A graph theoretical approach to input design for identification of nonlinear dynamical models

Patricio E. Valenzuela [a], Cristian R. Rojas [a], Håkan Hjalmarsson [a]

[a] *Department of Automatic Control and ACCESS Linnaeus Center, School of Electrical Engineering, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden.*

Abstract

In this paper the problem of optimal input design for model identification is studied. The optimal input signal is designed by maximizing a scalar cost function of the information matrix, where the input signal is a realization of a stationary process with finite memory, with its range being a finite set of values. It is shown that the feasible set for this problem can be associated with the prime cycles in the graph of possible values and transitions for the input signal. A realization of the optimal input signal is generated by running a Markov chain associated with the feasible set, where the transition matrix is built using a novel algorithm developed for de Bruijn graphs. The proposed method can be used to design inputs for nonlinear output-error systems, which are not covered in previous results. In particular, since the input is restricted to a finite alphabet, it can naturally handle amplitude constraints. Finally, our approach relies on convex optimization even for systems having a nonlinear structure. A numerical example shows that the algorithm can be successfully used to perform input design for nonlinear output-error models.

Key words: System identification, input design, Markov chains.

1 INTRODUCTION

Input design considers the construction of an input signal to maximize the information obtained from an experiment. Some of the initial contributions in this line were presented in the works of Cox [4], Fedorov [7], and Goodwin and Payne [11], where the latter contribution is concerned with input design for the identification of dynamic systems. Since then, several contributions in input design have been developed (see [33,14,9], and the references therein).

In the case of dynamic systems, input design maximizes the information related to the estimated parameters of the system. By maximizing a scalar function of the Fisher information matrix [22] related to the accuracy of the estimated model for a particular application, we obtain an input signal that can be used to identify a good application model of the unknown system. The results in this area are mainly focused on input design for linear systems, where powerful tools can be applied to solve the problem [10,22,17,20,25]. Several methods have been reported in the literature involving, e.g., linear matrix inequalities (LMI) [17,20,26,32], Markov chains [3,2], and time domain gradient based schemes [10,27], among others.

In recent years, the interest on input design has shifted from linear to nonlinear systems. Unfortunately, most of the tools used for input design for linear systems based on frequency domain techniques are no longer valid for the nonlinear case, which implies that new techniques need to be developed in this domain. One approach to input design for the identification of nonlinear systems is introduced in [15], where a linear systems perspective is considered. Extensions to a class of finite-impulse-response type systems are developed in [19], where a characterization of probability density functions is employed. Input design for structured nonlinear identification is introduced in [30,31], where the system is assumed to be an interconnection of known linear systems and unknown static nonlinearities. An input design method for a general class of nonlinear systems is presented in

⋆ This paper was not presented at any IFAC meeting. Corresponding author Patricio E. Valenzuela.
This work was supported by the Swedish Research Council under contracts 621-2011-5890 and 621-2009-4017, and by the European Research Council under the advanced grant LEARN, contract 267381.

Email addresses: pva@kth.se (Patricio E. Valenzuela), crro@kth.se (Cristian R. Rojas), hjalmars@kth.se (Håkan Hjalmarsson).

[12], based on a particle filter used to approximate the cost function, which is optimized over a particular class of input vectors using stochastic approximation. The methods previously mentioned [15,19,12] in general are highly complex (usually ending up with non-convex optimization formulations, e.g., [12]) and are restricted to particular model structures (e.g., [15,19]) and/or particular classes of input signals (e.g., white noise filtered through an ARX filter [12]). Moreover, except for the results in [3,2,19], the methods introduced cannot handle input design with amplitude constraints. Amplitude constraints can arise due to power and/or physical limitations in the system. Therefore, input design with amplitude constraints also requires further considerations.

As a first contribution, in this article we develop a novel approach for input design in nonlinear systems. This approach considers the design of an input sequence for models with additive white noise at the output, which extends the class of nonlinear systems considered in [19]. The input is constrained to be a stationary process with a finite set of possible values, and where the associated probability mass function (pmf) has finite memory, i.e., a Markov chain of fixed order. Therefore, the optimization considers the design of an optimal pmf which maximizes the information obtained from the experiment, quantified as a scalar function of the information matrix. By using notions of graph theory, we can express the set of feasible pmf's as a convex combination of the pmf's of the prime cycles describing the vertices of the set. Since the prime cycles can be explicitly computed by known algorithms [34,18], the optimization problem becomes easy to pose. Furthermore, for standard choices of the cost function, the problem is convex even for nonlinear systems, which simplifies the problem formulation discussed in [3,2]. Finally, since the input is restricted to a finite set of possible values, the method naturally incorporates amplitude limitations.

Once the optimization problem is solved, we obtain the optimal stationary distribution over the possible states of the memory describing the pmf. To obtain an input with the desired stationary distribution, we must be able to design a feasible transition probability matrix satisfying the constraints of the graph associated with our problem. Unfortunately, due to the asymmetric structure of the graph, we cannot use standard Markov chain Monte Carlo (MCMC) methods [13,1] to determine a transition matrix for the graph. Therefore, and as a second contribution of this paper, we develop a method to design a valid transition probability matrix for graphs generated from stationary processes with finite memory.

The present article can be seen as an extension of the results in [19] and [3,2]. The main difference with [3,2] is that we optimize over the stationary pmf associated with the Markov chain, instead of directly optimizing over the transition probabilities. This approach results in a convex problem (which cannot be achieved in [3,2], where optimization techniques guaranteeing local optima must be employed). In [19] a similar approach to the one presented in our article is discussed, but restricted to the analysis to nonlinear FIR systems. By using the finite memory property of nonlinear FIR models, the input design problem in [19] is solved in terms of an input realization of finite length. However, the results in [19] cannot be employed to design input sequences for identification of more general nonlinear output-error models, since the models will generally depend on the entire past input sequence. In this line, our article extends the analysis to more general nonlinear model structures, which includes nonlinear FIR systems (see Example 1 in [29] where the results are consistent with those introduced in [19]).

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 [25] or via an adaptive procedure, where the input signal is re-designed as more information is being collected from the system [24,8]. This issue goes beyond the scope of this article and it will not be addressed here.

A previous description of the proposed method has been presented in [29]. In this paper we give a more detailed explanation of the input design technique, a method of the generation of the input signal from an optimal finite state Markov chain stationary distribution, and new numerical examples.

The rest of the paper is organized as follows. Section 2 introduces some background on graph theory. Section 3 presents the input design problem. In Section 4 we solve the input design problem using elements of graph theory. Section 5 presents a novel method to generate an input signal from the optimal stationary distribution obtained in Section 4. Section 6 illustrates the results with numerical examples. Finally, Section 7 presents conclusions.

Notation. In the sequel, we denote by $\mathbb{C}$ the complex set, by $\mathbb{Z}$ the integer set, 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. Given $z\in\mathbb{C}$, $|z|$ denotes its modulus. The expected value with respect to the random variable x and the probability measure are denoted by $\mathbf{E}_x\{\cdot\}$, and $\mathbf{P}\{\cdot\}$, respectively. det and tr stand for the determinant and the trace functions, respectively. Given a finite set T, $\#T$ denotes its cardinality.

2 PRELIMINARIES ON GRAPH THEORY

In this section we provide a brief background on the concepts of graph theory used in the next sections. Our notation follows that of [18, 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 $p_{vu} = (v = v_1, v_2, \ldots, v_k = u)$ such that $(v_i, v_{i+1}) \in \mathcal{X}$ for all $i \in \{1, \ldots, k-1\}$. 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.

An n-dimensional *de Bruijn graph* of m symbols [5] is a directed graph representing overlaps between sequences of symbols (c.f. Figure 2). It has m^n vertices, consisting of all possible sequences of length n derived from the given symbols. The same symbol can appear multiple times in a sequence. If we have a set of symbols $\mathcal{C} = \{s_1, \ldots, s_m\}$ then the set of n-dimensional vertices is

$$\begin{aligned}\mathcal{V} = \mathcal{C}^n = \{&(s_1, \ldots, s_1, s_1), (s_1, \ldots, s_1, s_2), \ldots, \\ &(s_1, \ldots, s_1, s_m), (s_1, \ldots, s_2, s_1), \ldots, \\ &(s_m, \ldots, s_m, s_m)\}\,. \end{aligned} \tag{1}$$

If one of the vertices can be expressed as another vertex by shifting all its symbols one place to the left and adding a new symbol at the end, then the latter has a directed edge to the former vertex. Thus the set of directed edges is

$$\begin{aligned}\mathcal{X} = \{&((v_1, v_2, \ldots, v_n), (r_1, r_2, \ldots, r_n)) : \\ &v_2 = r_1, v_3 = r_2, \ldots, v_n = r_{n-1}\}\,. \end{aligned} \tag{2}$$

In the following we will denote by $\mathcal{G}_{\mathcal{C}^n}$ the n-dimensional de Bruijn graph derived from $\mathcal{C}^n$.

3 PROBLEM FORMULATION

Consider the single-input, single-output time invariant system depicted in Figure 1. Here, G_0 is a dynamic system (possibly nonlinear), defined for $t \geq 1$ as

$$G_0(\mathcal{U}_t) := \begin{cases} x_{t+1} = f_0(x_t, u_t) \\ z_t = h_0(x_t, u_t) \\ x_1 = \mu \end{cases}, \tag{3}$$

$\{e_t\}$ is a white noise sequence with zero mean and finite variance λ_e, $u_t \in \mathbb{R}$ is the input, $x_t \in \mathbb{R}^{n_x}$ are the internal states of G_0 with initial condition $\mu \in \mathbb{R}^{n_x}$, $y_t \in \mathbb{R}$ is the measured output, and $\mathcal{U}_t := (u_t, \ldots, u_1)$. Notice that G_0 is defined as in (3) to simplify notation. We consider a model structure G, defined for any $\theta \in \Theta \subset \mathbb{R}^m$ as

$$G(\mathcal{U}_t; \theta) := \begin{cases} x_{t+1} = f(x_t, u_t; \theta) \\ z_t = h(x_t, u_t; \theta) \\ x_1 = \mu \end{cases}. \tag{4}$$

We assume that there exists a $\theta_0 \in \Theta$ such that $G(\mathcal{U}_t; \theta_0) = G_0(\mathcal{U}_t)$ [22], i.e., there is no undermodelling. Notice that the noise e_t is assumed to enter only at the output.

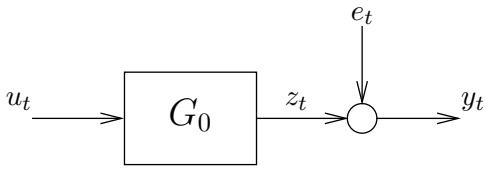

Fig. 1. Block diagram of a (possibly nonlinear) system.

To continue, we introduce the following definition:

Definition 1 *Consider a bounded signal* $\{u_t\}$, $|u_t| \leq K$ *($K > 0$), and a nonlinear system* $y_t = G_0(\mathcal{U}_t)$. *We say that* G_0 *is* exponentially stable *if and only if there are constants* $C > 0$ *(depending possibly on* K*),* $0 < \delta < 1$, *such that for all* t, $s \in \mathbb{Z}$,

$$|G_0(\mathcal{U}_t) - G_0(\mathcal{U}_t^s)| < C\delta^{t-s}\,, \tag{5}$$

where $\mathcal{U}_t^s := \{u_t, u_{t-1}, \ldots, u_{s+1}, 0, 0, \ldots\}$. ■

We notice that Definition 1 differs from that given by [21] since it considers deterministic systems, and it is not defined over moments of order 4 (as it is in the definition introduced in [21]).

The objective in this article is to design an input signal $\mathcal{U}_{n_{\text{seq}}} = (u_{n_{\text{seq}}}, \ldots, u_1)$ as a realization of a stationary process, such that the system (3) can be estimated with maximum accuracy as defined by a scalar function of the Fisher information matrix $\mathcal{I}_F$ [22]. In this article, the Fisher information matrix $\mathcal{I}_F$ is defined as the expected value of the conditioned Fisher information matrix $\mathcal{I}_F^e$ over the stationary process $\mathcal{U}_{n_{\text{seq}}}$. $\mathcal{I}_F^e$ can be computed as

$$\mathcal{I}_F^e = \frac{1}{\lambda_e}\mathbf{E}_e\left\{\sum_{t=1}^{n_{\text{seq}}} \psi_t^{\theta_0}(\mathcal{U}_t)\psi_t^{\theta_0}(\mathcal{U}_t)^T \middle| \mathcal{U}_{n_{\text{seq}}}\right\}, \tag{6}$$

where

$$\psi_t^{\theta_0}(\mathcal{U}_t) := \left.\frac{d\,\hat{y}_t(\mathcal{U}_t)}{d\theta}\right|_{\theta=\theta_0}, \tag{7a}$$

$$\hat{y}_t(\mathcal{U}_t) := G(\mathcal{U}_t; \theta)\,, \tag{7b}$$

and θ, $\theta_0 \in \Theta$. Since we are interested in computing $\mathcal{U}_{n_{\text{seq}}}$ as a realization of a stationary process, we will maximize a scalar cost function of the *per-sample* Fisher information matrix, defined as

$$\begin{aligned}\mathcal{I}_F &:= \mathbf{E}_{\mathcal{U}_{n_{\text{seq}}}}\{\mathcal{I}_F^e\} \\ &= \frac{1}{\lambda_e}\mathbf{E}_{\mathcal{U}_{n_{\text{seq}}},\,e}\left\{\sum_{t=1}^{n_{\text{seq}}} \psi_t^{\theta_0}(\mathcal{U}_t)\psi_t^{\theta_0}(\mathcal{U}_t)^T\right\}. \end{aligned} \tag{8}$$

On the other hand, we note that equation (7b) does not depend on the noise realization. Therefore, we can

rewrite (8) as

$$\mathcal{I}_F = \frac{1}{\lambda_e} \int_{\mathcal{U}_{n_{\text{seq}}} \in \mathbb{R}^{n_{\text{seq}}}} \sum_{t=1}^{n_{\text{seq}}} \psi_t^{\theta_0}(\mathcal{U}_t)\psi_t^{\theta_0}(\mathcal{U}_t)^T \, dP(\mathcal{U}_{n_{\text{seq}}}) \,, \tag{9}$$

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

We note that (9) depends on $P(\mathcal{U}_{n_{\text{seq}}})$. Therefore, the input design problem we will consider is to find a cdf $P^{\text{opt}}(\mathcal{U}_{n_{\text{seq}}})$ which optimizes a scalar function of (9). We define this function as $h : \mathbb{R}^{m\times m} \to \mathbb{R}$. As it is customary in input design [11,17,22], h is assumed to be a concave function. Several choices of h have been proposed in the literature [25]. Some examples for h are det, and $-\operatorname{tr}\{(\cdot)^{-1}\}$. In this work, we leave to the user the selection of h.

Since n_{seq} can be large, the input design problem involves dealing with a potentially very high dimensional integral (9), which can be computationally intractable. To address this issue, we can restrict the input signal u_t to be a stationary process of finite memory, i.e., u_t can be assumed to be a Markov process of order n_m (say) [1]. This means that $P(\mathcal{U}_{n_{\text{seq}}})$ can be completely described by its n_m-dimensional projection $P(\mathcal{U}_{n_m})$ [34]. Recall that an *n_m-dimensional projection* $P(\mathcal{U}_{n_m})$ of $P(\mathcal{U}_{n_{\text{seq}}})$ is a cdf $P(\mathcal{U}_{n_m})$ associated with the stationary vector $\mathcal{U}_{n_m}$, which is extended to the space of stationary vectors $\mathcal{U}_{n_{\text{seq}}}$ to define $P(\mathcal{U}_{n_{\text{seq}}})$ [34, Theorem 1].

Since $P(\mathcal{U}_{n_m})$ has to be the projection of a stationary cdf, the optimization must be constrained to the set [2]

$$\begin{aligned}\mathcal{P} := \Big\{ & F : \mathbb{R}^{n_m} \to \mathbb{R} |\, F(\mathbf{x}) \geq 0,\, \forall \mathbf{x} \in \mathbb{R}^{n_m}; \\ & F \text{ is monotone non-decreasing ;} \\ & \lim_{x_i \to \infty} F(x_1, \ldots, x_{n_m}) = 1, \quad \text{for } i = 1, \ldots, n_m \\ & \int_{v\in\mathbb{R}} dF(v,\, \mathbf{z}) = \int_{v\in\mathbb{R}} dF(\mathbf{z},\, v)\,, \forall \mathbf{z} \in \mathbb{R}^{(n_m-1)} \Big\} \,. \end{aligned} \tag{10}$$

The last condition in (10) (with some slight abuse of notation) guarantees that $F \in \mathcal{P}$ is the projection of the cdf of a stationary sequence [34]. Indeed, the last condition in (10) states that the cdf obtained by marginalizing over $u_{n_{\text{seq}}}$ is the same to the one obtained by marginalizing over u_1, which retrieves the shift invariant property associated with stationary processes.

[1] Notice that the assumption on n_m can be relaxed by making n_m close to n_{seq}.

[2] Recall that the joint cdf is defined as $F(x_1, \ldots, x_{n_m}) := \mathbf{P}\{X_1 \leq x_1, \ldots, X_{n_m} \leq x_{n_m}\}$, with $\{X_i\}_{i=1}^{n_m}$ as random variables.

For computational tractability, we further constrain u_t to belong to a finite alphabet $\mathcal{C}$ with cardinality c_{seq}. With this assumption, it is convenient to work with the pmf $p(\mathcal{U}_{n_m})$ rather than the cdf $P(\mathcal{U}_{n_m})$. In addition, we can define the constraint set of the pmf $p(\mathcal{U}_{n_m})$ as:

$$\begin{aligned}\mathcal{P}_{\mathcal{C}} := \{ & f : \mathcal{C}^{n_m} \to \mathbb{R} |\, f(\mathbf{x}) \geq 0,\, \forall \mathbf{x} \in \mathcal{C}^{n_m}; \\ & \sum_{\mathbf{x}\in\mathcal{C}^{n_m}} f(\mathbf{x}) = 1; \\ & \sum_{v\in\mathcal{C}} f(v,\, \mathbf{z}) = \sum_{v\in\mathcal{C}} f(\mathbf{z},\, v)\,, \forall \mathbf{z} \in \mathcal{C}^{(n_m-1)} \Big\} \,. \end{aligned} \tag{11}$$

Based on Definition 1, we require the following assumption over $\psi_t^{\theta_0}(\mathcal{U}_t)\psi_t^{\theta_0}(\mathcal{U}_t)^T$:

Assumption 1 *Consider $\psi_t^{\theta_0}(\mathcal{U}_t)$ defined in (7). Then the function $\psi_t^{\theta_0}(\mathcal{U}_t)\psi_t^{\theta_0}(\mathcal{U}_t)^T$ is exponentially stable with constants $C_\psi > 0$, $0 < \delta_\psi < 1$.* ■

Finally, the input design problem can be summarized as:

Problem 1 *Design an optimal input signal $\mathcal{U}_{n_{seq}}^{\text{opt}} \in \mathcal{C}^{n_{\text{seq}}}$ as a realization from the projected pmf $p^{\text{opt}}(\mathcal{U}_{n_m})$, given by*

$$p^{\text{opt}}(\mathcal{U}_{n_m}) := \arg\max_{p\in\mathcal{P}_{\mathcal{C}}} h(\mathcal{I}_F(p)) \,, \tag{12}$$

where $h : \mathbb{R}^{m\times m} \to \mathbb{R}$ is a concave function,

$$\mathcal{I}_F(p) = \frac{1}{\lambda_e} \sum_{\mathcal{U}_{n_{\text{seq}}}\in\mathcal{C}^{n_{\text{seq}}}} \sum_{t=1}^{n_{\text{seq}}} \psi_t^{\theta_0}(\mathcal{U}_t)\psi_t^{\theta_0}(\mathcal{U}_t)^T \, p(\mathcal{U}_{n_{\text{seq}}}) \,, \tag{13}$$

and where $\psi_t^{\theta_0}(\mathcal{U}_t) \in \mathbb{R}^m$ is defined as in (7), satisfying Assumption 1. ■

A subtle issue still needs to be addressed: how should $p(\mathcal{U}_{n_m})$ be parameterized? Since the information matrix $\mathcal{I}_F$ in (13) is defined in terms of $p(\mathcal{U}_{n_{\text{seq}}})$ (the full pmf of the input sequence) instead of $p(\mathcal{U}_{n_m})$, it is natural to attempt to parameterize $p(\mathcal{U}_{n_{\text{seq}}})$ in terms of a finite number of elements of $\mathcal{P}_{\mathcal{C}}$, i.e., projections of stationary pmf's, since these elements can be extended to a full stationary pmf $p(\mathcal{U}_{n_{\text{seq}}})$. Such a parameterization is described in the next section.

4 INPUT DESIGN VIA GRAPH THEORY

To parameterize the elements of $\mathcal{P}_{\mathcal{C}}$, we first notice that this is a convex set, and, in particular, a polyhedron [23, pp. 170], since it is described by linear inequalities. Hence, any element of $\mathcal{P}_{\mathcal{C}}$ can be described as a convex combination of its extreme points [23, Corollaries 18.3.1

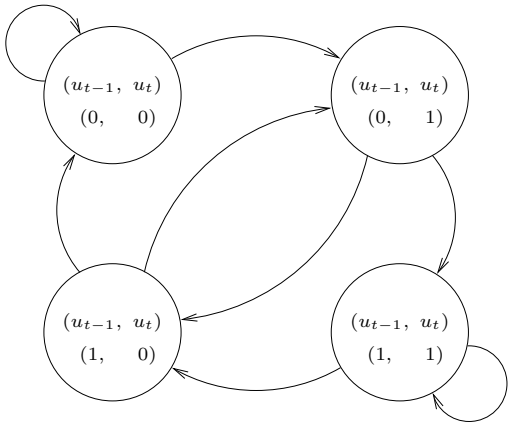

Fig. 2. Example of the de Bruijn graph derived from $\mathcal{C}^{n_m}$, with $c_{\text{seq}} = 2$, $n_m = 2$, and $\mathcal{C} := \{0,\ 1\}$.

and 19.1.1]. In other words, if we define $\mathcal{V}_{\mathcal{P}_\mathcal{C}} := \{w_i, i = 1, ..., n_\mathcal{V}\}$ as the set of all the extreme points of $\mathcal{P}_\mathcal{C}$, then for all $f \in \mathcal{P}_\mathcal{C}$ we have

$$f = \sum_{i=1}^{n_\mathcal{V}} \alpha_i \, w_i \,, \tag{14}$$

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

$$\sum_{i=1}^{n_\mathcal{V}} \alpha_i = 1 \,. \tag{15}$$

The set $\mathcal{V}_{\mathcal{P}_\mathcal{C}}$ can be characterized in a graph-theoretical manner. To this end, notice that the set of possible values for $(u_{t-n_m+1},\ \ldots,\ u_t)$, $\mathcal{C}^{n_m}$, is composed of $(c_{\text{seq}})^{n_m}$ elements, which can be viewed as nodes in a graph. In addition, the transitions between the elements in $\mathcal{C}^{n_m}$, as described by a stationary process of memory n_m, are given by the possible values of u_{t+1} when we move from $(u_{t-n_m+1}, \ldots,\ u_t)$ to $(u_{t-n_m+2}, \ldots,\ u_{t+1})$, for all integers $t \geq 0$. The edges between the elements in $\mathcal{C}^{n_m}$ denote the possible transitions between the states, represented by the nodes of the graph. The resulting graph corresponds to a de Bruijn graph (c.f. Section 2). 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 at node (0, 1) at time t, then we can only transit to node (1, 0) or (1, 1) at time $t + 1$.

In order to describe the elements of $\mathcal{V}_{\mathcal{P}_\mathcal{C}}$, we need 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 [34, pp. 678]. To continue we will need the following theorem:

Theorem 1 *The prime cycles of the de Bruijn graph of a Markov process of memory n_m are in one-to-one correspondence with the elements of $\mathcal{V}_{\mathcal{P}_\mathcal{C}}$. In addition, each $w_i \in \mathcal{V}_{\mathcal{P}_\mathcal{C}}$ corresponds to a uniform distribution whose support is the set of elements of a prime cycle.* ■

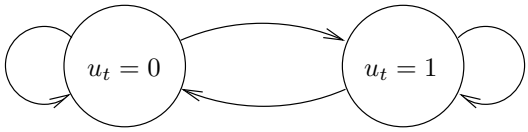

Fig. 3. Example of the de Bruijn graph derived from $\mathcal{C}^{n_m}$, with $c_{\text{seq}} = 2$, $n_m = 1$, and $\mathcal{C} := \{0,\ 1\}$.

Proof. See [34, Theorem 6] and [34, pp. 681] for a formal proof. ■

Using Theorem 1 we can describe all the elements in $\mathcal{V}_{\mathcal{P}_\mathcal{C}}$ by finding all the prime cycles associated with the de Bruijn graph $\mathcal{G}_{\mathcal{C}^{n_m}}$ drawn from $\mathcal{C}^{n_m}$. To find all the prime cycles in $\mathcal{G}_{\mathcal{C}^{n_m}}$ we will use the following lemma:

Lemma 1 *All the prime cycles associated with $\mathcal{G}_{\mathcal{C}^{n_m}}$ can be derived from the elementary cycles of $\mathcal{G}_{\mathcal{C}^{(n_m-1)}}$.* ■

Proof. See [34, Lemma 4] for the details. ■

Lemma 1 states that finding all the prime cycles in $\mathcal{G}_{\mathcal{C}^{n_m}}$ is equivalent to finding all the elementary cycles in $\mathcal{G}_{\mathcal{C}^{(n_m-1)}}$, which can be determined using standard graph algorithms [3] (see, e.g., [18,28]). To illustrate this procedure, consider the graph depicted in Figure 3. One elementary cycle for this graph is given by (0, 1, 0). Using Lemma 1, 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 ((0, 1), (1, 0), (0, 1)).

Once all the prime cycles of $\mathcal{G}_{\mathcal{C}^{n_m}}$ are found, the set $\mathcal{V}_{\mathcal{P}_\mathcal{C}}$ is fully determined. Then, for each $w_i \in \mathcal{V}_{\mathcal{P}_\mathcal{C}}$ we can generate a corresponding input signal by generating the corresponding prime cycle. Therefore, we can define the information matrix corresponding to prime-cycle i and element $w_i \in \mathcal{V}_{\mathcal{P}_\mathcal{C}}$ as

$$\begin{aligned} \mathcal{I}_F^{(i)} &:= \frac{1}{\lambda_e} \sum_{\mathcal{U}_{n_m} \in \mathcal{C}^{n_m}} \sum_{t=1}^{n_m} \psi_t^{\theta_0}(\mathcal{U}_t^i) \psi_t^{\theta_0}(\mathcal{U}_t^i)^T \, w_i(\mathcal{U}_{n_m}) \\ &\approx \frac{1}{\lambda_e \, N_{\text{sim}}} \sum_{t=1}^{N_{\text{sim}}} \psi_t^{\theta_0}(\mathcal{U}_t^i) \psi_t^{\theta_0}(\mathcal{U}_t^i)^T \,, \end{aligned} \tag{16}$$

for all $i \in \{1, \ldots, n_\mathcal{V}\}$, and N_{sim} sufficiently large (in relation to the length of the prime cycles). Notice that each $\mathcal{I}_F^{(i)}$ is associated with the i-th prime cycle. The approximation (16) is a necessary step since the explicit

[3] For the examples in Section 5, we have used the algorithm presented in [18, pp. 79–80] complemented with the one proposed in [28, pp. 157].

computation of $\mathcal{I}_F^{(i)}$ for the nonlinear model (4) is often intractable. Instead of approximating $\mathcal{I}_F^{(i)}$ as an average over different realizations of the input sequence $\mathcal{U}_{n_{\rm seq}}$, we consider an approximation of $\mathcal{I}_F^{(i)}$ as an average over one realization of length $N_{\rm sim}$, where $N_{\rm sim}$ is sufficiently large. The approximation (16) is possible since $\psi_t^{\theta_0}(\mathcal{U}_t)\psi_t^{\theta_0}(\mathcal{U}_t)^T$ satisfies Assumption 1 (see Appendix A for a proof of this statement). The approximation error incurred when (16) is employed to compute $\mathcal{I}_F^{(i)}$ is of order $\delta_\psi^{N_{\rm sim}}$.

As an example of how to generate $\{u_t^i\}_{t=0}^{N_{\rm sim}}$ for a particular w_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 $\{u_t^i\}_{t=0}^{N_{\rm sim}}$ is given by taking the last element of each node, i.e., $\{u_t^i\}_{t=0}^{N_{\rm sim}} = \{1, 0, 1, 0, \ldots, ((-1)^{N_{\rm sim}}+1)/2\}$.

Once the approximation (16) is made for all the elements in $\mathcal{V}_{\mathcal{P}_\mathcal{C}}$, we can compute an approximation of the information matrix $\mathcal{I}_F(p)$ associated with the elements in $\mathcal{P}_\mathcal{C}$ as a convex combination of the $\mathcal{I}_F^{(i)}$'s, $i \in \{1, \ldots, n_\mathcal{V}\}$. Indeed, if we define $\gamma := \{\alpha_1, \ldots, \alpha_{n_\mathcal{V}}\} \in \mathbb{R}^{n_\mathcal{V}}$, we can write

$$\mathcal{I}_F^{\rm app}(\gamma) := \sum_{i=1}^{n_\mathcal{V}} \alpha_i \mathcal{I}_F^{(i)}\,, \tag{17a}$$

$$\sum_{i=1}^{n_\mathcal{V}} \alpha_i = 1\,, \tag{17b}$$

$$\alpha_i \geq 0\,,\ \text{for all } i \in \{1, \ldots, n_\mathcal{V}\}\,, \tag{17c}$$

where $\mathcal{I}_F^{\rm app}(\gamma)$ is the approximation of the information matrix $\mathcal{I}_F(p)$ associated with the elements of $\mathcal{P}_\mathcal{C}$.

To summarize, the proposed method for the design of input 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., [18, pp. 79–80], [28, 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. [34, Lemma 4]).
(3) Generate the input signals $\{u_t^i\}_{t=0}^{N_{\rm sim}}$ from the prime cycles of $\mathcal{G}_{\mathcal{C}^{n_m}}$, for each $i \in \{1, \ldots, n_\mathcal{V}\}$.
(4) For each $i \in \{1, \ldots, n_\mathcal{V}\}$, approximate $\mathcal{I}_F^{(i)}$ using (16).
(5) Define $\gamma = \{\alpha_1, \ldots, \alpha_{n_\mathcal{V}}\} \in \mathbb{R}^{n_\mathcal{V}}$. Find $\gamma^{\rm opt} := \{\alpha_1^{\rm opt}, \ldots, \alpha_{n_\mathcal{V}}^{\rm opt}\}$ by solving the following approximation of Problem 1:

$$\gamma^{\rm opt} = \arg\max_{\gamma \in \mathbb{R}^{n_\mathcal{V}}} h(\mathcal{I}_F^{\rm app}(\gamma))\,, \tag{18}$$

where $h : \mathbb{R}^{m\times m} \to \mathbb{R}$ is a concave function,

$$\mathcal{I}_F^{\rm app}(\gamma) = \sum_{i=1}^{n_\mathcal{V}} \alpha_i \mathcal{I}_F^{(i)}\,, \tag{19a}$$

$$\sum_{i=1}^{n_\mathcal{V}} \alpha_i = 1\,, \tag{19b}$$

$$\alpha_i \geq 0\,,\ \text{for all } i \in \{1, \ldots, n_\mathcal{V}\}\,, \tag{19c}$$

and $\mathcal{I}_F^{(i)}$ is given by (16).

The procedure mentioned above computes $\gamma^{\rm opt}$ which defines the optimal pmf $p^{\rm opt}(\mathcal{U}_{n_m})$ via (14). Notice that $\mathcal{I}_F^{\rm app}(\gamma)$ in (19a) is linear in the decision variables. Therefore, if $-h$ is convex, problem (18)-(19) can be solved using convex optimization tools.

Remark 1 *Steps (1) to (3) are independent of the system for which the input is designed. Therefore, once the steps (1) to (3) are computed, they can be re-used for input design in different systems.*

Remark 2 *The approximate solution to Problem 1 given by* (18) *might not be unique. In that case,* (18) *will return the weighting vector associated with one optimal pmf. Moreover, even if the optimal pmf $p^{\rm opt}(\mathcal{U}_{n_m})$ is unique, the optimal input realization $\mathcal{U}_{n_{\rm seq}}$ is not unique. Indeed, $\mathcal{U}_{n_{\rm seq}}$ is a vector of random variables with stationary distribution $p^{\rm opt}(\mathcal{U}_{n_m})$.*

Remark 3 *The computational cost associated with the proposed technique is mostly described by the effort required to compute the elementary cycles. A time bound for the computation of elementary cycles is given by $O(c_{\rm seq}^{n_m}(c_{\rm seq}+1)(c_e+1))$, where c_e is the number of elementary cycles given by [18, p. 77]*

$$c_e := \sum_{i=1}^{c_{\rm seq}^{n_m}-1} \binom{c_{\rm seq}^{n_m}}{c_{\rm seq}^{n_m}-i+1} (c_{\rm seq}^{n_m}-i)!\,. \tag{20}$$

As expected, the computational time grows with the number of elementary cycles. How to avoid the computation of the entire set of elementary cycles is ongoing research.

5 INPUT SIGNAL GENERATION

In this section we develop a procedure to generate an input sequence $\mathcal{U}_{n_{\rm seq}}$ from the optimal projected pmf $p^{\rm opt}(\mathcal{U}_{n_m})$ computed above. To this end, notice that we can associate $\mathcal{G}_{\mathcal{C}^{n_m}}$ with the discrete-time Markov chain [6]

$$\pi_{k+1} = A\,\pi_k\,, \tag{21}$$

where $A \in \mathbb{R}^{\mathcal{C}^{n_m}\times\mathcal{C}^{n_m}}$ is a transition probability matrix, and $\pi_k \in \mathbb{R}^{\mathcal{C}^{n_m}}$ is the state vector. In this case, there

is a one-to-one correspondence between each entry of $\pi_k \in \mathbb{R}^{\mathcal{C}^{n_m}}$ and an element of $\mathcal{C}^{n_m}$.

Based on this association, $p^{\text{opt}}(\mathcal{U}_{n_m})$ corresponds to $\Pi^s \in \mathbb{R}^{\mathcal{C}^{n_m}}$, the stationary distribution of a Markov chain of the form (21). Therefore, in order to generate an input sequence $\mathcal{U}_{n_{\text{seq}}}$ from $p^{\text{opt}}(\mathcal{U}_{n_m})$, we can design a Markov chain having $p^{\text{opt}}(\mathcal{U}_{n_m})$ as its stationary distribution, and simulate this Markov chain to generate $\mathcal{U}_{n_{\text{seq}}}$ from its samples.

If $A_{rl} \in \mathbb{R}$ denotes the (r,l)-entry in A, then a valid A for the Markov chain (21) must satisfy

$$A_{rl} \geq 0\,, \text{ for all } r, l \in \mathcal{C}^{n_m}\,, \tag{22}$$

$$\sum_{r \in \mathcal{C}^{n_m}} A_{rl} = 1\,, \text{ for all } l \in \mathcal{C}^{n_m}\,, \tag{23}$$

$$A_{rl} = 0\,, \text{ if } (l,\, r) \notin \mathcal{X}\,. \tag{24}$$

Notice that the indices of A are not numerical, but belong to $\mathcal{C}^{n_m}$.

It can be proved that a matrix A satisfying (22)-(23) has 1 as an eigenvalue [16]. Furthermore, if the Markov chain is ergodic, the eigenvector $\Pi^s \in \mathbb{R}^{\mathcal{C}^{n_m}}$ associated with this eigenvalue is the stationary pmf of $\mathcal{C}^{n_m}$ (up to a scaling factor), satisfying

$$\Pi^s = A\Pi^s\,. \tag{25}$$

Our task is then to design a transition probability matrix satisfying (22)-(25). There is an extensive literature on how to optimize the mixing time of the resulting Markov chain (see, e.g., [1,13] and the references therein). However, these works assume that the graph is undirected or reversible, which implies that A must have a particular structure (e.g., A being a symmetric matrix). Since the structure of the graph $\mathcal{G}_{\mathcal{C}^{n_m}}$ does not satisfy in general the mentioned properties, most existing methods cannot be applied here.

Below we develop a method to design a transition probability matrix for the de Bruijn graph $\mathcal{G}_{\mathcal{C}^{n_m}}$. The idea is that, if we parameterize the transition probabilities of a Markov chain with memory n in terms of the stationary probabilities of a Markov chain with memory $n+1$, we obtain a convex problem, as discussed in Section 4. Once the stationary probabilities are optimized, the proposed algorithm gives a unique mapping between the optimized stationary probabilities and the transition matrix with memory n, by using

$$\mathbf{P}\{u_t | u_{t-1},\, \ldots,\, u_{t-n}\} = \frac{\mathbf{P}\{(u_t,\, \ldots,\, u_{t-n})\}}{\sum_{u_t \in \mathcal{C}} \mathbf{P}\{(u_t,\, \ldots,\, u_{t-n})\}}\,, \tag{26}$$

where $\mathbf{P}$ denotes the stationary probability measure of the Markov chain (i.e., defined by the entries of Π^s), and $\mathbf{P}\{\cdot|\cdot\}$ denotes the conditional probability measure.

To continue, we need to introduce the following result:

Fact 1 *The stationary probability measure* $\mathbf{P}$*, corresponding to* $p^{\text{opt}}(\mathcal{U}_{n_m})$*, satisfies*

$$\sum_{r=1}^{c_{\text{seq}}} \mathbf{P}\{(v_1,\, \ldots,\, v_{n_m-1},\, s_r)\} = \sum_{r=1}^{c_{\text{seq}}} \mathbf{P}\{(s_r,\, v_1,\, \ldots,\, v_{n_m-1})\}\,, \tag{27}$$

for all $(v_1,\, \ldots,\, v_{n_m-1}) \in \mathcal{C}^{n_m-1}$. ■

Fact 1 follows since $p^{\text{opt}}(\mathcal{U}_{n_m})$ is the projection of a stationary distribution, c.f. (11). Based on this fact, we can design a transition probability matrix A for $\mathcal{G}_{\mathcal{C}^{n_m}}$ as follows:

Algorithm 1 *Generation of a transition probability matrix* A *from* $p^{\text{opt}}(\mathcal{U}_{n_m})$*:*

- *For each* $r \in \mathcal{C}^{n_m}$*, define*

$$\mathcal{N}_r := \{l \in \mathcal{C}^{n_m} : (l,\, r) \in \mathcal{X}\}\,. \tag{28}$$

  In other words, $\mathcal{N}_r$ *is the set of ancestors of* r*.*
- *For each* $r,\, l \in \mathcal{C}^{n_m}$*, let*

$$A_{rl} = \begin{cases} \dfrac{\mathbf{P}\{r\}}{\sum\limits_{k \in \mathcal{N}_r} \mathbf{P}\{k\}}\,, & \text{if } l \in \mathcal{N}_r \text{ and } \sum\limits_{k \in \mathcal{N}_r} \mathbf{P}\{k\} \neq 0\,, \\ \dfrac{1}{\#\mathcal{N}_r}\,, & \text{if } l \in \mathcal{N}_r \text{ and } \sum\limits_{k \in \mathcal{N}_r} \mathbf{P}\{k\} = 0\,, \\ 0\,, & \text{otherwise.} \end{cases} \tag{29}$$

■

Algorithm 1 introduces a method to design valid transition probability matrices when Π^s satisfies Fact 1. The next theorem establishes the correctness of the algorithm.

Theorem 2 *Given a stationary probability measure* $\mathbf{P}$ *satisfying Fact 1, then the matrix* $A \in \mathbb{R}^{\mathcal{C}^{n_m} \times \mathcal{C}^{n_m}}$ *designed by Algorithm 1 is a transition probability matrix satisfying* (22)-(25). ■

Proof. Properties (22) and (24) are trivially satisfied by the construction of A. To establish (23) and (25), we

need to analyze the structure of the transition probability matrix A associated with a de Bruijn graph. From the definition of $\mathcal{X}$ (c.f. (2)), we have that

$$\sum_{l\in\mathcal{N}_r}\mathbf{P}\{l\}=\sum_{l=1}^{c_{\rm seq}}\mathbf{P}\{(s_l,\,r(1),\,\ldots,\,r(n_m-1))\}\,. \quad (30)$$

To proceed, we need to define

$$\mathcal{T}_r:=\{l\in\mathcal{C}^{n_m}:\,(r,\,l)\in\mathcal{X}\},\quad r\in\mathcal{C}^{n_m}, \quad (31)$$

i.e., (31) is the set of descendants of r. From the definition of $\mathcal{X}$, we have that $\#\mathcal{T}_r=\#\mathcal{N}_r=c_{\rm seq}$.

First, we will prove (23). Consider first a $l\in\mathcal{C}^{n_m}$ such that $\sum_{k\in\mathcal{N}_r}\mathbf{P}\{k\}\neq 0$ for all $r\in\mathcal{T}_l$. Then,

$$\mathcal{T}_l=\{(l(2),\,\ldots,\,l(n_m),\,s_1),\,\ldots,\,(l(2),\,\ldots,\,l(n_m),\,s_{c_{\rm seq}})\}\,. \quad (32)$$

In addition, for any $r\in\mathcal{T}_l$,

$$\mathcal{N}_r=\{(s_1,\,l(2),\,\ldots,\,l(n_m)),\,\ldots,\,(s_{c_{\rm seq}},l(2),\,\ldots,\,l(n_m))\}\,. \quad (33)$$

Equation (33) shows that the sets $\mathcal{N}_r$ are equal for all $r\in\mathcal{T}_l$. Therefore, the sums $\sum_{k\in\mathcal{N}_r}\mathbf{P}\{k\}$ are equal (and nonzero) for all $r\in\mathcal{T}_l$, hence

$$\begin{aligned}\sum_{r\in\mathcal{C}^{n_m}}A_{rl}&=\sum_{r\in\mathcal{T}_l}\frac{\mathbf{P}\{r\}}{\sum_{k\in\mathcal{N}_r}\mathbf{P}\{k\}}\\&=\frac{\sum_{r\in\mathcal{T}_l}\mathbf{P}\{r\}}{\sum_{k\in\mathcal{N}_{\tilde r}}\mathbf{P}\{k\}}\,,\end{aligned} \quad (34)$$

for any $\tilde r\in\mathcal{T}_l$. Furthermore, in the light of (32)-(33), we can rewrite (34) as

$$\begin{aligned}\sum_{r\in\mathcal{C}^{n_m}}A_{rl}&=\frac{\sum_{r=1}^{c_{\rm seq}}\mathbf{P}\{(l(2),\,\ldots,\,l(n_m),\,s_r)\}}{\sum_{k=1}^{c_{\rm seq}}\mathbf{P}\{s_k,\,l(2),\,\ldots,\,l(n_m)\}}\\&=1,\end{aligned} \quad (35)$$

where the last equality follows from Fact 1.

On the other hand, if $l\in\mathcal{C}^{n_m}$ is such that $\sum_{k\in\mathcal{N}_r}\mathbf{P}\{k\}=0$ for all $r\in\mathcal{C}^{n_m}$, we write

$$\sum_{r\in\mathcal{C}^{n_m}}A_{rl}=\sum_{r\in\mathcal{T}_l}\frac{1}{\#\mathcal{N}_r}=\sum_{r\in\mathcal{T}_l}\frac{1}{\#\mathcal{T}_l}=1\,, \quad (36)$$

The results presented in (35)-(36) establish (23).

Now we prove (25). For each $r\in\mathcal{C}^{n_m}$ such that $\sum_{k\in\mathcal{N}_r}\mathbf{P}\{k\}\neq 0$, we have that the r-th element of the product $A\Pi^s$ (denoted by π_r^s) is given by

$$\begin{aligned}\pi_r^s&=\frac{\mathbf{P}\{r\}}{\sum_{l\in\mathcal{N}_r}\mathbf{P}\{l\}}\sum_{k\in\mathcal{N}_r}\mathbf{P}\{k\}\\&=\mathbf{P}\{r\}\,.\end{aligned} \quad (37)$$

On the other hand, for each $r\in\mathcal{C}^{n_m}$ such that $\sum_{k\in\mathcal{N}_r}\mathbf{P}\{k\}=0$, we can consider an $l\in\mathcal{C}^{n_m}$ such that $r\in\mathcal{T}_l$. According to (32), (33), and using Fact 1, we can conclude that

$$\sum_{\tilde r\in\mathcal{T}_l}\mathbf{P}\{\tilde r\}=\sum_{k\in\mathcal{N}_r}\mathbf{P}\{k\}=0\,, \quad (38)$$

which implies that

$$\mathbf{P}\{k\}=0 \quad (39)$$

for all $k\in\mathcal{T}_l$, and in particular, for $k=r$. Since $l\in\mathcal{C}^{n_m}$ is arbitrary, (39) is true for all $l\in\mathcal{C}^{n_m}$ such that $r\in\mathcal{T}_l$. Hence, (37) is also satisfied for each $r\in\mathcal{C}^{n_m}$ such that $\sum_{k\in\mathcal{N}_r}\mathbf{P}\{k\}=0$, which establishes (25). This concludes the proof. ■

The transition probability matrix designed using Algorithm 1 has the following property:

Theorem 3 *The transition probability matrix $A\in\mathbb{R}^{\mathcal{C}^{n_m}\times\mathcal{C}^{n_m}}$ designed by Algorithm 1 has all its eigenvalues in the region $\mathbf{D}:=\{z\in\mathbb{C}:\,|z|\leq 1\}$. In addition, A has at most $(c_{\rm seq})^{n_m-1}$ nonzero eigenvalues in $\mathbf{D}$.* ■

Proof. The first statement follows since A is a transition probability matrix, according to Theorem 2.

To establish the second statement, notice that, from (32)-(33), for each $l\in\mathcal{C}^{n_m}$ we have that $\mathcal{N}_r$ is the same for all $r\in\mathcal{T}_l$, which means that the columns of A can be partitioned into $(c_{\rm seq})^{n_m-1}$ groups of $c_{\rm seq}$ identical columns. Therefore, the number of nonzero eigenvalues of A in $\mathbf{D}$ is at most $(c_{\rm seq})^{n_m-1}$. This concludes the proof. ■

Remark 4 *There are, in general, several transition matrices having a given $\mathbf{P}$ as stationary probability measure, subject to a graph constraint. Algorithm 1 provides only one such choice based on the constraint that A corresponds to a Markov chain of order n instead of $n+1$. Among those transition matrices, it would be preferable to select the one with the fastest mixing time, i.e., for which the Markov chain reaches the stationary distribution as quickly as possible, in a given sense. The most common criterion to define mixing time is the second largest eigenvalue modulus (SLEM). A Monte Carlo study, for $c_{\rm seq}=2$ and $n_m=2$, based on uniform sampling from*

the set of transition matrices giving a specific $\mathbf{P}$ *(which can be shown to be a polytope) has empirically shown that the* A *matrix given by Algorithm 1 is within the* $7\,\%$ *with lowest SLEM, which suggests that Algorithm 1 gives a reasonable (but improvable) mixing time. One way to further reduce the SLEM of* A *is by performing gradient descent, starting from the matrix designed in Algorithm 1. Another option to reduce the SLEM is by exploiting the full memory of the Markov chain; this, however, will be further explored in a future publication.*

Remark 5 *In some cases the optimal solution may not be ergodic, i.e., it may consist of disconnected sub graphs. To solve this, it is possible to add a small perturbation to the resulting Markov chain of the input, to make it ergodic.*

Remark 6 *For simplicity, the method introduced in this article is discussed for SISO models. However, an immediate extension of this technique to the MIMO case can be done. Indeed, in the MIMO case with* n_u *inputs, the states associated with each node in the de Bruijn graph are the possible values of an* $n_u \times n_m$ *matrix, where the* i*-th row describes the states for the stationary process in the* i*-th input. With this modification, the method can be directly employed to solve input design for MIMO models.*

6 NUMERICAL EXAMPLES

In this section we present numerical examples of the method proposed in this paper. Additional examples for this method can be found in [29].

Example 1 *Consider the block diagram depicted in Figure 1, where* $\{e_t\}$ *is Gaussian white noise sequence with zero mean and variance* $\lambda_e = 1$*. The system is described by*

$$G_0(\mathcal{U}_t) := \begin{cases} x_{t+1} = \dfrac{1}{\theta_1^0 + x_t^2} + u_t \\ z_t = \theta_2^0\, x_t^2 \\ x_1 = 0 \end{cases}, \qquad (40)$$

where $\theta_0 = \begin{bmatrix} \theta_1^0 & \theta_2^0 \end{bmatrix}^T = \begin{bmatrix} 0.8 & 2 \end{bmatrix}^T$*. Notice that the system* (40) *cannot be described as a Wiener-Hammerstein system, since the state equation is nonlinear on the state* x_t*.*

To analyze our method, we consider the model

$$G(\mathcal{U}_t;\theta) := \begin{cases} x_{t+1} = \dfrac{1}{\theta_1 + x_t^2} + u_t \\ z_t = \theta_2\, x_t^2 \\ x_1 = 0 \end{cases}, \qquad (41)$$

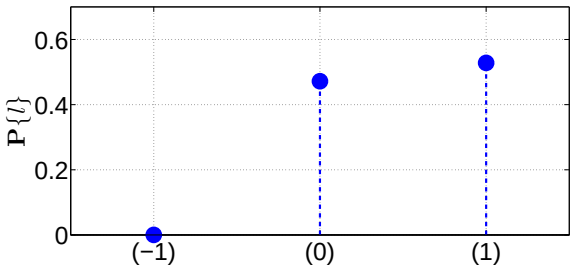

Fig. 4. Plot with the stationary probabilities for the optimal input signal in Example 1.

with $\theta = \begin{bmatrix} \theta_1 & \theta_2 \end{bmatrix}^T = \theta_0$*.*

We design an input of $n_{\text{seq}} = 10^4$ *samples as a realization of the pmf obtained by solving Problem 1, with* $n_m = 1$*, and* $\mathcal{C} = \{-1, 0, 1\}$*. Problem 1 will be solved for* $h(\cdot) = \log\{\det(\cdot)\}$*. Note that the cost function implies that the optimization problem* (18) *is convex in the decision variables* γ *defined in* (19a)*.*

The stationary probabilities $\mathbf{P}\{l\}$ *obtained as the solution of Problem 1 are presented in Figure 4. In this figure, we notice that the stationary probabilities cannot be associated with a random sample among the states (in which case we would see a uniform distribution).*

Given the stationary probabilities in Figure 4, we can use Algorithm 1 to design a transition matrix to generate samples with the desired distribution. The resulting transition matrix by using Algorithm 1 is given by

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0.47 & 0.47 & 0.47 \\ 0.53 & 0.53 & 0.53 \end{bmatrix}, \qquad (42)$$

where the rows and columns of A *are indexed in the order* (-1)*,* (0)*,* (1)*.*

The SLEM for A *defined as* (42) *is* 0*. Therefore, the number of nonzero eigenvalues is* 1*, as expected by Theorem 3.*

Finally, we generate the input signal by running the Markov chain with transition matrix (42) *with random initial state in* $\{0, 1\}$*, and recording the first* $n_{\text{seq}} = 10^4$ *samples. This is possible since the SLEM associated with the transition probability matrix* (42) *is zero, which implies that the chain will start in the stationary distribution. To compare the result of the new method with those of standard input signals, we compute the cost functions when the input vector of length* n_{seq} *is a realization of a uniformly distributed random variable with support* $[-1, 1]$*. Table 1 presents the results obtained for the cost function when the input is designed with the*

Table 1
Numerical results for the cost function in Example 1.

$h(\mathcal{I}_F)$	Graph	Uniform	Normal	Binary
$\log\{\det(\mathcal{I}_F)\}$	3.67	2.77	3.12	3.47

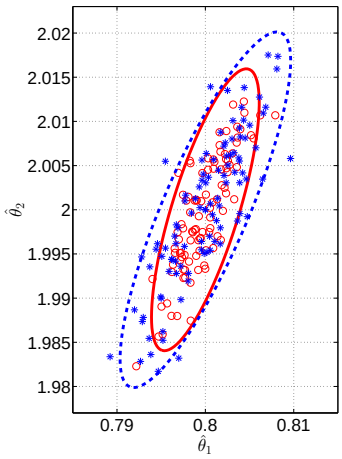

Fig. 5. Plot with the 95% confidence ellipsoids for input sequences of length $n_{\text{seq}} = 10^4$, and the estimated parameters, Example 1. **Blue, dashed line**: Confidence ellipsoid for a binary input sequence (realizations marked with $*$). **Red, continuous line**: Confidence ellipsoid for the optimal input sequence, Case 2 (realizations marked with circles).

proposed method (Graph), and when the input sequence is a realization of a sequence of independent and uniformly distributed random variables with support $[-1, 1]$ *(Uniform). The results show that the proposed method for input design outperforms the experiment result based on uniform samples. In addition, we can also compare our result with those obtained with the input being independent and Gaussian distributed, with zero mean and variance* 1 *(Normal in Table 1), and when the input is a realization of a binary white process with values* -1 *and* 1 *(Binary in Table 1). In this case, the results obtained by Gaussian and Binary distributed inputs are closer to the ones obtained by the proposed input design method. However, our method still improves the input based on random samples.*

As an additional exercise, we can also compute the results obtained when the input is designed for the following cases:

$$\textit{Case 1: } n_m = 2,\ \mathcal{C} = \{-1,\ 0,\ 1\}, \tag{43}$$
$$\textit{Case 2: } n_m = 1,\ \mathcal{C} = \{-1,\ -1/3,\ 1/3,\ 1\}, \tag{44}$$
$$\textit{Case 3: } n_m = 1,\ \mathcal{C} = \{-1,\ -0.5,\ 0,\ 0.5,\ 1\}. \tag{45}$$

Table 2 presents the results when the Markov chains associated with the Cases 1-3 are employed to generate $n_{\text{seq}} = 10^4$ *samples. From these results we conclude that we can increase the information obtained from the input if we extend the memory of the stationary process generating the input sequence. Moreover, the results are significantly better when we only extend the set* $\mathcal{C}$*.*

Table 2
Numerical results for the cost function in Example 1, Cases 1-3.

$h(\mathcal{I}_F)$	Case 1	Case 2	Case 3
$\log\{\det(\mathcal{I}_F)\}$	3.82	4.50	4.48

To analyze the accuracy of the method, we present in Figure 5 the 95% *confidence ellipsoids for the input sequence generated from a random binary distribution, and from an optimal input obtained by solving Case 2. In addition, we also plot* 10^2 *estimated parameters computed by using the data set generated with both input sequences. The results in the figure show that the proposed input design technique decreases the uncertainty region for the estimated parameters, compared with a random input sequence of length* n_{seq}*. This conclusion is also confirmed by the numerical estimates of* θ_0*, which obeys the distribution given by the theoretical bounds. Therefore, the proposed technique is an attractive alternative to increase the accuracy of the parameter estimates obtained with random samples.* ■

Example 2 *Consider the problem introduced in Example 1. As before, we will design an input of* $n_{\text{seq}} = 10^4$ *samples as a realization of the pmf obtained by solving Problem 1, for Case 2 and Case 3 in Example 1. In this case, Problem 1 is solved for* $h(\cdot) = -\operatorname{tr}\{(\cdot)^{-1}\}$*.*

Table 3
Numerical results for the cost function in Example 2.

$h(\mathcal{I}_F)$	Case 2	Case 3	Unif.	Normal	Binary
$\operatorname{tr}\{{\mathcal{I}_F}^{-1}\}$	0.49	0.45	1.21	0.96	0.83

The results obtained by different input sequences of length $n_{\text{seq}} = 10^4$ *are presented in Table 3, where Unif. Normal and Binary represent the results obtained with the random samples defined in Example 1. The results show that the proposed input design technique (Case 2 and Case 3 in Table 3) outperforms the inputs based on random samples.*

To analyze the accuracy for this example, Figure 6 presents the 95% *confidence ellipsoids for the input sequence generated from a random binary distribution, and from an optimal input obtained by solving Case 3. In addition, we also plot* 10^2 *estimated parameters computed by using the data set generated with both input sequences. In the same line than Figure 5, we see that the accuracy of the estimated parameters is improved for the confidence sets, when we compare the optimal input sequence with a random realization from a binary distribution. Therefore, the method presented in this article is an effective approach to design input sequences to identify the system* (40)*.* ■

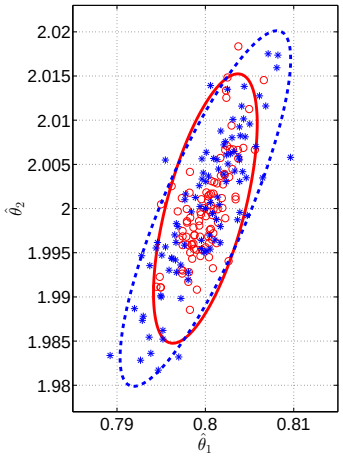

Fig. 6. Plot with the 95% confidence ellipsoids for input sequences of length $n_{\text{seq}} = 10^4$, and the estimated parameters, Example 2. **Blue, dashed line**: Confidence ellipsoid for a binary input sequence (realizations marked with $*$). **Red, continuous line**: Confidence ellipsoid for the optimal input sequence, Case 3 (realizations marked with circles).

7 CONCLUSIONS

In this paper we have developed a novel method to design input signals for system identification. The input signal is restricted to be the realization of a stationary process with finite memory. Thus, the method computes the pmf that maximizes a scalar cost function of the information matrix in the set of such processes. To describe the feasible set we use elements from graph theory, where the vertices of the feasible polyhedron are represented by the prime cycles in a de Bruijn graph. Therefore, the information matrix can be numerically computed as a convex combination of the information matrices associated with those prime cycles. The optimization problem then becomes convex even for nonlinear models.

Once the optimal pmf is computed, we run a Markov chain to obtain an input signal with the desired stationary distribution. For that purpose, we have developed an algorithm to design a transition matrix achieving the stationary pmf. Finally, the numerical example has shown that the proposed method can be successfully applied to design inputs for nonlinear output-error systems, and it outperforms inputs based on random samples.

As a future work on this topic, we will address the issue related with the computational complexity of the proposed method. In particular, we will analyze how the computation of the prime cycles can be reduced. Furthermore, we will consider extensions of the proposed technique to more general model structures.

Acknowledgements

The authors wish to thank the anonymous reviews for providing valuable suggestions to improve this paper.

References

[1] S. Boyd, P. Diaconis, and L. Xiao. Fastest mixing Markov chain on a graph. *SIAM Review*, 46(4):667–689, October 2004.

[2] C. Brighenti. On input design for system identification: input design using Markov chains, MSc. Thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2009.

[3] 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.

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

[5] N.G. de Bruijn and P. Erdos. A combinatorial problem. *Koninklijke Nederlandse Akademie v. Wetenschappen*, 49:758–764, 1946.

[6] J.L. Doob. *Stochastic processes*, volume 101. New York Wiley, 1953.

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

[8] 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.

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

[10] G.C. Goodwin, J.C. Murdoch, and R.L. Payne. Optimal test signal design for linear SISO system identification. *International Journal of Control*, 17(1):45–55, 1973.

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

[12] 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.

[13] W.K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[14] 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.

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

[16] R. Horn and C. Johnson. *Matrix Analysis*. Cambridge University Press (9th reprint), 1999.

[17] 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.

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

[19] 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*, pages 7220–7225, Atlanta, USA, 2010.

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

[21] L. Ljung. Convergence analysis of parametric identification methods. *IEEE Transactions on Automatic Control*, 23(5):770–783, 1978.

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

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

[24] 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.

[25] 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.

[26] B. Sanchez, C.R. Rojas, G. Vandersteen, R. Bragos, and J. Schoukens. On the calculation of the Data-optimal multisine excitation power spectrum for broadband impedance spectroscopy measurements. *Measurement Science and Technology*, 23(8):1–15, 2012.

[27] 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.

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

[29] P.E. Valenzuela, C.R. Rojas, and H. Hjalmarsson. Optimal input design for dynamic systems: a graph theory approach. Presented at the 52^{nd} IEEE Conference on Decision and Control (CDC), Florence, Italy. arXiv:1310.4706, October 2013.

[30] 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.

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

[32] B. Wahlberg, H. Hjalmarsson, and P. Stoica. On optimal input signal design for frequency response estimation. In *49th Conference on Decision and Control*, pages 302–307, Atlanta, USA, 2010.

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

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

A Convergence analysis of the approximation for $\mathcal{I}_F^{(i)}$

In this section we prove that the approximation (16) converges to $\mathcal{I}_F^{(i)}$ as $N_{\rm sim} \to \infty$:

Theorem 4 *If $\{u_t\}$ is periodic of period T satisfying $|u_t| \leq K$ for some $K \geq 0$, and $\psi_t^{\theta_0}(\mathcal{U}_t)\psi_t^{\theta_0}(\mathcal{U}_t)^T$ is exponentially stable, then*

$$\begin{aligned}\lim_{N_{\rm sim}\to\infty} &\frac{1}{N_{\rm sim}}\sum_{t=1}^{N_{\rm sim}} \psi_t^{\theta_0}(\tilde{\mathcal{U}}_{t,-\infty})\psi_t^{\theta_0}(\tilde{\mathcal{U}}_{t,-\infty})^T \\ &= \frac{1}{T}\sum_{t=1}^{T} \psi_t^{\theta_0}(\mathcal{U}_{t,-\infty})\psi_t^{\theta_0}(\mathcal{U}_{t,-\infty})^T \\ &= \int \psi_t^{\theta_0}(\mathcal{U}_{t,-\infty})\psi_t^{\theta_0}(\mathcal{U}_{t,-\infty})^T\, dP(\mathcal{U}_{t,-\infty})\,, \quad \text{(A.1)}\end{aligned}$$

where $\{\tilde{u}_t\}$ is equal to $\{u_t\}$ for $t > 0$ but $\tilde{u}_t = 0$ for $t \leq 0$, $\tilde{\mathcal{U}}_{t,-\infty} := \{\tilde{u}_k\}_{k=-\infty}^{t}, \mathcal{U}_{t,-\infty} := \{u_k\}_{k=-\infty}^{t}$, and P is the probability measure of a Markov chain generating $\{u_t\}$ (a uniform initial probability distribution on the set of possible values of $\mathcal{U}_T$). ■

Proof. Given $\varepsilon > 0$, take S as a multiple of T such that $C\delta_\psi^S < \varepsilon$. Then, for every $t > S$,

$$\begin{aligned}\Big|&\psi_t^{\theta_0}(\tilde{\mathcal{U}}_{t,-\infty})\psi_t^{\theta_0}(\tilde{\mathcal{U}}_{t,-\infty})^T \\ &-\psi_t^{\theta_0}(\tilde{\mathcal{U}}_{t,-\infty}^{t-S})\psi_t^{\theta_0}(\tilde{\mathcal{U}}_{t,-\infty}^{t-S})^T\Big| < C\delta_\psi^S < \varepsilon\,. \quad \text{(A.2)}\end{aligned}$$

On the other hand, since $\{u_t\}$ is periodic of period T, $\mathcal{U}_{t,-\infty}^{t-S}$ takes only a finite number of values for $t > S$ (at most S), we have that for $N_{\rm sim} = mS + n$ (with m, n positive integers):

$$\begin{aligned}&\frac{1}{N_{\rm sim}}\sum_{t=1}^{N_{\rm sim}} \psi_t^{\theta_0}(\tilde{\mathcal{U}}_{t,-\infty})\psi_t^{\theta_0}(\tilde{\mathcal{U}}_{t,-\infty})^T \\ &\quad= \frac{mS}{N_{\rm sim}}\frac{1}{mS}\left[\sum_{t=1}^{mS} \psi_t^{\theta_0}(\tilde{\mathcal{U}}_{t,-\infty})\psi_t^{\theta_0}(\tilde{\mathcal{U}}_{t,-\infty})^T \right. \\ &\qquad\left. + \sum_{t=mS+1}^{N_{\rm sim}} \psi_t^{\theta_0}(\tilde{\mathcal{U}}_{t,-\infty})\psi_t^{\theta_0}(\tilde{\mathcal{U}}_{t,-\infty})^T\right] \\ &\quad= \frac{mS}{N_{\rm sim}}\frac{1}{mS}\sum_{t=1}^{mS}\left[\psi_t^{\theta_0}(\tilde{\mathcal{U}}_{t,-\infty}^{t-S})\psi_t^{\theta_0}(\tilde{\mathcal{U}}_{t,-\infty}^{t-S})^T + \eta_t\right] \\ &\qquad+ \frac{1}{N_{\rm sim}}\sum_{t=mS+1}^{N_{\rm sim}} \psi_t^{\theta_0}(\tilde{\mathcal{U}}_{t,-\infty})\psi_t^{\theta_0}(\tilde{\mathcal{U}}_{t,-\infty})^T \\ &\qquad= \frac{mS}{N_{\rm sim}}\frac{1}{T}\sum_{t=1}^{T} \psi_t^{\theta_0}(\mathcal{U}_{t,-\infty}^{t-S})\psi_t^{\theta_0}(\mathcal{U}_{t,-\infty}^{t-S})^T \\ &\qquad\qquad+ \frac{1}{N_{\rm sim}}\sum_{t=1}^{mS}[\mu_t + \eta_t] \\ &\qquad+ \frac{1}{N_{\rm sim}}\sum_{t=mS+1}^{N_{\rm sim}} \psi_t^{\theta_0}(\tilde{\mathcal{U}}_{t,-\infty})\psi_t^{\theta_0}(\tilde{\mathcal{U}}_{t,-\infty})^T\,, \quad \text{(A.3)}\end{aligned}$$

where $\mu_t,\ \eta_t \in [-\varepsilon,\ \varepsilon]$. Thus, the second term in (A.3) is bounded by 2ε. Moreover, the third term in (A.3) tends to 0 as $N_{\rm sim} \to \infty$ (since it consists of a sum of a most S terms). Therefore,

$$\left| \lim_{N_{\rm sim}\to\infty} \frac{1}{N_{\rm sim}} \sum_{t=1}^{N_{\rm sim}} \psi_t^{\theta_0}(\tilde{\mathcal{U}}_{t,\,-\infty}) \psi_t^{\theta_0}(\tilde{\mathcal{U}}_{t,\,-\infty})^T - \frac{1}{T} \sum_{t=1}^{T} \psi_t^{\theta_0}(\mathcal{U}_{t,\,-\infty}^{t-S}) \psi_t^{\theta_0}(\mathcal{U}_{t,\,-\infty}^{t-S})^T \right| \leq 2\varepsilon\,, \quad \text{(A.4)}$$

and since ε was arbitrary, we conclude that

$$\lim_{N_{\rm sim}\to\infty} \frac{1}{N_{\rm sim}} \sum_{t=1}^{N_{\rm sim}} \psi_t^{\theta_0}(\tilde{\mathcal{U}}_{t,\,-\infty}) \psi_t^{\theta_0}(\tilde{\mathcal{U}}_{t,\,-\infty})^T = \frac{1}{T} \sum_{t=1}^{T} \psi_t^{\theta_0}(\mathcal{U}_{t,\,-\infty}^{t-S}) \psi_t^{\theta_0}(\mathcal{U}_{t,\,-\infty}^{t-S})^T\,. \quad \text{(A.5)}$$

The last equality in (A.1) follows since P assigns equal probability to T different sequences (corresponding to the possible sequences obtained by shifting $\{u_t\}$). ■