On robust input design for nonlinear dynamical models ⋆

Patricio E. Valenzuela [a] Johan Dahlin [b] Cristian R. Rojas [a] Thomas B. Schön [b]

[a] *Department of Automatic Control, KTH Royal Institute of Technology, SE-100 44, Stockholm, Sweden (e-mail: {pva, crro}@kth.se).*

[b] *Department of Information Technology, Uppsala University, SE-751 05, Uppsala, Sweden (e-mail: liu@johandahlin.com, thomas.schon@it.uu.se).*

Abstract

We present a method for robust input design for nonlinear state-space models. The method optimizes a scalar cost function of the Fisher information matrix over a set of marginal distributions of stationary processes. By using elements from graph theory we characterize such a set. Since the true system is unknown, the resulting optimization problem takes the uncertainty on the true value of the parameters into account. In addition, the required estimates of the information matrix are computed using particle methods, and the resulting problem is convex in the decision variables. Numerical examples illustrate the proposed technique by identifying models using the expectation maximization algorithm.

Key words: System identification, input design, particle filter, nonlinear systems.

1 Introduction

Input design is concerned with generating an excitation signal that maximizes the information retrieved from an experiment, quantified in terms of a cost function related to the intended model application. Some of the initial contributions are discussed in [7] and [20]. Since then, many contributions to the subject have been presented; see e.g. [16, 19, 22, 50] and the references therein.

In the case of dynamical systems, the existing results on input design are mostly focused on linear models. The assumption of a linear model structure can reduce the complexity of the problem, leading to formulations that are convex in the decision variables [29]. In this case, the convexity of the problem is achieved by designing the power spectrum of the input signal. Several approaches to input design for linear models have been proposed in the literature involving, e.g., linear matrix inequalities (LMI) [24, 27], Markov chains [2], and time domain techniques [40]. With the exception of the methods in [24, 27] that rely on convexification of the problem, the previous formulations are non-convex, which illustrates the difficulty of solving the input design problem.

⋆ This work was supported by the Swedish Research Council under contracts 621-2013-5524, 621-2011-5890 and 621-2009-4017, and by the European Research Council under the advanced grant LEARN, contract 267381.

In recent years, there has been an interest to extend the input design methods to nonlinear (NL) model structures. The main issue is that the frequency domain methods cannot be applied, which restricts the applicability of convex formulations [24, 27]. The first approaches to the problem considered NL finite impulse response (FIR) models [23, 26]. In [23] the input design problem is analyzed using the knowledge from linear systems, while in [26] the input design problem is solved over a set of marginal distributions of stationary processes.

An extension of the input design problem to structured NL models is presented in [47, 48], where the model is given by an interconnection of linear models and static nonlinearities. The class of NL model structures is also generalized in [17], where the input signal is optimized over an alphabet with finite cardinality. A multilevel excitation is also considered in [9] for identification of Wiener models. The restriction to a finite alphabet is relaxed in [21], where an ARX process is designed as input for the identification of NL state-space models (SSMs). A graph theoretical methodology to design inputs for identification of NL output-error models is developed in [45, 46], which is extended to NL-SSMs in [44].

The existing results on input design allow to optimize input signals when the system contains NL functions, but the restrictions on the system dynamics and/or the input structure are the main limitations of most of the

previous contributions. Moreover, with the exception of multilevel excitation [17, 26], and stationary processes [2, 44, 45, 46], most of the proposed methods cannot handle amplitude limitations on the input signal, which could arise due to physical and/or safety reasons.

The previously mentioned input design methods assume that a prior estimate of the model parameters is available for optimization. The requirement of such knowledge is a common issue in input design and different solutions to this difficulty have been proposed [18, 35, 36, 37, 49].

The main contribution of this article is to propose a robust input design method for the identification of NL-SSMs with input constraints, which extends the model structure considered in [45, 46], and the nominal input design presented in [44]. The optimal input signal is considered to be a realization of a stationary process, which maximizes a scalar function of the Fisher information matrix (FIM). To pose a tractable convex problem, we restrict the optimization to a set of marginal distributions of stationary processes with a finite alphabet. This set is a polytope and hence it can be described by a convex combination of its vertices. The vertices are cumulative distribution functions that can be found using de Bruijn graphs, as discussed in [44, 45]. Since the vertices of the set are known, we can draw an input realization and compute an estimate of the FIM for each vertex using particle methods [10, 15]. The estimates of the information matrices are computed using the method introduced in [39], which only requires one realization of the input-output data, and thus reducing the computational effort when estimating the FIM compared to [44].

To make the input design robust against model uncertainty, the optimization problem considers a measure of the uncertainty of the parameters, which relaxes the requirements on the knowledge of the system assumed in [44, 45, 46]. The method is illustrated through numerical examples, where the designed input is employed to identify a NL-SSM using the expectation-maximization (EM) algorithm [11, 30, 38].

The rest of this article is organized as follows. Section 2 states the problem and the main challenges when designing inputs for identification of NL-SSM. Section 3 describes the graph theoretical approach to input design. Section 4 discusses the estimation of the FIM using particle methods. A summary of the proposed robust input design method is presented in Section 5. The generation of the optimal input signal is addressed in Section 6. To illustrate the correctness and utility of the method, two numerical examples are discussed in Section 7. Finally, Section 8 concludes this work and presents future research directions.

Notation: Throughout this article, $\mathbb{N}$ denotes the set of natural numbers, $\mathbb{R}^p$ denotes the set of p-dimensional vectors with real entries, $\mathbb{R}^{p\times r}$ is the set of $p \times r$ matrices with real entries, and $\mathbb{R}_+$ the set of positive real numbers. $\mathbf{P}$, $\mathbf{E}$, and $\mathrm{Var}\{\cdot\}$ stand for a probability measure, the expected value, and the variance, respectively. Sometimes a subscript is added to $\mathbf{P}$ and $\mathbf{E}$ to clarify the stochastic process considered by these operators. Finally, for a finite set $\mathcal{A}$, $|\mathcal{A}|$ denotes its cardinality.

2 Problem formulation

Consider a NL-SSM described for all $t \geq 1$ by

$$x_t|x_{t-1} \sim f_\theta(x_t|x_{t-1}, u_{t-1}), \tag{1a}$$
$$y_t|x_t \sim g_\theta(y_t|x_t, u_t), \tag{1b}$$
$$x_0 \sim \mu_\theta(x_0), \tag{1c}$$

where f_θ, g_θ, and μ_θ denote probability density functions (pdf) parameterized by $\theta \in \Theta \subset \mathbb{R}^{n_\theta}$ (where Θ is an open set). Here, $u_t \in \mathbb{R}^{n_u}$ denotes the input signal, $x_t \in \mathbb{R}^{n_x}$ are the (unobserved/latent) internal states, and $y_t \in \mathbb{R}^{n_y}$ are the measured outputs.

The objective in this article is to design an input signal $u_{1:n_{\mathrm{seq}}} := (u_1, \ldots, u_{n_{\mathrm{seq}}})$, as a realization of a stationary process, such that the NL-SSM (1) can be identified with maximum accuracy as defined by a scalar function of the FIM [29]. In the sequel, we assume that there exists at least one parameter $\theta_0 \in \Theta$ such that the model (1) exactly describes the pdfs of the system, i.e., there is no undermodelling [29].

Given $u_{1:n_{\mathrm{seq}}}$, the FIM is defined as

$$\mathcal{I}_F^{n_{\mathrm{seq}}}(\theta_0) := \mathbf{E}\left\{\mathcal{S}(\theta_0)\mathcal{S}^\top(\theta_0)|\, u_{1:n_{\mathrm{seq}}}\right\}, \tag{2}$$

where $\mathcal{S}(\theta_0)$ denotes the score function, i.e.,

$$\mathcal{S}(\theta_0) := \nabla_\theta\, \ell_\theta(y_{1:n_{\mathrm{seq}}})\big|_{\theta=\theta_0}. \tag{3}$$

Here, $\ell_\theta(y_{1:n_{\mathrm{seq}}})$ denotes the log-likelihood function

$$\ell_\theta(y_{1:n_{\mathrm{seq}}}) := \log p_\theta(y_{1:n_{\mathrm{seq}}}|u_{1:n_{\mathrm{seq}}}). \tag{4}$$

We note that the expected value in (2) is with respect to the stochastic processes in (1). Since we consider $u_{1:n_{\mathrm{seq}}}$ as a realization of a stationary process, here we are interested in the *per-sample* FIM, defined as

$$\begin{aligned}\mathcal{I}_F(\theta_0) &:= \frac{1}{n_{\mathrm{seq}}}\mathbf{E}_u\left\{\mathcal{I}_F^{n_{\mathrm{seq}}}(\theta_0)\right\} \\ &= \frac{1}{n_{\mathrm{seq}}}\mathbf{E}\left\{\mathcal{S}(\theta_0)\mathcal{S}^\top(\theta_0)\right\},\end{aligned} \tag{5}$$

where the expected value in (5) is over both the stochastic processes in (1), and the stochastic vector $u_{1:n_{\mathrm{seq}}}$.

We note that (5) depends on the cumulative distribution function (cdf) of $u_{1:n_{\text{seq}}}$, denoted by $P_u(u_{1:n_{\text{seq}}})$. Therefore, the input design problem is to find a cdf $P_u^{\text{opt}}(u_{1:n_{\text{seq}}})$ which maximizes a scalar function of (5), $\mathcal{H} : \mathbb{R}^{n_\theta \times n_\theta} \times \Theta \to \mathbb{R}$, where [1] $\mathcal{H}$ is a matrix concave function in its first argument [1, pp. 108]. Different choices of $\mathcal{H}$ have been proposed in the literature, see e.g. [37]; some examples are $\mathcal{H}(A, \theta) = \log\det(A)$, and $\mathcal{H}(A, \theta) = -\operatorname{tr}\{A^{-1}\}$ for $A \in \mathbb{R}^{n_\theta \times n_\theta}$ non-singular.

To simplify our problem, we will assume that u_t can only adopt a finite number c_{seq} of values. We denote this set of values as $\mathcal{C}$. With the previous assumption, we can define the following set:

$$\mathcal{P}_{\mathcal{C}} := \left\{ p_u : \mathcal{C}^{n_{\text{seq}}} \to \mathbb{R} \middle| \begin{array}{c} p_u(\mathbf{x}) \geq 0,\ \forall \mathbf{x} \in \mathcal{C}^{n_{\text{seq}}}; \\ \sum_{\mathbf{x} \in \mathcal{C}^{n_{\text{seq}}}} p_u(\mathbf{x}) = 1; \\ \sum_{v \in \mathcal{C}} p_u(v,\, \mathbf{z}) = \sum_{v \in \mathcal{C}} p_u(\mathbf{z},\, v)\,, \forall \mathbf{z} \in \mathcal{C}^{n_{\text{seq}}-1} \end{array} \right\}. \quad (6)$$

The set $\mathcal{P}_{\mathcal{C}}$ introduced in (6) constrains the probability mass function (pmf) $u_{1:n_{\text{seq}}}$ of p_u to the set of marginal pmfs associated with stationary processes. We refer to [46] for more details about the derivation of the set $\mathcal{P}_{\mathcal{C}}$.

So far we have been concerned with the formulation of the problem in terms of the stationary process describing $u_{1:n_{\text{seq}}}$. However, we still need to consider a remaining issue: Equation (5) depends on the parameter θ_0 describing the true system (1). Therefore, the optimal input sequence $u_{1:n_{\text{seq}}}$ depends on the parameter we want to estimate, which limits the practical applicability of the previous formulation. To overcome this issue, we consider the function $\mathcal{R} : \Theta \to \mathbb{R}$ that measures the uncertainty over Θ. In the following, two definitions of $\mathcal{R}$ are considered: $\mathcal{R} = \mathbf{E}_\theta\{\cdot\}$ (where we assume that Θ is a measurable space with known cdf P_θ), and $\mathcal{R} = \min_{\theta \in \Theta}\{\cdot\}$.

To summarize, the problem we are interested in solving can be written as

Problem 1 *Design an optimal input signal $u_{1:n_{\text{seq}}} \in \mathcal{C}^{n_{\text{seq}}}$ as a realization from $p_u^{\text{opt}}(u_{1:n_{\text{seq}}})$, where*

$$p_u^{\text{opt}} := \arg\max_{p_u \in \mathcal{P}_{\mathcal{C}}} \mathcal{R}\{\mathcal{H}(\mathcal{I}_F(p_u, \theta),\, \theta)\}\,, \quad (7)$$

with $\mathcal{H} : \mathbb{R}^{n_\theta \times n_\theta} \times \Theta \to \mathbb{R}$ a matrix concave function, $\mathcal{R} : \Theta \to \mathbb{R}$, and $\mathcal{I}_F(p_u, \theta) \in \mathbb{R}^{n_\theta \times n_\theta}$ defined as in (5). ■

[1] We let $\mathcal{H}$ have an argument on Θ as the function can explicitly depend on the model parameter.

Problem 1 is difficult to solve. The main challenge is:

(A) The set Θ may be uncountably infinite, which implies that the computation of $\mathcal{R}\{\mathcal{H}(\mathcal{I}_F(\theta),\, \theta)\}$ can be intractable.

To address issue (A), we consider a procedure that depends on $\mathcal{R}$. For $\mathcal{R} = \mathbf{E}_\theta\{\cdot\}$, we solve a Monte-Carlo approximation of Problem 1 by sampling N_s points from the set Θ according to the cdf P_θ, and replacing the expected value by its sample mean estimate. In the case where $\mathcal{R} = \min_{\theta \in \Theta}\{\cdot\}$, we employ the scenario approach [3, 49]. By sampling N_s points from the set Θ according to a given cdf P_s, we can rewrite Problem 1 as an optimization problem over a finite number of points in Θ. In this case we will obtain a sub-optimal solution to Problem 1 which, however, can be made close to the optimal solution by increasing N_s; we refer to [3, 4] for more details.

In addition to issue (A), the parameterization of the set $\mathcal{P}_{\mathcal{C}}$ and the computation of the FIM (5) for the model (1) are also part of the complexity of solving Problem 1. In this article, these points will be addressed by employing existing techniques. To parameterize the set $\mathcal{P}_{\mathcal{C}}$, we follow the graph theoretical approach proposed in [46], which is briefly described in the next section. Finally, the computation of the FIM (5) will rely on particle methods and it will be considered in Section 4.

Remark 1 *The assumptions of prior knowledge about the structure of the pdfs characterizing* (1) *and the finite alphabet of u_t might seem very restrictive. However, the requirement of a structure for the pdfs in* (1) *is not more restrictive than the requirement of a model structure for linear models, cf. [29]. On the other hand, the finite alphabet assumption for u_t is introduced to make Problem 1 tractable, as it is difficult to parameterize in a tractable manner the set $\mathcal{P}_{\mathcal{C}}$ resulting from removing the finite alphabet requirement on u_t.* ■

Remark 2 *An alternative to solve Problem 1 is to directly optimize over* (2) *by designing every sample in $u_{1:n_{\text{seq}}}$. However, the resulting optimization is non-convex, and hence the designed input can only guarantee local optimality.* ■

3 Describing the set of stationary processes

Here we briefly discuss the graph theoretical approach in [45, 46] to characterize the set $\mathcal{P}_{\mathcal{C}}$, and we refer to [46] for more details about this method.

One of the difficulties associated with $\mathcal{P}_{\mathcal{C}}$ is that $p_u \in \mathcal{P}_{\mathcal{C}}$ is of dimension n_{seq}, where n_{seq} can be very large. This issue is relaxed by restricting p_u to the set of pmfs defined over $\mathcal{C}^{n_m}$, where $n_m < n_{\text{seq}}$, and corresponding to marginal pmfs of stationary processes. This assumption

allows to solve an approximation of Problem 1 in the sense that the difference between the optimal cost for the solution considering $p_u(u_{1:n_{\text{seq}}})$, and the optimal cost for $p_u(u_{1:n_m})$ can be made arbitrarily small by defining n_m sufficiently close to n_{seq}. We refer to Section 6 for the details on the computation of $u_{1:n_{\text{seq}}}$ from $p_u(u_{1:n_m})$.

A second difficulty associated with $\mathcal{P}_{\mathcal{C}}$ corresponds to its parameterization. To solve this issue, we first notice that all the elements in $\mathcal{P}_{\mathcal{C}}$ can be represented as a convex combination of its extreme points, since $\mathcal{P}_{\mathcal{C}}$ is described by a finite number of linear inequalities [34, Chapter 17]. We will refer to $\mathcal{V}_{\mathcal{P}_{\mathcal{C}}} := \{v_1, \ldots, v_{n_{\mathcal{V}}}\}$ as the set of the extreme points of $\mathcal{P}_{\mathcal{C}}$. As the set $\mathcal{P}_{\mathcal{C}}$ corresponds to pmfs, then $\mathcal{V}_{\mathcal{P}_{\mathcal{C}}}$ corresponds to pmfs describing every $p \in \mathcal{P}_{\mathcal{C}}$ as

$$p = \sum_{j=1}^{n_{\mathcal{V}}} \alpha_j v_j \,, \tag{8}$$

where $\alpha_j \geq 0$, for all $j \in \{1, \ldots, n_{\mathcal{V}}\}$, and

$$\sum_{j=1}^{n_{\mathcal{V}}} \alpha_j = 1 \,. \tag{9}$$

The problem of parameterizing $\mathcal{P}_{\mathcal{C}}$ turns then to find $\mathcal{V}_{\mathcal{P}_{\mathcal{C}}}$. To this end, we make use of the graph theoretical approach proposed in [46]. First, we note that every element in $\mathcal{C}^{n_m}$ can be viewed as one node in a graph [2] . In addition, the transitions (edges) between the elements in $\mathcal{C}^{n_m}$ are defined by the possible values of u_{k+1} when we move from $(u_{k-n_m+1}, \ldots, u_k)$ to $(u_{k-n_m+2}, \ldots, u_{k+1})$, for all integers $k > 0$. The resulting graph is referred to as a *de Bruijn* graph. This is illustrated in Figure 1 for $c_{\text{seq}} = 2$, $n_m = 2$, and $\mathcal{C} = \{0, 1\}$. We see that, if we are in node (0, 1) at time t, then we can only transit to node (1, 0) or (1, 1) at time $t + 1$. In the following, $\mathcal{G}_{\mathcal{X}} = \{\mathcal{X}, \mathcal{E}\}$ will denote a de Bruijn graph with set of nodes $\mathcal{X}$ and set of edges $\mathcal{E}$.

To find $\mathcal{V}_{\mathcal{P}_{\mathcal{C}}}$ we need the concept of prime cycles. A *prime cycle* is an elementary cycle [3] whose set of nodes does not have a proper subset which is an elementary cycle [51, pp. 678]. It is known that the prime cycles of a graph describe $\mathcal{V}_{\mathcal{P}_{\mathcal{C}}}$ [51, Theorem 6]. In other words, each prime cycle defines one element $v_j \in \mathcal{V}_{\mathcal{P}_{\mathcal{C}}}$. Furthermore, each v_j is a uniform distribution with support on the nodes of its prime cycle, for all $j \in \{1, \ldots, n_{\mathcal{V}}\}$ [51, pp. 681]. Therefore, the set $\mathcal{V}_{\mathcal{P}_{\mathcal{C}}}$ is described by finding all the prime cycles associated with the de Bruijn graph $\mathcal{G}_{\mathcal{C}^{n_m}}$. As an example, consider Figure 1. One prime cycle in this graph is given by ((0, 1), (1, 0), (0, 1)). For this prime cycle, the associated pmf is $v_j(0, 1) = v_j(1, 0) = 0.5$, and zero otherwise.

[2] Note that $|\mathcal{C}^{n_m}| = (c_{\text{seq}})^{n_m}$.

[3] An *elementary cycle* is a cycle where all nodes are different, except for the first and the last [25, p. 77].

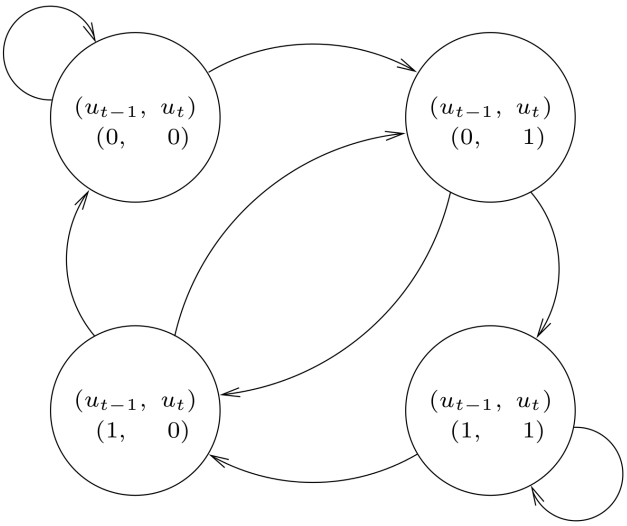

Fig. 1. Example of graph derived from $\mathcal{C}^{n_m}$, with $n_m = 2$, and $\mathcal{C} := \{0, 1\}$.

It is known that all the prime cycles associated with $\mathcal{G}_{\mathcal{C}^{n_m}}$ can be derived from the elementary cycles associated with $\mathcal{G}_{\mathcal{C}^{n_m-1}}$ [51, Lemma 4], which can be found using existing algorithms [4] .

Based on the prime cycles, it is possible to generate an input sequence $u^j_{1:n_{\text{seq}}}$ from v_j, which will be referred to as the *basis inputs*. As an example, consider the prime cycle ((0, 1), (1, 0), (0, 1)) of the graph in Figure 1. The sequence $u^j_{1:n_{\text{seq}}}$ associated with this prime cycle is given by taking the value of u_t in each node, i.e., $u^j_{1:n_{\text{seq}}} = \{1, 0, 1, 0, \ldots, ((-1)^{n_{\text{seq}}-1} + 1)/2\}$.

Given $u^j_{1:n_{\text{seq}}}$, we can use it to obtain the corresponding information matrix [5] for $v_j \in \mathcal{V}_{\mathcal{P}_{\mathcal{C}}}$, denoted by $\mathcal{I}_F^{(j)}$. However, in general the matrix $\mathcal{I}_F^{(j)}$ cannot be computed explicitly. This difficulty is overcome by using particle methods to approximate $\mathcal{I}_F^{(j)}$, as discussed in the next section.

Remark 3 *Following [46], the method presented here can also be extended to multiple inputs. In that case, the only required modification is the definition of the nodes, which are then associated with the possible states of an $n_u \times n_m$ matrix.* ■

Remark 4 *The number of prime cycles in a graph de-*

[4] For the examples in Section 7, we have used the algorithm presented in [25, pp. 79–80] complemented with the one proposed in [42, pp. 157].

[5] To simplify the discussion, here we use n_{seq} input samples to compute $\mathcal{I}_F^{(j)}$. However, the number of samples to compute $\mathcal{I}_F^{(j)}$ can be different from n_{seq} in general.

pends exponentially on n_m, which in turn results in an exponential increase in the computational time required to compute such cycles as n_m increases, which has already been noticed in [46]. On the other hand, the number of prime cycles depends polynomially in c_{seq}, resulting in a polynomial increase in the computational time to compute the prime cycles as c_{seq} increases. ■

4 Estimation of the FIM

From (8), we have that every $p \in \mathcal{P}_{\mathcal{C}}$ is a convex combination of the elements in $\mathcal{V}_{\mathcal{P}_{\mathcal{C}}}$. Hence, it is possible to approximate the FIM associated with p as

$$\mathcal{I}_F(\gamma, \theta) = \sum_{j=1}^{n_{\mathcal{V}}} \alpha_j \mathcal{I}_F^{(j)}(\theta)\,, \tag{10}$$

where $\gamma := \{\alpha_j\}_{j=1}^{n_{\mathcal{V}}}$ and $\mathcal{I}_F^{(j)}(\theta)$ denotes the FIM obtained using the basis input $u_{1:n_{\text{seq}}}^{(j)}$. Equation (10) is an approximation of the FIM associated with p in the sense that (10) converges to the value of the FIM for p as both n_{seq} and n_m tend to infinity [43].

The main problem for computing (10) is that the FIM $\mathcal{I}_F^{(j)}(\theta)$ is intractable for the NL-SSM. Instead, we propose to approximate $\mathcal{I}_F^{(j)}(\theta)$ for every $j \in \{1, \ldots, n_{\mathcal{V}}\}$ using particle methods [15, 28]. For brevity, we omit the superscript j corresponding to each vertex in $\mathcal{P}_{\mathcal{C}}$ and write $\mathbf{v} := v_{1:n_{\text{seq}}}$ for any vector $v_{1:n_{\text{seq}}}$.

4.1 Estimating the score function

From (2), we know that we can compute the FIM by the use of the score function. However, the score function is also intractable for a general SSM but it can be estimated using particle smoothers. The key ingredient for this is the *Fisher identity* [5] presented in Lemma 1. In the Fisher identity, $p_\theta(\mathbf{x}, \mathbf{y}|\mathbf{u})$ denotes the complete data log-likelihood for (1) given by

$$\log p_\theta(\mathbf{x}, \mathbf{y}|\mathbf{u}) = \log \mu_\theta(x_0) + \sum_{t=1}^{n_{\text{seq}}} \xi_\theta(x_{t-1:t}), \tag{11}$$
$$\xi_\theta(x_{t-1:t}) := \log f_\theta(x_t|x_{t-1}, u_{t-1}) + \log g_\theta(y_t|x_t, u_t),$$

where we make use of the notation $x_{t-1:t} = \{x_{t-1}, x_t\}$.

Lemma 1 (Fisher identity [6]) *Assume that the following holds:*

(i) For every $\theta \in \Theta$, $p_\theta(\mathbf{y}|\mathbf{u})$ is positive and finite.
(ii) For every $(\theta,\, \theta') \in \Theta \times \Theta$,

$$\int \Big| \log p_\theta(\mathbf{x}|\mathbf{y}, \mathbf{u}) \Big| p_{\theta'}(\mathbf{x}|\mathbf{y}, \mathbf{u})\, d\mathbf{x},$$

is finite.

(iii) $\ell_\theta(\mathbf{y}|\mathbf{u})$ is continuously differentiable on Θ.
(iv) For every $\theta' \in \Theta$, the function $\mathcal{L}_{\theta'} \colon \Theta \to \mathbb{R}$ defined as

$$\mathcal{L}_{\theta'}(\theta) := -\int \log p_\theta(\mathbf{x}|\mathbf{y}, \mathbf{u}) p_{\theta'}(\mathbf{x}|\mathbf{y}, \mathbf{u})\, d\mathbf{x},$$

is continuously differentiable on Θ. In addition, $\mathcal{L}_{\theta'}(\theta)$ is finite for any $(\theta,\, \theta') \in \Theta \times \Theta$, and

$$\nabla_\theta \int \log p_\theta(\mathbf{x}|\mathbf{y}, \mathbf{u}) p_{\theta'}(\mathbf{x}|\mathbf{y}, \mathbf{u})\, d\mathbf{x} = \int \nabla_\theta \log p_\theta(\mathbf{x}|\mathbf{y}, \mathbf{u}) p_{\theta'}(\mathbf{x}|\mathbf{y}, \mathbf{u})\, d\mathbf{x}.$$

Then,

$$\mathcal{S}(\theta') = \int \nabla_\theta \log p_\theta(\mathbf{x}, \mathbf{y}|\mathbf{u})\big|_{\theta=\theta'} p_{\theta'}(\mathbf{x}|\mathbf{y}, \mathbf{u})\, d\mathbf{x}\,. \tag{12}$$

■

Proof. We refer to [6] for a proof of this lemma. ■

Using (11) and (12), we arrive at the estimator

$$\mathcal{S}(\theta) = \sum_{t=1}^{n_{\text{seq}}} \underbrace{\int \nabla_\theta\, \xi_\theta(x_{t-1:t}) p_\theta(x_{t-1:t}|\mathbf{y}, \mathbf{u})\, \mathrm{d}x_{t-1:t}}_{:=\mathcal{S}_t(\theta)}\,. \tag{13}$$

Here, we require an estimate of the *two-step smoothing distribution* $p_\theta(x_{t-1:t}|\mathbf{y}, \mathbf{u})$. This can be provided by a so-called empirical distribution,

$$\widehat{p}_\theta(\mathrm{d}x_{t-1:t}|\mathbf{y}) := \sum_{i=1}^{N} w_t^{(i)} \delta_{x_{t-1:t}^{(i)}}(\mathrm{d}x_{t-1:t}), \tag{14}$$

where $x_t^{(i)}$ and $w_t^{(i)}$ denote particle i and its normalized weight at time t. Here, $\{x_t^{(i)}, w_t^{(i)}\}_{t=1}^{n_{\text{seq}}}$ denotes the *particle system* generated by a particle filter and $\delta_{x'}(x)$ denotes the Dirac measure located at $x = x'$.

In this paper, we generate the particle system using the bootstrap particle filter (bPF) [14, Section 1.3.3]. Algorithm 1 presents the pseudo code for the bPF, where $\mathsf{Cat}(\{p^{(i)}\}_{i=1}^N)$ denotes the categorical distribution with $p^{(i)}$ denoting the probability of selecting element i. However, the estimator in (14) based on the bPF often suffers from poor accuracy. This is due to problems with *path degeneracy*, see e.g. [15]. To mitigate this problem, we make use of a particle smoother that introduces a backward sweep after the forward run of the particle filter.

Here, we use the forward-filtering backwards simulator (FFBSi) with rejection sampling and early stopping [13, 41] presented in Algorithm 2. FFBSi makes

Algorithm 1 Bootstrap particle filter (bPF)

INPUTS: An SSM (1), $y_{1:n_{\text{seq}}}$ (observations), $u_{1:n_{\text{seq}}}$ (inputs), $N \in \mathbb{N}$ (number of particles).
OUTPUT: $\{x_t^{(i)}, w_t^{(i)}\}_{i=1}^N$, $t = 1, \ldots, n_{\text{seq}}$.
All operations are carried out over $i, j = 1, \ldots, N$.

1: Sample $x_0^{(i)} \sim \mu_\theta(x_0)$ and set $w_0^{(i)} = 1/N$.
2: **for** $t = 1$ to n_{seq} **do**
3: (Resampling) Sample $a_t^{(i)} \sim \mathsf{Cat}\Big(\{\tilde{w}_{t-1}^{(j)}\}_{j=1}^N\Big)$.
4: (Propagation) Sample $x_t^{(i)} \sim f_\theta\Big(x_t^{(i)}\Big|x_{t-1}^{a_t^{(i)}}, u_t\Big)$.
5: Set $x_{0:t}^{(i)} = \Big\{x_{0:t-1}^{a_t^{(i)}}, x_t^{(i)}\Big\}$.
6: (Weighting) Calculate $\widetilde{w}_t^{(i)} = g_\theta\Big(y_t\Big|x_t^{(i)}, u_t\Big)$.
7: Normalize $\widetilde{w}_t^{(i)}$ (over i) to obtain $w_t^{(i)}$.
8: **end for**

use of the output from a run of the bPF. The parameter ρ (line 13 in Algorithm 2) is an upper bound for the pdf f_θ in the sense that $f_\theta(x_t|x_{t-1}) \leq \rho$ for all $t \in \{1, \ldots, n_{\text{seq}}\}$. The computational complexity of FFBSi is of order $\mathcal{O}(NMn_{\text{seq}})$, where N and M denote the number of filter and smoother particles, respectively.

The statistical properties of the FFBSi are studied by [13] under some regularity assumptions. These include that the pdfs derived from the SSM (1) are bounded and that $\xi_\theta(x_{t-1:t})$ is measurable. These two conditions are usually fulfilled when the SSM is defined by densities f and g as in (1). Given these assumptions, it is possible to show that the error in the estimates obtained from the algorithm are bounded [13, Corollary 6] and obeys a central limit theorem [13, Corollary 9]. This implies that the estimates of the score function are (strongly) consistent and the variance decreases as $N, M \to \infty$. In practice, increasing N and M also increases the computational cost. We return to investigate the finite-data properties of FFBSi in Section 7.1.

Note that there are many alternative particle smoothers that could be useful, see [28] for a recent survey. In our previous work [44], we employed a fixed-lag (FL) particle smoother to estimate the score function. The main advantage with using FFBSi compared with FL is the consistency of the former. In comparison, the FL smoother gives biased estimates even when $N \to \infty$. However, the FL smoother has a computational complexity of $\mathcal{O}(Nn_{\text{seq}})$, which is smaller than that of the FFBSi smoother.

4.2 The resulting estimator

From (13), we note that $\mathcal{S}(\theta)$ can be written as a sum of conditional scores $\mathcal{S}_t(\theta)$. This can be seen as a martingale representation since each conditional score is conditionally independent given the past. Using results presented in [31, 39], we can compute an estimate of the FIM by

Algorithm 2 Fast forward-filtering backward-simulator with early stopping (fFFBSi-ES)

INPUTS: Inputs to Algorithm 1, $M \in \mathbb{N}$ (No. backward trajectories), $N_{\text{limit}} \in \mathbb{N}$ (Limit for when to stop using rejection sampling), $\rho > 0$.
OUTPUT: $\widehat{\mathcal{I}}_F(\theta)$ (estimate of the FIM).

1: Run Algorithm 1 to obtain the particle system $\Big\{x_t^{(i)}, w_t^{(i)}\Big\}_{i=1}^N$ for $t = 1, \ldots, n_{\text{seq}}$.
2: Sample $\big\{b_{n_{\text{seq}}}(j)\big\}_{j=1}^M \sim \mathsf{Cat}\big(\{w_{n_{\text{seq}}}^{(i)}\}_{i=1}^N\big)$.
3: Set $\tilde{x}_{n_{\text{seq}}}^{(j)} = x_{n_{\text{seq}}}^{(b_{n_{\text{seq}}}(j))}$ for $j = 1, \ldots, M$.
4: **for** $t = n_{\text{seq}} - 1$ **to** 1 **do**
5: $L \leftarrow \{1, \ldots, M\}$.
6: {Rejection sampling until N_{limit} trajectories remain.}
7: **while** $|L| \geq N_{\text{limit}}$ **do**
8: $n \leftarrow \mathsf{Card}(L)$.
9: $\delta \leftarrow \emptyset$.
10: Sample $\big\{I(k)\big\}_{k=1}^n \sim \mathsf{Cat}\big(\{w_t^{(i)}\}_{i=1}^N\big)$.
11: Sample $\big\{U(k)\big\}_{k=1}^n \sim \mathsf{Uniform}([0,1])$.
12: **for** $k = 1$ **to** n **do**
13: **if** $U(k) \leq f\big(\tilde{x}_{t+1}^{L(k)}|x_t^{I(k)}\big)/\rho$ **then**
14: $b_t(L(k)) \leftarrow I(k)$.
15: $\delta \leftarrow \delta \cup \{L(k)\}$.
16: **end if**
17: **end for**
18: $L \leftarrow L \setminus \delta$.
19: **end while**
20: {Use standard FFBSi for the remaining trajectories.}
21: **for** $j \in L$ **do**
22: Compute $\tilde{w}_{t|n_{\text{seq}}}^{(i,j)} \propto w_t^{(i)} f\big(\tilde{x}_{t+1}^{(j)}|x_t^{(i)}\big)$ for $i = 1, \ldots, N$.
23: Normalize the smoothing weights $\big\{\tilde{w}_{t|n_{\text{seq}}}^{(i,j)}\big\}_{i=1}^N$.
24: Draw $b_t(j) \sim \mathsf{Cat}\Big(\big\{\tilde{w}_{t|n_{\text{seq}}}^{(i,j)}\big\}_{i=1}^N\Big)$.
25: **end for**
26: Set $\tilde{x}_{t:n_{\text{seq}}}^{(j)} = \Big\{x_t^{b_t(j)}, \tilde{x}_{t+1:n_{\text{seq}}}^{(j)}\Big\}$ for $j = 1, \ldots, M$.
27: Estimate the score function at t using (13) by
$$\widehat{\mathcal{S}}_t(\theta) = \frac{1}{M}\sum_{j=1}^M \nabla_\theta \xi_\theta\Big(\tilde{x}_{t:t+1}^{(j)}\Big).$$
28: **end for**
29: Compute $\widehat{\mathcal{I}}_F(\theta)$ using (15).

$$\widehat{\mathcal{I}}_F(\theta) = \frac{1}{n_{\text{seq}}}\left\{\sum_{t=1}^{n_{\text{seq}}}\Big[\widehat{\mathcal{S}}_t(\theta)\Big]\Big[\widehat{\mathcal{S}}_t(\theta)\Big]^\top - \frac{1}{n_{\text{seq}}}\Big[\widehat{\mathcal{S}}(\theta)\Big]\Big[\widehat{\mathcal{S}}(\theta)\Big]^\top\right\}, \tag{15}$$

where

$$\widehat{\mathcal{S}}(\theta) := \sum_{t=1}^{n_{\text{seq}}} \widehat{\mathcal{S}}_t(\theta)\,. \tag{16}$$

We note that the estimator (15) is based on the output $y_{1:n_{\text{seq}}}$ generated from (1) using $\theta \in \Theta$ and $u_{1:n_{\text{seq}}}$ as input.

There are other alternatives for computing an estimate

of the FIM. One alternative approach is to make use of the Louis identity [6]. Another alternative is to compute the FIM using the sample covariance matrix of the score estimates (13) as proposed by [44]. The main advantage of the estimator (15) is that it only requires running a single particle smoother for estimating the score function. The approach based on the sample covariance matrix requires hundreds or thousands of runs to marginalise out the effect of the noisy realisation. That is, to estimate the expectation operator in (13) with respect to $\mathbf{y}$.

To see the later, we analyze the variance of this estimator using the martingale difference property when using Algorithm 2 to estimate the score function. To this end, we note that the expected value of (15) is given by

$$n_{\text{seq}}\,\mathbf{E}\left\{\widehat{\mathcal{I}}_F(\theta)\right\} = \frac{n_{\text{seq}}-1}{n_{\text{seq}}}\mathbf{E}\left\{\sum_{t=1}^{n_{\text{seq}}}\left[\widehat{\mathcal{S}}_t(\theta)\right]\left[\widehat{\mathcal{S}}_t(\theta)\right]^\top\right\}, \tag{17}$$

where the expected value in (17) is with respect to the stochastic processes characterizing the nonlinear state space model (1).

Before proceeding, let $\widehat{\mathcal{S}}_\ell(\theta)$ and $\widehat{\mathcal{I}}_{F,\ell m}(\theta)$ denote the ℓ-th and (ℓ, m) entry of $\widehat{\mathcal{S}}(\theta)$ and $\widehat{\mathcal{I}}_F(\theta)$, respectively. Then, the variance of the (ℓ, m) entry of (15) is given by

$$\begin{aligned} n_{\text{seq}}^2\,\text{Var}\left\{\widehat{\mathcal{I}}_{F,\ell m}(\theta)\right\} = \frac{(n_{\text{seq}}-1)^2}{n_{\text{seq}}^2}\,\text{Var}\left\{f_1(n_{\text{seq}},\ell,m)\right\} \\ + \frac{2}{n_{\text{seq}}}f_4(n_{\text{seq}},\ell,m) \\ + \frac{1}{n_{\text{seq}}^2}f_5(n_{\text{seq}},\ell,m)\,, \end{aligned} \tag{18}$$

with

$$\begin{aligned} f_4(n_{\text{seq}},\ell,m) := \mathbf{E}\left\{f_1^2(n_{\text{seq}},\ell,m)\right\} \\ - \mathbf{E}\left\{\sum_{t=1}^{n_{\text{seq}}} f_3^2(t,\ell,m)\right\}, \end{aligned} \tag{19}$$

$$\begin{aligned} f_5(n_{\text{seq}},\ell,m) := \mathbf{E}\left\{f_2^2(n_{\text{seq}},\ell,m)\right\} \\ - \mathbf{E}\left\{f_1^2(n_{\text{seq}},\ell,m)\right\}, \end{aligned} \tag{20}$$

and

$$f_1(n_{\text{seq}},\ell,m) := \sum_{t=1}^{n_{\text{seq}}}\widehat{\mathcal{S}}_{t,\ell}(\theta)\widehat{\mathcal{S}}_{t,m}(\theta)\,, \tag{21}$$

$$f_2(n_{\text{seq}},\ell,m) := \widehat{\mathcal{S}}_\ell(\theta)\widehat{\mathcal{S}}_m(\theta)\,, \tag{22}$$

$$f_3(t,\ell,m) := \widehat{\mathcal{S}}_{t,\ell}(\theta)\widehat{\mathcal{S}}_{t,m}(\theta)\,. \tag{23}$$

We refer to Appendix A for a complete derivation of Equation (18).

When the FFBSi is used to estimate the score function, we have by [13, Theorem 11] that the variance (18) is bounded for a fixed n_{seq}. Moreover, the estimator is consistent as $n_{\text{seq}}, N, M \to \infty$, which follows from [13, Corollary 9]. As previously mentioned, we investigate the finite sample accuracy of the estimate in Section 7.1.

We can carry out a similar analysis for the alternative approach based on the sample covariance matrix. Here, we estimate the score function $\widehat{\mathcal{S}}^{(k)}$ using particle methods over $k \in \{1, \ldots, K\}$ different noisy realizations based on a single realization of the input. The estimate of the FIM is computed by

$$\widehat{\mathcal{I}}_F(\theta) = \frac{1}{Kn_{\text{seq}}}\sum_{k=1}^{K}\left[\widehat{\mathcal{S}}^{(k)}(\theta)\right]\left[\widehat{\mathcal{S}}^{(k)}(\theta)\right]^\top. \tag{24}$$

The variance of this estimator is given by

$$n_{\text{seq}}^2\,\text{Var}\left\{\widehat{\mathcal{I}}_{F,\ell m}(\theta)\right\} = \frac{1}{K}\,\text{Var}\left\{\widehat{\mathcal{S}}_\ell^{(k)}(\theta)\widehat{\mathcal{S}}_m^{(k)}(\theta)\right\}, \tag{25}$$

for some $k \in \{1, \ldots, K\}$. This implies that the accuracy of (24) increases with the number of realizations K, provided that the variance of each element in $\left[\widehat{\mathcal{S}}^{(k)}(\theta)\right]\left[\widehat{\mathcal{S}}^{(k)}(\theta)\right]^\top$ is bounded, which again is satisfied by FFBSi as discussed above. Therefore, we conclude that the variance of the estimator (24) is bounded as $n_{\text{seq}}, N, M \to \infty$ and it approaches zero as $K \to \infty$.

As discussed above, the main benefit of using (15) instead of (24) is a smaller computational cost. This results from that only one run of the particle smoother is required for the former compared with K runs for the latter. In practice, this decreases the computational time for a single estimate of the FIM from over a day to about an hour. Moreover, it is difficult to establish theoretically which of the two estimators has better accuracy. Using numerical evaluations, we have observed that the new estimator outperforms the latter in terms of both accuracy and computational cost.

5 Final input design method

Under the considered approximations, the FIM (5) can be expressed as a convex combination of the information matrices evaluated at the vertex points of $\mathcal{P}_\mathcal{C}$ given by (10). Hence, the proposed method to design input

Algorithm 3 New input design method

INPUTS: $\mathcal{C}$ (input values), n_m (number of input samples), $\mathbf{P}$ (probability measure over Θ), and N_s (number of samples over Θ).
OUTPUT: $\gamma^\star$ (optimal weighting of the basis inputs).

1: Sample $\{\theta_i\}_{i=1}^{N_s}$ from Θ according to $\mathbf{P}$.
2: Compute all the elementary cycles of $\mathcal{G}_{\mathcal{C}^{n_m-1}}$ using, e.g., [25, pp. 79–80], [42, pp. 157].
3: Compute all the prime cycles of $\mathcal{G}_{\mathcal{C}^{n_m}}$ from the elementary cycles of $\mathcal{G}_{\mathcal{C}^{n_m-1}}$ as explained in [51, Lemma 4]. Denote by $\{v_j\}_{j=1}^{n_\mathcal{V}}$ the pmfs associated with the prime cycles of $\mathcal{G}_{\mathcal{C}^{n_m}}$.
4: Generate the input signals $u^j_{1:n_{\text{seq}}}$ from the prime cycles of $\mathcal{G}_{\mathcal{C}^{n_m}}$, for each $j \in \{1, \ldots, n_\mathcal{V}\}$.
5: **for** $i = 1$ to N_s **do**
6: Execute Algorithm 2 based on θ_i and $\{u^j_{1:n_{\text{seq}}}\}_{j=1}^{n_\mathcal{V}}$ to obtain $\{\widehat{\mathcal{I}}_F^{(j)}(\theta_i)\}_{j=1}^{n_\mathcal{V}}$.
7: **end for**
8: Solve the optimization problem (26) to obtain $\gamma^\star$.

signals in $\mathcal{C}^{n_m}$ for the identification of NL-SSMs is summarized in Algorithm 3. The method solves an approximation of Problem 1, which can be written as

$$\begin{aligned} \gamma^\star = \arg \max_{\gamma=\{\alpha_j\}_{j=1}^{n_\mathcal{V}}} \quad & \widehat{\mathcal{R}}(\mathcal{H}(\mathcal{I}_F^{\text{app}}(\gamma,\, \theta),\, \theta)) \\ \text{subject to} \quad & \mathcal{I}_F^{\text{app}}(\gamma,\, \theta_i) = \sum_{j=1}^{n_\mathcal{V}} \alpha_j\, \widehat{\mathcal{I}}_F^{(j)}(\theta_i)\,, \\ & \qquad\qquad \text{for all } i \in \{1, \ldots, N_s\} \\ & \alpha_j \geq 0\,, \text{ for all } j \in \{1, \ldots, n_\mathcal{V}\} \\ & \sum_{j=1}^{n_\mathcal{V}} \alpha_j = 1\,, \end{aligned} \tag{26}$$

where $\widehat{\mathcal{R}}$ denotes the approximation of the function $\mathcal{R}$ when the set Θ is approximated by $\{\theta_i\}_{i=1}^{N_s}$. The implementation of $\widehat{\mathcal{R}}$ for the cost functions considered in this article follows the solution presented for issue (A) in Section 2. Thus, for $\mathcal{R} = \mathbf{E}_\theta\{\cdot\}$ we have

$$\widehat{\mathcal{R}}(\mathcal{H}(\mathcal{I}_F^{\text{app}}(\gamma,\, \theta),\, \theta)) = \frac{1}{N_s} \sum_{i=1}^{N_s} \mathcal{H}(\mathcal{I}_F^{\text{app}}(\gamma,\, \theta_i),\, \theta_i)\,, \tag{27}$$

and for $\mathcal{R} = \min_{\theta\in\Theta}\{\cdot\}$ we obtain

$$\widehat{\mathcal{R}}(\mathcal{H}(\mathcal{I}_F^{\text{app}}(\gamma,\, \theta),\, \theta)) = \min_{\theta\in\{\theta_i\}_{i=1}^{N_s}} \mathcal{H}(\mathcal{I}_F^{\text{app}}(\gamma,\, \theta),\, \theta)\,. \tag{28}$$

Algorithm 3 computes the vector $\gamma^\star = \{\alpha_j^\star\}_{j=1}^{n_\mathcal{V}}$ which defines the optimal pmf p_u^{opt} as a convex combination of the measures associated with the elements in $\mathcal{V}_{\mathcal{P}_\mathcal{C}}$, according to

$$p_u^{\text{opt}} = \sum_{j=1}^{n_\mathcal{V}} \alpha_j^\star\, v_j\,. \tag{29}$$

Algorithm 4 Design of a transition probability matrix

INPUTS: A pmf $p \in \mathcal{P}_\mathcal{C}$, and n_m.
OUTPUT: A transition probability matrix A with stationary distribution p.

1: For each $r \in \mathcal{C}^{n_m}$, define

$$\mathcal{N}_r := \{l \in \mathcal{C}^{n_m} : (l,\, r) \in \mathcal{E}\}\,. \tag{30}$$

In other words, $\mathcal{N}_r$ is the set of ancestors of r, where $\mathcal{E}$ is the set of edges of $\mathcal{G}_{\mathcal{C}^{n_m}}$.
2: For each r, $l \in \mathcal{C}^{n_m}$, let

$$A_{rl} = \begin{cases} \frac{p(r)}{\sum_{k\in\mathcal{N}_r} p(k)}\,, & \text{if } l \in \mathcal{N}_r \text{ and} \\ & \sum_{k\in\mathcal{N}_r} p(k) \neq 0\,, \\ \frac{1}{|\mathcal{N}_r|}\,, & \text{if } l \in \mathcal{N}_r \text{ and} \\ & \sum_{k\in\mathcal{N}_r} p(k) = 0\,, \\ 0\,, & \text{otherwise.} \end{cases} \tag{31}$$

6 Input signal generation

In this section we present a procedure to generate an input sequence $u_{1:n_{\text{seq}}}$ from a given pmf $p(u_{1:n_m}) \in \mathcal{P}_\mathcal{C}$, which has been introduced in [46]. The method associates $\mathcal{G}_{\mathcal{C}^{n_m}}$ with the discrete-time Markov chain [12]

$$\pi_{k+1} = A\, \pi_k\,, \tag{32}$$

where $A \in \mathbb{R}^{\mathcal{C}^{n_m}\times\mathcal{C}^{n_m}}$ is a transition probability matrix [6], and $\pi_k \in \mathbb{R}^{\mathcal{C}^{n_m}}$ is the state vector. We note that every entry of $\pi_k \in \mathbb{R}^{\mathcal{C}^{n_m}}$ corresponds to an element in $\mathcal{C}^{n_m}$.

Based on the previous association, $p(u_{1:n_m})$ is $\Pi^s \in \mathbb{R}^{\mathcal{C}^{n_m}}$, the stationary distribution of the Markov chain (32). Thus, an input sequence $u_{1:n_{\text{seq}}}$ distributed according to $p(u_{1:n_m})$ can be obtained by running a Markov chain having $p(u_{1:n_m})$ as its stationary distribution.

Algorithm 4 presents a method to design a transition probability matrix A for $\mathcal{G}_{\mathcal{C}^{n_m}}$ with Π^s as stationary distribution. Here we will only make use of this result to generate $u_{1:n_{\text{seq}}}$ with distribution given by the optimal pmf $p_u^{\text{opt}}(u_{1:n_m})$. We refer to [46] for more details about

[6] Given a set of finite cardinality X, we denote by $\mathbb{R}^{X\times X}$ the matrices with real entries, with dimensions given by the cardinality of X.

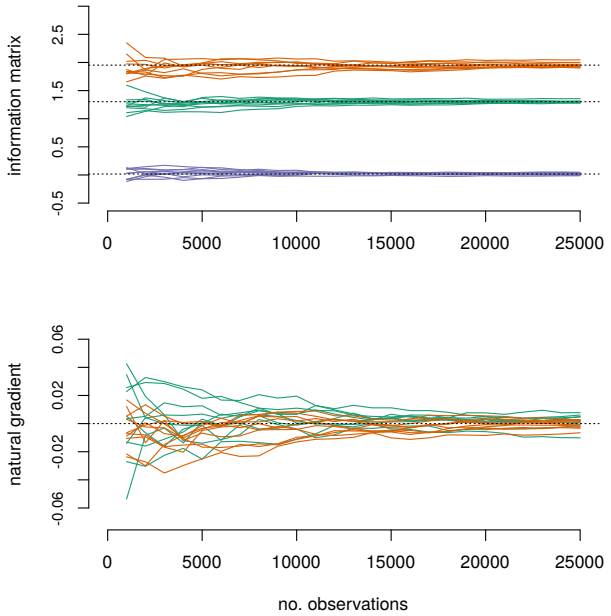

Fig. 2. Upper: the information matrix element for ϕ (orange), σ_v (green) and cross-term between ϕ and σ_v (purple). Lower: the natural gradient for different n_{seq}. The results are computed using 25 Monte Carlo simulations.

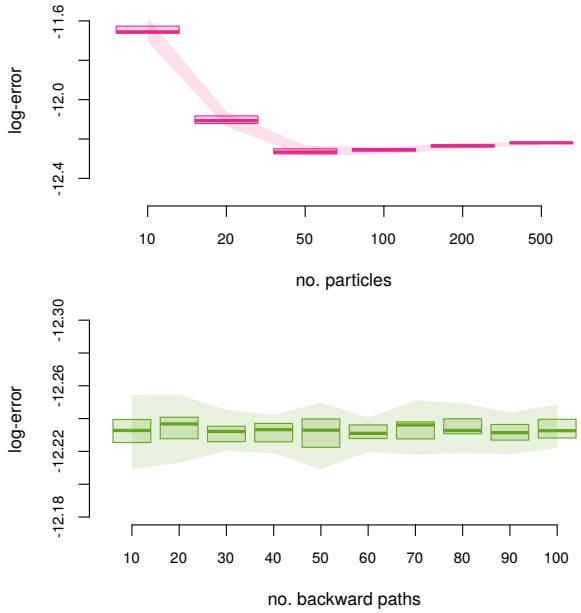

Fig. 3. The logarithm of the error in Frobenius norm of the estimate of the information matrix when varying N (upper) and M (lower). The plots show the results from 25 Monte Carlo simulations and the shaded areas indicate the span between the largest and smallest error.

Algorithm 4, where the correctness of the designed transition probability matrix is established.

7 Numerical illustrations

In this section, we present numerical simulations to illustrate some aspects of the proposed input design method in two SSMs. In the first illustration, we make use of a linear Gaussian state space (LGSS) model to evaluate the accuracy of the estimator for the FIM and the impact of the design parameters in Algorithm 2. We also compare the solution obtained for the input design problem to some standard input signals as a sanity check of how the proposed method works.

In the second illustration, we consider an NL-SSM, which is more challenging from an input design perspective. All implementation details and settings for the algorithm are presented in Appendix B.

7.1 Accuracy of information matrix estimation

Consider the LGSS model given by

$$x_t|x_{t-1} \sim \mathcal{N}\big(x_t; \phi x_{t-1} + u_{t-1}, \sigma_v^2\big), \tag{33a}$$
$$y_t|x_t \sim \mathcal{N}\big(y_t; x_t, 0.1^2\big), \tag{33b}$$

where the parameter vector is $\theta = \{\phi, \sigma_v\}$ with the state persistence $\phi \in (-1, 1)$ and the standard deviation of the innovations $\sigma_v \in \mathbb{R}_+$.

From Section 4.2, we know that the estimates of the FIM obtained by (15) are consistent. However, in practice we do not have an infinite amount of data and therefore the properties of the estimate can only be evaluated using numerical experiments. To this end, we make use of a Kalman method, which allows for computing the score function exactly for the LGSS model. Hence, we can isolate the influence of n_{seq} on the accuracy of the estimate.

Figure 2 presents the estimate and the resulting natural gradient (the gradient scaled by the inverse FIM) when varying n_{seq} for 25 Monte Carlo executions. We conclude that the estimator stabilizes after about $n_{\text{seq}} \approx 15 \cdot 10^3$ observations. Furthermore, we observe that the natural gradients are almost zero at this value of n_{seq}, indicating that the maximum likelihood estimate of parameter vector is indeed θ_0.

For a general SSM, we cannot make use of Kalman methods and have to resort to approximations using e.g., particle methods. In this setting, we would like to investigate the influence of N and M in Algorithm 2 on the accuracy of the estimates, when n_{seq} is fixed to $15 \cdot 10^3$ based on the results above. Figure 3 presents the log-error in the Frobenius norm of the information matrix when varying N and M. In the former case, we fix $M = \lfloor N/4 \rfloor$ (the closest integer to $N/4$ from below) and vary N. We conclude that 200 particles are enough to obtain reasonable accuracy in this model. The increase of the log-error for $N \geq 50$ is probably due to the randomness of the Monte Carlo simulations. In the latter case, we fix $N = 200$

Table 1
Parameter estimates with the 1.96 times the standard deviation (parenthesis), the log-MSE and $\mathsf{LD}(\mathcal{I})$ in the LGSS model computed from 40 independent runs of the EM algorithm using different (but fixed) input signals. Bold face marks the best values and KM/PM indicate that Kalman methods/Algorithm 2 is used to compute the Q-function in the EM algorithm. The true parameter values for θ are indicated between square brackets.

	Kalman method (KM)					Particle method (PM)				
Input	$\widehat{\phi}$ [0.5]	$\mathsf{MSE}(\widehat{\phi})$	$\widehat{\sigma}_v$ [0.1]	$\mathsf{MSE}(\widehat{\sigma}_v)$	$\mathsf{LD}(\mathcal{I})$	$\widehat{\phi}$ [0.5]	$\mathsf{MSE}(\widehat{\phi})$	$\widehat{\sigma}_v$ [0.1]	$\mathsf{MSE}(\widehat{\sigma}_v)$	$\mathsf{LD}(\mathcal{I})$
none	0.50 (0.02)	-9.06	0.10 (0.002)	-13.25	-22.53	0.30 (0.02)	-3.21	0.11 (0.002)	**-8.89**	-16.63
constant	0.50 (0.00)	**-15.33**	0.10 (0.002)	-13.47	**-28.83**	0.50 (0.00)	-14.15	0.12 (0.002)	-8.33	-23.56
uniform	0.50 (0.00)	-13.02	0.10 (0.002)	**-13.49**	-26.53	0.49 (0.00)	-10.31	0.11 (0.002)	-8.68	-21.56
binary	0.50 (0.00)	-14.06	0.10 (0.002)	-13.43	-27.69	0.50 (0.00)	-12.26	0.11 (0.002)	-8.52	-22.67
mean case	0.50 (0.00)	-14.56	0.10 (0.002)	-13.46	**-28.83**	0.50 (0.00)	-14.15	0.12 (0.002)	-8.33	-23.56
worst case	0.50 (0.00)	-14.52	0.10 (0.002)	-13.48	**-28.83**	0.50 (0.00)	**-14.18**	0.12 (0.001)	-8.34	**-23.67**

and vary the number of backward paths M. We conclude from this that the accuracy of the estimates is quite robust to the choice of the number of backward trajectories, and that $M = 10$ seems to be a reasonable choice for this example.

7.2 Input design for the LGSS model

The analysis above provides some guidance in selecting n_{seq}, N and M to obtain reasonable estimates of the FIM for the LGSS model. Therefore, we are ready to apply Algorithm 3 to construct an input sequence with the aim to accurately estimate $\theta = \{\phi, \sigma_v\}$ in (33). We make use of the EM algorithm proposed by [38] to estimate the parameters of the model when different inputs are applied. This is done to investigate if the designed input actually increases the accuracy of the estimates.

The main time spent by Algorithm 3 for designing an input for the LGSS model is Line 6 (5 minutes for each realisation i on a standard laptop computer from 2012) and Line 8 (1 minute). Note that it is possible to run Line 6 in parallel to decrease the computational time.

Table 1 presents the logarithm of the mean squared error (MSE) of the parameter estimates for different inputs: none ($u_t \equiv 0$), constant ($u_t \equiv 1$), uniform ($u_t \sim \mathsf{Uniform}([-1, 1])$) and binary ($u_t \sim 1 - 2 \cdot \mathsf{Bernoulli}(0.5)$). As mentioned above, Kalman methods can be applied to compute the score function (and the Q-function in the EM algorithm) exactly. Hence, Kalman methods illustrate the optimal performance of the proposed algorithm when N and M tends to infinity. Moreover, we make use of $\mathcal{R} = \mathbf{E}_\theta\{\cdot\}$ (mean case), and $\mathcal{R} = \min_{\theta\in\Theta}\{\cdot\}$ (worst case) to include robustness to the uncertainty in the parameters. The input design technique with particle methods (PM) attains similar log-MSE values than those obtained with a constant signal ($u_t = 1$), which is the optimal choice.

To complement the results in this example, we compute $\mathsf{LD}(\mathcal{I}) := \log\{\det(\mathcal{I}_{\text{MSE}}(\theta_0))\}$ for the different inputs and methods, where $\mathcal{I}_{\text{MSE}}(\theta_0)$ corresponds to the sample MSE matrix

$$\mathcal{I}_{\text{MSE}}(\theta_0) := \frac{1}{N_{\text{est}}} \sum_{i=1}^{N_{\text{est}}} (\widehat{\theta}_i - \theta_0)(\widehat{\theta}_i - \theta_0)^\top, \qquad (34)$$

where $\{\widehat{\theta}_i\}_{i=1}^{N_{\text{est}}}$ are the estimated parameters for the different inputs and methods in Table 1 ($N_{\text{est}} = 40$). The results for $\log\{\det(\mathcal{I}_{\text{MSE}}(\theta_0))\}$ are summarized in the rightmost column of Table 1. From this table we conclude that the proposed method attains the maximum value of $\log\{\det(\mathcal{I}_{\text{MSE}}(\theta_0))\}$ among the inputs considered in this example.

7.3 Input design for a nonlinear model

The second example is an NL-SSM given by

$$x_t | x_{t-1} \sim \mathcal{N}\Big(x_t; \frac{1}{\gamma + x_{t-1}^2} + u_{t-1}, 0.1^2\Big), \qquad (35a)$$

$$y_t | x_t \sim \mathcal{N}\Big(y_t; \beta x_t^2, 1^2\Big), \qquad (35b)$$

where the parameters are $\theta = \{\gamma, \beta\}$. The sign of the state is lost due to the term x_t^2 in the measurement process, which implies that two different values for the state can equally represent a value for y_t. Moreover, this and the NL state process prohibit the use of Kalman methods for this model and therefore only Algorithm 2 is considered. We apply the same approach as in Section 7.2 to construct an input to excite the system and then apply an EM algorithm to estimate the parameters. The computational time for executing Line 6 in Algorithm 3 increases to 38 minutes in this model.

Figure 4 presents estimates at each iteration of the EM algorithm from 35 independent runs on the same input/output data. The parameter β seems easier to estimate than the parameter γ, which is probably due to

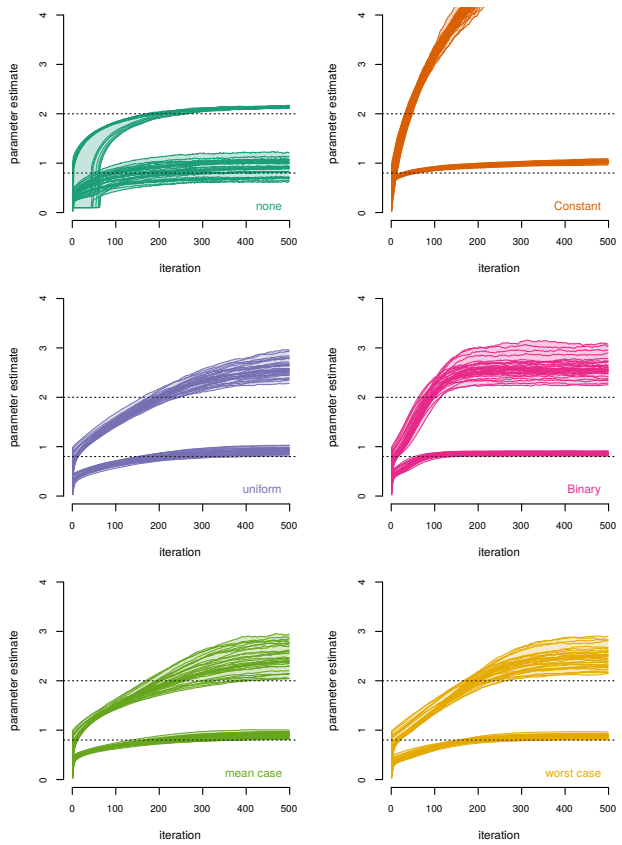

Fig. 4. The evolution of the parameter estimates for (35) obtained from 35 runs of the EM algorithm with random initializations and different inputs. The dotted lines indicate the values $\gamma_0 = 2$ and $\beta_0 = 0.8$ used to generate the data from the model. The shaded area indicates the span between the maximum and minimum values.

that the model is linear in β. Another interesting result from this figure can be seen by comparing the estimates for the different inputs. We observe that there exists a trade-off on the accuracy achieved for the estimates of γ and β. While a zero input gives more accurate estimates for γ than using a nonzero input, the accuracy for the estimates of β is increased by including a non-zero input.

We can also analyze the overall performance of the estimates for different inputs. In Table 2 the parameter estimates and the corresponding log-MSE for different inputs is presented. We note that the log-MSE is computed using θ_0 and not the maximum likelihood estimate of θ. We also present the half-length of the bootstrapped 95% confidence intervals (CIs) for the estimated parameters in parentheses. These are obtained after 10^3 resamplings and using the adjusted bootstrap percentile [8].

Table 2 shows that no overall best input signal can be determined from this experiment, as the decision depends on the relative importance of the two parameters. In the case of the optimal inputs designed by Algorithm 3, we see that we can improve the log-MSE for the estimate

Table 2
Parameter estimates for the NL-SSM model in analogue with the particle methods in Table 1 using 35 independent runs of the EM algorithm.

Input	$\widehat{\gamma}$ [2]	$\mathsf{MSE}(\widehat{\gamma})$	$\widehat{\beta}$ [0.8]	$\mathsf{MSE}(\widehat{\beta})$	$\mathsf{LD}(\mathcal{I})$
none	2.14 (**0.01**)	**-3.99**	0.90 (0.05)	-3.46	-7.76
constant	6.60 (0.23)	3.07	1.03 (**0.01**)	-2.96	-3.84
uniform	2.59 (0.05)	-0.97	0.91 (0.02)	-4.14	-7.31
binary	2.59 (0.06)	-0.95	0.86 (**0.01**)	**-5.38**	-7.94
mean case	2.47 (0.09)	-1.26	0.89 (0.02)	-4.47	-8.37
worst case	2.47 (0.06)	-1.36	0.88 (**0.01**)	-4.93	**-8.47**

of γ when compared to the uniform and binary inputs. As expected from the previous discussion, the improvement in the log-MSE for $\widehat{\gamma}$ is achieved by degrading the log-MSE for $\widehat{\beta}$ when compared to the binary input. However, the log-MSE of $\widehat{\beta}$ for the designed inputs is better than the one obtained with no input.

As a final comparison, the rightmost column of Table 2 presents the value of $\mathsf{LD}(\mathcal{I}) = \log\{\det(\mathcal{I}_{\mathrm{MSE}}(\theta_0))\}$ for the different inputs, with $\mathcal{I}_{\mathrm{MSE}}(\theta_0)$ given by (34) using $\{\widehat{\theta}_i\}_{i=1}^{N_{\mathrm{est}}}$ ($N_{\mathrm{est}} = 35$). The results presented in this table show that the designed inputs reduce the volume of the uncertainty set associated with $\widehat{\theta}$ when compared to the other inputs. Hence, the proposed method designs an input that optimally distributes the uncertainty over the estimated parameters in the sense that the volume of the associated uncertainty set is minimized.

In conclusion, the input design method presented here can be employed to provide a better trade-off between the accuracies of the parameter estimates in this challenging example, when compared to standard inputs.

8 Conclusions

We have proposed a robust input design method for the identification of NL-SSMs. It is based on designing an input signal as a realization of a stationary process maximizing a scalar cost function of the FIM. Since the model parameters are unknown a priori, the optimization of the experiment considers a measure of the uncertainty over the space of model parameters. Furthermore, we provide numerical illustrations indicating that the designed input signal improves the accuracy of parameter estimates compared with other standard inputs.

Acknowledgements

The authors thank to the Associate Editor and the anonymous Reviewers who provided valuable suggestions to improve this article.

A Variance for the estimator (15)

We start by writing

$$n_{\text{seq}}^2 \operatorname{Var}\left\{\widehat{\mathcal{I}}_{F,\ell m}(\theta)\right\} = n_{\text{seq}}^2 \left[\mathbf{E}\left\{\widehat{\mathcal{I}}_{F,\ell m}^2(\theta)\right\} - \mathbf{E}^2\left\{\widehat{\mathcal{I}}_{F,\ell m}(\theta)\right\}\right], \quad \text{(A.1)}$$

where the expected values in (A.1) are with respect to the stochastic processes characterizing the SSM (1).

Based on (15), we compute $\mathbf{E}\left\{\widehat{\mathcal{I}}_{F,\ell m}^2(\theta)\right\}$ as

$$\begin{aligned} n_{\text{seq}}^2 \mathbf{E}\left\{\widehat{\mathcal{I}}_{F,\ell m}^2(\theta)\right\} &= \mathbf{E}\left\{f_1^2(n_{\text{seq}},\ell,m)\right\} \\ &\quad + \frac{1}{n_{\text{seq}}^2}\mathbf{E}\left\{f_2^2(n_{\text{seq}},\ell,m)\right\} \\ &\quad - \frac{2}{n_{\text{seq}}}\mathbf{E}\left\{f_1(n_{\text{seq}},\ell,m)\, f_2(n_{\text{seq}},\ell,m)\right\}, \end{aligned} \quad \text{(A.2)}$$

where $f_1(n_{\text{seq}},\ell,m)$ and $f_2(n_{\text{seq}},\ell,m)$ are given by (21) and (22), respectively.

Using the martingale difference property, we obtain

$$\mathbf{E}\left\{f_1(n_{\text{seq}},\ell,m)\, f_2(n_{\text{seq}},\ell,m)\right\} = \mathbf{E}\left\{\sum_{t=1}^{n_{\text{seq}}} f_3^2(t,\ell,m)\right\}, \quad \text{(A.3)}$$

with $f_3(t,\ell,m)$ given by (23).

Replacing (A.3) into (A.2) we have

$$\begin{aligned} n_{\text{seq}}^2 \mathbf{E}\left\{\widehat{\mathcal{I}}_{F,\ell m}^2(\theta)\right\} &= \left[1 - \frac{(n_{\text{seq}}-1)^2}{n_{\text{seq}}^2}\right]\mathbf{E}\left\{f_1^2(n_{\text{seq}},\ell,m)\right\} \\ &\quad + \frac{1}{n_{\text{seq}}^2}\mathbf{E}\left\{f_2^2(n_{\text{seq}},\ell,m)\right\} - \frac{2}{n_{\text{seq}}}\mathbf{E}\left\{\sum_{t=1}^{n_{\text{seq}}} f_3^2(t,\ell,m)\right\} \\ &\quad + \frac{(n_{\text{seq}}-1)^2}{n_{\text{seq}}^2}\mathbf{E}\left\{f_1^2(n_{\text{seq}},\ell,m)\right\}, \end{aligned} \quad \text{(A.4)}$$

where we have added and subtracted the term

$$\frac{(n_{\text{seq}}-1)^2}{n_{\text{seq}}^2}\mathbf{E}\left\{f_1^2(n_{\text{seq}},\ell,m)\right\}.$$

Finally, replacing (17) and (A.4) into (A.1) and rearranging terms, we obtain (18).

B Implementation details

LGSS model: For the Monte Carlo study in Figure 2, we generate 25 data sets with $n_{\text{seq}} = 2.5 \cdot 10^4$ observations from (33) with $\theta_0 = \{0.5, 1.0\}$, no input and a known initial zero state. For the approach based on Kalman methods, a RTS smoother [33] is used to compute the score function and the FIM is estimated by (15). For the particle method, we make use of a fully adapted particle filter (faPA; [32]) in Algorithm 2 to estimate the FIM. This reduces the computational cost and increases the accuracy compared with bPF. However, the faPF can only be implemented for a small number of SSMs.

For Algorithm 3, we make use of $n_{\text{seq}} = 15 \cdot 10^3$, $n_m = 2$, and $c_{\text{seq}} = 3$ values (-1, 0 and 1). This results in 8 different basis inputs. Moreover, we use $\mathcal{H}(\mathcal{I}_F(\theta), \theta) = \log\det(\mathcal{I}_F(\theta))$ as the scalar cost function of the FIM. For Algorithm 2, we make use of $N = 200$, $M = 10$, $N_{\text{limit}} = 3$ and $\rho = 1$. To account for the uncertainty in the parameters, we sample $N_s = 100$ realizations uniformly from the parameter space $\Theta = \{(\phi, \sigma_v) : \phi \in [0.4, 0.6], \sigma_v \in [0.8, 1.2]\}$ and design the input accordingly. In the EM algorithm, we make use of $N = 200$, $M = 20$ and run it for 150 iterations.

NL-SSM: For the Monte Carlo study in Section 7.3, we generate a single data set with $n_{\text{seq}} = 10^3$ observations from (35) using $\theta_0 = \{2, 0.8\}$. For Algorithm 3, we use $n_m = 2$, and $c_{\text{seq}} = 4$ values (-1, $-1/3$, $1/3$ and 1), resulting in 24 different basis inputs. Moreover, we use $\mathcal{H}(\mathcal{I}_F(\theta), \theta) = \log\det(\mathcal{I}_F(\theta))$ as the scalar cost function of the FIM. We make use of the bPF in Algorithm 2 with $N = 2.5 \cdot 10^3$, $M = 100$, $M_{\text{limit}} = 10$ and $\rho = 5$. For the robustness, we sample $N_s = 30$ parameters uniformly from the set $\Theta = \{(\beta, \gamma) : \gamma \in (1.6, 2.4), \beta \in (0.6, 1)\}$. We run the EM algorithm for 500 iterations.

References

[1] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[2] C. Brighenti, B. Wahlberg, and C.R. Rojas. Input design using Markov chains for system identification. In *Proceedings of the joint 48th Conference on Decision and Control and 28th Chinese Conference*, pages 1557–1562, Shanghai, P.R. China, December 2009.

[3] G. Calafiore and M.C. Campi. Uncertain convex programs: randomized solutions and confidence levels. *Mathematical Programming*, 102(1):25–46, 2005.

[4] M.C. Campi and S. Garatti. The exact feasibility of randomized solutions of uncertain convex programs. *SIAM Journal on Optimization*, 19(3):1211–1230, 2008.

[5] O. Cappé, S.J. Godsill, and E. Moulines. An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE*, 95(5):899–924, 2007.

[6] O. Cappé, E. Moulines, and T. Rydén. *Inference in Hidden Markov Models.* Springer, 2005.

[7] D.R. Cox. *Planning of experiments*. New York: Wiley, 1958.

[8] A.C. Davison and D.V. Hinkley. *Bootstrap methods and their application*. Cambridge University Oress, 1997.

[9] A. De Cock, M. Gevers, and J. Schoukens. A preliminary study on optimal input design for nonlinear systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 4931–4936, Florence, Italy, December 2013.

[10] P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B*, 68(3):411–436, 2006.

[11] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 39(1):1–38, 1977.

[12] J.L. Doob. *Stochastic processes*. New York Wiley, 1953.

[13] R. Douc, A. Garivier, E. Moulines, and J. Olsson. Sequential Monte Carlo smoothing for general state space hidden Markov models. *Annals of Applied Probability*, 21(6):2109–2145, 2011.

[14] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo methods in practice*. Springer, 2001.

[15] A. Doucet and A. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. In D. Crisan and B. Rozovsky, editors, *The Oxford Handbook of Nonlinear Filtering*. Oxford University Press, 2011.

[16] V.V. Fedorov. *Theory of optimal experiments*. New York: Academic Press, 1972.

[17] M. Forgione, X. Bombois, P.M.J. Van den Hof, and H. Hjalmarsson. Experiment design for parameter estimation in nonlinear systems based on multilevel excitation. In *Proceedings of the* 13*th European Control Conference*, Strasbourg, France, June 2014.

[18] L. Gerencsér, H. Hjalmarsson, and J. Mårtensson. Identification of ARX systems with non−stationary inputs−asymptotic analysis with application to adaptive input design. *Automatica*, 45(3):623–633, 2009.

[19] M. Gevers. Identification for control: from the early achievements to the revival of experiment design. *European Journal of Control*, 11:1–18, 2005.

[20] G.C. Goodwin and R.L. Payne. *Dynamic System Identification: Experiment Design and Data Analysis*. Academic Press, New York, 1977.

[21] R.B. Gopaluni, T.B. Schön, and A.G. Wills. Input design for nonlinear stochastic dynamic systems - A particle filter approach. In *Proceedings of the 18th IFAC World Congress*, Milano, Italy, August 2011.

[22] R. Hildebrand and M. Gevers. Identification for control: Optimal input design with respect to a worst-case ν-gap cost function. *SIAM Journal of Control Optimization*, 41(5):1586–1608, 2003.

[23] H. Hjalmarsson and J. Mårtensson. Optimal input design for identification of non-linear systems: Learning from the linear case. In *Proceedings of the American Control Conference*, pages 1572–1576, New York, United States, 2007.

[24] H. Jansson and H. Hjalmarsson. Input design via LMIs admitting frequency-wise model specifications in confidence regions. *IEEE Transactions on Automatic Control*, 50(10):1534–1549, 2005.

[25] D.B. Johnson. Finding all the elementary circuits of a directed graph. *SIAM Journal on Computing*, 4(1):77–84, 1975.

[26] C. Larsson, H. Hjalmarsson, and C.R. Rojas. On optimal input design for nonlinear FIR-type systems. In *Proceedings of the 49th IEEE Conference on Decision and Control*, pages 7220–7225, Atlanta, USA, December 2010.

[27] K. Lindqvist and H. Hjalmarsson. Optimal input design using linear matrix inequalities. In *Proceedings of the IFAC Symposium on System Identification*, Santa Barbara, California, USA, July 2000.

[28] F. Lindsten and T.B. Schön. Backward simulation methods for Monte Carlo statistical inference. *Foundations and Trends in Machine Learning*, 6(1):1–143, 2013.

[29] L. Ljung. *System Identification. Theory for the User, 2nd ed.* Upper Saddle River, NJ: Prentice-Hall, 1999.

[30] G. McLachlan and T. Krishnan. *The EM algorithm and extensions*. John Wiley & Sons, 2008.

[31] I. Meilijson. A fast improvement to the EM algorithm on its own terms. *Journal of the Royal Statistical Society: Series B*, pages 127–138, 1989.

[32] M.K. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599, 1999.

[33] H.E. Rauch, F. Tung, and C.T. Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8):1445–1450, August 1965.

[34] R.T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[35] C.R. Rojas, J.C. Aguero, J.S. Welsh, G.C. Goodwin, and A. Feuer. Robustness in experiment design. *IEEE Transactions on Automatic Control*, 57(4):860–874, 2012.

[36] C.R. Rojas, H. Hjalmarsson, L. Gerencsér, and J. Mårtensson. An adaptive method for consistent estimation of real-valued non-minimum phase zeros in stable LTI systems. *Automatica*, 47(7):1388–1398, 2011.

[37] C.R. Rojas, J.S. Welsh, G.C. Goodwin, and A. Feuer. Robust optimal experiment design for system identification. *Automatica*, 43(6):993–1008, June 2007.

[38] T.B. Schön, A. Wills, and B. Ninness. System identification of nonlinear state-space models. *Automatica*, 47(1):39–49, January 2011.

[39] M. Segal and E. Weinstein. A new method for evaluating the log-likelihood gradient, the Hessian, and the Fisher information matrix for linear dynamic systems. *IEEE Transactions on Information Theory*, 35(3):682–687, 1989.

[40] H. Suzuki and T. Sugie. On input design for system identification in time domain. In *Proceedings of the European Control Conference*, Kos, Greece, July 2007.

[41] E. Taghavi, F. Lindsten, L. Svensson, and T.B. Schön. Adaptive stopping for fast particle smoothing. In *Proceedings of the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013.

[42] R. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, June 1972.

[43] P.E. Valenzuela. *Optimal input design for nonlinear dynamical systems: a graph-theory approach*. Licentiate thesis, School of Electrical Engineering, KTH Royal Institute of Technology, Sweden, 2014.

[44] P.E. Valenzuela, J. Dahlin, C.R. Rojas, and T.B. Schön. A graph/particle-based method for experiment design in nonlinear systems. In *Proceedings of the 19th IFAC World Congress*, Cape Town, South Africa, August 2014.

[45] P.E. Valenzuela, C.R. Rojas, and H. Hjalmarsson. Optimal input design for dynamic systems: a graph theory approach. In *Proceedings of the IEEE Conference on Decision and Control*, pages 5740–5745, Florence, Italy, December 2013.

[46] P.E. Valenzuela, C.R. Rojas, and H. Hjalmarsson. A graph theoretical approach to input design for identification of nonlinear dynamical models. *Automatica*, 51:233–242, 2015.

[47] T.L. Vincent, C. Novara, K. Hsu, and K. Poolla. Input design for structured nonlinear system identification. In *15th IFAC Symposium on System Identification*, pages 174–179, Saint-Malo, France, July 2009.

[48] T.L. Vincent, C. Novara, K. Hsu, and K. Poolla. Input design for structured nonlinear system identification. *Automatica*, 46(6):990–998, 2010.

[49] J.S. Welsh and C.R. Rojas. A scenario based approach to robust experiment design. In *Proceedings of the 15th IFAC Symposium on System Identification*, Saint-Malo, France, July 2009.

[50] P. Whittle. Some general points in the theory of optimal experiment design. *Journal of Royal Statistical Society: Series B*, 1:123–130, 1973.

[51] A. Zaman. Stationarity on finite strings and shift register sequences. *The Annals of Probability*, 11(3):678–684, 1983.