Estimation of white-box model parameters via artificial data generation: a two-stage approach

Simone Garatti * **Sergio Bittanti** *

* *Dipartimento di Elettronica ed Informazione, Politecnico di Milano, piazza L. da Vinci 32, 20133 Milano, Italy.*
e-mail: {bittanti,sgaratti}@elet.polimi.it.

Abstract: A main problem encountered in control engineering is the estimation of unknown parameters appearing in the plant equations. In this paper, a new off-line method to perform such estimation is proposed. The method is based on the use of the plant simulator and on the generation of *artificial data* from which the relationship between the unknown parameter vector and available measurements is estimated. A simple example is used to illustrate how effective the method is in comparison to those methods based on the Kalman filtering techniques (Extended Kalman Filter and Unscented Kalman filter).

1. INTRODUCTION AND PROBLEM POSITION

In control engineering, a common experience of any designer is that, at the end of the plant modeling phase, a further step is required, that of assigning the value to a number of model parameters. Often, the physical meaning of the uncertain parameters is well known, but their precise value is difficult to state. The basic idea is then to resort to experimental data to tune the model and identify the parameter values (grey-box identification, Bohlin [2006], Blanke and Knudsen [2006], Walter and Pronzato [1996, 1997]. In this context, Kalman Filter (KF) is often used. As is well known, KF has been originally conceived to estimate the state of a linear dynamical system based on measurements of input and output variables. In principle, the method can be used to estimate the unknown plant parameters too, by introducing in the model additional state variables to represent the parameters (*fake* state variables). However, this procedure leads to a nonlinear system, for which state estimation is still open to research studies. The basic method is the celebrated Extended Kalman filter (EKF) where the original setting of Kalman approach is recovered by recursive linearizations around the current estimate of the state. Alternatively, one can resort to other Montecarlo-based approaches where the variances and covariances appearing in the formula for the Kalman filter gain are evaluated by means of sample counterparts suitably constructed. This leads to a number of nonlinear KF variants, in particular to the Unscented Kalman Filter (UKF). There is a huge literature on linear and non-linear estimation covering the range of some decades; we will quote here a few main references: Anderson and Moore [1979], Gelb et al. [1974], Grewal and Andrews [2001], Julier et al. [2000], Julier and Uhlmann [2004], Kailath et al. [2000], Morall and Grizzle [1995], Simon [2006], Su and Kamen [1999], Sun et al. [1993], Walter and Pronzato [1997], Wan and van der Merwe [2001].

The nonlinear KF approaches (EKF and UKF) are indeed the methods we have mostly adopted in the plant modeling problems which were posed to us through industrial cooperations. However, we have encountered difficulties, mainly due to the fact that the convergence of the parameter estimate is very sensitive to the tuning of the filter equations. In particular, the tuning of the initial covariance matrix of the estimation error, a basic ingredient of Kalman filtering, is critical and is usually the result of a number of trial and error attempts. In some sense, this way of proceeding is like sweeping the dust under the rug. One simply forgets about the problem of estimating the unknown parameters in the model to convert it into the equally difficult problem of estimating the covariance matrix of the fake state of the extended system. Anyhow, there are no guarantees that the accuracy of the obtained estimate of the unknown parameters is satisfactory.

Motivated by these experiences, we have worked out a novel off-line methodology in the last couple of years. The new method is based on an extensive use of the the plant simulator, as worked out from the mathematical model. Thanks to the simulator, it is possible to generate a number of input-output simulated data sequences, each sequence being associated with a specific value of the plant parameters. This enables constructing the *simulated data chart*, namely a map leading from the value of the parameters to a corresponding data sequence. In principle, once such chart has been constructed, the estimation problem could be tackled as the inverse problem of searching that value of the plant parameters for which the corresponding simulated data are "as close as possible" to the measured sequence of input-output data. This optimization problem, however, is hardly affordable due to the high dimensionality of the data sequences. We suggest therefore a two-stage procedure to reduce the complexity of the optimization problem to an acceptable level. To be precise, the information contained in all data sequences is compressed in short sequences of *artificial data*. This compression procedure is applied not only to the original data set of plant measurements, but also to all sequences of *data* generated by means of the plant simulator. In this way one can construct the *compressed artificial data chart*, assigning a sequence of compressed artificial data to each value of the plant parameters. Thanks to the limited dimensionality of the compressed artificial data sequences, the inverse problem of assigning a value of the parameters to each compressed artificial data sequence can be tackled. Thus, the parameter estimation problem can eventually be faced by associating to the sequence

⋆ Paper supported by the MIUR national project “Identification and adaptive control of industrial systems” and by CNR - IEEIT.

of compressed artificial data generated from the original measured data the corresponding value of the parameters as indicated by the inverse of the *compressed artificial data chart*. This methodology, referred to as *two-stage method*, provided much more satisfactory results, and we believe it is advisable to bring it to the attention of the community of control engineers. To better clarify the rationale of the method, we have chosen a simple example of a second order state space model with a single unknown parameter, for which we will compare the performance of our method with that achievable via EKF and UKF.

1.1 Formal problem position

The problem we deal with is precisely the following one. Consider a dynamical system depending on a parameter vector $\theta \in \mathbb{R}^q$. The system is denoted by $P(\theta)$ as in Figure 1. For

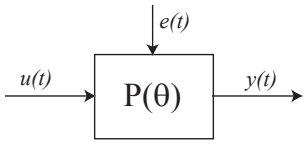

Fig. 1. The data generating system.

simplicity, in the sequel, we will focus on discrete time models only, with scalar input and output.
We suppose that the mathematical model for $P(\theta)$ is available, together with the associated simulator. The only unknown is the value of the parameter vector θ. The problem is that of estimating θ based on measurements from the plant.
To this purpose, we assume that the input and output of the plant are collected over a certain time interval so that a sequence of N input and output snapshots $\bar{D}^N = \{\bar{y}(1), \bar{u}(1), \ldots, \bar{y}(N), \bar{u}(N)\}$ is available. The issue is then how to exploit the information here contained in order to obtain a fair estimate of the uncertain parameter θ.

1.2 Structure of the paper

The paper is organized as follows. In Section 2, we preliminary study the mechanism of functioning of EKF and UKF as parameter estimators. By means of a simple example we will see that the estimation error may be remarkably spread. The new two-stage approach is introduced in Section 3, and then tested in the same example used to probe the applicability of Kalman filtering methods.
We anticipate that in this paper we propose the new method and test its effectiveness by simulations. As for EKF and other Kalman estimation techniques, we expect that the theoretical analysis is a nontrivial task. We do believe, however, that our approach can be a powerful and innovative tool to solve parameter estimation problems in physical models.

2. PARAMETER ESTIMATION WITH UKF AND EKF

Before introducing the new method, we are well advised to re-visit the issue of parameter estimation via Kalman filtering techniques. As is well known, the idea of EKF is to enlarge the state of the given model by considering the unknown parameter as an additional state variable, by means of the equation: $\theta(k+1) = \theta(k)$. Actually, to increase the responsiveness of the algorithm, the equation: $\theta(k+1) = \theta(k) + w(k)$ is normally preferred, where $w(k)$ is white noise with zero mean value and variance matrix W. We refer to such additional state components as the *fake state variables*. The state of the extended system can be then estimated by means of the Kalman filter, so obtaining also the estimate of the fake variables, i.e. of the parameter vector θ.

The main problems are: (i) the extended system is nonlinear, while the KF theory has been conceived for linear systems; (ii) the KF is a recursive estimation algorithm, and as such it requires an initialization; in particular, the value of the initial estimation error covariance matrix must be specified.

Point (i) is a main issue. Indeed, for linear systems, the analysis of the standard Riccati equation guarantees the stability of the KF along with the convergence of the estimation error covariance, provided that appropriate stabilizability and detectability conditions hold, see Bittanti et al. [1991].
When instead the system is nonlinear, the Kalman filtering equations can be used at the price of some approximation, and the general convergence results in the linear case do not hold anymore.
To be precise, both EKF and UKF are based on the same rationale, that is the iterated re-computation of the linear Bayes estimator around the current state estimate (as is well known, in Kalman filtering, the filter gain is given by the general Bayes estimator formula applied to the state estimation problem). The basic ingredients of the Bayes estimator are the cross variance between the output and the state and the variance of the output observations. In EKF these ingredients are (recursively) evaluated via system linearization by computing the Jacobians of the state and output equations around the current state estimate. In UKF, instead, the variance matrices are evaluated by means of sample counterparts. These counterparts are computed by letting the system equations evolve for one step starting from a cloud of a few points suitably generated around the current state estimate. In this way no system linearization is required and the estimated variance matrices suffer from a lower degree of approximation. In any case, both in EKF and UKF, the continuous re-computation of the filter gain introduces a sort of adaptation of the filter which should hopefully cope with the system nonlinearities. This could lead to the convergence of the state estimate notwithstanding the non-linearity of the underlying equations.
The question of convergence, however, is a well known critical issue of EKF. Indeed, there are celebrated (yet simple) examples showing the possible divergence/nonconvergence of EKF depending on the initialization of the filter (see e.g. Ljung [1979]). Despite many results on local convergence (Boutayeb et al. [1997], Ljung [1979], Reif and Unbehauen [1999], Song and Grizzle [1995], Zhou and Blanke [1989]), the global convergence of EKF is still an open problem. As for UKF, to the best knowledge of the authors, no analysis of convergence is available. In the following simple example we will show that UKF may provide unsatisfactory estimation performances too.

2.1 Example

Consider the time series y generated by the system:

$$x_1(k+1) = a \cdot x_1(k) + v_{11}(k) \tag{1a}$$

$$x_2(k+1) = x_1(k) + a^2 \cdot x_2(k) + v_{12}(k) \tag{1b}$$

$$y(k) = a \cdot x_1(k) + x_2(k) + v_2(k), \tag{1c}$$

where a is an unknown real parameter in the range $[-0.9, 0.9]$ while $v_{11} \sim WGN(0,1)$, $v_{12} \sim WGN(0,1)$, and $v_2 \sim WGN(0, 0.01)$ (WGN = White Gaussian Noise) are mutually uncorrelated noise signals.
For parameter estimation, system (1) is 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 the additional fake state variable representing parameter a. As for the fake state noise $w(k)$ we make the assumption that it is modeled as a further White Gaussian Noise uncorrelated with all other disturbances. The tuning of its variance is a rather complex issue, usually faced by a trial and error procedure; we have studied various cases but, for the sake of conciseness, we will only report the results obtained for a $WGN(0, 10^{-6})$. In all our experiments, system (1) was initialized with $x_1(0) = 0$ and $x_2(0) = 0$.
To probe the performance of EKF and UKF, we extracted 1000 values for the parameter a uniformly distributed in the interval $[-0.9, 0.9]$. For each extracted value, $N = 1000$ samples of the output signal y have been generated and used to estimate a by means of the 1-step ahead Kalman predictor of x_3, i.e. $\widehat{a} = \widehat{x}_3(1001|1000)$. Both EKF and UKF were applied for such

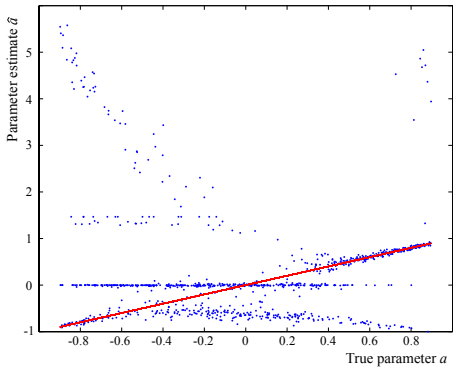

Fig. 2. Estimates of a via EKF (large initial variance).

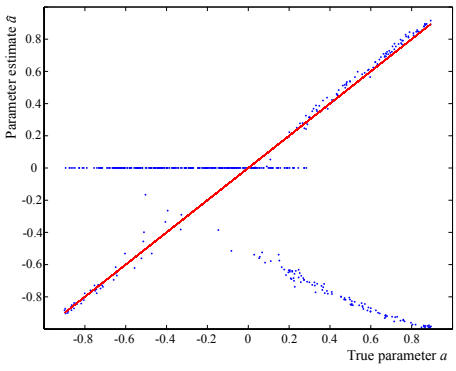

Fig. 3. Estimates of a via UKF (large initial variance).

estimation problem (the reader is referred to the literature for the updating equations of EKF and UKF, see Anderson and Moore [1979], Gelb et al. [1974], Grewal and Andrews [2001], Kailath et al. [2000], Simon [2006]).
Figures 2-5 display the result obtained in different operating conditions by plotting the obtained estimate versus the actual value of the parameter. In other words, for each point in the figure, the x-coordinate is the extracted value for a, while the y-coordinate is the corresponding estimate $\widehat{a}$ supplied by the used filter. Clearly a good estimator should return points concentrating around the bisector of the first and third quadrant.
As for the filter operating conditions, Figures 2 and 3 depict the results obtained when EKF and UKF, respectively, were initialized with $\widehat{x}_1(0) = 1$, $\widehat{x}_2(0) = 1$ and $\widehat{x}_3(0) = -0.4$. Moreover:

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

was taken as initial covariance matrix of the estimation error. Figures 4 and 5, instead, display the results obtained when

$$P(0) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 10^{-2} \end{bmatrix}. \tag{3}$$

As it appears, in many instances the estimate does not converge

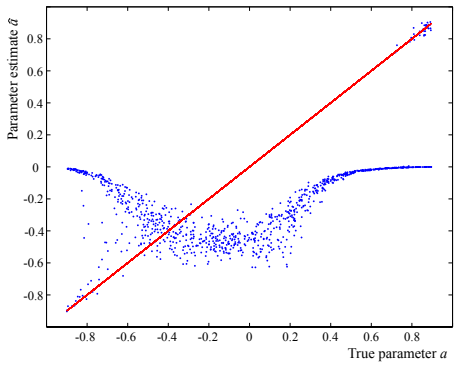

Fig. 4. Estimates of a via EKF (small initial variance).

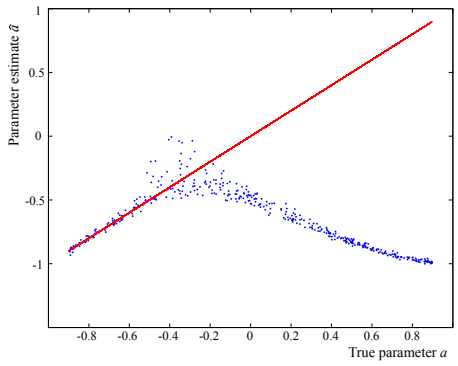

Fig. 5. Estimates of a via UKF (small initial variance).

to the true value of parameter a.
As it can be seen from Figures 2 and 3, when $P(0)$ is "large" as in the choice (2), $\widehat{a}$ form a large cloud. This means that, depending on the current realization of disturbances v_{11}, v_{12}, v_2, the estimate can converge to a variety of different values. This was already pointed out in Ljung [1979] for EKF. Somewhat more surprising is the fact that UKF suffers from the same drawback too.
When instead $P(0)$ is "small" as in the choice (3), the estimates obtained via EKF (Figure 4) are still rather spread whereas UKF provides estimates which are more concentrated. However, as it is apparent from Figure 5 the estimation error can be quite large for positive values of a. As a matter of fact, selecting a "small"

variance matrix means that parameter a is assumed to be located near the initial guess assumed in the filter initialization. In conclusion, the filters produce an acceptable estimate only when the true parameter a is in a relatively small neighborhood of -0.4. When instead a is far from -0.4, neither EKF nor UKF provide fair estimates of the actual value of a.
Further simulations were performed by changing the initialization of $\widehat{x}_3(0)$ (precisely, to -0.8, 0.4, and 0.8). The results were similar to those previously presented.

2.2 Final comments on UKF and EKF

As it appears from the simulations in Section 2.1 both EKF and UKF suffer from a major drawback, the possible non-convergence of the estimate to the true parameter value. Non-convergence occurs especially when the parameter to be estimated is subject to a large uncertainty and the correct value is away from its initial guess (a condition often encountered in application problems).
Furthermore, the behavior of EKF and UKF strongly depends on the choice of $\widehat{x}(0)$ and $P(0)$. In the Bayesian framework of Kalman filtering methods, these quantities should be selected as the mean and the variance matrix of the extended system state. However, in parameter estimation problems, θ is not a random variable in general, and the choice of its mean and variance is largely arbitrary and often obtained through heuristic trial and error procedures.
Summing up, the simulations above reported show that both EKF and UKF may be unsuitable for the estimation of an unknown parameter.

3. THE TWO-STAGE APPROACH

The new parameter estimation method is an off-line method fully developed in a non-Bayesian setting. The basic rationale is to resort to the plant simulator and to perform off-line intensive simulation trials in order to construct the relationship between the input/output data and the unknown parameter.
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 experiments one can work out a set of, say m, pairs $\{\theta_i, D_i^N\}$ as summarized in Table 1. Such set of data is referred to as the *simulated data chart*.

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

Suppose for a moment that, from such set of data, one can find the (nonlinear) map $\widehat{f} : \mathbb{R}^{2N} \to \mathbb{R}^q$ associating to each sequence of data a corresponding parameter vector as follows:

$$\widehat{f} \leftarrow \min_f \frac{1}{m} \sum_{i=1}^{m} \left\| \theta_i - f(y^i(1), u^i(1), \ldots, y^i(N), u^i(N)) \right\|^2. \quad (4)$$

Function $\widehat{f}$ can be seen as an appropriate estimator of the unknown parameter vector. Should $\widehat{f}$ be available, then the θ corresponding to the actual measurements $\overline{D}^N = \{\overline{y}(1), \overline{u}(1), \ldots, \overline{y}(N), \overline{u}(N)\}$ could be estimated as

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

Solving Problem (4) 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 Ljung [1999], Söderström and Stoica [1989].

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. The objective of the first step is to reduce the dimensionality of the estimation problem, by generating a new data chart 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 relationship between the original y-u data and θ 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 ($n \ll N$).

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 2. The compressed artificial data chart.

This leads to a new compressed artificial data chart constituted by the pairs $\{\theta_i, \widetilde{D}_i^n\}$, $i = 1, \ldots, m$, see Table 2.
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:

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

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 (Ljung [1999], Söderström and Stoica [1989]):

$$\begin{bmatrix} \alpha_1^i \\ \vdots \\ \alpha_n^i \end{bmatrix} = \Big[\sum_{t=1}^{N} \varphi^i(t) \varphi^i(t)^T \Big]^{-1} \cdot \sum_{t=1}^{N} \varphi^i(t) y^i(t), \qquad (5)$$

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

Remark 1. (Physical interpretation of the artificial data). While the original system $P(\theta)$ has a physical meaning, the simple model class selected to produce the compressed artificial data does not need to have any physical interpretation; 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 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 same 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 class of simple models and the corresponding identification algorithm.

Second stage. Once the compressed artificial data chart in Table 2 has been worked out, problem (4) becomes that of finding a map $\widehat{h} : \mathbb{R}^n \to \mathbb{R}^q$ which fits the set of m compressed artificial observations, i.e.

$$\widehat{h} \leftarrow \min_h \frac{1}{m} \sum_{i=1}^{m} \Big\| \theta_i - h(\alpha_1^i, \ldots, \alpha_n^i) \Big\|^2. \qquad (6)$$

Function minimization in (6) is reminiscent of the original minimization problem in (4). 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 (6) 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 link between the original data and θ is given by $\widehat{h}(\widehat{g}(\cdot))$, see Figure 6. When an actual input/ouput sequence is observed, say $\overline{D}^N =$

$$\underset{\text{data}}{\text{original}} \xrightarrow{\ \widehat{g}\ } \underset{\text{data}}{\text{artificial}} \xrightarrow{\ \widehat{h}\ } \theta$$

Fig. 6. The estimator function composition.

$\{\overline{y}(1), \overline{u}(1), \ldots, \overline{y}(N),\ \overline{u}(N)\}$, 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.

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_i^N, and this is by itself a nonlinear manipulation of data, see (5). 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. □

3.1 Example - continued

The two-stage approach was applied to the time series generated by system (1) so as to compare the performances in estimation with those achieved by EKF and UKF. To be precise, $m = 500$ values for $\theta = a$ were extracted uniformly from the interval $[-0.9, 0.9]$. Correspondingly, for each θ among this 500 extractions a simulated data sequence of 1000 output values was collected. Summing up, the simulated data chart consisted of 500 rows. Each row is associated to a value of θ and contains the corresponding sequence of 1000 snapshots.
For each sequence $y^i(1), \ldots, y^i(1000)$, the coefficients $\alpha_1^i, \ldots, \alpha_5^i$ of an AR(5) model ($y^i(t) = \alpha_1^i y(t-1) + \cdots + \alpha_5^i y(t-5)$) were identified by the least squares algorithm. We considered $\alpha_1^i, \ldots, \alpha_5^i$ for $i = 1, 2, \ldots, 500$ as compressed data. In other words, we replaced the 500 sequences of 1000 data each with another set of 500 sequences of 5 data each.
As for the second stage of the method, we have considered two possibilities. First, we have taken as function h a linear regression $h = c_1 \alpha_1^i + \ldots + c_5 \alpha_5^i$, with coefficients $c_1, \ldots, c_5$ obtained by the least squares algorithm. The second choice was to resort 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 compressed artificial observations composed by 5 samples each, by the usual back-propagation algorithm).

By composing the results of the two stages, the parameter estimator $\widehat{h}(\alpha_1^i,\ldots,\alpha_5^i)$ was eventually obtained. The performance of such estimator was tested by cross-validation. To this purpose, 1000 *new* values of $\theta = a$ were extracted in $[-0.9, 0.9]$. For each of these θ's, a sequence of $N = 1000$ outputs was generated from equation (1). The previously obtained estimator

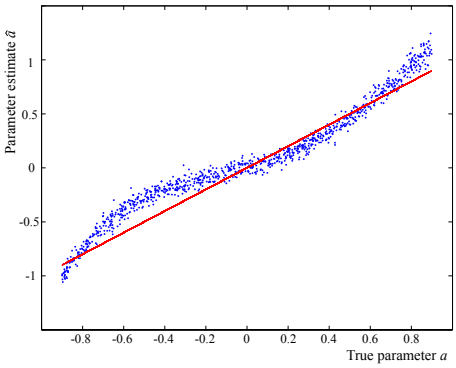

Fig. 7. Estimates of a (h linearly parameterized).

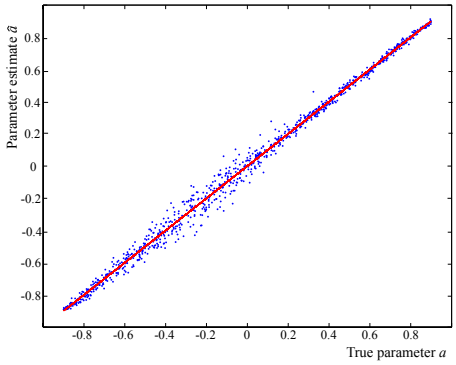

Fig. 8. Estimates of a (h parameterized via neural networks).

was used for each of these *new* sequences. The performance of the obtained estimates can be appreciate in Figure 7 (h linearly parameterized) and in Figure 8 (h parameterized via neural networks).

As can be seen, the two stage estimator works much better than EKF and UKF.

REFERENCES

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

S. Bittanti, A.J. Laub, and J.C. Willems, editors. *The Riccati equation*, volume 338 of *Communications and Control Engineering Series*. Springer-Verlag, Berlin, Gemrany, 1991.

M. Blanke and M. Knudsen. Efficient parameterization for grey-box model identification of complex physical systems. In *Proceedings of the 14th IFAC Symposium on System Identification (SYSID), Newcastle, Australia*, 2006.

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

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.

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

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

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

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.

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

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.

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

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.

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.

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

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

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.

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

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.

E. Walter and L. Pronzato. On the identifiability and distinguishability of nonlinear parametric models. *Mathematics and computers in simulation*, 32:125–134, 1996.

E. Walter and L. Pronzato. *Identification of parametric models from experimental data*. Springer, Englewood Cliffs, NJ, 1997.

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.

W.W. Zhou and M. Blanke. Identification of a class of nonlinear state space models using RPE techniques. *IEEE Transactions on Automatic Control*, 34:312–316, 1989.