# A randomized algorithm for nonlinear model structure selection

Alessandro Falsone [a], Luigi Piroddi [a], Maria Prandini [a]

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

**Abstract**

The identification of polynomial Nonlinear Autoregressive [Moving Average] models with eXogenous variables (NAR[MA]X) is typically carried out with incremental model building techniques that progressively select the terms to include in the model. The Model Structure Selection (MSS) turns out to be the hardest task of the identification process due to the difficulty of correctly evaluating the importance of a generic term. As a result, classical MSS methods sometimes yield unsatisfactory models, that are unreliable over long-range prediction horizons. The MSS problem is here recast into a probabilistic framework based on which a randomized algorithm for MSS is derived, denoted RaMSS. The method introduces a tentative probability distribution over models and progressively updates it by extracting useful information on the importance of each term from sampled model structures. The proposed method is validated over models with different characteristics by means of Monte Carlo simulations, which show its advantages over classical and competitor probabilistic MSS methods in terms of both reliability and computational efficiency.

*Key words:* Model identification; Nonlinear models; Polynomial NARX models; Randomized algorithms; Regressor selection; Prediction error minimization methods.

## 1 Introduction

System identification is the process of building a mathematical model of a dynamical system from input-output data [35]. In particular, the black-box identification of nonlinear systems is a singularly difficult and challenging problem, since it amounts to solving an optimization problem with a mixed combinatorial (model structure selection) and continuous (parameter estimation) nature, [34], [19], [6].

We are here mainly concerned with recursive input/output (I/O) models of the Nonlinear Autoregressive [Moving Average] with eXogenous variables (NAR[MA]X) class, [21], where the current value of the system output is obtained as a nonlinear functional expansion of lagged input and output (and possibly noise) terms. Polynomial NARX/NARMAX models have earned widespread interest in view of their flexibility and representation capabilities, and several applications are documented (see, *e.g.*, [31], [24], [2], [27], [12], [9], [22], [36], [28]). Various identification algorithms have been proposed in the literature for NARX/NARMAX models, mainly based on the Prediction Error Minimization (PEM) framework for parameter estimation, which is in fact computationally convenient in view of the linear-in-the-parameters structure of the model. The main difficulty addressed by such algorithms is the selection of an appropriate model structure, considering that functional approximation using families of basis functions often leads to an exponential increase in the number of candidate model structures (curse of dimensionality), a critical issue of polynomial expansions in particular.

Classical model selection techniques based on information criteria, such as the AIC (Akaike Information Criterion), the BIC (Bayesian Information Criterion), and similar indices, appear to be hardly applicable in the nonlinear framework. Essentially, these indices weigh the model accuracy against the model size (number of parameters) and are used in the linear framework to estimate the correct model size. In the nonlinear context, no simple relation between model size and accuracy can be established, because one can construct many models of the same size with very different regressors and these can have quite different performances. As a result, these criteria cannot be used to derive indications on whether to accept or discard a specific term [27], [11]. Regularization criteria like the Least Absolute Shrinkage Selection Operator (LASSO) operate in a similar direction by penalizing the model size in the model identification process. As such they are effective in reducing the model size, but not necessarily in the more difficult task of selecting the appropriate model structure [10].

An efficient method for tackling the Model Structure Selection (MSS) task for NARX/NARMAX models was first suggested in [20], based on an incremental model building procedure (*forward regression*). In detail, at each algorithm

⋆ Corresponding author A. Falsone. Tel. +39-02-23994028. Fax +39-02-23993412.

*Email addresses:* alessandro.falsone@polimi.it (Alessandro Falsone), luigi.piroddi@polimi.it (Luigi Piroddi), prandini@elet.polimi.it (Maria Prandini).

iteration a new term is included in the model based on an importance index, the Error Reduction Ratio (ERR), which evaluates the local accuracy improvement that can be gained by adding the term to the current model. The method also exploits Orthogonal Least Squares (OLS) to decouple the estimation of the various regressors. Accordingly, it is denoted Forward Regression Orthogonal Estimator (FROE). Several variants of this method have been introduced in the literature using both forward and backward regression schemes (see, *e.g.*, [8], [18], [26], [29], [23], [19], [39], [10], [15], [17]). The combinatorial optimization performed by the FROE in the space of all possible models follows a greedy scheme, so that there is no guarantee of convergence to the global minimum. Several other drawbacks, that may ultimately prevent the convergence towards the correct model, have been pointed out, *e.g.*, in [1], [7], [29]. They are essentially related to the inadequacy of the ERR index to express in an absolute way the importance of a regressor. Indeed, such measure depends on the specific model to which the regressor is to be added. Notice also that the PEM paradigm guarantees the unbiasedness of the parameter estimates only in ideal conditions, where the system is persistently excited and the model structure (including the disturbance model) exactly matches that of the target system, a condition which is typically not met in the model building process, precisely because the model is constructed iteratively.

Recently, some novel approaches have been introduced to address the nonlinear identification problem, based on randomized algorithms [37]. An algorithm based on the Expectation Maximization (EM) approach is presented in [4], that employs the particle filter to handle nonlinearities and jointly perform MSS and parameter identification. In [5], both tasks are dealt with in a unified Bayesian framework, that is suitable for describing the uncertainty in both parameters and structure. Structure and parameter variations are performed based on a statistical acceptance/rejection mechanism. Posterior distributions are inferred using the Reversible Jump Markov Chain Monte Carlo (RJMCMC) procedure. The introduction of random sampling favors the convergence to the global minimum. However, MCMC methods are known to require a burn-in period for the Markov chain to converge to the desired stationary distribution, and this calls for many iterations.

In this paper, a novel iterative randomized algorithm is introduced for the identification of nonlinear systems, based on a different probabilistic reformulation of the MSS problem. The method is here described for NARX models only, although the extension to the NARMAX case can be envisaged (and is a matter of current research endeavors). A Bernoulli random variable is associated to each regressor. These random variables are assumed to be independent and, at each iteration, the proposed algorithm generates a set of models, each one being independently extracted from the joint distribution of all regressors. More precisely, an extracted model will contain a specific regressor if the value taken by the Bernoulli random variable associated to that regressor is 1. Then, the parameters of the extracted models are estimated, and the performances of the parameterized models evaluated in terms of a suitable index based on the prediction and simulation errors. Finally, the Bernoulli distribution of each regressor is updated based on the performances of the entire population of extracted models. More precisely, a regressor probability is increased if, on average, the extracted models that contain that specific regressor perform better than those that do not, and decreased in the opposite case. The algorithm converges to a limit distribution corresponding to a specific model structure. Some examples are analyzed by means of Monte Carlo simulations to show the effectiveness of the adopted probabilistic formulation, and to illustrate the improved reliability of the proposed algorithm compared to currently available randomized methods.

The proposed approach has some features in common with evolutionary methods, such as genetic algorithms (GA) [32], in that it exploits randomness in choosing potential regressors and in that it processes populations of models. More in detail, a GA selects the fittest individuals in the current population and manipulates them to generate a new population, using specific pair-wise operators. In our framework, the "fitness" of each regressor is evaluated from an aggregate analysis of the whole population. All individuals of the population contribute to the evaluation of the regressors, either reinforcing or discouraging their selection. Then, the new population is generated from scratch, based on the aggregate information derived from the current population.

A preliminary version of this work is given in [14]. The present paper significantly extends that contribution both from a theoretical and a methodological viewpoint. More specifically, the iterative algorithm is here better formalized within an appropriately defined probabilistic framework, and its convergence properties are established. Furthermore, a more extensive assessment of the performance of the proposed approach has been carried out via numerical examples taken from the literature. In particular, the behavior of the algorithm is analyzed in critical operating conditions, such as when a slowly varying input signal is used or when the randomized procedure is only allowed to get partial information on the correct model structure. In the case of a slowly varying input signal, the possible advantages related to the use of the simulation error for performance evaluation purposes are also discussed.

The rest of the paper is organized as follows. Section 2 provides the basic framework and notation for nonlinear system identification of NARX models and briefly reviews the main approaches in the literature. Section 3 discusses the crucial issue of how to evaluate the importance of each regressor. The proposed method is illustrated in Section 4 and then tested in Section 5. Finally, some concluding remarks are drawn in Section 6.

## 2 Preliminaries

### *2.1 The NARX model class*

A NARX model [21] is described by the following input/output recursive equation:

$$\begin{aligned} y(k) =& f(y(k-1), \ldots, y(k-n_y), \\ & u(k-1), \ldots, u(k-n_u)) + e(k) \end{aligned} \tag{1}$$

where $y(k)$, $u(k)$, and $e(k)$ are the output, input, and (white) noise signals, respectively, $n_y$ and $n_u$ being suitable maximum lags, and $f(\cdot)$ is an unknown nonlinear function.

The objective of a NARX model identification process is to find an estimator $\hat{f}$ for function $f$ based on the available input/output data. Now, provided that $f$ is a continuous function, it can be approximated using various types of nonlinear functional expansions, such as piecewise linear models, rational polynomial models, radial basis function or sigmoidal neural networks, etc. [34], [19]. All these functional expansions are *universal approximators*, *i.e.* they can approximate $f$ to an arbitrary level of accuracy, provided they are endowed with sufficient degrees of freedom. The most frequently used nonlinear functional expansions are those that provide a *linear* combination of nonlinear basis functions, [34], [19]:

$$\hat{y}(k) = \hat{f}(\boldsymbol{x}(k)) = \sum_{j=1}^{m} \vartheta_j \, \varphi_j(\boldsymbol{x}(k)), \tag{2}$$

where $\boldsymbol{x}(k) = [y(k-1), \ldots, y(k-n_y), u(k-1), \ldots, u(k-n_u)]$, $\varphi_j(\boldsymbol{x}(k))$, $j = 1, \ldots, m$, are given nonlinear basis functions, $\vartheta_j$ are the corresponding coefficients, and $m$ is the number of basis functions. In vector form:

$$\hat{y}(k) = \boldsymbol{\varphi}^T(k)\, \boldsymbol{\vartheta}, \tag{3}$$

where $\boldsymbol{\varphi}(k) = [\varphi_1(\boldsymbol{x}(k)), \ldots, \varphi_m(\boldsymbol{x}(k))]^T$ is the regressor vector and $\boldsymbol{\vartheta} = [\vartheta_1, \ldots, \vartheta_m]^T$ the parameter vector. An often adopted functional expansion is the polynomial one, where the generic regressor is a monomial in the components of $\boldsymbol{x}(k)$:

$$\begin{aligned} \varphi_j(\boldsymbol{x}(k)) = y(k-d_{j1}) \cdots y(k-d_{ji}) \cdot \\ \cdot u(k-d_{j,i+1}) \cdots u(k-d_{j\ell}) \end{aligned} \tag{4}$$

with $d_{j1}, \ldots, d_{j\ell} \in \mathbb{N}$ and $0 \leq \ell \leq M$, $M$ being the maximum degree of the polynomial expansion (the regressor corresponding to $\ell = 0$ is the constant term $\varphi_j = 1$). Polynomial NARX models provide a fairly general and flexible class of models. They are typically more compact models compared to, *e.g.*, neural networks or support vector machines, and provide a better and clearer insight on the nonlinearities underlying the system dynamics.

Linear-in-the-parameter models, such as (3), have several features: i) they allow parameter estimation through simple algorithms of the Least Squares (LS) family, ii) they are well structured for adaptive learning, iii) the algorithms for their estimation have provable convergence conditions, and iv) they are amenable to a more direct interpretation and easier usage in control engineering applications, [19].

NARMAX models are a generalization of the NARX class, such that $\boldsymbol{x}(k)$ also includes past noise terms, therefore allowing for a more flexible representation of the disturbance model to account for unmodeled dynamics. The inclusion of a non-trivial disturbance model reduces the bias in the parameter estimates. However, it also aggravates the computational effort of the MSS task, since the set of candidate regressors for the model is typically largely increased. Furthermore, the parameter estimation algorithms that can be employed in the presence of the MA part of the model are more complex and computationally demanding. Besides, given the significant increase in the modeling capabilities of nonlinear models compared to linear ones, the role of the disturbance model is much de-emphasized in the considered framework. For these reasons, in the sequel we stick to the simpler NARX class, although the presented theory can also be extended to NARMAX models.

### *2.2 Parameter estimation and Student's t-test*

The linear-in-the-parameters structure (3) of the NARX model allows the use of LS algorithms for parameter estimation. Assume that a data-set $\{(u(k), y(k)),\ k = 1, \ldots, N\}$ of input/output pairs is available for identification purposes. The model performance is measured in terms of the Mean Squared Prediction Error (MSPE):

$$\mathrm{MSPE} = \frac{1}{N} \sum_{k=1}^{N} (y(k) - \hat{y}(k))^2, \tag{5}$$

where $\hat{y}(k)$ denotes the one-step-ahead prediction of $y(k)$. The optimal parameter estimate of a linear regression of type (3), which minimizes (5), is given by the LS formula:

$$\hat{\boldsymbol{\vartheta}} = \left( \sum_{k=1}^{N} \boldsymbol{\varphi}(k)\, \boldsymbol{\varphi}^T(k) \right)^{-1} \sum_{k=1}^{N} \boldsymbol{\varphi}(k)\, y(k)\,. \tag{6}$$

The variance of the estimated parameters can be estimated as:

$$\hat{\sigma}_j^2 \approx \hat{\sigma}_e^2 V_{jj}, \tag{7}$$

where $\hat{\sigma}_e^2$ is the estimated noise variance, obtained by scaling the mean squared residual by a factor $N/(N-m)$, and $V_{jj}$ is the $j^{\text{th}}$ diagonal element of $V = (\sum_{k=1}^{N} \boldsymbol{\varphi}(k)\, \boldsymbol{\varphi}^T(k))^{-1}$.

The parameter variances $\hat{\sigma}_j^2$ can be used in a Student's t-test to determine the statistical relevance of each regressor. More precisely, let $t_{\alpha,N-m}$ be the $100(1-\alpha)$ percentile of the Student's t distribution with $N-m$ degrees of freedom. Then, the $100(1-\alpha)\%$ confidence interval for each $\vartheta_j$ is given by:

$$[\hat{\vartheta}_j - \hat{\sigma}_j\, t_{\alpha,N-m}\,;\ \hat{\vartheta}_j + \hat{\sigma}_j\, t_{\alpha,N-m}]. \tag{8}$$

If the interval defined by expression (8) does not contain zero, $\vartheta_j$ is not zero with a confidence of $100(1-\alpha)\%$. Otherwise, $\vartheta_j$ is not significantly different from zero and the null hypothesis $\vartheta_j = 0$ cannot be rejected. In the latter case, the corresponding regressor $\varphi_j$ is considered to be statistically irrelevant for the given model, [16], and therefore redundant.

Concerning the model performance evaluation, it is sometimes useful to resort to the Mean Squared Simulation Error

(MSSE):

$$\mathrm{MSSE} = \frac{1}{N}\sum_{k=1}^{N}(y(k) - \hat{y}_s(k))^2, \tag{9}$$

where $\hat{y}_s$ denotes the simulated output of the model (sometimes referred to as free-run prediction). It is argued in [29] that using the MSSE as a model evaluation criterion can improve the robustness of the MSS process in partial identifiability conditions. In the sequel, to evaluate model performances, we will employ exponential versions of the MSPE and MSSE indices:

$$\mathcal{J}_p = e^{-K\cdot\mathrm{MSPE}}, \tag{10}$$

$$\mathcal{J}_s = e^{-K\cdot\mathrm{MSSE}}, \tag{11}$$

where $K$ is a tuning parameter. $\mathcal{J}_p$ and $\mathcal{J}_s$ provide indices of the model quality in the $[0, 1]$ range, higher performances corresponding to values of the criteria close to 1. The exponential transformation is also useful in the MSS task, since it tends to amplify the differences between models with similar performance, so that even small improvements can be detected. For further generality and in analogy with [3], one can employ the following combined performance index:

$$\mathcal{J} = \alpha\mathcal{J}_s + (1-\alpha)\mathcal{J}_p, \tag{12}$$

where $\alpha \in [0, 1]$ is a user defined parameter [1] .

### *2.3 MSS: the FROE algorithm*

The most popular MSS algorithms for NARX models is arguably the FROE [20]. This is an iterative model building procedure, that adds one term per iteration to the model based on the ERR criterion, which is used to rate the regressor importance. From a computational viewpoint, the FROE uses OLS to estimate the parameters, exploiting the orthogonalization to decouple the regressors.

The ERR coefficient measures the contribution of each regressor to the explained output variance, and it is computed by adding the regressor to the current model and evaluating the corresponding reduction of the MSPE normalized with respect to the model output variance [20]. At each iteration, the ERR coefficient is evaluated for all regressors not yet included in the model, and the regressor $\varphi_j$ with the highest ERR (*i.e.*, the regressor whose inclusion most improves the MSPE), is selected and added to the current model. Notice that the prediction error variance of the model is monotonically decreasing with the iterations.

The procedure applies to both NARX and NARMAX models, in the latter case the I/O structure being identified first and the model being subsequently complemented with additional noise dependent terms.

One of the most critical issues of the FROE is that it does not take into account the fact that the importance of a regressor as measured by the ERR index is not an absolute, "global" value that can be associated to the regressor itself independently of the particular model considered, but rather it is a function of the model with respect to which it is calculated, [29], which characterizes it as a "local" index. As a consequence, for example, a regressor that is included at an early stage of the procedure may turn out to be irrelevant when the algorithm identifies the complete model structure, [15]. Testing the included regressors for redundancy partially mitigates the problem [30]. Another strategy that aims at reducing the sensitivity to bad initial choices consists in iterating the FROE starting from different initial regressors, as done by the iterative orthogonal forward regression (iOFR) method [17].

Overall, the model building procedure is critically affected by the fact that the regressor inclusion policy is based on a local estimate of the regressor importance, as can be given by the ERR. Furthermore, the FROE algorithm suffers from the very incremental nature of the model building procedure, with its inherently local search mechanism in the space of model structures, which inevitably leads to suboptimal solutions, [20], [19], [29], [5]. Clearly, the MSS task would be greatly improved if the significance of a term could be established in absolute, global terms, independently of the model structure.

### *2.4 MSS: the RJMCMC-based approach of [5]*

A radically different approach to NARX model identification has been recently presented in [5], that employs a Bayesian approach to derive posterior distributions for both the model structure and its parameters by way of a sampling approach. The Bayesian framework also provides a way to quantify the uncertainty in the model structure determination and the parameter estimation task, a feature that is absent in all deterministic approaches.

The algorithm is based on the RJMCMC procedure, which extends the Metropolis-Hastings (MH) algorithm to account for "jumps" in the parameters dimension, as occur if the model structure is updated. It is an iterative algorithm, which at each iteration randomly performs one of the following actions:

i) *birth move*: a new regressor is randomly selected from a predefined pool of candidate terms and proposed for inclusion in the current model structure.
ii) *death move*: a regressor in the current model structure is chosen at random and tested for exclusion.
iii) *update move*: parameters are updated using a MH random walk.

The probabilities of performing birth or death moves are updated at each iteration, according to the likelihood that the size of the real model is larger or smaller than the current model. The proportion between update moves and birth/death moves is a design parameter. As prescribed by the MH procedure, a move is first proposed and then the algorithm decides if it can be accepted or not, based on the information collected at previous iterations. In the NARMAX case this proposal-acceptance mechanism is repeated twice: the first time for the process model and the second time for the noise model. After a burn-in period the algorithm should

[1] Optimization over $\alpha$ is not carried out, but rather $\mathcal{J}$ is optimized for a given value of $\alpha$.

converge to a distribution over both the regressors and the parameters.

The joint identification of both structure and parameters, while appealing in principle, turns out to be problematic, especially because the parameter distribution over different model structures is often quite complex. Indeed, the same parameter may assume significantly different values depending on the model structure in which the associated regressor appears. Figure 1 shows the histogram of the LS estimates of the same parameter when the rest of the model structure varies over a given model family.

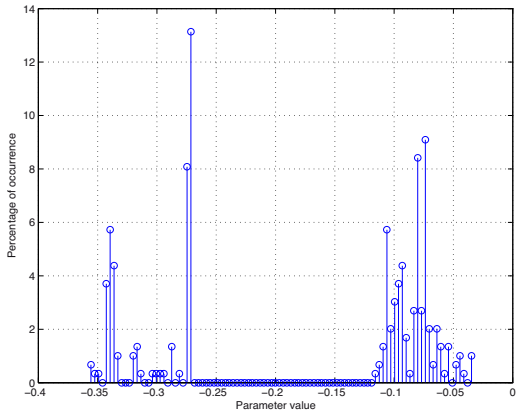


Figure 1. Distribution of the estimates of a parameter over different models that include the corresponding regressor.

## 3 Evaluating the regressor importance

A crucial step in the MSS task is the evaluation of the importance of the regressors, which ultimately drives the selection process. In the FROE algorithm the importance of each term is evaluated at each iteration by means of the ERR criterion. As already commented, the ERR provides a measure of the regressor importance which is only valid for a specific model, which makes it a quite erratic index depending on the considered model. A much more robust and reliable evaluation of the regressor significance can be obtained by analyzing a collection of models rather than a single one.

Consider for example the following system:

$$y(k) = u(k-1)^2 - 0.7y(k-2)u(k-1) + e(k), \quad (13)$$

where $u(\cdot) \sim \text{WGN}(0, 0.36)$ and $e(\cdot) \sim \text{WGN}(0, 0.01)$. Assume that, based on a set of input/output measurements, we want to perform MSS over the model family obtained by including all monomials up to order 2 of the terms $y(k-1)$, $y(k-2)$, $u(k-1)$, which amounts to a family of $m = 10$ regressors, for a total of $2^m = 1024$ possible models. The considered setting is sufficiently small to allow an exhaustive analysis of all the possible models. Accordingly, for each possible model we can estimate the parameters (with LS), remove all irrelevant regressors using the Student's t-test, and finally evaluate its performance based on $\mathcal{J}_p$. Let $I_j^{p+}$ be the average performance of the models that contain the $j^{\text{th}}$ regressor (after the elimination of redundant terms), and $I_j^{p-}$ the corresponding average performance of the remaining models. Then, as shown in Figure 2, index $I_j^p = I_j^{p+} - I_j^{p-}$ provides a much clearer indication of the presence or absence of the $j^{\text{th}}$ regressor, compared to the ERR coefficient calculated assuming an empty model as reference (as occurs at the onset of the FROE algorithm). In

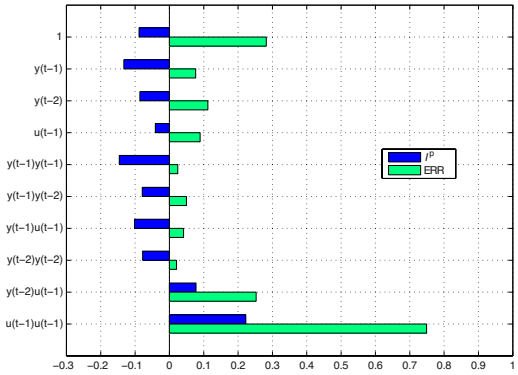


Figure 2. Evaluation of the importance of regressors: $I^p$ vs. ERR.

fact, $I_j^p$ is positive only for the two correct terms. On the other hand, the ERR is positive for all terms, and provides an incorrect ranking of the terms (the constant term is preferred over the term $y(k-2)u(k-1)$). Though there is a chance that this ranking will be corrected at the second iteration of the FROE, the example still serves the purpose of illustrating the capabilities of the proposed formulation.

Things get more involved if the input signal employed in the identification experiment has poor excitation properties. This is indeed the case when experiment design is not applicable, and the input-output data-set is an input to the identification process. Consider, *e.g.*, the case where system (13) is excited by a low-pass filtered white noise:

$$u(k) = 1.85u(k-1) - 0.855u(k-2) + \xi(k), \quad (14)$$

where $\xi(\cdot) \sim \text{WGN}\big(0, 2.28 \cdot 10^{-2}\big)$ and the variance of $u(k)$ is equal to the previous case. Signal (14) is an AR process, and as such it is persistently exciting (see, *e.g.*, [35]). Accordingly, if employed for parameter estimation in perfect model matching conditions (*i.e.*, the model structure coincides with that of the system generating the data), no bias is experimented. On the other hand, the MSS process is much more affected by this choice of input signal, as observed in several works [3], [29]. This occurs because such an input produces a slowly varying output, causing the difference between adjacent output samples to be very small. This makes several regressors look alike and complicates the discrimination of the correct ones.

The $I^p$ and ERR indices associated to the 10 regressors in this case are shown in Figure 3. It is immediately evident that the ERR is hardly capable of distinguishing the different regressors. In particular, regressors of the same "cluster" (*i.e.*, of the same type of nonlinearity), have very similar ERR values: compare, *e.g.*, regressors $y(k-1)u(k-1)$ and $y(k-2)u(k-1)$. By contrast, $I^p$ sharply discards 5 out

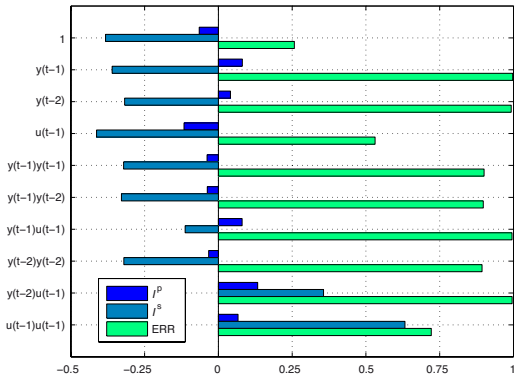


Figure 3. Evaluation of the importance of regressors: $I^p$, $I^s$, and ERR.

of 10 regressors and provides a positive indication for both correct regressors plus 3 redundant ones. Even better results are obtained if the same importance index is calculated with $\mathcal{J}_s$ in place of $\mathcal{J}_p$, as suggested in [29]. Indeed, as is apparent from Figure 3, the simulation-based index (denoted $I^s$) provides a sharp and precise classification of the regressors. This is very much in line with a number of previous results in the literature (see, *e.g.*, [29], [13], [3]).

## 4 A randomized approach to MSS

We now propose a novel MSS procedure where the decision regarding the inclusion of terms is taken based on a *population* of models, rather than a single one. MSS is reformulated as an optimization problem over a distribution of models, that is progressively refined using aggregate information obtained from a set of extracted models.

### *4.1 Probabilistic reformulation of the MSS problem*

Let $\mathcal{R} := \{\varphi_1, \dots, \varphi_m\}$ denote the set of the $m$ regressors in the chosen family, so that the power set of $\mathcal{R}$, *i.e.* $\mathcal{F} := 2^{\mathcal{R}}$, is the set of all possible model structures. In the following, we will assume that the true model, denoted $f^\star$, belongs to $\mathcal{F}$. Parameter estimation is carried out as explained in Section 2.2, and statistically irrelevant regressors terms are removed. The latter operation is accounted for by a function $\mathcal{T} : \mathcal{F} \to \tilde{\mathcal{F}}$, where the set of non-redundant models $\tilde{\mathcal{F}} = \{\tilde{f} = \mathcal{T}(f), f \in \mathcal{F}\}$ is a subset of $\mathcal{F}$. For the purpose of MSS, estimated models are rated with performance index $\mathcal{J} : \mathcal{F} \to \mathbb{R}^+$ as defined by (12).

According to the introduced notation, the structure selection problem can be formalized as that of finding the subset of regressors that maximizes the performance index over all non-redundant models. Throughout the paper we will assume that there exists only one such model and that this model coincides with the true one:

$$f^\star = \arg\max_{\tilde{f} \in \tilde{\mathcal{F}}} \mathcal{J}(\tilde{f}), \tag{15}$$

so that the identification problem is well-posed.

To solve the optimization problem (15), one should in principle explore all possible models in $f \in \mathcal{F}$, compute the corresponding $\tilde{f} = \mathcal{T}(f)$, and pick the non-redundant model with the best performance index. Such an exhaustive enumeration approach can hardly be considered viable, given that the number of possible models is $2^m$, and $m$ rapidly increases with $n_y$, $n_u$, and the polynomial degree $M$, a problem often referred to as "curse of dimensionality". More conveniently, the problem can be addressed by reformulating it in a probabilistic framework. For this purpose, we introduce the discrete random variable $\tilde{\Phi}$ which takes values in $\tilde{\mathcal{F}}$ according to a probability distribution $\mathcal{P}_{\tilde{\Phi}}$. The average performance of $\tilde{\Phi}$ is given by:

$$\mathbb{E}[\mathcal{J}(\tilde{\Phi})] = \sum_{\tilde{f} \in \tilde{\mathcal{F}}} \mathcal{J}(\tilde{f}) \mathcal{P}_{\tilde{\Phi}}(\tilde{f}). \tag{16}$$

Expression (16) is a convex combination of the performance indices of all models in $\tilde{\mathcal{F}}$. If we let $\mathcal{P}_{\tilde{\Phi}}$ vary over all possible distributions on $\tilde{\mathcal{F}}$, the maximum value of (16) is obtained by making all probability mass concentrate on the "true" model. Therefore, the solution of the optimization problem

$$\mathcal{P}^\star_{\tilde{\Phi}} = \arg\max_{\mathcal{P}_{\tilde{\Phi}}} \mathbb{E}[\mathcal{J}(\tilde{\Phi})] \tag{17}$$

is such that $\mathcal{P}^\star_{\tilde{\Phi}}(f^\star) = 1$, thus providing the same solution of problem (15).

A key feature of the proposed method is to provide a suitable parametrization for $\mathcal{P}_{\tilde{\Phi}}$. More specifically, we associate to each regressor $\varphi_j$ a (Bernoulli) random variable $\rho_j$, whose possible outcomes are 1 with probability $\mu_j$ and 0 with probability $1 - \mu_j$:

$$\rho_j \sim \mathrm{Be}(\mu_j)\,, \tag{18}$$

with $\mu_j \in [0, 1]$ and $j = 1, \dots, m$. We assume that all random variables $\rho_j$, $j = 1, \dots, m$, are independent. In the following, we will refer to $\mu_j$ as the Regressor Inclusion Probability (RIP) of the $j^{\text{th}}$ regressor, and $\boldsymbol{\mu} = [\mu_1 \ \dots \ \mu_m]^T$ is the vector of RIPs. Setting $\boldsymbol{\mu}$ induces a probability distribution $\mathcal{P}_\Phi$ over the models in $\mathcal{F}$. Precisely,

$$\mathcal{P}_\Phi(f) = \prod_{j:\varphi_j \in f} \mu_j \prod_{j:\varphi_j \notin f} (1 - \mu_j), \tag{19}$$

for any $f \in \mathcal{F}$. Note that, to each $f \in \mathcal{F}$ one can associate a limit distribution with $\mu_j = 1$ if $\varphi_j \in f$ and 0 otherwise, $j = 1, \dots, m$, such that $\mathcal{P}_\Phi(f) = 1$. In particular, we will denote as *target* limit distribution the one associated to $f^\star$.

Now, setting $\tilde{\Phi} = \mathcal{T}(\Phi)$ results in a probability distribution $\mathcal{P}_{\tilde{\Phi}}$ that is induced by $\mathcal{P}_\Phi$ through $\mathcal{T}$. Formally,

$$\mathcal{P}_{\tilde{\Phi}}(\tilde{f}) = \sum_{f \in \mathcal{F}:\, \mathcal{T}(f) = \tilde{f}} \mathcal{P}_\Phi(f) \tag{20}$$

with $\mathcal{P}_\Phi(f)$ depending on $\boldsymbol{\mu}$ through (19).

Exploiting the previous derivation, one can optimize (17) by tuning parameters $\mu_1, \dots, \mu_m$ so as to make $\mathcal{P}_\Phi$ concentrate

onto $f^\star$. Indeed, since from (20),

$$\mathcal{P}_{\tilde{\Phi}}(\tilde{f}) \geq \mathcal{P}_{\Phi}(\tilde{f}) \quad \forall \tilde{f} \in \tilde{\mathcal{F}}, \tag{21}$$

it holds that, if $\mathcal{P}_{\Phi}(f^\star)$ tends to 1, *i.e.* if the RIP distribution tends to the target limit one, then $\mathcal{P}_{\tilde{\Phi}}(f^\star)$ tends to 1 as well, so that the desired maximization of $\mathbb{E}[\mathcal{J}(\tilde{\Phi})]$ is achieved. Let

$$\mathcal{I}_j = \mathbb{E}[\mathcal{J}(\tilde{\Phi})|\varphi_j \in \tilde{\Phi}] - \mathbb{E}[\mathcal{J}(\tilde{\Phi})|\varphi_j \notin \tilde{\Phi}], \tag{22}$$

$j = 1, \ldots, m$, where the conditional expectations are set equal to 0 if the conditioning event has 0 probability to occur. Index $\mathcal{I}_j$ compares the average performance of the (non-redundant) models containing the $j^{\text{th}}$ regressor with that of the remaining ones. As such, it can be interpreted as a sort of global measure of the regressor importance, weighted in probability by the underlying model distribution induced by $\boldsymbol{\mu}$. More importantly, if the latter is not distant from the target limit distribution corresponding to $f^\star$, it can be shown (see Theorem 1 below) that $\mathcal{I}_j > 0$ iff $\varphi_j \in f^\star$.

**Theorem 1** *Let $\mathcal{P}_{\Phi}$ be the probability distribution induced by $\boldsymbol{\mu}$, according to* (19)*. Then, there exists $\delta \in (0, 1)$ such that if $\mathcal{P}_{\Phi}(f^\star) \geq \delta$ it holds that, for all $j \in \{1, \ldots, m\}$, $\mathcal{I}_j > 0$ if $\varphi_j \in f^\star$ and $\mathcal{I}_j < 0$ otherwise.*

**Proof** See Appendix A.1.

Theorem 1 suggests that the sign of indices $\mathcal{I}_j$, $j = 1, \ldots, m$, can provide reliable information for setting the $\mu_j$ parameters to the values of the target limit distribution. In practice, the expected values in (22) are estimated based on the empirical mean, and are, hence, affected by uncertainty. To mitigate this issue an adaptive rule is introduced in Section 4.2 for the tuning of the RIPs, to properly compromise between the already available information on the $\mathcal{I}_j$ indices and that gathered from new model samples. Another key issue related to Theorem 1 is that the property expressed therein is only guaranteed when $\mathcal{P}_{\Phi}(f^\star)$ is sufficiently high, and, hence, the RIP distribution is close to the target limit distribution. However, it will be shown in Section 5 that the result in Theorem 1 is quite conservative in this respect, since the proposed iterative RIP tuning method is capable of converging to the target limit distribution even if $\mathcal{P}_{\tilde{\Phi}}(f^\star) = 0$. In other words, the essential information regarding the correct model structure can be retrieved by proper processing of partial or incomplete models, that contain some but not all the correct regressors. In general, as discussed in the next section, indices $\mathcal{I}_j$ can provide correct information regarding the true model regressors even with relatively low values for $\mathcal{P}_{\tilde{\Phi}}(f^\star)$.

### *4.2 The RaMSS algorithm*

The probabilistic reformulation discussed in Section 4.1 can be exploited to construct an iterative randomized procedure that progressively refines the vector of RIPs $\boldsymbol{\mu}$ so as to concentrate the mass probability of $\mathcal{P}_{\Phi}$ (and hence that of $\mathcal{P}_{\tilde{\Phi}}$) onto $f^\star$. The idea is to extract a family of models based on the current RIP values, and update the latter based on the cumulative information obtained from the models, in the form of indices $\mathcal{I}_j$, $j = 1, \ldots, m$. Then, the procedure is repeated until convergence to a limit distribution corresponding to a specific model structure.

More in detail, each model extraction amounts to establishing for each regressor $\varphi_j$ if it belongs to the model, which is done through an extraction of the Bernoullian random variable $\rho_j$ associated to the regressor. Once the model structure has been defined, parameter estimation is carried out with LS (for computational reasons). The statistical significance of each parameter is established with a Student's t-test, and statistically non-significant regressors are eliminated. If the model is reduced, its parameters are re-estimated. Finally, all resulting models are evaluated based on the mixed index (12), and the RIPs are modified according to the tuning rule:

$$\mu_j(i+1) = \mu_j(i) + \gamma \mathcal{I}_j \tag{23}$$

for $j = 1, \ldots, m$, where the step size $\gamma > 0$ is a design parameter. In practice, $\mathcal{I}_j$ is estimated based on the sampled expected values of the model performances. In other words, each individual RIP is increased if the average performance of the models that include that specific regressor is greater than the average performance of the remaining ones. Expression (23) is similar to a gradient-based update rule, although $\mathcal{I}_j$ is not directly interpretable as the gradient of the cost function with respect to $\mu_j$. Nevertheless, the local convergence to the target limit distribution is still guaranteed thanks to the following result, which rests on Theorem 1.

**Theorem 2** *Let $\boldsymbol{\mu}$ be such that $\mathcal{P}_{\Phi}(f^\star) \geq \delta$, where $\delta$ is a value for which Theorem 1 holds. Then, the iterative application of* (23) *starting from $\boldsymbol{\mu}$ leads to the target limit distribution.*

**Proof** See Appendix A.2.

Algorithm 1 summarizes the whole model identification procedure.

Besides the input-output data and the candidate regressor set $\mathcal{R}$, the algorithm requires several parameters ($N_p$, $K$, $\alpha$, $\boldsymbol{\mu}$, $\mu_{\min}$, $\mu_{\max}$, $\varepsilon$), which are briefly discussed next. Parameter $N_p$ defines the number of models to be extracted at each iteration. Clearly, the more models are extracted at each iteration, the more robust will the regressor evaluation be (at an increased computational cost). In the examples (see Section 5) a value $N_p = 100$ has been used. The cost function depends on parameters $K$ and $\alpha$ (see expressions (10-12)). In particular, $K$ determines the sensitivity of the performance index (higher $K$ values amplify the performance difference between models) and ultimately influences the algorithm stopping (the algorithm stops when the regressor distribution converges, which occurs when no structural variation results in a significant improvement of $\mathcal{J}$). Parameter $\alpha$ determines the weight of the simulation performance index for model assessment. Vector $\boldsymbol{\mu}$ describes the initial regressor distribution. The algorithm is initialized by setting equal small probabilities for each regressor, say $\mu_j = 1/m$, $j = 1, \ldots, m$, thus encouraging the extraction of small-sized models at the early steps of the algorithm. If any information is available regarding the size $m^\circ$ of the "true" model, setting $\mu_j = m^\circ/m$ results in an initial set

**Algorithm 1** The RaMSS algorithm.

**Input:** $\{(u(k)\,, y(k)),\ k = 1, \ldots, N\}$, $\mathcal{R} = \{\varphi_j(k),\ j = 1, \ldots, m\}$, $N_p$, $K$, $\alpha$, $\boldsymbol{\mu}$, $\mu_{\min}$, $\mu_{\max}$, $\varepsilon$;

**Output:** $\boldsymbol{\mu}$

**repeat**
  **for** $i = 1$ **to** $N_p$ **do** ▷ Generate models
    $\boldsymbol{\psi}(k) = [\,]$;
    $\tau = 0$;
    **for** $j = 1$ **to** $m$ **do** ▷ Generate regressors
      Extract $r_j$ from $\mathrm{Be}(\mu_j)$;
      **if** $r_j = 1$ **then** ▷ Add regressor
        $\boldsymbol{\psi}(k) \leftarrow [\boldsymbol{\psi}^T(k)\ \ \varphi_j(k)]^T$;
        $\tau \leftarrow \tau + 1$;
      **end if**
    **end for**
    $\hat{\boldsymbol{\vartheta}} \leftarrow (\sum_{k=1}^{N} \boldsymbol{\psi}(k)\, \boldsymbol{\psi}^T(k))^{-1} \sum_{k=1}^{N} \boldsymbol{\psi}(k)\, y(k)$;
    $V \leftarrow (\sum_{k=1}^{N} \boldsymbol{\psi}(k)\, \boldsymbol{\psi}^T(k))^{-1}$;
    $\hat{\sigma}_e^2 \leftarrow \frac{1}{N-\tau} \sum_{k=1}^{N} (y(k) - \boldsymbol{\psi}^T(k)\, \hat{\boldsymbol{\vartheta}})^2$;
    **for** $h = 1$ **to** $\tau$ **do** ▷ Remove redundant terms
      $\hat{\sigma}_h^2 \leftarrow \hat{\sigma}_e^2 V_{hh}$;
      **if** $|\hat{\vartheta}_h| \leq \hat{\sigma}_h\, t_{\alpha, N-\tau}$ **then**
        Remove regressor $\psi_h(k)$ from $\boldsymbol{\psi}(k)$;
      **end if**
    **end for**
    $\hat{\boldsymbol{\vartheta}} \leftarrow (\sum_{k=1}^{N} \boldsymbol{\psi}(k)\, \boldsymbol{\psi}^T(k))^{-1} \sum_{k=1}^{N} \boldsymbol{\psi}(k)\, y(k)$;
    $\mathcal{J}^{(i)} \leftarrow \alpha \mathcal{J}_p^{(i)} + (1 - \alpha) \mathcal{J}_s^{(i)}$;
  **end for**
  **for** $j = 1$ **to** $m$ **do** ▷ Update RIPs
    $\mathcal{J}^+ \leftarrow 0$; $n^+ \leftarrow 0$; $\mathcal{J}^- = 0$; $n^- \leftarrow 0$;
    **for** $i = 1$ **to** $N_p$ **do**
      **if** $\varphi_j \in \boldsymbol{\psi}(k)$ **then**
        $\mathcal{J}^+ \leftarrow \mathcal{J}^+ + \mathcal{J}^{(i)}$; $n^+ \leftarrow n^+ + 1$;
      **else**
        $\mathcal{J}^- \leftarrow \mathcal{J}^- + \mathcal{J}^{(i)}$; $n^- \leftarrow n^- + 1$;
      **end if**
    **end for**
    $\mu_j \leftarrow \mu_j + \gamma \left( \frac{\mathcal{J}^+}{\max(n^+, 1)} - \frac{\mathcal{J}^-}{\max(n^-, 1)} \right)$;
    $\mu_j \leftarrow \max(\min(\mu_j, \mu_{\max}), \mu_{\min})$;
  **end for**
**until** $\min_{j=1,\ldots,m} |2\mu_j - 1| \geq 1 - \varepsilon$ ▷ Stopping criterion

of extracted models with an average number of terms equal to $m^\circ$. Notice that expression (23) does not ensure that parameters $\mu_j$ remain in the $[0, 1]$ interval. Therefore, suitable saturation thresholds $\mu_{\min}$ and $\mu_{\max}$ must be introduced to keep the $\mu_j$'s in the mentioned interval. Accordingly, the upper bound $\mu_{\max}$ is typically set to 1. On the other hand, the lower bound $\mu_{\min}$ is more conveniently set to a small nonzero value, so that the probability that a regressor can be picked out at any iteration can never go to 0. This prevents the algorithm from completely excluding the regressor (an event which can sometimes occur at the early stages of the procedure). Finally, parameter $\varepsilon$ defines the stopping criterion. The algorithm stops when the distribution converges to a limit distribution. We consider that this has occurred for all practical purposes if the values of $\boldsymbol{\mu}$ do not differ more than $\varepsilon/2$ from 0 or 1, where $\varepsilon$ is a suitably small value.

The step size is a crucial tuning parameter, since too small a value would lead to an extremely slow convergence rate, while too large a value might cause instability. For this reason we here employ an adaptive step size solution, in which we set

$$\gamma = \frac{1}{10(\mathcal{J}_{\max} - \bar{\mathcal{J}}) + 0.1}, \tag{24}$$

where $\mathcal{J}_{\max}$ represents the performance of the best model structure among all models extracted at the current iteration, while $\bar{\mathcal{J}}$ is the average value of model performances. The rationale underlying the adaptive step size (24) is as follows. If $\bar{\mathcal{J}}$ is far from the performance of the best model, $\gamma$ is kept small to take adequately into account the information dispersion in the considered population of models. Conversely, if $\bar{\mathcal{J}}$ is close to $\mathcal{J}_{\max}$, all extracted models have similar performance, and the parameter correction suggested based on this information is considered more reliable.

## 5 Simulation examples

In this section several simulation examples are discussed to show the effectiveness of the RaMSS algorithm. Consider the following five systems taken from the literature ($\mathcal{S}_1$ from [39], $\mathcal{S}_2$ from [10], $\mathcal{S}_3$ from [25], and $\mathcal{S}_4$ and $\mathcal{S}_5$ from [5]):

$$\begin{aligned}\mathcal{S}_1:\ & y(k) = -1.7y(k-1) - 0.8y(k-2) + u(k-1) \\ & \quad + 0.81u(k-2) + e(k), \\ & \text{with } u(k) \sim \mathrm{WUN}(-2, 2),\ e(k) \sim \mathrm{WGN}(0, 0.01)\end{aligned}$$

$$\begin{aligned}\mathcal{S}_2:\ & y(k) = 0.8y(k-1) + 0.4u(k-1) + 0.4u^2(k-1) \\ & \quad + 0.4u^3(k-1) + e(k), \\ & \text{with } u(k) \sim \mathrm{WGN}(0, 1),\ e(k) \sim \mathrm{WGN}\big(0, 0.33^2\big)\end{aligned}$$

$$\begin{aligned}\mathcal{S}_3:\ & y(k) = 0.2y^3(k-1) + 0.7y(k-1)\, u(k-1) \\ & \quad + 0.6u^2(k-2) - 0.7y(k-2)\, u^2(k-2) \\ & \quad - 0.5y(k-2) + e(k), \\ & \text{with } u(k) \sim \mathrm{WUN}(-1, 1),\ e(k) \sim \mathrm{WGN}(0, 0.01)\end{aligned}$$

$$\begin{aligned}\mathcal{S}_4:\ & y(k) = 0.7y(k-1)\, u(k-1) - 0.5y(k-2) \\ & \quad + 0.6u^2(k-2) - 0.7y(k-2)\, u^2(k-2) \\ & \quad + e(k), \\ & \text{with } u(k) \sim \mathrm{WUN}(-1, 1),\ e(k) \sim \mathrm{WGN}(0, 0.004)\end{aligned}$$

$$\begin{aligned}\mathcal{S}_5:\ & y(k) = 0.7y(k-1)\, u(k-1) - 0.5y(k-2) \\ & \quad + 0.6u^2(k-2) - 0.7y(k-2)\, u^2(k-2) \\ & \quad + 0.2e(k-1) - 0.3u(k-1)\, e(k-2) \\ & \quad + e(k), \\ & \text{with } u(k) \sim \mathrm{WUN}(-1, 1),\ e(k) \sim \mathrm{WGN}(0, 0.02)\end{aligned}$$

In all examples a white noise input is assumed, either with a Gaussian distribution $\mathrm{WGN}\big(\eta, \sigma^2\big)$, where $\eta$ and $\sigma$ are the mean and standard deviation, or a uniform one $\mathrm{WUN}(a, b)$

in the interval $[a, b]$. In particular, system $\mathcal{S}_4$ [5] is directly used to compare the RaMSS algorithm with the RJMCMC approach.

All realizations of the listed systems are composed of 500 samples.

In the sequel, parameter $\alpha$ is always assumed 0, except in Section 5.3, where the effect of different choices for that parameters is analyzed.

### *5.1 RaMSS performance and comparative analysis*

A candidate regressor set was constructed using all monomials of the input and output signals with maximum lag equal to 4 and maximum degree equal to 3, for a total of $m = 165$ regressors. The RaMSS algorithm was executed with $N_p = 100$ (number of models to be generated at each iteration), and initial RIPs equal to $\mu_j = 1/m$, $j = 1, \ldots, m$. The parameter $K$ in the performance indices $\mathcal{J}_p$ and $\mathcal{J}_s$ was set to 1, and a purely predictive index has been used for model evaluation ($\alpha = 0$ in (12), *i.e.* $\mathcal{J} = \mathcal{J}_p$).

Figure 4 illustrates a typical run of the RaMSS algorithm with reference to system $\mathcal{S}_4$. Notice how the RIPs associated to the correct regressors are consistently increasing until they reach 1, while all other RIPs tend eventually to 0. Interestingly enough, a spurious parameter initially displays a larger RIP than any of the correct regressors, but the algorithm is capable of rejecting it, when the information concerning the correct regressors is sufficiently reinforced. The algorithm is capable of providing the correct model structure after approximately 20 iterations (all correct terms have $\mu_j > 0.5$, while for all the others $\mu_j < 0.5$). Finally, the average model size (AMS) in the population of models extracted at each iteration is essentially monotonically increasing to the correct value, implying that small, incomplete models are typically tested.

To evaluate the consistency of the randomized approach we repeated the MSS process 100 times for each of the 5 systems over the same data-set of 500 input/output pairs. The aggregate results of the RaMSS algorithm are summarized in Table 1. The MSS process resulted in the exact model structure in all cases, gathering the necessary information for the RIP update from less than 5000 models overall, which represents a tiny fraction of the total number of possible models ($2^{165}$). The final values for the AMS show that the algorithm converges to the actual number of regressors in the model. The small difference between the final AMS and the true model size is due to the application of a lower bound greater than 0 on the RIPs ($\mu_{\min} > 0$ in Algorithm 1). With the exception of system $\mathcal{S}_1$, the average size of the explored models is only slightly larger than the true model size. In the case of system $\mathcal{S}_1$, the algorithm explores models of larger size than the correct one for some time before converging to the true model, as indicated by the maximum AMS value in Table 1. A possible reason for this resides in the fact that the chosen family is largely over-parameterized with respect to the simple linear structure of system $\mathcal{S}_1$.

The FROE algorithm was tested as well on all five 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 process variance). Only the

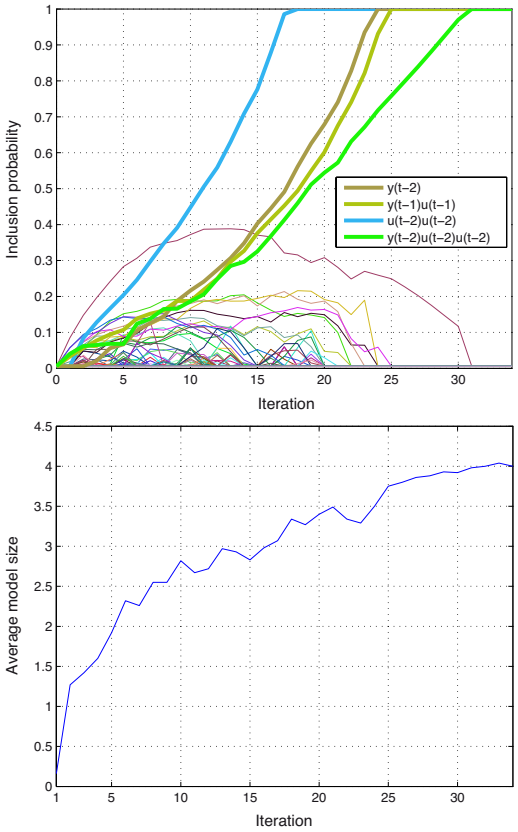


Figure 4. A typical evolution of the RaMSS algorithm for system $\mathcal{S}_4$: RIP (top) and AMS (bottom) evolution over 34 iterations.

model structure of system $\mathcal{S}_2$ is correctly identified. Two correct regressors ($y(k-1)$ and $u(k-1)$) are picked initially for system $\mathcal{S}_1$, but a wrong term selection ($y(k-3)$) at the $3^{rd}$ iteration jeopardizes the MSS process, which ends up in a model which misses two out of four terms. An incorrect model structure is returned for $\mathcal{S}_3$ as well, with a redundant constant term and $y(k-1)$ in place of $y^3(k-1)$. Finally, for both systems $\mathcal{S}_4$ and $\mathcal{S}_5$ a constant term is added by the FROE and the regressor $y(k-2)\,u^2(k-2)$ is not included.

Concerning system $\mathcal{S}_4$, we can also compare our results directly with those reported in [5], which documents a similar simulation experiment. Specifically, the RJMCMC approach retrieves the correct structure 7 times out of 10 runs, as opposed to the 100% correct selection obtained by the RaMSS algorithm. The RaMSS compares favorably also in terms of

Table 1
Average performance indicators of the RaMSS algorithm

| | $\mathcal{S}_1$ | $\mathcal{S}_2$ | $\mathcal{S}_3$ | $\mathcal{S}_4$ | $\mathcal{S}_5$ |
|---|---|---|---|---|---|
| Correct selection | 100% | 100% | 100% | 100% | 100% |
| # of Iterations | 42.65 | 36.00 | 55.09 | 37.19 | 36.13 |
| Elapsed Time [sec] | 51.3 | 57.1 | 80.4 | 58.4 | 57.1 |
| Maximum AMS | 5.83 | 4.01 | 5.25 | 4.04 | 4.06 |
| Final AMS | 4.02 | 4.00 | 5.11 | 4.02 | 4.01 |
| Explored Models | 4671 | 2428 | 4813 | 3260 | 3077 |

the computational effort. Specifically, 3260 model structures were evaluated on average, which amounts to about $1/6$ of the RJMCMC iterations, as reported in [5].

Notice that the RaMSS is capable of a $100\%$ correct performance even if system $\mathcal{S}_4$ is modified with the addition of a colored noise (see system $\mathcal{S}_5$).

### 5.2 *RaMSS performance under a slowly varying input signal*

As already observed earlier, the MSS process is generally sensitive to the excitation characteristics of the input, and particularly to slowly varying input signals. For this reason we performed an additional test on systems $\mathcal{S}_2$ and $\mathcal{S}_4$ using the low-pass filtered white noise signal (14) as input. More precisely, a Monte Carlo analysis of the performance of the FROE and RaMSS algorithms was conducted on 500 different data-sets (a single run of the RaMSS was carried out for each data-set). The FROE was applied with a $1\%$ threshold on the ERR for regressor acceptance. Also, a Student's t-test was performed on the returned model, and redundant terms were removed from the model, which was then re-estimated before evaluation. The RaMSS settings are the same used in the previous subsection. The quality of the selected models was finally evaluated both in terms of prediction and simulation performance through $\mathcal{J}_p$ and $\mathcal{J}_s$.

Figure 5 displays the performances of the models returned by both methods on systems $\mathcal{S}_4$ and $\mathcal{S}_2$. Concerning $\mathcal{S}_4$, on average the RaMSS improves little on the FROE in terms of prediction performance, but much more in simulation. It must be noted that the dispersion of the results is quite large, a clear sign of the high sensitivity of system $\mathcal{S}_4$ to the input signal realization. The results for system $\mathcal{S}_2$ are much more neatly clustered, showing that only the RaMSS algorithm is capable of obtaining models that perform accurately in simulation as well as prediction, while the FROE generally returns models that have scarce simulation performance (some are even unstable). This capability of the RaMSS is all the more surprising, given than parameter estimation is performed along the PEM framework and that $\alpha = 0$ in the performance evaluation index (during the MSS process).

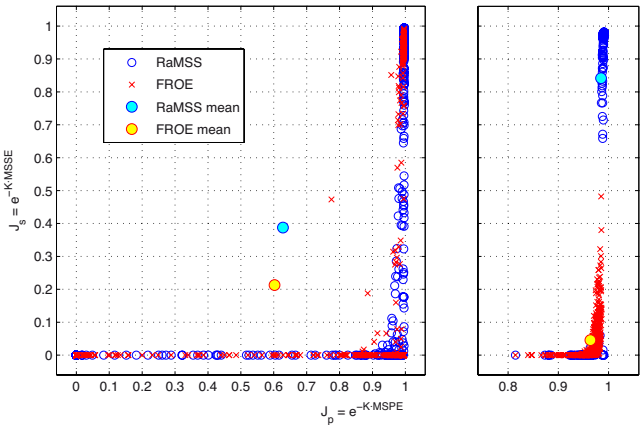


Figure 5. Comparison of the RaMSS and FROE algorithms on systems $\mathcal{S}_4$ (left) and $\mathcal{S}_2$ (right).

The iOFR method [17] was tested as well on system $\mathcal{S}_4$ to assess if its improved performance over the FROE can compare with the RaMSS. Figure 6 shows the performance and size distribution of all possible models obtained with the iOFR. Apparently, they are all dominated by the model identified by the RaMSS algorithm. This demonstrates that the difficulties experienced by the FROE in the MSS task are not limited to the initial choices but carry on throughout the whole process.

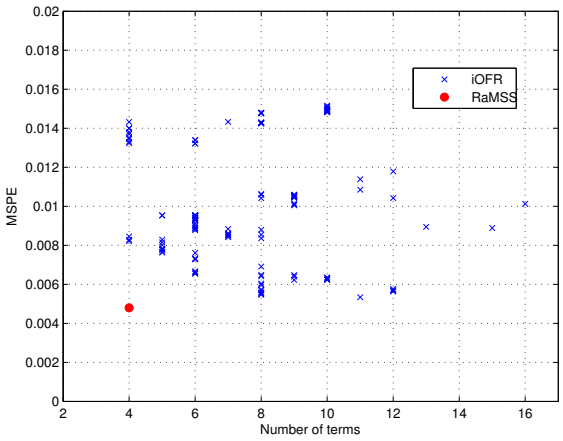


Figure 6. Performance distribution of the models identified with the iOFR algorithm, compared to the model returned by the RaMSS for system $\mathcal{S}_4$.

### 5.3 *Effects related to using $\mathcal{J}_s$ for model evaluation in the RaMSS*

Given the significant differences in terms of simulation performance, it is interesting to determine the possible improvements achievable by the RaMSS if $\alpha > 0$ is used for evaluating models in the MSS process. For this purpose, we ran another set of Monte Carlo simulations with the following setup. This time a single input realization was employed (again, using the low-pass filtered white noise signal (14)), but the RaMSS was carried out 500 times for each value of $\alpha$ in the set $\{0, 0.25, 0.5, 0.75, 1\}$. The prediction and simulation performances of the returned models are shown in Figures 7 and 8, for systems $\mathcal{S}_2$ and $\mathcal{S}_4$, respectively, and for increasing values of $\alpha$. The overlaid percentages quantify the number of trials in which the identified model has simulation performance less than or equal to 0.5. It appears that occasionally the RaMSS may return a model with scarce simulation performance, albeit with optimal predictive characteristics. However, the frequency of occurrence of such cases decreases with $\alpha$. Clearly, considering the simulation error in the performance index used for model evaluation purposes can be beneficial, in that it increases the ability of the algorithm to discard models with an apparently good performance, but that do not catch the underlying dynamics of the system. More in detail, the numerical simulations suggest that $\alpha$ should be set greater than 0.5 for appreciable results.

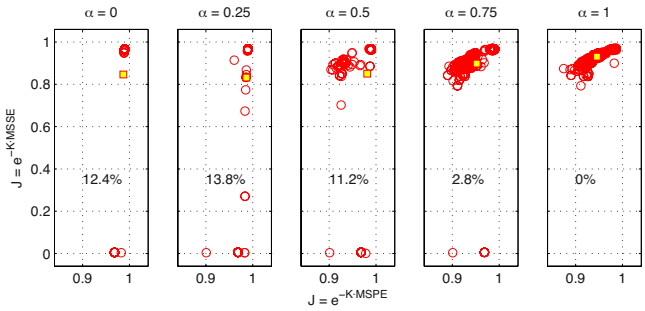


Figure 7. $\mathcal{S}_2$ identified model performances for increasing $\alpha$.

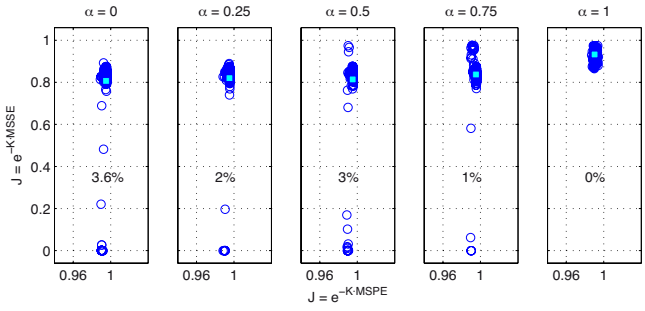


Figure 8. $\mathcal{S}_4$ identified model performances for increasing $\alpha$.

### 5.4 *Algorithm performance when* $\mathcal{P}_\Phi(f^\star) = 0$

One of the nice features of the presented approach is that the RaMSS is capable of extracting useful information on the model structure from *partially* correct models, *i.e.* models containing some of (but not all) the correct regressors among others. To emphasize this property, 1000 Monte Carlo simulations have been carried out on $\mathcal{S}_4$ (with a white noise input), where the RaMSS has been interrupted before convergence at the first extraction of the correct model (or any redundant one that includes it). In this way, no information is derived directly from the knowledge of the correct model structure. Figure 9 displays the maximum RIP value obtained for each regressor over all trials. There is an apparent gap between the values associated to the true regressors and those of the others, showing that the information extracted from partial models is overall enough to detect the true model.

### 5.5 *A real case study: the Wiener-Hammerstein benchmark*

While the RaMSS is designed for model structure selection it is also important to assess its performance when the model family does not include the exact system generating the data. For this purpose we considered a data-set concerning a nonlinear SISO electronic system with a Wiener-Hammerstein structure, originally documented in [38], which has been used as a benchmark nonlinear identification problem (see [33]). A data set of 188000 input-output data is available, the first 100000 for identification and the remaining for validation purposes. Following [28] we actually used a small portion of 2000 data only for model structure selection, then refined the parameter estimation over all the identification data and finally assessed the identified models over the validation data. The MSS task has been carried out over a candidate regressor set of 165 terms, including all monomials of the input and output signals with maximum lag equal to 4 and maximum degree equal to 3.

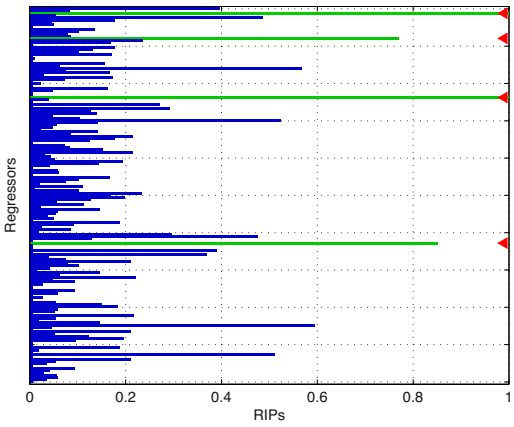


Figure 9. Monte Carlo analysis of the RaMSS performance when $\mathcal{P}_{\tilde{\Phi}}(f^\star) = 0$: maximum RIP value for each regressor (true regressors in green, other regressors in blue).

The results are reported in Table 2. Using $K = 5000$, the RaMSS achieves an MSPE of $1.3309 \cdot 10^{-6}$ and an MSSE of 0.0031 on the validation data, returning a model with 14 terms. For comparison purposes we tested the FROE and iOFR algorithms on the same problem limiting the model size to 14 terms. As evident from Table 2 the performances of the identified models are worse than for the RaMSS (the model returned by the FROE is even unstable). Actually, to reach the prediction performances of the RaMSS one has to allow the FROE to pick 144 terms (almost the entire regressor set), which reveals how such algorithm may have difficulties in discovering the appropriate terms. The iOFR cuts this figure to 117 terms, which is still a grossly over-parameterized model. Finally, note that the computational time associated to the RaMSS algorithm is much larger compared to the other two algorithms when applied with the same size limit. However, the iOFR requires more than double the time (534 s) to achieve comparable accuracy (with 117 terms). In any case, a computational time of 3.5 minutes appears largely affordable.

Table 2
Wiener-Hammerstein benchmark: validation performance of the identified models

| Algorithm | MSPE | MSSE | Number of terms | Elapsed time [s] |
|---|---|---|---|---|
| RaMSS | $1.3309 \cdot 10^{-6}$ | 0.0031 | 14 | 211 |
| FROE | $2.8967 \cdot 10^{-5}$ | $\infty$ | 14 | $< 1$ |
| iOFR | $1.9679 \cdot 10^{-6}$ | 0.0106 | 14 | 57 |

As a final remark, the iOFR performance can be greatly improved if a simulation-based criterion is used instead of the ERR for selecting regressors, similarly to what done in [29]. More specifically, at each iteration the regressor that most improves the MSSE is included in the model. The resulting MSS algorithm yields comparable results with the RaMSS,

both in terms of model size and accuracy. However, this comes at a great increase (almost a factor 10) in the computational time, due to the costly model simulations required.

## 6 CONCLUSIONS

A novel randomized algorithm denoted RaMSS is proposed for nonlinear system identification using the NARX model representation. More specifically, a probabilistic reformulation of the MSS process is given, that employs a model distribution defined in terms of the individual probabilities of each regressor to be included. Models extracted from the resulting distribution provide structural information that is exploited to update the mentioned regressor probabilities, thus adapting the model probability distribution. The local convergence of this iterative approach to the target limit distribution associated to the true model is proven in the paper, under mild conditions.

The performances of the RaMSS have been evaluated on five different systems using Monte Carlo analysis, the results showing that the RaMSS outperforms competitor approaches in terms of reliability of the selection. Compared to other randomized approaches it is also computationally more efficient. Moreover, it was shown that the algorithm is capable of retrieving the correct structure of the process model with a high percentage of success even with slowly varying inputs. Finally, the possibility of using a simulation error-oriented index for evaluation purposes has also been analyzed.

## References


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

[2] L.A. Aguirre, P.F. Donoso-Garcia, and R. Santos-Filho. Use of a priori information in the identification of global nonlinear models - a case study using a buck converter. *IEEE Transactions on Circuits and Systems, part I: Fund. Theory and Applications*, 47(7):1081–1085, 2000.

[3] Luis A. Aguirre, Bruno H. G. Barbosa, and Antonio P. Braga. Prediction and simulation errors in parameter estimation for nonlinear systems. *Mechanical Systems and Signal Processing*, 24(8):2855–2867, 2010.

[4] T. Baldacchino, S.R. Anderson, and V. Kadirkamanathan. Structure detection and parameter estimation for NARX models in a unified EM framework. *Automatica*, 48(5):857–865, 2012.

[5] T. Baldacchino, S.R. Anderson, and V. Kadirkamanathan. Computational system identification for Bayesian NARMAX modelling. *Automatica*, 49(9):2641–2651, 2013.

[6] S. A. Billings. *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*. Wiley, 2013.

[7] S. A. Billings and L. A. Aguirre. Effects of the sampling time on the dynamics and identification of nonlinear models. *International Journal of Bifurcation and Chaos*, 5(6):1541–1556, 1995.

[8] S.A. Billings, S. Chen, and M.J. Korenberg. Identification of MIMO nonlinear systems using a forward-regression orthogonal estimator. *International Journal of Control*, 49:2157–2189, 1989.

[9] O.M. Boaghe, S.A. Billings, L.M. Li, P.J. Fleming, and J. Liu. Time and frequency domain identification and analysis of a gas turbine engine. *Control Engineering Practice*, 10:1347–1356, 2002.

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

[11] S. Chen, X. Hong, and C. J. Harris. 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, 2003.

[12] N. Chiras, C. Evans, and D. Rees. Nonlinear gas turbine modeling using NARMAX structures. *IEEE Transactions on Instrumentation and Measurement*, 50:893–898, 2001.

[13] P. Connally, K. Li, and G.W. Irwin. Prediction-and simulation-error based perceptron training: solution space analysis and a novel combined training scheme. *Neurocomputing*, 70(4):819–827, 2007.

[14] A. Falsone, L. Piroddi, and M. Prandini. A novel randomized approach to nonlinear system identification. In *$53^{rd}$ IEEE Conference on Decision and Control*, pages 6516–6521, Los Angeles, USA, 2014.

[15] M. Farina and L. Piroddi. Identification of polynomial input/output recursive models with simulation error minimisation methods. *International Journal of Systems Science*, 43(2):319–333, 2012.

[16] S.A. Van De Geer. Least squares estimation. *Encyclopedia of Statistics in Behavioral Science*, 2:1041–1045, 2005.

[17] Y. Guo, L. Z. Guo, S. A. Billings, and H. L. Wei. An iterative orthogonal forward regression algorithm. *International Journal of Systems Science*, 46:776–789, 2015.

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

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

[20] M. Korenberg, S.A. Billings, Y.P. Liu, and P.J. McIlroy. Orthogonal parameter estimation algorithm for non-linear stochastic systems. *International Journal of Control*, 48(1):193–210, 1988.

[21] I.J. Leontaritis and S.A. Billings. Input-output parametric models for non-linear systems part I: deterministic non-linear systems. *International Journal of Control*, 41(2):303–328, 1985.

[22] A. Leva and L. Piroddi. NARX-based technique for the modeling of magneto-rheological damping devices. *Smart Materials and Structures*, 11:79–88, 2002.

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

[24] C.H. Loh and J.Y. Duh. Analysis of nonlinear system using NARMA models. *JSCE – Structural Engineering/Earthquake Engineering*, 13:11–21, 1996.

[25] K.Z. Mao and S.A. Billings. Algorithms for minimal model structure detection in nonlinear dynamic system identification. *International journal of control*, 68(2):311–330, 1997.

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

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

[28] L. Piroddi, M. Farina, and M. Lovera. Black box model identification of nonlinear input–output models: A Wiener–Hammerstein benchmark. *Control Engineering Practice*, 20(11):1109–1118, 2012.

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

[30] L. Piroddi and W. Spinelli. A pruning method for the identification of polynomial NARMAX models. In *$13^{th}$ IFAC Symposium on System Identification*, pages 1108–1113, Rotterdam, The Netherlands, August 27-28 2013.

[31] M. Pottmann, H. Unbehauen, and D.E. Seborg. Application of a general multi-model approach for identification of highly nonlinear processes – a case study. *International Journal of Control*, 57:97–120, 1993.

[32] K. Rodriguez-Vazquez, C. M. Fonseca, and P. J. Fleming. 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, July 2004.

[33] J. Schoukens, J. Suykens, and L. Ljung. Wiener-Hammerstein benchmark. In $15^{th}$ *IFAC Symposium on System Identification*, Saint-Malo, France, July 6-8 2009.

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

[35] T. Söderström and P. Stoica. *System identification*. Prentice-Hall, London, 1989.

[36] W. Spinelli, L. Piroddi, and K. Li. Nonlinear modeling of $NO_x$ emission in a coal-fired power generation plant. In *43rd IEEE Conference on Decision and Control*, pages 3850–3855, Atlantis, Paradise Island, Bahamas, 2004.

[37] R. Tempo, G. Calafiore, and F. Dabbene. *Randomized Algorithms for Analysis and Control of Uncertain Systems: with Applications*. Springer, 2004.

[38] G. Vandersteen. *Identification of linear and nonlinear systems in an errors-in-variables least squares and total least squares framework*. PhD thesis, Vrije Universiteit Brussel, 1997.

[39] H.-L. Wei and S.A. Billings. 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, 2008.

## A Theorem proofs

### *A.1 Proof of Theorem 1*

Consider first a regressor $\varphi_j \in f^\star$. Then, the following lower bound can be determined for $\mathcal{I}_j$, defined in (22):

$$\mathcal{I}_j \geq \mathcal{J}(f^\star)\mathcal{P}_\Phi(f^\star) - \overline{\mathcal{J}}_j, \tag{A.1}$$

where $\overline{\mathcal{J}}_j = \max_{\tilde{f}\in\tilde{\mathcal{F}}:\,\varphi_j\notin\tilde{f}} \mathcal{J}(\tilde{f})$. Indeed, notice that both expected values in (22) are positive, since $\mathcal{J}(\tilde{f}) > 0, \forall \tilde{f} \in \tilde{\mathcal{F}}$. Also, they can be bounded as explained in the following. As for the first term in (22),

$$\mathbb{E}[\mathcal{J}(\tilde{\Phi})|\varphi_j \in \tilde{\Phi}] = \sum_{\tilde{f}\in\tilde{\mathcal{F}}:\,\varphi_j\in\tilde{f}} \mathcal{J}(\tilde{f})\frac{\mathcal{P}_{\tilde{\Phi}}(\tilde{f})}{\tilde{\mu}_j},$$

where $\tilde{\mu}_j = \sum_{\tilde{f}\in\tilde{\mathcal{F}}:\,\varphi_j\in\tilde{f}} \mathcal{P}_{\tilde{\Phi}}(\tilde{f})$. The RHS of the previous expression can be bounded as follows:

$$\sum_{\tilde{f}\in\tilde{\mathcal{F}}:\,\varphi_j\in\tilde{f}} \mathcal{J}(\tilde{f})\frac{\mathcal{P}_{\tilde{\Phi}}(\tilde{f})}{\tilde{\mu}_j} \geq \mathcal{J}(f^\star)\frac{\mathcal{P}_{\tilde{\Phi}}(f^\star)}{\tilde{\mu}_j} \geq \mathcal{J}(f^\star)\mathcal{P}_\Phi(f^\star),$$

where the second inequality follows from (21) and upon observing that $0 < \tilde{\mu}_j \leq 1$. This finally leads to:

$$\mathbb{E}[\mathcal{J}(\tilde{\Phi})|\varphi_j \in \tilde{\Phi}] \geq \mathcal{J}(f^\star)\mathcal{P}_\Phi(f^\star). \tag{A.2}$$

On the other hand, the second term of (22) satisfies the following inequality:

$$\mathbb{E}[\mathcal{J}(\tilde{\Phi})|\varphi_j \notin \tilde{\Phi}] \leq \overline{\mathcal{J}}_j. \tag{A.3}$$

Applying the bounds (A.2) and (A.3) in (22), one obtains (A.1).

A similar reasoning applies to the case where $\varphi_j \notin f^\star$, leading to the following bound:

$$\mathcal{I}_j \leq \tilde{\mathcal{J}}_j - \mathcal{J}(f^\star)\mathcal{P}_\Phi(f^\star), \tag{A.4}$$

where $\tilde{\mathcal{J}}_j = \max_{\tilde{f}\in\tilde{\mathcal{F}}:\,\varphi_j\in\tilde{f}} \mathcal{J}(\tilde{f})$. Indeed, the first term in (22) can be bounded from above as follows:

$$\mathbb{E}[\mathcal{J}(\tilde{\Phi})|\varphi_j \in \tilde{\Phi}] \leq \tilde{\mathcal{J}}_j. \tag{A.5}$$

The second term can be reformulated as

$$\mathbb{E}[\mathcal{J}(\tilde{\Phi})|\varphi_j \notin \tilde{\Phi}] = \sum_{\tilde{f}\in\tilde{\mathcal{F}}:\,\varphi_j\notin\tilde{f}} \mathcal{J}(\tilde{f})\frac{\mathcal{P}_{\tilde{\Phi}}(\tilde{f})}{1-\tilde{\mu}_j} \geq \mathcal{J}(f^\star)\mathcal{P}_\Phi(f^\star). \tag{A.6}$$

Applying the bounds (A.5) and (A.6) in (22), one obtains (A.4).

Now, if we set

$$\delta > \max_{\tilde{f}\in\tilde{\mathcal{F}}\setminus\{f^\star\}} \frac{\mathcal{J}(\tilde{f})}{\mathcal{J}(f^\star)}$$

and $\mathcal{P}_\Phi(f^\star) \geq \delta$, we have that $\mathcal{I}_j > 0$ if $\varphi_j \in f^\star$, from bound (A.1). Conversely, $\mathcal{I}_j < 0$ if $\varphi_j \notin f^\star$, from bound (A.4).

### *A.2 Proof of Theorem 2*

Theorem 2 is proved upon showing that, if $\mathcal{P}_\Phi(f^\star)$ is sufficiently high, the execution of an algorithm iteration has the effect of rewarding correct regressors and discouraging wrong ones, thereby increasing $\mathcal{P}_\Phi(f^\star)$ further.

Let $\mathcal{P}_{\Phi(i)}$ denote the probability distribution induced by $\boldsymbol{\mu}(i)$, by way of (19), where $i$ is the iteration index in (23). Then, if $\mathcal{P}_{\Phi(i)}(f^\star) \geq \delta$, where $\delta$ satisfies the conditions of Theorem 1:

$$\begin{array}{ll} \mathcal{I}_j > 0 & \forall j : \varphi_j \in f^\star \\ \mathcal{I}_j < 0 & \forall j : \varphi_j \notin f^\star \end{array}$$

one obtains that:

$$\begin{aligned} \mu_j(i+1) &= \mu_j(i) + \gamma \mathcal{I}_j > \mu_j(i) \qquad \forall j : \varphi_j \in f^\star \\ \mu_j(i+1) &= \mu_j(i) + \gamma \mathcal{I}_j < \mu_j(i) \qquad \forall j : \varphi_j \notin f^\star. \end{aligned}$$

Now, recalling that

$$\mathcal{P}_{\Phi(i+1)}(f^\star) = \prod_{j:\varphi_j \in f^\star} \mu_j(i+1) \prod_{j:\varphi_j \notin f^\star} (1 - \mu_j(i+1)),$$

one obtains that

$$\mathcal{P}_{\Phi(i+1)}(f^\star) > \mathcal{P}_{\Phi(i)}(f^\star) \geq \delta.$$

Therefore, the repetitive application of (23) preserves the signs of $\mathcal{I}_j$, $j = 1, \ldots, m$, thus leading to the convergence of the vector of RIPs to the target limit distribution.