

International Journal of Control

ISSN: 0020-7179 (Print) 1366-5820 (Online) Journal homepage: http://www.tandfonline.com/loi/tcon20

Impulse response constrained LS-SVM modelling for MIMO Hammerstein system identification

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

To cite this article: Ricardo Castro-Garcia, Oscar Mauricio Agudelo & Johan A. K. Suykens (2017): Impulse response constrained LS-SVM modelling for MIMO Hammerstein system identification, International Journal of Control, DOI: 10.1080/00207179.2017.1373862

To link to this article: https://doi.org/10.1080/00207179.2017.1373862

Published online: 13 Sep 2017.

Submit your article to this journal

Article views: 38

View related articles

View Crossmark data

Full Terms & Conditions of access and use can be found at
http://www.tandfonline.com/action/journalInformation?journalCode=tcon20

INTERNATIONAL JOURNAL OF CONTROL, 2017
https://doi.org/10.1080/00207179.2017.1373862

Taylor & Francis
Taylor & Francis Group

Check for updates

Impulse response constrained LS-SVM modelling for MIMO Hammerstein system identification

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

KU Leuven, ESAT-STADIUS, Kasteelpark Arenberg 10, Leuven, Belgium

ABSTRACT

In this paper, a new methodology for identifying multiple inputs multiple outputs Hammerstein systems is presented. The proposed method aims at incorporating the impulse response of the system into a least-squares support vector machine (LS-SVM) formulation and therefore the regularisation capabilities of LS-SVM are applied to the system as a whole. One of the main advantages of this method comes from the fact that it is flexible concerning the class of problems it can model and that no previous knowledge about the underlying non-linearities is required except for very mild assumptions. Also, it naturally adapts to handle different numbers of inputs/outputs and performs well in the presence of white Gaussian noise. Finally, the method incorporates information about the structure of the system but still the solution of the model follows from a linear system of equations. The performance of the proposed methodology is shown through three simulation examples and compared with other methods in the literature.

ARTICLE HISTORY
Received 6 February 2017
Accepted 25 August 2017

KEYWORDS
System identification; Hammerstein systems; MIMO; impulse response; LS-SVM

1. Introduction

A considerable amount of research has been carried out in the last decades in the field of nonlinear system identification. A popular approach is to employ one of the block structured nonlinear models introduced in the literature (Billings & Fakhouri, 1982; Haber & Unbehauen, 1990). The Hammerstein model (Hammerstein, 1930) is one such model and consists of a static nonlinear part $f(\cdot)$, followed by a linear part $G_0(q)$ describing the dynamics of the process (see Figure 1). Hammerstein models have proved to be useful to represent nonlinear systems in many application areas including control (Fruzzetti, Palazoglu, & McDonald, 1997), power amplifiers (Kim & Konstantinou, 2001), signal processing (Stapleton & Bass, 1985), chemical processes (Eskinat, Johnson, & Luyben, 1991) and biological processes (Hunter & Korenberg, 1986).

There is an extensive body of literature around Hammerstein system identification. An overview of previous works can be found in Giri and Bai (2010). Furthermore, different classifications of these methods can be found in Bai (2003), Haber and Keviczky (1999) and Janczak (2005).

Least-squares support vector machines (LS-SVM) (Suykens, Van Gestel, De Brabanter, De Moor, & Vandewalle, 2002) have been used for system identification. For example, in De Brabanter et al. (2009) and Espinoza, Pelckmans, Hoegaerts, Suykens, and De Moor (2004), LS-SVM is used in the Wiener-Hammerstein dataset (Schoukens, Suykens, & Ljung, 2009). Furthermore, approaches where the information about the structure of the system is included into the LS-SVM models have been introduced (e.g. in Falck, Pelckmans, Suykens, and De Moor (2009) and Falck, Dreesen, De Brabanter, Pelckmans, De Moor, and Suykens (2012)). In addition, some methods have been devised specifically for taking into account the Hammerstein system's structure (e.g. in Goethals, Pelckmans, Suykens, and De Moor (2005) and Castro-Garcia, Tiels, Schoukens, and Suykens (2015)).

Despite the abundance of the existing literature on Hammerstein system identification, the vast majority focuses on single-input single-output (SISO) case. For the multiple-input multiple-output (MIMO) case, however, much less works exist. Examples of methods for the identification of MIMO Hammerstein systems include for instance: Gomez and Baeyens (2004) where basis functions are used to represent both the linear and nonlinear parts of Hammerstein models; in Jeng and Huang (2008) through the use of specially designed signals the impulse response of the system is estimated and using least squares the intermediate variables are mapped. Using this approximation and the known input, a mapping of the nonlinearity is done through the fitting of a polynomial; an overparametrisation approach is proposed in Goethals et al. (2005) in combination with a

CONTACT Ricardo Castro-Garcia rcastro@esat.kuleuven.be
© 2017 Informa UK Limited, trading as Taylor & Francis Group

reformulated version of LS-SVM, although the MIMO case is not actually tested. Other methods for MIMO Hammerstein system identification include Lee, Sung, Park, and Park (2004),Verhaegen and Westwick (1996) and Al-Duwaish and Karim (1997).

Our method consists of two stages: first, it takes into account the information about the structure of the system in order to approximate its impulse response. Then, using this estimation, an LS-SVM model of the whole system is obtained. The main advantages of the proposed method include its versatility with respect to the class of systems that can be modelled. For instance, whereas the work in Lee et al. (2004) is applicable only to the case where the nonlinearities are in terms of a polynomial, or in Gomez and Baeyens (2004) specific basis functions have to be chosen beforehand, the methodology presented in this work can work without these limitations due to the good generalisation properties of LS-SVM. Another advantage is that the proposed method can be naturally used for the identification of multivariate Hammerstein systems with arbitrary numbers of inputs and outputs while other works have heavier restrictions in this regard (e.g. in Jeng and Huang (2008) it is required from the system that the number of inputs is equal to the number of outputs).

The proposed method is tested in three examples through several Monte Carlo simulations. It is shown how the measurement noise (white Gaussian noise with zero mean) affects its behavior and also how its accuracy compares with other state of the art methodologies.

In this work, vectors are represented with bold lower case and matrices with bold upper case. For example, $\boldsymbol{x}$ is a vector and $\boldsymbol{X}$ is a matrix.

The paper is organised as follows. In Section 2, the concepts on which the proposed system identification technique is based are explained. In Section 3, the method itself is presented. Section 4 shows the results found when applying the described methodology on three simulation examples. Finally, in Section 5, the conclusions are presented.

2. Background

2.1 Problem statement

Following the notation used, a conversion between a time signal and a vector should be transparent, e.g. a time signal $u(t)$ with samples at $t = 0, 1, \ldots, N-1$ can be represented as a vector $\boldsymbol{u} \in \mathbb{R}^N$. Similarly, a set of p signals $u_i(t)$ with $i = 1, \ldots, p$ and samples at $t = 0, 1, \ldots, N-1$ could be represented as a matrix $\boldsymbol{U} \in \mathbb{R}^{N\times p}$. Throughout the paper, the time and vector notations will be used interchangeably.

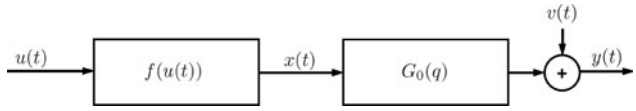

Figure 1. SISO Hammerstein system with $G_0(q)$ a linear dynamical system, $f(u(t))$ a static nonlinearity and $v(t)$ the measurement noise. Also, $u(t)$ and $y(t)$ are the input and output and $x(t)$ is the unknown intermediate variable. Throughout this paper, the q-notation will be used. The operator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

Given the structure of Hammerstein systems, the system to be identified is of the form

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{f}(\boldsymbol{U}), \tag{1}$$

where $\boldsymbol{H}$ is an impulse response matrix representing the linear part and $\boldsymbol{f}(\cdot)$ is the nonlinear part.

The impulse response matrix of a SISO linear time invariant (LTI) system can be constructed as follows (Ljung, 1999):

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

where the vector $\boldsymbol{y}_{\text{imp}} = [y_{\text{imp}}(0), y_{\text{imp}}(1), \ldots, y_{\text{imp}}(N-1)]$ is the response of the system to an impulse input.

For a system with p inputs, a matrix of the form $\boldsymbol{U} \in \mathbb{R}^{N\times p}$, where each column represents an input signal, can be used. It is assumed that there will be as many intermediate variables as inputs, therefore the intermediate variables will be $\boldsymbol{x}_i = \boldsymbol{f}_i(\boldsymbol{U})$ with $\boldsymbol{f}_i : \mathbb{R}^{N\times p} \to \mathbb{R}^N$ for $i = 1, \ldots, p$. Note then that for such a system $\boldsymbol{f} : \mathbb{R}^{N\times p} \to \mathbb{R}^{pN}$ with $\boldsymbol{f}(\boldsymbol{U}) = [\boldsymbol{f}_1(\boldsymbol{U})^\top, \ldots, \boldsymbol{f}_p(\boldsymbol{U})^\top]^\top$. For a system with r outputs, $\boldsymbol{y} \in \mathbb{R}^{rN}$. Finally, for a system with p inputs and r outputs, the impulse response matrix $\boldsymbol{H} \in \mathbb{R}^{rN\times pN}$ is as follows:

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H}_{11} & \boldsymbol{H}_{12} & \cdots & \boldsymbol{H}_{1p} \\ \boldsymbol{H}_{21} & \boldsymbol{H}_{22} & \cdots & \boldsymbol{H}_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{H}_{r1} & \boldsymbol{H}_{r2} & \cdots & \boldsymbol{H}_{rp} \end{bmatrix}, \tag{3}$$

with each $\boldsymbol{H}_{ij}$ as defined in (2) corresponding to the impulse response matrix from the jth input to the ith output. It is assumed that $\boldsymbol{f}_i(\boldsymbol{0}_{N\times p}) = \boldsymbol{0}_N$.

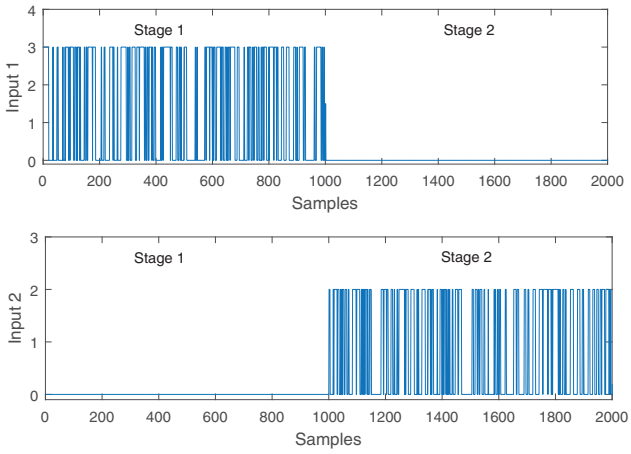

Figure 2. Example of the inputs for the identification of the LTI subsystem of a system with two inputs.

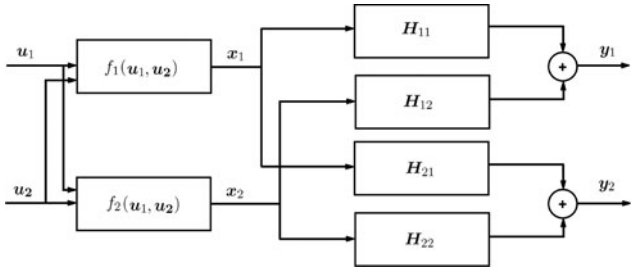

Figure 3. A MIMO Hammerstein system with two inputs and two outputs. $\boldsymbol{H}_{ij}$ are impulse response matrices corresponding to the linear dynamical systems of the ith output and the jth intermediate variable. $\boldsymbol{f}_1(\boldsymbol{u}_1, \boldsymbol{u}_2)$ and $\boldsymbol{f}_2(\boldsymbol{u}_1, \boldsymbol{u}_2)$ are static nonlinearities.

2.2 Impulse response of MIMO Hammerstein Systems

In Jeng and Huang (2008), a method for estimating the impulse response of a MIMO Hammerstein is introduced. Such method, where specially designed two level signals are used to take advantage of the inherent structure of a Hammerstein system, is adapted in this paper.

In order to excite the system for the identification of the LTI subsystem, the inputs are divided in as many stages as inputs. At each stage, a different input is a pseudo random binary signal (PRBS) switching between zero and a non-zero constant while all the other inputs are kept at zero. For a 2-input system, an illustration example is provided in Figure 2.

Given these inputs, the intermediate variables will be synchronised with the changing input at each stage and will have values of either 0 or a nonzero constant. This means that the intermediate variables will also be PRBS.

Without loss of generality, consider a system with two inputs $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ as the one represented in Figure 3. For the identification of the linear part, in this system there will be two stages then. In the first stage $u_{1i} \in \{0, \bar{u}_1\}$ and $u_{2i} = 0$ with $i = 1, \ldots, \left\lfloor \frac{N}{2} \right\rfloor$ and $\bar{u}_1$ the nonzero constant of the PRBS part of $\boldsymbol{u}_1$. Similarly, for the second stage $u_{1i} = 0$ and $u_{2i} \in \{0, \bar{u}_2\}$ for $i = \left\lfloor \frac{N}{2} \right\rfloor + 1, \ldots, 2\left\lfloor \frac{N}{2} \right\rfloor$ and $\bar{u}_2$ the nonzero constant of the PRBS part of $\boldsymbol{u}_2$. For such an excitation, the corresponding intermediate variables will be as stated in (4) for the first stage and as stated in (5) for the second one:

$$\begin{matrix} x_{1i} \in \{0,\ f_1(\bar{u}_1, 0)\} \\ x_{2i} \in \{0,\ f_2(\bar{u}_1, 0)\} \end{matrix}, i = 1, \ldots, \left\lfloor \frac{N}{2} \right\rfloor, \tag{4}$$

$$\begin{matrix} x_{1i} \in \{0,\ f_1(0, \bar{u}_2)\} \\ x_{2i} \in \{0,\ f_2(0, \bar{u}_2)\} \end{matrix}, i = \left\lfloor \frac{N}{2} \right\rfloor + 1, \ldots, 2\left\lfloor \frac{N}{2} \right\rfloor. \tag{5}$$

Let us define now $\boldsymbol{H}_{ij} \in \mathbb{R}^{N \times N}$ with $i = 1, \ldots, r$ and $j = 1, \ldots, p$ as the impulse response matrices corresponding to the different LTI blocks conforming the linear part of a system with p inputs and r outputs. Note that for $p = 2$ and $r = 2$, the system can then be represented as in (6) provided that both a_1 and a_2 are different from 0.

$$\begin{aligned} \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{bmatrix} &= \begin{bmatrix} \boldsymbol{H}_{11}\ \boldsymbol{H}_{12} \\ \boldsymbol{H}_{21}\ \boldsymbol{H}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_1(\boldsymbol{u}_1, \boldsymbol{u}_2) \\ \boldsymbol{f}_2(\boldsymbol{u}_1, \boldsymbol{u}_2) \end{bmatrix}, \\ &= \begin{bmatrix} \frac{\boldsymbol{H}_{11}}{a_1} & \frac{\boldsymbol{H}_{12}}{a_2} \\ \frac{\boldsymbol{H}_{21}}{a_1} & \frac{\boldsymbol{H}_{22}}{a_2} \end{bmatrix} \begin{bmatrix} a_1 \boldsymbol{f}_1(\boldsymbol{u}_1, \boldsymbol{u}_2) \\ a_2 \boldsymbol{f}_2(\boldsymbol{u}_1, \boldsymbol{u}_2) \end{bmatrix} \end{aligned} \tag{6}$$

with $\boldsymbol{f}_1 : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N$ and $\boldsymbol{f}_2 : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N$ the static nonlinear mappings; $\boldsymbol{y}_1 \in \mathbb{R}^N$ and $\boldsymbol{y}_2 \in \mathbb{R}^N$ the outputs and $\boldsymbol{u}_1, \boldsymbol{u}_2 \in \mathbb{R}^N$ the inputs of the system.

Equation (6) clearly shows that there could be a rescaling in either the linear or nonlinear parts and the other would have to compensate to keep the input–output relation. This is a known fact in the identification of block-oriented models (Boyd & Chua, 1983), where a rescaling of the blocks has no effect on the input-output behavior of the Hammerstein system (i.e. any pair of $\{\boldsymbol{f}(\boldsymbol{U})/\eta, \eta\boldsymbol{H}\}$ with $\eta \neq 0$ would yield identical input and output measurements).

Let us now define $a_1 = \bar{u}_1 / f_1(\bar{u}_1, 0)$ and $a_2 = \bar{u}_2 / f_2(0, \bar{u}_2)$. This definition necessarily means that during the first stage $a_1 \boldsymbol{x}_1 = \boldsymbol{u}_1$ and $a_2 \boldsymbol{x}_2 = \rho_2 \boldsymbol{u}_1$ with $\rho_2 = a_2 f_2(\bar{u}_1, 0)/\bar{u}_1$. In a similar way, for the second stage $a_1 \boldsymbol{x}_1 = \rho_1 \boldsymbol{u}_2$ and $a_2 \boldsymbol{x}_2 = \boldsymbol{u}_2$ with $\rho_1 = a_1 f_1(0, \bar{u}_2)/\bar{u}_2$.

From (6) then:

$$\begin{aligned} \boldsymbol{y}_1 &= \boldsymbol{H}_{11}\boldsymbol{x}_1 + \boldsymbol{H}_{12}\boldsymbol{x}_2, \\ \boldsymbol{y}_2 &= \boldsymbol{H}_{21}\boldsymbol{x}_1 + \boldsymbol{H}_{22}\boldsymbol{x}_2. \end{aligned} \tag{7}$$

For the first stage this means

$$\begin{aligned} \boldsymbol{y}_1 &= \left[\frac{\boldsymbol{H}_{11}}{a_1} + \rho_2 \frac{\boldsymbol{H}_{12}}{a_2}\right] \boldsymbol{u}_1 = \boldsymbol{H}_1^{(1)} \boldsymbol{u}_1, \\ \boldsymbol{y}_2 &= \left[\frac{\boldsymbol{H}_{21}}{a_1} + \rho_2 \frac{\boldsymbol{H}_{22}}{a_2}\right] \boldsymbol{u}_1 = \boldsymbol{H}_2^{(1)} \boldsymbol{u}_1. \end{aligned} \tag{8}$$

Similarly, for the second stage

$$\begin{aligned} \boldsymbol{y}_1 &= \left[\rho_1 \frac{\boldsymbol{H}_{11}}{a_1} + \frac{\boldsymbol{H}_{12}}{a_2}\right] \boldsymbol{u_2} = \boldsymbol{H}_1^{(2)} \boldsymbol{u}_2, \\ \boldsymbol{y}_2 &= \left[\rho_1 \frac{\boldsymbol{H}_{21}}{a_1} + \frac{\boldsymbol{H}_{22}}{a_2}\right] \boldsymbol{u}_2 = \boldsymbol{H}_2^{(2)} \boldsymbol{u_2}. \end{aligned} \tag{9}$$

Let us define now

$$\begin{aligned} \boldsymbol{Q} &= \begin{bmatrix} \boldsymbol{H}_1^{(1)} & \boldsymbol{H}_1^{(2)} \\ \boldsymbol{H}_2^{(1)} & \boldsymbol{H}_2^{(2)} \end{bmatrix}, \\ \boldsymbol{H} &= \begin{bmatrix} \boldsymbol{H}_{11} & \boldsymbol{H}_{12} \\ \boldsymbol{H}_{21} & \boldsymbol{H}_{22} \end{bmatrix}, \\ \boldsymbol{P} &= \begin{bmatrix} \frac{\boldsymbol{I}}{a_1} & \frac{\rho_1 \boldsymbol{I}}{a_1} \\ \frac{\rho_2 \boldsymbol{I}}{a_2} & \frac{\boldsymbol{I}}{a_2} \end{bmatrix}. \end{aligned} \tag{10}$$

Therefore,

$$\boldsymbol{Q} = \boldsymbol{H}\boldsymbol{P}. \tag{11}$$

From (8) and (9), it is clear that $\boldsymbol{H}_1^{(1)}$, $\boldsymbol{H}_2^{(1)}$, $\boldsymbol{H}_1^{(2)}$ and $\boldsymbol{H}_2^{(2)}$ can be directly calculated as $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ are known. In this paper, the least squares method was used to calculate $\boldsymbol{Q}$.

In order to calculate the impulse responses corresponding to the LTI blocks (i.e. $\boldsymbol{H}$), the only missing element now is $\boldsymbol{P}$.

Theorem 2.1: *Let us define the following matrix*

$$\tilde{\boldsymbol{P}} = \begin{bmatrix} \boldsymbol{I} & \tilde{\rho}_1 \boldsymbol{I} \\ \tilde{\rho}_2 \boldsymbol{I} & \boldsymbol{I} \end{bmatrix},$$

with

$$\begin{aligned} \tilde{\rho}_1 &= \beta_1 \rho_1, \\ \tilde{\rho}_2 &= \beta_2 \rho_2. \end{aligned}$$

Here, $\beta_1, \beta_2 \in \mathbb{R}$ with $\beta_1 \neq 0$ and $\beta_2 \neq 0$. β_1 is the proportion between the actual ρ_1, which is unknown, and the chosen $\tilde{\rho}_1$. Similarly, β_2 is the proportion between the actual ρ_2 and the chosen $\tilde{\rho}_2$.

Provided that $\tilde{\rho}_1 \tilde{\rho}_2 \neq 1$, matrix $\tilde{\boldsymbol{P}}$ can be used in place of $\boldsymbol{P}$ in (11). From an input–output perspective, the response of the identified MIMO Hammerstein system is then the same.

Proof:

$$\tilde{\boldsymbol{P}}^{-1} = \frac{1}{1 - \beta_1 \beta_2 \rho_1 \rho_2} \begin{bmatrix} \boldsymbol{I} & -\beta_1 \rho_1 \boldsymbol{I} \\ -\beta_2 \rho_2 \boldsymbol{I} & \boldsymbol{I} \end{bmatrix}.$$

As $\boldsymbol{Q}$ is known and instead of the actual $\boldsymbol{P}$ only $\tilde{\boldsymbol{P}}$ is available, from (11) an approximation to the linear block $\hat{\boldsymbol{H}}$ is found:

$$\begin{aligned} \hat{\boldsymbol{H}} &= \boldsymbol{Q}\tilde{\boldsymbol{P}}^{-1}, \\ &= \boldsymbol{H}\boldsymbol{P}\tilde{\boldsymbol{P}}^{-1}, \\ &= \boldsymbol{H} \begin{bmatrix} k_{11}\boldsymbol{I} & k_{12}\boldsymbol{I} \\ k_{21}\boldsymbol{I} & k_{22}\boldsymbol{I} \end{bmatrix}, \\ &= \boldsymbol{H}\boldsymbol{K}, \end{aligned}$$

with

$$\begin{aligned} k_{11} &= \frac{1 - \beta_2 \rho_1 \rho_2}{a_1(1 - \beta_1 \beta_2 \rho_1 \rho_2)}, k_{12} = \frac{(1 - \beta_1)\rho_1}{a_1(1 - \beta_1 \beta_2 \rho_1 \rho_2)}, \\ k_{21} &= \frac{(1 - \beta_2)\rho_2}{a_2(1 - \beta_1 \beta_2 \rho_1 \rho_2)}, k_{22} = \frac{1 - \beta_1 \rho_1 \rho_2}{a_2(1 - \beta_1 \beta_2 \rho_1 \rho_2)}. \end{aligned}$$

Note that in the case that $\rho_1 = \tilde{\rho}_1$ and $\rho_2 = \tilde{\rho}_2$, then

$$\beta_1 = \beta_2 = 1 \text{ and } \boldsymbol{K} = \begin{bmatrix} \frac{\boldsymbol{I}}{a_1} & \boldsymbol{0} \\ \boldsymbol{0} & \frac{\boldsymbol{I}}{a_2} \end{bmatrix}.$$

If $\boldsymbol{H}\boldsymbol{K}$ is used as an approximation to the linear part, all that is required for modelling the system, from an input-output perspective, is that the intermediate variables $\boldsymbol{x}$ are modified by $\boldsymbol{K}^{-1}$, this is:

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{K}\boldsymbol{K}^{-1}\boldsymbol{x}.$$

The only remaining consideration then is whether $\boldsymbol{K}$ is actually invertible. For $\boldsymbol{K}$ not invertible:

$$\begin{aligned} & k_{21}k_{12} = k_{11}k_{22} \\ \Rightarrow\ & \frac{1 - \beta_2 \rho_1 \rho_2}{a_1(1 - \beta_1 \beta_2 \rho_1 \rho_2)} \frac{1 - \beta_1 \rho_1 \rho_2}{a_2(1 - \beta_1 \beta_2 \rho_1 \rho_2)} = \frac{(1-\beta_1)\rho_1}{a_1(1-\beta_1\beta_2\rho_1\rho_2)} \frac{(1-\beta_2)\rho_2}{a_2(1-\beta_1\beta_2\rho_1\rho_2)} \\ \Rightarrow\ & 1 - \beta_2\rho_1\rho_2 - \beta_1\rho_1\rho_2 + \beta_1\beta_2(\rho_1\rho_2)^2 = (1 - \beta_1 - \beta_2 + \beta_1\beta_2)\rho_2\rho_1 \\ \Rightarrow\ & 1 + \beta_1\beta_2(\rho_1\rho_2)^2 = \rho_1\rho_2 + \beta_1\beta_2\rho_1\rho_2 \\ \Rightarrow\ & \tfrac{1}{\rho_1\rho_2} + \beta_1\beta_2\rho_1\rho_2 = 1 + \beta_1\beta_2 \\ \Rightarrow\ & \tfrac{1}{\rho_1\rho_2} - 1 = \beta_1\beta_2(1 - \rho_1\rho_2) \\ \Rightarrow\ & \tfrac{1-\rho_1\rho_2}{\rho_1\rho_2(1-\rho_1\rho_2)} = \beta_1\beta_2 \\ \Rightarrow\ & \tfrac{1}{\rho_1\rho_2} = \beta_1\beta_2 \\ \Rightarrow\ & 1 = \tilde{\rho}_1\tilde{\rho}_2. \end{aligned}$$

It can be seen that the non-invertible case of $\boldsymbol{K}$ is always avoided as $\tilde{\rho}_1\tilde{\rho}_2 \neq 1$ is already a constraint for selecting $\tilde{\rho}_1$ and $\tilde{\rho}_2$. ■

The number of inputs determines the number of existing ρ (and $\tilde{\rho}$) variables in the system. For a system with p inputs, this number is $p(p-1)$. In Theorem 2.1 it was shown, for a system with two inputs and two outputs, that as long as $\tilde{\rho}_1\tilde{\rho}_2 \neq 1$ the chosen values of $\tilde{\rho}_1$ and $\tilde{\rho}_2$ will

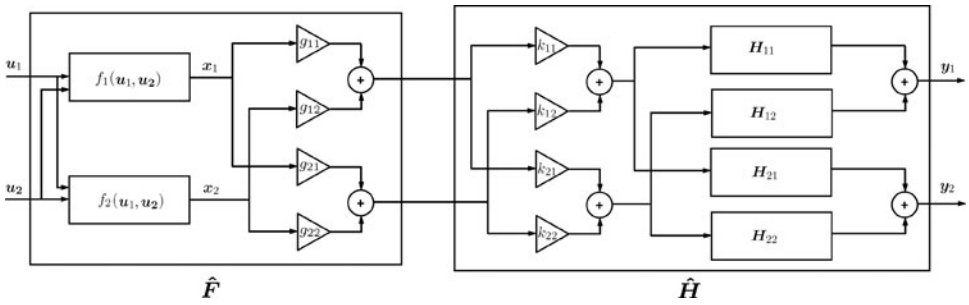

Figure 4. The model to be estimated. Note that once $\hat{\boldsymbol{H}}$ is calculated, $\hat{\boldsymbol{F}}$ can compensate the difference between $\boldsymbol{H}$ and $\hat{\boldsymbol{H}}$. Here, $g_{11} = \frac{k_{22}}{k_{11}k_{22}-k_{12}k_{21}}$, $g_{12} = -\frac{k_{12}}{k_{11}k_{22}-k_{12}k_{21}}$, $g_{21} = -\frac{k_{21}}{k_{11}k_{22}-k_{12}k_{21}}$ and $g_{22} = \frac{k_{11}}{k_{11}k_{22}-k_{12}k_{21}}$.

not affect the response of the identified MIMO Hammerstein system from an input–output perspective. In order to guarantee a proper selection of the $\tilde{\rho}$ values in systems with more inputs, an equivalent restriction must be made, this is: The chosen $\tilde{\rho}_i$ for $i = 1, \ldots, p(p-1)$ must be such that $\det(\tilde{\boldsymbol{P}}) \neq 0$.

In Figure 4, a graphical representation of the model to be found for a system with two inputs and two outputs is presented.

2.3 Function estimation using least squares support vector machines

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

$$\hat{x} = \boldsymbol{w}^\top \boldsymbol{\varphi}(\boldsymbol{u}) + b. \tag{12}$$

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

A constrained optimisation problem is then formulated:

$$\begin{aligned} \min_{\boldsymbol{w},b,e_i} \quad & \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + \frac{\gamma}{2}\sum_{i=1}^{N} e_i^2 \\ \text{subject to } & x_i = \boldsymbol{w}^\top \boldsymbol{\varphi}(\boldsymbol{u}_i) + b + e_i, \quad i = 1, \ldots, N, \end{aligned} \tag{13}$$

where e_i are the output residuals and γ represents the regularisation parameter.

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

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} &= 0 \to \boldsymbol{w} = \sum_{i=1}^{N} \alpha_i \boldsymbol{\varphi}(\boldsymbol{u}_i), \\ \frac{\partial \mathcal{L}}{\partial b} &= 0 \to \sum_{i=1}^{N} \alpha_i = 0, \\ \frac{\partial \mathcal{L}}{\partial e_i} &= 0 \to \alpha_i = \gamma e_i, \; i = 1, \ldots, N, \\ \frac{\partial \mathcal{L}}{\partial \alpha_i} &= 0 \to x_i = \boldsymbol{w}^\top \boldsymbol{\varphi}(\boldsymbol{u}_i) + b + e_i, \; i = 1, \ldots, N. \end{aligned} \tag{14}$$

Using Mercer's theorem (Mercer, 1909), the kernel matrix $\boldsymbol{\Omega}$ can be represented by the kernel function $\boldsymbol{\Omega}_{ij} = k(\boldsymbol{u}_i, \boldsymbol{u}_j) = \boldsymbol{\varphi}(\boldsymbol{u}_i)^\top \boldsymbol{\varphi}(\boldsymbol{u}_j)$ with $i, j = 1, \ldots, N$. 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:

$$k(\boldsymbol{u}_i, \boldsymbol{u}_j) = \exp\left(\frac{-\left\|\boldsymbol{u}_i - \boldsymbol{u}_j\right\|_2^2}{\sigma^2}\right), \tag{15}$$

where σ is the kernel parameter. The RBF kernel usually provides a good performance as it is a universal kernel approximator (Wang, Chen, & Chen, 2004) and is relatively easy to tune as it depends on only one parameter. This will be important later when the tuning of several kernels is necessary (i.e. see Sections 3.1 and 3.2). Note that the feature space of the RBF kernel has an infinite number of dimensions, however, thanks to the use of the kernel function and the kernel trick, this is not a problem at all.

The dual formulation is obtained then from (14) by elimination of $\boldsymbol{w}$ and e_i:

$$\begin{bmatrix} 0 & \mathbf{1}_N^T \\ \mathbf{1}_N & \boldsymbol{\Omega} + \gamma^{-1}\boldsymbol{I}_N \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{x} \end{bmatrix}, \tag{16}$$

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

$$\hat{x}(\boldsymbol{u}) = \sum_{i=1}^{N} \alpha_i k(\boldsymbol{u}, \boldsymbol{u}_i) + b. \tag{17}$$

3. Proposed method

3.1 MIMO case

For illustrative purposes and without loss of generality, we will show the case where the system has two inputs $\boldsymbol{u}_1, \boldsymbol{u}_2 \in \mathbb{R}^N$, two intermediate variables $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^N$ and two outputs $\boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathbb{R}^N$ (see Figure 3). In Section 3.2, this will be extended to a more general case with p inputs and r outputs.

In this system, the nonlinear part will be approximated by two separate expressions: $\hat{\boldsymbol{x}}_1 = \hat{\boldsymbol{f}}_1(\boldsymbol{u}_1, \boldsymbol{u}_2)$ and $\hat{\boldsymbol{x}}_2 = \hat{\boldsymbol{f}}_2(\boldsymbol{u}_1, \boldsymbol{u}_2)$. Also the matrix $\hat{\boldsymbol{H}}$ (i.e. see Theorem 2.1) will approximate $G(q)$ in the time domain and is composed by the approximation of the impulse response of the LTI blocks that map the intermediate variables to the outputs.

$$\hat{\boldsymbol{H}} = \begin{bmatrix} \hat{\boldsymbol{H}}_{11} & \hat{\boldsymbol{H}}_{12} \\ \hat{\boldsymbol{H}}_{21} & \hat{\boldsymbol{H}}_{22} \end{bmatrix}. \tag{18}$$

With the elements above, we can derive an expression to incorporate the linear part into the calculation of the whole model. Let us define

$$\begin{aligned} \boldsymbol{y} &= \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{bmatrix} \in \mathbb{R}^{2N}, \\ \boldsymbol{M}_1 &= \begin{bmatrix} \hat{\boldsymbol{H}}_{11} \\ \hat{\boldsymbol{H}}_{21} \end{bmatrix} \in \mathbb{R}^{2N \times N}, \\ \boldsymbol{M}_2 &= \begin{bmatrix} \hat{\boldsymbol{H}}_{12} \\ \hat{\boldsymbol{H}}_{22} \end{bmatrix} \in \mathbb{R}^{2N \times N}, \\ \boldsymbol{e} &= \begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \end{bmatrix} \in \mathbb{R}^{2N}, \end{aligned} \tag{19}$$

where $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ correspond to the errors associated with the outputs $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$, respectively.

Now, let the cost function be defined as

$$\min_{\boldsymbol{w}_1, \boldsymbol{w}_2, b_1, b_2, \boldsymbol{e}_1, \boldsymbol{e}_2} J = \frac{1}{2}\boldsymbol{w}_1^\top \boldsymbol{w}_1 + \frac{1}{2}\boldsymbol{w}_2^\top \boldsymbol{w}_2 + \frac{\gamma_1}{2}\boldsymbol{e}_1^\top \boldsymbol{e}_1 + \frac{\gamma_2}{2}\boldsymbol{e}_2^\top \boldsymbol{e}_2$$

$$\text{subject to } \boldsymbol{y} = \boldsymbol{M}_1 \left(\boldsymbol{\Phi}_1^\top \boldsymbol{w}_1 + \boldsymbol{1}_N b_1\right) + \boldsymbol{M}_2 \left(\boldsymbol{\Phi}_2^\top \boldsymbol{w}_2 + \boldsymbol{1}_N b_2\right) + \boldsymbol{e}. \tag{20}$$

Here b_1 and b_2 are the bias terms for each nonlinear function (i.e. $\hat{\boldsymbol{f}}_1(\boldsymbol{u}_1, \boldsymbol{u}_2)$ and $\hat{\boldsymbol{f}}_2(\boldsymbol{u}_1, \boldsymbol{u}_2)$), $\boldsymbol{\Phi}_1$ and $\boldsymbol{\Phi}_2$ are the aggregation of columns $\varphi_1(\boldsymbol{x}_i)$ and $\varphi_2(\boldsymbol{x}_i)$, respectively (i.e. the functions mapping the inputs to a higher dimensional feature space), with $i = 1, \ldots, N$, $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ are the weight vectors of the nonlinear functions and $\boldsymbol{1}_N$ is a column vector of ones with length N (the length of the data-set).

From the Lagrangian

$$\begin{aligned} &\mathcal{L}(\boldsymbol{w}_1, \boldsymbol{w}_2, b_1, b_2, \boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{\alpha}) \\ &= \frac{1}{2}\boldsymbol{w}_1^\top \boldsymbol{w}_1 + \frac{1}{2}\boldsymbol{w}_2^\top \boldsymbol{w}_2 + \frac{\gamma_1}{2}\boldsymbol{e}_1^\top \boldsymbol{e}_1 + \frac{\gamma_2}{2}\boldsymbol{e}_2^\top \boldsymbol{e}_2 \\ &\quad - \boldsymbol{\alpha}^\top \left(\boldsymbol{M}_1 \left(\boldsymbol{\Phi}_1^\top \boldsymbol{w}_1 + \boldsymbol{1}_N b_1\right) + \boldsymbol{M}_2 \left(\boldsymbol{\Phi}_2^\top \boldsymbol{w}_2 + \boldsymbol{1}_N b_2\right) \right. \\ &\quad \left. + \boldsymbol{e} - \boldsymbol{y}\right) \end{aligned} \tag{21}$$

with $\boldsymbol{\alpha} \in \mathbb{R}^{2N}$ the Lagrange multipliers, the optimality conditions are derived:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \boldsymbol{w}_1} &= 0 \rightarrow \boldsymbol{w}_1 = \boldsymbol{\Phi}_1 \boldsymbol{M}_1^\top \boldsymbol{\alpha}, \\ \frac{\partial \mathcal{L}}{\partial \boldsymbol{w}_2} &= 0 \rightarrow \boldsymbol{w}_2 = \boldsymbol{\Phi}_2 \boldsymbol{M}_2^\top \boldsymbol{\alpha}, \\ \frac{\partial \mathcal{L}}{\partial b_1} &= 0 \rightarrow 0 = \boldsymbol{1}_N^\top \boldsymbol{M}_1^\top \boldsymbol{\alpha}, \\ \frac{\partial \mathcal{L}}{\partial b_2} &= 0 \rightarrow 0 = \boldsymbol{1}_N^\top \boldsymbol{M}_2^\top \boldsymbol{\alpha}, \\ \frac{\partial \mathcal{L}}{\partial \boldsymbol{e}} &= 0 \rightarrow \boldsymbol{\alpha} = \boldsymbol{\Gamma} \boldsymbol{e}, \\ \frac{\partial \mathcal{L}}{\partial \alpha_i} &= 0 \rightarrow \boldsymbol{y} = \boldsymbol{M}_1 \left(\boldsymbol{\Phi}_1^\top \boldsymbol{w}_1 + \boldsymbol{1}_N b_1\right) \\ &\qquad + \boldsymbol{M}_2 \left(\boldsymbol{\Phi}_2^\top \boldsymbol{w}_2 + \boldsymbol{1}_N b_2\right) + \boldsymbol{e}, \end{aligned} \tag{22}$$

with $\boldsymbol{\Gamma} = \text{diag}([\gamma_1 \boldsymbol{1}_N^\top, \gamma_2 \boldsymbol{1}_N^\top]^\top)$.

The last equation in (22) can be rewritten by replacing $\boldsymbol{w}_1$, $\boldsymbol{w}_2$ and $\boldsymbol{e}$ as

$$\begin{aligned} \boldsymbol{y} = \boldsymbol{M}_1 &\left(\boldsymbol{\Phi}_1^\top \boldsymbol{\Phi}_1 \boldsymbol{M}_1^\top \boldsymbol{\alpha} + \boldsymbol{1}_N b_1\right) \\ &+ \boldsymbol{M}_2 \left(\boldsymbol{\Phi}_2^\top \boldsymbol{\Phi}_2 \boldsymbol{M}_2^\top \boldsymbol{\alpha} + \boldsymbol{1}_N b_2\right) + \boldsymbol{\Gamma}^{-1} \boldsymbol{\alpha}. \end{aligned} \tag{23}$$

Note again that Mercer's theorem can be used and therefore kernel functions $\boldsymbol{\Omega}_{ij}^{(1)} = k_1(\boldsymbol{u}_{\text{train},i}, \boldsymbol{u}_{\text{train},j})$ and $\boldsymbol{\Omega}_{ij}^{(2)} = k_2(\boldsymbol{u}_{\text{train},i}, \boldsymbol{u}_{\text{train},j})$ represent the kernel matrices $\boldsymbol{\Omega}_1$ and $\boldsymbol{\Omega}_2$, correspondingly. Note that this implies that $\boldsymbol{\Omega}^{(1)} = \boldsymbol{\Phi}_1^\top \boldsymbol{\Phi}_1$ and $\boldsymbol{\Omega}^{(2)} = \boldsymbol{\Phi}_2^\top \boldsymbol{\Phi}_2$.

The following linear system is finally obtained

$$\begin{bmatrix} 0 & 0 & \boldsymbol{1}_N^\top \boldsymbol{M}_1^\top \\ 0 & 0 & \boldsymbol{1}_N^\top \boldsymbol{M}_2^\top \\ \boldsymbol{M}_1 \boldsymbol{1}_N & \boldsymbol{M}_2 \boldsymbol{1}_N & \boldsymbol{M}_1 \boldsymbol{\Omega}^{(1)} \boldsymbol{M}_1^\top + \boldsymbol{M}_2 \boldsymbol{\Omega}^{(2)} \boldsymbol{M}_2^\top + \boldsymbol{\Gamma}^{-1} \end{bmatrix} \times \begin{bmatrix} b_1 \\ b_2 \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \boldsymbol{y} \end{bmatrix}. \tag{24}$$

The resulting model for a new input $\boldsymbol{U}_{\text{new}} \in \mathbb{R}^{N_n \times p}$ is then:

$$\hat{\boldsymbol{y}} = \bar{\boldsymbol{M}}_1 \mathbf{1}_N b_1 + \bar{\boldsymbol{M}}_2 \mathbf{1}_N b_2 + \left(\bar{\boldsymbol{M}}_1 \boldsymbol{K}^{(1)} \breve{\boldsymbol{M}}_1^\top + \bar{\boldsymbol{M}}_2 \boldsymbol{K}^{(2)} \breve{\boldsymbol{M}}_2^\top\right) \boldsymbol{\alpha}, \tag{25}$$

with $\boldsymbol{U}_{\text{train}}$ the set of inputs used to train the model, $\boldsymbol{K}_{ij}^{(1)} = k_1(\boldsymbol{u}_{\text{train},i}, \boldsymbol{u}_{\text{new},j})$ and $\boldsymbol{K}_{ij}^{(2)} = k_2(\boldsymbol{u}_{\text{train},i}, \boldsymbol{u}_{\text{new},j})$ with $i = 1, \ldots, N, j = 1, \ldots, N_n$ and therefore $\boldsymbol{K}^{(1)}, \boldsymbol{K}^{(2)} \in \mathbb{R}^{N \times N_n}$. $\bar{\boldsymbol{M}}_1, \bar{\boldsymbol{M}}_2 \in \mathbb{R}^{2N_n \times N}$ and $\breve{\boldsymbol{M}}_1, \breve{\boldsymbol{M}}_2 \in \mathbb{R}^{2N \times N_n}$. Note that if $N \neq N_n$ then $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ have to be truncated or expanded to generate $\bar{\boldsymbol{M}}_1, \bar{\boldsymbol{M}}_2, \breve{\boldsymbol{M}}_1, \breve{\boldsymbol{M}}_2$ and comply with the new dimensions. Given the way these matrices are constructed, this should be straightforward.

3.2 General case

The method's formulation can be extended to a p inputs and r outputs case easily though the solution of such systems becomes computationally expensive very quickly as the number of inputs and/or outputs increases. For this extension, the cost function in (20) is rewritten leading to a rewriting of (24).

$$\begin{aligned} \min_{\boldsymbol{w}_i, b_i, \boldsymbol{e}_j} \quad & \boldsymbol{J} = \sum_{i=1}^{p} \frac{1}{2} \boldsymbol{w}_i^\top \boldsymbol{w}_i + \sum_{j=1}^{r} \frac{\gamma_j}{2} \boldsymbol{e}_j^\top \boldsymbol{e}_j \\ \text{subject to } & \boldsymbol{y} = \sum_{i=1}^{p} \boldsymbol{M}_i \left(\boldsymbol{\Phi}_i^\top \boldsymbol{w}_i + \mathbf{1}_N b_i\right) + \boldsymbol{e}. \end{aligned} \tag{26}$$

Here,

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \vdots \\ \boldsymbol{y}_r \end{bmatrix}, \text{ with } \boldsymbol{y}_i \in \mathbb{R}^N \text{and } i = 1, \ldots, r,$$

$$\boldsymbol{e} = \begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \\ \vdots \\ \boldsymbol{e}_r \end{bmatrix}, \text{ with } \boldsymbol{e}_i \in \mathbb{R}^N \text{and } i = 1, \ldots, r. \tag{27}$$

Also let us define $\boldsymbol{M}_i \in \mathbb{R}^{rN \times N}$ for $i = 1, 2, \ldots, p$

$$\boldsymbol{M}_i = \begin{bmatrix} \hat{\boldsymbol{H}}_{1i} \\ \hat{\boldsymbol{H}}_{2i} \\ \vdots \\ \hat{\boldsymbol{H}}_{ri} \end{bmatrix}. \tag{28}$$

A matrix $\boldsymbol{M} \in \mathbb{R}^{rN \times p}$ can then be defined such that

$$\boldsymbol{M} = \left[\boldsymbol{M}_1 \mathbf{1}_N, \boldsymbol{M}_2 \mathbf{1}_N, \ldots, \boldsymbol{M}_p \mathbf{1}_N\right], \tag{29}$$

and a diagonal matrix $\boldsymbol{\Gamma} \in \mathbb{R}^{rN \times rN}$

$$\boldsymbol{\Gamma} = \text{diag}\left(\left[\gamma_1 \mathbf{1}_N^\top, \gamma_2 \mathbf{1}_N^\top, \ldots, \gamma_r \mathbf{1}_N^\top\right]^\top\right). \tag{30}$$

Finally, $\boldsymbol{b} \in \mathbb{R}^p$ is defined as

$$\boldsymbol{b} = [b_1, b_2, \ldots, b_p]^\top. \tag{31}$$

The corresponding linear system is then

$$\begin{bmatrix} \mathbf{0}_{p \times p} & \boldsymbol{M}^\top \\ \boldsymbol{M} & \boldsymbol{\Gamma}^{-1} + \sum_{i=1}^{p} \boldsymbol{M}_i \boldsymbol{\Omega}^{(i)} \boldsymbol{M}_i^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_p \\ \boldsymbol{y} \end{bmatrix}. \tag{32}$$

Note that for this system, $p + r$ parameters must be tuned if the RBF Gaussian kernel is used (i.e. $\sigma_1, \sigma_2, \ldots, \sigma_p$ and $\gamma_1, \gamma_2, \ldots, \gamma_r$). Also note that $\boldsymbol{\alpha}, \boldsymbol{y} \in \mathbb{R}^{rN}$.

The resulting model for a new input $\boldsymbol{U}_{\text{new}} \in \mathbb{R}^{N_n \times p}$ is then:

$$\hat{\boldsymbol{y}} = \bar{\boldsymbol{M}} \boldsymbol{b} + \left(\sum_{i=1}^{p} \bar{\boldsymbol{M}}_i \boldsymbol{K}^{(i)} \breve{\boldsymbol{M}}_i^\top\right) \boldsymbol{\alpha}, \tag{33}$$

with $\boldsymbol{U}_{\text{train}} \in \mathbb{R}^{N \times p}$ the set of inputs used to train the model, $\boldsymbol{K}_{lj}^{(i)} = k_i(\boldsymbol{u}_{\text{train},l}, \boldsymbol{u}_{\text{new},j})$ with $k_i(\cdot, \cdot)$ the kernel functions with $i = 1, \ldots, p$, $l = 1, \ldots, N$ and $j = 1, \ldots, N_n$ and therefore $\boldsymbol{K}^{(i)} \in \mathbb{R}^{N \times N_n}$. $\bar{\boldsymbol{M}}_i \in \mathbb{R}^{rN_n \times N}, \bar{\boldsymbol{M}} = \left[\bar{\boldsymbol{M}}_1 \mathbf{1}_N, \bar{\boldsymbol{M}}_2 \mathbf{1}_N, \ldots, \bar{\boldsymbol{M}}_p \mathbf{1}_N\right]$ and $\breve{\boldsymbol{M}}_i \in \mathbb{R}^{rN \times N_n}$. Again, note that if $N \neq N_n$ then $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ have to be truncated or expanded to generate $\bar{\boldsymbol{M}}_i, \breve{\boldsymbol{M}}_i$ and comply with the new dimensions.

3.3 SISO case

Given that the SISO case is intrinsically simpler than the MIMO one, a simplified version of the method is offered for such cases (Castro-Garcia, Agudelo, & Suykens, 2017). Let us define an impulse as a Kronecker delta function. This means for $t \in \mathbb{N}$:

$$u_{\text{imp}}(t) = \bar{u}_i \delta(t) = \begin{cases} \bar{u}_i & \text{for } t = 0 \\ 0 & \text{for } t \neq 0. \end{cases} \tag{34}$$

This representation shows that the $\delta(t)$ function, by definition a unit impulse, is rescaled by a factor $\bar{u}_i$.

In order to obtain an impulse response matrix from a SISO Hammerstein system, it is enough to apply 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 $x_{\text{imp}}(t)$ for the impulse input $u_{\text{imp}}(t)$ is a rescaled version of $u_{\text{imp}}(t)$. The initial value is simply the value of the impulse multiplied by an

Algorithm 1. Impulse response constrained LS-SVM for MIMO Hammerstein System Identification.

Input: Multi-stage pseudo-random binary input signals for estimation of the linear part $\boldsymbol{U}_{LP} \in \mathbb{R}^{N\times p}$ (i.e. p inputs signals of length N) and their corresponding r outputs $\boldsymbol{y}_{LP} \in \mathbb{R}^{rN}$.Training inputs for the LS-SVM $\boldsymbol{U}_{train} \in \mathbb{R}^{N\times p}$ and their corresponding outputs $\boldsymbol{y}_{train} \in \mathbb{R}^{rN}$.Validation inputs $\boldsymbol{U}_{val} \in \mathbb{R}^{N\times p}$ and their corresponding outputs $\boldsymbol{y}_{val} \in \mathbb{R}^{rN}$.Test inputs $\boldsymbol{U}_{test} \in \mathbb{R}^{N\times p}$.

Output: Evaluation of the test output signal $\boldsymbol{y}_{test} \in \mathbb{R}^{rN}$;

1: Use $\boldsymbol{U}_{LP}$ and $\boldsymbol{y}_{LP}$ to estimate the matrix $\boldsymbol{Q}$ as shown in (8), (9) and (10).
2: Estimate $\hat{\boldsymbol{H}} = \boldsymbol{Q}\tilde{\boldsymbol{P}}^{-1}$ with $\tilde{\boldsymbol{P}}$ as defined in Theorem 1.
3: Create matrices $\boldsymbol{M}_i$ with $i = 1, 2, \ldots, p$ using $\hat{\boldsymbol{H}}$ as shown in (29).
4: Use matrices $\boldsymbol{M}_i$ to create matrix $\boldsymbol{M}$ as in (29).
5: Assemble a linear system like the one in (32) and using $\boldsymbol{U}_{train}, \boldsymbol{y}_{train}, \boldsymbol{U}_{val}$ and $\boldsymbol{y}_{val}$ proceed to tune the parameters $\sigma_1, \sigma_2, \ldots, \sigma_p$ and $\gamma_1, \gamma_2, \ldots, \gamma_r$.
6: Obtain $\boldsymbol{\alpha}$ and $\boldsymbol{b}$ from (32) using the estimated parameters.
7: Apply the found model to $\boldsymbol{U}_{test}$ to obtain $\hat{\boldsymbol{y}}_{test}$ as in (33).
8: **return** $\hat{\boldsymbol{y}}_{test}$.

unknown constant η, that is:

$$x_{\text{imp}}(t) = \begin{cases} \eta\bar{u}_i & \text{for } t = 0 \text{ with } \eta \neq 0 \\ 0 & \text{for } t \neq 0. \end{cases} \tag{35}$$

The linear part will be excited then by $x_{\text{imp}}(t)$ and the corresponding output $y_{\text{imp}}(t)$ can be used to construct an impulse response matrix $\boldsymbol{M}_1$ as shown in (2).

This is very convenient as a rescaled version of the impulse response of the system can be easily obtained. Note however that this impulse response will contain measurement noise.

It is important to highlight 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 behaviour 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).

The optimisation problem is formulated then as follows:

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{e}} \quad & \tfrac{1}{2}\boldsymbol{w}^\top\boldsymbol{w} + \tfrac{\gamma}{2}\boldsymbol{e}^\top\boldsymbol{e} \\ \text{subject to } & \boldsymbol{y} = \boldsymbol{M}_1(\boldsymbol{\Phi}_{\boldsymbol{u}}^\top\boldsymbol{w} + \boldsymbol{1}_N b) + \boldsymbol{e}. \end{aligned} \tag{36}$$

Again, the the input is $\boldsymbol{u} \in \mathbb{R}^N$, $\boldsymbol{\varphi}(\cdot) : \mathbb{R} \to \mathbb{R}^{n_h}$ is the feature map to a high-dimensional (possibly infinite) space $\boldsymbol{\Phi}_{\boldsymbol{u}} = [\boldsymbol{\varphi}(u_1), \ldots, \boldsymbol{\varphi}(u_N)] \in \mathbb{R}^{n_h\times N}$ and $\boldsymbol{w} \in \mathbb{R}^{n_h}$ is the weights vector. Also $\boldsymbol{y}, \boldsymbol{e}, \boldsymbol{1}_N \in \mathbb{R}^N$ with $\boldsymbol{y}$ the output, $\boldsymbol{e}$ the errors and $\boldsymbol{1}_N$ a vector of ones. It is interesting to highlight the high resemblance between (36) and (26). As can be seen, the SISO case is just a special case of the MIMO one.

From the Lagrangian:

$$\begin{aligned} \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{e}; \boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{w}^\top\boldsymbol{w} + \gamma\frac{1}{2}\boldsymbol{e}^\top\boldsymbol{e} - \boldsymbol{\alpha}^\top(\boldsymbol{M}_1(\boldsymbol{\Phi}_{\boldsymbol{u}}^\top\boldsymbol{w} \\ + \boldsymbol{1}_N b) + \boldsymbol{e} - \boldsymbol{y}), \end{aligned} \tag{37}$$

the optimality conditions are then derived:

$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial\boldsymbol{w}} &= 0 \to \boldsymbol{w} = \boldsymbol{\Phi}_{\boldsymbol{u}}\boldsymbol{M}_1^\top\boldsymbol{\alpha}, \\ \frac{\partial\mathcal{L}}{\partial b} &= 0 \to 0 = \boldsymbol{1}_N^\top\boldsymbol{M}_1^\top\boldsymbol{\alpha}, \\ \frac{\partial\mathcal{L}}{\partial e_i} &= 0 \to \boldsymbol{e} = \boldsymbol{\alpha}/\gamma, \\ \frac{\partial\mathcal{L}}{\partial\alpha_i} &= 0 \to \boldsymbol{y} = \boldsymbol{M}_1(\boldsymbol{\Phi}_{\boldsymbol{u}}^\top\boldsymbol{w} + \boldsymbol{1}_N b) + \boldsymbol{e}. \end{aligned} \tag{38}$$

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

$$\boldsymbol{y} = \boldsymbol{M}_1(\boldsymbol{\Phi}_{\boldsymbol{u}}^\top\boldsymbol{\Phi}_{\boldsymbol{u}}\boldsymbol{M}_1^\top\boldsymbol{\alpha} + \boldsymbol{1}_N b) + \frac{\boldsymbol{\alpha}}{\gamma} \tag{39}$$

with and $\boldsymbol{\Omega} = \boldsymbol{\Phi}_{\boldsymbol{u}}^\top\boldsymbol{\Phi}_{\boldsymbol{u}}$ and the following linear system is obtained:

$$\begin{bmatrix} 0 & \boldsymbol{1}_N^T\boldsymbol{M}_1^\top \\ \boldsymbol{M}_1\boldsymbol{1}_N & \boldsymbol{M}_1\boldsymbol{\Omega}\boldsymbol{M}_1^\top + \frac{1}{\gamma}\boldsymbol{I}_N \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{y} \end{bmatrix}. \tag{40}$$

Again note the resemblance of (40) and (32).

Finally, for a new input $\boldsymbol{u}_{\text{new}} \in \mathbb{R}^{N_n}$ the resulting model is then:

$$\hat{\boldsymbol{y}} = \bar{\boldsymbol{M}}_1\boldsymbol{1}_N b + \left(\bar{\boldsymbol{M}}_1\boldsymbol{K}\check{\boldsymbol{M}}_1^\top\right)\boldsymbol{\alpha}, \tag{41}$$

with $\boldsymbol{u}_{\text{train}}$ the set of inputs used to train the model, $\boldsymbol{K}_{ij} = k(\boldsymbol{u}_{\text{train},i}, \boldsymbol{u}_{\text{new},j})$ and $\boldsymbol{K} \in \mathbb{R}^{N\times N_n}$, $\bar{\boldsymbol{M}}_1 \in \mathbb{R}^{rN_n\times N}$ and $\check{\boldsymbol{M}}_1 \in \mathbb{R}^{rN\times N_n}$. Once more, if $N \neq N_n$ then matrix $\boldsymbol{M}_1$ has to be truncated or expanded to generate $\bar{\boldsymbol{M}}_1$ and $\check{\boldsymbol{M}}_1$ to comply with the new dimensions.

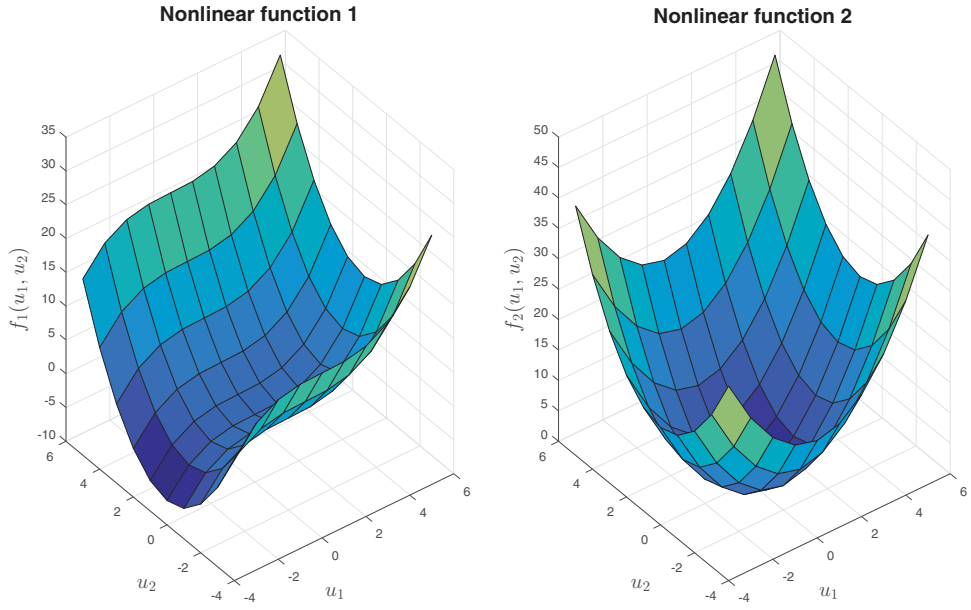

Figure 5. Nonlinear functions for Example 1. (Left) $f_1(u_1, u_2)$ and (right) $f_2(u_1, u_2)$.

4. Simulation results

4.1 Method steps

The proposed method was applied to three examples with two inputs and two outputs, consisting of two nonlinear functions and four LTI blocks as illustrated in Figure 3. The corresponding nonlinear functions of Example 1 are given in (42) and plotted in Figure 5.

$$f_1(u_1, u_2) = \frac{u_1^3}{10} + 0.9u_2^2, \tag{42a}$$

$$f_2(u_1, u_2) = u_1^2 + u_2^2. \tag{42b}$$

The transfer functions of the linear dynamic subsystems of Example 1 are presented in (43) and the magnitude of their frequency response is shown in Figure 6.

$$G_{11}(q) = \frac{0.9063}{q - 0.8187}, \tag{43a}$$

$$G_{12}(q) = \frac{1.572q + 1.323}{q^2 - 0.8828q + 0.6065}, \tag{43b}$$

$$G_{21}(q) = \frac{0.1969q^3 + 0.04616q^2 - 0.5395q - 0.1147}{q^4 - 0.9768q^3 + 0.1687q^2 - 0.268q + 0.2019}, \tag{43c}$$

$$G_{22}(q) = \frac{1.268q + 1.038}{q^2 - 1.452q + 0.5488}. \tag{43d}$$

Initially, the procedure described in Section 2.2 is applied in order to obtain an estimation of the impulse response of the different transfer functions. Note that as

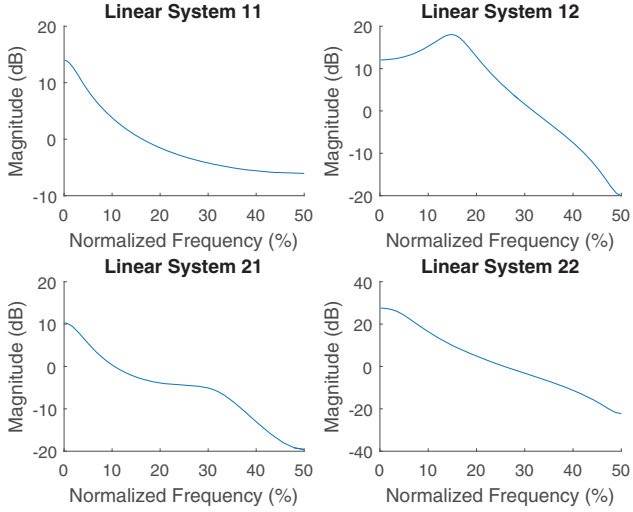

Figure 6. Magnitude of the frequency response of the LTI blocks for Example 1. (Upper left) $G_{11}(q)$, (upper right) $G_{12}(q)$, (lower left) $G_{21}(q)$ and (lower right) $G_{22}(q)$.

described, the parameters in matrix $\tilde{\boldsymbol{P}}$ are chosen arbitrarily and therefore the impulse responses obtained do not match exactly with the original ones. However, this will not impact the model from an input–output perspective as will be shown later. Once the impulse responses are calculated, the matrices $\boldsymbol{M}_i$ (i.e. see (28)) are created and the system in (32) can be solved. In this work, for the tuning of the parameters (i.e. σ_1, σ_2, γ_1 and γ_2) coupled simulated annealing (CSA) was used (Xavier-de Souza, Suykens, Vandewalle, & Bollé, 2009). In Figures 7 and 8, the results for the estimations of $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ in Example 1

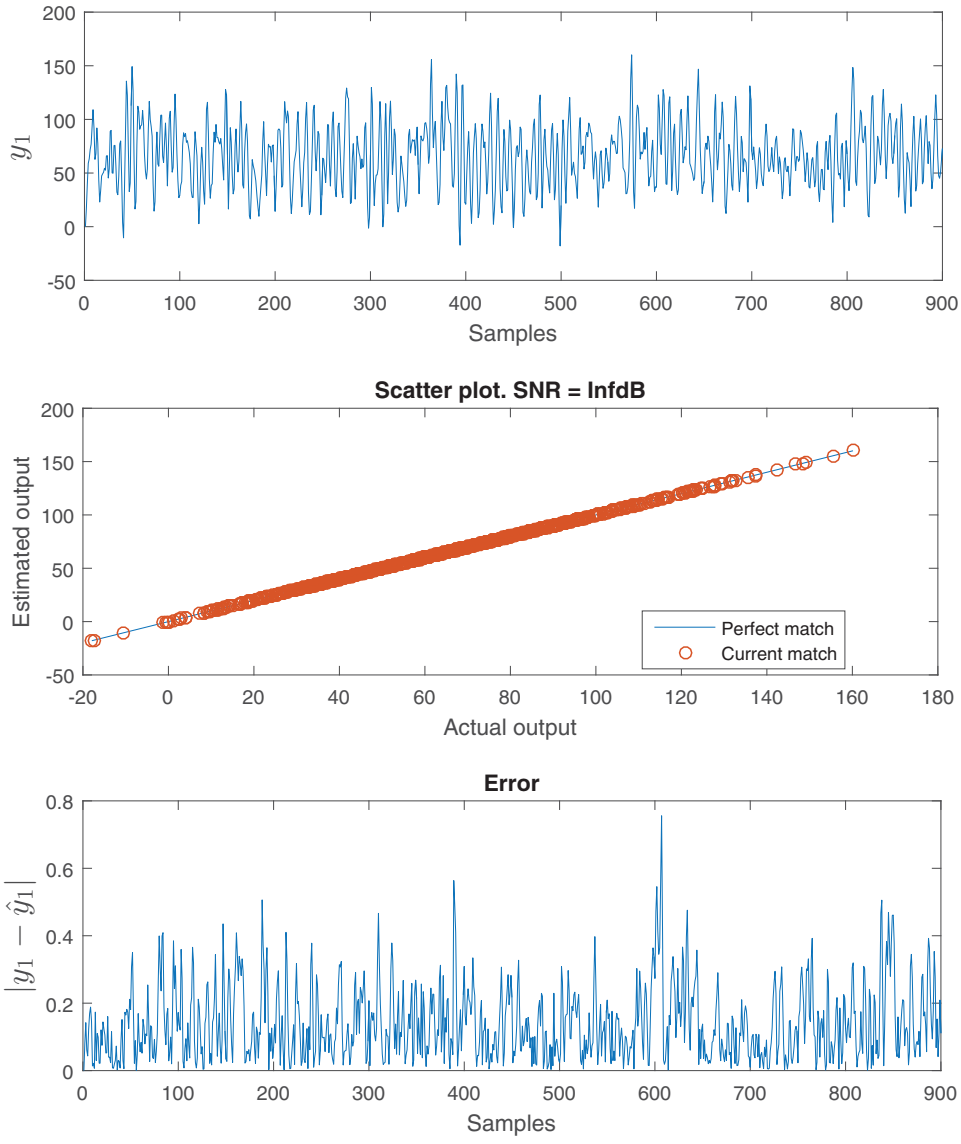

Figure 7. Example 1, output 1 simulation results. (Top) Real output. (Centre) Scatter plot between the actual and the estimated output. (Bottom) Absolute value of the difference between the actual and the estimated output.

are shown, respectively, for a no-noise case and an arbitrary $\tilde{\boldsymbol{P}} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}$.

In order to be able to make a comparison between the results, let us have the Normalised MAE defined as shown in (44) for a signal with N measurements. Note that the Normalised MAE uses the noise free output $\boldsymbol{y}_{\text{test}}$:

$$\%\text{MAE} = \frac{100}{N} \frac{\sum_{i=1}^{N} \left| \boldsymbol{y}_{\text{test},i} - \hat{\boldsymbol{y}}_{\text{test},i} \right|}{\left| \max(\boldsymbol{y}_{\text{test}}) - \min(\boldsymbol{y}_{\text{test}}) \right|}. \tag{44}$$

In Figure 9, the resulting estimations of the impulse response are shown for different values of $\tilde{\boldsymbol{P}}$ where the main diagonals are always 1. In Table 1, the corresponding values of $\tilde{\rho}_1$ and $\tilde{\rho}_2$ can be found together with the resulting %MAE in the test set. As can be seen, the results are quite similar despite the fact that the LTI approximations were different from the actual impulse response of

Table 1. Different entry values for $\tilde{\rho}_1$ and $\tilde{\rho}_2$ and the resulting %*MAE* in the test set.

$\tilde{\rho}_1$	$\tilde{\rho}_2$	%MAE of $\boldsymbol{y}_1$	%MAE of $\boldsymbol{y}_2$
7	1	0.268060647	0.1360575
7	5	0.120215049	0.037695636
2	10	0.425032546	0.17183245
10	9	0.230242274	0.122064228
2	9	0.223370104	0.137673664
4	6	0.639722418	0.196497981
4	9	0.132787457	0.096275247
10	6	0.308935392	0.154759024
6	10	0.194847013	0.092984275
10	10	0.151705355	0.085846982

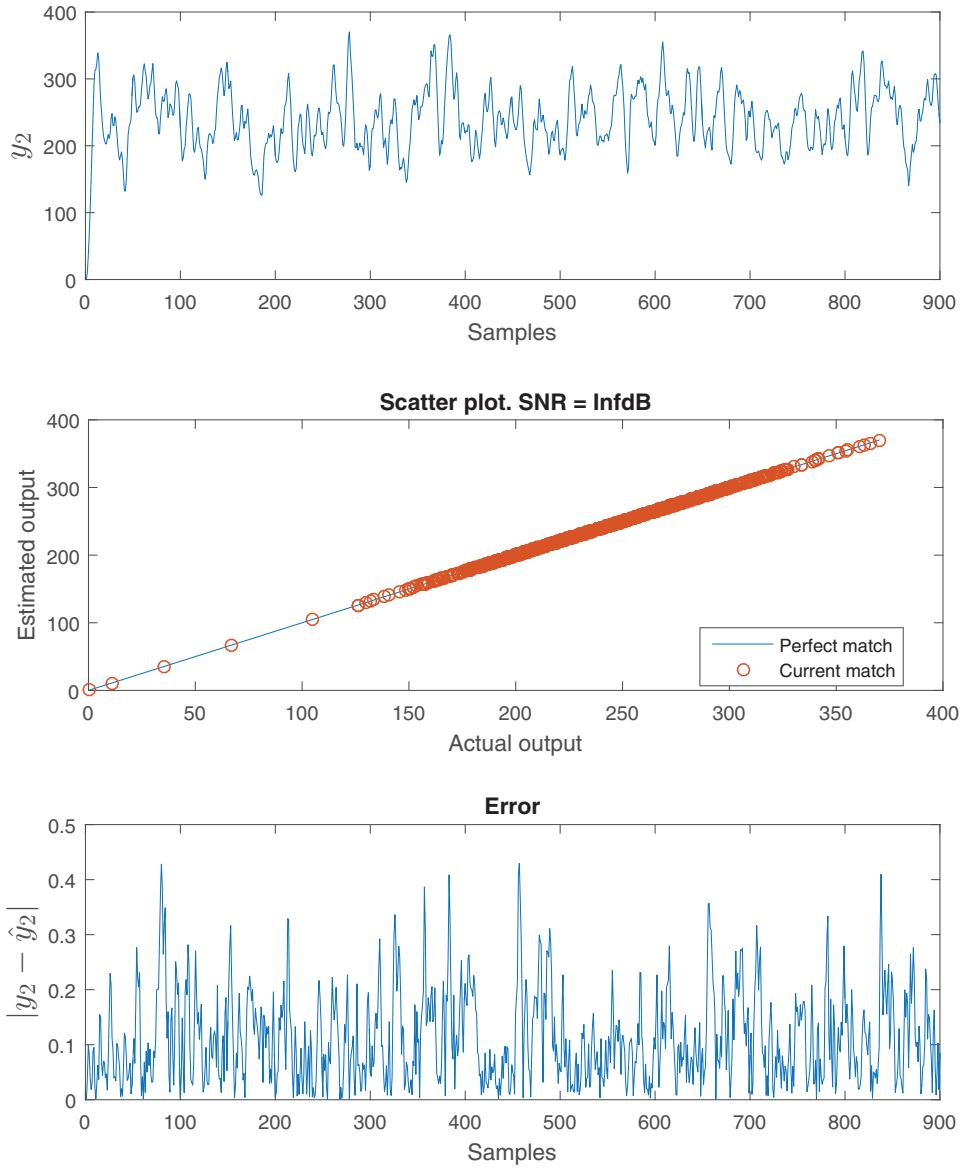

Figure 8. Example 1, output 2 simulation results. (Top) Real output. (Centre) Scatter plot between the actual and the estimated output. (Bottom) Absolute value of the difference between the actual and the estimated output.

the linear systems and among themselves. This was to be expected as explained in Theorem 2.1.

It is interesting to note that from (25) the estimated nonlinear functions can be retrieved by factorising $\bar{\boldsymbol{M}}_1$ and $\bar{\boldsymbol{M}}_2$ such that

$$\begin{aligned}\hat{\boldsymbol{y}} &= \bar{\boldsymbol{M}}_1(\mathbf{1}_N b_1 + (\boldsymbol{K}^{(1)}\breve{\boldsymbol{M}}_1^\top)\boldsymbol{\alpha}) \\ &\quad + \bar{\boldsymbol{M}}_2(\mathbf{1}_N b_2 + (\boldsymbol{K}^{(2)}\breve{\boldsymbol{M}}_2^\top)\boldsymbol{\alpha}), \\ &= \bar{\boldsymbol{M}}_1\hat{\boldsymbol{x}}_1 + \bar{\boldsymbol{M}}_2\hat{\boldsymbol{x}}_2, \\ &= \bar{\boldsymbol{M}}_1\hat{\boldsymbol{f}}_1(\boldsymbol{u}_1, \boldsymbol{u}_2) + \bar{\boldsymbol{M}}_2\hat{\boldsymbol{f}}_2(\boldsymbol{u}_1, \boldsymbol{u}_2).\end{aligned} \tag{45}$$

For Example 1, the estimated nonlinear functions are presented in Figure 10. Note that the estimated nonlinear functions are linear combinations of the original ones as illustrated in Figure 4. When the cross terms are comparatively smaller than the corresponding diagonal terms (i.e. $g_{12} \ll g_{11}$ and $g_{22} \ll g_{21}$), the estimated functions will be more similar to the original ones.

4.2 Signals description

The inputs used for the proposed method had $N = 900$ points. Note that as described in Section 2.2 a special set of signals is used for the identification of the impulse responses of the system. In this case, PRBSs ranging between 0 and 1 were used with a 5% switching probability and consisting of $2N$ points. Upper and lower limits for all inputs where set (i.e. $\boldsymbol{u}_1, \boldsymbol{u}_2 \in [-4, 5]$).

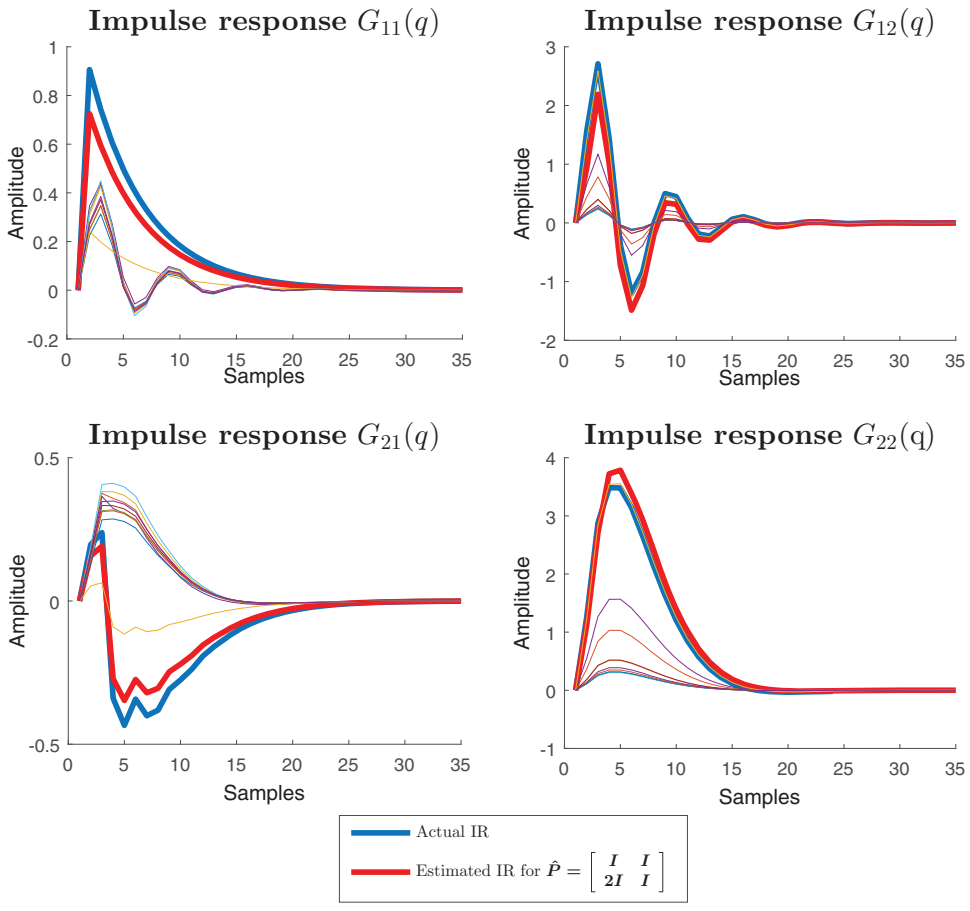

Figure 9. Estimation of the impulse responses. The actual impulse response and the approximated impulse response used are presented in thick lines. The other plots represent approximations obtained through the values of $\tilde{\rho}_1$ and $\tilde{\rho}_2$ presented in Table 1. (Upper left) $G_{11}(q)$, (upper right) $G_{12}(q)$, (lower left) $G_{21}(q)$ and (lower right) $G_{22}(q)$.

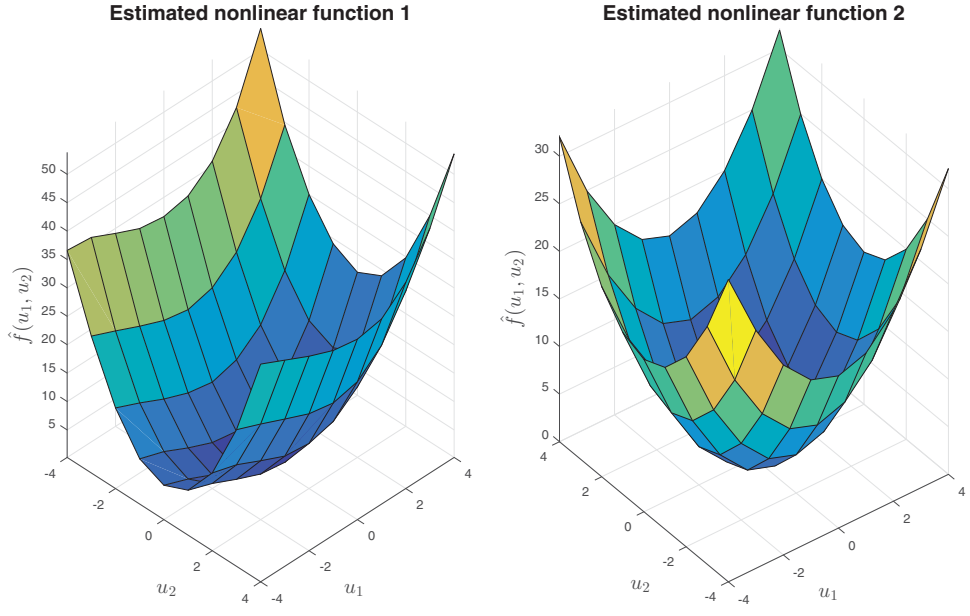

Figure 10. Estimated nonlinear functions for Example 1. (Left) $\hat{f}_1(u_1, u_2)$ and (Right) $\hat{f}_2(u_1, u_2)$.

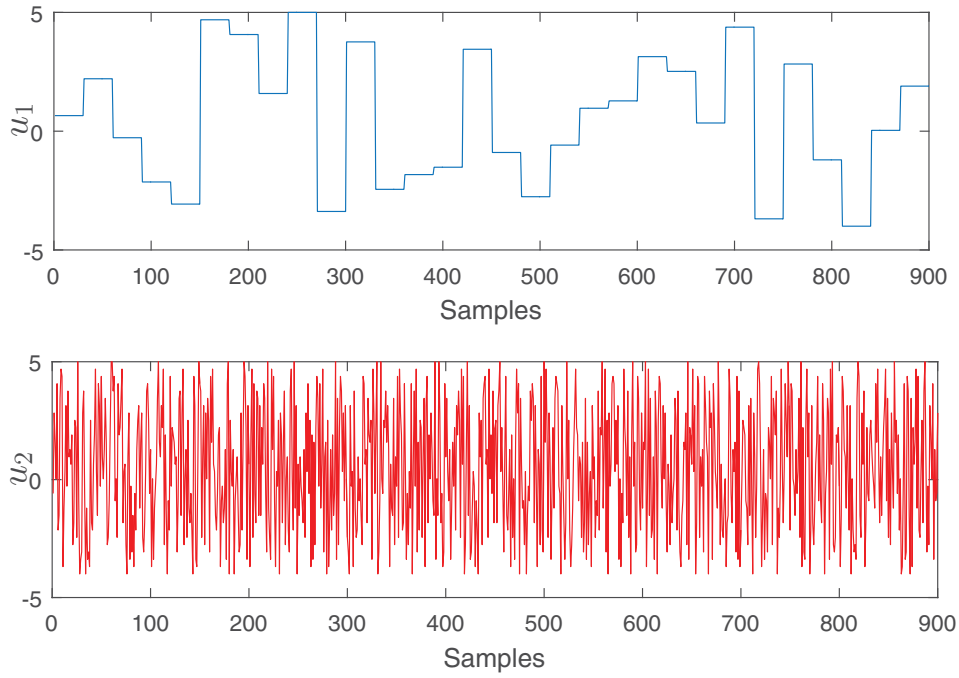

Figure 11. Training set for the LS-SVM part.

To generate the training set for the reformulated LS-SVM N is chosen such that $\sqrt{N} \in \mathbb{N}$. We select $\sqrt{N}$ values between the upper an lower limits of $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ (i.e. $v_i^{(1)}$ and $v_i^{(2)}$ for $i = 1, \ldots, \sqrt{N}$). An uniform distribution covering the whole range is used (i.e. the difference between any pair of consecutive values is constant). To guarantee that all the combinations of the selected values of $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ are considered, the procedure is as follows:

- For each value $v_i^{(1)}$ create a vector of size $\sqrt{N}$ where all the elements are $v_i^{(1)}$.
- Randomly permute the order of the resulting vectors and concatenate them. This constitutes $\boldsymbol{u}_1 \in \mathbb{R}^N$ which is the first input of the training set.
- Randomly permute the order of the selected values $v_i^{(2)}$ to create a vector of size $\sqrt{N}$.
- Repeat the last step $\sqrt{N}$ times and concatenate the results. This constitutes $\boldsymbol{u}_2 \in \mathbb{R}^N$ which is the second input of the training set.

With this procedure, it is guaranteed that all the possible combinations of $v_i^{(1)}$ and $v_i^{(2)}$ are considered, see Figure 11 for an illustration.

For the tuning of the parameters described in Section 3.1 validation was used where the inputs were taken randomly from a uniform distribution between the upper and lower bounds defined for the training set. In this case, we opted for plain validation and therefore used 10 validation sets, all of them generated in the same fashion. Note however that this could be replaced with a cross-validation scheme. In a similar way, the inputs of the test set were created by taking random values from a uniform distribution between the upper and lower bounds defined for the training set.

4.3 Noise effect analysis

In Section 4.1, the proposed method was explored step by step through Example 1 and to make the explanation more clear no noise was used. Nevertheless, for any method to be useful for system identification it is expected to be robust against noise. To show how the proposed method behaves in the presence of noise, Examples 2 and 3 are introduced and the results of 100 Monte Carlo simulations are offered for all examples and for different levels of white Gaussian noise with zero mean. For these simulations, signals as the ones described in Section 4.2 were employed.

For Example 2, the corresponding nonlinear functions are presented in (46) and plotted Figure 12.

$$f_1(u_1, u_2) = \frac{u_1^3}{5} + \sin(u_2)u_2^2, \tag{46a}$$

$$f_2(u_1, u_2) = 10\sin(u_1) + u_2^2. \tag{46b}$$

The transfer functions of the linear dynamic subsystems of Example 2 are given in (47) and the magnitude of their frequency response is shown in Figure 13.

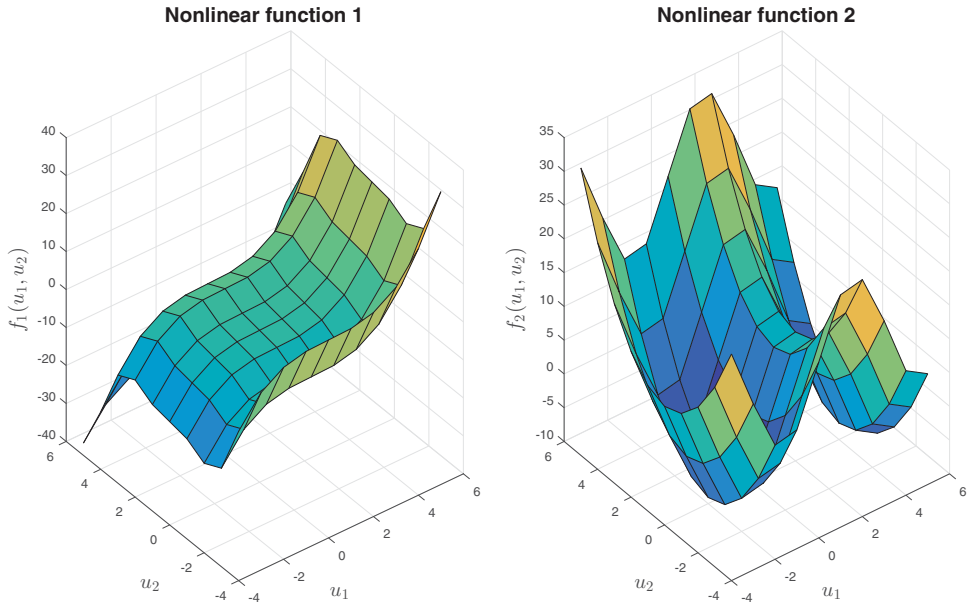

Figure 12. Nonlinear functions for Example 2. (Left) $f_1(\boldsymbol{u}_1, \boldsymbol{u}_2)$ and (right) $f_2(\boldsymbol{u}_1, \boldsymbol{u}_2)$.

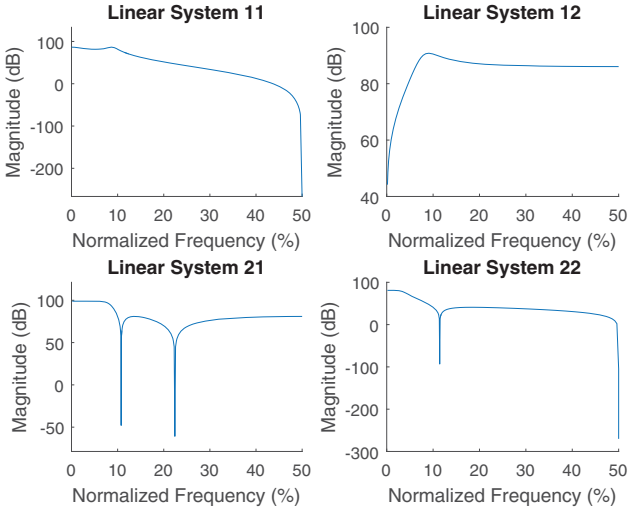

Figure 13. Magnitude of the frequency response of the LTI blocks for Example 2. (Upper left) $G_{11}(q)$, (upper right) $G_{12}(q)$, (lower left) $G_{21}(q)$ and (lower right) $G_{22}(q)$.

$$G_{11}(q) = \frac{100q^3 + 300q^2 + 300q + 100}{q^3 - 2.458q^2 + 2.262q - 0.7654}, \tag{47a}$$

$$G_{12}(q) = \frac{18,000q^2 - 32400q + 14400}{q^2 - 1.5q + 0.7225}, \tag{47b}$$

$$G_{21}(q) = \frac{10,000q^4 - 1.884e04q^3 + 2.506e04q^2 - 1.884e04q + 1e04}{q^4 - 2.485q^3 + 2.528q^2 - 1.184q + 0.2245}, \tag{47c}$$

$$G_{22}(q) = \frac{100q^3 - 50.64q^2 - 50.64q + 100}{q^3 - 2.564q^2 + 2.218q - 0.6456}. \tag{47d}$$

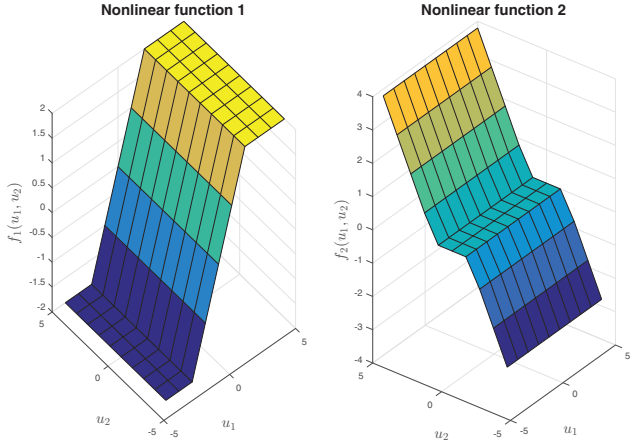

Figure 14. Nonlinear functions for Example 3. (Left) $f_1(u_1, u_2)$ and (right) $f_2(u_1, u_2)$.

Functions including saturation or dead-zones are generally regarded as difficult to learn. In Example 3, we include one of each of these functions. The corresponding nonlinear functions are presented in (48) and plotted Figure 14.

$$f_1(u_1, u_2) = \begin{cases} -2 & \text{for } u_1 < -2 \\ u_1 & \text{for } -2 \leq u_1 < 2, \\ 2 & \text{for } u_1 \geq 2 \end{cases} \tag{48a}$$

$$f_2(u_1, u_2) = \begin{cases} u_2 - 1.2 & \text{for } u_2 > 1.2 \\ 0 & \text{for } -1.2 \leq u_2 \leq 1.2. \\ u_2 + 1.2 & \text{for } u_2 < -1.2 \end{cases} \tag{48b}$$

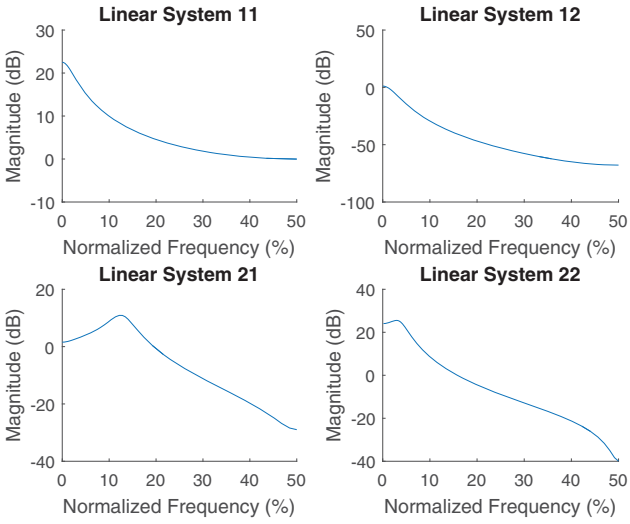

Figure 15. Magnitude of the frequency response of the LTI blocks for Example 3. (Upper left) $G_{11}(q)$, (upper right) $G_{12}(q)$, (lower left) $G_{21}(q)$ and (lower right) $G_{22}(q)$.

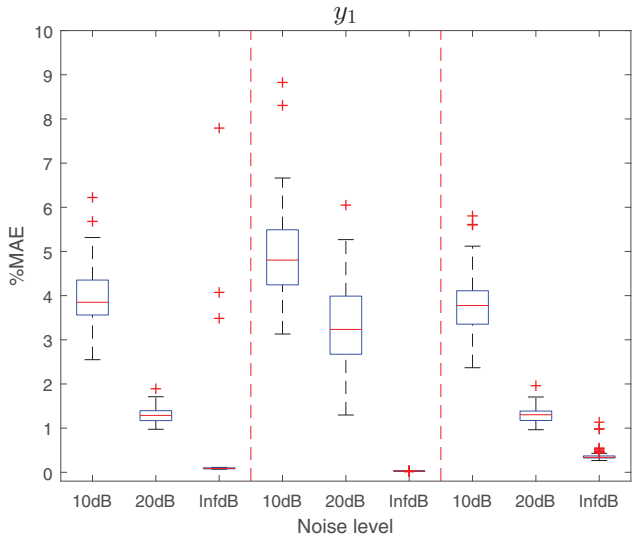

Figure 16. Monte Carlo simulation of the proposed method. Output 1 for Examples 1 (left), 2 (centre) and 3 (right).

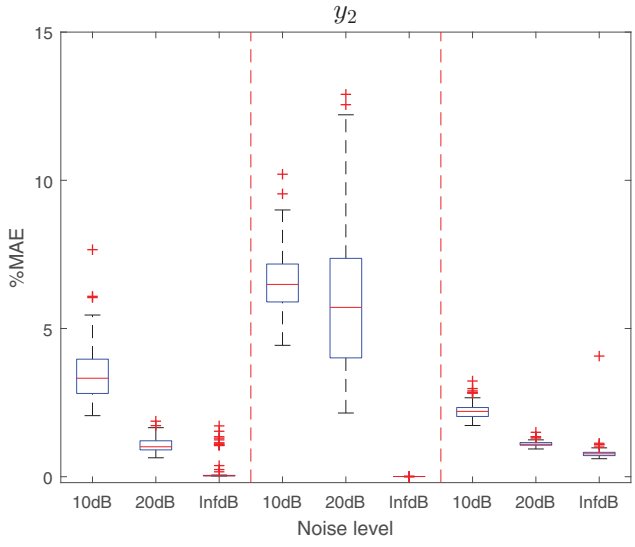

Figure 17. Monte Carlo simulation of the proposed method. Output 2 for Examples 1 (left), 2 (centre) and 3 (right).

The transfer functions of the linear dynamic subsystems of Example 3 are given in (49) and the magnitude of their frequency response is shown in Figure 15.

$$G_{11}(q) = \frac{1.857}{q - 0.8607}, \tag{49a}$$

$$G_{12}(q) = \frac{0.001429q^2 + 0.004898q + 0.001048}{q^3 - 2.44q^2 + 1.984q - 0.5379}, \tag{49b}$$

$$G_{21}(q) = \frac{-0.3612q^2 + 0.1623q + 0.3408}{q^3 - 1.954q^2 + 1.627q - 0.5543}, \tag{49c}$$

$$G_{22}(q) = \frac{0.4599q + 0.4245}{q^2 - 1.731q + 0.7866}. \tag{49d}$$

The results for all examples with different levels of noise are presented in Figures 16 and 17. In Table 2, the resulting medians are offered to summarise the results.

4.4 Methods comparison

The proposed method, from now on referred to as IR-H-LS-SVM, is compared with three other methods for MIMO Hammerstein system identification, namely:

- NARX LS-SVM (Suykens et al., 2002).
- The method in Jeng and Huang (2008) where an approximation to the input response of the system is obtained and with it and the known output an estimation of the intermediate variable is found. Using this approximation and the known input, a mapping of the nonlinearity is done through the fitting of a polynomial. From now on, this method will be referred to as IR H-MIMO.
- Using orthonormal bases for the identification of block-oriented nonlinear systems is proposed in Gomez and Baeyens (2004). This method will be referred to as ONBF.

The NARX LS-SVM approach is not a method specifically developed for the identification of MIMO Hammerstein Systems. However, it is still relevant in this work as it allows to see how the LS-SVM framework, though powerful by itself, can be improved by including additional information from the structure of the system. The training set was generated under the same guidelines as those described in Section 4.2 with the exception of the number

Table 2. %MAE comparison for the proposed method. Median values are offered for 100 Monte Carlo simulations for each case.

	Example 1		Example 2		Example 3	
	y_1	y_2	y_1	y_2	y_1	y_2
SNR 10dB	3.8499	3.3225	4.8041	6.4876	3.7755	2.2042
SNR 20dB	1.2885	1.0098	3.235	5.7129	1.3033	1.095
SNR InfdB	0.086366	0.032617	0.028782	0.0069262	0.33918	0.77252

of points as in this case $N = 2500$. For the hyper parameter tuning, a 10-fold cross validation scheme was used. Also, a preliminary Monte Carlo experiment was carried out to determine the optimal number of input and output lags to be used for each example. During these experiments, input and output lags from 4 to 10 were tested. For Example 1, 4 lags of input and 4 lags of output were selected. For Example 2, 10 lags of input and 10 lags of output were chosen. For Example 3, 9 lags of input and 9 lags of output were used.

For the IR H-MIMO method, PRBS of 800 samples were created in order to identify the linear part. These signals followed the same guidelines described in Section 2.2, that is: In a first stage, $\boldsymbol{u}_1$ was a PRBS and $\boldsymbol{u}_2$ was kept at 0. Then, in a second stage $\boldsymbol{u}_1$ was 0 and $\boldsymbol{u}_2$ was a PRBS. After the impulse responses were estimated, the nonlinear part was modeled. To do this, signals of 980 points were used of which the last 80 where included to make the corresponding linear system overdetermined. The initial 900 points where generated guaranteeing that all combinations of 30 points drawn from a uniform distribution between -5 and 5 were included. With these signals, the nonlinearities were estimated by fitting two-dimensional polynomials with degrees 3, 7 and 9 for Examples 1, 2 and 3, respectively.

For the ONBF method, polynomial basis functions were used for identifying the nonlinearity: For Example 1 until degree 3, for Example 2 until degree 5 and for Example 3 until degree 7. Empirically, it was found that the use of simpler basis functions yielded better results for the modeling of the linear part and in consequence q^{-n} was used. For Example 1, the number of bases used was 10 and for Examples 2 and 3 it was 40. These values were set by trial and error and were the ones that offered a good trade-off between complexity and accuracy. For Example 1, 1600 data points were used and for Examples 2 and 3, 3600 were employed.

The results are summarised in Table 3. For ease of comparison, the results from Table 2 are included again. The best results for each case are in bold case.

As can be seen, the proposed method performs quite well when compared with the other methods considered. It is the best in 12 out of the 18 cases analysed and in general has a very good and stable performance. By this, it is meant that comparatively good results are obtained in all cases. Also, it can be seen that the proposed method is robust against the type of noise employed, this is particularly evident for the SNR $= 10dB$ case where our method performs better than the others.

Note that the class model of the nonlinear part of Example 1 is polynomial which makes its identification much easier for methods using polynomial bases (as is the case for IR H-MIMO and ONBF). However, representing the nonlinear part of Examples 2 and 3 using polynomial

Table 3. %MAE comparison for the different methods tested. Median values are offered for 100 Monte Carlo simulations. The best results for each case are highlighted in bold text.

		Example 1		**Example 2**		**Example 3**	
		y_1	y_2	y_1	y_2	y_1	y_2
SNR 10dB	**IR H-LS-SVM**	**3.8499**	**3.3225**	4.8041	**6.4876**	**3.7755**	**2.2042**
	NARX LS-SVM	8.6475	10.5307	12.3816	9.9141	5.3544	3.2651
	IR H-MIMO	6.6443	4.9827	15.5643	20.2475	9.0841	21.1168
	ONBF	4.529	4.6303	**2.954**	6.8069	4.587	13.883
SNR 20dB	**IR H-LS-SVM**	**1.2885**	**1.0098**	3.235	5.7129	**1.3033**	**1.095**
	NARX LS-SVM	5.2555	5.5722	13.119	6.3674	3.9329	2.319
	IR H-MIMO	2.2933	2.9713	3.1664	7.2392	3.2163	6.6558
	ONBF	2.8237	1.6546	**1.5856**	**4.3417**	4.0638	6.0554
SNR InfdB	**IR H-LS-SVM**	0.086366	0.032617	**0.028782**	**0.0069262**	**0.33918**	0.77252
	NARX LS-SVM	3.9083	2.9626	14.9459	7.743	2.7757	1.0772
	IR H-MIMO	$\mathbf{1.8149 \times 10^{-6}}$	$\mathbf{2.3505 \times 10^{-7}}$	0.23985	0.98531	0.51268	**0.43857**
	ONBF	0.98313	0.9845	1.3339	3.8984	0.9663	1.3401

basis functions can be more difficult as it does not belong to the class problem anymore. However, this is not a problem for the proposed method which does not require previous knowledge about the class problem.

Finally, note that if the cross-validation methodology were employed, the proposed method would be the one using the least number of points as no validation sets would be required.

5. Conclusions

In this paper, a new methodology for identifying MIMO Hammerstein systems is presented where the impulse response of the system is incorporated into an LS-SVM formulation. This means that the model found benefits from the regularisation capabilities of LS-SVM.

The method proposed is very flexible with regard to the number of inputs and outputs it can handle and it is shown that it can model both SISO and MIMO systems. In addition, it has very good generalisation capabilities and can work with different classes of problem. This constitutes a nice advantage when the class of problem is unknown or is difficult to model with certain basis functions.

It is shown that the proposed method is robust against the type of noise employed as even in the presence of high levels of noise it has a comparatively good performance, in fact, for the examples presented it performed better than the other methods.

It is interesting to note that even though $\tilde{\boldsymbol{P}}$ is chosen arbitrarily, the results are largely unchanged. This allows for certain flexibility when modelling the system.

Information from the structure of the system is included into the LS-SVM formulation allowing for an improvement of the modelling capabilities of LS-SVM for this type of systems. Nevertheless, the model is still obtained from a linear system of equations which represents an advantage.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

Agentschap voor Innovatie door Wetenschap en Technologie [grant number POM II SBO 100031]; Federaal Wetenschapsbeleid [grant number IUAP P7/19]; Fonds Wetenschappelijk Onderzoek [grant number G.0377.12,G.088114N]; H2020 European Research Council [grant number AdG A-DATADRIVE-B (290923),FP7/2007-2013]; Research Council KUL [grant number CoE PFV/10/002 (OPTEC),BIL12/11T]; iMinds [grant number SBO 2015].

ORCID

Ricardo Castro-Garcia http://orcid.org/0000-0002-2332-9381

References

Al-Duwaish, H., & Karim, M. N. (1997). A new method for the identification of Hammerstein model. *Automatica, 33*(10), 1871–1875.

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

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

Boyd, S., & Chua, L. O. (1983). Uniqueness of a basic nonlinear structure. *IEEE Transactions on Circuits and Systems, 30*(9), 648–651.

Castro-Garcia, R., Agudelo, O. M., & Suykens, J. A. K. (2017). Impulse response constrained LS-SVM modeling for Hammerstein system identification. *Proceedings of the 20th World Congress of the International Federation of Automatic Control (IFAC2017)*, Toulouse, France. July, 2017.

Castro-Garcia, R., , Tiels, K., Schoukens, J., & 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)* (pp. 7392–7397). IEEE, 10.1109/cdc.2015.7403387.

De Brabanter, K., , Dreesen, P., Karsmakers, P., Pelckmans, K., De Brabanter, J., Suykens, J. A. K., & De Moor, B. (2009). Fixed-size LS-SVM applied to the Wiener-Hammerstein benchmark. In *Proceedings of the 15th IFAC Symposium on System Ientification (SYSID 2009)*, Saint-Malo, France (pp. 826–831).

Eskinat, E., Johnson, S. H., & 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., & De Moor, B. (2004). A comparative study of LS-SVMs applied to the silverbox identification problem. *Proceeding of the 6th IFAC Symposium on Nonlinear Control Systems (NOLCOS)*, Stturgart, Germany.

Falck, T., Dreesen, P., De Brabanter, K., Pelckmans, K., De Moor, B., & 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., & De Moor, B. (2009). Identification of Wiener-Hammerstein systems using LS-SVMs. In *Proceedings of the 15th IFAC Symposium on System Identification (SYSID 2009)*, Saint-Malo, France (pp. 820–825).

Fruzzetti, K. P., Palazoglu, A., & McDonald, K. A. (1997). Nolinear model predictive control using Hammerstein models. *Journal of Process Control, 7*(1), 31–41.

Giri, F., & Bai, E. -W. (Eds.). (2010). *Block-oriented nonlinear system identification*. Vol. 1, Springer-Verlag Berlin Heidelberg.

Goethals, I., Pelckmans, K., Suykens, J. A. K., & De Moor, B. (2005). Identification of MIMO Hammerstein models using

least-squares support vector machines. *Automatica, 41*(7), 1263–1272.

Gomez, J. C., & Baeyens, E. (2004). Identification of block-oriented nonlinear systems using orthonormal bases. *Journal of Process Control, 14*(6), 685–697.

Haber, R., & Unbehauen, H. (1990). Structure identification of nonlinear dynamic systems – A survey on input/output approaches. *Automatica, 26*(4), 651–677.

Haber, R., & Keviczky, L. (1999). *Nonlinear system identification – input-output modeling approach*. Vol. 1, The Netherlands: Springer.

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

Hunter, I. W., & Korenberg, M. J. (1986). The identification of nonlinear biological systems: Wiener and Hammerstein cascade models. *Biological Cybernetics, 55*(2–3), 135–144.

Janczak, A. (2005). *Identification of nonlinear systems using neural networks and polynomial models: A block-oriented approach* Vol. 310. Berlin: Springer-Verlag Berlin Heidelberg.

Jeng, J.-C., & Huang, H.-P. (2008). Nonparametric identification for control of MIMO Hammerstein systems. *Industrial & Engineering Chemistry Research, 47*(17), 6640–6647.

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

Lee, Y. J., Sung, S. W., Park, S., & Park, S. (2004). Input test signal design and parameter estimation method for the Hammerstein-Wiener processes. *Industrial & Engineering Chemistry Research, 43*(23), 7521–7530.

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

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* (pp. 415–446).

Schoukens, J., Suykens, J. A. K., & Ljung, L. (2009). Wiener-Hammerstein benchmark. *Proceedings of the 15th IFAC Symposium on System Identification*, Saint-Malo, France. July, 2009.

Stapleton, J., & Bass, S. (1985). Adaptive noise cancellation for a class of nonlinear, dynamic reference channels. *IEEE transactions on circuits and systems, 32*(2), 143–150.

Suykens, J. A. K., Van Gestel, T., De Brabanter, J., De Moor, B., & Vandewalle, J. (2002). *Least squares support vector machines*, Singapore. World Scientific.

Verhaegen, M., & Westwick, D. (1996). Identifying MIMO Hammerstein systems in the context of subspace model identification methods. *International Journal of Control, 63*(2), 331–349.

Wang, J., Chen, Q., & Chen, Y. (2004). RBF kernel based support vector machine with universal approximation and its application. In *International symposium on neural networks* (pp. 512–517). Springer.

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