Parameter estimation in the Pacejka's tyre model through the TS method [★]

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: In this paper, we consider the problem of estimating unknown parameters of the so-called Pacejka's model of a tyre from measurements of the car behavior. Differently from other estimation problems, the tyre parameters may change during the car life due to tyre aging, the variability of the inflation pressure, or many further reasons. Rather then solving a single estimation problem, thus, the issue is to set-up an automatic *estimator*, able to to supply reliable estimates of the Pacejka's parameters for any type of tyre and any type of operating condition. Existing methods for parameter estimation have been conceived for the estimation of the value taken by the parameter in a given functioning condition, and are not suitable for the problem here considered. In this paper we present a novel approach, the TS (two-stage) approach, specifically tailored to the problem of returning accurate estimates notwithstanding the tyres changeability. We compare its performances with those achievable with other parameter estimation techniques such as Prediction Error and Kalman Filter based methods. It turns that the TS approach offers significant improvements.

Keywords: Parameter estimation, White box identification, Pacejka's tyre model.

1. INTRODUCTION

One of the most critical aspects in modeling the dynamics of a vehicle is the determination of the lateral force generated by the interaction between tyres and the soil. The underlying physical phenomenon is rather complex and, for its description, one often resorts to empirical models, the most renowned of which is undoubtedly the *Pacejka's magic formula*, Pacejka [2005]. This is a non-linear function supplying the lateral force as a function of the steering angle. The formula contains a number of parameters, the value of which has to be tuned in order to distinguish between different kinds of tyres, with their own characteristics in term of size, constitutive material, inflation pressure, deterioration, etc. In order to complete the model for the vehicle behavior, thus, a further step is required, that of assigning a sensible value to the tyre Pacejka's parameters. In the Pacejka's magic formula, parameters have no clear physical meaning, and usually they are estimated from experimental data (white-box identification, Bohlin [2006]).

Estimating unknown parameters in a given plant from observed data is a basic problem in control engineering, Bittanti and Picci [1996], Bohlin [2006], Gelb et al. [1974], Ljung [1999], Söderström and Stoica [1989], and many techniques, such as those based on Kalman Filtering (KF), Anderson and Moore [1979], Grewal and Andrews [2001], Simon [2006], Su and Kamen [1999], or those based on Prediction Error Methods (PEM), Bohlin [2006], Ljung [1999], Söderström and Stoica [1989], or Linear Fractional Transformation (LFT), Wolodkin et al. [1997], have been developed to this purpose. All these techniques are well understood by theorists and practitioners, and can be effectively used to retrieve the parameters of a given tyre in a certain operating condition, corresponding to the available set of data.

During the car life, however, the operating condition may change due to the tyre aging, the variability of the inflation pressure, or many further reasons (including the fact that the car owner may decide to substitute the tyres!). Then, to guarantee the accuracy of the model, the parameters value must be re-estimated with new fresh data.

Thus, rather then solving a single estimation problem, the issue here is to set-up an automatic *estimator* to be embedded in an electronic device installed in the car. The estimator task is to supply reliable estimates of the Pacejka's parameters for any type of tyre and any type of operating condition.

Clearly, this conceptual framework poses some additional requirements:

- The estimation algorithm must be fully automatic as no human-supervision is allowed onboard.
- The computational effort must be low due to the limited computational resources.

As we will see, these requirements prevent the use of classical estimation approaches.

Conceptually, an estimator, or estimation algorithm, is nothing but a map from measured observations to the value of the parameters estimate. Typically, such a map is implicitly defined by the filter equation (KF based methods), or by the minimization of a mean square error cost function (PEM) or of a likelihood (LFT). All these approaches, however, may encounter serious drawbacks in the tyre parameters estimation problem.

More specifically, PEM and LFT call for the minimization of a non-linear cost function, a task which might be very hard. Indeed, simple gradient-based methods suffers from the obnoxious problem of local minima leading to biased (inconsistent) estimation unless the user supervises the minimization process.

[★] Paper supported by the MIUR national project "Identification and adaptive control of industrial systems" and by CNR - IEEIT. We are grateful to Dynasim for the support and to Carlo Sandroni for performing simulations.

On the other hand, more sophisticated gridding-based automatic methods run into the curse of dimensionality for which – in the words of Bohlin [2006] – “simulation would require supercomputers, and optimization an order of magnitude more”.
As for KF methods, Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) suffer from serious convergence problems if the initial estimate and the initial error covariance are not suitably initialized, Ljung [1979], Garatti and Bittanti [2008]. Moreover, such initialization depends on the actual value of the true parameters one want to estimate, so that it is often impossible to find an initialization ensuring convergence no matter what the true parameters are. Thus, both EKF and UKF require the adaptation of the “algorithm tuning knobs” through some human-supervision, every time the parameters value changes.
Another approach, still settled in the Kalman Filter setup, is the so-called Particle Filter (PF) which basically reconstructs the a-posteriori probability distribution of unknown parameters by letting a cloud of initial values for the unknown parameters evolving through the systems equation. Differently from EKF and UKF, PF has the great advantage of guaranteeing the convergence of the estimate independently of the initialization, Hu et al. [2008], 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. For this reason, PF is computationally demanding, and in general clashes with the limitation on available resources.

In this paper, a new estimation method, named the TS (*two-stage*) approach, is considered for the problem of tyre Pacejka’s parameters estimation. The TS approach has been recently introduced in Garatti and Bittanti [2008], Bittanti and Garatti [2008] and is specifically tailored to the problem of returning accurate estimates notwithstanding the tyres changeability. The basic rationale underlying the TS method is to off-line reconstruct the relationship linking the data to parameters through extensive simulation runs of the model. This is achieved thanks to an intermediary step aiming at the generation of a set of *artificial data*. Precisely, the procedure develops in two phases: the first one transfers the information contained in the original data into the artificial data, while the second one enables establishing the link between these last data and the unknown parameters. As a final product, the TS approach returns an explicit formula for the parameter estimator. The effectiveness of the proposed approach will be discussed by a comparison with other estimation techniques.

The paper is organized as follows. First, a model for the vehicle dynamics including interactions with tyres is described in Section 2. This will permit us to precisely formulate the estimation problem. For the sake of completeness, the TS approach is then resumed in Section 3, while Section 4 presents the experimental results obtained for the tyres parameters problem.

2. CAR MODEL DESCRIPTION AND PROBLEM FORMULATION

For simplicity we consider a car moving at constant speed V, and we model the car lateral dynamics only. In the sequel, x and y will denote the car longitudinal and lateral axis, respectively, while X and Y will denote an inertial coordinates frame. Our model input is the steering angle δ, i.e. the angle between the front wheels longitudinal axis and the car longitudinal axis (we suppose the steering angle is the same for both front wheels), while the output is the car body lateral acceleration denoted by a_y. $\overline{V}$ is the velocity vector in the inertial coordinates frame, while β is the car sideslip angle (i.e. the angle between the car longitudinal axis and $\overline{V}$) and ψ is the car yaw angle (i.e. the angle between the car longitudinal axis and the X axis). Clearly, $\overline{V} = Ve^{j(\beta+\psi)}$. In the following, we will assume that all angles are small enough to approximate function sin with its argument and cos with the constant 1.

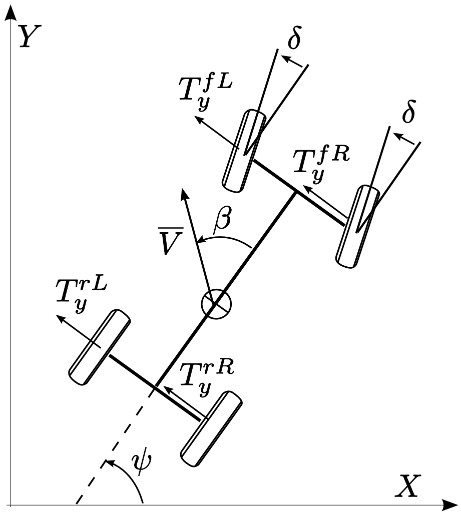

Fig. 1. The car model.

The model of the car lateral dynamics is obtained as a force and moment equilibrium (see Figure 1):

$$\begin{cases} ma_y = T_y^{fL} + T_y^{fR} + T_y^{rL} + T_y^{rR} \\ J\ddot{\psi} = (T_y^{fL} + T_y^{fR}) \cdot l_f - (T_y^{rL} + T_y^{rR}) l_r, \end{cases} \tag{1}$$

where T_y denotes the lateral forces generated by one tyre, while superscripts f, r, L, and R distinguish between *front*, *rear*, *Left*, and *Right* wheels. l_f and l_r indicate the distance between front and rear wheels and the car center of mass. Finally, m and J denotes the car mass and moment of inertia, respectively.
Under the current assumptions,

$$\begin{aligned} \frac{d\overline{V}}{dt} &= \frac{d}{dt}Ve^{j(\beta+\psi)} \\ &= [\dot{V}\cos\beta - V(\dot{\beta}+\dot{\psi})\sin\beta]e^{j\psi} + \\ &\quad +[\dot{V}\sin\beta + V(\dot{\beta}+\dot{\psi})\cos\beta]e^{j(\psi+\frac{\pi}{2})} \\ &\approx \left[\dot{V} - V(\dot{\beta}+\dot{\psi})\beta\right]e^{j\psi} + \left[\dot{V}\beta - V(\dot{\beta}+\dot{\psi})\right]e^{j(\psi+\frac{\pi}{2})}. \end{aligned}$$

It follows that

$$a_y \approx \dot{V}\beta - V(\dot{\beta}+\dot{\psi}) = -V(\dot{\beta}+\dot{\psi}),$$

being V constant. Then, (1) becomes

$$\begin{aligned} -mV(\dot{\beta}+\dot{\psi}) &= T_y^{fL} + T_y^{fR} + T_y^{rL} + T_y^{rR} \\ J\ddot{\psi} &= (T_y^{fL} + T_y^{fR}) \cdot l_f - (T_y^{rL} + T_y^{rR}) l_r. \end{aligned}$$

As for the lateral forces generated by tyres, we resort to the Pacejka’s magic formula, Pacejka [2005]:

$$T_y = D\sin\left\{C\arctan\left[B\overline{\alpha} - E\left(B\overline{\alpha} - \arctan\left(B\overline{\alpha}\right)\right)\right]\right\} + S_V$$

where

$$
\begin{aligned}
S_H &= p_{H1} + p_{H2} f_z + p_{H3}\gamma \\
S_V &= Q\left[(p_{V1} + p_{V2} f_z) + (p_{V3} + p_{V4} f_z)\gamma\right] \\
\bar{\alpha} &= \alpha + S_H \\
C &= p_{C1} \\
\mu &= (p_{D1} + p_{D2} f_z)(1 - p_{D3}\gamma^2) \\
D &= \mu Q \\
E &= (p_{E1} + p_{E2} f_z)\left[1 - (p_{E3} + p_{E4}\gamma)\mathrm{sign}(\bar{\alpha})\right] \\
K &= p_{K1} F_z \sin\left[2\arctan(Q/(p_{K2}\cdot F_z)))(1 - p_{K3}|\gamma|)\right] \\
B &= K/(C\cdot D)
\end{aligned}
$$

In this formula α is the wheel sideslip angle (i.e. the angle between the wheel longitudinal axis and the wheel velocity), while Q is the vertical load acting on the wheel. γ is the so called camber angle, here supposed to be constant, F_z the wheel nominal load (which is constant too), and $f_z = (Q - F_z)/F_z$ is the relative load. All other parameters appearing in this formula are the so called Pacejka's tyre parameters and their value determines the tyre response. Summarizing, letting θ be the vector of Pacejka's parameters, the Pacejka's magic formula is nothing but a parametric in θ nonlinear function of variables α and Q supplying as output the generated lateral force: $T_y = F_\theta(\alpha, Q)$.

In order to complete the model, it remains to show how to compute α and Q from β, ψ, and δ.

As for the wheel sideslip angle, we need first to compute the longitudinal and lateral speed of front and rear wheels. These are (right and left do not matter here):

$$
\begin{aligned}
V_x^f &= V\cos\beta \approx V \quad V_y^f = V\sin\beta + l_f\dot{\psi} \approx V\beta + l_f\dot{\psi} \\
V_x^r &= V\cos\beta \approx V \quad V_y^r = V\sin\beta - l_r\dot{\psi} \approx V\beta - l_r\dot{\psi}.
\end{aligned}
$$

The sideslip angles are then as follows:

$$
\begin{aligned}
\alpha^f &= \arctan V_y^f/V_x^f - \delta \approx \beta + l_f\cdot\dot{\psi}/V - \delta \\
\alpha^r &= \arctan V_y^r/V_x^r \approx \beta - l_r\cdot\dot{\psi}/V.
\end{aligned}
$$

In order to compute the vertical load acting on each wheel, instead, we need to model the car roll dynamics to take into account the vertical load shift. Precisely, let ϑ be the roll angle, g the gravitational acceleration, h the altitude of the center of mass, l the car semi-axis, and $N = m\cdot g$ the car vertical load. Then, we have that

$$
J_r\ddot{\vartheta} + C_r\dot{\vartheta} + K_r\vartheta = h\cdot m\cdot g\sin(\vartheta) - a_y\cdot h\cdot m\cdot\cos(\vartheta),
$$

where K_r, C_r and J_r are suitable constants. From the roll angle, then the lateral shift of the center of mass can be computed as $h\cos(\vartheta)$, so obtaining the following expression for the vertical loads on right and left wheels:

$$
Q^R = N\frac{l/2 - h\cos(\vartheta)}{l} \qquad Q^L = N\frac{l/2 + h\cos(\vartheta)}{l}.
$$

Thus, altogether, the car lateral dynamics can be modeled through a continuous time nonlinear system $P(\theta)$, depending on an uncertain parameter vector $\theta \in \mathbb{R}^q$. The system input u is equal to the steering angle and output y to the car lateral acceleration. See Figure 2. In order to retrieve the unknown value of the parameter vector θ, the system input and output are observed through an angular position sensor and an accelerometer for a certain time interval over which a number N of input and output observations $\bar{D}^N = \{\bar{y}(1), \bar{u}(1), \ldots, \bar{y}(N), \bar{u}(N)\}$ are collected. The issue then is how to build a suitable parameter estimator, i.e. a map $\widehat{f} : \mathbb{R}^{2N} \to \mathbb{R}^q$ which exploits the information contained in the observations $\bar{D}^N = \{\bar{y}(1), \bar{u}(1), \ldots, \bar{y}(N), \bar{u}(N)\}$ so as to produce fair estimates of the values taken by the uncertain parameter θ.

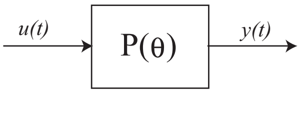

Fig. 2. The data generating system.

3. THE TS APPROACH FOR PARAMETER ESTIMATION

In this section we introduce the TS approach for estimators construction. The basic rationale is to resort to the plant simulator and to perform off-line intensive simulation runs in order to reconstruct from synthetic data the function $\widehat{f}$ mapping measured input/output data into an estimate for the parameter θ.

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

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

for $\theta = \theta_1$; N measurements

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

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

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

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

Should $\widehat{f}$ be found, then the 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 Problem (2) 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, the TS approach is proposed. In this method, the selection of the family of functions $\mathscr{F}$ is split in two steps. This splitting is the key to select a proper class $\mathscr{F}$ and, in turn, to obtain a good estimator $\widehat{f}$.

To be more precise, the objective of the first step is to reduce the dimensionality of the estimation problem, by generating a new data chart. The new chart is composed again of m

sequences; however, each sequence is constituted by a limited number n of samples ($n \ll N$). We will call such sequences *compressed artificial data sequences* and the corresponding chart the *compressed artificial data chart*. In the second step, the map between the compressed artificial observations and parameter θ is identified. By combining the results of the two steps, the map $\widehat{f}$ is finally unveiled.
We now 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 $\{\theta_i, \widetilde{D}_i^n\}$, $i = 1, \dots, m$, see Table 2.

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

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 $D_i^N = \{y^i(1), u^i(1), \dots, y^i(N), u^i(N)\}$; then, the parameters of this model, say $\alpha_1^i, \alpha_2^i, \dots, \alpha_n^i$, are seen as compressed artificial data, i.e. $\widetilde{D}_i^n = \{\alpha_1^i, \dots, \alpha_n^i\}$.
To fix ideas, we suggest the following as a typical method for the generation of compressed artificial data. For each $i = 1, 2, \dots, m$, the data sequence

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

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 $\alpha_1^i, \dots, \alpha_n^i$ of this model can be worked out by means of the least squares algorithm (Ljung [1999], Söderström and Stoica [1989]):

$$\begin{bmatrix} \alpha_1^i \\ \vdots \\ \alpha_n^i \end{bmatrix} = \Big[\sum_{t=1}^{N} \varphi^i(t) \varphi^i(t)^T \Big]^{-1} \cdot \sum_{t=1}^{N} \varphi^i(t) y^i(t), \qquad (3)$$

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

Remark 1. (Physical interpretation of the artificial data). While $P(\theta)$ is a mathematical description of a real plant, based on the physics of the problem, the simple model class selected to produce the compressed artificial data does not need to have any physical meaning; this class plays a purely instrumental and intermediary role in the process of bringing into light the hidden relationship between the unknown parameter and the original collected data. □

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

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

$$\widehat{h} \leftarrow \min_h \frac{1}{m} \sum_{i=1}^{m} \Big\| \theta_i - h(\alpha_1^i, \dots, \alpha_n^i) \Big\|^2. \qquad (4)$$

Function minimization in (4) is reminiscent of the original minimization problem in (2). 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: $h(\alpha_1^i, \dots, \alpha_n^i) = c_1 \alpha_1^i + \ldots + c_n \alpha_n^i$, $c_i \in \mathbb{R}^q$, i.e. each component of h is just a linear combination of the compressed artificial data $\alpha_1^i, \dots, \alpha_n^i$. 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 (4) 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 TS 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 TS estimator. Once function $\widehat{g}$ has been chosen and function $\widehat{h}$ has been identified, the function $\widehat{f}$ mapping input/output data into the estimate for θ is given by $\widehat{h}(\widehat{g}(\cdot))$, see Figure 3. When an actual input/ouput sequence is observed, say $\bar{D}^N = \{\bar{y}(1), \bar{u}(1), \dots, \bar{y}(N), \bar{u}(N)\}$, the corresponding unknown parameter can then be estimated as: $\widehat{\theta} = \widehat{h}(\widehat{g}(\bar{D}^N))$.

$$\begin{tikzcd} \begin{matrix}\textit{original}\\ \textit{data}\end{matrix} \arrow[r, "\widehat{g}"] & \begin{matrix}\textit{artificial}\\ \textit{data}\end{matrix} \arrow[r, "\widehat{h}"] & \widehat{\theta} \end{tikzcd}$$

Fig. 3. The estimator function composition.

As previously discussed, viewing this data vs. θ map as the composition of $\widehat{g}$ and $\widehat{h}$ is the key to transform a numerically intractable problem into an affordable one.

Remark 3. (Construction of the TS estimator). As is apparent, the TS approach relies on intensive simulations of the plant model and this fact can be computationally demanding. Yet, differently from other approaches, all these simulations have to performed once for all, by using the car simulator. The result then is an explicit map $\widehat{f}$ (i.e. $\widehat{f} = \widehat{h}(\widehat{g}(\cdot))$ which can be easily applied over and over, for estimating all possible values of θ at low computational cost (i.e. the cost of evaluating a function $\widehat{f}$ for a given data sequence) and without any supervision from an human-operator. Thus, the TS approach is well-suited for the Pacejka's parameters estimation problem as studied in this paper. □

4. EXPERIMENTAL RESULTS

The car model described in Section 2 has been implemented in `Dymola/Modelica` and then simulated through `Matlab`. We then applied the TS approach in order to construct an estimator able to retrieve the Pacejka's tyre parameters from the measurements of the car lateral acceleration while the car, moving at constat speed $V = 10$ m/s, is steered as in Figure 4. This manoeuver lasts about 10 seconds and corresponds to performing a *chicane*. In order to take into account measurement errors, we added a zero mean white noise with suitable variance 0.01 m/s^2 to the lateral acceleration a_y returned by the simulator. A typical trend of the lateral acceleration (for a particular type of tyres) is shown in Figure 5. Finally, for simplicity, we supposed that the Pacejka's tyre parameters were the same for the four tyres of the car (i.e. the four tyres mounted on the car are of the same type with same aging and pressure condition), and that among all Pacejka's parameters, only p_{K1} and p_{K2} were subject to changeability (while all others were fixed). In other words, the vector θ to be estimated is constituted by $\theta_1 = p_{K1}$ and $\theta_2 = p_{K2}$. This choice was dictated by the fact that the lateral force as calculated by means of the Pacejka's magic formula is much more sensitive to variation of parameters p_{K1} and p_{K2} than all other parameters (this can be easily verified by simple experiments). Typically, p_{K1} ranges between -42.83 and -27.32, while p_{K2} between -1.22 and -0.98.

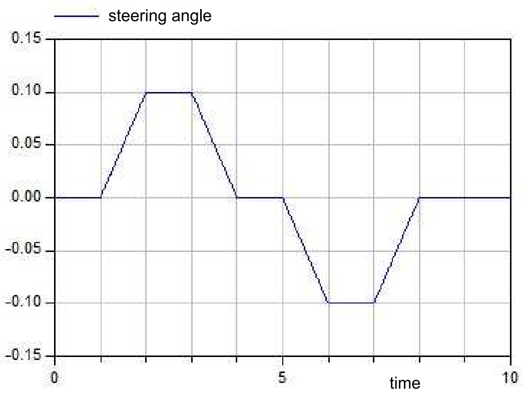

Fig. 4. Standard steering angle trend.

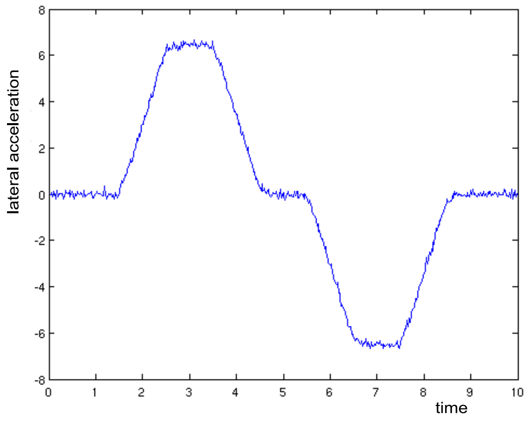

Fig. 5. Measured lateral acceleration.

In order to apply the TS method, $m = 2000$ values for θ were extracted uniformly from the rectangle $[-42.83, -27.32] \times [-1.22, -0.98]$ and, correspondingly, we run 2000 simulations of the car model, each time adopting the steering angle singal in Figure 4 as input. By sampling input and output signals with a period of 0.02 s, we then obtained 2000 sequences $u^i(1), y^i(1), \dots, u^i(500), y^i(500)$, $i = 1, \dots, 2000$, 500-samples long each, which, together with the 2000 extracted values for θ, formed the simulated data chart.

For each data set $u^i(1), y^i(1), \dots, u^i(500), y^i(500)$, the compressed artificial data sequence was obtained by identifying through the least squares algorithm the coefficients $\alpha_1^i, \dots, \alpha_8^i$ of an ARX(4,4) model $y^i(t) = \alpha_1^i y^i(t-1) + \cdots + \alpha_4^i y^i(t-4) + \alpha_5^i u^i(t-1) + \cdots + \alpha_8^i u^i(t-4)$. Such a choice for the model order was derived by a sequence of trials and errors.

The final estimator $\widehat{h}(\alpha_1^i, \dots, \alpha_8^i)$ was instead derived by resorting to a feed-forward 2-layers neural network, with 20 neurons in the hidden layer and 2 linear neurons in the output layer. The network weights were trained by the usual back-propagation algorithm. Again, the order as well as the structure of the neural network was also chosen by means of a trials and errors procedure.

The entire process for the derivation of the TS estimator took about 20 minutes on a standard 2.40 GHz dual-processor computer, and it produced an explicit estimation map $\widehat{f} = \widehat{h} \circ \widehat{g}$ defined as the composition of the least squares algorithm and the trained neural network.

In order to test the performance of the so obtained TS estimator, we applied it to 800 new data sequences, generated from 800 fresh values for parameter θ. Then, 800 estimates $\widehat{\theta}$ were then computed by simply evaluating $\widehat{f}$ for each data sequence. This validation process took few seconds only.

The performance of the TS estimator can be appreciated in Figures 6 and 7 where the estimates $\widehat{\theta}_1$ and $\widehat{\theta}_2$ of the first and second parameter are compared with the true values of θ_1 and θ_2. More precisely, Figure 6 [Figure 7] depicts in the x-axis the value of the first [second] parameter, while in the y-axis the returned estimate. Thus, each point in the graph corresponds to a pair true-parameter vs. returned-estimate, and the more the

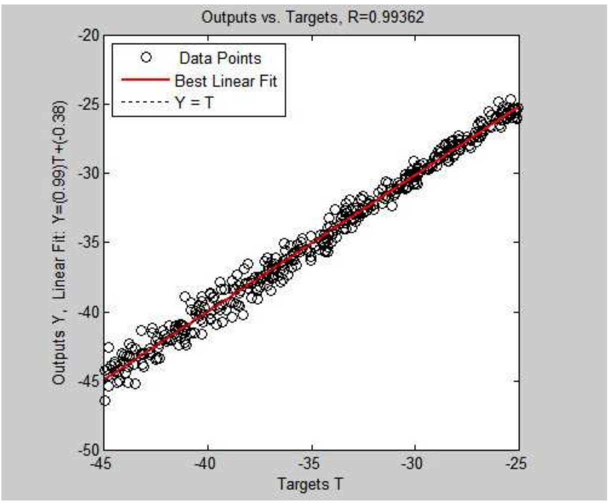

Fig. 6. Estimates of θ_1 as returned by the TS estimator (x-axis) vs. corresponding true values (y-axis).

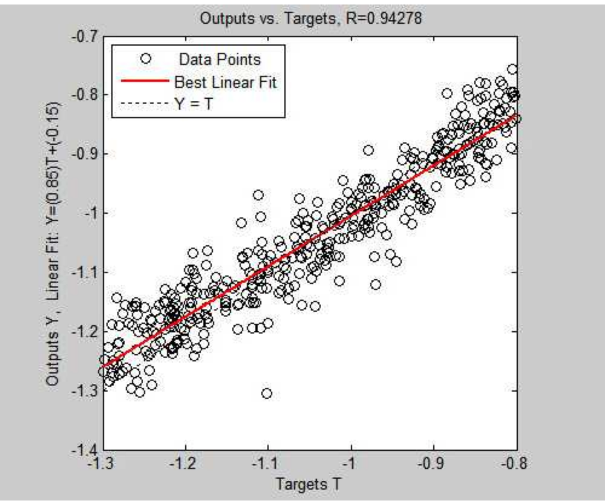

Fig. 7. Estimates of θ_2 as returned by the TS estimator (x-axis) vs. corresponding true values (y-axis).

points concentrate around the graph bisector (here plotted as a dashed line), the better the estimation performance. To ease the result visualization, we have also drawn a continuous line representing an optimal linear fit of the points in the graph.

As it appears, the TS estimator returns fairly accurate estimates, especially for the first parameter. In this respect, we note that the first parameter is one order of magnitude more important than the second one in determining the lateral force supplied by the Pacejka's magic formula, and this reflects into an easier identifiability of θ_1.

For the sake of comparison, we also applied the Particle Filter estimation algorithm to the first 50 data sequences out of the 800 we used for validating the TS approach. The estimates were computed by letting evolving a cloud of 500 particles through the filter equations. As for the Particle Filter equations, the reader is referred to the literature (see e.g. Simon [2006]).

Figures 8 and 9 plot the estimates $\widehat{\theta}_1$ and $\widehat{\theta}_2$ returned by Particle Filter vs. the true parameters value. As it appears, the PF estimates are consistent although much more scattered around the graph bisector. Furthermore, note that the PF algorithm required 5 hours to compute each estimate (the computation of all the 50 estimates took about 250 hours). This has to be compared with the few seconds required by the TS estimator for computing 800 estimates.

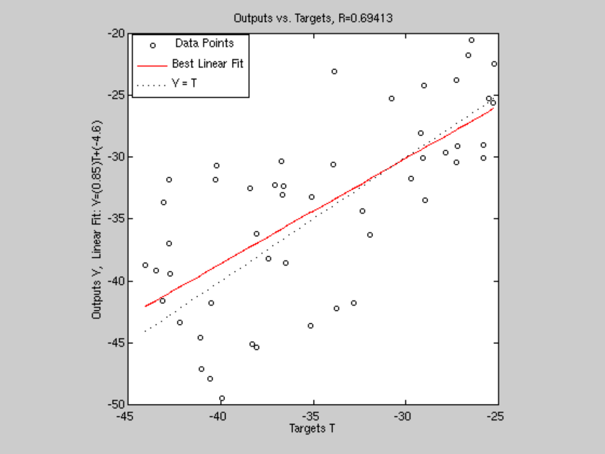

Fig. 8. Estimates of θ_1 as returned by the Particle Filter (x-axis) vs. corresponding true values (y-axis).

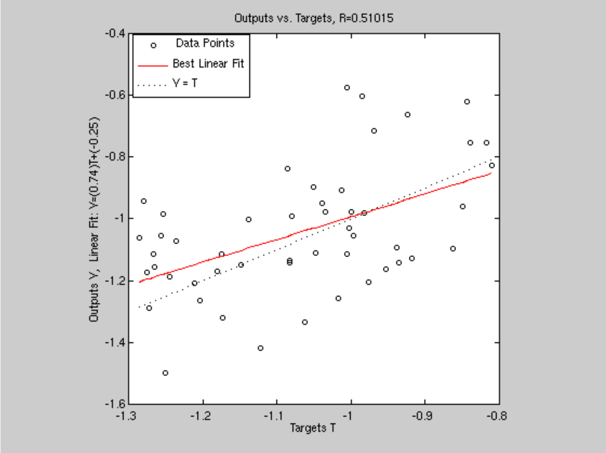

Fig. 9. Estimates of θ_2 as returned by the Particle Filter (x-axis) vs. corresponding true values (y-axis).

REFERENCES

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

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.

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

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

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, South Korea*, 2008.

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.

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.

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

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.

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

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

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

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

G. Wolodkin, S. Rangan, and K. Poolla. An lft approach to parameter estimation. In *Proceedings of the American Control Conference (ACC)*, 1997.