Nonlinear system identification with model structure selection via distributed computation

Federico Bianchi, Alessandro Falsone, Maria Prandini and Luigi Piroddi

Abstract— **The problem of identifying a model of a system from input/output observations is typically formulated as an optimization problem over all available data that are collected by a central unit, in the same operating conditions. However, the massive diffusion of networked systems is changing this paradigm: data are collected separately by multiple agents and cannot be made available to some central unit due to, *e.g.,* privacy constraints. In this paper, we address this novel set-up and consider the case in which multiple agents are cooperatively aiming at identifying a model for a nonlinear system, by performing local computations on their private data sets. The problem of identifying the structure and parameters of the system has a mixed discrete and continuous nature, which hampers the application of classical distributed schemes. Here, we propose a method that overcomes this limit by adopting a probabilistic reformulation of the model structure selection problem.**

I. INTRODUCTION

The objective of system identification is to determine a mathematical representation (model) of a dynamical system from observed data. In particular, the identification of nonlinear systems has been extensively studied [15], using various classes of nonlinear models, such as Volterra series, block-oriented systems (*e.g.*, Hammerstein and Wiener models), and difference equation models.

A frequently adopted representation of the last class is the nonlinear auto-regressive with exogenous input (NARX) model [8], [9], which consists in a recursive input-output expression, where the current output is obtained by means of a nonlinear functional expansion involving lagged inputs and outputs. The functional expansion is often in the form of a polynomial, which yields a *linear-in-the-parameters* structure of the model that is particularly convenient for parameter estimation purposes. On the downside, the number of monomials in the expansion grows rapidly with the model order and nonlinearity degree, which motivates the interest in the problem of model structure selection (MSS), *i.e.*, the selection of the appropriate monomials to include in the model. This is a combinatorial problem that cannot generally be handled by exhaustive approaches, due to the excessive size of the solution space. Not surprisingly, most effort in the literature has been devoted to the development of heuristic search methods aimed at identifying a parsimonious model. Among all, the orthogonal forward regression (OFR) [3] method represents a milestone, and several variants of this method have been proposed (see, *e.g.*, [14], [7]). The OFR method adopts an incremental greedy scheme to progressively augment the model by adding the most promising term according to a model performance based criterion. A key feature of this approach is an orthogonalization technique which allows to evaluate independently the importance of each new candidate term, relative to the current model structure. Unfortunately, such relative measure of the importance of model terms can greatly vary depending on the model structure, which impairs the reliability of the selection process and may ultimately lead to suboptimal solutions. This issue has lead to new approaches based on the introduction of randomness in the search strategy and different criteria for the ranking of candidate model terms. In [2] the problems of MSS and the parameter estimation have been tackled jointly within a Bayesian framework where the probability distribution defined over models is dealt with a Reversible Jump Markov Chain Monte Carlo procedure. In [5] an iterative randomized algorithm (RaMSS) has been proposed in which independent Bernoulli variables are associated to the model terms, representing the probability that those terms are present in the true model structure. The distributions of such Bernoulli variables are tuned based on the information gathered from a population of extracted models, according to a randomized approach.

In system identification, data are typically collected by a single entity and the identification problem is formulated in terms of an optimization problem involving all data at once. However, when data are collected separately by multiple entities and cannot be made available to a central unit, due to, *e.g.,* privacy or communication constraints, the problem of identifying the same model based on separately available data sets arises. If the entities collecting data have computing and communication capabilities, one can formulate the problem in a distributed computation framework, where a network of agents are cooperatively solving the identification problem by local optimization.

In the literature, there are various examples of *linear-in-the-parameters* regression problems solved according to distributed approaches, as documented *e.g.* in [6], [12], [16]. However, none of the mentioned methods deals explicitly with MSS, which makes the optimization problem hard to solve because of the introduced discrete decision variables. In this respect, there are only a few attempts to solve the MSS problem in a distributed fashion. Recently, in [18], [10] the authors have proposed an extension of OFR-type algorithms to select a common-structure sparse model from multiple data sets, within the NARX modeling framework. The rationale behind this technique is to evaluate indepen-

Authors are with the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, via Ponzio 34/5, 20133 Milano (Italy) {name.surname}@polimi.it

dently the importance of each term in each data set, and then selecting that term which maximizes the (weighted) average importance. The selected term is hence removed from the candidate set, and the procedure is repeated. Once the common structure has been selected, the final parameter estimate is the (weighted) average of the least-square estimates obtained from all the data sets, which is however not guaranteed to be optimal according to any global criterion.

In this paper, we propose a novel method for NARX model identification in a distributed computation framework, where each agent i has its own data set and its own cost function $f_i : \mathbb{R}^n \to \mathbb{R}$ to assess the quality of the model in terms of its parametrization $\vartheta \in \mathbb{R}^n$ (which also encompasses the model structure, in that only the terms included in the model have nonzero parameters). The N agents aim at reaching consensus on a common value for ϑ that optimizes $\sum_{i=1}^N f_i(\vartheta)$, but without sharing their local data sets and costs. Unfortunately, standard privacy-preserving distributed schemes, such as those based on the subgradient, [13], and on proximal minimization, [11], are not applicable in our framework, because the MSS task introduces binary decision variables and makes the identification problem a mixed integer optimization problem. Inspired by [5], we reformulate the MSS problem in terms of the optimization of a common probability distribution over the space of all possible model structures, thus transforming the purely combinatorial MSS task into a continuous optimization problem. Based on this reformulation, we develop a distributed computation method to address both MSS and parameter estimation and show its effectiveness on some numerical examples.

The rest of the paper is organized as follows. Section II provides a formulation of the NARX multi-agent identification problem and sets the fundamental notation. The probabilistic reformulation of the MSS problem leading to a randomized approach for NARX model identification is discussed in Section III-A. The proposed method is developed in Section III-B and Section III-C, and its performance illustrated through the analysis of some numerical examples in Section IV. Some conclusive remarks end the paper.

II. PROBLEM STATEMENT

Consider a NARX model with scalar input u and output y, represented as

$$y(t) = g\left(x(t); \vartheta\right) + e(t), \tag{1}$$

where $x(t) = [y(t-1) \ \dots \ y(t-n_y) \ u(t-1) \ \dots \ u(t-n_u)]^T$ is a finite-dimensional vector containing lagged input and output values (n_y and n_u being suitable maximum lags), $e(t)$ is a stochastic process characterized as a sequence of i.i.d. zero mean random variables, and g is an unknown nonlinear function parameterized via a vector $\vartheta = [\vartheta_1 \ \dots \ \vartheta_n]^T$ of coefficients, with $\vartheta_j \in \mathbb{R}$. We represent the nonlinear function g as a polynomial functional expansion, whereby model (1) takes the form of a linear regression:

$$y(t) = \varphi(t)^T \vartheta + e(t), \tag{2}$$

where the regressors φ_j (collected in vector $\varphi = [\varphi_1 \ \dots \ \varphi_n]^T$) are monomials of the elements in $x(t)$ up to a given degree n_d.

We assume that N input-output data sets $\mathcal{D}_i = \{x^i(t), y^i(t)\}_{t=1}^{T_i}$ with length T_i, $i = 1, \dots, N$, have been collected from system (1) separately by N agents, possibly in different experimental set-ups. Let $\sigma_1^2, \dots, \sigma_N^2$ be the corresponding output process variances.

We can then formulate the identification of ϑ as the following optimization problem:

$$\min_{\vartheta} f(\vartheta) = \min_{\vartheta} \sum_{i=1}^{N} f_i(\vartheta), \tag{3}$$

with the function $f_i : \mathbb{R}^n \to \mathbb{R}$ defined as

$$f_i(\vartheta) = \frac{1}{\sigma_i^2} \sum_{t=1}^{T_i} \left(y^i(t) - \varphi^i(t)^T \vartheta\right)^2 + \lambda \|\vartheta\|_0 T_i, \tag{4}$$

where the first term accounts for the accuracy of the identified NARX model ϑ on the data set $\mathcal{D}_i$, and the second for the model complexity (as measured by the zero norm of ϑ, $\|\vartheta\|_0 = \text{card}\{\vartheta_j : \ \vartheta_j \neq 0\}$, *i.e.*, the number of non-zero entries of that vector, which corresponds to the actual model size). The latter is a regularization term introduced to prevent redundant terms from entering the model structure. In (4), parameter $\lambda > 0$ tunes the accuracy-complexity trade-off. We shall denote with $\vartheta^\star$ the optimal solution of (3) and by $f^\star$ the corresponding optimal cost $\sum_{i=1}^N f_i(\vartheta^\star)$.

Note that the ℓ_0 penalty term $\|\vartheta\|_0$ in (4) disrupts the continuity of the optimization problem and makes it a mixed integer program, since one has to count the non-zero terms in ϑ representing the model structure. A possible workaround is replacing the ℓ_0 norm of ϑ with its closest convex approximation $\|\vartheta\|_1 = \sum_{j=1}^n |\vartheta_j|$ (the ℓ_1 norm), as done in LASSO-based (least-absolute shrinkage and selection operator) approaches (see, *e.g.*, [12]). However, this would denature the model selection problem, which is intrinsically combinatorial, and would lead to not much accurate results in terms of structure selection [4].

In this work, we then address the NARX model identification problem (3) with local costs given by (4). Our aim is to introduce a distributed computation framework, where the agents do not share their data sets. This goal is challenging because of the mixed integer nature of the optimization program (3), which hampers the adoption of standard distributed schemes.

III. DISTRIBUTED SCHEME FOR NARX MODEL IDENTIFICATION VIA RANDOMIZATION

In this section we first introduce a variant of the Randomized Model Structure Selection (RaMSS) algorithm in [5] and then guide the reader through the derivation of a new distributed computation algorithm for NARX model identification.

A. A modified RaMSS algorithm

The RaMSS algorithm in [5] leverages a probabilistic reformulation of the MSS task to tackle its combinatorial nature. Differently from [5], we rely on the ℓ_0 penalty term in the cost function instead of an *a posteriori* statistical test to prune redundant regressors from the model.

Let $m = [m_1 \ \cdots \ m_n]^T$ be a vector collecting n discrete variables $m_j \in \{0, 1\}$, $j = 1, \ldots, n$, where m_j encodes the presence or absence of the term φ_j in the model, *i.e.* $m_j = 0$ if $\vartheta_j = 0$ and $m_j = 1$ otherwise. Then, vector $m \in \mathcal{M} \equiv \{0, 1\}^n$ encodes a model structure, and its performance can be measured as the cost associated to the best parametrization compatible with the model structure defined by m, *i.e.*,

$$J(m) = \min_{\vartheta \in \Theta_m} f(\vartheta) \tag{5}$$

where $\Theta_m = \{\vartheta : \vartheta_j = 0, \forall j : m_j = 0\}$. Let us further denote as

$$m^\star = \arg\min_{m \in \mathcal{M}} J(m) \quad \text{and} \quad J^\star = J(m^\star)$$

the best model structure and its (optimal) performance, which is assumed to be unique.

Now, let ϕ be a discrete random variable taking values in $\mathcal{M}$ according to some probability distribution $\mathcal{P}$. By definition of expectation, the average performance of ϕ is given by

$$\mathbb{E}_{\mathcal{P}}[J(\phi)] = \sum_{m \in \mathcal{M}} J(m)\mathcal{P}(\phi = m), \tag{6}$$

and, if we let $\mathcal{P}$ vary over all possible distributions over $\mathcal{M}$, we notice that the minimum value of (6) as a function of $\mathcal{P}$ is obtained by making all probability mass concentrate on the true model. Formally, denoting as

$$\mathcal{P}^\star = \arg\min_{\mathcal{P}} \ \mathbb{E}_{\mathcal{P}}[J(\phi)], \tag{7}$$

it holds that $\mathcal{P}^\star(\phi = m^\star) = 1$ and $\mathcal{P}^\star(\phi = m) = 0$ for all $m \in \mathcal{M} \setminus \{m^\star\}$.

As it stands, the optimization of (6) with respect to $\mathcal{P}$ is hardly solvable since it requires an exponential number of variables to completely characterize $\mathcal{P}$. A suitable parametrization of $\mathcal{P}$ is therefore needed to make the problem tractable. The idea in [5] is to assign to each m_j, $j = 1, \ldots, n$ a Bernoulli random variable γ_j, whose success probability $\mu_j \in [0, 1]$ represents the belief that m_j takes value 1, *i.e.* that the regressor φ_j is present in the model. Taking $\gamma_1, \ldots, \gamma_n$ to be independent from each other we have that

$$\mathcal{P}(\phi = m) = \prod_{j:m_j=1} \mu_j \prod_{j:m_j=0} (1 - \mu_j), \tag{8}$$

for any $m \in \mathcal{M}$. The reader should note that $\mathcal{P}^\star$ can be obtained from (8) setting $\mu_j = \mu_j^\star = m_j^\star$, $j = 1, \ldots, n$.

The RaMSS algorithm in [5] seeks $\mathcal{P}^\star$ by iteratively refining $\mu = [\mu_1 \ \cdots \ \mu_n]^T$ based on the following update rule

$$\mu(k+1) = \mu(k) - \alpha(k) \ \nabla_\mu \mathbb{E}_{\mathcal{P}}[J(\phi)]\big|_{\mu(k)} \tag{9}$$

where $\alpha(k)$ is a variable step-size, $\nabla_\mu = \left[\frac{\partial}{\partial \mu_1} \ \cdots \ \frac{\partial}{\partial \mu_n}\right]^T$,

$$\left.\frac{\partial \mathbb{E}_{\mathcal{P}}[J(\phi)]}{\partial \mu_j}\right|_{\mu(k)} = \mathbb{E}_{\mathcal{P}_k}[J(\phi)|\gamma_j = 1] - \mathbb{E}_{\mathcal{P}_k}[J(\phi)|\gamma_j = 0], \tag{10}$$

and $\mathcal{P}_k$ is given by (8) when $\mu = \mu(k)$. Note that a saturation shall be employed to ensure $\mu(k+1) \in [0, 1]^n$.

In [5], local convergence of the RaMSS algorithm to $\mathcal{P}^\star$ is proven.

B. An intuitive extension based on distributed computation

Given that (9) represents a gradient descent algorithm and considering the separable structure of $f(\vartheta)$ in (3), one might be tempted to derive a distributed version of the RaMSS update rule as follows

$$\begin{aligned} \bar{\mu}(k) &= \frac{1}{N}\sum_{i=1}^{N} \mu^i(k) \\ \mu^i(k+1) &= \bar{\mu}(k) - \alpha(k) \ \nabla_\mu \mathbb{E}_{\mathcal{P}}[J_i(\phi)]\big|_{\bar{\mu}(k)} \end{aligned} \tag{11}$$

where μ^i represents agent i's local estimate of the common μ vector and

$$\left.\frac{\partial \mathbb{E}_{\mathcal{P}}[J_i(\phi)]}{\partial \mu_j}\right|_{\bar{\mu}(k)} = \mathbb{E}_{\bar{\mathcal{P}}_k}[J_i(\phi)|\gamma_j = 1] - \mathbb{E}_{\bar{\mathcal{P}}_k}[J_i(\phi)|\gamma_j = 0],$$

with $J_i(m)$ defined as in (5) with $f_i(\vartheta)$ in place of $f(\vartheta)$ and $\bar{\mathcal{P}}_k$ given by (8) when $\mu = \bar{\mu}(k)$. Notice that (11) allows each agent to locally compute $\nabla_\mu \mathbb{E}_{\mathcal{P}}[J_i(\phi)]\big|_{\bar{\mu}(k)}$ based on its own dataset only. Briefly, in (11) at each step the local estimates $\mu^i(k)$ are averaged and each agent performs an iteration of rule (9) starting from the common average $\bar{\mu}(k)$.

Unfortunately, this intuitively simple strategy presents some drawbacks. Indeed, if we compute the average of $\mu^1(k+1), \ldots, \mu^n(k+1)$, we obtain

$$\begin{aligned} \bar{\mu}(k+1) &= \frac{1}{N}\sum_{i=1}^{N} \mu^i(k+1) \\ &= \frac{1}{N}\sum_{i=1}^{N} \bar{\mu}(k) - \alpha(k)\frac{1}{N}\sum_{i=1}^{N} \nabla_\mu \mathbb{E}_{\mathcal{P}}[J_i(\phi)]\big|_{\bar{\mu}(k)} \\ &= \bar{\mu}(k) - \frac{\alpha(k)}{N} \ \nabla_\mu \left[\sum_{i=1}^{N} \mathbb{E}_{\mathcal{P}}[J_i(\phi)]\right]\Bigg|_{\bar{\mu}(k)} \\ &= \bar{\mu}(k) - \frac{\alpha(k)}{N} \ \nabla_\mu \mathbb{E}_{\mathcal{P}} \left[\sum_{i=1}^{N} J_i(\phi)\right]\Bigg|_{\bar{\mu}(k)}, \end{aligned}$$

which means that the update rule (11) is minimizing the cost function $\mathbb{E}_{\mathcal{P}}\left[\sum_{i=1}^{N} J_i(\phi)\right]$, rather than $\mathbb{E}_{\mathcal{P}}[J(\phi)]$ as in (6). The two cost functions can actually be different, considering that for all $m \in \mathcal{M}$ the following holds:

$$\sum_{i=1}^{N} J_i(m) = \sum_{i=1}^{N} \min_{\vartheta^i \in \Theta_m} f_i(\vartheta^i) \leq \min_{\vartheta \in \Theta_m} \sum_{i=1}^{N} f_i(\vartheta) = J(m), \tag{12}$$

where we used ϑ^i with superscript i in the left hand side of (12) to emphasize the fact that the optimal parameterizations might be different from one agent to another.

Albeit showing that minimizing $\mathbb{E}_{\mathcal{P}}\left[\sum_{i=1}^{N} J_i(\phi)\right]$ does not necessarily go into the direction of minimizing $\mathbb{E}_{\mathcal{P}}[J(\phi)]$, (12) provides us with a precious intuition on how to modify (11) to correctly distribute equation (9). Indeed, if we could force all agents to agree on a common ϑ while evaluating $J_i(m)$ for any m, then we could turn (12) into an equality and correct iteration (11) to make it minimize $\mathbb{E}_{\mathcal{P}}[J(\phi)]$.

C. A distributed scheme for NARX model identification

Inspired by the proximal algorithm presented in [11], we propose to correct (11) modifying how the agents assess the performance of a generic model structure m, and specifically replacing $J_i(m)$ in (11) with

$$J_{i,k}(m) = \min_{\vartheta^i \in \Theta_m} f_i(\vartheta^i) + \frac{\rho(k)}{2}\|\vartheta^i - \bar{\vartheta}(k)\|_2^2, \qquad (13)$$

where the additional proximal term $\|\vartheta^i - \bar{\vartheta}(k)\|_2^2$ penalizes the distance of the parameter estimate ϑ^i computed by agent i from an average parameter vector $\bar{\vartheta}(k)$ and $\rho(k) > 0$ tunes the trade-off between the agent's local cost $f_i(\vartheta^i)$ and the disagreement among the agents.

The calculation of quantity $\bar{\vartheta}(k)$ requires some further explanation. This quantity represents a "common" parameter vector among the agents, obtained, *e.g.*, by averaging the current best parameterizations of the agents. However, there is not an obvious characterization of such best local parametrization, since each agent is endowed with a probability distribution over the model collection $\mathcal{M}$, as opposed to a specific model structure. To calculate $\bar{\vartheta}(k)$ we therefore associate first to each agent the single model structure $\hat{m}^i(k)$ that currently has the highest probability of being the optimal one, according to the local probability distribution. Then, we find for each agent the parametrization $\hat{\vartheta}^i(k)$ minimizing (13) with $m = \hat{m}^i(k)$. Finally, we average the minimizers $\hat{\vartheta}^i(k)$, $i = 1, \ldots, N$ to get $\bar{\vartheta}(k)$.

Accordingly, the proposed algorithm is based on the following iteration:

$$\begin{aligned}
\bar{\vartheta}(k) &= \frac{1}{N}\sum_{i=1}^{N} \hat{\vartheta}^i(k) \\
\bar{\mu}(k) &= \frac{1}{N}\sum_{i=1}^{N} \mu^i(k) \\
\mu^i(k+1) &= \bar{\mu}(k) - \alpha(k)\, \nabla_{\mu}\mathbb{E}_{\mathcal{P}}[J_{i,k}(\phi)]\big|_{\bar{\mu}(k)} \qquad (14) \\
\hat{m}^i(k+1) &= \arg\max_{m \in \mathcal{M}} \mathcal{P}^i_{k+1} \\
\hat{\vartheta}^i(k+1) &= \arg\min_{\vartheta^i \in \Theta_{\hat{m}^i(k+1)}} f_i(\vartheta^i) + \frac{\rho(k)}{2}\|\vartheta^i - \bar{\vartheta}(k)\|_2^2
\end{aligned}$$

where a saturation is applied to ensure $\mu^i_j(k+1) \in [0, 1]$, $j = 1, \ldots, n$. Note that the last three equations in (14) are calculated by the individual agents, based on their private data sets, whereas the computation of the first two equations requires either a broadcasting mechanism or the introduction of a central unit.

Intuitively, if we increase $\rho(k)$ at a proper rate, we can push the agents towards the (common) $\bar{\vartheta}(k)$ while they keep optimizing their own local objective functions. Once the $\hat{\vartheta}^i(k)$ are sufficiently close to each other, then it holds that

$$\sum_{i=1}^{N} J_i(m) \approx J(m),$$

which implies that the algorithm is actually minimizing $\mathbb{E}_{\mathcal{P}}[J(\phi)]$. The performance of the proposed algorithm is evaluated on some numerical examples in the following section.

IV. NUMERICAL EXAMPLES

We considered two different scenarios, where either all the agents have access to homogenously obtained data (*i.e.*, data resulting from experiments in similar conditions, with equal input and noise signal characterizations), or one of them has data from a different type of experiment on the unknown system.

Model selection was carried out over a candidate regressor set including all monomials with lags not larger than $n_y = n_u = 3$ and maximum degree $n_d = 3$, amounting to $n = 84$ terms and $2^{84} = 1.9 \cdot 10^{25}$ possible model structures. The initial μ^i_j, $\forall j, i$, were set to $\mu^i_j(0) = 1/n$. $\bar{\vartheta}_j(0)$ were set to zero, $\forall j$. The λ value in (4) has been suitably selected for each studied NARX system by using the *L-curve criterion*, which is a convenient graphical tool for displaying the trade-off between the size and the accuracy of the model as a function of the regularization parameter. The stepsize α in (14) is a decreasing function of time according to the following (often adopted) rule:

$$\alpha(k) = \beta/\sqrt{k},\ \beta > 0. \qquad (15)$$

We adopted an increasing factor $\rho(k) = 2k$ in (14), with k being the iteration index.

Notice that an exact computation of the expected values appearing in (14), as well as those in (9), cannot be obtained in practice, since it would require to consider exhaustively all the possible structures. We hence adopt a Monte Carlo approach to approximate such values with their sampled counterparts. More precisely, at each iteration each agent draws $N_p = 1000$ sample model structures from $\mathcal{P}$, evaluates them in terms of $J_{i,k}(m)$, and calculates the corresponding sampled averages.

To account for the randomization inherent in the algorithm, a Monte Carlo analysis has been carried out in all the experiments, by running the algorithm 100 times on the same data sets.

All the tests have been performed in MATLAB 2017a environment, on an HP ProBook 650 G1 CORE i7-4702MQ CPU @2.20 GHz with 8GB of RAM.

A. Experiment 1

We considered the following benchmark systems taken from the literature [17], [2], [4], [1]:

$$\begin{aligned}
S_1: \quad & y(t) = -1.7y(t-1) - 0.8y(t-2) + u(t-1) \\
& \quad +0.8u(t-2) + e(t), \\
& \text{with } u(t) \sim WUN(-2, 2),\ e(t) \sim WGN(0, 0.01)
\end{aligned}$$

S_2: $y(t) = 0.7y(t-1)u(t-1) - 0.5y(t-2) - 0.7y(t-2)u(t-2)^2 + 0.6u(t-2)^2 + e(t)$, with $u(t) \sim WUN(-1, 1)$, $e(t) \sim WGN(0, 0.04)$

S_3: $y(t) = 0.8y(t-1) + 0.4u(t-1) + 0.4u(t-1)^2 + 0.4u(t-1)^3 + e(t)$, with $u(t) \sim WGN(0, 0.333)$, $e(t) \sim WGN(0, 0.1)$

S_4: $y(t) = 0.25u(t-1) + 0.75y(t-2) - 0.2y(t-2)u(t-1) + e(t)$, with $u(t) \sim WGN(0, 0.25)$, $e(t) \sim WGN(0, 0.02)$

where $WGN(\eta, \sigma^2)$ is a White Gaussian Noise with mean η and variance σ^2, while $WUN(a, b)$ denotes a White Uniform Noise defined in the interval $[a, b]$. The employed λ values are: $\lambda_{S_1} = \lambda_{S_2} = \lambda_{S_3} = 0.01$, $\lambda_{S_4} = 0.001$. $\beta_{S_1} = \beta_{S_2} = \beta_{S_3} = 0.01$, $\beta_{S_4} = 0.1$, in (15). Four data sets of length 2000 have been generated, one for each agent.

TABLE I

EXPERIMENT 1: AVERAGE STATISTICS.

	S_1	S_2	S_3	S_4
Correct selection	100%	100%	100%	100%
# of Iterations	45.9	33.6	42.3	45.8
Elapsed Time [sec]	27.8	12.3	24.7	24.9
MSE on parameter estimate	3.9E-7	1.3E-5	2.2E-5	2.3E-4

TABLE II

EXPERIMENT 1: AVERAGE PARAMETER ESTIMATES.

S_1	True	-1.7	-0.8	1	0.8
	Estimated	-1.6993	-0.8002	0.9999	0.7990
S_2	True	0.7	-0.5	-0.7	0.6
	Estimated	0.6987	-0.4977	-0.7066	0.6007
S_3	True	0.8	0.4	0.4	0.4
	Estimated	0.7975	0.4069	0.4057	0.3983
S_4	True	0.25	0.75	-0.2	
	Estimated	0.2528	0.7414	-0.1751	

The proposed algorithm has been applied to this case, and the aggregated results are reported in Table I, where each cell reports the average value of the corresponding parameter. Specifically, the following statistics have been considered: the correctness of the structure selection, the elapsed time and number of iterations required to reach consensus, and the mean square error (MSE) of the parameter estimates. Table II displays the average parameter estimates.

The proposed algorithm performed well in all cases both regarding the model structure selection and the estimation of the parameters.

B. Experiment 2

In this experiment, we considered the following system:

S_5: $y(t) = 0.5y(t-1) + 0.8u(t-2) + 0.1u(t-1)^2 + e(t)$, with $e(t) \sim WGN(0, 0.01)$

Again, 4 data sets $\mathcal{D}_i$, $i = 1, \ldots, 4$ were collected, of length 5000 each, but this time the data are originated from different experimental set-ups. Specifically, in the first 3 experiments, $u(t) \sim WGN(0, 0.01)$, while in the last, $u(t) \sim WGN(0, 1)$.

The peculiarity of this example lies in the impossibility to identify the full model structure based on the data sets $\mathcal{D}_1$, $\mathcal{D}_2$, or $\mathcal{D}_3$, since the input amplitude is insufficient to excite the nonlinear dynamics in the model. On the other hand, the nonlinear dynamics is fully excited when $u(t) \sim WGN(0, 1)$ is employed (data set $\mathcal{D}_4$). To see this, check Table III, which reports the first eight terms selected by the OFR method, applied separately to each data set. The model terms are listed in the same order as they have been selected, which reflects their importance in the model. The terms in bold represent the final model structure, as selected according to the BIC criterion. Apparently, while the correct model structure is identified for $\mathcal{D}_4$ (albeit with a redundant term), only the linear sub-model is correctly selected in the other three cases, and the nonlinear missing term is not even among those suggested by the OFR immediately after the two correct regressors.

The explained difficulty of this example causes the failure of approaches such as that explained in [10] (denoted as PRESS-based OFR), as documented in Table IV, where the final model structure has been selected according to the Average-BIC criterion defined in [10]. Again, only the linear sub-part of the model has been correctly identified, while the nonlinear term has been masked by the prevailing linear data, and has been selected only as fourth term, to be rejected by the Average-BIC criterion.

Table V displays the aggregated results obtained by running our algorithm. The design parameters were set to $\lambda = 0.005$ and $\beta = 0.01$. Apparently, the algorithm proposed in the paper fruitfully combines the information gathered from all the data sets, ultimately leading to the identification of the correct model structure and accurate parameter estimates.

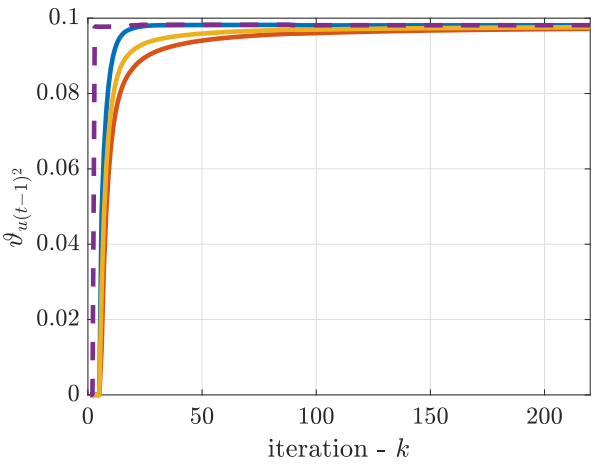

Fig. 1. Experiment 2: Nonlinear parameter estimation results with $\rho(k) = 2k$ for the proposed approach. The dashed curve is associated to agent 4.

Figure 1 shows the evolution for all agents of the parameter associated to the nonlinear term (denoted $\vartheta^i_{u(t-1)^2}$), during a single execution of the algorithm, using $\rho(k) = 2k$ in (14). As expected, the 4^{th} agent immediately recognizes

TABLE III

EXPERIMENT 2: MODEL STRUCTURE SELECTION RESULTS WITH THE OFR METHOD ON DIFFERENT DATA SETS.

Sel. order	$\mathcal{D}_1$	$\mathcal{D}_2$	$\mathcal{D}_3$	$\mathcal{D}_4$
1	$\boldsymbol{u(t-2)}$	$\boldsymbol{u(t-2)}$	$\boldsymbol{u(t-2)}$	$\boldsymbol{u(t-2)}$
2	$\boldsymbol{y(t-1)}$	$\boldsymbol{y(t-1)}$	$\boldsymbol{y(t-1)}$	$\boldsymbol{y(t-1)}$
3	$y(t-2)y(t-3)$	$y(t-3)$	$y(t-2)u(t-2)u(t-3)$	$\boldsymbol{u(t-1)^2}$
4	$y(t-3)u(t-3)$	$u(t-2)u(t-3)$	$u(t-2)^2u(t-3)$	$\boldsymbol{y(t-3)u(t-2)u(t-3)}$
5	$u(t-3)$	$y(t-3)u(t-2)u(t-3)$	$y(t-2)u(t-1)^2$	$y(t-1)u(t-2)^2$
6	$y(t-1)y(t-3)$	$u(t-1)^2u(t-2)$	$u(t-1)^2u(t-2)$	$u(t-1)^2u(t-2)$
7	$y(t-1)u(t-1)u(t-3)$	$y(t-1)u(t-1)$	$y(t-3)u(t-1)u(t-2)$	$y(t-3)u(t-1)$
8	$u(t-1)^2$	$u(t-1)u(t-3)$	$u(t-2)u(t-3)$	$u(t-1)^2u(t-3)$

TABLE IV

EXPERIMENT 2: RESULTS WITH THE PRESS-BASED OFR.

Sel. order	Model Term	Par. estimate
1	$\boldsymbol{u(t-2)}$	0.794143
2	$\boldsymbol{y(t-1)}$	0.508702
3	$y(t-2)u(t-2)u(t-3)$	–
4	$u(t-1)^2$	–

TABLE V

EXPERIMENT 2: AVERAGE STATISTICS OF THE PROPOSED APPROACH.

Correct selection	100%
# of Iterations	222.6
Elapsed Time [sec]	85.5
Selected model terms	$y(t-1),\ u(t-2),\ u(t-1)^2$
Parameter estimates	$0.5050,\ 0.7983,\ 0.0977$
MSE on parameter estimates	1.1E-5

the importance of the nonlinear term and provides a very accurate estimate of the corresponding parameter, while the others are slower, given that their data does not clearly emphasize the nonlinearity. However, they are still able to identify the presence of that term and its parameter value, which is very close to the true one.

V. CONCLUSIONS

A novel distributed scheme with model structure selection was proposed for nonlinear system identification using the NARX model representation. The proposed approach relies on the standing assumption that there are multiple data sets collected from several experiments which cannot be made centrally available, and hence the identification problem has to be solved by distributing the computation among agents. Its performance was evaluated using Monte Carlo simulations over two different scenarios. In both cases, the algorithm was capable of retrieving the correct structure and parameterization of the process model, in a computationally efficient way. Furthermore, it was shown that the presented method outperforms an OFR-based competitor in terms of reliability of the structure selection.

REFERENCES

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

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

[3] S. Billings, S. Chen, and M. Korenberg. Identification of MIMO non-linear systems using a forward-regression orthogonal estimator. *International Journal of Control*, 49(6):2157–2189, 1989.

[4] 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.

[5] A. Falsone, L. Piroddi, and M. Prandini. A randomized algorithm for nonlinear model structure selection. *Automatica*, 60:227–238, 2015.

[6] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin, and S. Madden. Distributed regression: an efficient framework for modeling sensor network data. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 1–10. ACM, 2004.

[7] 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.

[8] I. 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.

[9] I. Leontaritis and S. A. Billings. Input-output parametric models for non-linear systems part II: stochastic non-linear systems. *International journal of control*, 41(2):329–344, 1985.

[10] P. Li, H.-L. Wei, S. A. Billings, M. A. Balikhin, and R. Boynton. Nonlinear model identification from multiple data sets using an orthogonal forward search algorithm. *Journal of Computational and Nonlinear Dynamics*, 8(4):041001, 2013.

[11] K. Margellos, A. Falsone, S. Garatti, and M. Prandini. Distributed constrained optimization and consensus in uncertain networks via proximal minimization. *IEEE Transactions on Automatic Control*, 63(5):1372–1387, May 2018.

[12] G. Mateos, J. A. Bazerque, and G. B. Giannakis. Distributed sparse linear regression. *IEEE Transactions on Signal Processing*, 58(10):5262–5276, 2010.

[13] A. Nedic and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

[14] L. Piroddi. Simulation error minimization methods for NARX model identification. *International Journal of Modelling, Identification and Control*, 3(4):392–403, 2008.

[15] 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.

[16] S. Sundhar Ram, A. Nedić, and V. V. Veeravalli. A new class of distributed optimization algorithms: Application to regression of distributed data. *Optimization Methods and Software*, 27(1):71–88, 2012.

[17] H.-L. Wei and S. 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.

[18] H.-L. Wei and S. A. Billings. Improved model identification for non-linear systems using a random subsampling and multifold modelling (rsmm) approach. *International Journal of Control*, 82(1):27–42, 2009.