A novel randomized approach to nonlinear system identification

Alessandro Falsone, Luigi Piroddi, Maria Prandini

Abstract—Classical incremental approaches for the identification of polynomial NARX/NARMAX models often yield unsatisfactory results in terms of structure selection, which is crucial for model reliability over long-range prediction horizons. This paper embeds the nonlinear identification problem into a probabilistic framework and presents a novel randomized algorithm for structure selection. The approach is validated over different models by means of Monte Carlo simulations, and is shown to outperform competitor probabilistic methods in terms of both reliability and computational efficiency.

I. INTRODUCTION

System identification refers to the procedure of building a mathematical description of the dynamic behavior of a system from input-output data [1]. This topic is of outmost importance in control theory, given the widespread use of model-based approaches to control design.

The identification of nonlinear systems is particularly challenging, [2]. We are here mainly concerned with NARX/NARMAX models [3], which are recursive input-output models where the current value of the system output is obtained as a nonlinear functional expansion of regressors, i.e., lagged input and output (and possibly noise) terms.

A system identification problem typically involves two tasks: model structure identification and parameter estimation. The latter is usually carried out following the prediction error minimization (PEM) paradigm, which requires a precise matching of the model structure — including the noise contribution — to that of the underlying system in order to guarantee unbiased parameter estimates. This further emphasizes the role of model structure selection, which, however, is interleaved with the parameter estimation task, since different model structures can only be compared after their parameters have been estimated.

Model structure selection has received much attention in the literature, due to the fact that functional approximation using families of basis functions often leads to an exponential increase in the number of possible regressors to be considered for inclusion in the model, making it crucial to identify significant terms, [4], [5], [6], [7], [8]. Iterative incremental methods are typically employed that update the current model by adding or removing terms in a greedy fashion, optimizing an objective function at each step, [9], [10], [4], [5], [11], [6]. As a result, there is no guarantee of convergence to the global minimum.

Recently some new randomized approaches [12] have been presented to address the nonlinear identification problem. In [7] an approach based on the Expectation Maximization algorithm is presented, which uses the particle filter in order to handle nonlinearities and jointly perform structure and parameter identification. In [8], parameter identification and regressor selection is performed in a Bayesian framework that is suitable for describing uncertainty in both parameters and structure. Posterior distributions are inferred using the Reversible Jump Markov Chain Monte Carlo (RJMCMC) procedure. Though the introduction of random sampling typically requires many iterations, which makes randomized approaches less computationally attractive than incremental methods in terms of speed, it should also favor convergence to the global minimum, [8].

In this work we introduce a novel iterative randomized algorithm for the identification of nonlinear systems. The approach is described with reference to the NARX model class, although the extension to the NARMAX case can be envisaged. Each regressor is associated with a Bernoulli distribution describing the probability that the regressor is present in the “true” model. At each iteration, the algorithm generates a set of models, each of which is obtained by independently extracting a value from the Bernoulli distribution of each regressor: if the value extracted is 1, then the regressor is included in the model, whereas if the value extracted is 0, then the regressor is not included. The parameters of the extracted models are estimated and a statistical test is performed to remove redundant terms. Finally, the Bernoulli distribution of each regressor is updated based on the performances of the entire population of extracted models.

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 rest of the paper is organized as follows. Section II provides the basic framework and notation for nonlinear system identification of NARX models and briefly reviews the main approaches in the literature. The proposed method is illustrated in Section III and then tested in Section IV. Finally, some concluding remarks are drawn in Section V.

II. PRELIMINARIES

A. NARX system identification

The class of nonlinear auto-regressive systems with exogenous input (NARX), introduced in [3], is described by the following input/output recursive equation:

\[ y(k) = f(\boldsymbol{x}(k)) + e(k), \]
where \( x(k) = [y(k-1), \ldots, y(k-n_y), u(k-1), \ldots, u(k-n_u)] \), \( y(k) \) and \( u(k) \) are the output and input signals, respectively, \( n_y \) and \( n_u \) are suitable maximum lags, \( e(k) \) is a (white) noise signal, and \( f(\cdot) \) is an unknown nonlinear function to be identified.

In NARX system identification, the goal is to find an estimator \( \hat{f} \) for function \( f \) based on an available data set of input/output measurements. Provided that \( f \) is continuous, many types of nonlinear functional expansions can be chosen for approximating \( f \), such as piecewise linear models, rational polynomial models, radial basis function or sigmoidal neural networks, etc. [2], [5]. All these functional expansions are \textit{universal approximators}, i.e., they can represent a given nonlinear continuous function to an arbitrary level of accuracy, provided they are endowed with sufficient degrees of freedom.

A nonlinear function can be conveniently approximated by a linear combination of nonlinear basis functions, [2], [5]:

\[
\hat{f}(x(k), \vartheta) = \sum_{j=1}^{m} \vartheta_j \varphi_j(x(k)),
\]

where \( \varphi_j(x(k)) \), with \( j = 1, \ldots, m \), are known nonlinear basis functions, \( \vartheta_j \) are their unknown coefficients, and \( m \) is the total number of basis functions employed. A popular choice consists in adopting polynomials as basis functions, thus setting

\[
\varphi_j(x(k)) = y(k-d_{j1}) \cdots y(k-d_{j_{\ell_j}}) \cdot u(k-d_{j_{\ell_j}+1}) \cdots u(k-d_{jM}),
\]

with \( d_{j1}, \ldots, d_{j\ell} \in \mathbb{N} \) and \( 0 \leq \ell_j \leq M \), if \( M \) is the maximum degree of the polynomial expansion, whereas \( \ell = 0 \) stands for the zero-degree term (i.e., \( \varphi_j = 1 \)). As opposed to more general NARMAX models, the NARX class does not include a noise model, i.e., its regressors include only input/output lagged terms. This simplifying assumption is often justified in practice by the increased modeling capabilities of nonlinear models compared to linear ones, which deemphasizes the role of the noise model to account for unmodeled dynamics. Moreover, in practical situations it is very difficult to make reasonable \textit{a priori} assumptions on the noise model structure, especially in the nonlinear case, [6]. Polynomial NARX models provide a fairly general and flexible class of models. Also, compared with other classes such as neural networks or support vector machines, they are typically more compact and give a better insight on the nonlinearities underlying the system dynamics.

Defining the regressor vector \( \varphi(k) = [\varphi_1(x(k)), \ldots, \varphi_m(x(k))]^T \) and the parameter vector \( \vartheta = [\vartheta_1, \ldots, \vartheta_m]^T \), and substituting (2) in (1), one obtains the linear regression model:

\[
\hat{y}(k) = \varphi^T(k) \vartheta,
\]

Linear-in-the-parameter models 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, [5].

In the following subsections two algorithms for NARX identification are briefly reviewed, namely the Forward Regression Orthogonal Estimator (FROE) and a randomized algorithm based on the RJMCMC approach.

B. The Forward Regression Orthogonal Estimator

The FROE is an iterative procedure for nonlinear systems identification, which is based on the orthogonal LS (OLS) algorithm for parameter estimation. OLS is a slightly modified version of the common LS estimation procedure, which operates a decoupling of the regressors through the orthogonalization of the regression matrix. The structure selection is driven by the \textit{Error Reduction Ratio} (ERR) coefficient that can be easily computed using the OLS procedure, and which accounts for the improvement in the model prediction performance guaranteed by the inclusion of an additional term in the model. The model performance is measured in terms of the mean squared prediction error (MSPE):

\[
\text{MSPE} = \frac{1}{N} \sum_{k=1}^{N} (y(k) - \hat{y}(k))^2.
\]

At each iteration of the FROE algorithm, the regressor \( \varphi_j \) with the highest ERR (i.e., the regressor whose inclusion most improves the MSPE) is added to the current model, thus progressively decreasing the prediction error. In fact, the relative importance of a regressor changes as a new term is introduced in the model [4], and a regressor included at an early stage of the procedure may turn out to be irrelevant when the algorithm identifies the complete model structure, [6]. Testing included regressors for exclusion may partially mitigate the problem [13]. However, the whole model building procedure can be jeopardized by the fact that the regressor inclusion policy is based on a local estimation of the regressor importance, which can be quite different depending on the specific model structure considered. More in general, 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, [9], [5], [4], [8]. New approaches, where the significance of a term is established in absolute, global terms, over a whole range of models, are highly desirable.

C. The RJMCMC approach

A new Bayesian approach to nonlinear model identification has been recently presented in [8], which aims at performing regression selection alongside parameter estimation. The Bayesian framework is also convenient in that it allows to quantify the uncertainty in the model structure determination and the parameter identification task, a feature that is absent in all deterministic approaches.

The RJMCMC procedure extends the Metropolis-Hastings (MH) algorithm to account for “jumps” in the parameters dimension. It is an iterative algorithm, which at each iteration randomly performs one out of three possible actions:
i) **birth move**: a new regressor is randomly selected from predefined pool 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 real model has more or fewer terms compared to 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 in previous iterations. In the NARMAX case this proposal-acceptance mechanism is repeated twice: first for the process model and then for the noise model. After a burn-in period the algorithm should 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, both the regressors and the parameters.

The joint identification of both structure and parameters, while appealing in principle, turns out to be problematic, both the regressors and the parameters. The joint optimization problem formulated above can hardly be solved by an exhaustive enumeration approach, given that the number of possible models is exponential in the number of regressors \( m \) (which rapidly increases with \( n_u, n_y \), and the polynomial degree \( M \)), a problem often referred to as “curse of dimensionality”. More conveniently, it can be addressed by adopting a probabilistic reformulation. Let \( P \) denote a probability distribution over \( \tilde{F} \) and \( \Phi \) the random variable that corresponds to an extraction of a model in \( \tilde{F} \) according to \( P \).

\[
E_P[J] = \sum_{\tilde{f} \in \tilde{F}} J(\tilde{f}) P(\Phi = \tilde{f}).
\]

Expression (8) is a convex combination of the performance indices of all models in \( \tilde{F} \). If we let \( P \) vary over all possible distributions on \( \tilde{F} \), the maximum value of (8) is obtained by making all probability mass concentrate on the “true” model. Thus, the optimization problem

\[
\max_P E_P[J],
\]

provides the same solution of problem (6).

A key feature of the proposed algorithm is the adopted parametrization for \( P \). We associate to each regressor \( \varphi_j(x(k)) \) a (Bernoulli) random variable \( \rho_j \sim \text{Be}(\mu_j) \), whose possible outcomes are 1 with probability \( \mu_j \) and 0 with probability \( 1 - \mu_j \), with \( \mu_j \in [0,1] \) and \( j = 1, \ldots, m \).

A model \( f \in \tilde{F} \) is obtained by extracting independently a value for each \( \rho_j, j = 1, \ldots, m \). Regressor \( \varphi_j \) is present in model \( \tilde{f} \) only if \( \rho_j = 1 \). We will refer to \( \mu = [\mu_1, \ldots, \mu_m] \) as the vector of Regressor Inclusion Probabilities (RIPs). Each extracted model \( f \in \tilde{F} \) undergoes a statistical test, that allows to detect any redundant terms, which are then removed resulting in a model \( \tilde{f} \in \tilde{F} \). Accordingly, \( \rho_j \sim \text{Be}(\tilde{\mu}_j) \) will denote a random variable that is equal to 1 when regressor \( \varphi_j \) belongs to the non-redundant model \( \tilde{f} \) extracted according to the outlined procedure. Notice that it always holds that \( \tilde{\mu}_j \leq \mu_j \), since the statistical test can only result in the removal of some terms.

The aim of the proposed algorithm is to tune parameter vector \( \mu \) so that the resulting \( \tilde{\mu} = [\tilde{\mu}_1, \ldots, \tilde{\mu}_m] \) is a binary vector with 1 only in correspondence to the indices of the true regressors, hence ensuring that the probability mass of \( P \) is concentrated on the true model. Now, observe that the following expression holds for each \( j = 1, \ldots, m \):

\[
E_P[J] = \tilde{\mu}_j E_P[J|\tilde{\mu}_j = 1] + (1 - \tilde{\mu}_j) E_P[J|\tilde{\mu}_j = 0].
\]
Deriving (10) with respect to \( \tilde{\mu}_j \), one obtains
\[
\frac{\partial E_P[J]}{\partial \tilde{\mu}_j} = E_P[J|\tilde{\rho}_j = 1] - E_P[J|\tilde{\rho}_j = 0].
\] (11)

Expression (11), according to the discussion above, suggests setting \( \tilde{\mu}_j = 1 \) for those regressors that have \( \partial E_P[J]/\partial \tilde{\mu}_j > 0 \), and \( \tilde{\mu}_j = 0 \) otherwise. If we map these conditions back to the \( \mu \) parametrization, the former one implies that \( \mu_j = 1 \), while the latter is guaranteed if \( \mu_j = 0 \).

We can then use the information provided by (11) to update directly parameter \( \mu_j \) according to:
\[
\mu_j(i+1) = \mu_j(i) + \gamma \frac{\partial E_P[J]}{\partial \tilde{\mu}_j},
\] (12)
where the step size \( \gamma > 0 \) is a design parameter. This update rule should progressively refine the RIP distribution, thus shaping the \( \mathcal{P} \) distribution over \( \mathcal{F} \) so as to ultimately reduce it to a point mass distribution.

Unfortunately, the exact value of \( \partial E_P[J]/\partial \tilde{\mu}_j \) cannot be computed in practice, since it involves exploring all possible models. Nevertheless, one can obtain an estimate of such value by simply extracting a representative sample of the population. The procedure we propose in this paper prescribes to iteratively extract a given number of models \( f \in \mathcal{F} \) according to the probability distribution induced by the current values of \( \mu_1, \ldots, \mu_m \), and compute estimates \( E_P[J|\tilde{\rho}_j = 1] \) and \( \hat{E}_P[J|\tilde{\rho}_j = 0] \) on the population of extracted models.

Notice that expression (12) does not guarantee by itself that parameters \( \mu_j \) remain in the \([0, 1]\) interval. Therefore, suitable saturation thresholds must be introduced. In case \( E_P[J|\tilde{\rho}_j = 1] = E_P[J|\tilde{\rho}_j = 0] \), the algorithm cannot decide whether there is some advantage in including regressor \( \varphi_j \), so the corresponding RIP is left unchanged.

The step size \( \gamma \) 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:
\[
\gamma = \frac{1}{10(J_{\text{max}} - J) + 0.1},
\] (13)
where \( J_{\text{max}} \) represents the performance of the best model structure among all models extracted at the current iteration, while \( J \) is the average value of the model performances. The rationale underlying the adaptive step size (13) is as follows. If \( J \) is far from the performance of the best model, \( \gamma \) is kept small so as to progressively update the parameters while taking into account the information dispersion in the considered population of models. Conversely, if \( J \) is close to \( J_{\text{max}} \), many of the extracted models have the same performance, and the suggested parameter correction is considered more reliable.

In view of the discussion above, we can now outline the RnMSS algorithm. In the absence of other \textit{a priori} information an initial Bernoulli distribution is assumed with equal RIPs for all regressors, \( \mu_j = \mu_0 \), for \( j = 1, \ldots, m \). Then, at each iteration of the algorithm a set of \( N_p \) models is generated according to the following procedure. A value is extracted at random, independently, for each \( \rho_j, j = 1, \ldots, m \). Regressor \( \varphi_j \) is included in the model if the extracted value for \( \rho_j \) is 1. Once the structure of all \( N_p \) models has been defined, the parameter vector \( \vartheta \) associated to each model \( \mathcal{M}_p \) is estimated using LS. The resulting models \( f \in \mathcal{F} \) undergo a statistical test and non-redundant regressors are eliminated, thus obtaining a set of models \( f \in \mathcal{F} \), which are then evaluated in terms of performance index \( J \). Finally, vector \( \mu \) is modified according to the update equation (12). In other words, each individual RIP is updated based on the performances of the \( N_p \) observed models, and specifically increased if the average performance of the models that include that specific regressor is greater than the average performance of the remaining ones. The algorithm stops when a desired level of accuracy is met or when the distribution has converged.

B. Implementation aspects

Once the parameters \( \vartheta \) of a tentative model structure \( f \in \mathcal{F} \) have been estimated, a Student’s \( t \)-test is performed to determine whether or not a regressor is relevant for the current model. More precisely, an estimate of the variance \( \hat{\sigma}_j^2 \) of the LS solution \( \hat{\vartheta}_j \) can be obtained \textit{a posteriori} as \( \hat{\sigma}_j^2 \approx \hat{\sigma}_e^2 V_{jj} \), where \( V_{jj} \) is the \( j \)-th diagonal element of the information matrix \( V = (\sum_{k=1}^{N} \varphi(k) \varphi^T(k))^{-1} \). Notice that the estimated noise variance \( \hat{\sigma}_e^2 \) can be obtained directly by scaling the MSPE by a factor \( N/(N-\tau) \), where \( \tau \) is the number of regressors in the given structure. Now let \( t_{\alpha,N-\tau} \) be the \((1-\alpha)\text{th}\) percentile of a Student’s \( t \) distribution with \( N-\tau \) degrees of freedom. If \( 0 \) belongs to the \((1-\alpha)\cdot100\% \) confidence interval \( [\hat{\vartheta}_j - \hat{\sigma}_j t_{\alpha,N-\tau}; \hat{\vartheta}_j + \hat{\sigma}_j t_{\alpha,N-\tau}] \), \( \vartheta_j \) is not significantly different from zero at the \( \alpha \cdot 100\% \) level, and we cannot reject the null hypothesis \( \vartheta_j = 0 \). In the latter case, the corresponding regressor \( \varphi_j \) is considered to be statistically irrelevant for the given model.

Concerning the update step (12), notice that conditional expected values are estimated based on the \( N_p \) extracted models. Now, if \( \mu_j = 0 \) for some regressors, no extracted model will contain \( \varphi_j \). To avoid this, a lower bound is introduced for the RIPs, thus ensuring that any regressor can be extracted with a nonzero probability at any algorithm iteration.

A suitable initialization of the algorithm is obtained by setting equal small probabilities for each regressor, \( \mu_0 = 1/m \), thus encouraging the extraction of small models at the early steps of the algorithm. As a more general criterion, if any information is available regarding the estimated size \( \hat{m} \) of the “true” model, one can set \( \mu_0 = \hat{m}/m \) so as to force the algorithm to extract an initial set of models with an average number of terms equal to \( \hat{m} \).

Finally, as a stopping criterion one can monitor the convergence of the RIPs. If no RIP changes by more than a small prescribed factor \( \beta \) for \( K \) subsequent iterations, the algorithm stops. In the presented simulations \( \beta = 2 \cdot 10^{-3} \) and \( K = 3 \).
Consider the following systems taken from the literature: \[14\], \[11\], \[15\]...

<table>
<thead>
<tr>
<th>System</th>
<th>Maximum AMS</th>
<th>Final AMS</th>
<th>Explored Models</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S_1)</td>
<td>5.83 ± 0.01</td>
<td>4.02 ± 0.00</td>
<td>4671</td>
</tr>
<tr>
<td>(S_2)</td>
<td>5.25 ± 0.04</td>
<td>4.11 ± 0.02</td>
<td>2428</td>
</tr>
<tr>
<td>(S_3)</td>
<td>4.04 ± 0.02</td>
<td>3.26 ± 0.04</td>
<td>4813</td>
</tr>
<tr>
<td>(S_4)</td>
<td>4.01 ± 0.01</td>
<td>3.02 ± 0.02</td>
<td>3260</td>
</tr>
</tbody>
</table>

A family of \(m = 165\) model regressors was obtained setting \(n_y = n_u = 4\) and \(M = 3\). The initial estimate of the noise variance was assumed to be equal to \(\sigma^2 = 0.1\), and a 99% confidence interval was set for the LS parameter estimates. The number of models to be generated at each iteration was set to \(N_p = 100\) and the initial RIPs were set to \(\mu_j(0) = 1/m, \forall j\), to ensure an initial Average Model Size (AMS) of 1. The tuning parameter of the performance index \(J\) was set to \(K = 1\).

A typical run of the RaMSS algorithm produces the results reported in Figure 1. The convergence of the RIP parameters is depicted in the left part of Figure 1, while the right part illustrates the evolution of the AMS in the population of models extracted at each iteration. Notice that the RIPs associated to the correct regressors are consistently increasing until they reach 1, while all other RIPs tend eventually to 0. A spurious parameter has initially a larger RIP than the correct regressors, but the algorithm is capable of rejecting it, when the information concerning the correct regressors is sufficiently reinforced. A correct decision can be made regarding the model structure after 20 iterations only (all correct terms have \(\mu_j > 0.5\), while for all the others \(\mu_j < 0.5\)). Finally, the AMS is essentially monotonically increasing to the correct value.

To collect some statistics about the algorithm performance, we generated a single data-set (composed of 500 input/output pairs) for each system and we applied the RaMSS algorithm one hundred times over the same data. The distributions over the different runs of the number of iterations to convergence and of the maximum AMS (after the Student’s t-test) are reported in Figure 2. The aggregate results are summarized in Table I.

Each cell of the table reports the average value of the corresponding parameter. The RaMSS algorithm performed well in all cases and the number of explored models is mostly under 5000, a tiny fraction of the total number of possible models (which is greater than \(4.5 \cdot 10^{69}\)). The final values for the AMS show that the algorithm converges to the actual number of regressors in the model. Except for system \(S_1\) the algorithm tends to find the correct model without including unnecessary regressors, while with system \(S_1\) the algorithm explores redundant models for some time before convergence to the true model. This peculiarity may be due to the fact that the chosen family is largely over-parameterized with respect to the simple linear structure of system \(S_1\).

For comparison purposes we ran the FROE algorithm on all five systems with a value of \(\eta = 0.01\) as a threshold for regressor inclusion (the identification procedure stops if none of the regressors tested for inclusion at the current iteration has an ERR coefficient less than \(\eta\)). For system \(S_1\) the FROE algorithm correctly selects regressors \(y(k-1)\) and \(u(k-1)\) at the 1st and 2nd iterations respectively, but at the 3rd iteration it selects the wrong term \(y(k-3)\) and from that point on the structure selection is based on a wrong model. The identification ends without including two out of four terms. System \(S_2\) is correctly identified. The FROE algorithm returns a wrong model for \(S_3\), including a redundant constant term and \(y(k-1)\) in place of \(y^3(k-1)\). For systems \(S_4\) and \(S_5\) a constant term is added by the FROE and the regressor \(y(k-2)u^2(k-2)\) is not included.

Comparing the RaMSS performance on system \(S_4\) with the results in [8], while the RJMCMC approach retrieves the correct structure 7 times out of 10 runs, the RaMSS algorithm was able to identify the correct structure in all cases. Moreover, in terms of the computational effort, 3260 model structures were evaluated on average, which amounts to about 1/6 of the RJMCMC iterations, as reported in [8].

### V. CONCLUSIONS

A novel randomized algorithm was proposed for nonlinear system identification using the NARX model representation. Its performance was evaluated using Monte Carlo simulations over five different systems and the results show that it 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 in cases of a NARMAX model.

As pointed out in [4] using the prediction error for nonlinear system identification has some critical drawbacks. If the structures of the process and the noise models do not match those of the system, the parameter estimation is generally biased, [1], [6], which may indirectly affect the model structure selection as well. Furthermore, the use of the prediction error makes the nonlinear identification process more sensitive to the excitation characteristics of the input signal.
the input signal used in the identification experiment. This has been analyzed in [4], [6], with reference to the FROE algorithm: if a low frequency input signal is employed, the FROE will most likely include autoregressive terms in the model, regardless of their actual presence in the “real” model. Accordingly, future developments will focus on the use of the simulation error to evaluate model performance, so as to perform a reliable structure selection also in cases when the input signal is not fully exciting.

REFERENCES