Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

A graph/particle-based method for experiment design in nonlinear systems [⋆]

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

[*] *Department of Automatic Control, KTH Royal Institute of Technology, SE-100 44, Stockholm, Sweden (e-mail: {pva, crro}@kth.se).*
[**] *Division of Automatic Control, Linköping University, SE-581 83 Linköping, Sweden (e-mail: johan.dahlin@liu.se).*
[***] *Division of Systems and Control, Uppsala University, SE-751 05 Uppsala, Sweden (e-mail: thomas.schon@it.uu.se).*

Abstract: We propose an extended method for experiment design in nonlinear state space models. The proposed input design technique optimizes a scalar cost function of the information matrix, by computing the optimal stationary probability mass function (pmf) from which an input sequence is sampled. The feasible set of the stationary pmf is a polytope, allowing it to be expressed as a convex combination of its extreme points. The extreme points in the feasible set of pmf's can be computed using graph theory. Therefore, the final information matrix can be approximated as a convex combination of the information matrices associated with each extreme point. For nonlinear systems, the information matrices for each extreme point can be computed by using particle methods. Numerical examples show that the proposed technique can be successfully employed for experiment design in nonlinear systems.

Keywords: System identification, input design, particle filter, nonlinear systems.

1. INTRODUCTION

Experiment design deals with the generation of an input signal that maximizes the information retrieved from an experiment. Some of the initial contributions are discussed in Cox [1958] and Goodwin and Payne [1977]. Since then, many contributions to the subject have been developed; see e.g. Fedorov [1972], Whittle [1973], Hildebrand and Gevers [2003], Gevers [2005] and the references therein.

In this article, a new method for experiment design in nonlinear systems is presented, which extends the input design methods proposed in Gopaluni et al. [2011] and Valenzuela et al. [2013]. The objective is to design an experiment as a realization of a stationary process, such that the system is identified with maximum accuracy as defined by a scalar function of the Fisher information matrix, and under the assumption that the input can adopt a finite set of values. The assumption on the input class modifies the class of input sequences considered in Gopaluni et al. [2011]. The optimization of the stationary probability mass function (pmf) is done by maximizing a scalar cost function of the information matrix over the feasible set of pmf's. Using concepts from graph theory [Zaman, 1983, Johnson, 1975, Tarjan, 1972], we can express the feasible set of pmf's as a convex combination of the measures for the extreme points of the set. Therefore, the information matrix corresponding to a feasible pmf can be expressed as the convex combination of the information matrices associated with the extreme points of the feasible set. Since the exact computation of the information matrices for nonlinear systems is often intractable, we use particle methods to compute sampled information matrices for the extreme points of the feasible set. This allows us to extend the technique of Valenzuela et al. [2013] to more general nonlinear model structures. An attractive property of the method is that the optimization problem is convex even for nonlinear systems. In addition, since the input is restricted to a finite set of possible values, the method can naturally handle amplitude limitations.

Previous results on input design have mostly been concerned with linear systems. A Markov chain approach to input design is presented in Brighenti et al. [2009], where the input is modelled as the output of a Markov chain. Suzuki and Sugie [2007] presents a time domain experiment design method for system identification. Linear matrix inequalities (LMI) are used to solve the input design problem in Jansson and Hjalmarsson [2005] and Lindqvist and Hjalmarsson [2000]. A robust approach for input design is presented in Rojas et al. [2007], where the input signal is designed to optimize a cost function over a set where the true parameter is assumed to lie.

In recent years, the interest in input design for nonlinear systems has increased. The main problem here is that the frequency domain approach for experiment design used in linear systems is no longer valid. An analysis of input design for nonlinear systems using the knowledge of lin-

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

ear systems is considered in Hjalmarsson and Mårtensson [2007]. In Larsson et al. [2010] an input design method for a particular class of nonlinear systems is presented. Input design for structured nonlinear systems is discussed in Vincent et al. [2009]. Gopaluni et al. [2011] introduces a particle filter method for input design in nonlinear systems. An analysis of input design for a class of Wiener systems is considered in Cock et al. [2013]. A graph theory approach for input design for output-error like nonlinear system is presented in Valenzuela et al. [2013]. The results presented allow to design input signals when the system contains nonlinear 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 Brighenti et al. [2009], Larsson et al. [2010] and Valenzuela et al. [2013], the proposed methods cannot handle amplitude limitations on the input signal, which could arise due to physical and/or safety reasons.

2. PROBLEM FORMULATION

In this article, the objective is to design an input signal $u_{1:n_{\text{seq}}} := \{u_t\}_{t=1}^{n_{\text{seq}}}$, as a realization of a stationary process. This is done such that a state space model (SSM) can be identified with maximum accuracy as defined by a scalar function of the Fisher information matrix $\mathcal{I}_F$ [Ljung, 1999]. An SSM with states $x_{1:T} := \{x_t\}_{t=1}^{T}$, inputs $u_{1:T}$ and measurements $y_{1:T}$ is given by

$$x_t|x_{t-1} \sim f_\theta(x_t|x_{t-1}, u_{t-1}), \tag{1a}$$

$$y_t|x_t \sim g_\theta(y_t|x_t, u_t). \tag{1b}$$

Here, $f_\theta(\cdot)$ and $g_\theta(\cdot)$ denote known probability distributions parametrised by $\theta \in \Theta \subset \mathbb{R}^d$. For the remainder of this article, we make the rather restrictive albeit standard assumption that we know the initial state x_0 and the true model structure (1) with true parameters θ_0. Hence, we can write the joint distribution of states and measurements for (1) as

$$p_\theta(x_{1:T}, y_{1:T}|u_{1:T}) = \prod_{t=1}^{T} f_\theta(x_t|x_{t-1}, u_{t-1}) g_\theta(y_t|x_t, u_t). \tag{2}$$

This quantity is used in the sequel for estimating $\mathcal{I}_F$ by

$$\mathcal{I}_F := \mathbf{E}_\theta \left\{ \mathcal{S}(\theta_0)\mathcal{S}^\top(\theta_0) \right\} , \tag{3a}$$

$$\mathcal{S}(\theta_0) := \nabla \log l_\theta(y_{1:n_{\text{seq}}})\big|_{\theta=\theta_0} , \tag{3b}$$

where $l_\theta(y_{1:n_{\text{seq}}})$ and $\mathcal{S}(\theta)$ denote the likelihood function and the score function, respectively. Note, that the expected value in (3a) is with respect to the stochastic processes in (1) and the realizations of $u_{1:n_{\text{seq}}}$.

We note that (3a) depends on the cumulative density function (cdf) of $u_{1:n_{\text{seq}}}$, say $P_u(u_{1:n_{\text{seq}}})$. Therefore, the input design problem is to find a cdf $P_u^{\text{opt}}(u_{1:n_{\text{seq}}})$ which optimizes a scalar function of (3a). We define this scalar function as $h_m : \mathbb{R}^{d\times d} \to \mathbb{R}$. To obtain the desired results, h_m must be a nondecreasing matrix function [Boyd and Vandenberghe, 2004, pp. 108]. Different choices of h_m have been proposed in the literature, see e.g. Rojas et al. [2007]; some examples are $h_m = \det$, and $h_m = -\operatorname{tr}\{(\cdot)^{-1}\}$. In this work, we leave the selection of h_m to the user.

Since $P_u^{\text{opt}}(u_{1:n_{\text{seq}}})$ has to be a stationary cdf, the optimization must be constrained to the set

$$\left.\begin{aligned}\mathcal{P} := \{P_u : \mathbb{R}^{n_{\text{seq}}} \to \mathbb{R}|\, &P_u(\mathbf{x}) \geq 0,\, \forall \mathbf{x} \in \mathbb{R}^{n_{\text{seq}}};\\ &P_u \text{ is monotone non-decreasing };\\ &\lim_{\substack{x_i \to \infty\\ i=\{1,\,\ldots,\,n_{\text{seq}}\}\\ \mathbf{x}=(x_1,\,\ldots,\,x_{n_{\text{seq}}})}} P_u(\mathbf{x}) = 1;\\ \int_{v\in\mathbb{R}} dP_u(v,\, \mathbf{z}) = \int_{v\in\mathbb{R}} &dP_u(\mathbf{z},\, v)\,, \forall \mathbf{z} \in \mathbb{R}^{n_{\text{seq}}-1}\end{aligned}\right\} . \tag{4}$$

The last condition in (4) (with slight abuse of notation) guarantees that $P_u \in \mathcal{P}$ is the cdf of a stationary sequence [Zaman, 1983].

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

$$\left.\begin{aligned}\mathcal{P}_{\mathcal{C}} := \{p_u : \mathcal{C}^{n_{\text{seq}}} \to \mathbb{R}|\, &p_u(\mathbf{x}) \geq 0,\, \forall \mathbf{x} \in \mathcal{C}^{n_{\text{seq}}};\\ &\sum_{\mathbf{x}\in\mathcal{C}^{n_{\text{seq}}}} p_u(\mathbf{x}) = 1;\\ \sum_{v\in\mathcal{C}} p_u(v,\, \mathbf{z}) = \sum_{v\in\mathcal{C}} &p_u(\mathbf{z},\, v)\,, \forall \mathbf{z} \in \mathcal{C}^{(n_{\text{seq}}-1)}\end{aligned}\right\} . \tag{5}$$

The set introduced in (5) will constrain the pmf $p_u(u_{1:n_{\text{seq}}})$.

The problem described can be summarized 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

$$p_u^{\text{opt}} := \arg\max_{p_u\in\mathcal{P}_{\mathcal{C}}} h_m(\mathcal{I}_F(p_u)) , \tag{6}$$

with $h_m : \mathbb{R}^{d\times d} \to \mathbb{R}$ a matrix nondecreasing function, and $\mathcal{I}_F \in \mathbb{R}^{d\times d}$ defined as in (3). ■

3. NEW INPUT DESIGN METHOD

In this section, we discuss the proposed input design method, which is based on three steps. In the first step, we calculate basis input signals, which are used to excite the system. In the second step, we iteratively calculate the information matrix estimate and the optimal weighting of the basis inputs in a Monte Carlo setting. In the third step, we generate an optimal input sequence using the estimated optimal weighting of the basis inputs.

3.1 Graph theoretical input design

Problem 1 is often hard to solve explicitly since

(i) we need to represent the elements in $\mathcal{P}_{\mathcal{C}}$ as a linear combination of its basis functions, and
(ii) the stationary pmf p_u is of dimension n_{seq}, where n_{seq} could potentially be very large.

These issues make Problem 1 computationally intractable.

To solve issue (ii), we assume that p_u is an extension from the subspace of stationary pmf's of memory length n_m, where $n_m << n_{\text{seq}}$.

To address issue (i), notice that all the elements in $\mathcal{P}_{\mathcal{C}}$ can be represented as a convex combination of its extreme points [Valenzuela et al., 2013]. 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 $\mathcal{P}_{\mathcal{C}}$.

To find all the elements in $\mathcal{V}_{\mathcal{P}_{\mathcal{C}}}$, we will make use of graph theory as follows. $\mathcal{C}^{n_m}$ is composed of $(c_{\text{seq}})^{n_m}$ elements.

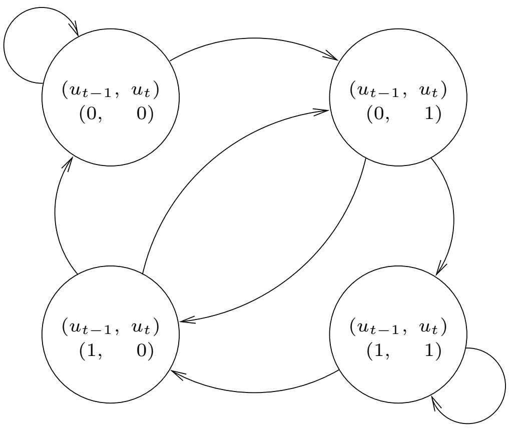
$(u_{t-1},\ u_t)$ $(0,\ 0)$
$(u_{t-1},\ u_t)$ $(0,\ 1)$
$(u_{t-1},\ u_t)$ $(1,\ 0)$
$(u_{t-1},\ u_t)$ $(1,\ 1)$

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

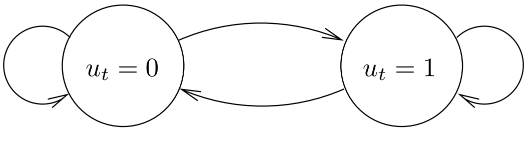

Fig. 2. Example of graph derived from $\mathcal{C}^{n_m}$, with $c_{\text{seq}} = 2$, $n_m = 1$, and $\mathcal{C} := \{0, 1\}$.

Each element in $\mathcal{C}^{n_m}$ can be viewed as one node in a graph. In addition, the transitions (edges) between the elements in $\mathcal{C}^{n_m}$ are given by the feasible 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 $k \geq 0$. Figure 1 illustrates this idea, when $c_{\text{seq}} = 2$, $n_m = 2$, and $\mathcal{C} = \{0, 1\}$. From this figure we can 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 $t + 1$.

To find all the elements in $\mathcal{V}_{\mathcal{P}_\mathcal{C}}$ we rely on the concept of prime cycles. A *prime cycle* is an elementary cycle whose set of nodes do not have a proper subset which is an elementary cycle [Zaman, 1983, pp. 678]. It has been proved that the prime cycles of a graph describe all the elements in the set $\mathcal{V}_{\mathcal{P}_\mathcal{C}}$ [Zaman, 1983, Theorem 6]. In other words, each prime cycle defines one element $v_j \in \mathcal{V}_{\mathcal{P}_\mathcal{C}}$. Furthermore, each v_j corresponds to a uniform distribution whose support is the set of elements of its prime cycle, for all $j \in \{1, \ldots, n_\mathcal{V}\}$ [Zaman, 1983, pp. 681]. Therefore, the elements in $\mathcal{V}_{\mathcal{P}_\mathcal{C}}$ can be described by finding all the prime cycles associated with the stationary graph $\mathcal{G}_{\mathcal{C}^{n_m}}$ drawn from $\mathcal{C}^{n_m}$.

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 $\mathcal{G}_{\mathcal{C}^{(n_m-1)}}$ [Zaman, 1983, Lemma 4], which can be found by using existing algorithms[1]. To illustrate this, we consider the graph depicted in Figure 2. One elementary cycle for this graph is given by $(0, 1, 0)$. Using [Zaman, 1983, Lemma 4], the elements of one prime cycle for the graph $\mathcal{G}_{\mathcal{C}^2}$ are obtained as a concatenation of the elements in the elementary cycle $(0, 1, 0)$. Hence, the prime cycle in $\mathcal{G}_{\mathcal{C}^2}$ associated with this elementary cycle is given by $((0, 1), (1, 0), (0, 1))$.

Since we know the prime cycles, it is possible to generate an input sequence $\{u_t^j\}_{t=0}^T$ from v_j, which will be referred to as the *basis inputs*. As an example, we use the graph depicted in Figure 1. One prime cycle for this graph is given by $((0, 1), (1, 0), (0, 1))$. Therefore, the sequence $\{u_t^j\}_{t=0}^T$ is given by taking the last element of each node, i.e., $\{u_t^j\}_{t=0}^T = \{1, 0, 1, 0, \ldots, ((-1)^T + 1)/2\}$.

Given $\{u_t^j\}_{t=0}^T$, we can use them to obtain the corresponding information matrix for $v_j \in \mathcal{V}_{\mathcal{P}_\mathcal{C}}$, say $\mathcal{I}_F^{(j)}$. However, in general the matrix $\mathcal{I}_F^{(j)}$ cannot be computed explicitly. To overcome this problem, we use Sequential Monte Carlo (SMC) methods to approximate $\mathcal{I}_F^{(j)}$, as discussed in the next subsection.

3.2 Estimation of the score function

SMC methods are a family of methods that can be used e.g. to estimate the filtering and smoothing distributions in SSMs. General introductions to SMC samplers are given in e.g. Doucet and Johansen [2011] and Del Moral et al. [2006]. Here, we introduce the *auxiliary particle filter* (APF) [Pitt and Shephard, 1999] and the fixed-lag (FL) particle smoother [Kitagawa and Sato, 2001] to estimate the score function for (1). In the next subsection, the score function estimates are used to estimate the information matrix using (3a).

The APF estimates the smoothing distribution by

$$\widehat{p}_\theta(\mathrm{d}x_{1:t}|y_{1:t}) := \sum_{i=1}^{N} \frac{w_t^{(i)}}{\sum_{k=1}^{N} w_t^{(k)}} \delta_{x_{1:t}^{(i)}}(\mathrm{d}x_{1:t}), \tag{7}$$

where the particle system is denoted by $\{x_{1:t}^{(i)}, w_t^{(i)}\}_{i=1}^N$. Here, $w_t^{(i)}$ and $x_{1:t}^{(i)}$ denote the weights and the particle trajectories computed by the APF. Here, $\delta_z(\mathrm{d}x_{1:t})$ denotes the Dirac measure at z.

The particle system is sequentially computed using two steps: (i) sampling/propagation and (ii) weighting. The first step can be seen as sampling from a proposal kernel,

$$\{a_t^{(i)}, x_t^{(i)}\} \sim \frac{w_{t-1}^{a_t}}{\sum_{k=1}^{N} w_{t-1}^{(k)}} R_{\theta,t}(x_t|x_{t-1}^{a_t}, u_{t-1}), \tag{8}$$

where we append the sampled particle to the trajectory by $x_{1:t}^{(i)} = \{x_{1:t-1}^{(i)}, x_t^{(i)}\}$. Here, $R_{\theta,t}(\cdot)$ denotes the propagation kernel and the *ancestor index* $a_t^{(i)}$ denotes the index of the *ancestor* at time $t-1$ of particle $x_t^{(i)}$. In the second step, we calculate the (unnormalised) importance weights,

$$w_t^{(i)} \triangleq \frac{g_\theta(y_t|x_t^{(i)}, u_t) f_\theta(x_t|x_{t-1}^{(i)}, u_{t-1})}{R_{\theta,t}(x_t|x_{t-1}^{(i)}, u_{t-1})}. \tag{9}$$

We make use of *Fisher's identity* [Fisher, 1925, Cappé et al., 2005] to rewrite the score function into a form that can be used in combination with SMC methods. This results in that we can write

$$\nabla \log\, l_\theta(y_{1:T}) = \mathbf{E}_\theta\Big[\nabla \log\, p_\theta(x_{1:T}, y_{1:T}|u_{1:T})\Big|y_{1:T}, u_{1:T}\Big],$$

where we insert (2) to obtain the form

$$\nabla \log\, l_\theta(y_{1:T}) = \sum_{t=1}^{T} \int \xi_\theta(x_{t-1:t}) p_\theta(x_{t-1:t}|y_{1:T})\, \mathrm{d}x_{t-1:t},$$

[1] For the examples in Section 4, we have used the algorithm presented in [Johnson, 1975, pp. 79–80] complemented with the one proposed by [Tarjan, 1972, pp. 157].

Algorithm 1 PF for score function estimation

INPUTS: An SSM (1), $y_{1:T}$ (obs.), $u_{1:T}$ (inp.) and N (no. particles).
OUTPUT: $\widehat{\mathcal{S}}(\theta)$ (est. of the score).

```
Initialise particles $x_0^{(i)}$ for $i = 1$ to $N$.
for $t = 1$ to $T$ do
    Resample the particles with weights $\{w_{t-1}^{(i)}\}_{i=1}^N$.
    Propagate the particles using $R_{\theta,t}(\cdot)$.
    Compute (9) to obtain $\{w_t^{(i)}\}_{i=1}^N$.
end for
Compute (10) to obtain $\widehat{\mathcal{S}}(\theta)$.
```

with

$$\xi_\theta(x_{t-1:t}) = \nabla\Big[\log f_\theta(x_t|x_{t-1}, u_{t-1}) + \log g_\theta(y_t|x_t, u_t)\Big],$$

which depends on the two-step marginal smoothing densities. The APF can be used the estimate these quantities but this leads to poor estimates with high variance, due to problems with *particle degeneracy.*

Instead, we use an FL-smoother to estimate the smoothing densities, which reduces the variance of the score estimates [Olsson et al., 2008]. The fixed-lag smoother assumes that

$$p_\theta(x_t|y_{1:T}, u_{1:T}) \approx p_\theta(x_t|y_{1:\kappa_t}, u_{1:\kappa_t}),$$

for $\kappa_t = \min(t + \Delta, T)$ with some fixed-lag Δ. This means that measurements after some time has a negligible effect on the state, see Dahlin et al. [2014] for more details about the FL-smoother and its use for score estimation. The resulting expression is obtained as

$$\widehat{\mathcal{S}}(\theta) := \sum_{t=1}^{T}\sum_{i=1}^{N} w_{\kappa_t}^{(i)} \xi_\theta\Big(x_t^{a^{(i)}_{\kappa_t,t}}, x_{t-1}^{a^{(i)}_{\kappa_t,t-1}}, u_t\Big), \qquad (10)$$

where $a^{(i)}_{\kappa_t,t}$ denotes the particle at time t which is the ancestor of particle i at time κ_t. The complete procedure for estimating the score function using the FL smoother is outlined in Algorithm 1.

3.3 Monte Carlo-based optimisation

Given $\{\mathcal{I}_F^{(j)}\}_{j=1}^{n_\mathcal{V}}$ associated with the elements in $\mathcal{V}_{\mathcal{P}_\mathcal{C}}$, we can find the corresponding information matrix associated with any element in $\mathcal{P}_\mathcal{C}$ as a convex combination of the $\mathcal{I}_F^{(j)}$'s. By defining $\gamma := \{\alpha_1, \ldots, \alpha_{n_\mathcal{V}}\} \in \mathbb{R}^{n_\mathcal{V}}$, we introduce $\mathcal{I}_F^{\text{app}}(\gamma)$ as the information matrix associated with one element in $\mathcal{P}_\mathcal{C}$ for a given γ such that $\alpha_j \geq 0$, $j \in \{1, \ldots, n_\mathcal{V}\}$, $\sum_{j=1}^{n_\mathcal{V}} \alpha_j = 1$. Therefore, finding the optimal $\mathcal{I}_F^{\text{app}}(\gamma)$ is equivalent to determining the optimal weighting vector γ.

Hence, we can rewrite Problem 1 as

$$\gamma^{\text{opt}} = \arg\max_{\gamma\in\mathbb{R}^{n_\mathcal{V}}} h_m(\mathcal{I}_F^{\text{app}}(\gamma)), \qquad (11a)$$

$$\text{st.} \quad \mathcal{I}_F^{\text{app}}(\gamma) := \sum_{j=1}^{n_\mathcal{V}} \alpha_j \, \mathcal{I}_F^{(j)}, \qquad (11b)$$

$$\sum_{j=1}^{n_\mathcal{V}} \alpha_j = 1, \qquad (11c)$$

$$\alpha_j \geq 0, \text{ for all } j \in \{1, \ldots, n_\mathcal{V}\}, \qquad (11d)$$

To solve the optimisation problem (11), we need to estimate the information matrix for each basis input.

Algorithm 2 Optimal input estimating using Monte Carlo

INPUTS: Algorithm 1, K (no. MC runs) and M (size of each batch).
OUTPUT: $\gamma^\star$ (est. of the optimal weighting of the basis inputs).

```
for $k = 1$ to $K$ do
    Generate $M$ samples using Algorithm 1 for each basis input.
    Estimate the information matrix by (12) for each basis input.
    Solve the problem in (11).
    Set $\gamma_k$ as the weighting factors obtained from the solver.
end for
Compute the sample mean of $\gamma = \{\gamma_1, \ldots, \gamma_K\}$, denote it as $\gamma^\star$.
```

In the SMC literature, the observed information matrix is often estimated by the use of *Louis' identity* [Louis, 1982, Cappé et al., 2005]. However, this approach does not guarantee that the information matrix estimate is positive semi-definite. In the authors' experience, this standard approach also leads to poor accuracy in the estimates.

Instead, we make use of the fact that the information matrix can be expressed as (3), i.e. the variance of the score function. Hence, a straight-forward method for estimating the information matrix is to use the Monte Carlo covariance estimator over some realisations of the system. If we denote each Monte Carlo estimate of the score function by $\widehat{\mathcal{S}}_m(\theta)$, the information matrix can be estimated using

$$\widehat{\mathcal{I}}_F = \frac{1}{M-1}\sum_{m=1}^{M} \widehat{\mathcal{S}}_m(\theta)\widehat{\mathcal{S}}_m^\top(\theta), \qquad (12)$$

where M denotes the number of score estimates. Note, that this is an estimate of Fisher information matrix as the Monte Carlo estimator averages over the system realisations. The estimate is positive semi-definite by construction but inherits some bias from the FL-smoother, see Olsson et al. [2008] for more information. This problem can be handled by using more computationally costly particle smoother. Later, we present results indicating that this bias does not effect the resulting input signal to any large extent.

The information matrix estimate in (12) can be used to estimate $\mathcal{I}_F^{(j)}$ for each basis input. A simple solution is therefore to plug-in the estimates and solve the convex optimisation problem (11) using some standard solver. However, by doing this we neglect the stochastic nature of the estimates and disregard the uncertainty. In practice, this leads to bad estimates of γ.

Instead, we propose the use of a Monte Carlo method which iterates two different steps over K iterations. In step (a), we compute the information matrix estimates $\mathcal{I}_F^{(j)}$ for each input using (12). In step (b), we solve the optimisation problem in (11) using the estimates to obtain γ_k at iteration k. The estimate of the optimal weighting vector $\gamma^\star$ is found using the sample mean of $\gamma = \{\gamma_1, \ldots, \gamma_K\}$, which can be complemented with an confidence interval (CI). Such CI could be useful in determining which of the basis inputs that are significant and should be included in the optimal input sequence. The outline of the complete procedure is presented in Algorithm 2.

Algorithm 3 New input design method

```
INPUTS: Algorithm 2, $\mathcal{C}$ (input values), $n_m$ (memory) and $T$ (no. input samples).
OUTPUT: $\gamma^\star$ (est. of the optimal weighting of the basis inputs).

1: Compute all the elementary cycles of $\mathcal{G}_{\mathcal{C}^{(n_m-1)}}$ by using, e.g., [Johnson, 1975, pp. 79–80], [Tarjan, 1972, pp. 157].
2: 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 above (c.f. [Zaman, 1983, Lemma 4]).
3: Generate the input signals $\{u_t^j\}_{t=0}^T$ from the prime cycles of $\mathcal{G}_{\mathcal{C}^{n_m}}$, for each $j \in \{1, \ldots, n_\mathcal{V}\}$.
4: Execute Algorithm 2.
```

Table 1. $h_m(\widehat{\mathcal{I}}_F)$, Example 1.

Input / $h_m(\widehat{\mathcal{I}}_F)$	$\log\det(\widehat{\mathcal{I}}_F)$	$\text{tr}\left\{(\widehat{\mathcal{I}}_F)^{-1}\right\}$
Optimal (det)	20.67(0.01)	$1.51 \cdot 10^{-4}(5.18 \cdot 10^{-7})$
Optimal (tr)	20.82(0.01)	$1.32 \cdot 10^{-4}(4.45 \cdot 10^{-7})$
Binary	**20.91**(0.01)	$\mathbf{1.21 \cdot 10^{-4}}(4.51 \cdot 10^{-7})$
Uniform	19.38(0.01)	$5.32 \cdot 10^{-4}(2.12 \cdot 10^{-6})$

3.4 Summary of the method

The proposed method for designing of input signals in $\mathcal{C}^{n_m}$ is summarized in Algorithm 3.

The algorithm computes $\gamma^\star$ which defines the optimal pmf $p_u^{\text{opt}}(u_{1:n_m})$ as a convex combination of the measures associated with the elements in $\mathcal{V}_{\mathcal{P}_\mathcal{C}}$, with $\gamma^\star$ as the weighting vector. Notice that $\mathcal{I}_F^{\text{app}}(\gamma)$ in (11b) is linear in the decision variables. Therefore, the optimization (11) is convex.

4. NUMERICAL EXAMPLES

The following examples present some applications of the proposed input design method.

Example 1. Consider the linear Gaussian state space (LGSS) system with parameters $\theta = \{\theta_1, \theta_2\}$,

$$\begin{aligned} x_{t+1} &= \theta_1 x_t + u_t + v_t, & v_t &\sim \mathcal{N}(0, \theta_2^2), \\ y_t &= x_t + e_t, & e_t &\sim \mathcal{N}(0, 0.1^2), \end{aligned}$$

where the true parameters are $\theta_0 = \{0.5, 0.1\}$. We design experiments to identify θ with $n_{\text{seq}} = 5 \cdot 10^3$ time steps, memory length $n_m = 2$, and an input assuming values $\mathcal{C} = \{-1, 0, 1\}$. The optimal experiments maximize $h_m(\mathcal{I}_F^{\text{app}}(\gamma)) = \det(\mathcal{I}_F^{\text{app}}(\gamma))$, and $h_m(\mathcal{I}_F^{\text{app}}(\gamma)) = -\text{tr}\left\{(\mathcal{I}_F^{\text{app}}(\gamma))^{-1}\right\}$.

We generate $\{u_t^j\}_{t=0}^T$ for each $v_j \in \mathcal{V}_{\mathcal{P}_\mathcal{C}}$ ($T = 10^2$) to compute the approximation (12) for each $\mathcal{I}_F^{(j)}$. Finally, the optimal input $u_{1:n_{\text{seq}}}$ is computed by running a Markov chain with $p_u^{\text{opt}}(u_{1:n_m})$ as stationary pmf, where we discard the first $2 \cdot 10^6$ samples and keep the last $n_{\text{seq}} = 5 \cdot 10^3$ ones. In addition, we consider $K = 100$, $M = 5 \cdot 10^3$ and $N = 10^3$. As a benchmark, we generate n_{seq} input samples from uniformly distributed white noise with support $[-1, 1]$, and the same amount of samples from binary white noise with values $\{-1, 1\}$. These input samples are employed to compute an approximation of $\mathcal{I}_F$ via (12).

Table 1 condenses the results obtained for each input sequence, where *Optimal (det)* and *Optimal (tr)* represent the results for the input sequences obtained from optimizing $\det(\mathcal{I}_F^{\text{app}}(\gamma))$, and $-\text{tr}\left\{(\mathcal{I}_F^{\text{app}}(\gamma))^{-1}\right\}$, respectively.

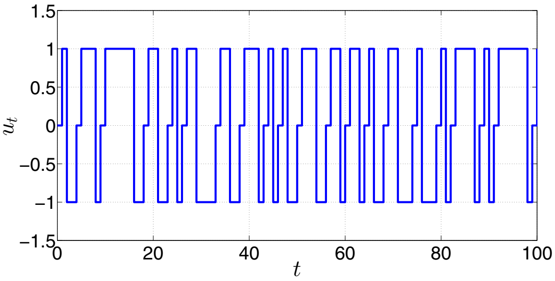

Fig. 3. Input realization, Example 2.

Table 2. $h_m(\widehat{\mathcal{I}}_F)$, Example 2.

Input / $h_m(\widehat{\mathcal{I}}_F)$	$\log\det(\widehat{\mathcal{I}}_F)$
Optimal	**25.34**(0.01)
Binary	24.75(0.01)
Uniform	24.38(0.01)

The 95% confidence intervals are given as ± the value in the parentheses. From the data we conclude that, for this particular example, the binary white noise seems to be the best input sequence. Indeed, the proposed input design method tries to mimic the binary white noise excitation, which is clear from the numbers in Table 1. ■

Example 2. In this example we consider the system in [Gopaluni et al., 2011, Section 6], given by

$$\begin{aligned} x_{t+1} &= \theta_1 x_t + \frac{x_t}{\theta_2 + x_t^2} + u_t + v_t, & v_t &\sim \mathcal{N}(0, 0.1^2), \\ y_t &= \frac{1}{2} x_t + \frac{2}{5} x_t^2 + e_t, & e_t &\sim \mathcal{N}(0, 0.1^2), \end{aligned}$$

where $\theta = \{\theta_1, \theta_2\}$ denotes the parameters with true values $\theta_0 = \{0.7, 0.6\}$. We design an experiment with the same settings as in Example 1 maximizing $h_m(\mathcal{I}_F^{\text{app}}(\gamma)) = \det(\mathcal{I}_F^{\text{app}}(\gamma))$. A typical input realization obtained from the proposed input design method is presented in Figure 2.

Table 2 presents the results obtained for each input sequence, where *Optimal* represents the result for the input sequence obtained from optimizing $\det(\mathcal{I}_F^{\text{app}}(\gamma))$. The 95% confidence intervals are given as ± the value in the parentheses. From these data we conclude that the extended input design method outperforms the experiment results obtained for binary and uniformly distributed samples. Therefore, our new input design method can be successfully employed to design experiments for this nonlinear system. ■

5. CONCLUSION

We have presented a new input design method for state space models, which extends existing input design approaches for nonlinear systems. The extension considers a more general model structure, and a new class for the input sequences. The method maximizes a scalar cost function of the information matrix, by optimizing the stationary pmf from which the input sequence is sampled. The elements in the feasible set of the stationary pmf are computed as a convex combination of its extreme points.

Under the assumption of a finite set of possible values for the input, we use graph theoretical tools to compute the information matrix as a convex combination of the information matrices associated with each extreme point. The information matrix for each extreme point is approximated using particle methods, where the information matrix is computed as the covariance of the score function. The numerical examples show that the extended input design method can be successfully used to design experiments for general nonlinear systems.

In a future work we will combine the proposed input design technique with parameter estimation methods, which will allow to simultaneously estimate the parameters and the optimal input for a nonlinear SSM. We will also consider alternative methods based on Gaussian process models for information matrix estimation. This could improve the accuracy and the efficiency in the information matrix estimation method outlined in this paper.

Finally, as with most optimal input design methods, the one proposed in this contribution relies on knowledge of the true system. This difficulty can be overcome by implementing a robust experiment design scheme on top of it [Rojas et al., 2007] or via an adaptive procedure, where the input signal is re-designed as more information is being collected from the system [Rojas et al., 2011]. This approach will be also addressed in a future work.

ACKNOWLEDGEMENTS

The authors thank to Dr. Fredrik Lindsten for his comments to improve this article.

REFERENCES

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

C. Brighenti, B. Wahlberg, and C.R. Rojas. Input design using Markov chains for system identification. In *Joint 48th Conference on Decision and Control and 28th Chinese Conference*, pages 1557–1562, Shangai, P.R. China, 2009.

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

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 (CDC'13) (accepted for publication)*, Florence, Italy, 2013.

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

J. Dahlin, F. Lindsten, and T.B. Schön. Second-order Particle MCMC for Bayesian Parameter Inference. In *Proceedings of the 19th IFAC World Congress*, Cape Town, South Africa, August 2014.

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.

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.

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

R.A. Fisher. Theory of statistical estimation. *Mathematical Proceedings of the Cambridge Philosophical Society*, 22(05):700–725, 1925.

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

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

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, 2011.

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.

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.

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, October 2005.

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

G. Kitagawa and S. Sato. Monte carlo smoothing and self-organising state-space model. In A. Doucet, N. de Fretias, and N. Gordon, editors, *Sequential Monte Carlo methods in practice*, pages 177–195. Springer, 2001.

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, 2010.

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.

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

T.A. Louis. Finding the observed information matrix when using the EM algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 44(02):226–233, 1982.

J. Olsson, O. Cappé, R. Douc, and E. Moulines. Sequential Monte Carlo smoothing with application to parameter estimation in nonlinear state space models. *Bernoulli*, 14(1):155–179, 2008.

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

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.

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.

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.

R. Tarjan. Depth-First Search and Linear Graph Algorithms. *SIAM Journal on Computing*, 1(2):146–160, June 1972.

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 (CDC'13) (accepted for publication)*, Florence, Italy, 2013. Available in http://arxiv.org/abs/1310.4706.

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

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

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