represents the mode of the posterior predictive distribution. In the low-noise case, we see that the Bayesian algorithm and SINDy both capture the dynamics very closely. We see that SINDy agrees slightly more closely with the trajectory as a result of its hard threshold regularization. Note that the posterior in this case is very small because the high number of data points and low measurement noise gives us high certainty in our estimate. In the high-noise case, we see that SINDy gives a similar result to what DMD gave when the measurements were noisy: the trajectory immediately flatlines. When the data are noisy like this, the procedure for SINDy is to denoise the data using total variation (TV) regularization [35] before executing the algorithm. However, the increased timestep between data makes it difficult to accurately denoise the data, and when the TV regularization is performed, SINDy ends up giving an unstable estimate. The Bayesian approach, however, is still able to identify the dynamics of the Van der Pol system. The posterior in this high-noise case is wider, signifying that the estimate holds more uncertainty than the low-noise and frequent measurements case.

6.6 Known model form

Finally, we consider the case where the model form is known, for instance from physical laws, but the parameters are uncertain. This is the classical inference setting and has seen a lot of development [61–68], including in the computational physics community. However, much of this literature either only considers deterministic dynamics according to some variation of Eq. (2) or

only static problems. In this section, we consider both a chaotic system and a reaction–diffusion PDE in which we impose process noise to aid in the parameter estimation. For the reaction–diffusion PDE, this implies that the process noise is added to the discretized dynamics. Our results suggest that these methods are applicable to spatial problems and are able to effectively learn chaotic dynamics with a much smaller amount of data than observed in the literature.

6.6.1 Lorenz '63

We first consider the chaotic Lorenz '63 system [69]

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix} = \begin{bmatrix}\theta_1(x_2 - x_1)\\ x_1(\theta_2 - x_3) - x_2\\ x_1 x_2 - \theta_3 x_3\end{bmatrix}, \quad x_0 = \begin{bmatrix}2.0181\\ 3.5065\\ 11.8044\end{bmatrix}. \tag{54}$$

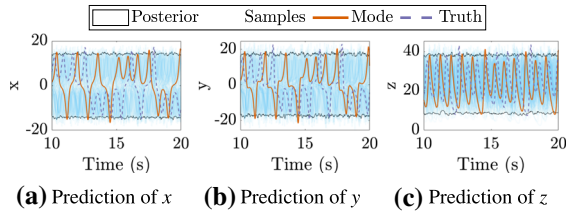
The initial condition of this system was chosen so as to sit on the attractor. We attempt only to learn the parameters $\theta_\Psi = (\theta_1, \theta_2, \theta_3)$. The difficulty with learning in chaotic systems is that the computation of the likelihood can be challenging. Since the likelihood involves running a filter, and filtering chaotic systems is well known to be challenging, it may seem that our approach would breakdown. Here, we show that our Gaussian filtering approach is still able to learn an approximate dynamical system without resorting to more complicated likelihood building processes, e.g., using correlation integrals [28,70].

The priors on the dynamics parameters are once again improper and uniform. In addition to learning the model parameters in this example, we also learn the process noise variance for each state and the measurement noise variance for a total of seven parameters. The parameterizations of the covariance matrices are shown:

$$\Sigma(\theta_\Sigma) = \begin{bmatrix}\theta_4 & 0 & 0\\ 0 & \theta_5 & 0\\ 0 & 0 & \theta_6\end{bmatrix}, \quad \Gamma(\theta_\Gamma) = \theta_7 I_{3\times 3}, \tag{55}$$

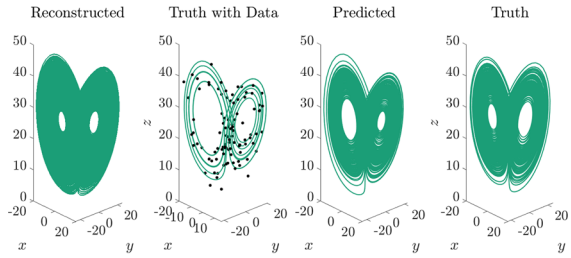
with half-normal priors as before.

One hundred data points uniformly spaced over 10 s are collected with a true measurement noise standard deviation of 2.0. The predicted state trajectories after 10 s of simulation using the parameter posterior mode



(a) Prediction of x **(b)** Prediction of y **(c)** Prediction of z

Fig. 17 Lorenz '63 prediction posteriors. Although the trajectories become misaligned rather quickly due to the chaotic nature of the system, the posterior phase diagram (Fig. 18) reveals that the algorithm has discovered that the dynamics exist on a low-dimensional attractor

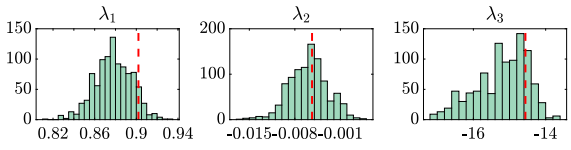


(a) Reconstruction **(b)** Prediction

Fig. 18 Reconstruction and prediction of the Lorenz '63 attractor. The right panel compares the predicted and true trajectories up to 200 s using the mode of the parameter posterior distribution. The proposed approach is able to successfully discover the Lorenz attractor from sparse, noisy data

are shown in Fig. 17. Similar to the Van der Pol oscillator, the dynamics exist on a low-dimensional attractor in phase space, and the wide, but constant, posterior distribution once again reflects this fact. Figure 18 shows the reconstructed and predicted attractors from the Bayesian algorithm. These figures show that while we cannot accurately capture the state, indeed all methods would eventually break down due to the chaotic nature of the system, we do predict a qualitatively similar attractor. As such, one would expect that most post-processing of these attractors, e.g., for control, would yield similar results.

Similar to the eigenvalues of the linear pendulum, the samples collected here can be used to investigate the probability distribution of any dynamical quantity of interest. Here, we show the estimation of the three Lyapunov exponents of the Lorenz system in Fig. 19. Lyapunov exponents are a measure of the exponential growth rates of generic perturbations of a system. A positive Lyapunov exponent implies that an arbitrarily small perturbation will grow exponentially large over time. Such systems are considered to be chaotic [71].



(a) λ_1 Estimate **(b)** λ_2 Estimate **(c)** λ_3 Estimate

Fig. 19 Posterior of the Lyapunov exponent estimation of the Lorenz '63 system. The distribution of λ_3 is wider than the other two because the behavior of the system is dominated by the first two exponents, making the third difficult to estimate with high certainty

Here, the Lyapunov exponents are computed using a function from MATLAB file exchange [72] that uses the algorithm proposed in [73]. The red line denotes the approximated value of the Lorenz system's Lyapunov exponents using the truth values of the parameters. When approximating the Lyapunov exponents of a system, the growth of the initial perturbation is dominated by the largest Lyapunov exponents, making the smaller Lyapunov exponents more difficult to estimate precisely. This fact is reflected in the distribution of λ_3, which is much wider than the other two. We see that for each exponent, the true value is contained in its respective distribution at a relatively high probability value.

6.6.2 Reaction diffusion

In the final example, we consider both a PDE and a case where the measurement operator h is not the identity. The reaction diffusion PDE is given by

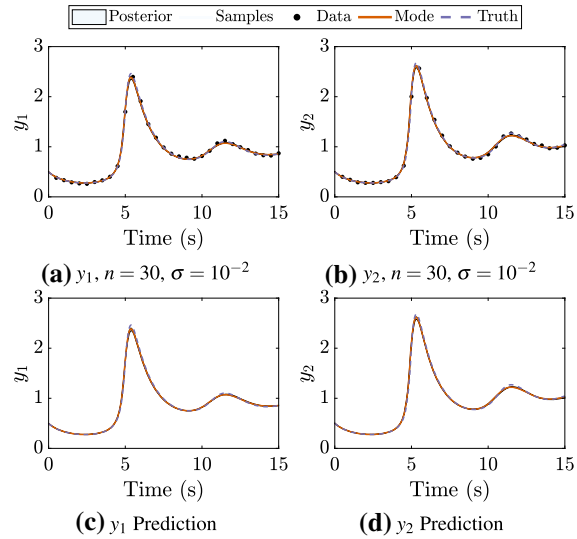
$$\begin{aligned} \frac{\partial C_1}{\partial t} &= \theta_1 \frac{\partial^2 C_1}{\partial x^2} + 0.1 - C_1 + \theta_3 C_1^2 C_2 \\ \frac{\partial C_2}{\partial t} &= \theta_2 \frac{\partial^2 C_2}{\partial x^2} C_2 + 0.9 - C_1^2 C_2 \end{aligned} \tag{56}$$

where C_1, C_2 specify the concentrations. A one-dimensional spatial grid was selected to have regular intervals of 0.4 units between boundaries of -40 and 40 for a total of 201 grid points for each of the two states. The boundary conditions at $x = \pm 40$ are

$$\frac{\partial C_1}{\partial x} = \frac{\partial C_2}{\partial x} = 0, \tag{57}$$

and the initial condition of the system was drawn from a uniform distribution as shown

$$(C_i)_j \sim \mathscr{U}(0.4, 0.6), \quad \text{for } t = 0; \ \forall i = 1, 2;$$



(a) y_1, $n = 30$, $\sigma = 10^{-2}$ **(b)** y_2, $n = 30$, $\sigma = 10^{-2}$

(c) y_1 Prediction **(d)** y_2 Prediction

Fig. 20 Reconstruction and prediction of the observables of the reaction diffusion system. The top row shows the reconstruction, and the bottom row shows the prediction for an alternate initial condition. The left column is the first measurement state (first moment), and the right column is the second measurement state (second moment). The estimates are very close to the truth, demonstrating the generality of the learned model

$$\forall j = 1, ..., 201. \tag{58}$$

Similar to the Lorenz example, for this system we attempt to learn only the model parameters, θ_1, θ_2, and θ_3 rather than the complete model. The measurement covariance matrix is assumed to be known, and the process noise covariance is fixed to be 1e-8 such that the total number of parameters that we are learning remains only three. The observation operator indirectly measures the concentration through only the first two moments of the concentration of the first species at certain time intervals

$$\begin{aligned} y_1(t) &= \int_{-40}^{40} C_1(t)\, \mathrm{d}x \\ y_2(t) &= \int_{-40}^{40} C_1^2(t)\, \mathrm{d}x. \end{aligned} \tag{59}$$

We collect measurements every 0.5 s for 15 s with noise standard deviation of 10^{-2}. The reconstructions and predictions of the moments from these data using the mode of the parameter posterior distribution are shown in Fig. 20. Additionally, the true and reconstructed contours of C_1 and C_2 are shown in Fig. 21.

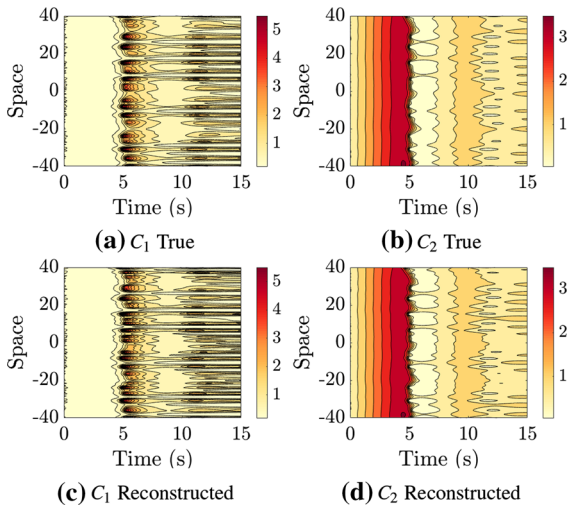


Fig. 21 The experiment is the same as in Fig. 20. The top row shows the true contours of C_1 and C_2. The bottom row shows the contours of C_1 and C_2 reconstructed using the mode of the parameter posterior distribution. Visually, the two rows appear very similar, reflecting the strong performance of the Bayesian algorithm

The Bayesian estimate shows close agreement with the truth.

7 Conclusion

In this paper, we have shown how data-driven system ID methods that consider only the measurement noise or only the process noise are impractical for many problems. When only the measurement noise is considered, increasingly many local minima arise as data collection is continued, making identification of the optimal solution difficult. When only process noise is considered, noisy and/or sparse measurements can cause the estimator to break down, even after incorporation of a denoising algorithm. By deriving a probabilistic model of our dynamical system from first principles, we were able to account for how parameter, model, and measurement uncertainty can each affect the learning problem in different ways. Then, using the UKF-MCMC algorithm, we compared the performance of the Bayesian approach to DMD and SINDy, which only consider model uncertainty, on a number of systems with varying values of measurement noise and frequency. It was shown that when substantial noise is introduced into the measurements, DMD and SINDy will fail, but the Bayesian algorithm continues to yield strong performance. Thus, it has been empirically shown that consideration of parameter, model, and measurement uncertainty leads to enhanced performance on a wider class of systems than that to which most least squares-based approaches can be reliably applied.

Acknowledgements This research was primarily supported by the DARPA Physics of AI Program under the grant "Physics Inspired Learning and Learning the Order and Structure of Physics," Agreement No. HR00111890030. It was also supported in part by the DARPA Artificial Intelligence Research Associate under the grant "Artificial Intelligence Guided Multi-scale Multi-physics Framework for Discovering Complex Emergent Materials Phenomena," Agreement No. HR00111990028.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

A Pseudocode

In this appendix, we provide the pseudocode for both the linear Kalman filter and nonlinear unscented Kalman filter algorithms. In the UKF algorithm, α and κ are parameters that determine the spread of the sigma points around the mean, β is a parameter used for incorporating prior information on the distribution of x, and the notation $[\cdot]_i$ denotes the i-th row of the matrix [22].

Algorithm 2 Kalman filtering for evaluating $p(\theta \mid \mathscr{Y}_n)$ (exact for linear models)

Input: System parameters $\theta = (\theta_\Psi, \theta_h, \theta_\Sigma, \theta_\Gamma)$;
Prior distribution $p(\theta)$;
Distribution on initial condition m_0, P_0;
Linear dynamical model parameterization $A(\theta_\Psi)$;
Linear observation model parameterization $H(\theta_h)$;
Covariance matrices $\Sigma(\theta_\Sigma)$ and $\Gamma(\theta_\Gamma)$
Output: Posterior evaluation $p(\theta \mid \mathscr{Y}_n)$
1: Compute the prior $p(\theta \mid \mathscr{Y}_0) = p(\theta)$
2: **for** $k = 1$ to n **do**
3: Predict $p(X_k|\theta, \mathscr{Y}_{k-1}) = \mathcal{N}(m_k^-, P_k^-)$
$m_k^-(\theta) = A(\theta_\Psi) m_{k-1}$
$P_k^-(\theta) = A(\theta_\Psi) P_{k-1} A^T(\theta_\Psi) + \Sigma(\theta_\Sigma)$
4: Compute the Evidence $p(y_k|\theta, \mathscr{Y}_{k-1}) = \mathcal{N}(\mu_k, S_k)$
$\mu_k(\theta) = H(\theta_h) m_k^-$
$S_k(\theta) = H(\theta_h) P_k^- H^T(\theta_h) + \Gamma(\theta_\Gamma)$
5: Update $p(X_k|\theta, \mathscr{Y}_k) = \mathcal{N}(m_k, P_k)$
$m_k(\theta) = m_k^- + P_k^- H^T(\theta_h) S_k^{-1}(y_k - \mu_k)$
$P_k(\theta) = P_k^- - P_k^- H^T(\theta_h) S_k^{-1} H(\theta_h) P_k^-$
6: Update $p(\theta|\mathscr{Y}_k) \propto p(y_k \mid \theta, \mathscr{Y}_{k-1}) p(\theta|\mathscr{Y}_{k-1})$
7: **end for**

Algorithm 3 Unscented Kalman filtering algorithm for approximating $p(\theta \mid \mathscr{Y}_n)$

Input: System parameters $\theta = (\theta_\Psi, \theta_h, \theta_\Sigma, \theta_\Gamma)$;
Prior distribution $p(\theta)$;
Distribution on initial condition m_0, P_0;
Dynamical model parametrization $\Psi(\theta_\Psi)$;
Observation model parameterization $h(\theta_h)$;
Covariance matrices $\Sigma(\theta_\Sigma)$ and $\Gamma(\theta_\Gamma)$;
UKF parameters α, κ, β

Output: Approximate evaluation of the posterior $p(\theta \mid \mathscr{Y}_n)$

Calculate $\lambda = \alpha^2(d + \kappa) - d$

Compute the weights

$W_0^{(m)} = \frac{\lambda}{d+\lambda}$

$W_0^{(c)} = \frac{\lambda}{d+\lambda} + (1 - \alpha^2 + \beta)$

$W_i^{(m)} = W_i^{(c)} = \frac{1}{2(d+\lambda)}, \quad \forall i = 1, \ldots, 2d$

Compute the prior $p(\theta \mid \mathscr{Y}_0) = p(\theta)$

for $k = 1$ to n **do**

Predict $p(X_k|\theta, \mathscr{Y}_{k-1}) \approx \mathscr{N}(m_k^-, P_k^-)$

Form the sigma points

$$\mathscr{X}_{k-1}^{(0)}(\theta) = m_{k-1}$$
$$\mathscr{X}_{k-1}^{(i)}(\theta) = m_{k-1} + \sqrt{d+\lambda}\left[\sqrt{P_{k-1}}\right]_i$$
$$\mathscr{X}_{k-1}^{(i+d)}(\theta) = m_{k-1} - \sqrt{d+\lambda}\left[\sqrt{P_{k-1}}\right]_i, \quad \forall i = 1, \ldots, d$$

Propagate the sigma points through the dynamical model

$$\hat{\mathscr{X}}_k^{(i)}(\theta) = \Psi(\mathscr{X}_k^{(i)}, \theta_\Psi), \quad \forall i = 0, \ldots, 2d$$

Compute the mean and covariance

$$m_k^-(\theta) = \sum_{i=0}^{2d} W_i^{(m)} \hat{\mathscr{X}}_k^{(i)}$$
$$P_k^-(\theta) = \sum_{i=0}^{2d} W_i^{(c)} (\hat{\mathscr{X}}_k^{(i)} - m_k^-)(\hat{\mathscr{X}}_k^{(i)} - m_k^-)^T + \Sigma(\theta_\Sigma)$$

Compute the Evidence $p(y_k|\theta, \mathscr{Y}_{k-1}) \approx \mathscr{N}(\mu_k, S_k)$

Update the sigma points

$$\mathscr{X}_{k-1}^{(0)}(\theta) = m_{k-1}$$
$$\mathscr{X}_{k-1}^{(i)}(\theta) = m_{k-1} + \sqrt{d+\lambda}\left[\sqrt{P_{k-1}}\right]_i$$
$$\mathscr{X}_{k-1}^{(i+d)}(\theta) = m_{k-1} - \sqrt{d+\lambda}\left[\sqrt{P_{k-1}}\right]_i, \quad \forall i = 1, \ldots, d$$

Propagate the sigma points through the observation model

$$\hat{\mathscr{Y}}_k^{(i)}(\theta) = h(\mathscr{X}_k^{(i)}, \theta_h), \quad \forall i = 0, \ldots, 2d$$

Compute the mean and covariance

$$\mu_k(\theta) = \sum_{i=0}^{2d} W_i^{(m)} \hat{\mathscr{Y}}_k^{(i)}$$
$$S_k(\theta) = \sum_{i=0}^{2d} W_i^{(c)} (\hat{\mathscr{Y}}_k^{(i)} - \mu_k^-)(\hat{\mathscr{Y}}_k^{(i)} - \mu_k^-)^T + \Gamma(\theta_\Gamma)$$

Update $p(X_k|\theta, \mathscr{Y}_k) \approx \mathscr{N}(m_k, P_k)$

$$C_k(\theta) = \sum_{i=0}^{2d} W_i^{(c)} (\mathscr{X}_k^{(i)} - m_k^-)(\hat{\mathscr{Y}}_k^{(i)} - \mu_k)^T$$
$$m_k(\theta) = m_k^- + (C_k S_k^{-1})(y_k - \mu_k)$$
$$P_k(\theta) = P_k^- - (C_k S_k^{-1}) S_k^{-1} (C_k S_k^{-1})^T$$

Update $p(\theta|\mathscr{Y}_k) \propto p(y_k \mid \theta, \mathscr{Y}_{k-1}) p(\theta|\mathscr{Y}_{k-1})$

end for

References

1. Glahn, H.R., Lowry, D.A.: The use of model output statistics (MOS) in objective weather forecasting. J. Appl. Meteorol. **11**(8), 1203 (1972)
2. Chitsazan, M.A., Fadali, M.S., Trzynadlowski, A.M.: Wind speed and wind direction forecasting using echo state network with nonlinear functions. Renew. Energy **131**, 879 (2019)
3. Scher, S.: Toward data-driven weather and climate forecasting: approximating a simple general circulation model with deep learning. Geophys. Res. Lett. **45**(22), 12 (2018)
4. Kevrekidis, I.G., Gear, C.W., Hyman, J.M., Kevrekidid, P.G., Runborg, O., Theodoropoulos, C., et al.: Equation-free, coarse-grained multiscale computation: enabling mocroscopic simulators to perform system-level analysis. Commun. Math. Sci. **1**(4), 715 (2003)
5. Raissi, M.: Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Mach. Learn. Res. **19**(1), 932 (2018)
6. Takeishi, N., Kawahara, Y., Yairi, T.: Learning Koopman invariant subspaces for dynamic mode decomposition. In *Advances in Neural Information Processing Systems*, pp. 1130–1140 (2017)
7. Liu, Y.J., Li, J., Tong, S., Chen, C.P.: Neural network control-based adaptive learning design for nonlinear systems with full-state constraints. IEEE Trans. Neural Netw. Learn. Syst. **27**(7), 1562 (2016)
8. Cui, R., Yang, C., Li, Y., Sharma, S.: Adaptive neural network control of AUVs with control input nonlinearities using reinforcement learning. IEEE Trans. Syst. Man Cybern. Syst. **47**(6), 1019 (2017)
9. Sun, K., Jianbin, Q., Karimi, H.R., Fu, Y.: Event-triggered robust fuzzy adaptive finite-time control of nonlinear systems with prescribed performance. IEEE Trans. Fuzzy Syst. (2020). https://doi.org/10.1109/TFUZZ.2020.2979129
10. Polycarpou, M.M., Ioannou, P.A.: A robust adaptive nonlinear control design: a robust adaptive nonlinear control design. In *1993 American Control Conference*, pp. 1365–1369. IEEE (1993)
11. Ott, E., Hunt, B.R., Szunyogh, I., Zimin, A.V., Kostelich, E.J., Corazza, M., Kalnay, E., Patil, D., Yorke, J.A.: A local ensemble Kalman filter for atmospheric data assimilation. Tellus A Dyn. Meteorol. Oceanogr. **56**(5), 415 (2004)
12. Hunt, B., Kalnay, E., Kostelich, E., Ott, E., Patil, D., Sauer, T., Szunyogh, I., Yorke, J., Zimin, A.: Four-dimensional ensemble Kalman filtering. Tellus A **56**(4), 273 (2004)
13. Sirovich, L.: Turbulence and the dynamics of coherent structures. I. Coherent structures. Q. Appl. Math. **45**(3), 561 (1987)
14. Lumley, J.L.: Stochastic Tools in Turbulence. Courier Corporation, North Chelmsford (2007)

15. Schmid, P.J.: Dynamic mode decomposition of numerical and experimental data. J. Fluid Mech. **656**, 5–28 (2010)
16. Leffens, E., Markley, F., Shuster, M.: Kalman filtering for spacecraft attitude estimation. J. Guid. Control Dyn. **5**(5), 417 (1982)
17. Slotine, J.J.E., Li, W.: On the adaptive control of robot manipulators. Int. J. Robot. Res. **6**(3), 49 (1987)
18. Craig, J.J., Hsu, P., Sastry, S.S.: Adaptive control of mechanical manipulators. Int. J. Robot. Res. **6**(2), 16 (1987)
19. Li, K., Kou, J., Zhang, W.: Deep neural network for unsteady aerodynamic and aeroelastic modeling across multiple Mach numbers. Nonlinear Dyn. **96**(3), 2157 (2019)
20. De Paula, N., Marques, F.: Multi-variable Volterra kernels identification using time-delay neural networks: application to unsteady aerodynamic loading. Nonlinear Dyn. **97**(1), 767 (2019)
21. Li, W., Laima, S., Jin, X., Yuan, W., Li, H.: A novel long short-term memory neural-network-based self-excited force model of limit cycle oscillations of nonlinear flutter for various aerodynamic configurations. Nonlinear Dyn. **100**, 2071–2087 (2020)
22. Särkkä, S.: Bayesian Filtering and Smoothing. Institute of Mathematical Statistics Textbooks. Cambridge University Press, Cambridge (2013)
23. Law, K., Stuart, A., Zygalakis, K.: Data Assimilation. Springer, Cham (2015)
24. Barfoot, T.D.: State Estimation for Robotics. Cambridge University Press, Cambridge (2017)
25. Berger, J.O.: Statistical Decision Theory and Bayesian Analysis. Springer, New York (1985)
26. Brunton, S.L., Proctor, J.L., Kutz, J.N.: Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc. Nat. Acad. Sci. **113**(15), 3932 (2016)
27. Erazo, K., Nagarajaiah, S.: An offline approach for output-only Bayesian identification of stochastic nonlinear systems using unscented Kalman filtering. J. Sound Vib. **397**, 222–240 (2018)
28. Haario, H., Kalachev, L., Hakkarainen, J.: Generalized correlation integral vectors: a distance concept for chaotic dynamical systems. Chaos Interdiscip. J. Nonlinear Sci. **25**(6), 063102 (2015)
29. Noh, S.: Posterior inference on parameters in a nonlinear DSGE model via Gaussian-based filters. Comput. Econ. (2019). https://doi.org/10.1007/s10614-019-09944-5
30. Drovandi, C., Everitt, R.G., Golightly, A., Prangle, D.: Ensemble MCMC: Accelerating Pseudo-Marginal MCMC for State Space Models using the Ensemble Kalman Filter. arXiv preprint arXiv:1906.02014 (2019)
31. Khalil, M., Sarkar, A., Adhikari, S., Poirel, D.: The estimation of time-invariant parameters of noisy nonlinear oscillatory systems. J. Sound Vib. **344**, 81–100 (2015)
32. Andrieu, C., Roberts, G.O.: The Pseudo-marginal approach for efficient Monte Carlo computations. Ann. Stat. **37**(2), 697 (2009)
33. Gelman, A.: Prior distributions for variance parameters in hierarchical models. Bayesian Anal. **1**, 515–533 (2006)
34. Hemati, M., Rowley, C., Cattafesta, L.: De-biasing the dynamic mode decomposition for applied Koopman spectral analysis. Theor. Comput. Fluid Dyn. **10**, 10 (2017). https://doi.org/10.1007/s00162-017-0432-2
35. Chartrand, R., Appl, I.S.R.N.: Numerical differentiation of noisy. Nonsmooth Data Math. (2011). https://doi.org/10.5402/2011/164564
36. Yoshida, K., Takamatsu, H., Matsumoto, S.: Nonlinear identification of torsional driveshaft vibrations in a full-scale automotive vehicle during acceleration. Nonlinear Dyn. **86**(1), 711 (2016)
37. Cheng, C., Peng, Z., Dong, X., Zhang, W., Meng, G.: Nonlinear system identification using Kautz basis expansion-based Volterra-PARAFAC model. Nonlinear Dyn. **94**(3), 2277 (2018)
38. Yuan, L., Yang, Q., Zeng, C.: Chaos detection and parameter identification in fractional-order chaotic systems with delay. Nonlinear Dyn. **73**(1–2), 439 (2013)
39. Venkataraman, H.K., Seiler, P.J.: Recovering robustness in model-free reinforcement learning. In: 2019 American Control Conference (ACC), pp. 4210–4216. IEEE (2019)
40. Peng, H., Li, L., Yang, Y., Liu, F.: Parameter estimation of dynamical systems via a chaotic ant swarm. Phys. Rev. E **81**(1), 016207 (2010)
41. Evensen, G., Dee, D.P.,Schröter, J.: Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. In: Ocean Modeling and Parameterization, pp. 373–398. Springer (1998)
42. Wu, K., Xiu, D.: Data-driven deep learning of partial differential equations in modal space. J. Comput. Phys. **408**, 109307 (2020)
43. Constantine, P.G., Wang, Q.: Residual minimizing model interpolation for parameterized nonlinear dynamical systems. SIAM J. Sci. Comput. **34**(4), A2118 (2012)
44. Chen, T.Q., Rubanova, Y., Bettencourt, J., Duvenaud, D.K.: Neural ordinary differential equations. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 31, pp. 6571–6583. Curran Associates Inc, Red Hook (2018)
45. Tsoulos, I.G., Gavrilis, D., Glavas, E.: Solving differential equations with constructed neural networks. Neurocomputing **72**(10–12), 2385 (2009)
46. Lagaris, I.E., Likas, A., Fotiadis, D.I.: Artificial neural networks for solving ordinary and partial differential equations. IEEE Trans. Neural Netw. **9**(5), 987 (1998)
47. Proctor, J.L., Brunton, S.L., Kutz, J.N.: Dynamic mode decomposition with control, dynamic mode decomposition with control SIAM. J. Appl. Dyn. Syst. **15**, 142 (2014)
48. Rowley, C.W., Mezić, I., Bagheri, S., Schlatter, P., Henningson, D.S.: Spectral analysis of nonlinear flows. J. Fluid Mech. **641**, 115–127 (2009)
49. Golub, G.H., Loan, C.F.V.: An analysis of the total least squares problem. SIAM J. Numer. Anal. **17**(6), 883 (1980)
50. Huffel, S.V., Vandewalle, J.: Analysis and properties of the generalized total least squares problem $AX \approx B$ when some or all columns in A are subject to error. SIAM J. Matrix Anal. Appl. **10**(3), 294 (1989)
51. Takeishi, N., Kawahara, Y., Tabei, Y., Yairi, T.: Bayesian dynamic mode decomposition. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, pp. 2814–2821 (2017)
52. Evensen, G.: Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods

to forecast error statistics. J. Geophys. Res. Oceans **99**(C5), 10143 (1994)
53. Arasaratnam, I., Haykin, S.: Cubature Kalman filters. IEEE Trans. Autom. Control **54**(6), 1254 (2009)
54. Julier, S.J., Uhlmann, J.K.: New extension of the Kalman filter to nonlinear systems. In: I. Kadar (ed.) Signal Processing, Sensor Fusion, and Target Recognition VI, vol. 3068. International Society for Optics and Photonics (SPIE), vol. 3068, pp. 182–193 (1997)
55. Gordon, N.J., Salmond, D.J., Smith, A.F.: Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In: IEE Proceedings F (Radar and Signal Processing), vol. 140, pp. 107–113. IET (1993)
56. Andrieu, C., Doucet, A., Holenstein, R.: Particle Markov chain Monte Carlo methods. J. R. Stat. Soc. Ser. B (Stat. Methodol.) **72**(3), 269 (2010)
57. Haario, H., Laine, M., Mira, A., Saksman, E.: Chaotic dynamical systems. Stat. Comput. **16**, 339 (2006)
58. Houtzager, I.: Total least squares with mixed and/or weighted disturbances. MATLAB File Exchange (2019). Retrieved 5 Dec 2019
59. Colton, D., Kress, R.: Inverse Acoustic and Electromagnetic Scattering Theory, vol. 93. Springer, New York (2019)
60. Wirgin, A.: arXiv preprint arXiv:math-ph/0401050 (2004)
61. Chen, Y., Pi, D., Wang, B.: Enhanced global flower pollination algorithm for parameter identification of chaotic and hyper-chaotic system. Nonlinear Dyn. **97**(2), 1343 (2019)
62. Lu, Z.R., Liu, G., Liu, J., Chen, Y.M., Wang, L.: Parameter identification of nonlinear fractional-order systems by enhanced response sensitivity approach. Nonlinear Dyn. **95**(2), 1495 (2019)
63. Narayanan, M., Narayanan, S., Padmanabhan, C.: Parametric identification of nonlinear systems using multiple trials. Nonlinear Dyn. **48**(4), 341 (2007)
64. Marzouk, Y.M., Najm, H.N., Rahn, L.A.: Stochastic spectral methods for efficient Bayesian solution of inverse problems. J. Comput. Phys. **224**(2), 560 (2007)
65. Marzouk, Y.M., Najm, H.N.: Dimensionality reduction and polynomial chaos acceleration of Bayesian inference in inverse problems. J. Comput. Phys. **228**(6), 1862 (2009)
66. Brastein, O.M., Perera, D.W.U., Pfeifer, C., Skeie, N.O.: Parameter estimation for grey-box models of building thermal behaviour. Energy Build. **169**, 58 (2018)
67. Dokos, S., Lovell, N.H.: Parameter estimation in cardiac ionic models. Prog. Biophys. Mol. Biol. **85**(2–3), 407 (2004)
68. Kivman, G.A.: Sequential parameter estimation for stochastic systems. Nonlinear Processes Geophys. **10**(3), 253 (2003)
69. Lorenz, E.N.: Deterministic nonperiodic flow. J. Atmos. Sci. **20**(2), 130 (1963)
70. Springer, S., Haario, H., Shemyakin, V., Kalachev, L., Shchepakin, D.: Robust parameter estimation of chaotic systems. Inverse Probl. Imaging **13**(6), 1189 (2019)
71. Politi, A.: Lyapunov exponent. Scholarpedia **8**(3), 2722 (2013). https://doi.org/10.4249/scholarpedia.2722. Revision #137286
72. Govorukhin, V.: Calculation Lyapunov exponents for ode. MATLAB File Exchange (2020). Retrieved 29 June 2020
73. Wolf, A., Swift, J.B., Swinney, H.L., Vastano, J.A.: Determining Lyapunov exponents from a time series. Physica D **16**(3), 285 (1985)