Preprints of the 20th World Congress
The International Federation of Automatic Control
Toulouse, France, July 9-14, 2017

Impulse Response Constrained LS-SVM modeling for Hammerstein System Identification ⋆

Ricardo Castro-Garcia *, **Oscar Mauricio Agudelo** *,
Johan A. K. Suykens *

* *KU Leuven, ESAT-STADIUS, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium. (ricardo.castro@esat.kuleuven.be, mauricio.agudelo@esat.kuleuven.be, johan.suykens@esat.kuleuven.be).*

Abstract: Hammerstein systems are composed by a static nonlinearity followed by a linear dynamic system. The proposed method for identifying Hammerstein systems consists of a formulation within the Least Squares Support Vector Machines (LS-SVM) framework where the Impulse Response of the system is incorporated as a constraint. A fundamental aspect of this work is that the structure of the Hammerstein system allows to obtain an impulse response that approximates the linear block while LS-SVM models the nonlinearity. When the resulting model is trained, the regularization capabilities of LS-SVM are applied to the whole model. One of the main advantages of this method comes from the fact that while it incorporates information about the structure of the system, the solution of the model still follows from a simple linear system of equations. The performance of the proposed methodology is shown through two simulation examples and for different hyper-parameter tuning techniques.

Keywords: Impulse Response, Machine Learning, LS-SVM, System Identification, Hammerstein Systems, Nonlinear Systems, SISO.

1. INTRODUCTION

For the modeling of nonlinear systems an important subject are the block structured nonlinear models. Many block structured models have been introduced in the literature (Billings and Fakhouri, 1982) and a commonly used nonlinear model structure is the Hammerstein model (Hammerstein, 1930) which consists of a static nonlinear part $f(\cdot)$, followed by a linear part $G_0(q)$ containing the dynamics of the process (see Fig. 1). Hammerstein models have proved to be able to describe accurately many nonlinear systems like chemical processes (Eskinat et al., 1991) and power amplifiers (Kim and Konstantinou, 2001) among others.

Many identification methods of Hammerstein systems have been reported in the literature. An overview of previous works can be found in Giri and Bai (2010). Also different classifications of these methods can be found in Bai (2003), Haber and Keviczky (1999) and Janczak (2005).

For system identification Least Squares Support Vector Machines (LS-SVM) (Suykens et al., 2002) have been used before. Results on well known benchmark data sets like the Wiener-Hammerstein data set (Schoukens et al., 2009) have been presented (e.g. (De Brabanter et al., 2009) and (Espinoza et al., 2004)). Other approaches where the information about the structure of the system is included into the LS-SVM models have been introduced (e.g. (Falck et al., 2009, 2012)). Also, some methods using this concept have been devised specifically for Hammerstein system identification (e.g. (Goethals et al., 2005; Castro-Garcia et al., 2015)).



Fig. 1. Hammerstein system with $G_0(q)$ a linear dynamical system, $f(u(t))$ a static nonlinearity and $v(t)$ the measurement noise. Throughout this paper, the q-notation will be used. The operator q is a time shift operator of the form $q^{-1}x(t) = x(t-1)$.

The proposed methodology in this paper not only takes into account the information about the structure of the system, but also exploits the structural information so that the impulse response of the linear system can be approximated.

The method proposed can be separated in two stages: First the system's impulse response is estimated. Then, using this, a LS-SVM model of the whole system is estimated (i.e. as opposed to modeling its component blocks). Even though at the second stage a model for the whole system is obtained, it can still be separated into the corresponding models of the linear block and the nonlinearity.

The capabilities of the method will be illustrated through several Monte Carlo simulations covering two examples and it will be shown how the measurement noise (white Gaussian

⋆ EU: The research leading to these results has received funding from the European Research Council under the European Unions Seventh Framework Programme (FP7/2007-2013) / ERC AdG A-DATADRIVE-B (290923). This paper reflects only the authors views and the Union is not liable for any use that may be made of the contained information. Research Council KUL: CoE PFV/10/002 (OPTEC), BIL12/11T; PhD/Postdoc grants Flemish Government: FWO: projects: G.0377.12 (Structured systems), G.088114N (Tensor based data similarity); PhD/Postdoc grant iMinds Medical Information Technologies SBO 2015 IWT: POM II SBO 100031 Belgian Federal Science Policy Office: IUAP P7/19 (DYSCO, Dynamical systems, control and optimization, 2012-2017).

Copyright by the International Federation of Automatic Control (IFAC)

noise with zero mean) affects the behavior of the proposed methodology.

Given that a modified formulation of LS-SVM is used in order to include the estimated Impulse Response, two different methods for tuning the parameters are also discussed. These methods are Genetic Algorithms and Simulated Annealing for global optimization on a validation set.

In this work scalars are represented in lower case and lower case followed by (t) is used for signals in the time domain. E.g. x is a scalar and $x(t)$ is a signal in the time domain. Also, vectors are represented with bold lower case and matrices with bold upper case e.g. $\boldsymbol{x}$ is a vector and $\boldsymbol{X}$ is a matrix.

The paper is organized as follows: In Section 2 the methods used to implement the proposed system identification technique are explained. In Section 3 the proposed method is presented. Section 4 shows the results found when applying the described methodology on two simulation examples. Finally, in Section 5, the conclusions are presented.

2. BACKGROUND

2.1 Hammerstein Impulse Response

Throughout this paper, we use the discrete time framework. Given this, we define an impulse as a Kronecker delta function. This means for $t \in \mathbb{N}$:

$$u_{imp}(t) = u_i \delta(t) = \begin{cases} u_i & \text{for } t = 0 \\ 0 & \text{for } t \neq 0. \end{cases} \tag{1}$$

This representation shows that the $\delta(t)$ function, by definition a unit impulse, is rescaled by a factor u_i.

In order to obtain an impulse response matrix from a Hammerstein system, it is enough to apply such an impulse as input and measure the corresponding output. This can be easily understood if we consider that the first block contains a static nonlinearity and therefore, the resulting intermediate variable $x_{imp}(t)$ for the impulse input $u_{imp}(t)$ is a rescaled version of $u_{imp}(t)$. The initial value is simply the value of the impulse multiplied by an unknown constant η, that is:

$$x_{imp}(t) = \begin{cases} \eta u_i & \text{for } t = 0 \text{ with } \eta \neq 0 \\ 0 & \text{for } t \neq 0. \end{cases} \tag{2}$$

The linear part will be excited then by $x_{imp}(t)$ and the corresponding output $y_{imp}(t)$ can be used to construct an Impulse Response Matrix $\boldsymbol{M}_{IR}$ (Ljung, 1999):

$$\boldsymbol{M}_{IR} = \begin{bmatrix} y_{imp}(0) & 0 & 0 & \cdots & 0 \\ y_{imp}(1) & y_{imp}(0) & 0 & \cdots & 0 \\ y_{imp}(2) & y_{imp}(1) & y_{imp}(0) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_{imp}(N-1) & y_{imp}(N-2) & y_{imp}(N-2) & \cdots & y_{imp}(0) \end{bmatrix}. \tag{3}$$

This is very convenient as we can easily obtain a rescaled version of the impulse response of the system. Note however that this measured impulse response will contain noise.

Note that this rescaling does not represent a problem as the LS-SVM model will take care of it in the next stage. In other words, the rescaling of the approximated linear block has no effect on the input-output behavior of the Hammerstein model (i.e. any pair of $\{f(u(t))/\eta,\ \eta G(q)\}$ with $\eta \neq 0$ would yield identical input and output measurements), as will be shown in Section 3.1.

It is important to highlight that that using an impulse excitation has the disadvantage of not being persistently exciting. In practice the amplitude of such signals is limited and hence, within the available experiment time, more information can be collected by using richer excitations. In this sense it is possible to use for instance a Pseudo Random Binary Signal (PRBS) input $u_{pr}(t)$ switching between zero and a non zero constant $\bar{u}$. This means that

$$\begin{aligned} y_{pr}(t) &= \boldsymbol{M}_{IR} x_{pr}(t) \\ &= \eta \boldsymbol{M}_{IR} u_{pr}(t), \end{aligned} \tag{4}$$

with $f(\bar{u}) = \eta \bar{u}$ and therefore $\hat{\boldsymbol{M}}_{IR} = \eta \boldsymbol{M}_{IR}$ can be estimated from the known $u_{pr}(t)$ and $y_{pr}(t)$.

2.2 Function estimation using LS-SVM

LS-SVM (Suykens et al., 2002) has been proposed within the framework of a primal-dual formulation. Having a data set $\{\boldsymbol{u}_i, x_i\}_{i=1}^N$, the objective is to find a model

$$\hat{x} = \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{u}) + b. \tag{5}$$

Here, $\boldsymbol{w} \in \mathbb{R}^{n_h}$, $\boldsymbol{u} \in \mathbb{R}^n$, $\hat{x} \in \mathbb{R}$ represents the estimated output value, $\boldsymbol{\varphi}(\cdot) : \mathbb{R}^n \to \mathbb{R}^{n_h}$ is the feature map to a high dimensional (possibly infinite) space and b is a bias term.

A constrained optimization problem is then formulated (Suykens et al., 2002):

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{e}} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{\gamma}{2}\sum_{i=1}^{N} e_i^2 \\ \text{subject to } & x_i = \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{u}_i) + b + e_i,\ i = 1, ..., N, \end{aligned} \tag{6}$$

with e_i the errors and γ the regularization parameter.

Using Mercer's theorem (Mercer, 1909), the kernel matrix $\boldsymbol{\Omega}$ can be represented by the kernel function $K(\boldsymbol{u}_i, \boldsymbol{u}_j) = \boldsymbol{\varphi}(\boldsymbol{u}_i)^T\boldsymbol{\varphi}(\boldsymbol{u}_j)$ with $i, j = 1, ..., N$. It is important to note that in this representation $\boldsymbol{\varphi}(\cdot)$ does not have to be explicitly known as it is implicitly used through the positive definite kernel function. In this paper the radial basis function kernel (i.e. RBF kernel) is used:

$$K(\boldsymbol{u}_i, \boldsymbol{u}_j) = \exp\left(\frac{-\left\|\boldsymbol{u}_i - \boldsymbol{u}_j\right\|_2^2}{\sigma^2}\right), \tag{7}$$

where σ is the kernel parameter.

From the Lagrangian $\mathcal{L}(\boldsymbol{w}, b, e; \alpha) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum_{i=1}^N e_i^2 - \sum_{i=1}^N \alpha_i(\boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{u}_i) + b + e_i - x_i)$ with $\alpha_i \in \mathbb{R}$ the Lagrange multipliers, the optimality conditions are derived:

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = 0 \to \boldsymbol{w} = \displaystyle\sum_{i=1}^{N} \alpha_i \boldsymbol{\varphi}(\boldsymbol{u}_i) \\ \dfrac{\partial \mathcal{L}}{\partial b} = 0 \to \displaystyle\sum_{i=1}^{N} \alpha_i = 0 \\ \dfrac{\partial \mathcal{L}}{\partial e_i} = 0 \to \alpha_i = \gamma e_i, i = 1, ..., N \\ \dfrac{\partial \mathcal{L}}{\partial \alpha_i} = 0 \to x_i = \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{u}_i) + b + e_i, i = 1, ..., N. \end{cases} \tag{8}$$

By elimination of $\boldsymbol{w}$ and e_i the following linear system is obtained:

$$\begin{bmatrix} 0 & \mathbf{1}_N^T \\ \mathbf{1}_N & \boldsymbol{\Omega} + \gamma^{-1}\boldsymbol{I}_N \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{x} \end{bmatrix} \tag{9}$$

with $\boldsymbol{x} = [x_1, ..., x_N]^T$ and $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_N]^T$. The resulting model is then:

$$\hat{x}(\boldsymbol{u}) = \sum_{i=1}^{N} \alpha_i K(\boldsymbol{u}, \boldsymbol{u}_i) + b. \tag{10}$$

3. PROPOSED METHOD

3.1 Impulse Response Constrained LS-SVM

The proposed method aims to integrate the Impulse Response Matrix $\boldsymbol{M}_{IR}$ of the Hammerstein system as defined in Section 2.1 into the LS-SVM formulation presented in Section 2.2. To do this, the constrained optimization problem is reformulated as follows for any input/output data:

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{e}} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{\gamma}{2}\boldsymbol{e}^T\boldsymbol{e} \\ \text{subject to } & \boldsymbol{y} = \boldsymbol{M}_{IR}(\boldsymbol{\Phi}_{\boldsymbol{U}}^T\boldsymbol{w} + \mathbf{1}_N b) + \boldsymbol{e}. \end{aligned} \tag{11}$$

Here, $\boldsymbol{w} \in \mathbb{R}^{n_h}$, the input matrix is $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_N]^T$ and the elements of the input signal $\boldsymbol{u}_i \in \mathbb{R}^n$. Also $\boldsymbol{y}, \boldsymbol{e}, \mathbf{1}_N \in \mathbb{R}^N$ with $\boldsymbol{y} = [y_1, y_2, \ldots, y_N]$ the output, $\boldsymbol{e} = [e_1, e_2, \ldots, e_N]$ the errors and $\mathbf{1}_N$ a vector of ones. Finally, $\boldsymbol{\Phi}_{\boldsymbol{U}} \in \mathbb{R}^{n_h \times N}$, or equivalently:

$$\boldsymbol{\Phi}_{\boldsymbol{U}} = [\boldsymbol{\varphi}(\boldsymbol{u}_1), \boldsymbol{\varphi}(\boldsymbol{u}_2), \ldots, \boldsymbol{\varphi}(\boldsymbol{u}_N)] \tag{12}$$

with $\boldsymbol{\varphi}(\cdot) : \mathbb{R}^n \to \mathbb{R}^{n_h}$ the feature map to a high dimensional (possibly infinite) space. Also, we have that $\boldsymbol{\Omega} = \boldsymbol{\Phi}_{\boldsymbol{U}}^T\boldsymbol{\Phi}_{\boldsymbol{U}}$.

Note that in the constraint of (11) the separation of the blocks is clear: The Impulse Response Matrix $\boldsymbol{M}_{IR}$ models the linear block and is multiplied with the output of the nonlinear model given by $\boldsymbol{\Phi}_{\boldsymbol{U}}^T\boldsymbol{w} + \mathbf{1}_N b$. However, note also that we only consider one global error for the whole model. This means that the tuning of the Impulse Response Constrained LS-SVM eventually will have to deal with the errors of the two blocks, i.e. the errors introduced by the Impulse Response Matrix and the errors introduced by the selection of the parameters in the nonlinear block.

From the Lagrangian:

$$\begin{aligned} \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{e}; \boldsymbol{\alpha}) = \; & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\boldsymbol{e}^T\boldsymbol{e} \\ & -\boldsymbol{\alpha}^T(\boldsymbol{M}_{IR}(\boldsymbol{\Phi}_{\boldsymbol{U}}^T\boldsymbol{w} + \mathbf{1}_N b) + \boldsymbol{e} - \boldsymbol{y}), \end{aligned} \tag{13}$$

the optimality conditions are then derived:

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = 0 \to \boldsymbol{w} = \boldsymbol{\Phi}_{\boldsymbol{U}}\boldsymbol{M}_{IR}^T\boldsymbol{\alpha} \\ \dfrac{\partial \mathcal{L}}{\partial b} = 0 \to 0 = \mathbf{1}_N^T\boldsymbol{M}_{IR}^T\boldsymbol{\alpha} \\ \dfrac{\partial \mathcal{L}}{\partial e_i} = 0 \to \boldsymbol{e} = \boldsymbol{\alpha}/\gamma \\ \dfrac{\partial \mathcal{L}}{\partial \alpha_i} = 0 \to \boldsymbol{y} = \boldsymbol{M}_{IR}(\boldsymbol{\Phi}_{\boldsymbol{U}}^T\boldsymbol{w} + \mathbf{1}_N b) + \boldsymbol{e}. \end{cases} \tag{14}$$

By elimination of $\boldsymbol{w}$ and $\boldsymbol{e}$, the last equation can be rewritten as:

$$\boldsymbol{y} = \boldsymbol{M}_{IR}(\boldsymbol{\Phi}_{\boldsymbol{U}}^T\boldsymbol{\Phi}_{\boldsymbol{U}}\boldsymbol{M}_{IR}^T\boldsymbol{\alpha} + \mathbf{1}_N b) + \frac{\boldsymbol{\alpha}}{\gamma} \tag{15}$$

and the following linear system is obtained:

$$\begin{bmatrix} 0 & \mathbf{1}_N^T\boldsymbol{M}_{IR}^T \\ \boldsymbol{M}_{IR}\mathbf{1}_N & \boldsymbol{M}_{IR}\boldsymbol{\Omega}\boldsymbol{M}_{IR}^T + \dfrac{1}{\gamma}\boldsymbol{I}_N \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{y} \end{bmatrix}. \tag{16}$$

For a new input signal $\boldsymbol{U}_d \in \mathbb{R}^{n \times D}$ with elements $\boldsymbol{d} \in \mathbb{R}^n$ and the training input $\boldsymbol{U} \in \mathbb{R}^{n \times N}$ with elements $\boldsymbol{u} \in \mathbb{R}^n$, let us define a matrix $\boldsymbol{K} \in \mathbb{R}^{D \times N}$ whose entries are defined as

$$\boldsymbol{K}_{i,j} = \exp\left(\frac{-\|\boldsymbol{u}_j - \boldsymbol{d}_i\|_2^2}{\sigma^2}\right), \tag{17}$$

with $i = 1, \ldots, D$ and $j = 1, \ldots, N$. Note that in the case where $\boldsymbol{U}_d = \boldsymbol{U}$ then $\boldsymbol{K} = \boldsymbol{\Omega}$.

If $N \neq D$ we also need to define an additional matrix $\boldsymbol{M}_{New}$. This matrix will be a re-sized version of $\boldsymbol{M}_{IR}$ in order to make it coincide with the new data set (i.e. $\boldsymbol{M}_{New} \in \mathbb{R}^{D \times D}$). Note that if the new data set is longer than the training one, and assuming that the impulse response $\boldsymbol{y}_{imp}$ is long enough to allow the system to settle down, $\boldsymbol{M}_{New}$ can be generated by extending $\boldsymbol{y}_{imp}$ with zeros. On the other hand, if the new data set is shorter than the training one, $\boldsymbol{M}_{New}$ can be generated by truncating $\boldsymbol{y}_{imp}$. Of course, if $N = D$ then $\boldsymbol{M}_{IR} = \boldsymbol{M}_{New}$.

Finally, we can define the estimated output for $\boldsymbol{U}_d$ as:

$$\hat{\boldsymbol{y}}(\boldsymbol{U}_d) = \boldsymbol{M}_{New}\boldsymbol{K}\boldsymbol{M}_{IR}^T\boldsymbol{\alpha} + \boldsymbol{M}_{New}\mathbf{1}_N b. \tag{18}$$

In this final formulation, the clear separation between the linear and nonlinear blocks present in (11) is lost. However, it is still possible to make a separation between the two blocks by factorizing $\boldsymbol{M}_{New}$. This leads then to

$$\hat{\boldsymbol{y}}(\boldsymbol{U}_d) = \boldsymbol{M}_{New}(\boldsymbol{K}\boldsymbol{M}_{IR}^T\boldsymbol{\alpha} + \mathbf{1}_N b). \tag{19}$$

In Section 4 it will be illustrated how from (19) we can recover a good approximation to the nonlinearity.

3.2 Role of Regularization

It is important to highlight the importance of the regularization in the found model.

As shown in (16), $\boldsymbol{y}$ can be expressed as:

$$\boldsymbol{y} = \boldsymbol{M}_{IR}\mathbf{1}_N b + \boldsymbol{M}_{IR}\boldsymbol{\Omega}\boldsymbol{M}_{IR}^T\boldsymbol{\alpha} + \frac{\boldsymbol{I}_N}{\gamma}\boldsymbol{\alpha}. \tag{20}$$

If we were to calculate the output of the found model for the input of the training data set, we would have then:

$$\tilde{\boldsymbol{y}} = \boldsymbol{M}_{IR}\mathbf{1}_N b + \boldsymbol{M}_{IR}\boldsymbol{\Omega}\boldsymbol{M}_{IR}^T\boldsymbol{\alpha} \tag{21}$$

It is clear then from (20) and (21) that $\tilde{\boldsymbol{y}} = \boldsymbol{y} - \boldsymbol{\alpha}/\boldsymbol{\gamma}$, this leads to

$$\tilde{\boldsymbol{y}} = \boldsymbol{y} - \frac{1}{\gamma}\left(\boldsymbol{M}_{IR}\boldsymbol{\Omega}\boldsymbol{M}_{IR}^T + \frac{\boldsymbol{I}_N}{\gamma}\right)^{-1}(\boldsymbol{y} - \boldsymbol{M}_{IR}\mathbf{1}_N b) \tag{22}$$

Now, let us assume that a change $\boldsymbol{\Delta}_v$ in the measurement noise occurs and let us analyze the effect it has in $\tilde{\boldsymbol{y}}$:

$$\begin{aligned} \tilde{\boldsymbol{y}} + \boldsymbol{\Delta}_{\tilde{y}} = \; & \boldsymbol{y} + \boldsymbol{\Delta}_v - \frac{1}{\gamma}(\boldsymbol{M}_{IR}\boldsymbol{\Omega}\boldsymbol{M}_{IR}^T + \frac{\boldsymbol{I}_N}{\gamma})^{-1} \\ & (\boldsymbol{y} + \boldsymbol{\Delta}_v - \boldsymbol{M}_{IR}\mathbf{1}_N b) \\ = \; & \boldsymbol{y} - \frac{1}{\gamma}(\boldsymbol{M}_{IR}\boldsymbol{\Omega}\boldsymbol{M}_{IR}^T + \frac{\boldsymbol{I}_N}{\gamma})^{-1} \\ & (\boldsymbol{y} - \boldsymbol{M}_{IR}\mathbf{1}_N b) + \boldsymbol{\Delta}_v \\ & -\frac{1}{\gamma}(\boldsymbol{M}_{IR}\boldsymbol{\Omega}\boldsymbol{M}_{IR}^T + \frac{\boldsymbol{I}_N}{\gamma})^{-1}\boldsymbol{\Delta}_v. \end{aligned} \tag{23}$$

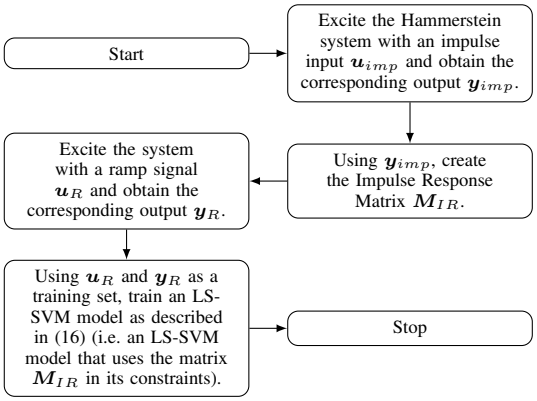

Fig. 2. Summary of the method showing the main steps.

Therefore

$$\begin{aligned}\boldsymbol{\Delta}_{\tilde{y}} &= \boldsymbol{\Delta}_v - \frac{1}{\gamma}(\boldsymbol{M}_{IR}\boldsymbol{\Omega}\boldsymbol{M}_{IR}^T + \frac{\boldsymbol{I}_N}{\gamma})^{-1}\boldsymbol{\Delta}_v \\ &= (\boldsymbol{I} - \frac{1}{\gamma}(\boldsymbol{M}_{IR}\boldsymbol{\Omega}\boldsymbol{M}_{IR}^T + \frac{\boldsymbol{I}_N}{\gamma})^{-1})\boldsymbol{\Delta}_v.\end{aligned} \tag{24}$$

From (24) it is evident that the effect of $\boldsymbol{\Delta}_v$ in $\tilde{\boldsymbol{y}}$ is heavily dependent on γ. Let us now assume that $\gamma \to 0$:

$$\boldsymbol{\Delta}_{\tilde{y}} \approx \left(\boldsymbol{I} - \frac{1}{\gamma}\left(\frac{\boldsymbol{I}_N}{\gamma}\right)^{-1}\right)\boldsymbol{\Delta}_v = 0. \tag{25}$$

Here, $\boldsymbol{\Delta}_v$ has no effect over $\tilde{\boldsymbol{y}}$. This result was to be expected as the errors considered in (11) have no impact at all given that γ is so small.

Let us assume now that $\gamma \to \infty$ (i.e. the errors in (11) are given a very high weigth):

$$\boldsymbol{\Delta}_{\tilde{y}} \approx (\boldsymbol{I} - \frac{1}{\gamma}(\boldsymbol{M}_{IR}\boldsymbol{\Omega}\boldsymbol{M}_{IR}^T)^{-1})\boldsymbol{\Delta}_v = \boldsymbol{\Delta}_v. \tag{26}$$

Here the model would follow the training points perfectly regardless of the noise. This is an undesirable effect as it clearly leads to overfitting.

3.3 Method Summary

In Fig. 2 the algorithm of the proposed method is summarized. Note that the elements in the input signals used are scalars and therefore, we drop the matrix notation.

4. SIMULATION RESULTS

The proposed methodology was applied to two systems in the discrete time framework. The first system (i.e. Example 1) was generated through a nonlinear block:

$$x(t) = u(t)^3 \tag{27}$$

and a linear block:

$$y(t) = \frac{B_1(q)}{A_1(q)}x(t) \tag{28}$$

where

$$\begin{aligned}B_1(q) &= q^6 + 0.8q^5 + 0.3q^4 + 0.4q^3 \\ A_1(q) &= q^6 - 2.789q^5 + 4.591q^4 - 5.229 \\ &\quad q^3 + 4.392q^2 - 2.553q + 0.8679.\end{aligned} \tag{29}$$

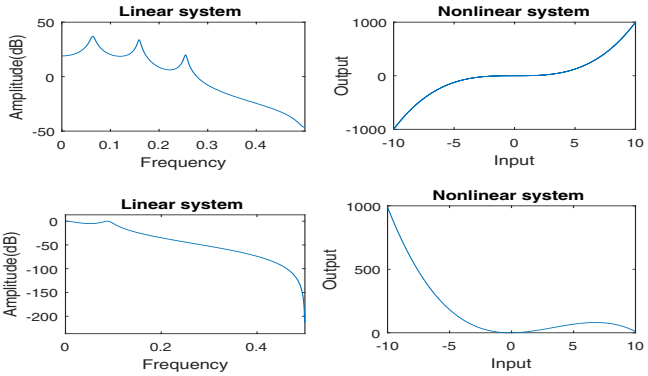

Fig. 3. Example 1: (Left) Linear block in the frequency domain (normalized). (Rigth) Nonlinear block. (Top) Example 1. (Bottom) Example 2.

The second system (i.e. Example 2) was generated through a nonlinear block:

$$x(t) = -0.5u^3 + 5u^2 + u \tag{30}$$

and a linear block:

$$y(t) = \frac{B_2(q)}{A_2(q)}x(t) \tag{31}$$

where

$$\begin{aligned}B_2(q) &= 0.004728q^3 + 0.01418q^2 + \\ &\quad 0.01418q + 0.004728 \\ A_2(q) &= q^3 - 2.458q^2 + 2.262q - 0.7654.\end{aligned} \tag{32}$$

The examples can be visualized in Fig. 3.

Both systems were initially excited using an impulse signal $\boldsymbol{u}_{imp}$ (i.e. $\boldsymbol{u}_{imp} = [10, 0, \ldots, 0]^T$) and the corresponding $\boldsymbol{y}_{imp}$ were retrieved. The Impulse Response Matrices $\boldsymbol{M}_{IR}$ were created using the corresponding $\boldsymbol{y}_{imp}$. Next, a ramp like signal is used to excite the systems (i.e. $\boldsymbol{u}_R$) and their corresponding outputs $\boldsymbol{y}_R$ are retrieved. Finally, using $\boldsymbol{u}_R$ and $\boldsymbol{y}_R$ and $\boldsymbol{M}_{IR}$, models for the systems are estimated as explained in Section 3.

The estimated models were tested in an independent test set. The inputs of this set $\boldsymbol{u}_T$ are Multilevel Pseudo Random Signals with 2% switching probability and amplitude values drawn from a uniform distribution with amplitudes in the interval $[-10, 10]$. All the signals in the presented examples consist of 500 samples.

In order to compare between the results of the two examples, let us have the Normalized MAE defined as shown in (33) for a signal with N measurements. Note that the Normalized MAE uses the noise free signal $y_T(t)$ and its estimated counterpart $\hat{y}_T(t)$.

$$\%\text{MAE} = \frac{100}{N}\frac{\sum_{t=1}^{N}|y_T(t) - \hat{y}_T(t)|}{|\max(y_T(t)) - \min(y_T(t))|}. \tag{33}$$

Two different methods for tuning the hyper-parameters (i.e. σ and γ) were tried, namely Genetic Algorithms and Simulated Annealing. These methods were used through validation sets. Results for Examples 1 and 2 are shown in Fig. 4 for the different tuning methods.

White Gaussian noise with zero mean was added to the systems' output such that a resulting Signal to Noise Ratio (SNR) of 20 dB was obtained. It is clear that even in the presence of noise, the proposed methodology works very well when Simulated Annealing or Genetic Algorithms are used.

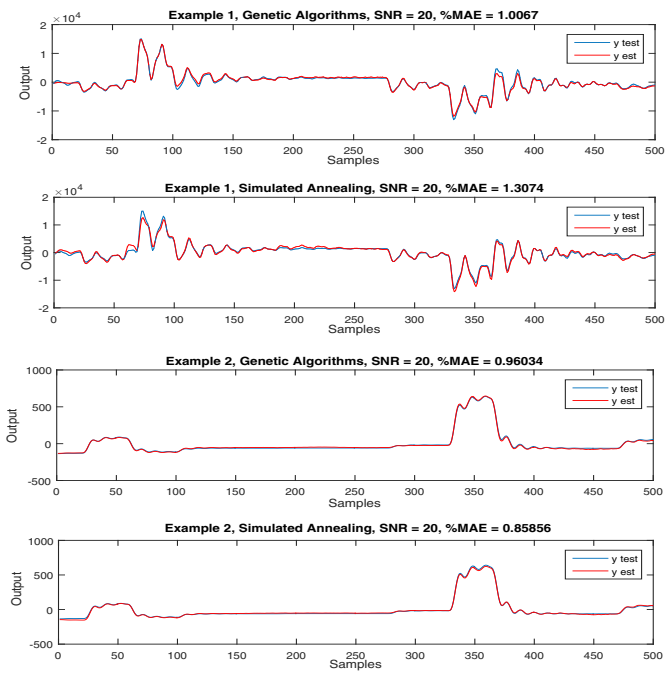

Fig. 4. Results for Examples 1 and 2.

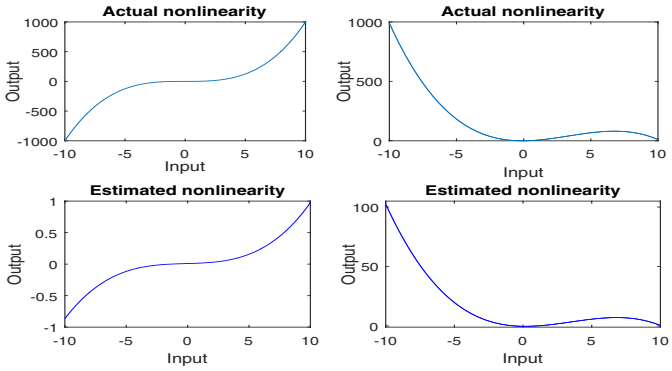

Fig. 5. (Left) Nonlinearity of Example 1. (Rigth) Nonlinearity of Example 2. (Top) Actual Nonlinearities. (Bottom) Estimated nonlinearities. Results corresponding to the examples of Figs. 4.

In Fig. 5, the found nonlinearities for Examples 1 and 2 are depicted. These estimations were done following the separation between the linear and nonlinear blocks in the found model explained in (19). This is, $\hat{\boldsymbol{x}} = \boldsymbol{K}\boldsymbol{M}_{IR}^T\boldsymbol{\alpha} + \boldsymbol{1}_N b$ with a $\boldsymbol{K}$ matrix generated as explained in Section 3.1 using the input of the training data $\boldsymbol{U}$ and an input $\boldsymbol{U}_{NL} = [-10, -9, \ldots, 9, 10]$.

As can be seen, even though the scales are different, the shapes of the estimated nonlinearities are very similar to the actual ones. Again, it is important to note that this scaling factor is of no consequence in the input-output behavior as any pair of $\{f(u(t))/\eta,\ \eta G(q)\}$ with $\eta \neq 0$ would yield identical input and output measurements.

In addition, in order to show the effect of parameter tuning during the modeling of the system Figs. 6 and 7 are presented. There σ and γ are alternatively fixed while the other varies in a wide range. The corresponding errors are displayed for the training and test set of Examples 1 and 2 for Genetic Algorithms and Simulated Annealing correspondingly.

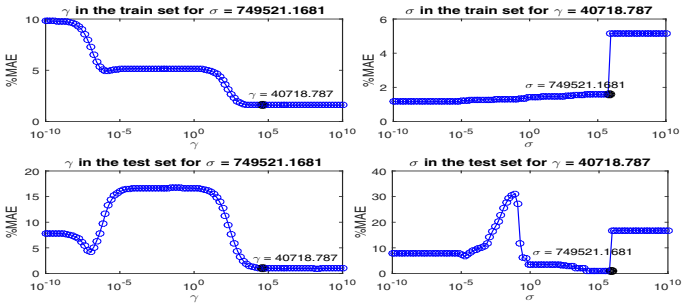

Fig. 6. Example1. Behavior of the error with respect to γ (left) and σ (right). (Top) Training set results. (Bottom) Test set results. The black dot shows the selected value.

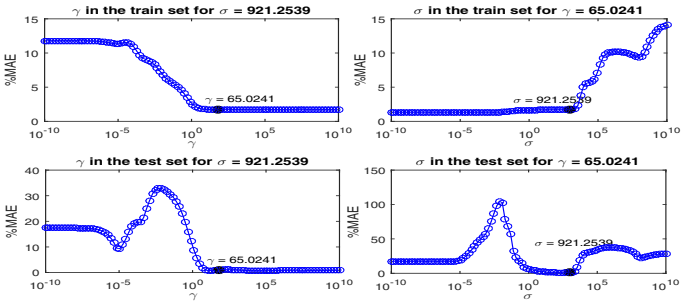

Fig. 7. Example 2. Behavior of the error with respect to γ (left) and σ (right). (Top) Training set results. (Bottom) Test set results. The black dot shows the selected value.

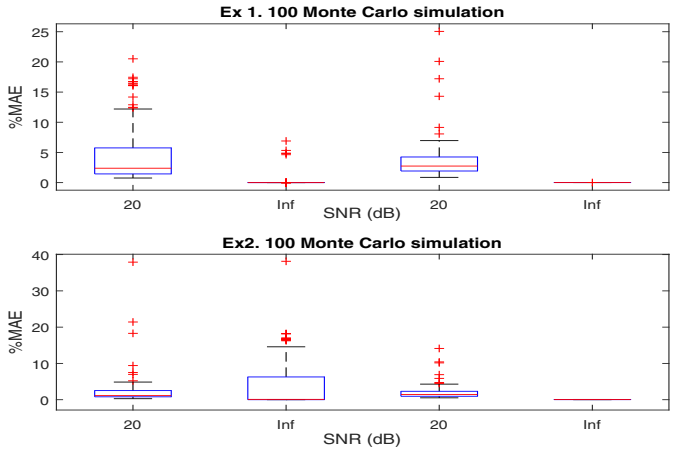

Fig. 8. Results of 100 Monte Carlo simulations using (Left) Genetic Algorithms and (Rigth) Simulated Annealing with 20dB SNR and no noise for Example 1 (Top) and Example 2 (Bottom).

Fig. 8 summarizes the result of 100 Monte Carlo simulations for each example and each tuning methodology.

From these results it can be seen that both Genetic Algorithms and Simulated Annealing can achieve very good results even in the presence of noise. However, it is also clear that the levels of noise used lead the results obtained with Genetic Algorithms to be slightly less homogeneous.

The proposed method takes the underlying structure of the system into account through the modified constraint in (11), therefore it is expected to produce better models than those obtained with purely black box methods like the NARX LS-SVM discussed in Suykens et al. (2002). Table 1 shows the results of the comparison between the proposed method (i.e.

Table 1. $\%MAE$ Comparison. Median values are offered for 100 Monte Carlo simulations for each case.

	SNR 20dB		No Noise	
Method	Ex 1	Ex 2	Ex 1	Ex 2
NARX LS-SVM	2.8154	2.0174	0.5668	0.4592
IR+LS-SVM (SA)	2.742	1.4452	0.0048	**0.03433**
IR+LS-SVM (GA)	2.3843	**1.1288**	$\mathbf{8.6216 \times 10^{-5}}$	0.0905
SITB PWLinear	**1.9487**	4.1317	0.2559	0.447
SITB SigmoidNet	3.3196	6.8528	0.3486	0.1992

IR+LS-SVM) and a NARX LS-SVM with 10 lags input and 10 lags output in a test set. Additionally, the method is compared against the MathWorks' System Identification Toolbox (SITB) (Ljung et al., 2007). Both, the single hidden layer neural network with sigmoid neurons (i.e. SigmoidNet) and the piecewise linear estimator are considered (i.e. PWlinear).

For the NARX LS-SVM, a ramp signal $\boldsymbol{u}_R$ is used for training and a 10-fold cross-validation scheme was used with a combination of Coupled Simulated Annealing (Xavier-de Souza et al., 2009) and simplex search for the tuning of the hyper-parameters (i.e. LS-SVMlab v1.8 [1]). For the SigmoidNet 25 neurons are used. Similarly, for the PWlinear 25 points are used for the nonlinear modeling. In both cases the order of the linearity is chosen by observation of the behavior of the noiseless case. This means that the order of numerator and denominator are the same in both examples and both methodologies, this is 6 and 3 for Examples 1 and 2 respectively.

To train the SITB methods, a 500 points ramp signal ranging from -15 to 15 was created. This signal was randomly shuffled so the resulting training signal is rich in its frequency content while covering all the input range.

It can be seen that the proposed method clearly outperforms the purely black box approach of NARX LS-SVM. Also, when compared with the SITB methods the results of the proposed method are in general better. Note that the order of the linear block was manually picked for the SITB methods in a noiseless environment while for the proposed method the process is fully automated.

5. CONCLUSIONS

The proposed method includes information about the structure of the system within a LS-SVM formulation. We exploit the structure of the Hammerstein system for obtaining a rescaled impulse response and the fact that such a rescaling is not a problem for the modeling of the system as a whole.

The results indicate that when the structure of the system is taken into account, a substantial improvement can be achieved in the resulting modeling. Also, they show that the method is effective in the presence of zero mean white Gaussian noise.

For this method, the kernel parameter σ and the regularization parameter γ have to be selected. To this end, two techniques were used and compared using Monte Carlo simulations.

It is interesting to note that in the initial formulation, a clear separation in the modeling of the linear and nonlinear blocks is present. However, when the final model to be used is derived from the dual, that separation is no longer clear anymore.

The solution of the model follows from solving a linear system of equations. This is a clear advantage over other methodologies like the overparametrization presented in Goethals et al. (2005).

Future work for the presented method includes the extension of the method to the MIMO case.

[1] http://www.esat.kuleuven.be/stadius/lssvmlab/

REFERENCES

Bai, E.W. (2003). Frequency domain identification of Hammerstein models. *IEEE Transactions on Automatic Control*, 48(4), 530–542.

Billings, S. and Fakhouri, S. (1982). Identification of systems containing linear dynamic and static nonlinear elements. *Automatica*, 18(1), 15–26.

Castro-Garcia, R., Tiels, K., Schoukens, J., and Suykens, J.A.K. (2015). Incorporating Best Linear Approximation within LS-SVM-Based Hammerstein System Identification. In *Proceedings of the 54th IEEE Conference on Decision and Control (CDC 2015)*, 7392–7397. IEEE.

De Brabanter, K., Dreesen, P., Karsmakers, P., Pelckmans, K., De Brabanter, J., Suykens, J.A.K., and De Moor, B. (2009). Fixed-size LS-SVM applied to the Wiener-Hammerstein benchmark. In *Proceedings of the 15th IFAC symposium on system identification (SYSID 2009)*, 826–831.

Eskinat, E., Johnson, S.H., and Luyben, W.L. (1991). Use of Hammerstein models in identification of nonlinear systems. *AIChE Journal*, 37(2), 255–268.

Espinoza, M., Pelckmans, K., Hoegaerts, L., Suykens, J.A.K., and De Moor, B. (2004). A comparative study of LS-SVMs applied to the Silverbox identification problem. In *Proc. of the 6th IFAC Symposium on Nonlinear Control Systems (NOLCOS)*.

Falck, T., Dreesen, P., De Brabanter, K., Pelckmans, K., De Moor, B., and Suykens, J.A.K. (2012). Least-Squares Support Vector Machines for the identification of Wiener-Hammerstein systems. *Control Engineering Practice*, 20(11), 1165–1174.

Falck, T., Pelckmans, K., Suykens, J.A.K., and De Moor, B. (2009). Identification of Wiener-Hammerstein systems using LS-SVMs. In *Proceedings of the 15th IFAC symposium on system identification (SYSID 2009)*, 820–825.

Giri, F. and Bai, E.W. (eds.) (2010). *Block-oriented nonlinear system identification*, volume 1. Springer.

Goethals, I., Pelckmans, K., Suykens, J.A.K., and De Moor, B. (2005). Identification of MIMO Hammerstein models using Least-Squares Support Vector Machines. *Automatica*, 41(7), 1263–1272.

Haber, R. and Keviczky, L. (1999). *Nonlinear System Identification Input-Output Modeling Approach*, volume 1. Springer The Netherlands.

Hammerstein, A. (1930). Nichtlineare Integralgleichungen nebst Anwendungen. *Acta Mathematica*, 54, 117–176.

Janczak, A. (2005). *Identification of Nonlinear Systems Using Neural Networks and Polynomial Models: A Block-Oriented Approach*, volume 310. Springer-Verlag Berlin Heidelberg.

Kim, J. and Konstantinou, K. (2001). Digital predistortion of wideband signals based on power amplifier model with memory. *Electronics Letters*, 37(23), 1417 – 1418.

Ljung, L. (1999). *System identification : theory for the user*. Prentice Hall information and system sciences series. Prentice Hall PTR, Upper Saddle River (NJ).

Ljung, L., Zhang, Q., Lindskog, P., Iouditski, A., and Singh, R. (2007). An integrated system identification toolbox for linear and nonlinear models. In *Proceedings of the 4th IFAC Symposium on System Identification, Newcastle, Australia*.

Mercer, J. (1909). Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, 415–446.

Schoukens, J., Suykens, J.A.K., and Ljung, L. (2009). Wiener-Hammerstein benchmark. In *Proceedings of the 15th IFAC Symposium on System Identification*.

Suykens, J.A.K., Van Gestel, T., De Brabanter, J., De Moor, B., and Vandewalle, J. (2002). *Least Squares Support Vector Machines*. World Scientific.

Xavier-de Souza, S., Suykens, J.A.K., Vandewalle, J., and Bollé, D. (2009). Coupled simulated annealing. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 40(2), 320–335.