Parameter estimation in induction motors: a comparison between the PE and the TS paradigm

S. Bittanti * **S. Garatti** * **M. Sarati**

* *Dipartimento di Elettronica e Informazione - Politecnico di Milano, piazza Leonardo da Vinci 32, 20133 Milano, Italia. (E-mails:* {bittanti,sgaratti}@elet.polimi.it, *web-sites:* http://home.dei.polimi.it/sgaratti/, http://home.dei.polimi.it/bittanti/*)*

Abstract: In this paper, we consider the problem of estimating unknown parameters of the model of an induction motor in a sensorless environment. A common practice consists in performing estimation through a series of bench tests, by letting the motor operate in steady-state. This way of proceeding, however, does not take into account the high sensitivity of some parameters to the motor operating condition. The issue, then, is to set-up an automatic *estimator*, able to to supply reliable estimates of the parameters from measurements taken from the actual functioning of the motor. Two different estimation paradigms are compared, namely the Prediction Error (PE) paradigm, which has become a standard in the practice of system identification, and the recently introduced Two-Stage (TS) paradigm. Advantages and drawbacks of such methods in the context of induction motors are spotted out by means of simulation experiments. It turns out that the TS method may offer a valid alternative to the PE method.

Keywords: Parameter estimation, Grey-box identification, TS method, Prediction Error Methods.

1. INTRODUCTION

The target of this paper is to discuss the use of parameter estimation techniques for the setup from experimental data of models of induction motors in a sensorless environment.

Like all motors, induction motors have a fixed stator and a mobile rotor, but, differently from the others, they are characterized by a poly-phase stator windings besides a three-phase rotor windings. Feeding the windings of the stator with a symmetrical alternating voltage, a magnetic field is generated which induces a current flow in the windings of the rotor. This current generates an additional magnetic field which tends to oppose to the stator current. While the motor is kept powered, this effort produces an electromagnetic torque which makes the rotor move.

Many different sets of equations describing the behavior of an induction motor can be found in the literature, Leonhard (1985), Krause (1986). In this paper, we consider the following well known fifth order model taking into account both the dynamics of the stator currents and of the rotor fluxes, see e.g. Marino et al. (1993):

$$\begin{aligned}
\frac{d\omega}{dt} &= \mu(\psi_a i_b - \psi_b i_a) - \frac{T_l}{J} \\
\frac{d\psi_a}{dt} &= -\alpha\psi_a - \omega\psi_b + \alpha M i_a \\
\frac{d\psi_b}{dt} &= -\alpha\psi_b + \omega\psi_a + \alpha M i_b \\
\frac{di_a}{dt} &= -(\frac{R_s}{\sigma} + \beta\alpha M) i_a + \frac{1}{\sigma}u_a + \beta\alpha\psi_a + \beta\omega\psi_b \\
\frac{di_b}{dt} &= -(\frac{R_s}{\sigma} + \beta\alpha M) i_a + \frac{1}{\sigma}u_b + \beta\alpha\psi_b - \beta\omega\psi_a
\end{aligned} \tag{1}$$

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

In this model, all variables refers to the fixed reference frame of the rotor. The state variables are the rotor speed ω, the rotor fluxes (ψ_a, ψ_b), and the stator currents (i_a, i_b). The currents i_a and i_b are also the measurable outputs of the system, while u_a and u_b, denoting the stator voltages, are the control inputs. T_l instead is the load torque and can be regarded as an external, not measurable, disturbance. The model depends on a number of parameters, that is the rotor moment of inertia J, the resistances (R_r, R_s) and the inductances (L_r, L_s) of the rotor and stator windings, and the mutual inductance M. In the model equations, the following notation was adopted: $\alpha = \frac{R_r}{L_r}$, $\beta = \frac{M}{\sigma L_r}$, $\mu = \frac{M}{JL_r}$, $\sigma = L_s(1 - \frac{M^2}{L_s L_r})$.

Model (1) provides a description of an induction motor which is accurate enough for most applications, as discussed in many contributions such as Lin et al. (2000), Feemster et al. (2001), Marino et al. (1993, 2000, 2005, 2008). Yet, the problem of determining the values of parameters J, R_r, R_s, L_r, and L_s, in order to fit the model to the real motor data, arises.

According to the literature, Leonhard (1985), Krause (1986), Feemster et al. (2000), the unknown parameters are typically estimated through a series of bench-tests by letting the system operate in steady-state with nominal inputs. The sought values for the parameters are then retrieved by processing measured signals, according to equations derived from a simplified model of the motor steady-state operating condition.

This way of proceeding typically returns estimates which are reliable for many situations of interest, so that it has become a standard in the practice of electrical motors. On the other hand, it may present serious limitations. To be precise, bench-tests basically assume that parameters are constant, without taking into account the high sensitivity of some of them to the working condition of the motor. For instance, it is well known that resistive parameters may vary with temperature up

to 100% of their nominal value. In turn, temperature depends in a nontrivial way on the motor state variables, the applied load torque, and other exogenous sources of variation, so that temperature is subject to unpredictable changes. In a sensorless framework (where e.g. the temperature cannot be measured by means of a dedicated sensor), and in those applications where the motor working conditions may have a considerable variation (see e.g. Lin et al. (2000), Feemster et al. (2001), Marino et al. (2000, 2005, 2008)), parameters must be estimated based on input/output measurements taken during the actual operating condition of the system. This calls for more sophisticated parameter estimation methods, and this paper aims at discussing advantages and drawbacks of a recently developed method in comparison with more standard techniques.

The estimation problem we want to tackle is a grey-box (sometimes also called white-box, Bohlin (2006)) identification problem, Ljung (1999), Bohlin (2006), being the model of the system completely given and the uncertainty restricted to the lack of knowledge of a number of parameters with physical meaning. Moreover, the system is continuous-time and highly nonlinear. Here, we consider two different estimation paradigms which seem to better fit with the problem at hand. Precisely, we will consider both the Prediction Error (PE) paradigm, which has become a standard in the practice of system identification, Ljung (1999), and the Two-Stage (TS) paradigm, which has been recently introduced in Garatti and Bittanti (2008), Bittanti and Garatti (2008) and which is specifically tailored to grey-box identification problems like the one discussed here.

The remainder of the paper is organized as follows. Section 2 provides a description of the PE and TS paradigms. Then, the experimental setting is given in Section 3, while the simulation results and the comparison between PE and TS are presented in Section 4. Some conclusions are eventually drawn in Section 5.

2. ESTIMATION METHODS

We will adopt the following notation. Letting t_0 be the sampling starting time and T the sampling period, $u(k) = [u_a(t_0 + kT)\ u_b(t_0 + kT))]'$ is the sampled input vector, while $y(k) = [i_a(t_0 + kT))\ i_b(t_0 + kT))]' + e(k)$ the sampled output vector corrupted by some measurements noise. We assume that N measurements are collected. The vector of unknown parameters will be denoted by $\theta \in \mathbb{R}^q$, where q is the number of uncertain parameters (for model (1) q may range from 1 to 5).

The estimation problem is solved by designing a suitable estimator, that is a function $\widehat{f} : \mathbb{R}^{4N} \to \mathbb{R}^q$ mapping the measured observation $D^N = \{y(1), u(1), \ldots, y(N), u(N)\}$ into an estimate $\widehat{\theta}$ for the unknown parameter vector θ. The quality of the designed estimator must be then evaluated based on its capability of making the estimation error $\|\widehat{\theta} - \theta\|$ small for θ ranging over a suitable domain, but also on its computational complexity, that is the time required to compute the estimate from available observations. This latter is a somewhat overlooked aspect which however is of great importance in practice, especially when estimation is required for real-time applications.

The problem of parameter estimation is of vital importance and it has been considered since the early days of IFAC. In the proceedings of the first World Congress in Moscow, 1960, one can already find a paper on this issue written by John Westcott, one of the signers of the IFAC resolution of 1956, Bittanti (2008). Further seminal works are Åström and Bohlin (1965), Mayne (1966). A solid estimation framework is the of Prediction Error (PE) methods, see Ljung (1999). Although the PE method has proven effective in a number of applications, its main criticality is that the estimator $\widehat{f}$ is implicitly defined, and this can possibly lead to a computationally demanding algorithm unless a severe deterioration of the estimation error is accepted, See next Section 2.1.

To prevent this drawback of PE, which is common to many other estimation approaches, a second estimation paradigm is introduced, namely the Two-Stage paradigm which aims at finding an *explicit* estimator by means of intensive simulations of the motor model. The TS paradigm thus operates in a complementary way with respect to the PE paradigm and it may offer a valid alternative in some cases, see Section 2.2.

2.1 Prediction Error method

In the Prediction Error method the loss function

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

is considered, where $\widehat{y}(i,\theta)$ is a predictor of the system output derived from the model equations and based on the input/output data up to time $i-1$. The estimate of θ is then obtained by minimizing $V(\theta)$, viz.

$$\widehat{\theta} = \arg\min_{\theta} V(\theta),$$

so that the estimator $\widehat{f}$ mapping observations into estimates is implicitly defined by the optimization problem itself. This latter is typically tackled by resorting to gradient-like methods.

The PE paradigm has been around for decades and has been analyzed in great detail. It has become the mainstream in black-box identification problems, but it applies to grey-box identification as well, with no conceptual twisting.

The main advantages of such approach are the solid theoretical background for consistency analysis, Ljung (1999), Söderström and Stoica (1989) and its general applicability. As for this latter aspect, observe that the gradient of the prediction error can be computed with generality once a model of the plant is available, possibly via numerical approximations.

However, the PE paradigm may suffer from computational drawbacks mainly due to the fact that $V(\theta)$ can be non-convex with many local minima which may trap the numerical solution far away from the true minimizer, Bohlin (1971), Söderström (1975), Ljung (1999). Ignoring this problem would lead to biased (inconsistent) estimates, so that minimization is carried out by means of multiple attempts, i.e. by running the numerical resolution method many times with different initializations chosen from a grid in the parameter space. As is clear, the finer the grid, the better the chance to converge to the global minimizer, but in this case one may run into the curse of dimensionality leading to a high computational complexity.

2.2 Two Stage paradigm

The TS method resorts to the system model to synthetically generate by *simulation* input/output data for a number of different values of the unknown parameter θ, say $\theta_1, \theta_2, \ldots, \theta_m$, extracted from the uncertainty domain. 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. By repeated simulation experiments one

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 TS method.

can eventually work out a set of m pairs $\{\theta_i, D_i^N\}$ as summarized in Table 1. Such set of data is referred to as the *simulated data chart*.

From the simulated data chart, $\widehat{f}: \mathbb{R}^{4N} \to \mathbb{R}^q$ is reconstructed as that map minimizing the mean square estimate error over simulated data. That is,

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

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

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

As is clear, solving Problem (2) requires the preliminary choice of a suitable class of functions $\mathscr{F}$ within which performing optimization. This is indeed a critical issue, since 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).

Due to the high dimensionality of the problem (f, indeed, depends upon $4N$ variables, normally a very large number if compared to the number of experiments m), a direct selection of $\mathscr{F}$ that achieves a sensible compromise between bias and variance errors is difficult to perform. In the TS method, hence, the selection of the family of functions $\mathscr{F}$ is split in two steps. This splitting is the key to determine 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 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 constituted by the pairs $\{\theta_i, \widetilde{D}_i^n\}$, $i = 1, \ldots, m$, see Table 2.

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.

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 $D_i^N = \{y^i(1), u^i(1), \ldots, y^i(N), u^i(N)\}$; then, the coefficients of this model, say $\alpha_1^i, \alpha_2^i, \ldots, \alpha_n^i$, are seen 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 a Multi-Input-Multi-Output (MIMO) ARX model (here, each α_k^i is a 2×2 matrix):

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

with a total number of coefficients $n = n_y + n_u$. The coefficients $\alpha_1^i, \ldots, \alpha_n^i$ of this model can be worked out by means of the Least Squares algorithm, Ljung (1999), Söderström and Stoica (1989).

Note that 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 finding a function $\widehat{g}: \mathbb{R}^{4N} \to \mathbb{R}^{4n}$ 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. Since as compressed artificial data we take the coefficients of a simple model, identified from D_i^N, function $\widehat{g}$ turns out to be defined by the chosen class of simple models together with the corresponding identification algorithm.

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

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

Function minimization in (3) is reminiscent of the original minimization problem in (2). However, being n small, the bias versus variance error trade-off is no more a big issue as before, and it is possible to resort to one of the many methods available in the literature for function fitting.

As for the choice of h one can e.g. select a suitable Neural Networks or a NARX model, while 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. At this stage, cross-validation can be profitably employed to select the network/NARX model order.

Use of the TS estimator. The TS estimator 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 compressed artificial data to the parameter estimate. While $\widehat{g}$ is chosen by the designer by selecting the intermediary identification algorithm in the first stage, in the second stage the designer chooses a suitable class of functions and $\widehat{h}$ is identified by fitting the extracted parameter values to the corresponding com-

pressed artificial data. The estimator $\widehat{f}$ mapping input/output data into the estimate for θ is eventually given by $\widehat{h}(\widehat{g}(\cdot))$, i.e. by the composition of $\widehat{g}$ and $\widehat{h}$. When the actual input/ouput sequence $D^N = \{y(1), u(1), \ldots, y(N), u(N)\}$ is observed, the corresponding unknown parameter can then be estimated as: $\widehat{\theta} = \widehat{h}(\widehat{g}(D^N))$. See Figure 1 for a pictorial representation.

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

Fig. 1. The estimator function composition.

Although the training of the TS estimator relies on intensive simulations of the model, this computational effort is relegated to the training of the estimator only, and it is worth stressing that the estimator eventually provided by the TS method is a map $\widehat{f} = \widehat{h}(\widehat{g}(\cdot))$ which is explicitly given and which permits to generate estimates, by evaluating $\widehat{f}$ in correspondence of the seen data sequence, at extremely low computational cost. This is in contrast with other estimation methods, where, being $\widehat{f}$ implicitly defined, each generation of an estimate requires a computationally demanding data processing.

Remark 1. (Sampling issues). An implementation choice the user is required to perform in TS method regards the sampling of the parameter vector θ. The idea is to randomly extract values for the unknown parameter vector θ over the range of interest. While in general uniform distribution can be considered, other probability distributions can be used if some a-priori information is available (see e.g. Tempo et al. (2005) for algorithms to perform random extractions from various probability distribution).
The issue, then, is how many samples need to be randomly extracted to ensure a given accuracy, that is to ensure that the empirical estimation error variance

$$\frac{1}{N}\sum_{i=1}^{N} \|\theta_i - h(\alpha_1^i, \ldots, \alpha_n^i)\|^2$$

is close enough to its probabilistic counterpart

$$E\|\theta - h(\alpha_1, \ldots, \alpha_n)\|^2.$$

If this was the case, indeed, the map reconstructed by optimizing the empirical cost would be satisfactory not only for the extracted θ_i, but also for other, unseen, instances of θ. In the context of the TS paradigm, the easiest way is to resort to cross-validation to a-posteriori assess the accuracy of the obtained estimator.
It is perhaps worth remarking, however, that the convergence of the empirical cost to the probabilistic one does not depend on the dimensionality of θ, i.e. convergence does not suffer from the curse of dimensionality. The reason is the same why Montecarlo methods are successful in computing multiple integrals: the overall effort for computing an integral in high dimensional spaces is the same as for a 1-dimensional integral. This is the magic of randomization, Tempo et al. (2005).

3. EXPERIMENTAL SETTING

The induction motor model in (1) was implemented and simulated in MATLAB and both the PE and the TS estimation methods were applied to retrieve the parameters from measurements of input and output signals. More precisely, we manipulated the voltages applied to stator windings by means of a square-wave three-phase inverter controller, while, according to our sensorless framework, we assumed that stator currents could be measured only. In order to take into account possible measurement errors, we added to each output a zero mean white noise with suitable variance (standard deviation $= 100mA$).
The assumed uncertainty was in line with Lin et al. (2000), Marino et al. (2008).
Among other parameters, the rotor resistance R_r is the one that is more affected by uncertainty due its sensitivity to temperature [1]. We hence assumed that R_r was unknown and that its range of variability was $[0.5R_{rn}, 2R_{rn}]$, where $R_{rn} = 3.3\Omega$ represents the resistance nominal value. This is typical for a number of motors and applications as revealed by experimental tests. For simplicity, all other parameters were supposed to be constant and equal to their nominal value (that is, $L_r = 0.375H$, $L_s = 0.365H$, $J = 0.0075kgm^2$, and $M = 0.34H$).
Besides R_r, we supposed that another source of uncertainty was present, namely the imprecise knowledge of the load torque T_l being this latter dependent on the particular use of the motor. To be precise, the range of variability for T_l was $[0.88T_{ln}, 5T_{ln}]$, where $T_{ln} = 5.104Nm$ is the nominal value for T_l.
Note that the load torque T_l is not properly a model parameter although it can be treated as such. T_l indeed is a constant disturbance input whose lack of knowledge makes the estimation problem even more difficult. As such, T_l was not required to be estimated. Rather, the target of our problem was that of estimating R_r robustly with respect to the values taken by T_l.

3.1 PE estimator

The PE method was implemented by resorting to function `pem` of the System Identification Toolbox of MATLAB, Ljung (2009). This function required to specify (by means of the `idnlgrey` data-structure) the output-prediction model of the motor. This latter can be derived from equations (1), taking into account the disturbances acting on the system: the load torque T_l and the error measurements on the output. Since computing the predictor model with respect to both these two disturbances is quite complicated, we decided to treat T_l as an unknown parameter and let the `pem` function estimate both R_r and T_l. This way, the model became a standard output error model from which computing the predictor model is trivial.
As for the initialization of the PE algorithm, 3 different initializations chosen at random from the parameter uncertainty domain were used so as to compromise between computational complexity and avoidance of local minima.

3.2 Training of the TS estimator

In order to apply the TS method, $m = 2500$ values for $\theta = R_r$ were extracted uniformly from the interval $[0.5R_{rn}, 2R_{rn}]$ and, correspondingly, we ran 2500 simulations of the motor model, each time adopting the control returned by the square-wave three-phase inverter controller as input and a constant value randomly chosen from $[0.88T_{ln}, 5T_{ln}]$ as torque load. This way we trained the TS estimator to be robust with respect to the imprecise knowledge of T_l.
By sampling at $1kHz$ the input and output signals, we obtained 2500 input/output sequences each $N = 1000$ samples long:

$$u^i(1), y^i(1), u^i(2), y^i(2), \ldots, u^i(1000), y^i(1000),$$

[1] It is worth noticing that the stator resistance R_s presents the same sensitivity to temperature. Yet, R_s can be trivially estimated being the measurements of both stator voltages and currents available.

$i = 1,2,\ldots,2500$. These sequences together with the 2500 extracted values for θ formed the simulated data chart.
For the generation of the compressed artificial data chart a MIMO ARX(3,3) model was considered:

$$y^i(t) = \alpha_1^i y^i(t-1) + \cdots + \alpha_3^i y^i(t-3) + \alpha_4^i u^i(t-1) + \cdots + \alpha_6^i u^i(t-3).$$

The parameters $\alpha_1^i, \alpha_2^i, \ldots, \alpha_6^i$, $i = 1,2,\ldots,2500$, obtained by performing identification over the sequence

$$u^i(1), y^i(1), u^i(2), y^i(2), \ldots, u^i(1000), y^i(1000)$$

constituted the compressed artificial data chart.
The final estimator $\widehat{h}(\alpha_1^i, \alpha_2^i, \ldots, \alpha_6^i)$ was instead derived by resorting to a feed-forward 4-layers neural network, with a total of 15 neurons in the hidden layers and 1 linear neurons in the output layer, Haykin (1998). The network weights were trained by the usual back-propagation algorithm. The order as well as the structure of the neural network was chosen by means of cross-validation.
The entire process for the training of the TS estimator took about 20 minutes on a standard 2.40 GHz dual-processor computer, and it produced an explicit estimator $\widehat{f}(\cdot) = \widehat{h}(\widehat{g}(\cdot))$ defined as the composition of the least squares algorithm and the trained neural network.

4. SIMULATION RESULTS

In order to test both the PE estimators and the TS estimator, we picked at random new 100 values for the uncertain parameter R_r, and correspondingly we ran new 100 simulations of the motor model with input generated by the square-wave three-phase inverter controller and torque load T_l extracted uniformly in $[0.88T_{ln}, 5T_{ln}]$. The 100 data sequences obtained by sampling input and output signals at $1kHz$ were made available to the PE and TS estimators so as to generate 100 PE estimates and 100 TS estimates. These estimates were eventually compared to the true values of the parameters so as to evaluate the performance of the two estimators.

Figures 2 and 3 depict the estimation results for the PE and TS methods respectively. In each figure, the estimates of R_r

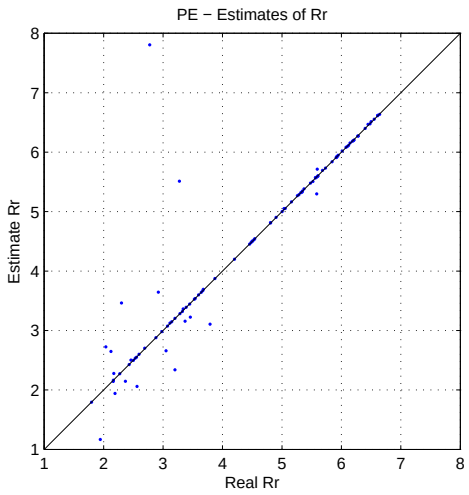

Fig. 2. Estimation results for the PE method.

as returned by the corresponding estimator are plotted against

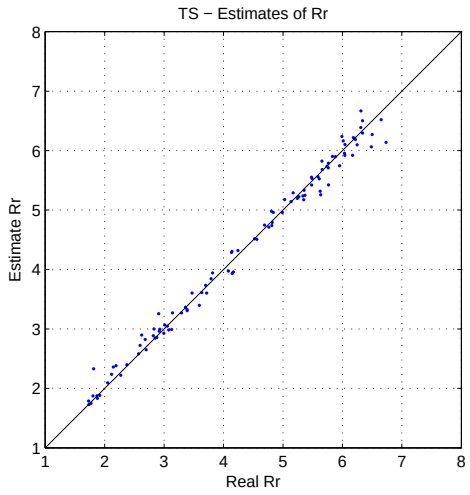

Fig. 3. Estimation results for the TS method.

the true parameter values. That is, for each point, the x-axis is the extracted value for the resistance parameter R_r, while the y-axis is the corresponding estimate supplied by the implemented estimation method. Of course, a good estimator must return points displaced nearby the plot bisector.
As it can be seen, both estimators are unbiased and provide satisfactory results for most of the cases. However, the behavior of PE and TS is quite different.
If on one hand in the 90% of the cases the PE and TS methods return comparable results (admittedly, with PE estimates which are closer to the true values in many cases), on the other hand, in the other 10% of cases, the PE estimates are completely unreliable with an estimation error that can even reach 5Ω. TS instead provides better overall results, with an estimation error which is more uniformly distributed and anyway bounded by 0.6Ω. The

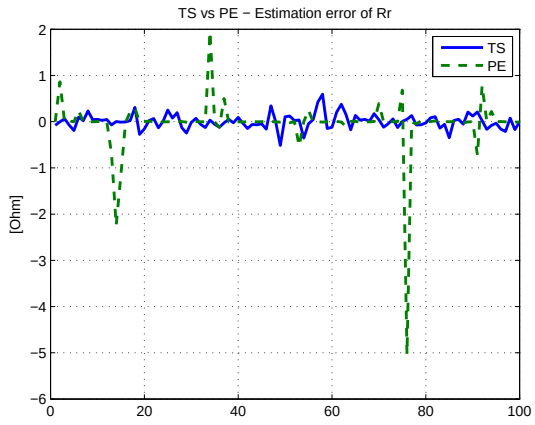

Fig. 4. Estimation errors - PE = dashed lined, TS = continuous line

difference between PE and TS can be also appreciated in Figure 4 where the estimation errors of PE and TS are plotted on the same graph.
The presence of unreliable PE estimates has to be ascribed to the problem of local minima, which in some cases trap the solution far away from the parameter true value in spite of the redundancy of initializations adopted. To reduce the number of incorrect estimates one could increase the number of initializations of the PE algorithm for each estimate. Yet, this would increase the PE computational effort, which is already quite

high if compared to that required by the TS estimator.
Computational time is indeed another important aspect to take into account in order to provide a fair evaluation of the performance of an estimator. In this respect, note that, on a standard 2.40 GHz dual-processor computer, the PE method took 1311 seconds (i.e. about 22 minutes) to generate the whole 100 estimates with an average of 13 seconds per estimate, while the TS method overall required 2.11 seconds only with an average of 0.021 seconds per estimate. Thus, the required computational effort for the TS estimator is 3 orders of magnitude below that for the PE method.
The overall performances of PE and TS are summarized in

	ARE	MRE	AET
TS:	0.0230	0.1201	0.021*s*
PE:	0.0352	1.0163	13*s*

Table 3. PE performances vs. TS performances

Table 3 where the following acronyms are used: ARE = Average Relative Error, MRE = Maximum Relative Error, AET = Average Estimation Time.

5. CONCLUSIONS

In this paper, the well known Prediction Error (PE) paradigm and the recently introduced Two Stage (TS) paradigm for parameter estimation have been evaluated in the context of grey-box identification of a model for an induction motor. Both the estimation error and and the estimator computational complexity have been taken into account. Although PE provides good results most of times, occasionally it returns completely unreliable estimates as due to the obnoxious problem of local minima. Multiple initializations may alleviate this problem, at the price, however, of increasing the already high computational complexity. The TS paradigm instead presents a better overall behavior, with an estimation error more uniformly distributed and in the worst case much smaller than in the PE method. As for the computational complexity, the TS estimator turns out to be 3 orders of magnitude faster than the PE method.

REFERENCES

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

Bittanti, S. (ed.) (2008). *Control Science Evolution – Proceedings of the 50th anniversary of Control Science in Italy*. The National Research Council, Milan, Italy.

Bittanti, S. and Garatti, S. (2008). Revisiting the basic issue of parameter estimation in system identification - a new approach for multi-value estimation. In *Proceedings of the 47th IEEE Conference on Decision and Control*. Cancun, Mexico.

Bohlin, T. (1971). On the problem of ambiguities in maximum likelihood identification. *Automatica*, 7, 199–210.

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

Feemster, M., Aquino, P., Dawson, D., and Behal, A. (2000). An online identification method for both stator and rotor resistances of induction motors without rotational transducers. *IEEE Transactions on Industrial Electronics*, 47, 842–853.

Feemster, M., Aquino, P., Dawson, D., and Behal, A. (2001). Sensorless rotor velocity tracking control for induction motors. *IEEE Transactions on Control Systems Technology*, 9, 645–653.

Garatti, S. and Bittanti, S. (2008). Estimation of white-box model parameters via artificial data generation: a two stage approach. In *Proceedings of the 17th IFAC World Congress*. Seoul, Korea.

Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation*. Prentice Hall, Upper Saddle River, NJ, USA.

Krause, P. (1986). *Analysis of electric machinery*. McGraw-Hill, New York, NY.

Leonhard, W. (1985). *Control of Electrical Drives*. Springer, Berlin, Germany.

Lin, Y., Fu, L., and Tsai, C. (2000). Non-linear sensorless indirect adaptive speed control of induction motor with unknown rotor resistance and load. *International Journal of Adaptive Control and Signal Processing*, 14, 109–140.

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

Ljung, L. (2009). *System Identification Toolbox™ 7 – User Guide*. The Mathworks Inc.

Marino, R., Peresada, S., and Valigi, P. (1993). Adaptive input/output linearizing control of induction motors. *IEEE Transactions on Automatic Control*, 38, 208–221.

Marino, R., Tomei, P., and Verrelli, C. (2000). A global tracking control for speedsensorless induction motors. *Automatica*, 40, 1071–1077.

Marino, R., Tomei, P., and Verrelli, C. (2005). A nonlinear tracking control for sensorless induction motors. *Automatica*, 41, 1071–1077.

Marino, R., Tomei, P., and Verrelli, C. (2008). An adaptive tracking control from current measurements for induction motors with uncertain load torque and rotor resistance. *Automatica*, 44, 2593–2599.

Mayne, D. (1966). Parameter estimation. *Automatica*, 3, 245–255.

Söderström, T. (1975). On the uniqueness of maximum likelihood identification. *Automatica*, 11, 193–197.

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

Tempo, R., Calafiore, G., and Dabbene, F. (2005). *Randomized algorithms for analysis and control of uncertain systems*. Springer-Verlag, London, UK.