A randomized approach for NARX model identification based on a multivariate Bernoulli distribution

F. Bianchi[a], A. Falsone[a], M. Prandini[a], and L. Piroddi[a*]

[a] *Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Via Ponzio 34/5, 20133 Milano, Italy;*

Email: federico12.bianchi@mail.polimi.it, alessandro.falsone@polimi.it, maria.prandini@polimi.it, luigi.piroddi@polimi.it

[*] *Corresponding author*

(*Received 00 Month 20XX; final version received 00 Month 20XX*)

Research was supported by the European Commission under the project UnCoVerCPS, grant number 643921.

Abstract – The identification of polynomial NARX models is typically performed by incremental model building techniques that progressively select from a candidate set the terms (regressors) to include in the model. The main limitation of these methods stems from the difficulty to correctly assess the importance of each regressor based on the evaluation of partial individual models, which may ultimately lead to erroneous model selections. A more robust assessment of the significance of a specific model term can be obtained by considering ensembles of models, as demonstrated by the recently developed RaMSS algorithm. In that context, the identification task is formulated in a probabilistic fashion and a Bernoulli distribution is employed to represent the probability that a regressor is actually part of the target model. Then, a randomized method is used to sample from the model distribution and gather reliable information to update the distribution, until convergence to a specific model is achieved. The basic version of the RaMSS algorithm employs multiple independent univariate Bernoulli distributions associated to the different candidate model terms, thus overlooking the correlations between different terms, which are typically important in the selection process. In this research endeavor a more complex multivariate Bernoulli distribution is employed, in which the sampling of a given term is conditioned by the sampling of the others. The added complexity inherent in considering the regressor correlation properties is more than compensated by the achievable improvements in terms of accuracy of the model selection process.

Keywords: Nonlinear model identification; Model structure selection; Polynomial NARX models; Randomized algorithms; Multivariate Bernoulli distribution; Prediction error minimization methods.

1. Introduction

The identification of nonlinear dynamical models is a challenging problem which has deserved much attention by researchers in the last few decades (Haber & Unbehauen, 1990), (Sjöberg et al., 1995), (Hong et al., 2008), (S. A. Billings, 2013). In this context, one frequently employed model representation is the nonlinear auto-regressive (moving average) with exogenous inputs (NAR[MA]X) model (Leontaritis & Billings, 1985a), (Leontaritis & Billings, 1985b), which consists in a recursive input-output expression, where the current output is obtained by means of a nonlinear functional expansion of lagged inputs, outputs (and possibly noise) elementary terms. If a polynomial expansion is used for this purpose, the output is a linear function of (nonlinear) monomials of the elementary terms, thus configuring a linear regression problem (or a pseudo-linear one in case the model contains a moving average part). This is particularly convenient in that algorithms of the Least Squares (LS) family can be employed for parameter estimation in a prediction error minimization (PEM) framework.

The hardest problem in NAR[MA]X model identification is however not the estimation of the parameters, but rather the estimation of the correct model structure. Indeed, numerical ill-conditioning and over-parameterization issues typically arise if one attempts to use full polynomial expansions of the elementary terms, since the number of possible model terms increases rapidly with the number, maximum lag and degree of the elementary terms (curse of dimensionality) (Aguirre & Billings, 1995), (Piroddi & Spinelli, 2003). Over-parameterization has a number of unwanted consequences, and typically results in models with poor generalization capabilities. Note also that PEM methods guarantee unbiased parameter estimates, but only if the model structure exactly matches that of the underlying system.

Model structure selection is a combinatorial problem which aims at finding the optimal subset of regressors among a set of candidate ones. Most model selection algorithms consist of a policy to explore the model structure space and compare different model structures in terms of their performance. The naive exhaustive approach of generating all possible structures and comparing their performance (which is, in fact, what one typically does with linear AR[MA]X models) is hardly applicable in practice in the nonlinear case, due to the mentioned curse of dimensionality. Also, statistical indices such as the final prediction error (FPE), the Akaike information criterion (AIC), the Bayesian information criterion (BIC), the minimum description length (MDL), etc., which are used in the linear case to select the model structure are unsuitable in the nonlinear case (Palumbo & Piroddi, 2000), (Chen, Hong, & Harris, 2003). Finally, regularization approaches, such as the LASSO (Least Absolute Shrinkage Selection Operator) can help reducing the candidate regressor set, but are not suitable for exact model selection (Bonin, Seghezza, & Piroddi, 2010).

The forward-regression orthogonal estimator (FROE) (S. Billings, Chen, & Korenberg, 1989) represents a milestone in the research on model structure selection algorithms, and several variants of this method have been proposed in the literature (Mao & Billings, 1999), (Piroddi & Spinelli, 2003), (Li, Peng, & Irwin, 2005), (Wei & Billings, 2008), (Guo, Guo, Billings, & Wei, 2015). The FROE adopts an incremental greedy scheme in which the model is progressively augmented, adding the term that most improves the current model. For this purpose, the regressors are rated by means of the error reduction ratio (ERR) criterion. The FROE also uses a smart scheme based on orthogonal least squares (OLS) to decouple the estimation of the parameters associated to additional terms from that of the parameters related to terms already included in the model. The drawbacks of the FROE have been extensively reviewed in the literature (see, *e.g.*, the discussion in (Piroddi & Spinelli, 2003)). Most of these are related to the PEM setting and the unreliability of the ERR criterion as a measure of the importance of regressors. Indeed, in the PEM framework the parameter estimates are guaranteed to be unbiased only if the model structure is exact (and persistence of excitation conditions hold on the input signal), a condition which obviously does not hold in the early stages of the algorithm. This leads to inaccurate parameter estimation, which in turn may influence the incremental selection process, driving it away from the target model structure. Furthermore, the ERR criterion measures the added portion of explained output variance when a regressor is added to the current model. Such relative importance of a regressor appears to be highly variable with respect to the model structure and may vary significantly during the selection process, so that a regressor considered crucial at the beginning may progressively lose importance as others are added to the model (Piroddi & Spinelli, 2003). These effects can be mitigated using pruning techniques to eliminate wrong terms at the expense of an increased computational complexity (Piroddi & Spinelli, 2013).

Some recent research endeavors have significantly strayed from the classical FROE scheme, by reformulating the model structure selection process in a probabilistic framework and using random sampling methods. In (Baldacchino, Anderson, & Kadirkamanathan, 2013) the problems of model structure selection and parameter estimation have been dealt with together in a unified Bayesian framework. The method operates on a distribution of models tuned by means of a Reversible Jump Markov Chain Monte Carlo (RJMCMC) procedure which is based on three operations: sampling of unselected terms to include in the model, sampling of previously selected terms to remove from the model, and updating of the existing parameters. At each step only one of these operations (randomly chosen) is attempted and its result is accepted or not according to an acceptance ratio. Unfortunately, a long burn-in period is required to achieve proper convergence of the method. Also, as discussed in (Falsone, Piroddi, & Prandini, 2015), the joint identification of both structure and parameters is problematic, since parameters do not vary continuously across structure variations.

In (Falsone et al., 2015) a novel iterative randomized algorithm for model structure selection (RaMSS) has been introduced focusing on NARX models only. This method has some features in common with both RJMCMC and methods based on genetic algorithms (GA) such as (Rodriguez-Vazquez, Fonseca, & Fleming, 2004). As RJMCMC it defines distributions (in this case Bernoulli distributions) over model terms, that represent the probability that the term is present in the "true" model. Also, a sampling procedure is used to update the model distribution. However, contrary to the RJMCMC approach, it does not adopt an incremental/decremental strategy based on the current single model, but gathers the information to update the model distribution from a population of models. More precisely, at each iteration a new population of models is generated according to the Bernoulli distributions. The performances of all the extracted models are then used to update the individual Bernoulli distributions associated to the various terms. The probability of extracting a specific term is increased if, on average, the performance of the models where that term appears is higher than that of the remaining models, and reduced otherwise. As the number of iterations increases, the number of different explored models decreases, and the distribution converges to a limit distribution corresponding to a specific model structure. Note, that differently from GA-based approaches the population of models is not adapted but regenerated at every step (from the adapted Bernoulli distributions).

In the basic version of the RaMSS algorithm the correlation between regressors is not exploited, and the regressors are extracted independently. However, a deeper analysis of the inclusion mechanism of regressors into the model reveals that regressors often tend to enter or disappear jointly in a model. This fact has motivated the present work, which proposes a revised version of the RaMSS that accounts for second order information regarding the relations among regressors in the updating of the Bernoulli distributions. The presented method displays improved performance characteristics, in terms of selection accuracy, number of explored models and number of iterations required to achieve convergence. Using a multivariate Bernoulli distribution to represent the regressor inclusion probabilities has a number of consequences on the structure of the algorithm. For example, a full specification of the joint distribution becomes impractical as the number of variables grows, therefore one needs to consider an approximation of such a distribution. As a consequence the sampling mechanism that generates the models is completely different. Correlated binary variables with specified mean vector and correlation matrix can be simulated using the Conditional Linear Family (CLF) of multivariate binary distributions (Qaqish, 2003). Essentially, regressors are extracted sequentially, the extraction of a given regressor being dependent on the previously extracted ones. The correlation matrix used in the regressor extraction process is constructed using a measure of the tendency of pairs of regressors to appear jointly in the extracted models, weighted by the model performances. Besides the sampling mechanism, the update rules employed to tune the distribution must be integrated to account for second order information.

The rest of the paper is organized as follows. Section 2 briefly describes the nonlinear system identification setting with reference to NARX models and reviews the RaMSS model selection method. Section 3 is devoted to the explanation of the new algorithm. Some simulation examples that well illustrate the features and improvements achievable with the proposed method are presented in Section 4, followed by some conclusions.

2. Preliminaries

2.1. *Nonlinear system identification and the NARX model class*

System identification is about constructing mathematical models from data. We are here concerned, for simplicity, with single input single output systems. We will therefore assume that a series of N input-output pairs $\{(u(t), y(t)), t = 1, \ldots, N\}$ are observed through some experiments performed on the dynamical system of interest, t being a discrete time index. We will further denote as

$$u^t = [u(1), u(2), \cdots, u(t)]$$

and

$$y^t = [y(1), y(2), \cdots, y(t)]$$

the vectors of input and output data up to a certain time index. Then, the objective of the identification process is to look for a relationship between past observations $[u^{t-1}, y^{t-1}]$ and future outputs $y(t)$, in the form of an input-output recursive model:

$$y(t) = g(u^{t-1}, y^{t-1}) + e(t),$$

where the noise signal $e(t)$ accounts for the unavoidable mismatches between model and real data, due to noise and un-modeled phenomena. The available observations can be interpreted as being a finite length realization of a stochastic process, whose characteristics are explicitly represented by $e(t)$. When the model structure is unknown the identification task can be decomposed in two subtasks:

- model structure selection (MSS);
- parameter estimation.

The aim of MSS is to find the form of the nonlinear dependence between data, within a family of functions which is usually parameterized by means of a finite-dimensional parameter vector ϑ:

$$g(u^{t-1}, y^{t-1}; \vartheta). \tag{1}$$

Amongst all the possible parameterizations of the chosen family, we are interested in the one that provides the best approximation of $y(t)$. The parameter estimation task deals with this approximation problem, minimizing w.r.t. ϑ a cost function of the following type:

$$\sum_{t=1}^{N} \left(y(t) - g(u^{t-1}, y^{t-1}; \vartheta)\right)^2 \tag{2}$$

For finite-order systems, the input-output relationship covers a limited time stretch, which is related to the system order, so that $y(t)$ actually depends on a finite-dimensional vector of the most recent past observations:

$$x(t) = [y(t-1), \cdots, y(t-n_y), u(t-1), \cdots, u(t-n_u)]$$

where n_y and n_u are suitable maximum lags, resulting in the following model equation:

$$y(t) = g(x(t);\vartheta) + e(t), \tag{3}$$

which is often referred as nonlinear autoregressive model with exogenous variables (NARX), in view of its input-output recursive structure that is identical to that of linear ARX models. The nonlinear mapping $g(\cdot)$ is often represented by means of a functional expansion:

$$g(x(t);\vartheta) = \varphi(t)^T\vartheta = \sum_{j=1}^{n} \vartheta_j \varphi_j(t), \tag{4}$$

where $\varphi(t) = \varphi(x(t))$ is a mapping that projects the observations onto a finite-dimensional space (the basis functions space). This mapping is particularly convenient in that it makes model (3) linear-in-the-parameters and configures a linear regression problem (all nonlinearities are confined in $\varphi(t)$):

$$y(t) = \varphi(t)^T\vartheta + e(t) \tag{5}$$

For this reason, the elements $\varphi_j(t)$, $j = 1, \ldots, n$ are called *regressors* and $\varphi(t)$ is referred to as the *regressor vector*. Linear-in-the-parameters models are convenient from a computational point of view, since their parameters can be estimated by simple algorithms of the LS family.

There are several families of basis functions that can be used in (4) to approximate any continuous function on a compact domain to a given precision level, provided that the expansion is endowed with sufficient degrees of freedom (*i.e.*, n is large enough). Polynomials, splines, multi-layer perceptron neural networks (NN), radial basis function (RBF), wavelets are examples of such universal approximating functions. A popular choice in nonlinear identification is the polynomial functional expansion which is a linear combination of all monomials of $x(t)$ up to a given order. This functional expansion extends gracefully from linear models and typically allows an easier model interpretation. Indeed, the terms in the model are often associated to physical aspects of the system and so the parameters can be interpreted as importance factors for the relative physical phenomena. Furthermore, nonlinear combinations of physical variables reveal the existence of specific nonlinear dependencies. Other functional expansions do not share this feature. For example, artificial NNs are very powerful estimators, but do not provide transparent models.

On the other hand, polynomial expansions suffer from the well known curse of dimensionality, in that the number of terms grows rapidly with the number of elementary arguments $(n_u + n_y)$ and the degree of the polynomial. However, it is common experience that polynomial models with few terms can provide highly accurate and robust models. It is therefore crucial to identify and select the essential terms of a model, which motivates the interest in MSS algorithms.

2.2. *MSS: the RaMSS method*

The MSS problem consists in finding the best subset of regressors from a set of candidate regressors, such that the corresponding model has maximum accuracy. In the absence of *a priori* information the candidate regressor set is constructed as the set of terms of a full polynomial expansion of a given low degree (typically, 2 or 3) of the elements of $x(t)$. The larger the candidate set, the greater are the model flexibility and the optimal model accuracy, but also the greater is the difficulty of the MSS task. It is trivial to convince oneself that an exhaustive comparison of all possible model structures is not generally feasible in practice.

An iterative randomized algorithm for MSS (briefly, RaMSS), has been introduced in (Falsone et al., 2015). The method exploits a probabilistic framework to tackle the model identification procedure. A Bernoulli random variable $\rho_j \sim Be(\mu_j)$ is associated to each regressor φ_j, and its success probability μ_j (named Regressor Inclusion Probability, or RIP) represents the belief that the regressor belongs to the true model. Accordingly, the random variable $m = (\rho_1, \dots, \rho_n)$ represents a random model structure. The RIPs naturally induce a probability distribution over the model structure space. For simplicity, the random variables $\rho_1, \dots, \rho_n$ are assumed to be independent in (Falsone et al., 2015).

For each model structure m one can compute a performance index

$$J(m) = e^{-K \cdot \mathrm{MSPE}(m)}, \tag{6}$$

K being a scaling parameter that can be used to magnify small performance differences, while MSPE is the mean square prediction error,

$$\mathrm{MSPE}(m) = \frac{1}{N} \sum_{t=1}^{N} \left(y(t) - \hat{y}_m(t)\right)^2,$$

where $\hat{y}_m(t)$ denotes the one-step-ahead prediction of $y(t)$ computed according to the model structure represented by m. The reader should note that in the considered setting both MSPE(m) and $J(m)$ are random variables.

The RaMSS algorithm exploits this probabilistic framework to progressively refine the value of the RIPs based on the average performance of the models, given the current distribution. Specifically, at the $(k+1)$-th iteration the jth RIP is updated according to:

$$\mu_j(k+1) = \mu_j(k) + \gamma \cdot \left(E_{P_j(k)}[J(m)] - E_{P_{-j}(k)}[J(m)]\right), \tag{7}$$

where γ is a step size, and $E_{P_j(k)}$ and $E_{P_{-j}(k)}$ denote the expected values computed according to the probability distributions $P_j(k)$ and $P_{-j}(k)$ on the model structure space at the kth iteration, which are respectively conditioned to the presence or absence of the jth regressor. Before computing $J(m)$, the relevance of each regressor in m is tested using a t-test. All statistically non-significant regressors are not considered in the computation of $J(m)$. The probability distributions $P_j(k)$ and $P_{-j}(k)$ are computed accordingly (see (Falsone et al., 2015) for a detailed explanation).

The rationale behind (7) is that the extraction of a regressor will be encouraged or discouraged (by modifying accordingly the mean of the associated Bernoulli distribution), depending on the difference between the average performance of the models which contains the regressor, and the one of those models which do not.

Since an exact computation of the expected values in (7) would lead to an exhaustive enumeration of all model structures, the RaMSS algorithm, at each iteration, approximates the right hand side of (7) using samples. Specifically, at each iteration, N_p model structures are extracted. Each model structure m is obtained by performing n independent Bernoulli trials (one for each regressor), whereby the regressor φ_j is included in the model if and only if the corresponding Bernoulli trial is successful (*i.e.*, $\rho_j = 1$). Once the N_p model structures have been generated, their parameters are estimated by means of an LS procedure. Subsequently, for each model all statistically non-significant regressors are removed via a t-test, and the parameters re-estimated after the removal. Then, the performance index $J(\cdot)$ is computed for all extracted models, the RIPs are updated according to

(7) (where sample estimates are used in place of the expected values), and the algorithm proceeds to the next iteration.

Note that (7) does not guarantee that $\mu_j(k+1) \in [0,1]$. To this end a thresholding is applied to ensure the well-definiteness of the Bernoulli distributions. The algorithm ends when a stopping criterion is met. This can either be associated to a maximum number of iterations, or a practical convergence of the RIPs. The practical convergence is achieved when the relative difference between the RIPs calculated at subsequent iterations are lower than a given threshold.

Finally, let us remark that the RaMSS is based on an independence assumption between regressors in the model extraction phase, whereby each regressor is extracted independently from the others. The present work studies the benefits of relaxing such hypothesis.

2.3. *An illustrative example*

Consider the following system:

$$\begin{aligned} y(t) = &- 1.7y(t-1) - 0.8y(t-2) + \\ &+ u(t-1) + 0.8u(t-2) + e(t), \end{aligned}$$

where $e(t)$ is a white Gaussian noise. Now, construct a candidate regressor set with all the monomials of degree less or equal 2 obtained from $x(t) = [y(t-1), y(t-2), u(t-1), u(t-2)]$. Denote as:

$$\begin{aligned} R = (\varphi_1, \varphi_2, \cdots, \varphi_n) = (1, y(t-1), y(t-2), u(t-1), \\ u(t-2), y(t-1)^2, y(t-1)y(t-2), y(t-1)u(t-1), \cdots) \end{aligned}$$

the ordered set of all possible regressors. Notice that the correct regressors are φ_2, φ_3, φ_4, and φ_5.

Let us examine a run of the algorithm that resulted in the selection of the exact model structure. Specifically, consider the second iteration. For the sake of simplicity, focus only on 5 models, structured as synthetically represented by the following model-regressor matrix (a 1 in cell (r,c) indicates that model m_r includes regressor φ_c):

$$\mathrm{W} = \begin{array}{c} \\ m_1 \\ m_2 \\ m_3 \\ m_4 \\ m_5 \end{array} \begin{array}{c} \begin{array}{ccccccc} \varphi_1 & \varphi_2 & \varphi_3 & \varphi_4 & \varphi_5 & \varphi_6 & \cdots \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

The fourth model includes two regressors out of the correct ones, while all the others include only one. In addition, m_1 and m_2 are subparts of m_4, therefore the latter has higher performance than the others. Indeed, the performance vector J was as follows:

$$J = (0.0389, 0.2143, 0.0472, 0.8125, 0.0335)^T$$

Consider regressors φ_2 and φ_4. The models that include only one of these two regressors, specifically m_1 and m_2, have lower performance than m_4 which includes both of them. This means that taking these two regressors together is more convenient than considering them separately. In other words, model m_4 is highlighting a relation between the two regressors. To exploit this relation, one can encourage the extraction of models containing both φ_2 and φ_4, using second order information (*e.g.*, the correlation) in the extraction mechanism. For a better understanding of this idea, consider the

RIP increments for these two regressors (see Table 1), as would result from the application of Equation (7) to the 5 considered models. It is clear how these two regressors will benefit from the presence of m_4 in the extracted population. In particular, if model m_4 had not been extracted, the RIP of φ_4 would be much lower.

Table 1. Update factor values for the RIPs of the example.

Regressor	RIP increments	
	including m_4	without m_4
φ_2	0.4735	0.1744
φ_3	-0.2276	-0.048
φ_4	0.3274	-0.0594
φ_5	-0.2447	-0.0666

3. Including second order information in the RaMSS scheme

We now propose a variant of the RaMSS algorithm where second order information concerning the correlation between regressors is introduced in the MSS process. In this new scenario, a multivariate Bernoulli distribution is associated to the set of candidate regressors, which defines the probability of extraction of the model terms. Unlike the classic RaMSS, the regressors are not independent but they are correlated with a given mean vector and a covariance matrix. The proposed variant will be referred to as *Correlated* - RaMSS, briefly C-RaMSS.

3.1. *CLF: The Conditional Linear Family method*

Correlated random binary variables can be described by means of a multivariate (or joint) Bernoulli distribution. If the multivariate distribution is known, it is straightforward to simulate the associated correlated random binary variables by means of sampling methods (Devroye, 1986). However, a full specification of an n-variate distribution is impractical because it would require to specify the probabilities of all the possible combinations of n binary variables, thus requiring that 2^n parameters be defined. In practice, one can approximate the multivariate distribution by specifying just the first and second moments, namely the mean vector μ and the covariance matrix V. In the literature, several methods have been proposed to simulate correlated random binary variables with specified mean vector and covariance matrix (Lee, 1997), (Lunn & Davies, 1998), (Oman & Zucker, 2001). The previously mentioned methods do not simultaneously allow the distribution to have unequal means and arbitrary sign of the correlation coefficients, which are instead needed in our framework. For this reason, we here rely on the method introduced in (Qaqish, 2003) to simulate the correlated binary variables.

Let $Y = [Y_1, \dots, Y_n]^T$ denote an $n \times 1$ vector of Bernoulli random variables, with mean $E(Y) = [\mu_1, \dots, \mu_n]^T = \mu$, and covariance matrix $\mathrm{cov}(Y) = V$. According to (Qaqish, 2003), among all possible multivariate binary distributions with the same mean μ and covariance V, the one known as *conditional linear family* (CLF), is fully characterized by μ and V and satisfies the following property:

$$\Pr(Y_i = 1 | Y_1 = y_1, \dots, Y_{i-1} = y_{i-1}) = \mu_i + \sum_{j=1}^{i-1} b_{ij}(y_j - \mu_j) \tag{8}$$

where $b_i = G_i^{-1} s_i$, with $G_i = \mathrm{cov}([Y_1, \dots, Y_{i-1}]^T)$, and $s_i = [V_{i1}, \dots, V_{i,i-1}]^T$.

Note that this formulation poses some restrictions on μ and V in that the quantity in (8) is a probability and therefore belongs to $[0, 1]$.

Simulating Y from a multivariate binary distribution with the above structure simply amounts to simulating first Y_1 as a Bernoulli random variable with mean μ_1, and then, for $i = 2, \dots, n$, simulating Y_i as a Bernoulli random variable with conditional probability given by Equation (8).

Notice that it is sometimes convenient to operate in terms of the matrix of correlation coefficients R. In this respect, specifying (μ, V) is equivalent to specifying (μ, R), since for Bernoulli variables $\mathrm{var}(Y_i) = V_{ii} = \mu_i(1-\mu_i)$, and therefore:

$$V_{ij} = R_{ij}\sqrt{\mu_i(1-\mu_i)}\,\sqrt{\mu_j(1-\mu_j)}. \tag{9}$$

3.2. *Update rule for the correlation matrix*

As discussed previously, the simulation algorithm requires the mean vector and the correlation (or covariance) matrix. While the mean vector and its update rule are already defined in the RaMSS (see Eq. (7)), the method lacks a corresponding rule for the correlation matrix. The idea is to iteratively construct this matrix starting from an identity matrix. The updating at the $(k+1)$-th iteration takes the form:

$$R_{ij}^{(k+1)} = \nu \cdot R_{ij}^{(k)} + (1-\nu)\cdot \Delta R_{ij}^{(k)} \tag{10}$$

where ν is a learning factor and $\Delta R_{ij}^{(k)}$ is the update factor for R_{ij}. Informally, we can characterize the correlation between the inclusion probabilities of two regressors as the property that assesses when it is convenient to extract them together in the model. Following this rationale, we define the update factor as a measure of the tendency of regressors φ_i and φ_j to appear jointly in the extracted models, weighted with the performances of the models extracted at the k-th iteration. In other words, when two regressors always appear together in good models we want to increase their correlation, so that the probability of extracting models with both of them is higher.

An easy way to characterize the tendency of two regressors to appear jointly is to represent the model structures of the population of models generated from the current distribution using a vector space model (VSM) defined on regressors and models. More specifically, the VSM can be described by means of a $N_p \times n$ binary model-regressor matrix W, such that:

$$W_{ij} = \begin{cases} 1, & \text{if regressor } \varphi_j \text{ appears in model } m_i \\ 0, & \text{otherwise} \end{cases}$$

where n is the number of regressors. In W, the columns associated to regressors always appearing together in the extracted models are equal. We can therefore measure the tendency that two regressors appear jointly in the extracted models by evaluating the distance between the corresponding column vectors. A widely used distance measure in a VSM is the cosine of the angle between vectors. Specifically, given two regressors φ_i and φ_j the cosine distance is defined as:

$$C_{ij} = \frac{\langle W_{\cdot,i}, W_{\cdot,j}\rangle}{\|W_{\cdot,i}\|\|W_{\cdot,j}\|} = \frac{\sum_l W_{l,i}W_{l,j}}{\sqrt{\sum_l W_{l,i}^2}\sqrt{\sum_l W_{l,j}^2}}. \tag{11}$$

where $W_{\cdot,l}$ denotes the lth column of W. Notice that $0 \le C_{ij} \le 1$.

This kind of similarity measure is widely used in the Information Retrieval field to assess the relevance of a document to a given query (Berry, Drmac, & Jessup, 2006). The cosine distance on its own is a measure of the tendency of regressors to appear jointly in the extracted models, but it does not take into account their performances. The added feature here is to weigh such measure according to the performances of the extracted models. Let

$$\tilde{W} = \mathrm{diag}(\sqrt{J_1}, \ldots, \sqrt{J_{N_p}})\, W \tag{12}$$

be the weighted similarity matrix, J_i being the performance associated to model m_i. Accordingly, define the weighted cosine distance $\tilde{C}$ as:

$$\tilde{C}_{ij} = \frac{\langle \tilde{W}_{\cdot,i}, \tilde{W}_{\cdot,j} \rangle}{\|\tilde{W}_{\cdot,i}\| \|\tilde{W}_{\cdot,j}\|}, \tag{13}$$

The weighted cosine distance $\tilde{C}$ is low for regressors never (or rarely) appearing jointly, but also for regressors that appear together in scarcely accurate models. Conversely, high values of $\tilde{C}$ are obtained for pairs of regressors that appear jointly in accurate models.

The update rule in Equation (10) can be thus expressed as:

$$R_{ij}(k+1) = \nu \cdot R_{ij}(k) + (1-\nu) \cdot \tilde{C}_{ij}. \tag{14}$$

By construction, matrix R is symmetric, positive semi-definite and with unit diagonal, so it is a valid matrix of correlation coefficients.

3.3. *Cont'd illustrative example*

To appreciate the impact of (14) into the selection algorithm, let us continue with the illustrative example introduced in Section 2.3. The final idea emerging from that analysis was to encourage the extraction of models containing both regressors φ_2 and φ_4 since these models had higher performance w.r.t. those containing the two regressors separately.

Now, let us compute how would the extraction probabilities of the five models be affected by the application of only rule (7), as in the plain RaMSS, or by the same rule combined with the second order update equation (14) (see Table 2 for a comparison).

Table 2. Extraction probabilities for the models in the example.

Model	Extraction probability	
	1st order	2nd order
m_1	0.0616	0.0008
m_2	0.1031	0.0733
m_3	0.0365	0.0082
m_4	0.0157	0.0514
m_5	0.0430	0.0227

Apparently, compared to the plain RaMSS, the use of second order information in the regressor distribution discourages the extraction of models containing either φ_2 or φ_4 *separately* (*i.e.* m_1 and m_2), while the extraction of m_4 is encouraged. Notice that, based on the use of only first order information, the algorithm would not recognize m_4 as a promising model. On the other hand, with the envisaged correction the probability of extracting m_4 is significantly increased.

3.4. *The C-RaMSS algorithm*

We here summarize the C-RaMSS algorithm and discuss its implementation. A Bernoulli random variable ρ_i is associated to each regressor φ_i, and its success rate μ_i, describes the probability of extracting that regressor in a model. Regressors are initially assumed to be uncorrelated, in the absence of *a priori* information regarding their pairwise performance. Then, at all iterations, the conditional probability of extraction of each regressor is updated according to Equation (8). The RIP vector and the correlation matrix R are used by the CLF method as reference statistics for the new population of extracted models. The n regressors are initially uniformly distributed ($\rho_i = \frac{1}{n}, \forall i$) while the correlation matrix R is initialized as an identity matrix. At each iteration a population of N_p models is extracted. If the model turns out to be empty, it is discarded and a new model is generated. Once the models have been extracted, parameter estimation is carried out with LS. A statistical t-test is performed in order to remove redundant regressors and then the parameters are re-estimated. The significance confidence level α for the statistical test is a tuning parameter. The performance of each model is evaluated through the index defined in Equation (6), where factor K is a tuning parameter which determines the sensitivity of the performance index.

Finally, the correlation matrix and the RIPs are updated according to the rules in Equations (7) and (14). The learning factor ν is a tuning parameter. Since Equation (7) does not automatically ensure that the parameters μ_j remain in the $[0, 1]$ interval, suitable saturation thresholds $\mu_{\min}$ and $\mu_{\max}$ are introduced for this purpose. Algorithms 1 and 2 represent the two main contributions introduced in the C-RaMSS with respect to RaMSS, specifically the CLF simulation algorithm and the computation of the weighted cosine distance matrix needed to update the correlation values. Algorithm 3 summarizes the whole model identification procedure.

Algorithm 1 CLF simulation

```
Require: $\mu$, $V$, $\Re = \{\varphi_j(t), j = 1, \cdots, n\}$,
Ensure: $\{(\psi_p(t)), p = 1, \cdots, N_p\}$, $W$
1:  for $p = 1$ to $N_p$ do                                   ▷ Generate first regressor
2:      Extract $W_{p,1}$ from $Be(\mu_1)$;
3:      $\psi_p(t) \leftarrow [\,]$;
4:      if $W_{p,1} = 1$ then                                 ▷ Add regressor
5:          $\psi_p(t) \leftarrow [\psi_p(t)^T \ \varphi_1(t)]^T$;
6:      end if
7:  end for
8:  for $i = 2$ to $n$ do                                     ▷ Generate regressors
9:      $G_i \leftarrow V[1 : i-1, 1 : i-1]$;
10:     $s \leftarrow V[i, 1 : i-1]^T$;
11:     $b_i \leftarrow G_i^{-1} \cdot s$;                    ▷ Generate linear coefficients
12:     for $p = 1$ to $N_p$ do
13:         $\lambda_i \leftarrow \mu_i + \sum_{j=1}^{i-1} b_{ij}(W_{p,j} - \mu_j)$;   ▷ Compute conditional probabilities
14:         Extract $W_{p,i}$ from $Be(\lambda_i)$;
15:         if $W_{p,i} = 1$ then                             ▷ Add regressor
16:             $\psi_p(t) \leftarrow [\psi_p(t)^T \ \varphi_i(t)]^T$;
17:         end if
18:     end for
19: end for
```

Algorithm 2 Computation of weighted cosine distance

```
Require: ${(J_p), p = 1, \cdots, N_p}, W$
Ensure: $\tilde{C}$
1: $\tilde{W} \leftarrow diag(\sqrt{J_1}, \ldots, \sqrt{J_{N_p}}) \cdot W$;        ▷ Weighted regressor-model matrix
2: $M \leftarrow \tilde{W} \cdot \tilde{W}^T$;
3: for $i = 1$ to $n$ do
4:     for $j = 1$ to $n$ do
5:         $\tilde{C}_{ij} \leftarrow \frac{M_{ij}}{\|\tilde{w}_i\| \|\tilde{w}_j\|}$;
6:     end for
7: end for
```

Algorithm 3 C-RaMSS algorithm

```
Require: $\{(u(t), y(t)), t = 1, \cdots, N\}$, $\Re = \{\varphi_j(t), j = 1, \cdots, n\}$, $N_p$, $K$, $\nu$, $\alpha$, $\mu$, $\mu_{\min}$, $\mu_{\max}$, $\varepsilon$
Ensure: $\mu, R$
1: $\mu \leftarrow \frac{1}{n} \cdot \mathbf{1}_{n \times 1}$;
2: $R \leftarrow I_n$;
3: repeat
4:     Compute the covariance matrix $V$ with Equation (9);
5:     Generate regressors $\{(\psi_p(t)), p = 1, \cdots, N_p\}$ and matrix $W$ (Algorithm 1);
6:     for $p = 1$ to $N_p$ do
7:         $S \leftarrow \left(\sum_{t=1}^{N} \psi_p(t)\psi_p(t)^T\right)^{-1}$;
8:         $\hat{\vartheta} \leftarrow S \sum_{t=1}^{N} \psi_p(t)y(t)$;        ▷ Estimation
9:         $\hat{\sigma}_e^2 \leftarrow \frac{1}{N-\tau_p} \sum_{t=1}^{N} (y(t) - \psi_p(t)^T \hat{\vartheta})$;
10:        $\tau_p = \sum_{h=1}^{n} W_{p,h}$
11:        for $h = 1$ to $\tau_p$ do        ▷ T-test
12:            $\sigma_h^2 \leftarrow \sigma_e^2 \cdot S_{h,h}$;
13:            if $|\hat{\vartheta}_h| \le \hat{\sigma}_h \cdot t_{\alpha, N-\tau_p}$ then
14:                Remove regressor $\psi_{p,h}(t)$ from $\psi_p(t)$;
15:                $W_{p,h} \leftarrow 0$;
16:            end if
17:        end for
18:        $\hat{\vartheta} \leftarrow \left(\sum_{t=1}^{N} \psi_p(t)\psi_p(t)^T\right)^{-1} \sum_{t=1}^{N} \psi_p(t)y(t)$;        ▷ Re-estimation
19:        for $t = 1$ to $N$ do
20:            $\hat{y}(t) = \psi_p(t)^T \hat{\vartheta}$;
21:        end for
22:        $\text{MSPE} = \frac{1}{N} \sum_{t=1}^{N} (y(t) - \hat{y}(t))^2$
23:        $J_p \leftarrow e^{-K \cdot \text{MSPE}_p}$;        ▷ Model evaluation
24:    end for
25:    for $j = 1$ to $n$ do        ▷ Update RIPs
26:        $J^+ \leftarrow 0$; $n^+ \leftarrow 0$; $J^- \leftarrow 0$; $n^- \leftarrow 0$;
27:        for $p = 1$ to $N_p$ do
28:            if $\varphi_j(t) \in \psi_p(t)$ then
29:                $J^+ \leftarrow J^+ + J_p$; $n^+ \leftarrow n^+ + 1$;
30:            else
31:                $J^- \leftarrow J^- + J_p$; $n^- \leftarrow n^- + 1$;
32:            end if
33:        end for
34:        $\mu_j \leftarrow \mu_j + \gamma \left(\frac{J^+}{\max(n^+,1)} - \frac{J^-}{\max(n^-,1)}\right)$;
35:        $\mu_j \leftarrow \max(\min(\mu_j, \mu_{\max}), \mu_{\min})$;
36:    end for
37:    Compute $\tilde{C}$ matrix (Algorithm 2);
38:    $R \leftarrow \nu R + (1-\nu)\tilde{C}$;        ▷ Update matrix R
39: until $\max_{j=1,\ldots,n} (\max\{\mu_{\max} - \mu_j, \mu_j - \mu_{\min}\}) \le \varepsilon$        ▷ Stopping criterion
```

4. Simulation examples

In this section several simulation examples are discussed to show the effectiveness of the C-RaMSS algorithm. First, a typical run of the algorithm is shown to illustrate its behavior (Section 4.1). Then, various non ideal identification problems are discussed in Sections 4.2-4.4, *i.e.* cases where the system generating the data does not belong to the model family (*e.g.*, because of a non trivial noise model or because of the complexity of the nonlinearity of the input-output model) or where the data are oversampled (which is simulated by exciting the system with a low frequency input signal). The effects of an increase in the noise levels are studied in Section 4.5. Subsection 4.6 is devoted to a computational analysis of the algorithm for increasing number of candidate regressors. Finally, a comparative analysis w.r.t. the RaMSS, FROE and iOFR algorithms has been carried out on various different identification problems to assess the improved performance of the C-RaMSS method (see Section 4.7). Many of the systems considered in the examples are taken from the literature (Wei & Billings, 2008), (Baldacchino et al., 2013), (Piroddi & Spinelli, 2003), to facilitate comparisons.

In all tests, 300 runs of the algorithm have been carried out for the same system and the results have been aggregated. The data-set used in the identification process is composed of 1000 randomly generated input/output pairs. Seventy percent of the data is used for the training of the model while the remaining 30% is used for the validation of the obtained model. Concerning the RaMSS and C-RaMSS methods, the following statistics have been studied:

number of iterations: average number of iterations required to achieve convergence,
elapsed time: average time required to obtain the final model,
correctness: percentage of exact model selections,
explored models: average number of (distinct) models explored by the algorithm,
maximum AMS: average of the maximum AMS obtained for each complete run of the algorithm,
final AMS: average AMS at the last iteration,

where AMS (average model size) denotes the average size of the extracted models at a given iteration. As for the FROE and iOFR algorithms, we considered in particular the **model size**, the **number of correct/wrong identified terms**, as well as the **correctness** in the model selection.

All tests have been performed in a MATLAB 2014b environment (MATLAB, 2014), on an HP ProBook 650 G1 CORE i7-4702MQ CPU @2.20 GHz with 8GB of RAM.

4.1. *Example 1: A typical C-RaMSS run*

A typical run of the C-RaMSS algorithm is illustrated with reference to the following NARX system (Baldacchino et al., 2013)

$$S_1 : y(t) = 0.7y(t-1)u(t-1) - 0.5y(t-2) - 0.7y(t-2)u(t-2)^2 + 0.6u(t-2)^2 + e(t)$$

with $u(t) \sim WUN(-1,1)$ and $e(t) \sim WGN(0, 0.04)$, where $WUN(a,b)$ is a White Uniform Noise defined in the interval $[a,b]$, while $WGN(\eta,\sigma^2)$ is a White Gaussian Noise with mean η and standard deviation σ. The candidate regressor set contains all the monomials obtained as combinations of the lagged input and output signals with maximum lags $n_u = n_y = 4$ and maximum degree equal to 3, for a total of $n = 165$ regressors. The number of models generated at each iteration is set to $N_p = 200$ and the initial RIPs are equal to $\mu_j = \frac{1}{n}$, $j = 1,\dots,n$. By doing so, all the regressors have initially the same probability of being extracted and the models extracted at early iterations have typically a small size. The tuning parameter K in the performance index J is set to 1 and the learning factor ν is set to 0.1.

Figures 1 and 2 illustrate respectively the evolution of the RIPs and that of the AMS. The RIPs associated to the correct regressors are gradually increasing until they reach 1. On the other hand the RIPs associated to the other regressors, after a transient, tend to 0. The AMS is (almost) monotonically increasing to the correct value. This emphasizes the incremental nature of the selection procedure which starts from very small models and progressively increases the (average) size of the explored models.

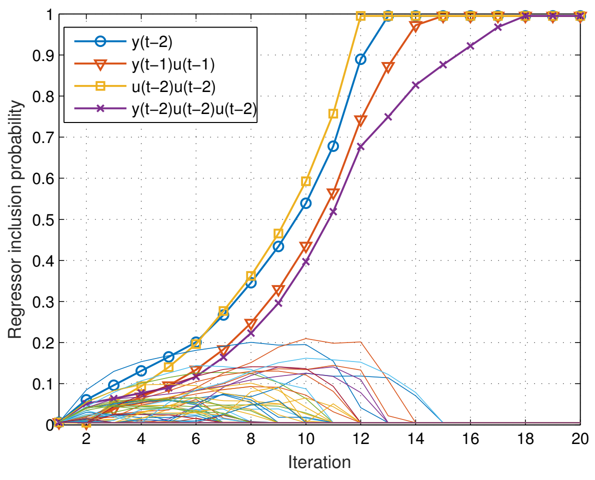

Figure 1. Example 1: A typical evolution of the RIPs (the true terms are emphasized).

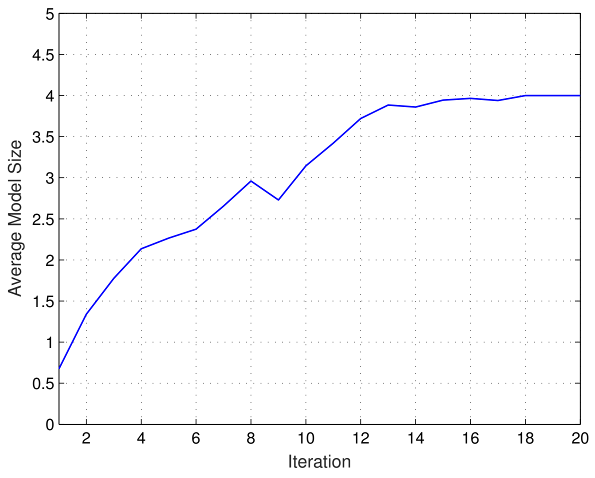

Figure 2. Example 1: A typical evolution of the AMS.

Figure 3 displays the evolution of the correlation coefficients associated to pairs of correct regressors. The correlation coefficients, not reported in Figure 3, associated to pairs of spurious regressors or pairs of regressors that do not frequently occur jointly tend to zero.

Finally, it is interesting to study how the regressor distribution evolves in time compared with the RaMSS algorithm. Figure 4 shows the distribution of the RIPs at different stages of the RaMSS and C-RaMSS algorithms. For graphical reasons the distributions are sorted in descending RIP order, and the correct regressors are emphasized. Apparently, the C-RaMSS is much faster in recognizing the correct regressors and at the same time rejecting the other ones.

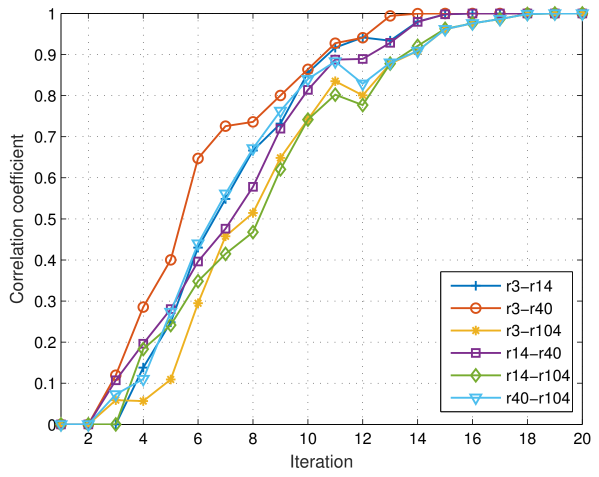

Figure 3. Example 1: Evolution of the correlation associated to the true regressors ($r_3(t) = y(t-2)$, $r_{14}(t) = y(t-1)u(t-1)$, $r_{40}(t) = u(t-2)^2$, and $r_{104}(t) = y(t-2)u(t-2)^2$).

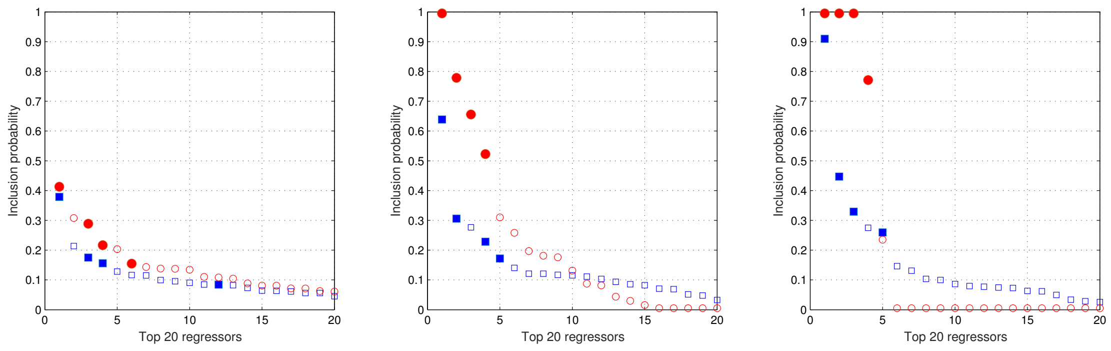

Figure 4. Example 1: Evolution of the RIP distribution with the RaMSS (squares) and the C-RaMSS (circles): early (left), intermediate (middle), and advanced (right) stages. Distributions are sorted in descending order and the correct regressors are emphasized.

4.2. *Example 2: A system with correlated noise*

The aim of this analysis is to assess how the C-RaMSS fares, compared to the RaMSS, in the identification of the *process* model structure when the noise model is not limited to an additive white noise term (the identification of the noise model is not addressed). To this purpose, consider the following NARMAX system (Baldacchino et al., 2013):

$$S_2 : y(t) = 0.7y(t-1)u(t-1) - 0.5y(t-2) - 0.7y(t-2)u(t-2)^2 + 0.6u(t-2)^2 + \\ 0.2e(t-1) - 0.3u(t-1)e(t-2) + e(t)$$

with $u(t) \sim WUN(-1, 1)$, $e(t) \sim WGN(0, 0.02)$. The candidate regressor set includes all monomials of degree up to $n_d = 3$, and with maximum lags $n_u = n_y = 4$. The parameters $K = 1$, $\alpha = 0.997$ have been used for both algorithms and a learning factor $\nu = 0.1$ has been used for the C-RaMSS. Both algorithms have been tested for different sizes of the population of extracted models.

Some indicators regarding the performance of the two algorithms are reported in Table 3. The absolute correctness shows the effectiveness of the probabilistic framework behind both RaMSS and C-RaMSS methods. However, the C-RaMSS requires much fewer iterations to achieve convergence, as well as fewer explored models. The maximum and final AMS are not significantly affected by the new method. The increasing elapsed time is due to the extra computation required to generate models according to the CLF method.

For completeness sake, the FROE algorithm has been tested as well on the NARMAX systems using a 1% threshold for regressor inclusion (at each iteration the most improving regressor is added, provided it improves the MSPE by at least 1% of the output variance). A wrong constant term is added by the FROE and the regressor $y(t-2)u(t-2)^2$ is not included. The constant term can be removed *a posteriori* by applying a statistical t-test as in the C-RaMSS. Finally, the iOFR was run starting from the suboptimal solution identified by the FROE (after the t-test), and it was able to identify the correct process model.

Table 3. Example 2: Comparison between C-RaMSS and RaMSS on the NARMAX system S_2.

	RaMSS		C-RaMSS	
	$N_p = 100$	$N_p = 200$	$N_p = 100$	$N_p = 200$
Correct selection	100%	100%	100%	100%
# of Iterations	31.82	31.73	21.44	19.42
Elapsed Time [sec]	1.5688	2.7231	9.5869	12.292
Maximum AMS	3.9924	3.9893	4.0898	4.0121
Final AMS	3.9816	3.9794	3.9896	3.9917
# of Explored Models	592.1	974.3	364.24	628.32

4.3. *Example 3: A system with a non-polynomial nonlinearity*

Consider now the system

$$S_3 : y(t) = \exp(-y(t-1)) - 1 + 0.3u(t-1) + e(t),$$

with $u(t) \sim WUN(-1,1)$ and $e(t) \sim WGN(0,0.01)$. Though the system belongs to the NARX family it cannot be exactly represented by a finite polynomial expansion. Still, it is important to assess the performance of the proposed algorithm when the system does not belong to the model family (as occurs in normal practice). The candidate regressor set and the algorithm settings are identical to the previous example.

Figure 5 shows the prediction quality of the model identified with C-RaMSS on some validation data. To further assess the C-RaMSS on this test case, we provide in Table 4 some aggregate results in terms of the MSPE achieved with various methods. The "t" superscript associated to the FROE and iOFR algorithms indicates that the selected models are *a posteriori* subjected to a t-test to remove statistically non-significant terms. While the two incremental approaches settle for a 4-term model (or a 3-term one after the t-test), both randomized approaches select 5 terms and obtain a 30% improvement in the prediction quality over the validation data. Clearly, the FROE and iOFR have got stuck in some local minimum, from which no significant improvement is possible.

Table 4. Example 3: Comparative prediction performance analysis on validation data for system S_3.

Method	MSPE	Model size	Identified terms
FROE	0.0124	4	$1, y(t-1), u(t-1), y(t-1)^2$
FROE^t	0.0128	3	$y(t-1), u(t-1), y(t-1)^2$
iOFR	0.0124	4	$1, y(t-1), u(t-1), y(t-1)^2$
iOFR^t	0.0128	3	$y(t-1), u(t-1), y(t-1)^2$
RaMSS	0.0088	5	$y(t-1), u(t-1), y(t-1)^2, y(t-1)^2y(t-2), y(t-1)^2u(t-2)$
C-RaMSS	0.0084	5	$y(t-1), u(t-1), y(t-1)^2, y(t-1)^3, y(t-1)^2y(t-2)$

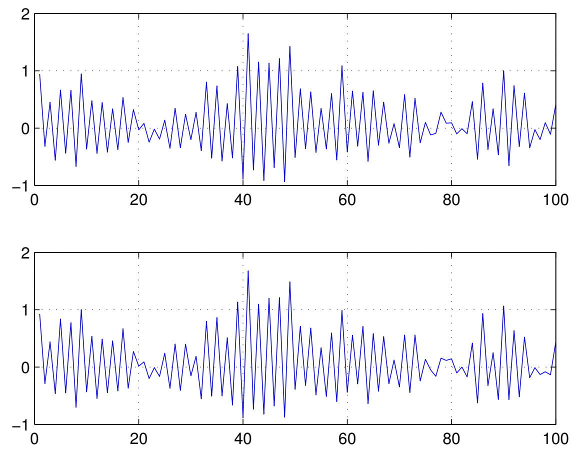

Figure 5. Example 3: Performance of the model identified with C-RaMSS on validation data: output data (top) and one-step ahead predictions (bottom).

4.4. *Example 4: Low frequency input signal*

The MSS process is generally sensitive to the excitation characteristics of the input, and particularly to slowly varying input signals (Piroddi & Spinelli, 2003), essentially because subsequent output samples turn out to be very similar (*e.g.*, $y(t) \sim y(t-1)$). This is also what typically happens when data are oversampled, a condition that is often encountered in normal practice. The analysis in this section aims at assessing how the C-RaMSS algorithm behaves with respect to the RaMSS algorithm when the input signal has a low frequency content, obtained by low-pass filtering a white noise signal. An AR(2) process was employed for this purpose, placing the two poles according to the desired filter cut-off frequency. Indeed, as the poles move from the origin to the border of the unit circle, the filtering effect is greater and thus the input signal has a lower frequency content.

Both the RaMSS ($N_p = 200$) and the C-RaMSS ($N_p = 200, \nu = 0.1$) have been tested on system S_1, this time excited with several slow input signals, with varying filter cut-off frequency. As expected, both methods experience a small performance drop at some point when the input signal is too slow. However, satisfactory performance (greater than 0.98) can be achieved with both methods on almost all the frequency range (see Figure 6), which is a remarkable result in itself.

The most interesting fact regards the size of the selected model structure. In Figure 7 the final AMS is compared for both methods as a function of the filter cut-off frequency. Apparently, for low frequencies the C-RaMSS tends to identify more compact models than the RaMSS. Furthermore, comparing Figures 6 and 7, this smaller model size is not costly in terms of performance. Indeed, the performance of the C-RaMSS is generally slightly better (see Figure 6). In summary, the second order information employed by the C-RaMSS to account for the relations between regressors focuses the exploration of the solution space leading to more compact models than the RaMSS without loss of efficiency. This is very important, since compact models are typically more robust and more easily interpretable.

4.5. *Example 5: Increasing noise levels*

This analysis aims to prove the robustness of the proposed model as the noise level increases. We focus again on system S_1 for which, with the noise level assumed in Example 1, both RaMSS and C-RaMSS are always able to identify the correct structure. As expected, the performance of both methods in terms of J (not represented her for brevity) decreases, almost in an equivalent way, as the noise variance increases. However, Figure 8 reveals that the correctness of the C-RaMSS is

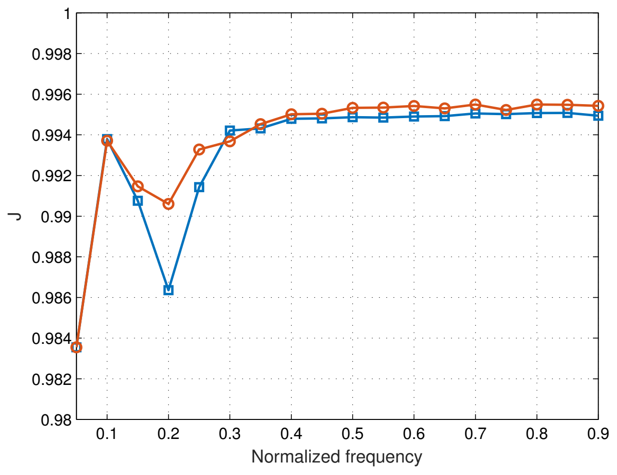

Figure 6. Example 4: Average performance of the models identified with RaMSS (squares) and C-RaMSS (circles) as a function of the input angular frequency.

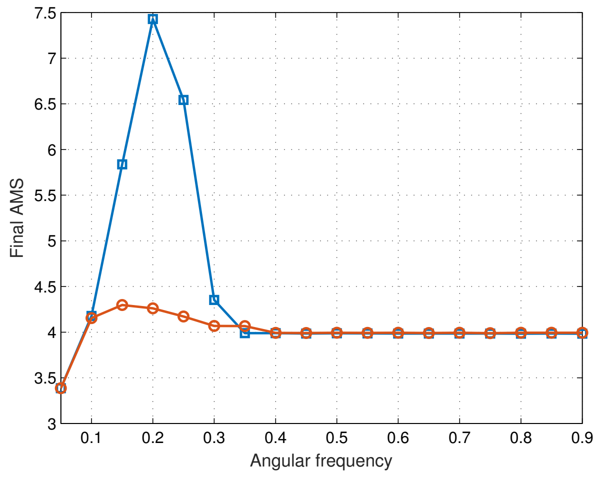

Figure 7. Example 4: Final AMS of the RaMSS (squares) and C-RaMSS (circles) as a function of the input angular frequency.

always greater than 95%, while that of the RaMSS is significantly affected by the increase in the noise level.

4.6. *Example 6: Oversized candidate regressor pool*

We here analyze the computational effort required by the C-RaMSS on system S_1 for an increasing size of the candidate regressor pool. Figure 9 reports some indicators associated to the computational load, such as the elapsed time to convergence and the total number of explored models required to achieve convergence as a function of the size of the regressor set. The latter is determined as $\frac{(n_d+n_y+n_u)!}{n_d!(n_y+n_u)!}$, where n_d is the maximum degree and n_u and n_y are the maximum lags.

Remarkably, the size of the problem did not influence the results in terms of correctness (that is almost always 100%), indicating how robust the two algorithms are. As expected, the C-RaMSS is costlier than the RaMSS, but overall extracts and processes fewer models, confirming its improved exploration capabilities (less iterations are also typically required to achieve convergence). The computational time (< 10 min) is largely acceptable if less than 500 regressors are employed, while

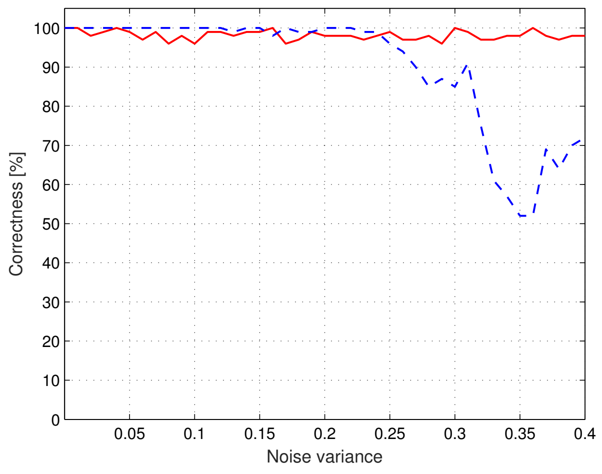

Figure 8. Example 5: Robustness of the RaMSS (dashed line) and C-RaMSS (solid line) algorithms under increasing noise levels.

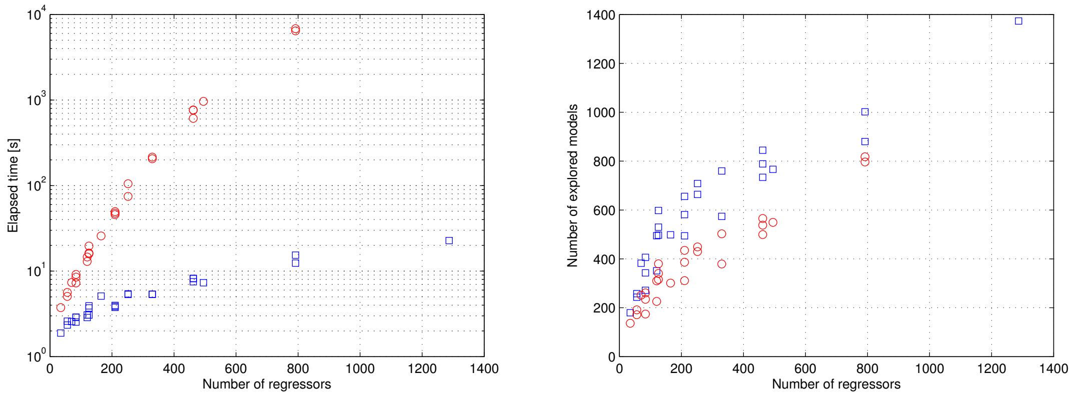

Figure 9. Example 6: Elapsed time (left) and total number of explored models (right) for the RaMSS (squares) and the C-RaMSS (circles).

for larger problem instances the RaMSS is more viable. Apparently, the latter can process up to 3 thousands regressors in less than 100 s. Consider also that these figures have been obtained using non-optimized Matlab code, so that overall the computational effort does not seem to be a severe issue. Regarding the explored models, it is noticeable that their increase is linear with respect to the size of the problem for both algorithms.

4.7. *Comparative analysis*

An extensive comparative analysis of the C-RaMSS with the FROE, the FROE^t, the iOFR, the iOFR^t, and the RaMSS has been carried out to assess the effectiveness of the proposed algorithm. The primary indicator studied here is the correctness of the model selection, reported in Table 5. The randomized algorithms (RaMSS and C-RaMSS) are evaluated in terms of the percentage of runs returning the correct model. The other algorithms being deterministic a single solution is obtained, regarding which we indicate how many correct terms are actually selected (c) or missing (m), and how many incorrect terms are included (w). Besides the correctness of the model selection, we also report in Table 6 the performance of the identified models in terms of the MSPE evaluated on a test data-set (not used for identification purposes). In Table 7 we report the standard deviation on the performance for the two randomized algorithms. Finally, the elapsed time for both the RaMSS and the C-RaMSS is provided in Table 8. As for the FROE and iOFR algorithms, the elapsed time is always less than 1 s in all experiments.

The studied systems are the already introduced S_1 and S_2, plus some additional ones taken from the literature, denoted S_4 (Cheng, Wang, & Jinglu, 2011), S_5 (Wei & Billings, 2008), and S_6 (Piroddi & Spinelli, 2003):

$$\begin{aligned} S_4\text{: } y(t) &= 0.3u(t-3) - 0.6u(t-2)^2y(t-1) + 0.7u(t-1)u(t-2)^2 - 0.5y(t-2) \\ &\quad -0.9u(t-1)u(t-2)u(t-3)y(t-3) - 0.8y(t-2)^3u(t-2) + e(t), \\ &\text{with } u(t) \sim WUN(-1,1),\ e(t) \sim WGN(0,0.02) \end{aligned}$$

$$\begin{aligned} S_5\text{: } y(t) &= -1.7y(t-1) - 0.8y(t-2) + u(t-1) + 0.8u(t-2) + e(t), \\ &\text{with } u(t) \sim WUN(-2,2),\ e(t) \sim WGN(0,0.01) \end{aligned}$$

$$\begin{aligned} S_6\text{: } y(t) &= 0.5y(t-1) + 0.8u(t-2) + u(t-1)^2 - 0.05y(t-2)^2 + 0.5 + e(t), \\ &\text{with } u(t) \sim WGN(0,0.3),\ e(t) \sim WGN(0,0.01). \end{aligned}$$

We chose the linear system S_5 to prove the effectiveness of the method in the case of a largely overparameterized model class. S_6 is a nonlinear system that also includes a constant term. Finally, S_4 has a higher degree than all the other considered systems. The previously used system S_3 is not considered in the analysis of correctness since the model family does not include the true system.

Regarding the model selection process, the candidate regressor set includes all terms up to degree $n_d = 3$ and with maximum lags $n_u = n_y = 4$ (165 terms) for all systems, except the last for which the parameters $n_d = 4$ and $n_u = n_y = 3$ (210 terms) have been employed.

Table 5. Comparative analysis: correctness.

	RaMSS	C-RaMSS	FROE			FROE^t			iOFR			iOFR^t		
			c	m	w	c	m	w	c	m	w	c	m	w
S_1	100%	100%	3	1	1	3	1	0	4	0	0	4	0	0
S_2	100%	100%	3	1	1	3	1	0	4	0	0	4	0	0
S_3	–	–	–	–	–	–	–	–	–	–	–	–	–	–
S_4	100%	100%	3	3	5	3	3	5	4	2	3	4	2	1
S_5	89%	95%	2	2	1	2	2	1	2	2	1	2	2	1
S_6	78.5%	95%	4	1	2	3	1	1	4	1	2	4	1	1

Table 6. Comparative analysis: MSPE.

	RaMSS	C-RaMSS	FROE	FROE^t	iOFR	iOFR^t
S_1	$3.420 \cdot 10^{-3}$	$3.420 \cdot 10^{-3}$	$1.127 \cdot 10^{-2}$	$1.201 \cdot 10^{-2}$	$3.420 \cdot 10^{-3}$	$3.420 \cdot 10^{-3}$
S_2	$9.045 \cdot 10^{-3}$	$9.045 \cdot 10^{-3}$	$1.725 \cdot 10^{-2}$	$1.811 \cdot 10^{-2}$	$9.045 \cdot 10^{-3}$	$9.045 \cdot 10^{-3}$
S_3	$8.800 \cdot 10^{-3}$	$8.400 \cdot 10^{-3}$	$1.240 \cdot 10^{-2}$	$1.280 \cdot 10^{-2}$	$1.240 \cdot 10^{-2}$	$1.280 \cdot 10^{-2}$
S_4	$1.692 \cdot 10^{-2}$	$1.692 \cdot 10^{-2}$	$2.347 \cdot 10^{-2}$	$2.347 \cdot 10^{-2}$	$2.349 \cdot 10^{-2}$	$2.363 \cdot 10^{-2}$
S_5	$9.373 \cdot 10^{-3}$	$8.627 \cdot 10^{-3}$	$9.187 \cdot 10^{-2}$	$9.187 \cdot 10^{-2}$	$9.187 \cdot 10^{-2}$	$9.187 \cdot 10^{-2}$
S_6	$9.727 \cdot 10^{-3}$	$9.232 \cdot 10^{-3}$	$1.240 \cdot 10^{-1}$	$1.241 \cdot 10^{-1}$	$1.239 \cdot 10^{-1}$	$1.241 \cdot 10^{-1}$

Table 7. Comparative analysis: standard deviation on the MSPE for RaMSS and C-RaMSS.

	RaMSS	C-RaMSS
S_1	0	0
S_2	0	0
S_3	$5.7 \cdot 10^{-4}$	$4.6 \cdot 10^{-4}$
S_4	0	0
S_5	$3.4 \cdot 10^{-3}$	$1.2 \cdot 10^{-3}$
S_6	$1.1 \cdot 10^{-3}$	$3.3 \cdot 10^{-5}$

Table 8. Comparative analysis: average elapsed time (in seconds) for RaMSS and C-RaMSS.

	RaMSS	C-RaMSS
S_1	3.81	14.93
S_2	3.40	13.83
S_3	89.10	112.38
S_4	11.11	40.05
S_5	6.53	12.88
S_6	20.44	44.06

The C-RaMSS outperforms (or at least equals) the other methods for all the tested systems, both in terms of correctness and performance. The most interesting result is obtained for S_6 for which a significant gain in correctness is observed w.r.t. RaMSS and also a significant improvement in accuracy is achieved w.r.t. the incremental methods. This shows in general the greater effectiveness of randomized methods compared to incremental approaches. The analysis also implies that the second order information employed by the C-RaMSS sometimes yields a significant payoff in terms of improved correctness, thank to the perfected exploration of the model space.

5. Conclusions

A variant of the recently introduced randomized algorithm RaMSS denoted C-RaMSS is proposed for nonlinear system identification based on the NARX model family. The RaMSS recasts the MSS problem in a probabilistic framework where a model distribution is defined in terms of the probabilities of each regressor to be present in the model, or RIPs (regressor inclusion probabilities). A sampling procedure is applied to generate a population of models according to the said distribution. Then, the RIPs are updated according to the performance of the entire population of extracted models. The algorithm is iterated until it converges to a limit model distribution.

The novel variant introduces a multivariate Bernoulli distribution to formalize the dependencies between regressors, as opposed to the RaMSS which is based on an independence assumption. More specifically, the CLF method is used to simulate these correlated binary variables with given mean and correlation, without having to fully specify the joint distribution. In so doing, the probability of extracting each regressor depends on the previously extracted ones. The RIPs are updated as in the RaMSS, and a new update rule is introduced for the correlation matrix. The latter is based on a weighted cosine distance measure calculated on a VSM where each regressor is defined in terms of the average performance of the models in which it appears.

The proposed algorithm is validated over different systems taken from the literature. The obtained results show its advantages in terms of robustness and reliability with respect to the RaMSS at the expense of an increase in the computational efficiency, due to the heavier simulation mechanism required to extract the models according to a multivariate correlated Bernoulli distribution. Nonetheless, this loss in efficiency is acceptable since the algorithm works on a batch of collected system observations, rather than on on-line data. In this respect, it must be also observed that the

C-RaMSS typically requires fewer iterations to converge and explores fewer model structures, which partially balances the increased computational burden. Furthermore, the proposed algorithm tends to identify more compact models than the RaMSS, when the exciting signal has a low frequency content. This is an appreciable result since compact models are more robust and easier to interpret.

References

Aguirre, L., & Billings, S. (1995). Dynamical effects of overparametrization in nonlinear models. *Physica D: Nonlinear Phenomena*, *80*(1), 26–40.

Baldacchino, T., Anderson, S., & Kadirkamanathan, V. (2013). Computational system identification for Bayesian NARMAX modelling. *Automatica*, *49*(9), 2641–2651.

Berry, M. W., Drmac, Z., & Jessup, E. R. (2006). Matrices, vector spaces, and information retrieval. *SIAM Review*, *41*(2), 335–362.

Billings, S., Chen, S., & Korenberg, M. (1989). Identification of MIMO nonlinear systems using a forward-regression orthogonal estimator. *International Journal of Control*, *49*, 2157–2189.

Billings, S. A. (2013). *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains.* Wiley.

Bonin, M., Seghezza, V., & Piroddi, L. (2010). NARX model selection based on simulation error minimisation and LASSO. *IET Control Theory & Applications*, *4*(7), 1157–1168.

Chen, S., Hong, X., & Harris, C. J. (2003). Sparse kernel regression modeling using combined locally regularized orthogonal least squares and d-optimality experimental design. *IEEE Transactions on Automatic Control*, *48*(6), 1029–1036.

Cheng, Y., Wang, L., & Jinglu, H. (2011). A two-step scheme for polynomial narx model identification based on moea with prescreening process. *IEEJ Transactions on Electrical and Electronic Engineering*, *6*(3), 253–259.

Devroye, L. (1986). *Nonuniform random variate generation.* New York: Springer-Verlag.

Falsone, A., Piroddi, L., & Prandini, M. (2015). A randomized algorithm for nonlinear model structure selection. *Automatica*, *60*, 227–238.

Guo, Y., Guo, L. Z., Billings, S. A., & Wei, H. L. (2015). An iterative orthogonal forward regression algorithm. *International Journal of Systems Science*, *46*, 776–789.

Haber, R., & Unbehauen, H. (1990). Structure identification of nonlinear dynamic systems – a survey on input/output approaches. *Automatica*, *26*(4), 651–677.

Hong, X., Mitchell, R., Chen, S., Harris, C., Li, K., & Irwin, G. (2008). Model selection approaches for non-linear system identification: a review. *International Journal of Systems Science*, *39*(10), 925–946.

Lee, A. (1997). Some simple methods for generating correlated categorical variates. *Computational Statistics & Data Analysis*, *26*, 133–148.

Leontaritis, I., & Billings, S. (1985a). Input-output parametric models for non-linear systems part I: deterministic non-linear systems. *International Journal of Control*, *41*(2), 303–328.

Leontaritis, I., & Billings, S. (1985b). Input-output parametric models for non-linear systems part II: stochastic non-linear systems. *International Journal of Control*, *41*(2), 329–344.

Li, K., Peng, J., & Irwin, G. (2005). A fast nonlinear model identification method. *IEEE Transactions on Automatic Control*, *50*(8), 1211–1216.

Lunn, A., & Davies, S. (1998). A note on generating correlated binary variables. *Biometrika*, *85*(2), 487–490.

Mao, K., & Billings, S. (1999). Variable selection in non-linear systems modelling. *Mechanical Systems and Signal Processing*, *13*(2), 351–366.

MATLAB. (2014). *Version 2014b.* Natick, Massachusetts: The MathWorks Inc.

Oman, S., & Zucker, D. (2001). Modeling and generating correlated binary variables. *Biometrika*, *88*(1), 287–290.

Palumbo, P., & Piroddi, L. (2000). Seismic behaviour of buttress dams: nonlinear modelling of a damaged buttress based on ARX/NARX models. *Journal of Sound and Vibration*, *239*, 405–422.

Piroddi, L., & Spinelli, W. (2003). An identification algorithm for polynomial NARX models based on simulation error minimization. *International Journal of Control*, *76*(17), 1767–1781.

Piroddi, L., & Spinelli, W. (2013, August 27-28). A pruning method for the identification of polynomial NARMAX models. In 13th *IFAC symposium on system identification* (pp. 1108–1113). Rotterdam, The Netherlands.

Qaqish, B. F. (2003). A family of multivariate binary distributions for simulating correlated binary variables with specified marginal means and correlations. *Biometrika*, *90*(2), 455–463.

Rodriguez-Vazquez, K., Fonseca, C. M., & Fleming, P. J. (2004, July). Identifying the structure of nonlinear dynamic systems using multiobjective genetic programming. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, *34*(4), 531–545.

Sjöberg, J., Zhang, Q., Ljung, L., Benveniste, A., Delyon, B., Glorennec, P. Y., … Juditsky, A. (1995). Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, *31*(12), 1691–1724.

Wei, H.-L., & Billings, S. (2008). Model structure selection using an integrated forward orthogonal search algorithm assisted by squared correlation and mutual information. *International Journal of Modelling, Identification and Control*, *3*(4), 341–356.