

# Optimal input design for non-linear dynamic systems: a graph theory approach

Patricio E. Valenzuela, Cristian R. Rojas and Håkan Hjalmarsson

**Abstract**—In this article a new algorithm for the design of stationary input sequences for system identification is presented. The stationary input signal is generated by optimizing an approximation of a scalar function of the information matrix, based on stationary input sequences generated from prime cycles, which describe the set of finite Markov chains of a given order. This method can be used for solving input design problems for nonlinear systems. In particular it can handle amplitude constraints on the input. Numerical examples show that the new algorithm is computationally attractive and that is consistent with previously reported results.

**Index Terms**—System identification, input design, Markov chains.

## I. INTRODUCTION

Input design concerns the generation of an input signal to maximize the information obtained from an experiment. Some of the first contributions in this area have been introduced in [1], [2]. From the roots of [1], [2], many contributions on the subject have been developed (see [3], [4], [5], [6] and the references therein).

In the case of dynamic systems, the input design problem can be formulated as the optimization of a cost function related to the model to be identified. The results in this area are mainly focused on linear systems. In [7], [8], linear matrix inequalities (LMI) are employed to solve the input design problem. In [9], the input signal is modelled as the output of a Markov chain. Robust input design is covered in [10], where the input signal is designed to optimize a cost function over the feasible set of the true parameters. A time domain approach for input design for system identification is developed in [11]. The results presented above (except for [9]) design an input signal without amplitude constraints. However, in practical applications, amplitude constraints on the input signal are required due to physical and/or performance limitations. Therefore, input design with amplitude constraints needs more analysis.

In recent years, we see growing interest to extend the results of input design to nonlinear systems. An approach to input design for nonlinear systems by using the knowledge of linear systems is presented in [12]. Input design for structured nonlinear systems has been introduced in [13]. An approach of input design for a particular class of nonlinear systems is presented in [14]. A particle filter method for

input design for nonlinear systems is presented in [15]. The results presented allow to design input signals when the system contains nonlinear functions, but the constraints on the system dynamics and the computational cost required to solve the problem are the main limitations of these results. Therefore, it is necessary to develop a method for input design suited for a wide class of nonlinear models and requiring low computational effort.

In this article we present a method to solve input design problems with amplitude limitations. The proposed technique also includes nonlinear systems with more general structures than those presented in [14]. The method designs an input signal which is restricted to a finite set of values, and it is a realization of the optimal stationary process. Since the problem is solved over the set of stationary processes, the feasible set needs to be described by basis functions. However, finding the basis functions for this set is a hard task. This drawback is solved by using ideas from graph theory [16], [17], [18]. By deriving the prime cycles of the de Bruijn's graph associated to the feasible set, we can express any element in the set as a convex sum of a finite number of elements. The information matrices associated to these elements can be approximated by a simple average, which reduces the computational costs compared to the method in [15]. A nice feature of this approach is that, if the cost function is convex, the optimization problem can be solved by using convex tools, even if the system is nonlinear. The numerical examples show that this method is consistent with the results presented in [14], and that it can be successfully applied to solve input design problems with amplitude limitations.

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 [10] or via an adaptive procedure, where the input signal is re-designed as more information is being collected from the system [19]. Due to space limitations, however, we will not address these issues in the present paper.

The rest of this paper is organized as follows. Section II presents basic concepts in graph theory. Section III introduces the input design problem. Section IV presents the newly proposed method to compute an optimal input signal. In Section V some numerical examples are presented. Finally, Section VI concludes the paper.

**Remark:** In the sequel, we denote by  $\mathbb{R}$  the real set, by  $\mathbb{R}^p$  the set of real  $p$ -dimensional vectors, and by  $\mathbb{R}^{r \times s}$  the set of real  $r \times s$  matrices. The expected value and

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

Automatic Control Lab and ACCESS Linnaeus Center, School of Electrical Engineering, KTH–Royal Institute of Technology, SE-100 44 Stockholm, Sweden. (e-mail: {pva, crro, hjalmars}@kth.se.)

the probability measure are denoted by  $\mathbf{E}\{\cdot\}$ , and  $\mathbf{P}\{\cdot\}$ , respectively. Finally,  $\det$  and  $\text{tr}$  stand for the determinant and the trace functions, respectively.

## II. PRELIMINARIES ON GRAPH THEORY

The purpose in this section is to provide a brief background on graph theory to understand the discussion in the next sections. The definitions presented here come from [17, pp. 77].

A *directed graph*  $\mathcal{G}_{\mathcal{V}} := (\mathcal{V}, \mathcal{X})$  consists of a nonempty and finite set of vertices (or nodes)  $\mathcal{V}$  and a set  $\mathcal{X}$  of ordered pairs of distinct vertices called *edges*. A *path* in  $\mathcal{G}_{\mathcal{V}}$  is a sequence of vertices  $p_{vu} := (v = v_1, v_2, \dots, v_k = u)$  such that  $(v_i, v_{i+1}) \in \mathcal{X}$  for  $i \in \{1, \dots, k-1\}$ . A *cycle* is a path in which the first and last vertices are identical. A cycle is elementary if no vertex but the first and last appears twice. Two elementary cycles are distinct if one is not a cyclic permutation of the other.

## III. PROBLEM FORMULATION

Consider the single-input, single-output time invariant system depicted in Figure 1. Here,  $G_0$  is a dynamic system (possibly nonlinear),  $\{e_t\}$  is a white noise sequence with zero mean and variance  $\lambda_e$ ,  $u_t \in \mathbb{R}$  is the input and  $y_t \in \mathbb{R}$  is the measured output. We will assume that we have a model structure for  $G_0$ . Notice that we assume that the noise  $e_t$  enters only at the output.

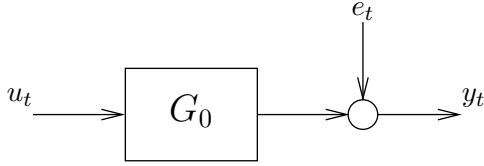


Fig. 1. Block diagram of a dynamic system (possibly nonlinear).

The objective in this article is to design an input signal  $\mathcal{U}_{n_{\text{seq}}} := (u_{n_{\text{seq}}}, \dots, u_1)$  as a realization of a stationary process, such that the system

$$y_t = G_0(\mathcal{U}_t^\infty) + e_t, \quad (1)$$

$$\mathcal{U}_t^\infty := (u_t, \dots, u_{-\infty}), \quad (2)$$

can be identified with maximum accuracy as defined by a scalar function of the Fisher information matrix  $\mathcal{I}_F$  [20].  $\mathcal{I}_F$  can be computed as

$$\mathcal{I}_F := \frac{1}{\lambda_e} \mathbf{E} \left\{ \sum_{t=1}^{n_{\text{seq}}} \psi(\theta_0) \psi(\theta_0)^T \right\}, \quad (3)$$

where

$$\psi(\theta_0) := \left. \frac{d\hat{y}(t)}{d\theta} \right|_{\theta=\theta_0}, \quad (4)$$

$$\hat{y}(t) := G(\mathcal{U}_t; \theta), \quad (5)$$

and  $\theta, \theta_0 \in \Theta \subset \mathbb{R}^m$ . The expected value in (3) is with respect to the realizations of  $\mathcal{U}_{n_{\text{seq}}}$ . In addition, the result introduced in (3)-(5) assumes that there exists a  $\theta_0 \in \Theta$

such that  $G(\mathcal{U}_t; \theta_0) = G_0(\mathcal{U}_t)$  [20], i.e., that there is no undermodelling; we will make this assumption in the sequel.

Equation (5) does not depend on the noise realization. Therefore, we can rewrite (3) as

$$\mathcal{I}_F = \frac{1}{\lambda_e} \int_{\mathcal{U}_{n_{\text{seq}}} \in \mathbb{R}^{n_{\text{seq}}}} \sum_{t=1}^{n_{\text{seq}}} \psi(\theta_0) \psi(\theta_0)^T dP(\mathcal{U}_{n_{\text{seq}}}), \quad (6)$$

where  $P(\mathcal{U}_{n_{\text{seq}}})$  is the cumulative distribution function of  $\mathcal{U}_{n_{\text{seq}}}$ .

We note that (6) depends on  $P(\mathcal{U}_{n_{\text{seq}}})$ . Therefore, the input design problem is to find a cumulative distribution function  $P^{\text{opt}}(\mathcal{U}_{n_{\text{seq}}})$  which optimizes a scalar function of (6). We define this scalar function as  $h : \mathbb{R}^{m \times m} \rightarrow \mathbb{R}$ . To obtain the desired results,  $h$  must be a matrix nondecreasing function [21, pp. 108]. Different choices of  $h$  have been proposed in the literature [10]. Some examples for  $h$  are  $h = \det$ , and  $h = -\text{tr}\{(\cdot)^{-1}\}$ . In this work, we leave to the user the selection of  $h$ .

Since  $P^{\text{opt}}(\mathcal{U}_{n_{\text{seq}}})$  has to be a stationary cumulative distribution function, the optimization must be constrained to the set

$$\mathcal{P} := \{F : \mathbb{R}^{n_{\text{seq}}} \rightarrow \mathbb{R} \mid F(\mathbf{x}) \geq 0, \forall \mathbf{x} \in \mathbb{R}^{n_{\text{seq}}};$$

$$F \text{ is monotone non-decreasing};$$

$$\lim_{\substack{x_i \rightarrow \infty \\ i=\{1, \dots, n_{\text{seq}}\} \\ \mathbf{x}=(x_1, \dots, x_{n_{\text{seq}}})}} F(\mathbf{x}) = 1;$$

$$\left. \int_{v \in \mathbb{R}} dF(v, \mathbf{z}) = \int_{v \in \mathbb{R}} dF(\mathbf{z}, v), \forall \mathbf{z} \in \mathbb{R}^{(n_{\text{seq}}-1)} \right\}. \quad (7)$$

The last condition in (7) (with slight abuse of notation) guarantees that  $F \in \mathcal{P}$  is the cumulative distribution function of a stationary sequence [16].

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

$$\mathcal{P}_{\mathcal{C}} := \{f : \mathcal{C}^{n_{\text{seq}}} \rightarrow \mathbb{R} \mid f(\mathbf{x}) \geq 0, \forall \mathbf{x} \in \mathcal{C}^{n_{\text{seq}}};$$

$$\sum_{\mathbf{x} \in \mathcal{C}^{n_{\text{seq}}}} f(\mathbf{x}) = 1;$$

$$\left. \sum_{v \in \mathcal{C}} f(v, \mathbf{z}) = \sum_{v \in \mathcal{C}} f(\mathbf{z}, v), \forall \mathbf{z} \in \mathcal{C}^{(n_{\text{seq}}-1)} \right\}. \quad (8)$$

The set introduced in (8) will be used to constrain the probability mass function  $p(\mathcal{U}_{n_{\text{seq}}})$ .

The discussion presented in this section can be summarized as

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

$$p^{\text{opt}}(\mathcal{U}_{n_{\text{seq}}}) := \arg \max_{p \in \mathcal{P}_{\mathcal{C}}} h(\mathcal{I}_F(p)), \quad (9)$$

where  $h : \mathbb{R}^{m \times m} \rightarrow \mathbb{R}$  is a matrix nondecreasing function,

$$\mathcal{I}_F(p) = \frac{1}{\lambda_e} \sum_{\mathcal{U}_{n_{\text{seq}}} \in \mathcal{C}^{n_{\text{seq}}}} \sum_{t=1}^{n_{\text{seq}}} \psi(\theta_0) \psi(\theta_0)^T p(\mathcal{U}_{n_{\text{seq}}}), \quad (10)$$

and  $\psi(\theta_0) \in \mathbb{R}^m$  is defined as in (4)-(5). ■

A solution for this problem will be discussed in the next section.

#### IV. INPUT DESIGN VIA GRAPH THEORY

Problem 1 is hard to solve explicitly. The main issues are

- 1) We need to describe the elements in the set  $\mathcal{P}_C$  as a linear combination of basis functions, and
- 2) the sum in (10) is of dimension  $n_{\text{seq}}$ , where  $n_{\text{seq}}$  could be potentially very large.

These issues make Problem 1 computationally intractable. Therefore, we need to develop an approach to solve this problem by using a computational feasible method.

Since  $n_{\text{seq}}$  could be large, Problem 1 can be unfeasible to solve. To address this, we restrict the memory of the stationary process  $u_t$ , i.e., we consider only finite stationary sequences of length, say,  $n_m$ .

To address the first issue, we notice that  $\mathcal{P}_C$  is a convex set. In particular,  $\mathcal{P}_C$  is a polyhedron [21, pp. 31]. Hence, any element of  $\mathcal{P}_C$  can be described as a convex combination of the extreme points of  $\mathcal{P}_C$  [21, pp. 24]. Therefore, if we define  $\mathcal{V}_{\mathcal{P}_C}$  as the set of all the extreme points of  $\mathcal{P}_C$ , composed by  $n_{\mathcal{V}}$  elements, then for all  $f \in \mathcal{P}_C$  we have

$$f = \sum_{i=1}^{n_{\mathcal{V}}} \alpha_i v_i, \quad (11)$$

where  $\alpha_i \geq 0$ ,  $i \in \{1, \dots, n_{\mathcal{V}}\}$ ,

$$\sum_{i=1}^{n_{\mathcal{V}}} \alpha_i = 1, \quad (12)$$

and  $v_i \in \mathcal{V}_{\mathcal{P}_C}$ , for all  $i \in \{1, \dots, n_{\mathcal{V}}\}$ .

Equation (11) says that all the elements in  $\mathcal{P}_C$  can be described by using  $n_{\mathcal{V}}$  elements in the set  $\mathcal{V}_{\mathcal{P}_C}$ .

To find all the elements in  $\mathcal{V}_{\mathcal{P}_C}$ , we need to shift our focus to graph theory. Indeed, we can analyze the set  $\mathcal{C}^{n_m}$  as follows.  $\mathcal{C}^{n_m}$  is composed of  $(c_{\text{seq}})^{n_m}$  elements. Each element in  $\mathcal{C}^{n_m}$  can be viewed as one node in a graph. In addition, the transitions among the elements in  $\mathcal{C}^{n_m}$  are given by the feasible values of  $u_{t+k+1}$  when we move from  $(u_{t+k}, \dots, u_k)$  to  $(u_{t+k+1}, \dots, u_{k+1})$ , for all integers  $k \geq 0$ . The edges among the elements in  $\mathcal{C}^{n_m}$  denote the possible transitions between the states, represented by the nodes of the graph. Figure 2 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 end at node  $(1, 0)$  or  $(1, 1)$  at time  $t+1$ .

The idea to use graph theory to find all the elements in  $\mathcal{V}_{\mathcal{P}_C}$  is related with the concept of prime cycles. In graph theory, a *prime cycle* is an elementary cycle whose set of nodes do not have a proper subset which is an elementary cycle [16, pp. 678]. It has been proved that the prime cycles of a stationary graph can describe all the elements in the set  $\mathcal{V}_{\mathcal{P}_C}$  [16, Theorem 6]. In other words, each prime cycle defines one element  $v_i \in \mathcal{V}_{\mathcal{P}_C}$ . Furthermore, each  $v_i$  corresponds to a uniform distribution whose support is the set of elements of its prime cycle, for all  $i \in \{1, \dots, n_{\mathcal{V}}\}$  [16, pp. 681].

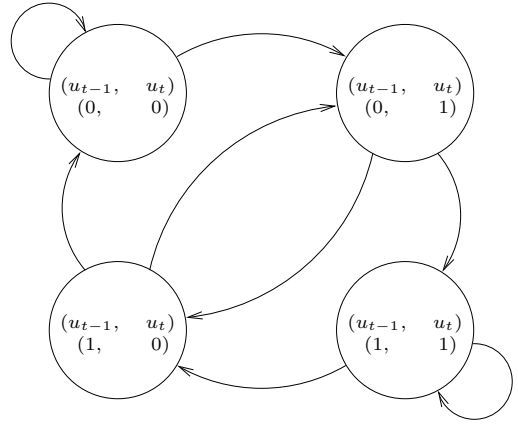


Fig. 2. Example of graph derived from  $\mathcal{C}^{n_m}$ , with  $c_{\text{seq}} = 2$ ,  $n_m = 2$ , and  $\mathcal{C} := \{0, 1\}$ .

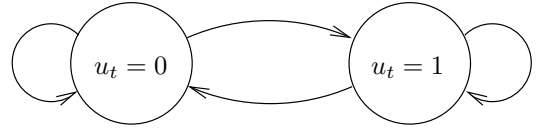


Fig. 3. Example of graph derived from  $\mathcal{C}^{n_m}$ , with  $c_{\text{seq}} = 2$ ,  $n_m = 1$ , and  $\mathcal{C} := \{0, 1\}$ .

Therefore, the elements in  $\mathcal{V}_{\mathcal{P}_C}$  can be described by finding all the prime cycles associated to the stationary graph  $\mathcal{G}_{\mathcal{C}^{n_m}}$  drawn from  $\mathcal{C}^{n_m}$ .

It is known that all the prime cycles associated to  $\mathcal{G}_{\mathcal{C}^{n_m}}$  can be derived from the elementary cycles associated to  $\mathcal{G}_{\mathcal{C}^{(n_m-1)}}$  [16, Lemma 4]. In the literature there are many algorithms for finding all the elementary cycles in a graph. For the examples in Section V, we have used the algorithm presented in [17, pp. 79–80] complemented with the one proposed in [18, pp. 157].

Once all the elementary cycles of  $\mathcal{G}_{\mathcal{C}^{(n_m-1)}}$  are found, we can find all the prime cycles associated to  $\mathcal{G}_{\mathcal{C}^{n_m}}$  by using the idea introduced in [16, Lemma 4]. To illustrate this, we consider the graph depicted in Figure 3. One elementary cycle for this graph is given by  $(0, 1, 0)$ . Using Lemma 4 in [16], 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 to this elementary cycle is given by  $((0, 1), (1, 0), (0, 1))$ .

With all the prime cycles clearly defined for  $\mathcal{G}_{\mathcal{C}^{n_m}}$ , then all the elements in the set  $\mathcal{V}_{\mathcal{P}_C}$  are found. Hence, we can use (11) to describe all the elements in  $\mathcal{P}_C$ . Thus, the solution described here presents a computationally feasible method to address the first issue.

Since we know the distribution  $v_i$  for each prime cycle, with  $v_i \in \mathcal{V}_{\mathcal{P}_C}$ , we can generate an input signal  $\{u_t^i\}_{t=0}^{t=N}$  drawn from  $v_i$ , so that

$$\mathcal{I}_F^{(i)} := \frac{1}{\lambda_e} \sum_{u_{n_m} \in \mathcal{C}^{n_m}} \sum_{t=1}^{n_m} \psi(\theta_0) \psi(\theta_0)^T v_i(u_{n_m})$$

$$\approx \frac{1}{\lambda_e N} \sum_{t=1}^N \psi(\theta_0) \psi(\theta_0)^T, \quad (13)$$

for all  $i \in \{1, \dots, n_\nu\}$ , and  $N$  sufficiently large<sup>1</sup> (in relation to the length of the prime cycles). Notice that  $\psi(\theta_0)$  depends implicitly on  $\{u_t^i\}_{t=0}^{t=N}$  through (4)-(5). Furthermore, each  $\mathcal{I}_F^{(i)}$  is associated to the  $i$ -th prime cycle, for all  $i \in \{1, \dots, n_\nu\}$ .

As an example of how to generate  $\{u_t^i\}_{t=0}^{t=N}$  from  $v_i$ , we use the graph depicted in Figure 2. One prime cycle for this graph is given by  $((0, 1), (1, 0), (0, 1))$ . Therefore, the sequence  $\{u_t^i\}_{t=0}^{t=N}$  is given by taking the last element of each node, i.e.,  $\{u_t^i\}_{t=0}^{t=N} = \{1, 0, 1, 0, \dots, ((-1)^N + 1)/2\}$ .

The approximation of each  $\mathcal{I}_F^{(i)}$  given by (13) reduces the sum (10) from dimension  $n_{\text{seq}}$  to dimension 1. This simplification reduces significantly the computation effort to obtain (10). With this approach, issue 2) is also addressed.

To summarize, the proposed method for input design of signals in  $\mathcal{C}^{n_m}$  can be described as follows

- 1) Compute all the elementary cycles of  $\mathcal{G}_{\mathcal{C}^{(n_m-1)}}$  by using, e.g., [17, pp. 79–80], [18, 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. [16, Lemma 4]).
- 3) Generate the input signals  $\{u_t^i\}_{t=0}^{t=N}$  from the prime cycles of  $\mathcal{G}_{\mathcal{C}^{n_m}}$ , for each  $i \in \{1, \dots, n_\nu\}$ .
- 4) For each  $i \in \{1, \dots, n_\nu\}$ , approximate  $\mathcal{I}_F^{(i)}$  by using (13).
- 5) Define  $\gamma := \{\alpha_1, \dots, \alpha_{n_\nu}\} \in \mathbb{R}^{n_\nu}$ . Find  $\gamma^{\text{opt}} := \{\alpha_1^{\text{opt}}, \dots, \alpha_{n_\nu}^{\text{opt}}\}$  by solving an approximation of Problem 1, given by

$$\gamma^{\text{opt}} := \arg \max_{\gamma \in \mathbb{R}^{n_\nu}} h(\mathcal{I}_F^{\text{app}}(\gamma)), \quad (14)$$

where

$$\mathcal{I}_F^{\text{app}}(\gamma) := \sum_{i=1}^{n_\nu} \alpha_i \mathcal{I}_F^{(i)}, \quad (15)$$

$$\sum_{i=1}^{n_\nu} \alpha_i = 1, \quad (16)$$

$$\alpha_i \geq 0, \text{ for all } i \in \{1, \dots, n_\nu\}, \quad (17)$$

and  $\mathcal{I}_F^{(i)}$  is given by (13), for all  $i \in \{1, \dots, n_\nu\}$ .

The procedure mentioned above computes  $\gamma^{\text{opt}}$  to describe the optimal probability density function  $p^{\text{opt}}(\mathcal{U}_{n_m})$  using (11). Notice that  $\mathcal{I}_F^{\text{app}}(\gamma)$  in (15) is linear in the decision variables. Therefore, for a suitable choice of  $h$ , the problem (14)-(17) becomes convex.

On the other hand, notice that the steps (1)-(3) mentioned above are independent of the system for which the experiment is designed. Therefore, once steps (1)-(3) are computed, then can be reused to design input sequences for different systems.

<sup>1</sup>Note that  $N$  is the number of Monte Carlo simulations to compute (13), and it is not necessarily equal to the length of the experiment  $n_{\text{seq}}$ .

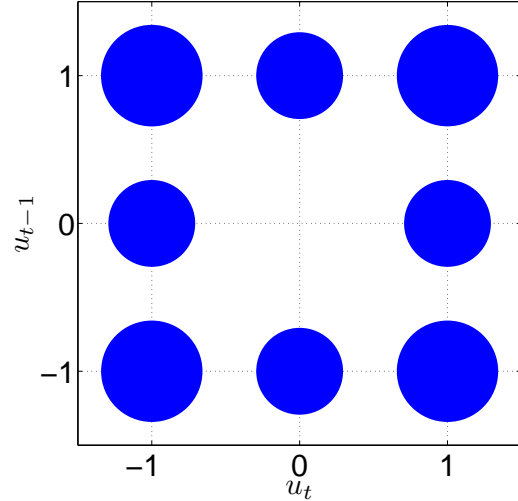


Fig. 4. Plot with the stationary probabilities for the optimal input signal of Example 1. The radius of each disc is proportional to the probability of the state  $\{u_t, u_{t-1}\}$ .

To obtain an input signal from  $p^{\text{opt}}(\mathcal{U}_{n_m})$ , we need to compute a Markov chain associated to the elements in  $\mathcal{C}^{n_m}$ . We can find one transition matrix  $A \in \mathbb{R}^{(c_{\text{seq}})^{n_m} \times (c_{\text{seq}})^{n_m}}$  for the equivalent Markov chain

$$\Pi_{k+1} := A \Pi_k, \quad (18)$$

by using algorithms presented in the literature (e.g., Metropolis-Hastings algorithm [22], [23]). Notice that each entry of  $\Pi_k \in \mathbb{R}^{(c_{\text{seq}})^{n_m}}$  in (18) represents one element in  $\mathcal{C}^{n_m}$ . To use the algorithms presented in [22], [23] we need to determine the stationary probabilities of each element in  $\mathcal{C}^{n_m}$ , which can be computed as follows. We know that each vertex in  $\mathcal{V}_{\mathcal{P}_C}$  has a uniform distribution with support equal to the set of input vectors in the associated prime cycle. Therefore, the stationary probability of each  $\mathbf{x} \in \mathcal{C}^{n_m}$  is given by

$$\mathbf{P}\{\mathbf{X} = \mathbf{x}\} = \sum_{i=1}^{n_\nu} \alpha_i^{\text{opt}} v_i(\mathbf{x}). \quad (19)$$

Equation (19) can be used to construct  $\Pi^s \in \mathbb{R}^{(c_{\text{seq}})^{n_m}}$ , where each entry in  $\Pi^s$  is associated to the stationary probability of one element in  $\mathcal{C}^{n_m}$ . Given  $\Pi^s$ , we can find one matrix  $A$  such that

$$\Pi^s = A \Pi^s. \quad (20)$$

Finally, the transition matrix  $A$  can be used to compute the input sequence by running the Markov chain with a random initial state  $\Pi_0$ .

## V. NUMERICAL EXAMPLES

The previous section described a method to compute a solution for Problem 1. In this section we will show that the method is consistent with reported algorithms in the literature.

**Example 1:** In this example we will solve the input design problem for the system in Figure 1, with

$$G_0(\mathcal{U}_t) = G_1(q, \theta) u_t + G_2(q, \theta) u_t^2, \quad (21)$$

where

$$G_1(q, \theta) = \theta_1 + \theta_2 q^{-1}, \quad (22)$$

$$G_2(q, \theta) = \theta_3 + \theta_4 q^{-1}, \quad (23)$$

and  $q$  denotes the shift operator, i.e.,  $q^{-1} u_t := u_{t-1}$ . We assume that  $e_t$  is Gaussian white noise with variance  $\lambda_e = 1$ . This system has been introduced as an example in [14].

We will solve Problem 1 by considering  $h(\cdot) = \det(\cdot)$ , and a ternary sequence ( $c_{\text{seq}} = 3$ ) of length  $n_m = 2$ . For this example, we define  $\mathcal{C} := \{-1, 0, 1\}$ .

To solve (14)-(17) we consider  $N = 5 \cdot 10^3$  in (13). The implementation of (14)-(17) was made in `Matlab` by using `cvx` toolbox [24].

The simulation results give an optimal cost  $\det(\mathcal{I}_F^{\text{app}}) = 0.1796$  (c.f.  $\det(P^{-1}) = 0.18$  for the same example in [14]). Figure 4 shows the optimal stationary probabilities for each state  $\{u_t, u_{t-1}\}$  (c.f. Figure 4(a) in [14])<sup>2</sup>. The results presented here show that the proposed method is consistent with previous results in the literature [14], when  $G_0$  is defined as (21)-(23). ■

Example 1 shows that this method is equivalent to the method introduced in [14] when  $G_0$  has a nonlinear FIR-type structure.

The results in this article can be also employed when amplitude constraints are considered in the input sequence by forcing  $u_t$  to belong to a finite alphabet. The next example shows an application in that direction.

**Example 2:** In this example we consider the mass-spring-damper system introduced in [9]. The continuous input  $u$  is the force applied to the mass and the output  $y$  is the mass position. The continuous-time system is described by the transfer function

$$G_0(s) = \frac{\frac{1}{m}}{s^2 + \frac{c}{m}s + \frac{k}{m}}, \quad (24)$$

with  $m = 100$  [Kg],  $k = 10$  [N/m], and  $c = 6.3246$  [Ns/m]. This choice results in the natural frequency  $\omega_n = 0.3162$  [rad/s], and the damping  $\xi = 0.1$ . The noise  $e_t$  is white with zero mean and variance  $\lambda_e = 10^{-4}$ . The system (24) is sampled by using a zero-order-hold with sampling period  $T_s = 1$  [s]. This gives the discrete-time system

$$G_0(\mathcal{U}_t) = \frac{4.86 \cdot 10^{-3} q^{-1} + 4.75 \cdot 10^{-3} q^{-2}}{1 - 1.84 q^{-1} + 0.94 q^{-2}} u_t. \quad (25)$$

As a model, we define

$$G(\mathcal{U}_t; \theta) = \frac{\theta_1 q^{-1} + \theta_2 q^{-2}}{1 + \theta_3 q^{-1} + \theta_4 q^{-2}} u_t, \quad (26)$$

where

$$\theta = [\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4]^T. \quad (27)$$

<sup>2</sup>The use of disc plots to represent the optimal input in Figure 4 is considered to ease comparison with the results in [14], where this visual representation is used.

We will solve Problem 1 for two cost functions:  $h(\cdot) = -\text{tr}\{(\cdot)^{-1}\}$  and  $h(\cdot) = \det(\cdot)$ , subject to a binary sequence ( $c_{\text{seq}} = 2$ ) of length  $n_m = 2$ . In this example, we define  $\mathcal{C} := \{-1, 1\}$ , and  $N = 5 \cdot 10^3$ . The optimization is carried out on `Matlab` by using `cvx` toolbox.

The solution of Problem 1 for this example gives  $\text{tr}\{(\mathcal{I}_F^{\text{app}})^{-1}\} = 0.1108$  and  $\det(\mathcal{I}_F^{\text{app}}) = 1.8036 \cdot 10^{12}$ . Figure 5 presents the stationary probabilities of the optimal input signal for both cost functions. We can see that the stationary probabilities depend on the cost function  $h$ . However, we see that both cost functions assign higher stationary probabilities to the states  $(-1, -1)$  and  $(1, 1)$ .

We can compare the performance of our approach with the method introduced in [9]. For this purpose, we generate an input sequence of length  $N$  by running the Markov chain associated to the stationary distribution in Figure 5(a), and the 4-states Markov chain presented in [9]. To guarantee that the input is a realization of a stationary process, we discard the first  $10^6$  outputs of the Markov chain. The results for the sampled information matrix are<sup>3</sup>  $\text{tr}\{(\mathcal{I}_F)^{-1}\} = 1.8233 \cdot 10^{-4}$  for the 4-states Markov chain presented in [9], and  $\text{tr}\{(\mathcal{I}_F)^{-1}\} = 1.6525 \cdot 10^{-4}$  for our method. Therefore, the approach in this paper gives better results for the example introduced in [9].

To have an idea of the computation time required for this example, the optimization was solved in a laptop Dell Latitude E6430, equipped with Intel Core i7 2.6 [GHz] processor, and 8 [Gb] of RAM memory. The time required from the computation of elementary cycles to the computation of stationary probabilities is 1.9 seconds<sup>4</sup>. ■

The numerical examples presented in this section show that the proposed method is suitable for input design for systems with output-error-type structure, and when amplitude constraints on the input are required.

## VI. CONCLUSIONS

In this paper we have developed a new method to compute input signals for systems with arbitrary nonlinearities. The method is based on the optimization of a scalar cost function of the information matrix with respect to the probability density function of a stationary input. The optimal probability density function is used to compute the optimal input signal. An approach based on graph theory is used to derive a computationally efficient algorithm. This approach assumes that the input can adopt a finite set of values. An important feature of this method is that, by a suitable definition of the cost function, the optimization problem is convex even for nonlinear systems. Numerical examples show that this method is consistent with previous results in the literature, when we assume that the system has a particular structure. The method can also be used for input design with amplitude limitations.

<sup>3</sup>Notice that our results are consistent with those reported in [9], since the scaling factor  $N$  is not considered here.

<sup>4</sup>A time bound for the computation of elementary cycles is given by  $O(c_{\text{seq}}^m (c_{\text{seq}} + 1)(c_e + 1))$ , where  $c_e$  is the number of elementary cycles [17, p. 77].

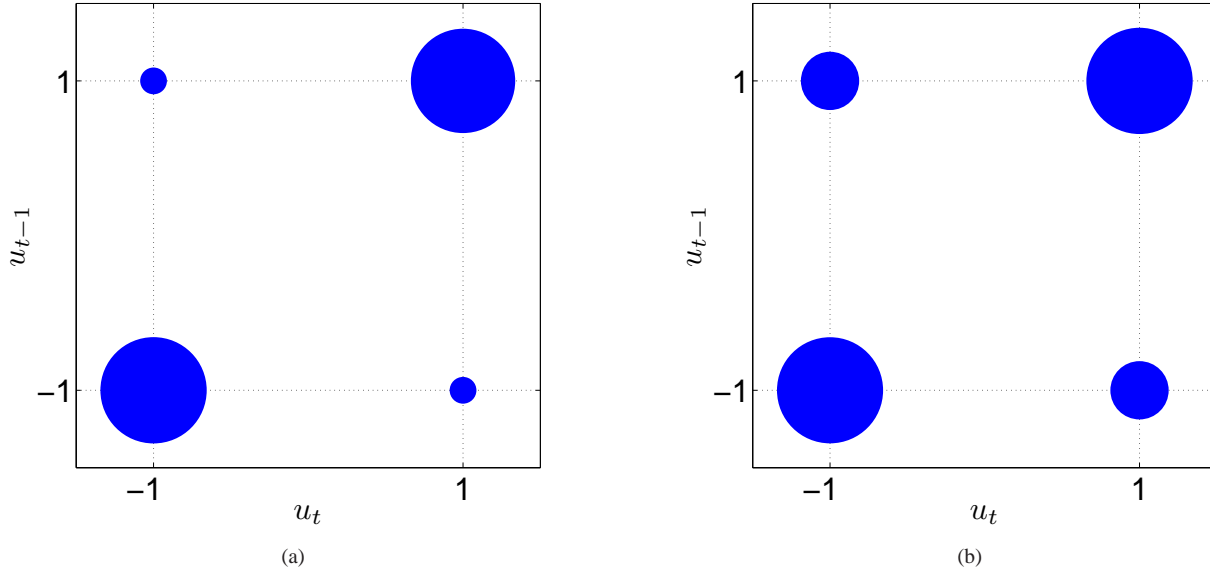


Fig. 5. Plot with the stationary probabilities for the optimal input signal in Example 2. The radius of each disc is proportional to the probability of the state  $\{u_t, u_{t-1}\}$ . Figure 5(a):  $h(\cdot) = -\text{tr}\{(\cdot)^{-1}\}$ . Figure 5(b):  $h(\cdot) = \det(\cdot)$ .

## VII. ACKNOWLEDGMENTS

The authors thank to Marco Forgone for the fruitful discussions during his time at the Automatic Control Lab in KTH.

## REFERENCES

- [1] D.R. Cox, *Planning of experiments*, New York: Wiley, 1958.
- [2] G.C. Goodwin and R.L. Payne, *Dynamic System Identification: Experiment Design and Data Analysis*, Academic Press, New York, 1977.
- [3] V.V. Fedorov, *Theory of optimal experiments*, New York: Academic Press, 1972.
- [4] P. Whittle, "Some general points in the theory of optimal experiment design," *Journal of Royal Statistical Society*, vol. 1, pp. 123–130, 1973.
- [5] R. Hildebrand and M. Gevers, "Identification for control: Optimal input design with respect to a worst-case  $\nu$ -gap cost function," *SIAM Journal of Control Optimization*, vol. 41, no. 5, pp. 1586–1608, 2003.
- [6] M. Gevers, "Identification for control: from the early achievements to the revival of experiment design," *European Journal of Control*, vol. 11, pp. 1–18, 2005.
- [7] H. Jansson and H. Hjalmarsson, "Input design via LMIs admitting frequency-wise model specifications in confidence regions," *IEEE Transactions on Automatic Control*, vol. 50, no. 10, pp. 1534–1549, Oct. 2005.
- [8] K. Lindqvist and H. Hjalmarsson, "Optimal input design using linear matrix inequalities," in *IFAC Symposium on System Identification*, Santa Barbara, California, USA, July 2000.
- [9] 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*, Shanghai, P.R. China, 2009, pp. 1557–1562.
- [10] C.R. Rojas, J.S. Welsh, G.C. Goodwin, and A. Feuer, "Robust optimal experiment design for system identification," *Automatica*, vol. 43, no. 6, pp. 993–1008, June 2007.
- [11] 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.
- [12] H. Hjalmarsson and J. Mårtensson, "Optimal input design for identification of non-linear systems: Learning from the linear case," in *American Control Conference*, New York, United States, 2007, pp. 1572–1576.
- [13] T.L. Vincent, C. Novara, K. Hsu, and K. Poola, "Input design for structured nonlinear system identification," in *15th IFAC Symposium on System Identification*, Saint-Malo, France, 2009, pp. 174–179.
- [14] C. Larsson, H. Hjalmarsson, and C.R. Rojas, "On optimal input design for nonlinear FIR-type systems," in *49th IEEE Conference on Decision and Control*, Atlanta, USA, 2010, pp. 7220–7225.
- [15] R.B. Gopaluni, T.B. Schön, and A.G. Wills, "Input design for nonlinear stochastic dynamic systems - A particle filter approach," in *18th IFAC World Congress*, Milano, Italy, 2011.
- [16] A. Zaman, "Stationarity on finite strings and shift register sequences," *The Annals of Probability*, vol. 11, no. 3, pp. 678–684, Aug. 1983.
- [17] D.B. Johnson, "Finding all the elementary circuits of a directed graph," *SIAM Journal on Computing*, vol. 4, no. 1, pp. 77–84, March 1975.
- [18] R. Tarjan, "Depth-First Search and Linear Graph Algorithms," *SIAM Journal on Computing*, vol. 1, no. 2, pp. 146–160, June 1972.
- [19] 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*, vol. 47, no. 7, pp. 1388–1398, 2011.
- [20] L. Ljung, *System Identification. Theory for the User, 2nd ed.*, Upper Saddle River, NJ: Prentice-Hall, 1999.
- [21] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
- [22] W.K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.
- [23] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing Markov chain on a graph," *SIAM Review*, vol. 46, no. 4, pp. 667–689, Oct. 2004.
- [24] M.C. Grant and S.P. Boyd, *The CVX users' guide*, CVX Research, Inc., 2nd. edition, January 2013.