Proceedings of the 31st Chinese Control Conference
July 25-27, 2012, Hefei, China

System Identification and Control: a fruitful cooperation over half a century, and more

Sergio Bittanti , Simone Garatti
Dipartimento di Elettronica ed Informazione, Politecni co di Milano, piazza L. da Vinci 32, 20133 Milano, Italy.
E-mail: {bittanti,garatti}@elet.polimi.it

Abstract: Control science is basically a model based discipline. The performance of control is determined by the accuracy of the model representing data. Therefore the successes of identification reflects into the successes of control. This is why identification and control have had a never-ending interplay over so many decades. In this paper we try to outline such interplay by making reference to one of the main problems encountered in control engineering modeling: the estimation of unknown or uncertain parameters in the plant equations.

Key Words: System Identification, Parameter Estimation, Identification for Control

1 Introduction and problem position

In the development of modern control science and technology, the first World Congress of the *International Federation of Automatic Control* (IFAC) represents a milestone. Held in Moscow in the summer of 1960, that congress was gifted with the participation of many outstanding control engineers, such as, in alphabetical order, G. Axelby, R. Bellman, G. Evangelisti, A.A. Feldbaum, A.F. Filippov, E.I. Jury, R.E. Kalman, N.N. Krasovski, C.T. Leondes, A. Lepschy, N.B. Nichols, E. Popov, A. Ruberti, G. Quazza, J. Song, T. Vamos, M. Thoma, J. Westcott, H. Zemanek, J.G. Ziegler, and many others, all hosted at the Ukraina Hotel. Interestingly enough, Norbert Wiener was also in Moscow during the congress days, as confirmed by Manfred Thoma in private conversations with the first author of this paper. Thoma and Wiener met various times in the lobby of the hotel. Surprisingly enough, there is no Wiener papers in the Congress Proceedings. However, by sure he gave a seminar in a set of three lectures organized as a side event of the congress, the three lecturers being Solomon Lefschetz, Rudolph Kalman, and, as already said, Norbert Wiener (this information was passed to the first author of this paper in a private communication with Kalman).

Among the fundamental contributions presented at that congress, there was the celebrated Kalman's paper "On the General Theory of Control Systems", [20],which had to produce a paradigm shift in control, as well as in many other engineering disciplines. In that paper, the derivation of the control law was based on a rigorous pattern starting from the mathematical model of the plant. Ever since, control had to became more and more a *model based discipline*.

Fifty years later, in the summer of 2011, the 18th IFAC World Congress was held in Milan, with an exceptionally high number of attendees, more than 2800. In the last day of the Congress, a panel session entitled "Plugging into the origins of IFAC: The IFAC World Congress of 1960" was held with a wide audience as the last plenary event. The video of that session is available at the web site ifac2011.org. The goal of this historical session was to evoke the early days of IFAC by the accounts of a team of distinguished scholars, including Rudolph Kalman. During his commemorative speech, Kalman said:

Fifty years is a fairly long time and one should certainly look back and see what the picture looks like at that point as regard the future and now, since the future is arrived, say something about what the final developments were that was started at the congress.

Then, Kalman remarked that

In any engineering application the success of control is determined by the accuracy with which the model represents data, and this leads to the realm of identification.

His conclusion was that, 50 years later, the field of identification is far from being settled.

In this paper, we would like to corroborate Kalman statement by illustrating the difficulties one can encounter even today to solve elementary identification problems. We will make reference to one of the simplest.

Parameter estimation problem in white box models: *the model of the system is completely given and the uncertainty is restricted to the lack of knowledge of a parameter appearing in it*.

This estimation problem is very frequently encountered in all fields of applications, as any control engineer knows very well. Some examples are in order.

Induction motor
A typical description of an induction motor consists of a state space model with 5 state variables: two rotor fluxes, two stator currents and the angular velocity. The model is nonlinear since various products of state variables appear in it. The parameters of the model are typically estimated through a series of bench-tests by letting the system operate in steady-state with nominal inputs. The 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 presents

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

serious limitations which may prevent its use in some cases. 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, the resistive parameters are subject to large variations with temperature. 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. [23, 10, 28, 29, 30]), parameters must be estimated based on measurements of input and output taken during the actual operating condition of the system. This leads to the need for more sophisticated parameter estimation methods other than bench-tests, and this paper aims at discussing advantages and drawbacks of some estimation methods developed by the system identification community. ■

Tyre parameters
A typical model of a tyre relating the lateral force at the tyre to the steering angle, is provided by the so-called magic formula of Pacejka, [32], which constitute a basic ingredient of most vehicle dynamic models. In this Pacejka's model, the dependence of the lateral force upon the steering angle involves various trigonometric expressions with a number of unknown parameters depending on the tyre size, the inflation pressure and other conditions. The issue is to estimate such parameters from lab data or from data collected during specific manoeuvres. ■

The parameter estimation problem has been studied in the systems and control community since long time. In the same Proceedings of the IFAC World Congress of 1960 in Moscow, one can find an article by John Westcott entitled "The Parameter Estimation Problem", [41], addressing exactly this problem. By the way, John Westcott, now in his nineties, is one of the signers of the IFAC resolution of 1956, where the creation of IFAC was promoted by a small team of gifted and far-seeing scholars. Actually Westcott is the only one among the 18 signers who is still alive. See also [5] for a survey on many advances in the field of identification.

For parameter estimation, there are two main approaches. The first one is based on least squares techniques in the prediction error identification framework (maximum likelihood methods in the statistical field), the second one is based on extended Kalman filter concepts. By the way, the subject of the lecture given by Kalman in the set of the three side seminars given at the IFAC World Congress of 1960 was filtering and prediction in the state space framework.

The aim of the present paper is that of surveying these existing estimation methods. By means of a test bed problem, we will show that they may fail to provide sensible results. Then, we present a new estimation paradigm, namely the two-stage (TS) approach, which has been recently introduced by the authors of this paper to prevent some of the drawbacks of existing methods, [4, 11, 12]. Clearly, our contention is not that the TS paradigm always outperforms previously proposed approaches, since each method has its own range of applicability, with successes and flaws. Yet, the proposed test bed problem reveals that the TS method may be a valid alternative. An application of the TS method to a more concrete estimation problem in induction motors is also given.

2 Parameter estimation: formal problem position

Let $P(\theta)$ be 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 $\theta \in \mathbb{R}^q$, as pictorially depicted in Figure 1. Here, $u(t)$ and $y(t)$ are the input and output measurable signals while $e(t)$ is a nonmeasurable exogenous input. For simplicity, we will assume that $u(t)$ and $y(t)$ take value in $\mathbb{R}$, i.e. $P(\theta)$ is SISO (single-input-single-output).

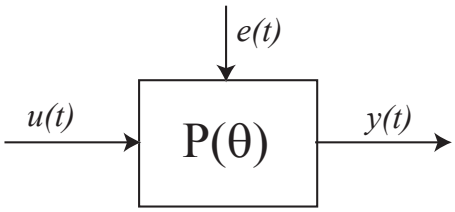

Figure 1: The data generating system.

We assume that an exact mathematical model (and a corresponding simulator) for $P(\theta)$ is available. The current value of parameter θ, however, is unknown and it has to be retrieved based on an experiment on the plant (white-box identification, [7]). To this purpose, the system behavior is observed for a certain time interval over which a number N of input and output observations $\bar{D}^N = \{\bar{y}(1), \bar{u}(1), \bar{y}(2), \bar{u}(2), \ldots, \bar{y}(N), \bar{u}(N)\}$ are collected (in case of continuous-time systems, observations has to be intended as sampled data points, e.g. $\bar{y}(i), \bar{u}(i) = \bar{y}(t_0 + iT), \bar{u}(t_0 + iT)$, where t_0 is the initial time and T is the sampling period). The issue, then, is how to exploit the information contained in the data in order to obtain a fair estimate of the uncertain parameter θ.

To be more concrete, we will consider a test bed problem where the mathematical model of system $P(\theta)$ is as follows:

$$\begin{aligned} x_1(k+1) &= \frac{1}{2}x_1(k) + u(k) + v_{11}(k) \\ x_2(k+1) &= (1-\theta^2)\sin(50\,\theta^2)\cdot x_1(k) - \theta \cdot x_2(k) + \\ &\quad + \frac{\theta}{1+\theta^2}\cdot u(k) + v_{12}(k) \\ y(k) &= x_2(k) + v_2(k), \end{aligned} \tag{1}$$

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 $v_2 \sim WGN(0, 0.1)$ (WGN = White Gaussian Noise) are mutually uncorrelated disturbances. We suppose that the initialization of system (1) is always $x_1(0) = 0 = x_2(0)$.
In order to test the behavior of various estimation approaches, we extracted 200 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 associated with an input u generated as $WGN(0,1)$ uncorrelated with the disturbances. The $N = 1000$ pairs of input/output observations are given to the estimation algorithms, producing an estimate $\widehat{\theta}$ for the corresponding θ. Thus, for each estimation algorithm, we obtain 200 estimates $\widehat{\theta}$ which then are compared with the corresponding 200 true values of θ.

3 Existing estimation paradigms revisited

3.1 Kalman Filtering

In Kalman filter based methods, see e.g. [1, 39, 14, 35], 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.
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, one resorts to nonlinear Kalman filtering.
Most common approaches are the Extended Kalman Filter (EKF), [1, 13, 14], and the Unscented Kalman Filter (UKF), [19, 40, 18]. For the EKF and UKF equations the reader is referred to the literature, see e.g. [35].
Apart from the difficulties one can encounter when the system is continuous-time and/or infinite dimensional, the major issue of EKF and UKF is that an initial guess for the initial estimation error mean and covariance matrix must be supplied. However, the convergence of the parameter estimate is very sensitive to the tuning of such initialization, and there are celebrated (yet simple) examples showing the possible divergence/nonconvergence (see e.g. [25]). In general, local convergence is achievable only, [25, 31, 38, 8, 33].
In order to obtain reasonable estimates, the initialization of the algorithm must be suitably tuned according to the current value of θ. When, however, no accurate a-priori information is available, such a tuning is performed by trial and error empirical attempts, with questionable findings.

In order to apply both EKF and UKF to estimate the unknown parameter in (1), the system was rewritten as:

$$\begin{aligned} x_1(k+1) &= \frac{1}{2}x_1(k) + u(t) + v_{11}(k) \\ x_2(k+1) &= (1 - x_3(k)^2)\sin(50\, x_3(k)^2)\cdot x_1(k) + \\ &\quad -x_3(k)\cdot x_2(k) + \frac{x_3(k)}{1+x_3(k)^2}\cdot u(t) + \\ &\quad +v_{12}(k) \\ x_3(k+1) &= x_3(k) + w(k) \\ y(k) &= x_2(k) + v_2(k), \end{aligned}$$

where x_3 is an additional state variable representing parameter θ. Herein, we will report the simulation results obtained by taking as $w(k)$ a $WGN(0, 10^{-6})$.
For each extracted value of θ in the range $[-0.9, 0.9]$, the estimate was obtained as the 1-step ahead prediction $\widehat{\theta} = \widehat{x}_3(1001|1000)$. Such a computation was carried over with the EKF, UKF algorithms.

To provide a graphical visualization of the obtained results, we plot the obtained estimates against the true parameter values. In other words, for each point in the figures to follows, the x-axis is the extracted value for θ, while the y-axis is the corresponding estimate $\widehat{\theta}$ supplied by the implemented estimation method. Clearly, the more points displaced nearby the bisector of the first and third quadrants, the better the estimation result.

Figures 2,3 display the result of EKF and UKF in different operating conditions. Precisely, Figure 2 depicts the

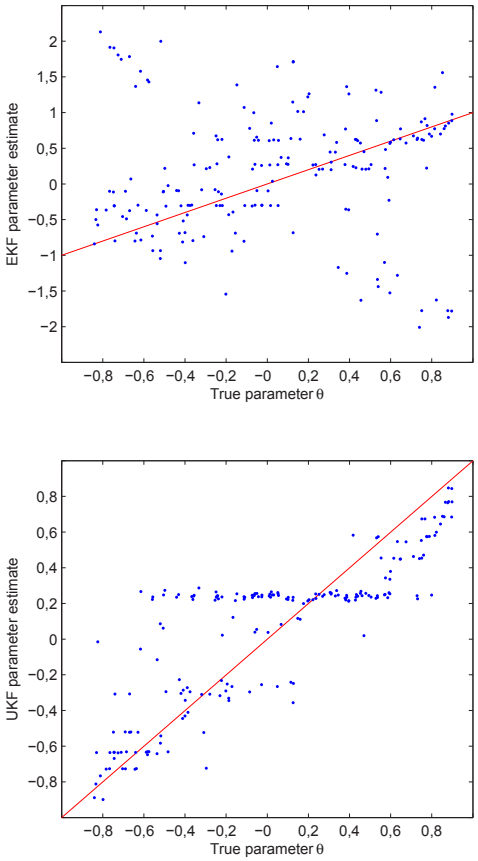

Figure 2: Estimates of θ vs. true parameter values (large initial variance) – EKF on the left, UKF on the right.

results obtained with the following initialization[2]: $\widehat{x}_1(0) = \widehat{x}_2(0) = 0$, $\widehat{x}_3(0) = 0$, and

$$P(0) = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.5 \end{bmatrix} \qquad (2)$$

($P(0)$ is the initial covariance of the estimation error). Fig-

[1] Often the additional equation takes the form $\theta(k+1) = \theta(k) + w(k)$ or $\dot{\theta}(t) = w(t)$ where w is white noise with suitable variance so as to improve the reactivity of the algorithm.

[2] Perhaps it is worth noticing that further simulations were performed by changing the initialization of $\widehat{x}_3(0)$ (precisely, to -0.8, -0.3, 0.3, and 0.8), but such simulations are not reported here due to space limitations. The results, however, were similar to those we have presented, and the conclusions drawn below remain still valid.

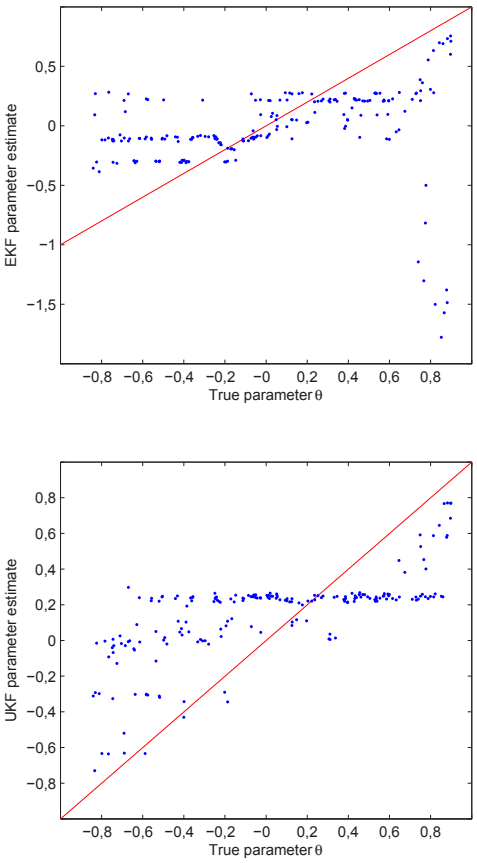

Figure 3: Estimates of θ vs. true parameter values (small initial variance) – EKF on the left, UKF on the right.

ure 3, instead, displays the results obtained when

$$P(0) = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 10^{-2} \end{bmatrix}. \tag{3}$$

As for the computational complexity, EKF took about 11 seconds to return the whole 200 estimates (with an average time of 0.055 seconds per estimate), while UKF required overall about 200 seconds (with an average time of 1 second per estimate).
As it appears, the EKF and UKF behavior is quite different from the optimal expected one. In many instances the estimate does not converge to the true value of θ. Furthermore, the estimator behavior strongly depends on the choice of $\widehat{x}(0)$ and $P(0)$, and, anyhow, local convergence can be achieved at most.

Another approach, still settled in the Kalman Filter realm, is the so-called Particle Filter (PF). PF basically reconstructs the a-posteriori probability distribution of θ by letting a cloud of possible parameter values evolve through the system equations, [35]. Differently from EKF and UKF, PF has the great advantage of guaranteeing the convergence of the estimate, [9, 17], and this is one reason for its increasing popularity. On the other hand, the PF estimation algorithm requires an intensive simulation of the model evolution before returning an estimate for the unknown parameters. Therefore, PF is very demanding from a computational point of view, and in general its potentiality clashes with the limitation on available resources.

As for the test bed problem, the PF paradigm results obtained with 1000 particles are depicted in Figure 4.
As can be seen, PF provides more satisfactory estimates. Although the performance can be further improved by increasing the number of particles, the actual bottleneck of the PF estimation method remains its computational complexity. Indeed, PF took 4675.89 seconds to generate the estimates with an average time of 23.38 seconds per estimate.

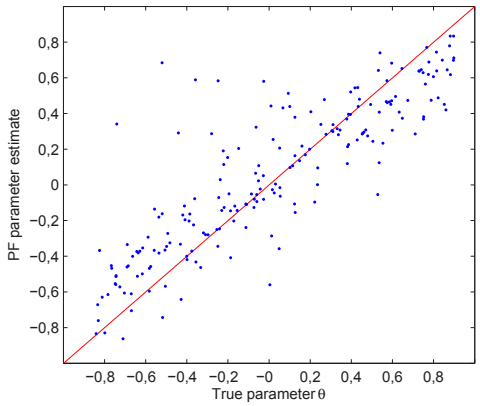

Figure 4: Estimates of θ vs. true parameter values for the Particle Filter estimator.

3.2 Prediction Error and Maximum Likelihood paradigms

In the Prediction Error (PE) approach, [24], the loss function

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

is introduced, where $\widehat{y}(i,\theta)$ is a predictor of the system output derived through the model equation for $P(\theta)$ and the available input/output data up to time $i-1$. The estimate of θ is obtained by minimizing $V(\theta)$, viz.

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

a problem which 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 white-box identification as well, with no conceptual twisting.
The main advantages of the PE approach are the solid theoretical background for consistency analysis and its general applicability. As for this latter, observe that the gradient of the prediction error can be computed with generality once a model of the plant is available, possibly by numerical approximations.
However, the PE paradigm may suffer from computational drawbacks as discussed in [7]. $V(\theta)$, indeed, is typically a non-linear non-convex function with many local minima which may trap the numerical solution far away from the true minimizer, [6, 36]. Ignoring this problem would lead

to biased (inconsistent) estimates. Hence, minimization is typically carried out by means of multiple attempts, i.e. by running the gradient-like method many times with different initializations chosen from a grid in the parameter space and then by choosing the estimate returning the smallest value for the loss function. As is clear, the finer the grid, the better the chance to converge to the global minimizer, but in this case one runs into the curse of dimensionality for which – in the words of [7] – "simulation would require supercomputers, and optimization an order of magnitude more".

The PEM estimation method was implemented for the test bed problem by resorting to the `idgrey` models of the `System Identification Toolbox` of `Matlab`, see [26].
Figure 5 depicts the results obtained with a single initialization of the PEM algorithm obtained by choosing a value at random for θ in the interval $[-0.9, 0.9]$ (note that initial states were known, $x_1(0) = 0 = x_2(0)$, and they needed not to be estimated). Overall, calculations took 18.23 seconds

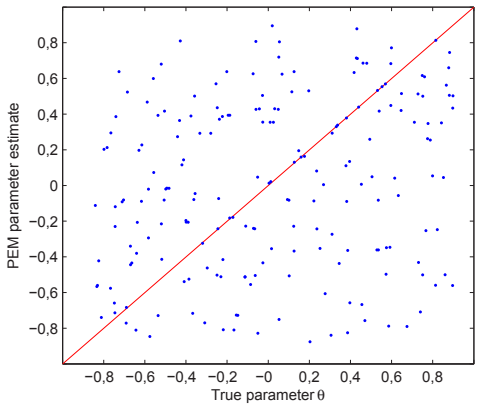

Figure 5: Estimates of θ vs. true parameter values for PEM estimator (1 initialization only).

with an average of 0.09 seconds per estimate. The returned estimates are rather spread, revealing the presence of many local minima trapping the PEM solution far away from the true parameter value.
Then, we ran the PEM algorithm with 5 different initializations first and with 10 initializations later, and then we chose the estimate returning the lowest loss. See Figure 6.
Although the performance is not the best one can hope for even with 10 initializations, PEM provides better and better results as the number of initializations is let increase. However, time complexity get worse since the algorithm with 5 initializations requires 94.28 seconds to generate the 200 estimates (with an average of 0.47 seconds per estimate), while the algorithm with 10 initializations took $179,14$ seconds (with an average of 0.9 seconds per estimate).

The maximum likelihood (ML) approach is another well known estimation method taken from statistics, [3, 2, 15].
ML amounts to computing the likelihood of possible values of θ given the observed data, and then finding the maximum of such likelihood function.
In case of complex systems, ML suffers from major drawbacks since it requires to reconstruct from the model equa-

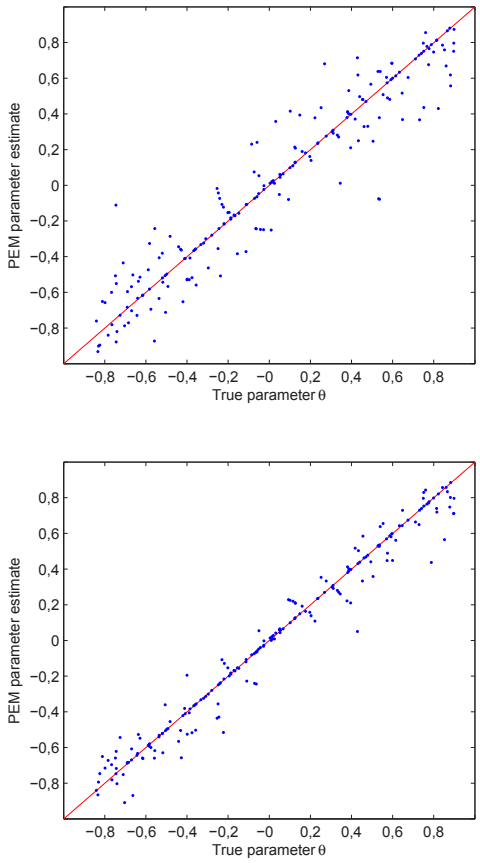

Figure 6: Estimates of θ vs. true parameter values for PEM estimator – 5 initializations on the left, 10 initializations on the right.

tions the probability density of observed data as a function of the parameter θ. This is a hard task requiring full probabilistic knowledge of disturbances, and it can be solved in closed form in few exceptional cases only (basically, for linear systems with Gaussian disturbances). On top of that, the maximization of the likelihood function presents the same criticality in terms of local minima as in the PE paradigm, see [6, 36]. Indeed, under suitable conditions, the negative logarithm of the likelihood is equal to the PE cost function, and hence the two optimization problems are equivalent.
Recently, an interesting ML approach based on Particle Filtering and the so called EM algorithm has been proposed in [34]. This approach gets rid of some of the drawbacks of ML, but still its computational complexity remains very critical.

4 A new paradigm in parameter estimation: the Two-Stage (TS) approach

Main Idea. The idea underlying the TS approach is to resort to off-line intensive simulation runs in order to explicitly reconstruct the estimator, i.e. a function $\widehat{f} : \mathbb{R}^{2N} \to \mathbb{R}^q$ mapping measured input/output data into an estimate for the parameter θ.
To be precise, we use the *simulator* of model $P(\theta)$ to generate input/output data for a number of different values of the unknown parameter θ chosen so as to densely cover a certain

range of interest. 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 forth and so on, so as to work out a set of,

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.

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*.
From the simulated data chart, the function $\widehat{f}$ is reconstructed as that map minimizing the estimation error over simulated data, i.e.

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

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

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

As is clear, solving (4) requires the preliminary choice of a suitable class of functions $\mathcal{F}$ within which performing optimization. This is indeed a critical issue, due to the high dimensionality of the problem (f depends upon $2N$ variables, normally a very large number if compared to the number m of experiments). Correspondingly, the notorious bias vs. variance dilemma arises, [24]: if $\mathcal{F}$ 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 $\mathcal{F}$ is a class of high-complexity functions, then the over-fitting issue arises (variance error).
In order to achieve a sensible compromise between the bias and variance errors, the two-stage approach is proposed. In this method, the selection of the family of functions $\mathcal{F}$ is split in two steps. This splitting is the key to select a proper family 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 problem, by generating a new data chart composed of m "short" sequences, each with $n \ll N$ points. 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 these artificial observations and parameter θ is identified. By combining the results of the two steps, the estimator $\widehat{f}$ is finally retrieved.
We give now more details on each of the two stages.

First stage. The first step consists in a compression of the information conveyed by input/output sequences D_i^N in order to obtain new 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 contained in a short sequence of n samples ($n \ll N$). This leads to a new compressed artificial data

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.

chart constituted by the pairs $\{\theta_i, \widetilde{D}_i^n\}$, $i = 1, \ldots, m$, see Table 2.
Each compressed artificial data sequence $\widetilde{D}_i^n$ can be derived from D_i^N by resorting to standard identification procedures. That is, one fits a simple model to each sequence $D_i^N = \{y^i(1), u^i(1), \ldots, y^i(N), u^i(N)\}$ and then takes 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 each $i = 1, 2, \ldots, m$, the data sequence

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

is concisely described by an ARX model:

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

with a total number of parameters $n = n_y + n_u$. The parameters $\alpha_1^i, \alpha_2^i, \ldots, \alpha_n^i$ of this model can be worked out by means of the least squares algorithm ([37, 24]),

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

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

and are used as compressed artificial data. It is worth noticing that, while $P(\theta)$ is a mathematical description of a real plant, the simple model class selected to produce the compressed artificial data does not 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. Hence, it does not matter if the ARX models do not tightly fit data sequences D_i^N; what really matters is that $\alpha_1^i, \alpha_2^i, \ldots, \alpha_n^i$ capture the variability of the θ_i. In this connection, observe that the choice of the ARX model order (which must be the same for all the data sequences in the simulated data chart) is not very critical and it can be performed by successive trials.
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 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 (4) becomes that of finding a map $\widehat{h} : \mathbb{R}^n \to \mathbb{R}^q$ which fits the m compressed artificial observations into the corresponding parameter vectors, i.e.

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

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, 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 linear function: $h(\alpha_1^i, \ldots, \alpha_n^i) = c_1\alpha_1^i + c_2\alpha_2^i + \ldots + c_n\alpha_n^i$, $c_i \in \mathbb{R}^q$, i.e. each component of h is just a linear combination of the compressed artificial data $\alpha_1^i, \alpha_2^i, \ldots, \alpha_n^i$. The parameters c_i appearing here can then 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, [16], or NARX models. The minimization in (6) can be performed by resorting to standard algorithms developed for these classes of nonlinear functions.

Use of the two-stage method. The two-stage method 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 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 compressed artificial data.
Once $\widehat{g}$ and $\widehat{h}$ are available, the estimator $\widehat{f}$ mapping input/output data into the estimate for θ is given by $\widehat{h} \circ \widehat{g} = \widehat{h}(\widehat{g}(\cdot))$, i.e. by the composition of $\widehat{g}$ and $\widehat{h}$ as pictorially represented in Figure 7. In this way, $\widehat{f}$ is

Figure 7: The estimator function composition.

explicitly given. When a new input/ouput sequence, say $\bar{D}^N = \{\bar{y}(1), \bar{u}(1), \ldots, \bar{y}(N), \bar{u}(N)\}$, is observed from the real plant, the corresponding unknown parameter θ is simply estimated as $\widehat{\theta} = \widehat{h}(\widehat{g}(\bar{D}^N))$.

Application to the test bed problem. According to the previous discussion, the TS estimator was obtained by means of a training over a set of simulation data. Once the training was terminated, the performance was tested against the same 200 experiments previously used for the other methods.
As for the training, $m = 1500$ new values of θ were extracted uniformly from the interval $[-0.9, 0.9]$ and correspondingly 1500 sequences of 1000 pairs of input/output values were simulated so as to construct the *simulated data chart*.
For each data sequence $y^i(1), u^i(1) \ldots, y^i(1000), u^i(1000)$, $i = 1, \ldots, 1500$, the compressed artificial data sequence was obtained by identifying through the least squares algorithm the coefficients $\alpha_1^i, \ldots, \alpha_{10}^i$ of an ARX(5,5) model ($y^i(t) = \alpha_1^i y(t-1) + \cdots + \alpha_5^i y(t-5) + \alpha_6^i u(t-1) + \cdots + \alpha_{10}^i u(t-5)$). The final estimator $\widehat{h}(\alpha_1^i, \ldots, \alpha_{10}^i)$, instead, was computed by resorting to a neural network with 10 inputs $(\alpha_1^i, \ldots, \alpha_{10}^i)$ and 1 output $(\widehat{\theta})$. The network was a standard feed-forward neural network with 2 layers (10 neurons in the first layer and 1 neuron in the second one which was also the output layer). The network was trained with the 1500 artificial observations by the usual back-propagation algorithm, [16]. Overall the training took 31.45 seconds.
The obtained estimator was then applied to the 200 data sequences previously used[3]. Again, the returned 200 estimates $\widehat{\theta}$ were compared with the corresponding 200 values of θ. The TS estimators required 1.95 seconds to generated the

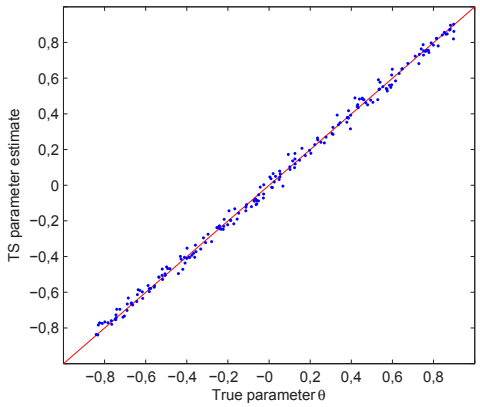

Figure 8: Estimates of θ vs. true parameter values for the TS estimator.

whole 200 estimates (that is, an average time of 0.01 seconds per estimate), while its performance can be appreciate in Figure 8.
As it can be seen, the two-stage estimator works much better than other methods, with time complexity some orders of magnitude below.

5 Parameter estimation in induction motors

In this section, we discuss the use of the TS method for the estimation from experimental data of the parameters of an induction motor in a sensorless environment.

[3] Perhaps it is worth to stress that this 200 values of θ and corresponding data sequences were not used in the training phase of the two-stage approach.

Model description and experimental setting

Many different sets of equations describing the behavior of an induction motor can be found in the literature, [22, 21]. Here, 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. [27, 23, 10, 28, 29, 30]:

$$\begin{array}{rcl} \frac{d\omega}{dt} & = & \mu(\psi_a i_b - \psi_b i_a) - \frac{T_l}{J} \\ \frac{d\psi_a}{dt} & = & -\alpha\psi_a - \omega\psi_b + \alpha M i_a \\ \frac{d\psi_b}{dt} & = & -\alpha\psi_b + \omega\psi_a + \alpha M i_b \\ \frac{di_a}{dt} & = & -(\frac{R_s}{\sigma} + \beta\alpha M) i_a + \frac{1}{\sigma} u_a + \beta\alpha\psi_a + \beta\omega\psi_b \\ \frac{di_b}{dt} & = & -(\frac{R_s}{\sigma} + \beta\alpha M) i_a + \frac{1}{\sigma} u_b + \beta\alpha\psi_b - \beta\omega\psi_a \end{array} \quad (7)$$

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, namely 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 above equations, the following notation was adopted:

$$\alpha = \frac{R_r}{L_r}, \quad \beta = \frac{M}{\sigma L_r}, \quad \mu = \frac{M}{J L_r}, \quad \sigma = L_s(1 - \frac{M^2}{L_s L_r}).$$

In our estimation problem, the stator voltages are manipulated by means of a square-wave three-phase inverter controller, while, according to a sensorless framework, we assume that the stator currents can be measured only. In order to take into account possible measurement errors, a zero mean white noise with standard deviation $= 100mA$ is added to each output. The assumed uncertainty was in line with [23, 30].

Among other parameters, the rotor resistance R_r is the one that is more affected by uncertainty due its sensitivity to temperature[4]. We hence assume that R_r is unknown and that its range of variability is $[0.5R_{rn}, 2R_{rn}]$, 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 are 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, another source of uncertainty is present in the model, namely the imprecise knowledge of the load torque T_l, being this latter user-dependent. The range of variability for T_l is $[0.88T_{ln}, 5T_{ln}]$, 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 is not required to be estimated. Rather, the target of our problem is that of estimating R_r from input/output data, robustly with respect to the values taken by T_l.

[4] 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.

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:

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

$i = 1, 2, \ldots, 2500$. 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:

$$\begin{array}{rcl} y^i(t) & = & \alpha_1^i y^i(t-1) + \cdots + \alpha_3^i y^i(t-3) + \\ & & \alpha_4^i u^i(t-1) + \cdots + \alpha_6^i u^i(t-3). \end{array}$$

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, [16]. 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.

Simulation results

In order to test 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 TS estimator so as to generate 100 estimates. These estimates are compared to the true values of the parameters in Figure 9. As in the test bed example, the estimates of R_r as returned by the TS estimator are plotted against the true parameter values, and the closer the points to the plot bisector the better the estimation results.

As it can be seen, the TS method provided unbiased estimates with an estimation error which is no greater than

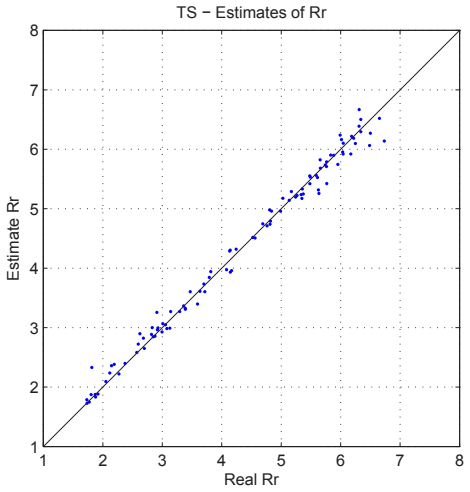

Figure 9: Estimation results for the TS method.

0.6Ω. As for the computational complexity, on a standard 2.40 GHz dual-processor computer, the TS method required 2.11 seconds only to generate the whole estimates, with an average of 0.021 seconds per estimate. The overall performances of the TS approach are summarized in Table 3 where the following acronyms are used: ARE = Average Relative Error, MRE = Maximum Relative Error, AET = Average Estimation Time.

	ARE	MRE	AET
TS:	0.0230	0.1201	$0.021s$

Table 3: TS estimation performances

6 Conclusions

The dawn of control science and engineering in modern times can be traced back to the mid of past century. Ever since the literature in the field has been flourishing more and more, its evolution being appropriately outlined in the Proceedings of the 18 IFAC World Congresses held so far. The interest for the field is now so wide that in the IFAC World Congress of Milan, 2011, we counted 3629 submitted papers, for a total of 7140 authors of 73 nationalities.

A basic conclusion one can draw is that control science and engineering is more and more a model based field. In turn this implies that estimation and identification is a fundamental pillar for control. Not surprisingly, system identification has been the subject of many conferences. Within IFAC, one can count 16 System Identification symposia, the first being held 42 years ago, Prague, The Check Republic (1970), only twelve years after the first IFAC World Congress of 1960. Then, the SYSID conference was hosted in Hague/Delft, The Netherlands (1973), Tblisi, URSS (1976), Darmstadt, Germany (1979), Arlington, USA (1982), York, UK (1985), Beijing, China (1988), Budapest, Hungary (1991), Copenhagen, Denmark (1994), Fukuoka, Japan (1997), Santa Barbara, USA (2000), Rotterdam, The Netherlands (2003), Newcastle, Australia (2006), St. Malo, France (2009), Brussels, Belgium (2012). The next one in the sequence is scheduled again in Beijing - October 2015.

Although many estimation and identification methods have been developed and studied in half century of research around the world, still there are very interesting unsettled problems, and this paper tries to bring into light one of them. Furthermore, new challenging problems will arise in the forthcoming future, due to the new arising fields of control applications, such as intelligent transportation systems, smart cities, systems biology, renewable energy systems, to name but a few.

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 S. Garatti. 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, 2008.

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

[6] T. Bohlin. On the problem of ambiguities in maximum likelihood identification. *Automatica*, 7:199–210, 1971.

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

[8] 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.

[9] D. Crisan and A. Doucet. A survey of convergence results on particle filtering methods for practitioners. *IEEE Transactions on signal processing*, 50(3):736–746, 2002.

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

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

[12] S. Garatti and S. Bittanti. Parameter estimation in the Pacejka's tyre model through the TS method. In *Proceedings of the 15th IFAC Symposium on System Identification*, Saint-Malo, France, 2009.

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

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

[15] E.J. Hannan and M. Deistler. *The statistical theory of linear systems*. John Wiley and sons, 1988.

[16] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, Upper Saddle River, NJ, USA, 1998.

[17] X.L. Hu, T.B. Schön, and L. Ljung. A basic convergence result for particle filtering. *IEEE Transaction on Signal Processing*, 56(4):1337–1348, 2008.

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

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

[20] R.E. Kalman. On the general theory of control systems. In *Proceedings of the 1st IFAC World Congress*, 1960.

[21] P. Krause. *Analysis of electric machinery*. McGraw-Hill, New York, NY, 1986.

[22] W. Leonhard. *Control of Electrical Drives*. Springer, Berlin, Germany, 1985.

[23] Y.C. Lin, L.C. Fu, and C.Y. Tsai. 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, 2000.

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

[25] 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.

[26] L. Ljung. *System Identification Toolbox™ 7 – User Guide*. The Mathworks Inc., 2009.

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

[28] R. Marino, P. Tomei, and C.M. Verrelli. A global tracking control for speedsensorless induction motors. *Automatica*, 40:1071–1077, 2000.

[29] R. Marino, P. Tomei, and C.M. Verrelli. A nonlinear tracking control for sensorless induction motors. *Automatica*, 41: 1071–1077, 2005.

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

[31] 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.

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

[33] 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.

[34] T. Schon, A. Wills, and B. Ninness. System identification of nonlinear state-space models. *Automatica*, 37(1):39–49, 2011.

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

[36] T. Söderström. On the uniqueness of maximum likelihood identification. *Automatica*, 11:193–197, 1975.

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

[38] 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.

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

[40] 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.

[41] J.H. Westcott. The parameter estimation problem. In *Proceedings of the 1st IFAC World Congress*, 1960.