DOI: 10.48011/asba.v2i1.1440

Comparative Study of Three Output-Error Multi-Input Multi-Output Identification Methods

Renato A. L. de Andrade [*] **Péricles R. Barros** [**]

[*] *Researcher at Laboratório de Instrumentação Eletrônica e Controle (e-mail: renato.andrade@ee.ufcg.edu.br.)*
[**] *Universidade Federal de Campina Grande, Campina Grande, Brazil (e-mail: prbarros@dee.ufcg.edu.br*

Abstract: Multi-input multi-output (MIMO) systems have been a major concern for decades. However, due to the intrinsic complexity raised by the process interactions and optimization issues, MIMO approaches have not been developed as extensively as the single-input single-output ones. Recently, nevertheless, several algorithms have been proposed to address this problem, most of them based on recursive algorithms and many dependent on the assumption that the transfer function denominator polynomials are the same for all subsystems. In this article, an iterative least-squares-based algorithm, a pseudolinear regression and a Gauss-Newton optimization-based algorithm are proposed to provide a continuous-time output-error multi-input single-output model by means of iterative strategies. The numerical simulations indicate the iterative least-squares-based and the pseudo-linear regression algorithms have similar performances and generate more accurate and precise estimates than the Gauss-Newton one, which presented averages and standard deviations of the parameters ranging from twice as large to one order of magnitude higher than those of the other two algorithms.

Keywords: System identification; MISO models; output-error; least squares optimization.

1. INTRODUCTION

Multivariable systems are not uncommon in industrial processes, however, complexity rises significantly in identification procedures, both in terms of analysis and computational efficiency, as the number of subsystems increases in comparison to the single-input single-output (SISO) context, which occurs due to process interactions and optimization issues. Therefore, despite the SISO approach has been treated more extensively due to its simplicity, multi-input multi-output identification formulations have always been a major concern and some effort has been done on the past towards a multivariable identification approach (El-Sherief and Sinha, 1979) (Ljung, 1987).

In Rajapandiyan and Chidambaram (2012a) and Rajapandiyan and Chidambaram (2012b), the time-delay in the estimation and nonlinear optimization is presented as a rather faster option to obtain first- and second-order models with time delay in closed-loop than genetic algorithm-based methods. More recently, a set of algorithms have been developed to address the possibility of using diverse model structures, not including the time-delay as an estimated variable: FIR (Chan et al., 2019), ARX (Jin et al., 2014), ARMAX (Ding, 2014), ARARMAX (Liu et al., 2018), OE (He et al., 2019), OE with moving average (Zhang et al., 2011) and autoregressive error (Ding, 2018), BJ (Wang and Ding, 2016a) etc.. These algorithms originate from different methods, for instance, gradient-based algorithms (Ding, 2018), generalised instrumental variables (Söderström, 2012) or linear squares (Liu and Ding, 2013), and allow the addition of features which improve their performance, such as hierarchical iterative principle (Ma et al., 2019), coupled-least squares (Ding, 2013), data filtering (Wang and Ding, 2016b) and bias compensator (Chan et al., 2019). Part of them assumes a common denominator polynomial associated to every subsystem.

Most of the proposed algorithms are recursive ones, although some iterative options are available, for example, Ding (2014) and Ding (2018). This latter category is characterized by providing a progressive approach to the true parameters, however, differently to the recursive algorithms, they use the whole batch of data to obtain each new estimate.

Decomposition of a multi-input single-output (MISO) model into their SISO submodels is a strategy used in Yan et al. (2016) and He et al. (2019). In this article, an idea similar to that used in He et al. (2019) is followed, in which an output error MISO model is decomposed into a set of SISO models and their responses are accumulated to produce the overall output estimation that composes the prediction error. Nevertheless, instead of following the recursive approach, three iterative algorithms are proposed to estimate the parameter vector: an iterative fixed uncorrelated variables linear squares, which is proposed in this work, along with a pseudo-linear regression, and a Gauss-Newton-based optimization, all of them providing continuous-time models for open-loop plants. The three algorithms performances are compared by means of a numerical simulation example.

2. PROBLEM STATEMENT

Let $G(s)$ be the transfer matrix that describes the dynamics of a MISO continuous-time linear time-invariant (LTI) system with m inputs:

$$\begin{aligned} G(s) &= \begin{bmatrix} G_1(s) & G_2(s) & \dots & G_m(s) \end{bmatrix} \\ &= \begin{bmatrix} \dfrac{B_1(s)}{F_1(s)} & \dfrac{B_2(s)}{F_2(s)} & \cdots & \dfrac{B_m(s)}{F_m(s)} \end{bmatrix} \end{aligned} \tag{1}$$

$$B_i(s) = b_{i,1}s^{n-1} + b_{i,2}s^{n-2} + \ldots + b_{i,n} \tag{2}$$

$$F_i(s) = s^n + f_{i,1}s^{n-1} + f_{i,2}s^{n-2} + \ldots + f_{i,n}, \tag{3}$$

in which $i = 1,2,...,m$, and n is the model order.

If one decides to identify this system with an ARX model structure, one should make some decisions relative to the model order. For example, if the subsystems are of first order and it is made the choice of using n-th-order models to describe each input-output dynamics, since this is the structure which best models their relation, the overall model will lack degrees of freedom because there will be only n poles available to describe the dynamics of all subsystems. In order words, the model will be under-parametrized.

Instead, one may choose to provide more degrees of freedom to the model by using a higher-order model. For instance, if the system has three inputs and each submodel is of first order, a third-order model can be used in the identification so that one will have one pole available in the model for each pole in the true system. In this case, the overall model will be over-parametrized, since there will be two extra zeros and poles in the model for each input-output combination, providing the model with exceeding degrees of freedom. Ideally, this should not be an issue due to the fact that the extra zeros in each sub-model are expected to cancel with the poles that approximate the other sub-models dynamics. Whenever disturbances are present, however, such cancellations are not guaranteed.

Another possibility would be to apply a model reduction technique, for example, balanced truncation, on the high-order models so one may reduce each submodel to n-th-order ones.

In summary, there exists an issue regarding matching the model order with the true system one. One option to solve this problem would be the selection of a corresponding model structure to identify the true model, with each input-output combination modelled with independent poles, as shown in equation (1). The immediate obstacle to applying this structure is the fact that, in order to obtain the correct parameters for each sub-model, it would require the knowledge of the contributions from each input to the output, which are unknown. The goal in this article is, therefore, to solve this problem by proposing the identification of the multivariable system with a model structure which accounts for independent poles for each subsystem.

3. PROPOSED METHOD

Converting the transfer matrix of equation (1) into the continuous-time domain by means of equivalent transfer operators, the measured output $y(t)$ is given by:

$$y(t) = \sum_{i=1}^{m} G_i(\rho)u_i(t) + n(t) \tag{4}$$

where the signals $\{y_1(t),...,y_m(t)\}$ are the system responses to the inputs $u_1(t),...,u_m(t)$, respectively, which compound the output $y(t)$, $n(t)$ is white noise and ρ is the derivative operator.

The following output error model structure will be used:

$$y(t) = \sum_{i=1}^{m} \frac{B_i(\rho)}{F_i(\rho)}u_i(t) + e(t) \tag{5}$$

$$B_i(\rho) = b_{i,1}\rho^{n-1} + \ldots + b_{i,n} \tag{6}$$

$$F_i(\rho) = \rho^n + f_{i,1}\rho^{n-1} + \ldots + f_{i,n}. \tag{7}$$

where ρ is the derivative operator. It is important to highlight that not only $B_i(\rho)$ must be different for each subsystem but also $F_i(\rho)$ must be assigned as an independent polynomial for each input.

3.1 Initial Model

A MISO ARX model is estimated in order to provide an initial model for the system. It may be of any order but there exists a trade-off to make, which is discussed in the next section.

It is proved in Ljung and Wahlberg (1992) that, if both the number of samples and model order tend to infinity, and the former increases much faster than the latter, the identified rational model tends to the true system. Thus, it is suggested to start assuming this condition is approached, choosing an initial high-order model. Assuming the subsystems to be of order n, the initial model is suggested to be of order $n \times m$.

The initial ARX model structure is thus given by:

$$A(\rho;\eta)y(t) = \sum_{i=1}^{m} \Gamma(\rho;\eta)u_i(t) + e(t) \tag{8}$$

$$A(\rho;\eta) = \rho^{n\times m} + a_1\rho^{n\times m-1} + \ldots + a_{n\times m} \tag{9}$$

$$\Gamma_i(\rho;\eta) = \gamma_{i,1}\rho^{n\times m-1} + \ldots + \gamma_{i,n\times m}, \tag{10}$$

where $e(t)$ is the estimation error.

Applying the state-variable filter method (see section 4) and understanding that an operator ρ with a negative operator j is equivalent to an integration of the corresponding function, executed j times over the integration interval, the data vector is defined therefore as:

$$\begin{aligned} \phi(t) = [&-\rho^{-1}y(t) \quad \dots \quad -\rho^{-n\times m}y(t) \\ &\rho^{-1}u_1(t) \quad \dots \quad \rho^{-n\times m}u_1(t) \quad \dots \\ &\rho^{-1}u_m(t) \quad \dots \quad \rho^{-n\times m}u_m(t)]^T \end{aligned} \tag{11}$$

and the parameter vector:

$$\eta = [a_1 \ \dots \ a_{n\times m} \ \gamma_{1,1} \ \dots \ \gamma_{1,n\times m} \ \dots \ \gamma_{m,1} \ \dots \ \gamma_{m,n\times m}]^T, \tag{12}$$

the initial model predictor is given by:

$$\hat{y}^{IM}(t,\eta) = [1 - \rho^{-n\times m}A(\rho,\eta)]y(t) + \sum_{i=1}^{m}[\rho^{-n\times m}\Gamma_i(\rho,\eta)u_i(t)] \tag{13}$$

and the error to be minimized:

$$\begin{aligned} e(t,\eta) &= y(t) - \hat{y}^{IM}(t;\eta) \\ &= y(t) - \phi(t)^T\eta. \end{aligned} \tag{14}$$

The previous formulas are applied at each sample instant $t = t_r$, where $r = 1,2,\ldots,N$ are the data samples, and the parameter estimate is obtained by means of least squares as follows:

$$\hat{\eta} = \Big[\sum_{r=1}^{N} \phi(t_r)\phi(t_r)^T\Big]^{-1}\Big[\sum_{r=1}^{N} \phi(t_r)y(t_r)\Big] \tag{15}$$

Now that one has a model, one may proceed to the computation of a lower-order model, whose subsystems are of order n.

3.2 Iterative fixed uncorrelated variables linear squares

The first method consists in finding a parameter vector $\hat{\theta}$ that makes a vector $\zeta(t)$ — compounded by past values of $y(t)$, $u_i(t)$ and $\hat{\theta}$ — to be uncorrelated with the error vector, *i.e.* if Z^N is the experimental data, $\hat{\theta}$ is the solution for the equation:

$$f_N(\theta;Z^N) = \frac{1}{N}\sum_{r=1}^{N} \zeta(t_r;\theta)\varepsilon(t_r;\theta) = 0 \tag{16}$$

$$\varepsilon(t;\theta) = y(t) - \varphi^T(t;\theta)\theta. \tag{17}$$

The $\varphi(t,\theta)$ vector will comprise the inputs and their individual responses. Since the latter are not available, the simulated noise-free responses generated by the initial model which was assumed to tend to the true model will be used instead. The model structure now changes to the output error one described by equations (5) to (7) so that one may estimate m sets of transfer functions of order n, each one corresponding to an input and with its own and independent set of poles and zeros.

The simulated noise-free response is defined as:

$$w(t) = \sum_{i=1}^{m} w_i(t) \tag{18}$$

$$w_i(t) = \sum_{i=1}^{m} \frac{B_i(\rho)}{F_i(\rho)}u_i(t), \tag{19}$$

in which $w_i(t)(i = 1,2,...,m)$ are the contributions from each input to the output response. The output predictor for the output error model based on its past values is given as follows:

$$\hat{y}^{IV}(t;\theta) = \sum_{i=1}^{m} \hat{y}_i^{IV}(t;\theta) \tag{20}$$

$$\begin{aligned}\hat{y}_i^{IV}(t;\theta) = &-f_{i,1}\rho^{-1}w_i(t) - \ldots - f_{i,n}\rho^{-n}w_i(t)\\ &+ b_{i,1}\rho^{-1}u_i(t) + \ldots + b_{i,n}\rho^{-n}u_i(t).\end{aligned} \tag{21}$$

The output predictor is now defined in function of a data and a parameter vectors:

$$\hat{y}^{IV}(t;\theta^{IV}) = \varphi^{IV}(t)\theta^{IV} \tag{22}$$

$$\varphi^{IV}(t) = [\varphi_1^{IV}(t)^T\ \varphi_2^{IV}(t)^T\ \ldots\ \varphi_m^{IV}(t)^T]^T \tag{23}$$

$$\begin{aligned}\varphi_i^{IV}(t) = [&-\rho^{-1}w_i(t)\ -\rho^{-2}w_i(t)\ \ldots\ -\rho^{-n}w_i(t)\\ &+\rho^{-1}u_i(t)\ +\rho^{-2}u_i(t)\ \ldots\ +\rho^{-n}u_i(t)]^T.\end{aligned} \tag{24}$$

$$\theta^{IV} = [(\theta_1^{IV})^T\ (\theta_2^{IV})^T\ \ldots\ (\theta_m^{IV})^T]^T \tag{25}$$

$$\theta_i^{IV} = [f_{i,1}\ f_{i,2}\ \ldots\ f_{i,n}\ u_{i,1}\ u_{i,2}\ \ldots\ u_{i,n}]^T \tag{26}$$

One should account for the fact that the contributions from each input are not available and, therefore, their values are computed based on a previous model by means of filtering with (19). Assuming the last estimated model to be the best approximation for the true response, and defining k as the iteration index, it will be used in the composition of $\varphi^{IV}(t;\theta)$. then, the equations (20) to (24) will be rewritten as follows:

$$\hat{y}^{IV}(t;\hat{\theta}^{IV,k},\theta^{IV}) = \sum_{i=1}^{m} \hat{y}_i^{IV}(t;\theta^{IV,k},\theta^{IV}) \tag{27}$$

$$\begin{aligned}\hat{y}_i^{IV}(t;\hat{\theta}^{IV,k},\theta^{IV}) = &-f_{i,1}\rho^{-1}w_i(t;\hat{\theta}^{IV,k}) - \ldots\\ &- f_{i,n}\rho^{-n}w_i(t;\hat{\theta}^{IV,k})\\ &+ b_{i,1}\rho^{-1}u_i(t) + \ldots + b_{i,n}\rho^{-n}u_i(t).\end{aligned} \tag{28}$$

$$\hat{y}_i^{IV}(t;\hat{\theta}^{IV,k},\theta^{IV}) = \varphi_i^{IV}(t;\hat{\theta}^{(IV,k)})\theta_i^{IV} \tag{29}$$

$$\begin{aligned}&\varphi^{IV}(t;\hat{\theta}^{IV,k}) =\\ &\quad[\varphi_1^{IV}(t;\hat{\theta}^{IV,k})^T\ \varphi_2^{IV}(t;\hat{\theta}^{IV,k})^T\ \ldots\ \varphi_m^{IV}(t;\hat{\theta}^{IV,k})^T]^T\end{aligned} \tag{30}$$

$$\begin{aligned}\varphi_i^{IV}(t;\hat{\theta}^{IV,k}) = [&-\rho^{-1}w_i(t;\hat{\theta}^{IV,k})\ \ldots\ -\rho^{-n}w_i(t;\hat{\theta}^{IV,k})\\ &\rho^{-1}u_i(t)\qquad\ldots\qquad\rho^{-n}u_i(t)]^T.\end{aligned} \tag{31}$$

Thus, the prediction error is defined as:

$$\begin{aligned}\varepsilon^k(t;\hat{\theta}^{IV,k},\theta^{IV}) &= y(t) - \hat{y}^{IV}(t;\hat{\theta}^{IV,k},\theta^{IV})\\ &= y(t) - \sum_{i=1}^{m} \hat{y}_i^{IV}(t;\hat{\theta}^{IV,k},\theta^{IV})\\ &= y(t) - \sum_{i=1}^{m} \varphi_i^{IV}(t;\hat{\theta}^{IV,k})^T\theta_i^{IV}\end{aligned} \tag{32}$$

It is worth explaining that the first iteration should be performed based on the initial model, so that:

$$\varepsilon^1(t;\eta,\theta^{IV}) = y(t) - \varphi^{IV}(t;\eta)^T\theta^{IV} \tag{33}$$

$$\varphi^{IV}(t;\eta) = [\varphi_1^{IV}(t;\eta)^T\ \varphi_2^{IV}(t;\eta)^T\ \ldots\ \varphi_m^{IV}(t;\eta^T]^T \tag{34}$$

$$\begin{aligned}\varphi_i^{IV}(t;\eta) = [&-\rho^{-1}z_i(t;\eta)\ \ldots\ -\rho^{-n}z_i(t;\eta)\\ &+\rho^{-1}u_i(t)\quad\ldots\quad+\rho^{-n}u_i(t)]^T\end{aligned} \tag{35}$$

$$z(t;\hat{\eta}) = \hat{y}^{IM}(t;\hat{\eta}) = \sum_{i=1}^{m} \frac{\Gamma_i(\rho;\hat{\eta})}{A(\rho;\hat{\eta})}u_i(t). \tag{36}$$

Similarly to what happens in a instrumental variables algorithm, the vector $\zeta(t;\theta)$ is chosen in such a way it is uncorrelated with the noise, which is equivalent to determine that it does not depend on the output. The instruments will be chosen to be the input and the model output response but, this time, the initial model will be always used to obtain the responses from each input. Hence:

$$\zeta^{IV}(t;\hat{\eta}) = [\zeta_1^{IV}(t;\hat{\eta})^T\ \zeta_2^{IV}(t;\hat{\eta})^T\ \ldots\ \zeta_m^{IV}(t;\hat{\eta})^T]^T \tag{37}$$

$$\begin{aligned}\zeta_i^{IV}(t;\hat{\eta}) = [&-\rho^{-1}z(t;\hat{\eta})\ \ldots\ -\rho^{-n}z(t;\hat{\eta})\\ &+\rho^{-1}u_i(t)\ \ldots\ +\rho^{-n}u_i(t)]^T.\end{aligned} \tag{38}$$

The parameter vector $\hat{\theta}^{IV}$ is finally estimated with a bootstrap method to solve the perform the regression also analogously to that found in instrumental variables method, so that one has:

$$\hat{\theta}^{IV,k+1} = \Big[\sum_{r=1}^{N} \zeta^{IV}(t_r;\hat{\eta})\varphi^{IV}(t_r;\hat{\theta}^{IV,k})^T\Big]^{-1} \times \Big[\sum_{r=1}^{N} \zeta^{IV}(t_r;\hat{\eta})y(t_r)\Big] \tag{39}$$

3.3 Pseudo-linear regression

Another application of a bootstrap method would be for estimating the parameter vector by means of solving a pseudo-linear regression. In this case, most of the theory presented in the previous section remains the same, but the definitions of the vectors $\varphi(t;\theta)$ and $\zeta(t;\theta)$ will change.

For this case, $\zeta(t;\theta) = \varphi(t;\theta)$ and $\varphi(t;\theta)$ will be no longer given as a function of the initial model, but will be recomputed at each iteration as function of the last estimated model. Thus, the definitions are adapted as follows:

- Output predictor:

$$\hat{y}^{PLR}(t;\theta^{PLR},\hat{\theta}^{PLR,k}) = \sum_{i=1}^{m} \hat{y}_i^{PLR}(t;\theta^{PLR},\hat{\theta}^{PLR,k}) \tag{40}$$

$$\hat{y}_i^{PLR}(t;\theta^{PLR},\hat{\theta}^{PLR,k}) = \varphi_i^k(t;\hat{\theta}^{PLR,k})\theta_i^{PLR} \tag{41}$$

- Data vector:

$$\varphi^{PLR}(t;\hat{\theta}^{PLR,k}) = [\varphi_1^{PLR}(t;\hat{\theta}^{PLR,k})^T \ \ldots \ \varphi_m^{PLR}(t;\hat{\theta}^{PLR,k})^T]^T \tag{42}$$

$$\varphi_i^{PLR}(t;\hat{\theta}^{PLR,k}) = [-\rho^{-1}w_i^{PLR,k}(t;\hat{\theta}^{PLR,k}) \ \ldots \ -\rho^{-n}w_i^{PLR,k}(t;\hat{\theta}^{PLR,k}) \ +\rho^{-1}u_i(t) \ \ldots \ +\rho^{-n}u_i(t)]^T \tag{43}$$

- Simulated noise-free output responses:

$$w_i^{PLR,k}(t;\theta^{(PLR,k)}) = \frac{B_i(\rho;\theta^{\hat{PLR},k})}{F_i(\rho;\theta^{\hat{PLR},k})}u_i(t) \tag{44}$$

- Parameter vector:

$$\theta^{PLR} = [(\theta_1^{PLR})^T \ (\theta_2^{PLR})^T \ \ldots \ (\theta_m^{PLR})^T]^T \tag{45}$$

$$\theta_i^{PLR} = [f_{i,1} \ f_{i,2} \ \ldots \ f_{i,n} \ u_{i,1} \ u_{i,2} \ \ldots \ u_{i,n}]^T \tag{46}$$

- Prediction error:

$$\begin{aligned}\varepsilon(t;\theta^{PLR},\hat{\theta}^k) &= y(t) - \hat{y}^{PLR}(t;\theta^{PLR},\hat{\theta}^{PLR,k}) \\ &= y(t) - \sum_{i=1}^{m} \hat{y}_i^{PLR}(t;\theta^{PLR},\hat{\theta}^{PLR,k}) \\ &= y(t) - \varphi^{PLR}(t;\hat{\theta}^{PLR,k})^T\theta^{PLR}\end{aligned} \tag{47}$$

- Parameter estimation:

$$\hat{\theta}^{PLR,k+1} = \Big[\sum_{r=1}^{N} \varphi^{PLR}(t_r;\hat{\theta}^{PLR,k})^T\varphi^{PLR}(t_r;\hat{\theta}^{PLR,k})\Big]^{-1} \times \Big[\sum_{r=1}^{N} \varphi^{PLR}(t_r;\hat{\theta}^{PLR,k})y(t_r)\Big] \tag{48}$$

3.4 Prediction-Error Method and Non-linear Optimization

Finally, a last method consists on solving a non-linear optimization problem by means of the application of the Gauss-Newton method.

Following the developments in Ljung (1987) and Zhu (2001), let the experiment data be referred to as the variable Z^N and the cost function be defined by:

$$V_N(\theta;Z^N) = \frac{1}{N}\sum_{r=1}^{N}\frac{1}{2}\varepsilon(t_r,\theta)^2 = \frac{1}{N}\sum_{r=1}^{N}\frac{1}{2}[y(t_r) - \hat{y}(t_r,\theta)]^2. \tag{49}$$

Its gradient at a given point θ^k is obtained from the equation:

$$V_N'(\theta^k;Z^N) = -\frac{1}{N}\sum_{r=1}^{N}\psi(t_r;\theta^k)\varepsilon(t_r,\theta^k), \tag{50}$$

$\psi(t;\theta^k)$ in which the gradient of the predicted output $\hat{y}(t,\theta^k)$. Here, the latter will be defined directly as the filtered input:

$$\hat{y}(t;\theta) = \sum_{i=1}^{m}\frac{B_i(\rho)}{F_i(\rho)}u_i(t). \tag{51}$$

The derivatives of $\hat{y}(t;\theta)$ relative to the parameters are:

$$\frac{\partial}{\partial f_{i,j}}\hat{y}(t;\theta) = -\frac{B_i(\rho;\theta)}{F_i(\rho;\theta)^2}\rho^{n-j}u_i(t) \tag{52}$$

$$\frac{\partial}{\partial b_{i,j}}\hat{y}(t;\theta) = \frac{1}{F_i(\rho;\theta)^2}\rho^{n-j}u_i(t), \tag{53}$$

which leads to the gradient of $\hat{y}(t;\theta)$, at a point θ^k. Applying the filter ρ^{-n} to this gradient, it is given by:

$$\begin{aligned}\rho^{-n}\psi(t;\theta^k) &= \rho^{-n}\frac{\partial}{\partial\theta^T}\hat{y}(t;\theta^k)\Big|_{\theta=\theta^k} \\ &= \rho^{-n}\frac{\partial}{\partial\theta^T}\sum_{i=1}^{m}\frac{B_i(\rho;\theta^k)}{F_i(\rho;\theta^k)}u_i(t)\Big|_{\theta=\theta^k} \\ &= \begin{bmatrix} \Big[-\frac{B_1(\rho;\theta^k)}{F_1(\rho;\theta^k)^2}\rho^{-1}u_1(t) \ \ldots \ -\frac{B_1(\rho;\theta^k)}{F_1(\rho;\theta^k)^2}\rho^{-n}u_1(t)\Big]^T \\ \vdots \\ \Big[-\frac{B_m(\rho;\theta^k)}{F_m(\rho;\theta^k)^2}\rho^{-1}u_m(t) \ \ldots \ -\frac{B_m(\rho;\theta^k)}{F_m(\rho;\theta^k)^2}\rho^{-n}u_m(t)\Big]^T \\ \Big[\frac{1}{F_1(\rho;\theta^k)}\rho^{-1}(t) \quad \ldots \quad \frac{1}{F_1(\rho;\theta^k)}\rho^{-n}u_1(t)\Big]^T \\ \vdots \\ \Big[\frac{1}{F_m(\rho;\theta^k)}\rho^{-1}u_m(t) \quad \ldots \quad \frac{1}{F_m(\rho;\theta^k)}\rho^{-n}u_m(t)\Big]^T \end{bmatrix}\end{aligned} \tag{54}$$

In the Gauss-Newton method, the value $H_N(\theta_N^{GN,k})$ is used as an approximation of the Hessian of the predicted error $\varepsilon(t,\theta)$ and is given by:

$$H_N(\theta_N^{GN,k}) = \frac{1}{N}\sum_{r=1}^{N}\psi(t_r;\theta^{GN,k})\psi^T(t_r;\theta^{GN,k}). \tag{55}$$

The Gauss-Newton method is finally applied by means of the following equation:

$$\begin{aligned}\hat{\theta}_N^{GN,k+1} &= \hat{\theta}_N^{GN,k} - [H_N(\theta_N^{GN,k})]^{-1} V_N'(\theta^{GN,k}; Z^N) \\ &= \hat{\theta}_N^{GN,k} - \Big[\frac{1}{N}\sum_{r=1}^{N} \psi(t_r;\theta^{GN,k})\psi^T(t_r;\theta^{GN,k})\Big]^{-1} \\ &\quad \times \Big[-\frac{1}{N}\sum_{r=1}^{N} \psi(t_r;\theta^{GN,k})\varepsilon(t_r,\theta^{GN,k})\Big] \\ &= \hat{\theta}_N^{GN,k} + \Big[\frac{1}{N}\sum_{r=1}^{N} \rho^{-n}\psi(t_r;\theta^{GN,k})\rho^{-n}\psi^T(t_r;\theta^{GN,k})\Big]^{-1} \\ &\quad \times \Big[\frac{1}{N}\sum_{r=1}^{N} \rho^{-n}\psi(t_r;\theta^{GN,k})\rho^{-n}\varepsilon(t_r,\theta^{GN,k})\Big].\end{aligned} \tag{56}$$

An important issue in this case is related to the initialization of the method. Since the initial model is not of the same model structure, equation (54) does not apply to it. A solution for this problem would be provided by finding a model of the same order as the one used in the Gauss-Newton method, which could be accomplished for instance, through a model reduction technique, applying one of the bootstrap methods presented in the previous subsections in the first iteration or even an ARX estimate of equivalent order.

4. IMPLEMENTATION ISSUES

Now some considerations are made about the computational issues for the implementation of this algorithm. First, as regards the suggestion of using an ARX model with an order equal to the number of inputs to obtain the initial model. Since the ARX model structure has only one common denominator polynomial to describe all the input-output dynamics, the purpose of identifying with a higher order model,i.e. a larger number of parameters, is to have enough degrees of freedom to take into account of the different poles from each input-output subsystem and, therefore, individualize as much as possible each input contribution to the output, so one can start up the iterations with an already rather accurate representation of the responses. However, this choice implicates an over-parameterization. The exceeding zeros and poles should tend to cancel or at least diminish the effect of each other, but, as the number of inputs grows, this strategy may be subject to numerical issues because of ill-conditioning of the least squares covariance matrix, besides the fact that the model becomes more subject to variance errors due to noise influence. If the higher-order model is not adequate, regularization techniques could be applied or even lower-order initial models could be tried instead.

Remarkably, not necessarily the initial model will be a nice representation for the system: even if the corresponding approximation presents a small error, the model system for a stable system may present unstable poles, for instance. However, this will not be an issue if the responses approximations actually are close to the true ones. The purpose at this point is solely to have an initial separation of the responses generated by each input, having this purpose achieved, the iterations performed in the second step will refine the parameter estimations so that a reasonable model can be raised.

Finally, it is important to highlight the fact that, following the continuous-time approach presented therein, the derivative operations over the collected data are actually not realizable, which requires the use of some strategy to perform the model estimation. One possible solution, which was used in the methods, is the use of the state-variable filter (SVF) method (Wang and Garnier, 2008). The filters to be used are:

$$L(\rho) = \rho^{-n\times m}, \tag{57}$$

for the initial model, and:

$$L(\rho) = \rho^{-n}, \tag{58}$$

for the n-th-order models ultimately obtained.

SVF was used in the development of the methods by changing the linear regression equation. For example, to obtain the data vector defined in (11), the inputs and outputs in equation (8) are substituted by their filtered versions, so that one has:

$$L(\rho)A(\rho;\eta)y(t) = \sum_{i=1}^{m} L(\rho)\Gamma(\rho;\eta)u_i(t) + e(t), \tag{59}$$

from which the further definitions in the section may be derived.

Example 1. Let a system be modelled by the following transfer matrix:

$$G(s) = \begin{bmatrix} \dfrac{5}{s+2} & \dfrac{2}{s+10} \end{bmatrix} \tag{60}$$

In order to illustrate the algorithms evolution along the iterations, a numerical simulation is performed with this model such that a step signal is applied to each input at different instants. The output is noise-free. The input-output data from this simulation can be observed in Figure 1.

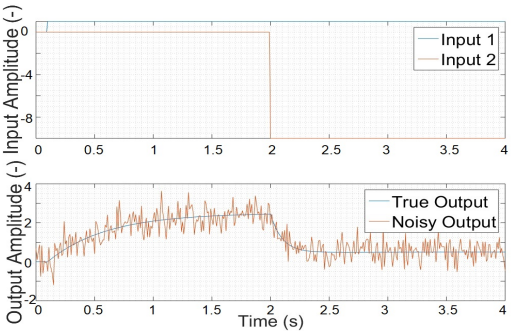

Figure 1. Input-output data of numerical simulation described in Example 1.

The IFUV-LS is applied to estimate a model from the simulation data. In Table 1, some of the models obtained along the identification procedure are listed, namely, the second-order ARX initial model (iteration 0), the first model — already made of 1st-order subsystems — obtained by means of the bootstrap algorithm (iteration 1), the fifth one (iteration 5) and the final one (iteration F), which is the one delivered for the user. The algorithm stopping criteria consists in reaching a relative reduction in the approximation error, evaluated by means of the

Table 1. True model and evolution of some of the models computed throughout the IFUV-LS algorithm for Example 1.

Model	Transfer Matrix	MSE
True Model	$\begin{bmatrix} \frac{5}{s+2} & \frac{2}{s+10} \end{bmatrix}$	0
Initial	$\begin{bmatrix} \frac{2.085s+34.87}{s^2+8.643s+14.11} & \frac{1.501s+2.701}{s^2+8.643s+14.11} \end{bmatrix}$	0.0081
1st Iteration	$\begin{bmatrix} \frac{4}{s+1.522} & \frac{1.903}{s+9.514} \end{bmatrix}$	0.0112
5th Iteration	$\begin{bmatrix} \frac{4.108}{s+1.567} & \frac{2.236}{s+11.15} \end{bmatrix}$	0.0083
Final Model	$\begin{bmatrix} \frac{4.109}{s+1.566} & \frac{2.095}{s+9.938} \end{bmatrix}$	0.0045

mean square value (MSE), less than 10^{-5} or a maximum of 50 iterations. The algorithm converged, in this case, with 31 iterations. It is possible to notice that the response generated by the models resulting from the first iterations are rather less accurate than the one generated by the initial higher-order model, nevertheless the error reduces gradually until it provides a better approximation than the one obtained with the initial model.

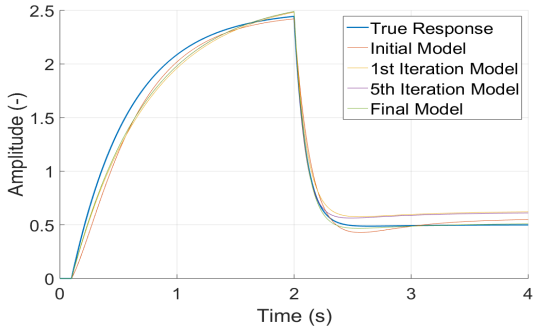

Figure 2. Iterative fixed uncorrelated variables least squares models responses through the algorithm iterations.

5. NUMERICAL SIMULATIONS

Numerical simulations to evaluate the algorithms performance are presented in the following. The simulated system is the Wood-Berry distillation column (Wood and Berry, 1973). Transport delay assumed known and, without loss of generality, set to zero in this paper. PRBS input signals were applied to inputs 1 and 2 with switching periods of 3.6 min and 3.0 min, respectively. White noise with $\sigma^2 = 4$ is added to output 1, which is used in the identification process; $T_s = 0.1$ min and $T_{sim} = 200$ min;

Table 2 shows the values of the true parameters used in the simulated models.

For each case, 100 simulations are executed, with the same input but with different realizations of the output noise. The parameters are then computed with each algorithm and their averages and standard deviations are showed in Table 3. This table presents also the average approximation errors, evaluated by means of the MSE (mean square error), obtained with each algorithm. For the three iterative algorithms, the stopping criteria is defined by a relative reduction in the MSE less than 10^{-5} or a maximum of 50 iterations. Figure 3 shows one input-output

Table 2. True parameters relative to the first input of the Wood-Berry distillation column (delays set to zero).

	Parameters	Values
True Parameters	b_1	0.7665
	f_1	0.05988
	b_2	-0.9
	f_2	0.04762

realization and Figures 4 to 6 illustrate the responses from the 100-model set obtained with each algorithm along with the true response.

Table 3. Numeric simuations results with the three evaluated algorithms: pseudo-linear regression (PLR), iterative fixed uncorrelated variables least squares (IFUV-LS) and Gauss-Newton optimization (GN).

Algorithm	Parameters	Average of Estimated Parameters	Relative Error to the True Parameters (%)	Estimated Parameters Standard Deviation
PLR	$\hat{b}_1$	0.76842	0.25039	0.01946
	$\hat{f}_1$	0.06073	1.42751	0.00291
	$\hat{b}_2$	-0.90129	0.14333	0.01742
	$\hat{f}_2$	0.04796	0.72014	0.00200
	MSE	0.01261		0.00678
IFUV-LS	$\hat{b}_1$	0.77050	0.52175	0.02014
	$\hat{f}_1$	0.06033	0.75282	0.00319
	$\hat{b}_2$	-0.89599	0.44505	0.01669
	$\hat{f}_2$	0.04740	0.46635	0.00215
	MSE	0.01399		0.00859
GN	$\hat{b}_1$	0.74366	2.97933	0.13636
	$\hat{f}_1$	0.05926	1.03072	0.00638
	$\hat{b}_2$	-0.88936	1.18226	0.11882
	$\hat{f}_2$	0.04812	1.05590	0.00437
	MSE	0.38144		3.17610

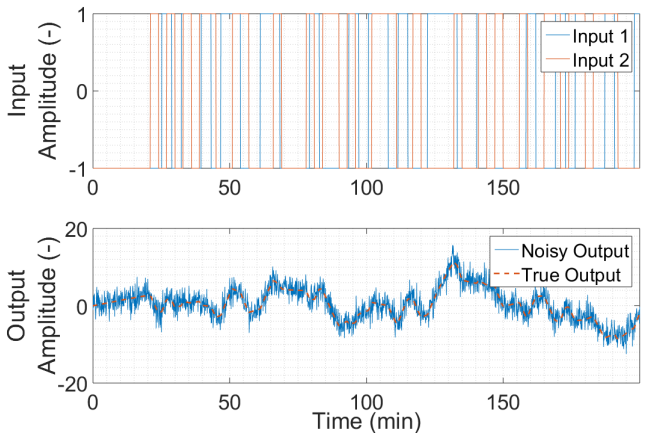

Figure 3. Wood-Berry distillation column (delays set to zero) simulation: inputs, true and and one out of the 100 realizations of the noisy output.

From the MSE values, it is possible to observe that the pseudo-linear regression and the IFUV-LS perform rather

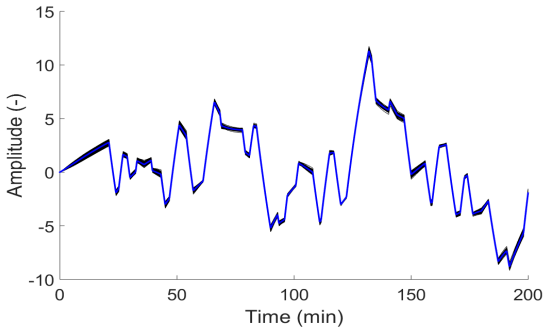

Figure 4. Response of models obtained from IFUV-LS method (thin black line) and true response (thick blue line).

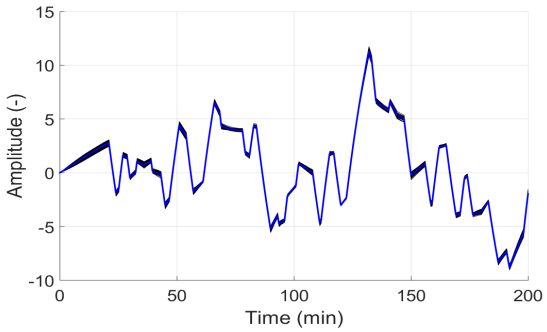

Figure 5. Response of models obtained from Pseudo-linear Regression method (thin black line) and true response (thick blue line).

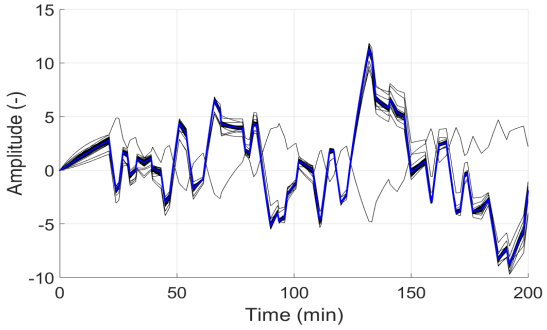

Figure 6. Response of models obtained from Gauss-Newton Method (thin black line) and true response (thick blue line).

similarly, with a slight difference in accuracy and precision. The Gauss-Newton method, on the other hand, had a poor performance as compared to the other two algorithms, with essentially all averages and standard deviations ranging from twice as large to one order of magnitude higher than those of the other two algorithms. These numbers are well elucidated by the Figures 4 to 6, where one can visualize that the 100 models generated by the first two algorithms almost overlap the true response, evidencing a reasonable accuracy and precision for the PRBS input signal, while the Gauss-Newton models present a number of responses clearly different to the one generated by the true model.

6. CONCLUSION

In this work three iterative algorithms to render a continuous-time output-error MISO model have been presented and compared. One of them is proposed herein and is inspired in the instrumental variables method, with the difference of using a fixed matrix with the initial model responses as the uncorrelated variable in the estimation, while the data vector is updated throughout the iterations. The other ones consist in an adaptation of the pseudo-linear regression approach and the Gauss-Newton method to address the model output-error model structure as treated herein, by means of a decomposition into SISO models whose responses are accumulated to form the prediction error. The results demonstrate that the Gauss-Newton method was outperformed by the other two methods, which presented similar performance, both in accuracy and precision.

Further research is to be performed in order to: investigate the impact in performance due to the use of lower-order models as the initial model, as well as due to the choice of different model reduction strategies in the Gauss-Newton method; verify the proposed algorithms robustness in the identification of higher-order models; develop on consistency analysis; and extend the methods to allow the inclusion of time-delays.

REFERENCES

Chan, S.C., Lin, J.Q., Sun, X., Tan, H.J., and Xu, W.C. (2019). A new variable forgetting factor-based bias-compensation algorithm for recursive identification of time-varying multi-input single-output systems with measurement noise. *IEEE Transactions on Instrumentation and Measurement*.

Ding, F. (2013). Coupled-least-squares identification for multivariable systems. *IET Control Theory & Applications*, 7(1), 68–79.

Ding, F. (2014). Hierarchical estimation algorithms for multivariable systems using measurement information. *Information Sciences*, 277, 396–405.

Ding, J. (2018). Recursive and iterative least squares parameter estimation algorithms for multiple-input–output-error systems with autoregressive noise. *Circuits, Systems, and Signal Processing*, 37(5), 1884–1906.

El-Sherief, H. and Sinha, N. (1979). Identification of multivariable systems in the transfer-function matrix form. *Journal of Cybernetics*, 9(2), 113–125.

He, C., Yu, C., and Wang, P. (2019). Distributed identification of large-scale miso output-error systems. In *2019 IEEE 15th International Conference on Control and Automation (ICCA)*, 818–823. IEEE.

Jin, Q., Wang, Z., Yang, R., and Wang, J. (2014). An effective direct closed loop identification method for linear multivariable systems with colored noise. *Journal of Process Control*, 24(5), 485–492.

Liu, Q., Ding, F., and Yang, E. (2018). Parameter estimation algorithm for multivariable controlled autoregressive autoregressive moving average systems. *Digital Signal Processing*, 83, 323–331.

Liu, Y. and Ding, F. (2013). Convergence properties of the least squares estimation algorithm for multivariable systems. *Applied Mathematical Modelling*, 37(1-2), 476–483.

Ljung, L. (1987). *System Identification: Theory for the User*. Prentice Hall.

Ljung, L. and Wahlberg, B. (1992). Asymptotic properties of the least-squares method for estimating transfer functions and disturbance spectra. *Advances in Applied Probability*, 412–440.

Ma, H., Pan, J., Lv, L., Xu, G., Ding, F., Alsaedi, A., and Hayat, T. (2019). Recursive algorithms for multivariable output-error-like arma systems. *Mathematics*, 7(6), 558.

Rajapandiyan, C. and Chidambaram, M. (2012a). Closed-loop identification of multivariable systems by optimization method. *Industrial & engineering chemistry research*, 51(3), 1324–1336.

Rajapandiyan, C. and Chidambaram, M. (2012b). Closed-loop identification of second-order plus time delay (soptd) model of multivariable systems by optimization method. *Industrial & engineering chemistry research*, 51(28), 9620–9633.

Söderström, T. (2012). A generalised instrumental variable estimator for multivariable errors-in-variables identification problems. *International journal of control*, 85(3), 287–303.

Wang, L. and Garnier, H. (2008). *Identification of Continuous-time Models from Sampled Data*. Springer London.

Wang, Y. and Ding, F. (2016a). Filtering-based iterative identification for multivariable systems. *IET Control Theory & Applications*, 10(8), 894–902.

Wang, Y. and Ding, F. (2016b). Novel data filtering based parameter identification for multiple-input multiple-output systems using the auxiliary model. *Automatica*, 71, 308–313.

Wood, R. and Berry, M. (1973). Terminal composition control of a binary distillation column. *Chemical Engineering Science*, 28(9), 1707–1717.

Yan, W., Pang, C.K., and Du, C. (2016). Identification of miso systems using periodic inputs and its application to dual-stage hard disk drives. *IEEE Transactions on Magnetics*, 53(3), 1–7.

Zhang, Z., Ding, F., and Liu, X. (2011). Hierarchical gradient based iterative parameter estimation algorithm for multivariable output error moving average systems. *Computers & Mathematics with Applications*, 61(3), 672–682.

Zhu, Y. (2001). *Multivariable system identification for process control*. Elsevier.