Revisiting the basic issue of parameter estimation in system identification - a new approach for multi-value estimation

Sergio Bittanti and Simone Garatti

***Abstract*— In this paper, we consider one of the basic estimation problem, that of identifying an unknown parameter in a given model from measurements of input/output data. The existing methods have been conceived for the estimation of the value taken by the parameter in a given functioning condition. However, there are situations where one has to provide an estimator equally valid for different values of the parameter associated with various functioning conditions (multi-value estimation problem). The application of the available techniques lead then to poor accuracy in estimation. In this paper we propose a novel approach, the two-stage approach, tailored to the multi-value estimation problem. We compare its performances with those achievable with other parameter estimation techniques such as Prediction Error and Kalman Filter based methods. By means of a benchmark example, we spot out advantages and drawbacks of each method, by also discussing their domain of applicability. It turns that the two stage approach offers significant improvements.**

***Index Terms*— System identification, Parameter estimation, White box identification, Extended Kalman filter methods.**

I. Introduction

This paper focuses on the basic problem of estimating unknown parameters in a given plant from observed data. To be precise, suppose that data are generated by a dynamical system (continuous time or discrete time, linear or nonlinear, finite or infinite dimensional, noise free or subject to disturbances) depending on a certain parameter vector $\theta \in \mathbb{R}^q$. The system is denoted by $P(\theta)$ as in Figure 1. While a mathematical model (and a corresponding simulator)

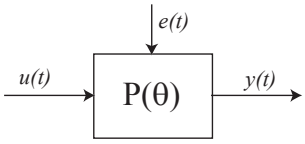

Fig. 1. The data generating system.

for $P(\theta)$ is available, the current value of parameter θ is unknown and it has to be retrieved based on an experiment on the plant (white-box identification, [5]). The system behavior is thus observed for a certain time interval over which a number N of input and output observations $\bar{D}^N = \{\bar{y}(1), \bar{u}(1), \ldots, \bar{y}(N), \bar{u}(N)\}$ are collected. The issue is then how to exploit the information contained in the data in order to obtain a fair estimate of the uncertain parameter θ. Of course, the above setting applies also in a purely time-series framework where no input signal is present. In that case, $\bar{D}^N = \{\bar{y}(1), \ldots, \bar{y}(N)\}$.

This basic problem has been addressed many times in the literature, [4], [5], [8], [14], [19], but still there are situations where a satisfactory solution is not available. In particular, problems are often encountered when the parameter vector θ can cover a wide range of values corresponding to different dynamics of the plant P. In those situations, the estimation method should be designed so as to be equally valid for all possible values taken by θ. We will refer to such a problem as *multi-value* estimation problem.

In order to apply currently available estimation algorithms to multi-value problems, it is necessary to adapt the "algorithm tuning knobs" to the current situation associated with the unknown value taken by θ. This means that a human-supervised tuning is required from time to time. However, in many application frameworks such a human-supervision is not possible and the estimation process should be fully automatic, able to properly work independently of the value taken by the unknown θ. To be more concrete, this is illustrated in the following example.

Example 1 (Pacejka's model parameters estimation):
The determination of the lateral force generated by a tyre and acting on a car can be made by resorting to the so-called Pacejka's magic formula which supplies the lateral force as a function of the steering angle, [16]. As is well known such a formula is a non-linear function depending on five parameters. Hence the problem of determining the lateral force is indeed that of estimating these parameters.
Depending on the tyre in use, with its own characteristics in term of size, constitutive material, inflation pressure, deterioration, etc., the Pacejka's parameters may cover a wide variety of values. The issue is to set-up an estimation algorithm of these parameters supplying a reliable estimate of the lateral force for any tyre operating conditions. Clearly, if the estimation algorithm has to be embedded in a device installed in the car, no human-supervision is allowed, and furthermore the estimation algorithm should work in absence of any information on the specific tyre characteristics, so as to obtain fair estimates notwithstanding the tyre changes during the life of the car. □

In this paper, we propose a new estimation method, named the *two-stage* approach, which is suitably tailored to *multi-value* estimation. Its basic rationale is to reconstruct off-line the relationship linking the data to parameter θ through simulation trials. This is achieved thanks to an intermediary

Paper supported by the MIUR national project "Identification and adaptive control of industrial systems" and by CNR - IEEIT

S. Bittanti and S. Garatti are with the Dipartimento di Elettronica ed Informazione, Politecnico di Milano, p.zza L. da Vinci 32, 20133 Milano, Italy. E-mail: {bittanti,sgaratti}@elet.polimi.it

step aiming at the generation of a set of *artificial data*. The procedure develops in two phases: the first one transfers the information contained in the original data into the artificial data, while the second one enables establishing the link between these last data and parameter θ.
The two-stage method has a range of applicability which looks much wider than that of other approaches today available.

The paper is organized as follows. First, traditional approaches to parameter estimation are briefly summarized in Section II and their advantages and drawbacks are spotted out. The new two-stage approach is then discussed in Section III, while Section IV presents a benchmark example allowing the comparison between different techniques.

II. Traditional approaches in multi-value estimation

Conceptually, a parameter estimator is nothing but a function $\widehat{f}\colon \mathbb{R}^{2N} \to \mathbb{R}^q$ which maps the measured observations $\bar{D}^N = \{\bar{y}(1), \bar{u}(1), \ldots, \bar{y}(N), \bar{u}(N)\}$ into a value for θ. The design of an estimator consists in finding such a map so that the returned estimate is as close as possible to the true value. Normally, $\widehat{f}$ is deduced by exploiting the model equations for $P(\theta)$, and turns out to be implicitly given through some optimization procedure.
Three widely used methods are now outlined.

A. Prediction Error approaches

In this approach, a prediction error loss function

$$V(\theta) = \sum_{i=1}^{N} \left(y(i) - \widehat{y}(i,\theta)\right)^2$$

is considered, where $\widehat{y}(i,\theta)$ is a predictor of the system output derived through the model equation for $P(\theta)$. Then, the estimate of θ is obtained by minimizing $V(\theta)$, i.e.

$$\widehat{\theta} = \arg\min V(\theta).$$

Here, the function $\widehat{f}$ mapping observations into an estimated value is implicitly defined by this optimization procedure.
Although very intuitive, this approach suffers from severe drawbacks as reported in the literature, see e.g. [5]. First of all, the derivation of $\widehat{y}(i,\theta)$ may not be easy when the model $P(\theta)$ is complex (nonlinear, infinite dimensional, etc.). Moreover, when the predictor $\widehat{y}(i,\theta)$ is not perfectly tuned, the method is highly sensitive to disturbances, especially at low frequencies, see e.g. [5].
On top of that, one cannot neglect the computational burden required by these methods. Indeed, $V(\theta)$ is typically a non convex function and its minimization may be tough. If one resorts to simple gradient-based methods, the obnoxious problem of local minima cannot be avoided. Alternatively, one can consider gridding methods, but then "simulation would require supercomputers, and optimization an order of magnitude more", [5].
All these difficulties make it impossible the efficient use of these methods when parameter θ can cover a multitude of values as required in multi-value estimation problems.

B. Maximum likelihood

The maximum likelihood (ML) approach [7], [3], [2] is another well known estimation method taken from statistics. It consists in computing the likelihood of possible values of θ given the observed data; then, the estimate of θ is defined as the value maximizing the likelihood.
In case of complex systems, ML suffers from major drawbacks since it requires the full knowledge of the probability distribution of the disturbances in order to construct the probability density of data as a function of the unknown parameter. Furthermore, the calculation and maximization of the likelihood raises all the computational complexity issues mentioned before for the prediction error approaches.

C. Kalman filter based approaches

In Kalman filter based methods, [1], [8], [9], [10], [11], [12], [15], [18], [21], [22], [23], parameter θ is seen as a state variable by introducing an additional state equation of the type: $\theta(k+1) = \theta(k)$ or $\dot{\theta}(t) = 0$, depending if time is discrete or continuous[1]. Then, the estimation problem is reformulated as a state prediction problem. In this way, the function $\widehat{f}$ mapping the data into the estimate is implicitly defined by the Kalman filter equations.
As is well known, even if $P(\theta)$ were a linear model, the resulting prediction problem would be nonlinear due to the introduction of the additional state equation. Thus, typically one has to resort to nonlinear Kalman filtering, for which the two most common approaches are the so-called Extended Kalman Filter (EKF), or the Unscented Kalman Filter (UKF). There is a huge literature on KF methods, see e.g. [1], [8], [9], [10], [12], [18], to which we refer the reader for the EKF and UKF equations.
Apart from the difficulties one can encounter when the system is continuous-time and/or infinite dimensional, the actual critical issue of EKF and UKF is that, being settled in a Bayesian framework, an initial guess for the estimation error mean and covariance matrix must be supplied. However, the convergence of the parameter estimate is very sensitive to the tuning of this mean and covariance matrix, and there are celebrated (yet simple) examples showing the possible divergence/nonconvergence depending on the initialization (see e.g. [13]). In multi-value estimation problems, the only possibility to obtain reasonable estimates is the re-tuning of mean and covariance for the current value of θ in each operating condition. Indeed only local convergence is achievable, as shown in [6], [13], [17], [20]). Normally, however, no a-priori information is available on the current θ and the re-initialization can be performed only by data manipulation with trial and error empirical attempts and questionable findings.
In conclusion, it appears that there is no way of making the estimation process via KF methods fully automatic independent of a human-operator supervision. This prevents

[1]Perhaps it is worth noticing that many times an additional equation of the type $\theta(k+1) = \theta(k) + w(k)$ or $\dot{\theta}(t) = w(t)$ where w is white noise with suitable variance is preferred in order to increase the reactivity of the algorithm.

the use of these approaches in many problems encountered in practice.

III. The two-stage approach

In this section we propose a new parameter estimation method which is suitably tailored for the multi-value estimation problem.

The basic rationale is to resort to the plant simulator and to perform off-line intensive simulation trials in order to reconstruct the function $\widehat{f}$ mapping measured input/output data into an estimate for the parameter θ.

To be precise, we use the *simulator* to generate input/output data for a number of different values of the unknown parameter θ. That is, we collect N measurements

$$D_1^N = \{y^1(1), u^1(1), \ldots, y^1(N), u^1(N)\}$$

for $\theta = \theta_1$; N measurements

$$D_2^N = \{y^2(1), u^2(1), \ldots, y^2(N), u^2(N)\}$$

for $\theta = \theta_2$; and so on and so forth. By repeated simulation

θ_1	$D_1^N = \{y^1(1), u^1(1), \ldots, y^1(N), u^1(N)\}$
θ_2	$D_2^N = \{y^2(1), u^2(1), \ldots, y^2(N), u^2(N)\}$
$\vdots$	$\vdots$
θ_m	$D_m^N = \{y^m(1), u^m(1), \ldots, y^m(N), u^m(N)\}$

TABLE I

The simulated data chart as the starting point of the two-stage method.

experiments one can work out a set of, say m, pairs $\{\theta_i, D_i^N\}$ as summarized in Table I. Such set of data is referred to as the *simulated data chart*.

From the simulated data chart, $\widehat{f} : \mathbb{R}^{2N} \to \mathbb{R}^q$ is reconstructed as that map minimizing the estimate error over simulated data, i.e.

$$\widehat{f} \leftarrow \min_f \frac{1}{m} \sum_{i=1}^{m} \left\| \theta_i - f(y^i(1), u^i(1), \ldots, y^i(N), u^i(N)) \right\|^2. \quad (1)$$

Should $\widehat{f}$ be found, then the θ corresponding to actual measurements $\overline{D}^N = \{\overline{y}(1), \overline{u}(1), \ldots, \overline{y}(N), \overline{u}(N)\}$ is estimated as

$$\widehat{\theta} = \widehat{f}(\overline{y}(1), \overline{u}(1), \ldots, \overline{y}(N), \overline{u}(N)).$$

As is clear, solving Problem (1) requires the preliminary choice of a suitable class of functions $\mathscr{F}$ within which performing optimization. This is indeed a critical issue, due to the high dimensionality of the problem. Indeed f depends upon $2N$ variables, normally a very large number if compared to the number m of experiments. If $\mathscr{F}$ is a class of low-complexity functions, then it is difficult to replicate the relationship linking D^N to θ for all values of θ (bias error). On the opposite, if $\mathscr{F}$ is a class of high-complexity functions, then the over-fitting issue arises (variance error), see [14], [19].

In order to achieve a sensible compromise between bias and variance error, the two-stage approach is proposed. In this method, the selection of the family of functions $\mathscr{F}$ is split in two steps. This splitting is the key to select a proper class $\mathscr{F}$ and, in turn, to obtain a good estimator $\widehat{f}$.

To be more precise, the objective of the first step is to reduce the dimensionality of the estimation problem, by generating a new data chart: the new chart is composed again of m sequences; however, each sequence is constituted by a limited number n of samples ($n \ll N$). We will call such sequences *compressed artificial data sequences* and the corresponding chart the *compressed artificial data chart*. In the second step, the map between the compressed artificial observations and parameter θ is identified. By combining the results of the two steps, the map $\widehat{f}$ is finally unveiled.

We now will give more details on each of the two stages.

First stage. The first step consists in a compression of the information conveyed by measured input/output sequences D_i^N in order to obtain data sequences $\widetilde{D}_i^n$ of reduced dimensionality. While in the data D_i^N the information on the unknown parameter θ_i is scattered in a long sequence of N samples, in the new compressed artificial data $\widetilde{D}_i^n$ such information is compressed in a short sequence of n samples ($n \ll N$). This leads to a new compressed artificial data chart

θ_1	$\widetilde{D}_1^n = \{\alpha_1^1, \ldots, \alpha_n^1\}$
θ_2	$\widetilde{D}_2^n = \{\alpha_1^2, \ldots, \alpha_n^2\}$
$\vdots$	$\vdots$
θ_m	$\widetilde{D}_m^n = \{\alpha_1^m, \ldots, \alpha_n^m\}$

TABLE II

The compressed artificial data chart.

constituted by the pairs $\{\theta_i, \widetilde{D}_i^n\}$, $i = 1, \ldots, m$, see Table II.

The compressed artificial data sequence $\widetilde{D}_i^n$ can be derived from D_i^N by resorting to a standard identification method. To be precise, one can fit a simple model to each sequence $D_i^N = \{y^i(1), u^i(1), \ldots, y^i(N), u^i(N)\}$ and then adopts the parameters of this model, say $\alpha_1^i, \alpha_2^i, \ldots, \alpha_n^i$, as compressed artificial data, i.e. $\widetilde{D}_i^n = \{\alpha_1^i, \ldots, \alpha_n^i\}$.

To fix ideas, we suggest the following as a typical method for the generation of compressed artificial data. For each $i = 1, 2, \ldots, m$, the data sequence

$$D_i^N = \{y^i(1), u^i(1), \ldots, y^i(N), u^i(N)\}$$

can be concisely described by an ARX model:

$$\begin{aligned} y^i(t) \;=\; & \alpha_1^i y^i(t-1) + \cdots \alpha_{n_y}^i y^i(t-n_y) + \\ & \alpha_{n_y+1}^i u^i(t-1) + \cdots + \alpha_{n_y+n_u}^i u^i(t-n_u), \end{aligned}$$

with a total number of parameters $n = n_y + n_u$. The parameters $\alpha_1^i, \ldots, \alpha_n^i$ of this model can be worked out by means of the least squares algorithm ([14], [19]):

$$\begin{bmatrix} \alpha_1^i \\ \vdots \\ \alpha_n^i \end{bmatrix} = \Big[\sum_{t=1}^{N} \varphi^i(t) \varphi^i(t)^T \Big]^{-1} \cdot \sum_{t=1}^{N} \varphi^i(t) y^i(t), \quad (2)$$

$\varphi^i(t) = [y^i(t-1) \cdots y^i(t-n_y)\; u^i(t-1) \cdots u^i(t-n_u)]^T$.

Remark 1 (Physical interpretation of the artificial data): While $P(\theta)$ is a mathematical description of a real plant, the simple model class selected to produce the compressed

artificial data does not need to have any physical meaning; this class plays a purely instrumental and intermediary role in the process of bringing into light the hidden relationship between the unknown parameter and the original collected data. In this connection, we observe that the choice of the ARX model order is not a critical issue. Anyhow, one can resort to the available complexity selection criteria such as FPE or AIC. □

In conclusion, the first stage of the method aims at finding a function $\widehat{g}:\mathbb{R}^{2N}\to\mathbb{R}^n$ transforming each simulated data sequence D_i^N into the a new sequence of compressed artificial data $\widetilde{D}_i^n$ conveying the information on θ_i. As compressed artificial data we take the parameters of a simple model, identified from D_i^N. In this way, function $\widehat{g}$ is implicitly defined by the chosen class of simple models together with the corresponding identification algorithm.

Second stage. Once the compressed artificial data chart in Table II has been worked out, problem (1) becomes that of finding a map $\widehat{h}:\mathbb{R}^n\to\mathbb{R}^q$ which fits the set of m compressed artificial observations to the corresponding parameter vectors, i.e.

$$\widehat{h}\leftarrow\min_h\frac{1}{m}\sum_{i=1}^{m}\left\|\theta_i-h(\alpha_1^i,\ldots,\alpha_n^i)\right\|^2. \tag{3}$$

Function minimization in (3) is reminiscent of the original minimization problem in (1). However, being n small, the bias versus variance error trade-off is no more an issue.
As for the choice of h one can select a linear function: $h(\alpha_1^i,\ldots,\alpha_n^i)=c_1\alpha_1^i+\ldots+c_n\alpha_n^i$, $c_i\in\mathbb{R}^q$, i.e. each component of h is just a linear combination of the compressed artificial data $\alpha_1^i,\ldots,\alpha_n^i$. As in any linear regression, the parameters c_i appearing here can be easily computed via least squares, at a low computational cost. Of course such a way of parameterizing h is computationally cheap but possibly loose. Better results are expected by choosing a class of nonlinear functions, such as Neural Networks or NARX models. The minimization in (3) can be performed by resorting to the back-propagation algorithm or to other standard algorithms developed for these classes of nonlinear functions.

Remark 2 (The functions $\widehat{g}$ and $\widehat{h}$): The two-stage methods is based on two functions: $\widehat{g}$ and $\widehat{h}$. The former is the *compression function*, transforming simulated data into compressed artificial data. The latter is the *fitting function* providing the map from the compressed artificial data to the unknown parameter. While $\widehat{g}$ is chosen by the designer, $\widehat{h}$ is identified by fitting the parameter values to the corresponding compressed artificial data. □

Use of the two-stage method. Once function $\widehat{g}$ has been chosen and function $\widehat{h}$ has been identified, the function $\widehat{f}$ mapping input/output data into the estimate for θ is given by $\widehat{h}(\widehat{g}(\cdot))$, see Figure 2. When an actual input/ouput sequence is observed, say $\overline{D}^N=\{\overline{y}(1),\overline{u}(1),\ldots,\overline{y}(N),\overline{u}(N)\}$, the corresponding unknown parameter can then be estimated as: $\widehat{\theta}=\widehat{h}(\widehat{g}(\overline{D}^N))$.
As previously discussed, viewing this data-θ function as the composition of $\widehat{g}$ and $\widehat{h}$ is the key to transform a numerically intractable problem into an affordable one.

Fig. 2. The estimator function composition.

Remark 3 (Nonlinearity in estimation): Suppose that both in the first stage and in the second one, a linear parametrization is used. In other words: in the first stage, the simple class of models is the ARX one and in the second stage a linear regression of the compressed artificial data sequences is used to fit θ. Even in such case, the final estimation rule is nonlinear. Indeed, the generation of the compressed artificial data in the first stage requires the use of the LS algorithm applied to the simulated data sequences $D^N i$, and this is by itself a nonlinear manipulation of data, see (2). Hence only the second stage is actually linear.
As a matter of fact, in some cases, such nonlinearity limited to the first stage of elaboration suffices for capturing the relationship between the unknown θ and the data $y(1),u(1),\ldots,y(N),u(N)$. In other cases, instead, introducing also a nonlinearity in the second stage (namely, taking h as a nonlinearly parameterized function of the compressed artificial data) is advisable and leads to better global results. □

Remark 4 (Two-stage and multi-value estimation): As it appears the two-stage approach relies on intensive simulations of the plant model and this fact can be computationally demanding. Yet, differently from other approaches, all these simulations have to performed once for all, through a single laboratory experiment. The result then is an explicit expression for $\widehat{f}$ (i.e. $\widehat{f}=\widehat{h}(\widehat{g}(\cdot))$ which can be easily applied over and over, for estimating all possible values of θ without any supervision from an human-operator. Thus, the two-stage approach is well-suited for multi-value estimation. □

IV. A BENCHMARK-EXAMPLE

Consider the following data-generation mechanism:

$$\begin{aligned} x_1(k+1) &= \theta\cdot x_1(k)+v_{11}(k) && \text{(4a)}\\ x_2(k+1) &= x_1(k)+\theta^2\cdot x_2(k)+v_{12}(k) && \text{(4b)}\\ y(k) &= \theta\cdot x_1(k)+x_2(k)+v_2(k), && \text{(4c)}\end{aligned}$$

where θ is an unknown real parameter in the range $[-0.9,0.9]$ and $v_{11}\sim WGN(0,1)$, $v_{12}\sim WGN(0,1)$, and $v_2\sim WGN(0,0.01)$ (WGN = White Gaussian Noise) are mutually uncorrelated noise signals. In all our experiments, system (4) was initialized with $x_1(0)=0=x_2(0)$.
In order to test the behavior of various approaches in a multi-value estimation problem, we extracted 800 values for the parameter θ uniformly in the interval $[-0.9,0.9]$ and, for each extracted value of θ, we generated $N=1000$ observations of the output variable y. Each time, the $N=1000$ observations were made available to the considered estimation algorithms which returned an estimate of the corresponding θ. Thus, for each estimation algorithm, we

obtained 800 estimates $\widehat{\theta}$ which then was compared with the corresponding 800 true values of θ.

A. Prediction Error approaches

The system output can be written as an ARMA process of the type[2]:

$$y(k) = \frac{N(z,\theta)}{z^2-(\theta+\theta^2)z+\theta^3}e(k), \quad e(k) \sim WN(0,\lambda^2),$$

where the numerator $N(z,\theta)$ is a second order polynomial whose coefficients depends on θ in a complex way.
Even in this simple case, due to the complexity of $N(z,\theta)$, an explicit expression for $\widehat{y}(k|\theta)$ is difficult to find and, moreover, the minimization of the loss function $\sum_{i=1}^{N}(y(i)-\widehat{y}(i,\theta))^2$ is extremely hard. Yet, the ARMA representation suggests that θ can be estimated by: 1. modeling the system output as

$$y(k) = \frac{z^2+c_1z+c_2}{z^2+a_1z+a_2}e(k), \quad e(k) \sim WN(0,\lambda^2),$$

where all numerator and denominator coefficients are free; 2. using the Prediction Error approach to identify the numerator and denominator coefficients $\widehat{c}_1$, $\widehat{c}_2$, $\widehat{a}_1$, $\widehat{a}_2$; 3. retrieving an estimate for θ according to the expression $\widehat{\theta} = \sqrt[3]{\widehat{a}_2}$ (note that if the true values were exactly identified then $\widehat{a}_2 = \theta^3$).
The obtained results are displayed in Figure 3 by plotting the estimates versus the parameter actual values. In other words, for each point in the figure, the x-coordinate is the extracted value for θ, while the y-coordinate is the corresponding estimate $\widehat{\theta}$ supplied by the used filter. Clearly a good estimator should return points concentrating around the bisector of the first and third quadrant.
As it appears the returned estimates are rather spread showing that this approach is not suitable for parameter estimation.

B. Kalman filters

In order to apply both EKF and UKF, system (4) was rewritten as:

$$\begin{aligned} x_1(k+1) &= x_3(k)\cdot x_1(k)+v_{11}(k) \\ x_2(k+1) &= x_1(k)+x_3(k)^2\cdot x_2(k)+v_{12}(k) \\ x_3(k+1) &= x_3(k)+w(k) \\ y(k) &= x_3(k)\cdot x_1(k)+x_2(k)+v_2(k), \end{aligned}$$

where x_3 is an additional state variable representing parameter a. Herein, we will report the simulation results obtained by taking as $w(k)$ a $WGN(0,10^{-6})$.
For each extracted value, θ was estimated as the 1-step ahead prediction of x_3 when 1000 values of the output y were observed, i.e. $\widehat{\theta} = \widehat{x}_3(1001|1000)$. In such a computation, both EKF and UKF were applied, but since the obtained results were quite similar, here the results for EKF are reported only.

[2]Note that the ARMA model is fed by a single white noise, while system (4) is affected by three exogenous disturbances; this is made possible by the well known spectral factorization theorem, [12].

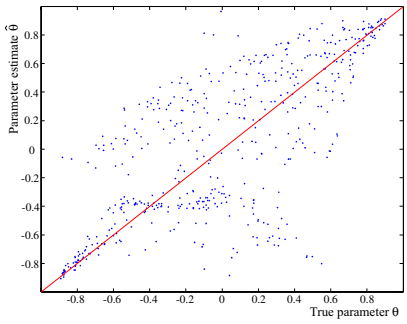

Fig. 3. Estimates of θ (h linearly parameterized).

Figures 4-5 display the result obtained in different operating conditions. Precisely, Figure 4 depicts the results obtained when EKF was used with the following initialization: $\widehat{x}_1(0) = \widehat{x}_2(0) = 1$, $\widehat{x}_3(0) = -0.4$, and

$$P(0) = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 2 \end{bmatrix} \tag{5}$$

($P(0)$ is the initial covariance of the estimation error). Figure 5, instead, displays the results obtained when

$$P(0) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 10^{-2} \end{bmatrix}. \tag{6}$$

As it appears, the filter behavior is quite different from

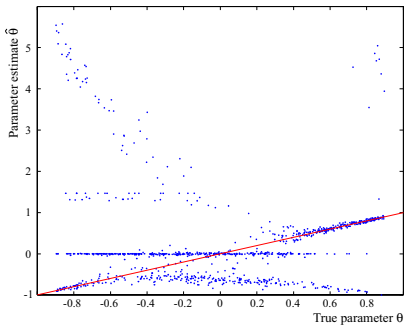

Fig. 4. Estimates of θ via EKF (large initial variance).

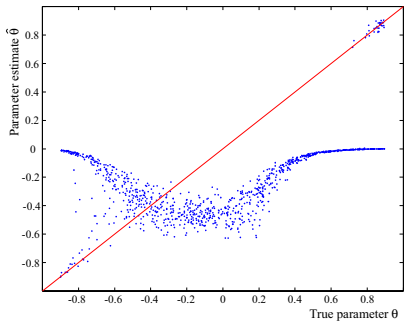

Fig. 5. Estimates of θ via EKF (small initial variance).

the optimal expected one. In many instances the estimate does not converge to the true value of θ. Furthermore, the filter behavior strongly depends on the choice of $\widehat{x}(0)$ and $P(0)$. Such a selection, however, is in general non trivial and

an human-operator supervision is needed to achieve sensible results. This makes Kalman filters ill-suited for multi-value estimation problems.

Perhaps it is worth noticing that further simulations were performed by changing the initialization of $\widehat{x}_3(0)$ (precisely, to -0.8, 0.4, and 0.8), but such simulations are not reported here due to space limitations. The results, however, were similar to those previously presented, and the conclusions drawn above remain still valid.

C. The two-stage approach

In order to apply the two-stage approach to system (4), $m = 500$ *new* values of θ were extracted uniformly from the interval $[-0.9, 0.9]$ and correspondingly 500 sequences of 1000 output values were simulated so as to construct the *simulated data chart*.

For each sequence $y^i(1),\ldots,y^i(1000)$, $i = 1,\ldots,500$, the compressed artificial data sequence was obtained by identifying through the least squares algorithm the coefficients $\alpha_1^i,\ldots,\alpha_5^i$ of an AR(5) model ($y^i(t) = \alpha_1^i y(t-1) + \cdots + \alpha_5^i y(t-5)$). The final estimator $\widehat{h}(\alpha_1^i,\ldots,\alpha_5^i)$, instead, was

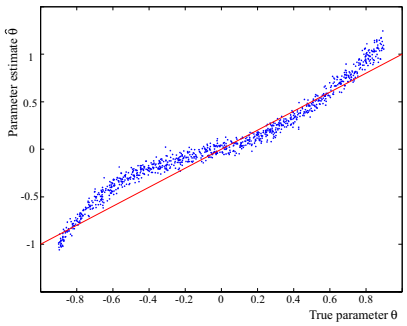

Fig. 6. Estimates of θ (h linearly parameterized).

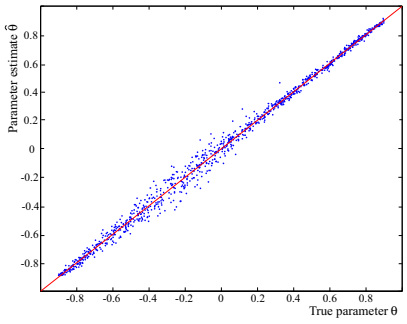

Fig. 7. Estimates of θ (h parameterized via neural networks).

computed by resorting, first, to a linear parametrization ($h = c_1\alpha_1^i + \ldots + c_5\alpha_5^i$)), with coefficients $c_1,\ldots,c_5$ estimated again by the least squares algorithm. As an alternative, $\widehat{h}$ was also derived by resorting to a neural network (to be precise, we considered an Elman neural network with 2 layers, 10 neurons in the first layer and one neuron in the second one; the network was trained with the 500 artificial observations by the usual back-propagation algorithm).

The obtained estimator was then applied to the 800 data sequences used also to test the other estimation approaches. The returned 800 estimates $\widehat{\theta}$ were compared with the corresponding 800 values of θ [3]. The performance of the obtained estimates can be appreciate in Figure 6 (h linearly parameterized) and in Figure 7 (h parameterized via neural networks).

As can be seen, the two-stage estimator works much better than other methods.

REFERENCES

[1] B.D.O. Anderson and J.B. Moore. *Optimal Filtering*. Prentice Hall, 1979.

[2] K.J. Åström. Maximum likelihood and prediction error methods. *Automatica*, 16:551–574, 1980.

[3] K.J. Åström and T. Bohlin. Numerical identification of linear dynamic systems from normal operating records. In *IFAC symposium on self-adaptive systems*, 1965.

[4] S. Bittanti and G. Picci, editors. *Identification, adpatation, learning – the science of learning models from data*. Springer-Verlag, Berlin, Gemrany, 1996.

[5] T. Bohlin. *Practical grey-box identification: theory and applications*. Springer-Verlag, London, UK, 2006.

[6] M. Boutayeb, H. Rafaralay, and M. Darouch. Convergence analysis of the extended kalman filter used as an observer for nonlinear deterministic discrete-time systems. *IEEE Transaction on Automatic Control*, 42(4):581–586, 1997.

[7] R.A. Fisher. On an absolute criterion for fitting frequency curves. *Mess. Math.*, 41:155, 1910.

[8] A. Gelb, Jr. J.F. Kasper, Jr. R.A. Nash, C.F. Price, and Jr. A.A. Sutherland. *Applied Optimal Estimation*. MIT press, 1974.

[9] M.S. Grewal and A.P. Andrews. *Kalman Filtering - theory and practice using MATLAB*. John Wiley & Sons, 2001.

[10] S.J. Julier and J.K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedingds of the IEEE*, 92(3):401–402, 2004.

[11] S.J. Julier, J.K. Uhlmann, and H.F. Durrant-Whyte. A new method for the nonlinear transformation of means andcovariances in filters and estimators. *IEEE Transaction on Automatic Control*, 45(3):477–482, 2000.

[12] T. Kailath, A.H. Sayed, and B. Hassabi. *Linear Estimation*. Prentice-Hall, 2000.

[13] L. Ljung. Asymptotic behavior of the extended kalman filter as a parameter estimator for linear systems. *IEEE Transaction on Automatic Control*, 24(1):36–50, 1979.

[14] L. Ljung. *System Identification: Theory for the User*. Prentice-Hall, Upper Saddle River, NJ, 1999.

[15] P.E. Morall and J.W. Grizzle. Observer design for nonlinear systems with discrete-time measurements. *IEEE Transaction on Automatic Control*, 40(3):395–404, 1995.

[16] H.B. Pacejka. *Tire and Vehicle Dynamics*. SAE International, 2005.

[17] K. Reif and R. Unbehauen. The extended kalman filter as an exponential observer for nonlinear systems. *IEEE Transaction on Signal Processing*, 47(8):2324–2328, 1999.

[18] D. Simon. *Optimal state estimation*. John Wiley & Sons, Hoboken, NJ, 2006.

[19] T. Söderström and P. Stoica. *System Identification*. Prentice-Hall, Englewood Cliffs, NJ, 1989.

[20] Y. Song and J.W. Grizzle. The extended kalman filter as a local asymptotic observer for nonlinear discrete-time systems. *J. Math. Systems Estim. Contr.*, 5(1):59–78, 1995.

[21] J.K. Su and E.W. Kamen. *Introduction to Optimal Estimation*. Springer, Englewood Cliffs, NJ, 1999.

[22] W. Sun, K.M. Nagpal, and P.P. Khargonekar. H_∞ control and filtering for sampled-datasystems. *IEEE Transaction on Automatic Control*, 38(8):1162–1175, 1993.

[23] E.A. Wan and R. van der Merwe. The unscented kalman filter. In S. Haykin, editor, *Kalman filtering and Neural Networks*, New York, NY, USA, 2001. John Wiley & Sons.

[3] Perhaps it is worth noticing that the validation was thus performed by using parameters and observations different from those used in the training phase of the two-stage approach.