

# Support Vector Machines for Nonlinear Kernel ARMA System Identification

Manel Martínez-Ramón, *Senior Member, IEEE*, José Luis Rojo-Álvarez, *Member, IEEE*,  
Gustavo Camps-Valls, *Member, IEEE*, Jordi Muñoz-Marí, Ángel Navia-Vázquez, *Senior Member, IEEE*,  
Emilio Soria-Olivas, Aníbal R. Figueiras-Vidal, *Senior Member, IEEE*

## Abstract

Nonlinear system identification based on Support Vector Machines (SVM) has been usually addressed by means of the standard SVM regression (SVR), which can be seen as an implicit nonlinear Auto-Regressive and Moving Average (ARMA) model in some Reproducing Kernel Hilbert Spaces (RKHS). The proposal of this paper is twofold. First, the explicit consideration of an ARMA model in RKHS (SVM-ARMA<sub>2K</sub>) is originally proposed. We show that stating the ARMA equations in RKHS leads to solving the regularized normal equations in that RKHS, in terms of the auto-correlation and cross-correlation of the (nonlinearly) transformed input and output discrete time processes. Second, a general class of SVM-based system identification nonlinear models is presented, based on the use of composite Mercer's kernels. This general class can improve model flexibility by emphasizing the input-output cross information (SVM-ARMA<sub>4K</sub>), which lead to straightforward and natural combinations of implicit and explicit ARMA models (SVR-ARMA<sub>2K</sub> and SVR-ARMA<sub>4K</sub>). Capabilities of the different SVM-based system identification schemes are illustrated with two benchmark problems.

## I. INTRODUCTION

A common problem in digital signal processing is to model a functional relationship between two simultaneously recorded discrete-time processes (DTP) [1]. When this relationship is linear and time-invariant, it is usually addressed with Auto-Regressive and Moving Average (ARMA) modeling, but if linearity can not be assumed, nonlinear system identification techniques are required. General nonlinear models, such as artificial neural networks, wavelet, and fuzzy models, are common and effective choices [1], [2], but the temporal structure of these nonlinear models can not be easily analyzed, because it remains inside a black-box model.

Support Vector Machines (SVM) were originally conceived for pattern recognition and classification tasks [3], and Support Vector Regression (SVR) was subsequently proposed as the SVM implementation for regression and function approximation [4]. A main advantage of SVM current algorithms is their capability for giving nonlinear algorithms by the statement of a well-known linear data model (classification or regression) in a nonlinearly

MMR, JLRA, ANV, and ARFV, are with Dep. Teor. Señal y Comunic., Univ. Carlos III Madrid, Spain ({manel,jlrojo,navia,arfv}@tsc.uc3m.es).  
GCV, JMM, and ESO, are with Dep. d'Enginyeria Electrònica, Univ. València, Spain ({gcamps,jordi,soriae}@uv.es).

transformed domain, known as Reproducing Kernel Hilbert Space (RKHS). In [5]–[9], SVR algorithm was used for nonlinear system identification, but the time series structure of the data was not scrutinized. In [10], SVM was explicitly formulated for modeling linear time-invariant ARMA systems (linear SVM-ARMA), and this kind of formulation has been recently extended to a general framework for linear signal processing problems [11].

This letter introduces an explicit formulation of the ARMA data structure in RKHS by using the well known *kernel trick*. The so-called SVM-ARMA<sub>2k</sub> allows us to study the time series structure on a straightforward and natural way. Additionally, we introduce a general and still simple class of SVM-based system identification algorithms, by using *composite kernels*. In this context, a second algorithm is presented to take into account the input-output cross information (SVM-ARMA<sub>4k</sub>). A full family of natural combinations of implicit and explicit ARMA models (SVR-ARMA<sub>2k</sub> and SVR-ARMA<sub>4k</sub> algorithms) is finally proposed.

The scheme of this work is as follows. Section II summarizes the SVR algorithm for nonlinear system identification. Section III presents the novel formulation of an explicit ARMA models in the RKHS. Section IV introduces the use of composite kernels for further model flexibility. Section V shows the advantages of the proposed methods with benchmark examples. Section VI gives conclusions and outlines future work.

## II. IMPLICIT ARMA SYSTEM IDENTIFICATION WITH SVR

Previous SVM-based approaches to nonlinear system identification have taken advantage of both the kernel trick and the well-developed SVR algorithmic implementations [5]–[9]. The nonlinear SVR-based system identification is briefly presented in this section, with the aims of reviewing the kernel trick, highlighting the implicit ARMA nature of this problem statement, and introducing the  $\varepsilon$ -Huber cost function in this setting.

### A. Mercer's kernels and nonlinearity

Let us consider two DTP,  $\{u_n\}$  and  $\{y_n\}$ , which are the input and the output, respectively, of a nonlinear system. Let  $\mathbf{y}_{n-1} = [y_{n-1}, y_{n-2}, \dots, y_{n-P}]^T$  and  $\mathbf{u}_n = [u_n, u_{n-1}, \dots, u_{n-Q+1}]^T$  denote the states of input and output DTP at time instant  $n$ , so that  $\mathbf{z}_n = [\mathbf{y}_{n-1}^T, \mathbf{u}_n^T]^T$  is just the vector concatenation of input and output states at that instant. We assume that  $P$  and  $Q$  are large enough so that the predictable part of the process is completely captured. The SVR-based system identification uses a nonlinear transformation  $\phi_z(\mathbf{z}_n) : \mathbb{R}^P \times \mathbb{R}^Q \rightarrow \mathcal{H}_z$ , which maps the concatenation vector to an RKHS  $\mathcal{H}_z$ , or *feature space*. For a properly chosen transformation  $\phi_z$ , a linear regression model can be built in  $\mathcal{H}_z$ , and it is given by

$$y_n = \langle \mathbf{v}, \phi_z(\mathbf{z}_n) \rangle + e_n \quad (1)$$

where  $\mathbf{v} \in \mathcal{H}_z$ ,  $\langle \cdot, \cdot \rangle$  denotes the dot product, and  $\{e_n\}$  is a DTP standing for the effect of measurement errors.

In general, the RKHS dimension ( $H_z$ ) will be greater than the input space dimension ( $P + Q$ ), and for some choices of the transformation it can be even infinite. However, the SVM methodology allows to still work in that high-dimensional RKHS by using Mercer's kernels [12]. If a bivariate function  $K_z(\mathbf{z}_i, \mathbf{z}_j)$  fulfils the Mercer's condition, i.e.,  $\int K_z(\mathbf{z}_i, \mathbf{z}_j) f(\mathbf{z}_i) f(\mathbf{z}_j) \geq 0$  for any square integrable functions  $f(\mathbf{z})$ , then there exist a Hilbert

space  $\mathcal{H}_z$  and a mapping  $\phi_z$ , such that  $K_z(\mathbf{z}_i, \mathbf{z}_j) = \langle \phi_z(\mathbf{z}_i), \phi_z(\mathbf{z}_j) \rangle$ . Therefore, the kernel trick in SVM consists of stating the problem at hand (such as classification, regression, and many others) in terms of dot products in the RKHS, and then substituting these products by Mercer's kernels. The kernel expression is actually used in a given SVM algorithm, but neither the mapping function  $\phi_z$ , nor the RKHS, need to be explicitly known.

The widely used Gaussian Mercer's kernel is given by  $K_z(\mathbf{z}_i, \mathbf{z}_j) = \exp\left(-\frac{\|\mathbf{z}_i - \mathbf{z}_j\|^2}{2\sigma^2}\right)$ , where  $\sigma$  is the kernel width and can be seen as a parameter that controls the distortion in the input space to provide with a RKHS in which linear regression (1) is an appropriate model. For the Gaussian kernel, the explicit expression of nonlinear transformation  $\phi_z(\mathbf{z})$  is unknown, and the corresponding RKHS dimension is infinite [12].

### B. The $\varepsilon$ -Huber residual cost

SVM algorithms minimize a cost function of the residuals (CFR) that is regularized with the  $L_2$  norm of the model coefficients in a RKHS. Two CFRs have been used in previous SVM-based system identification research: (1) the  $\varepsilon$ -insensitive CFR [5], [6], which yields sparse solutions, is essentially a  $L_1$  cost, and hence appropriate for dealing with outliers, and (2) the quadratic CFR used in Least-Squares SVM approaches [7]–[9], which is optimal when Gaussian noise is present, but it is sensitive to outliers, and more, it does not produce sparse solutions. The  $\varepsilon$ -Huber cost [10] contains the preceding ones as particular cases, and it is expressed as

$$\ell_P(e_n) = \begin{cases} 0, & |e_n| \leq \varepsilon \\ \frac{1}{2\gamma}(|e_n| - \varepsilon)^2, & \varepsilon < |e_n| \leq e_c \\ C(|e_n| - \varepsilon) - \frac{1}{2}\gamma C^2, & |e_n| > e_c \end{cases} \quad (2)$$

where  $e_c = \varepsilon + \gamma C$ . Parameter  $\gamma$  controls the width of the  $L_2$  interval between  $\varepsilon$  and  $e_c$ , so that the function is continuous and derivable, and the  $L_1$  interval has slope  $C$ . The  $\varepsilon$ -insensitivity zone provides with sparse solutions in SVM formulation, which is a very desirable characteristic in nonlinear formulations. The quadratic cost is optimal, in a Maximum Likelihood (ML) sense, when the noise is Gaussian, whereas the linear cost is optimal for exponential noise. Here, we propose to use the  $\varepsilon$ -Huber CFR because it has the ability to deal simultaneously with different kinds of noise [10]. The use of  $\varepsilon$ -insensitive CFR is not appropriate when Gaussian noise can be present in the data, whereas a quadratic CFR (according to LS-SVM) does not produce sparse solutions.

### C. Algorithm statement for SVR system identification

The algorithm for SVR system identification using the proposed  $\varepsilon$ -Huber cost reduces to the minimization of

$$L_P^{SVR}(v_j, \xi_n^{(*)}) = \frac{1}{2} \sum_{j=1}^{H_z} v_j^2 + \frac{1}{2\gamma} \sum_{n \in I_1} (\xi_n^2 + \xi_n^{*2}) + C \sum_{n \in I_2} (\xi_n + \xi_n^*) - \sum_{n \in I_2} \frac{\gamma C^2}{2}, \quad (3)$$

where  $\xi_n, \xi_n^*$  are the slack variables or losses,  $I_1$  is the set of samples for which  $\varepsilon \leq \xi_n^{(*)} \leq e_c$ ,  $I_2$  is the set of samples for which  $\xi_n^{(*)} > e_c$ , and constrained to

$$y_n - \mathbf{v}^T \phi_z(\mathbf{z}_n) \leq \varepsilon + \xi_n, \quad \forall n = n_0, \dots, N, \quad (4)$$

$$-y_n + \mathbf{v}^T \phi(\mathbf{z}_n) \leq \varepsilon + \xi_n^*, \quad \forall n = n_0, \dots, N \quad (5)$$

where  $\xi_n^{(*)} \geq 0$ ,  $n_0$  is given by the required initial conditions (without loss of generality,  $n_0 = 1$  and null initial conditions),  $N$  is the number of available samples, and  $\xi_n^{(*)}$  denotes both  $\xi_n$  and  $\xi_n^*$ .

The Lagrangian for this problem is obtained by introducing a coefficient (Lagrange multiplier) for each constraint [13]. In particular,  $\alpha_n$  and  $\alpha_n^*$  are the (non-negative) Lagrange multipliers corresponding to (4) and (5), respectively. By making zero the gradient of the Lagrangian, with respect to  $v_j$  and  $\xi_n^{(*)}$ , we obtain

$$\mathbf{v} = \sum_{n=1}^N (\alpha_n - \alpha_n^*) \phi_z(\mathbf{z}_n) = \sum_{n=1}^N \beta_n \phi_z(\mathbf{z}_n) \quad (6)$$

and  $0 \leq \alpha_n^{(*)} \leq C$ , where  $\beta_n = \alpha_n - \alpha_n^*$ . After introducing these conditions into the Lagrangian, the primal variables are removed, and a term-grouping can be done by writing down the Gram matrix of dot products in the RKHS, or kernel matrix,

$$\mathbf{G}_{ij} = \langle \phi_z(\mathbf{z}_i), \phi_z(\mathbf{z}_j) \rangle = K_z(\mathbf{z}_i, \mathbf{z}_j). \quad (7)$$

The dual problem is to maximize with constrains

$$L_D^{SVR}(\alpha_n^{(*)}) = -\frac{1}{2} (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)^T [\mathbf{G} + \gamma \mathbf{I}] (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)^T \mathbf{y} - \varepsilon \mathbf{1}^T (\boldsymbol{\alpha} + \boldsymbol{\alpha}^*) \quad (8)$$

where  $\boldsymbol{\alpha}^{(*)} = [\alpha_1^{(*)}, \dots, \alpha_N^{(*)}]^T$  and  $\mathbf{y} = [y_1, \dots, y_N]^T$ . This is a constrained quadratic programming (QP) problem that has a single minimum. It can be shown that the predicted output for a new observed sample  $y_r$ , given  $\mathbf{z}_r$ , is

$$\hat{y}_r = \sum_{n=1}^N \beta_n K_z(\mathbf{z}_n, \mathbf{z}_r) \quad (9)$$

This solution is expressed in terms of the observation vectors, and hence, if sparsity is allowed in the  $\varepsilon$ -Huber CFR by making  $\varepsilon > 0$ , only some of the Lagrange multipliers are nonzero. Those samples with a nonzero coefficient are called *support vectors*, and they contain all the information that is relevant for building the model.

*Property 1:* The following nonlinear relationship between the residuals and the model coefficients holds:

$$\beta_n = g_{nl}(e_n) = \begin{cases} \text{sign}(e_n)C, & |e_n| \geq e_c \\ \frac{\text{sign}(e_n)}{\gamma} (|e_n| - \varepsilon), & \varepsilon \leq |e_n| \leq e_c \\ 0, & |e_n| < \varepsilon \end{cases} \quad (10)$$

*Proof.* When using the  $\varepsilon$ -Huber CFR in SVM regression-like problems, a straightforward relationship between the residuals and the Lagrange multipliers can be derived from the Karush-Khun-Tucker (KKT) conditions [10], [11]. We have that  $\alpha_n = C$  for  $e_n \geq e_c$ , that  $\alpha_n = \frac{1}{\gamma}(e_n - \varepsilon)$  for  $\varepsilon \leq e_n \leq e_c$ , and that  $\alpha_n = 0$  for  $e_n < \varepsilon$ . Also, we have that  $\alpha_n^* = C$  for  $e_n \leq -e_c$ , that  $\alpha_n^* = \frac{1}{\gamma}(-e_n - \varepsilon)$  for  $-\varepsilon \geq e_n \geq -e_c$ , and that  $\alpha_n^* = 0$  for  $e_n > -\varepsilon$ . Given that  $\beta_n = \alpha_n - \alpha_n^*$ , then (10) holds.  $\square$

According to (9), nonlinear relationship (10) can be conveniently used to control the impact of an outlier on the final solution by choosing appropriate values of the cost function parameters [11], i.e., an outlier will have, at most, a weight  $|\beta_n| = C$ . However,  $\beta_n$  for  $\varepsilon < |e_n| < e_C$  can be more flexibly valued than in  $\varepsilon$ -insensitive CFR. Free parameters of both the CFR and the Mercer's kernel are usually determined in SVM algorithms by using cross-validation search.

### III. EXPLICIT ARMA MODELS IN RKHS

The formulation presented in the section before uses a time series model given by a nonlinear regression in a RKHS, whose input space is given by the concatenated vector  $\mathbf{z}_n$ . Therefore, it can only be considered an ARMA model in a wide and implicit sense, given that both the AR and the MA component of the observed DTP are stacked and jointly transformed into that RKHS. Although this can be a valid and powerful approach, no useful insight about the temporal statistical properties of the data can be gained. This is a similar situation to the NN-based analysis of time series, where the temporal structure remains inside a black-box equation [2]. Alternatively, we propose here to build an explicit ARMA model in some given RKHS, by taking advantage of Mercer's kernels, which will allow us to study the time series structure of the data, even if this is a nonlinear model for system identification.

Assume that both the input and the output DTP state vectors can be separately mapped to  $\mathcal{H}_u, \mathcal{H}_y$ , by using two possibly different nonlinear mappings,  $\phi_u(\mathbf{u}_n) : \mathbb{R}^Q \rightarrow \mathcal{H}_u$ , and  $\phi_y(\mathbf{y}_n) : \mathbb{R}^P \rightarrow \mathcal{H}_y$ , respectively. A linear MA (AR) model component can be built in  $\mathcal{H}_u$  ( $\mathcal{H}_y$ ), and now, the ARMA difference equation is:

$$y_n = \mathbf{a}^T \phi_y(\mathbf{y}_{n-1}) + \mathbf{b}^T \phi_u(\mathbf{u}_n) + e_n \quad (11)$$

where  $\mathbf{b} = [b_1, \dots, b_{H_u}]^T$  and  $\mathbf{a} = [a_1, \dots, a_{H_y}]^T$  are vectors determining the MA and the AR coefficients of the system, respectively, in the RKHS; and  $H_u, H_y$  are the RKHS dimensions.

The primal problem can be here formulated as the minimization of

$$L_P^{2k}(a_i, b_j, \xi_n^{(*)}) = \frac{1}{2} \sum_{i=1}^{H_y} a_i^2 + \sum_{j=1}^{H_u} b_j^2 + \frac{1}{2\gamma} \sum_{n \in I_1} (\xi_n^2 + \xi_n^{*2}) + C \sum_{n \in I_2} (\xi_n + \xi_n^*) - \sum_{n \in I_2} \frac{\gamma C^2}{2} \quad (12)$$

constrained to

$$y_n - \mathbf{a}^T \phi_y(\mathbf{y}_{n-1}) - \mathbf{b}^T \phi_u(\mathbf{u}_n) \leq \varepsilon + \xi_n \quad (13)$$

$$-y_n + \mathbf{a}^T \phi_y(\mathbf{y}_{n-1}) + \mathbf{b}^T \phi_u(\mathbf{u}_n) \leq \varepsilon + \xi_n^* \quad (14)$$

and  $\xi_n^{(*)} \geq 0$ , for  $n = 1, \dots, N$ . By stating the Lagrangian and making its gradient zero, the AR and MA vector coefficients are given by

$$\mathbf{a} = \sum_{n=1}^N \beta_n \phi_y(\mathbf{y}_{n-1}), \quad \mathbf{b} = \sum_{n=1}^N \beta_n \phi_u(\mathbf{u}_n) \quad (15)$$

which is a different expression for the model coefficients in (6), because AR and MA coefficients are now uncoupled in the RKHS.

After introducing (15) into the Lagrangian, we can identify two different Gram matrices, one for the input and another for the output DTP, denoted as

$$\mathbf{R}_{y,ij} = \langle \phi_y(\mathbf{y}_{i-1}), \phi_y(\mathbf{y}_{j-1}) \rangle = K_y(\mathbf{y}_{i-1}, \mathbf{y}_{j-1}) \quad (16)$$

$$\mathbf{R}_{u,ij} = \langle \phi_u(\mathbf{u}_i), \phi_u(\mathbf{u}_j) \rangle = K_u(\mathbf{u}_i, \mathbf{u}_j) \quad (17)$$

Equations (16) and (17) can be seen as uncoupled Gram matrices that also account for the sample estimators of input and output DTP autocorrelation functions [14], respectively, in the RKHS. The dual problem consists now in

the constrained maximization of

$$L_D^{2k}(\alpha_n^{(*)}) = -\frac{1}{2}(\alpha - \alpha^*)^T [\mathbf{R}_u + \mathbf{R}_y + \gamma \mathbf{I}] (\alpha - \alpha^*) + (\alpha - \alpha^*)^T \mathbf{y} - \varepsilon \mathbf{1}^T (\alpha + \alpha^*) \quad (18)$$

The output for a new observation vector is obtained as

$$\hat{y}_r = \sum_{n=1}^N \beta_n (K_y(\mathbf{y}_{n-1}, \mathbf{y}_{r-1}) + K_u(\mathbf{u}_n, \mathbf{u}_r)) \quad (19)$$

Note that the model complexity, in terms of number of coefficients, is, as in (9), equal to the number of training samples  $N$ . We will denote this algorithm as SVM-ARMA $_{2k}$ .

To gain further insight about the structure that SVM-ARMA $_{2k}$  has, we can analyze the temporal structure of the proposed model. According to [1], there are two main general classes of system identification algorithms: Prediction Error Methods (PEM), which are based on the minimization of a function of the residual variance for a given model (e.g. least-squares and/or maximum a posteriori estimators); and Correlation Methods (CM), which minimize the cross-correlation between a (possibly nonlinear) function of the residuals and some transformation of the data. In [10], a comparison of linear SVM-ARMA system identification with PEM and CM was presented. The SVM-ARMA $_{2k}$  nonlinear system identification model solves the regularized normal equations in some RKHS while minimizing the cross-correlation between the data and a nonlinear function of the residuals, as shown in Appendices I and II. This nonlinear relationship is determined by the free parameters of the  $\varepsilon$ -Huber CFR.

#### IV. SVM SYSTEM IDENTIFICATION WITH COMPOSITE KERNELS

In the two preceding sections, we have described the SVR-based and the SVM-ARMA $_{2k}$  system identification algorithms. Two questions can be raised at this moment. First, note that (19) in SVM-ARMA $_{2k}$  shows an apparent uncoupling between the input and the output DTP in the final solution, with no explicit consideration of the (maybe relevant) cross information between them. Although we are solving the normal equations in the RKHS, and the cross correlation is indeed implicitly considered therein, the SVM-ARMA $_{2k}$  model could be somewhat limited in the cases when strong cross information were present. Therefore, we will look for a SVM-ARMA system identification model capable of considering a cross comparison between input and output DTP states, when this becomes necessary in the problem at hand. This new algorithm will be called SVM-ARMA $_{4k}$ . Second, if we observe prediction equations (9) and (19), we can think of the possibility of combining them into a joint model for improving its performance and flexibility simultaneously. These two additional algorithms are called SVR-ARMA $_{2k}$  and SVR-ARMA $_{4k}$ .

In this section, we firstly describe the elements of a generic nonlinear mapping into a RKHS in a SVM system identification problem. Then, we use composite kernels as direct sum of different RKHS (a well-known result of Functional Analysis Theory, see e.g. [15]), which allows us both to represent the previously described SVM models, and to formulate the above mentioned new system identification algorithms.

### A. Generic SVM algorithm for system identification

*Property 2:* Let  $\phi(\mathbf{z}_n)$  be a composite nonlinear transformation (into a RKHS  $\mathcal{H}$ ) given by the concatenation of  $M$  single nonlinear transformations to their RKHS, i.e.,

$$\phi(\mathbf{z}_n) = [\phi_1(\mathbf{z}_n)^T, \phi_2(\mathbf{z}_n)^T, \dots, \phi_M(\mathbf{z}_n)^T]^T \quad (20)$$

The corresponding SVM system identification model is given by  $y_n = \langle \mathbf{w}, \phi(\mathbf{z}_n) \rangle + e_n$ , where  $\mathbf{w} \in \mathcal{H}$ . The kernel matrix is  $\mathbf{K}_{ij} = \langle \phi(\mathbf{z}_i), \phi(\mathbf{z}_j) \rangle$ , and it allows to state a dual QP problem that yields the model coefficients. The dual problem is to maximize with constrains

$$L_D(\alpha_n^{(*)}) = -\frac{1}{2} (\alpha - \alpha^*)^T [\mathbf{K} + \gamma \mathbf{I}] (\alpha - \alpha^*) + (\alpha - \alpha^*)^T \mathbf{y} - \varepsilon \mathbf{1}^T (\alpha + \alpha^*) \quad (21)$$

and the prediction model is  $\hat{y}_r = \sum_{n=1}^N \beta_n K(\mathbf{z}_n, \mathbf{z}_r)$ .

*Proof.* The derivation is similar to the ones presented in Sections II and III.  $\square$

*Property 3: SVR from Composite Kernels.* It is straightforward to see that the SVR system identification model is obtained for  $\phi(\mathbf{z}_n) = \phi_z(\mathbf{z}_n)$ . In this case, the kernel matrix is (7). Given that prediction model is (9), where a single kernel is used, the explicit consideration of input and output DTP is lost, and the normal equations are generated in a RKHS where the input and output effects are coupled.

*Property 4: SVM-ARMA<sub>2k</sub> from Composite Kernels.* The above proposed SVM-ARMA<sub>2k</sub> algorithm can be obtained from a composite kernel formulation by using  $\phi(\mathbf{z}_n) = [\phi_y(\mathbf{y}_{n-1})^T, \phi_u(\mathbf{u}_n)^T]^T$ . It is straightforward to see that the model kernel is:

$$K(\mathbf{z}_i, \mathbf{z}_j) = \langle [\phi_y(\mathbf{y}_{i-1})^T, \phi_u(\mathbf{u}_i)^T]^T, [\phi_y(\mathbf{y}_{j-1})^T, \phi_u(\mathbf{u}_j)^T]^T \rangle = K_y(\mathbf{y}_{i-1}, \mathbf{y}_{j-1}) + K_u(\mathbf{u}_i, \mathbf{u}_j) \quad (22)$$

where the kernel (being the sum of two kernels) accounts for the input and output DTP.

### B. Composite kernels for input-output cross information

Composite kernels can be used to emphasize, if necessary, the cross information between input and output DTP. Assume a nonlinear mapping  $\varphi(\cdot)$  into a RKHS  $\mathcal{H}_\varphi$  and three linear transformations  $\mathbf{A}_i$  from  $\mathcal{H}_\varphi$  to  $\mathcal{H}_i$ ,  $i = 1, 2, 3$ . Note, however, that in this case,  $\mathbf{u}_n$  and  $\mathbf{y}_n$  need to have the same dimension for the formulation to be valid. For simplicity, we force  $P' = Q' = \max(P, Q)$ , which yields input and output vectors  $\mathbf{u}'_n$  and  $\mathbf{y}'_{n-1}$  that are ensured to contain all the relevant time series information (plus some amount of redundant information).

For the following composite transformation

$$\phi(\mathbf{z}') = [\mathbf{A}_1 \varphi(\mathbf{u}')^T, \mathbf{A}_2 \varphi(\mathbf{y}')^T, \mathbf{A}_3 (\varphi(\mathbf{u}') + \varphi(\mathbf{y}'))^T]^T, \quad (23)$$

the obtained kernel is

$$K(\mathbf{z}'_i, \mathbf{z}'_j) = \varphi^T(\mathbf{y}'_i) \mathbf{R}_1 \varphi(\mathbf{y}'_j) + \varphi^T(\mathbf{u}'_i) \mathbf{R}_2 \varphi(\mathbf{u}'_j) + \varphi^T(\mathbf{y}'_{i-1}) \mathbf{R}_3 \varphi(\mathbf{u}'_j) + \varphi^T(\mathbf{u}'_i) \mathbf{R}_3 \varphi(\mathbf{y}'_{j-1}), \quad (24)$$

where  $\mathbf{R}_1 = \mathbf{A}_1^T \mathbf{A}_1 + \mathbf{A}_3^T \mathbf{A}_3$ ,  $\mathbf{R}_2 = \mathbf{A}_2^T \mathbf{A}_2 + \mathbf{A}_3^T \mathbf{A}_3$  and  $\mathbf{R}_3 = \mathbf{A}_3^T \mathbf{A}_3$  are three (independent) definite positive matrices. The last two terms can be grouped into a Mercer's kernel  $K_{uy}$  accounting for cross information,

$$K_{uy}(\mathbf{u}'_i, \mathbf{y}'_{j-1}) = K_3(\mathbf{y}'_{i-1}, \mathbf{u}'_j) + K_3(\mathbf{u}'_i, \mathbf{y}'_{j-1}) \quad (25)$$

which is warranted to be a valid Mercer kernel if  $K_3$  is a valid Mercer kernel. Then, the final kernel expression is

$$K(\mathbf{z}'_i, \mathbf{z}'_j) = K_y(\mathbf{y}_{i-1}, \mathbf{y}_{j-1}) + K_u(\mathbf{u}_i, \mathbf{u}_j) + K_{uy}(\mathbf{u}'_i, \mathbf{y}'_{j-1}) \quad (26)$$

and using this kernel in the generic SVM system identification algorithm gives us the so-called SVM-ARMA $_{4k}$  algorithm. It can be seen that we are now using four different kernels to build the composite kernel. A cross information analysis of this algorithm can be made, according to the corresponding normal equations in the RKHS for this case (not included here).

### C. Composite kernels for improved versatility

Instead of using separately the proposed algorithms for SVM system identification, one can think on using in a collaborative way the different kernel structures that have been presented here.

*Property 5:* The first possibility is using three concatenated transformations into RKHS, one for input  $\mathbf{u}_n$ , one for output  $\mathbf{y}_{n-1}$ , and one for their concatenation  $\mathbf{z}_n$ . The result is a combination between the SVM-ARMA and the SVR structures described in the preceding sections. The transforming concatenation is

$$\phi(\mathbf{z}_n) = [\phi_y(\mathbf{y}_{n-1})^T, \phi_u(\mathbf{u}_n)^T, \phi_z(\mathbf{z}_n)^T]^T \quad (27)$$

It is straightforward to see that the corresponding kernel is

$$K(\mathbf{z}_i, \mathbf{z}_j) = K_y(\mathbf{y}_{i-1}, \mathbf{y}_{j-1}) + K_u(\mathbf{u}_i, \mathbf{u}_j) + K_z(\mathbf{z}_i, \mathbf{z}_j) \quad (28)$$

and its introduction in the generic SVM system identification model yields the so-called SVR-ARMA $_{2k}$  algorithm.

*Property 6:* A composite transformation given by

$$\phi(\mathbf{z}') = [\mathbf{A}_1\varphi(\mathbf{u}')^T, \mathbf{A}_2\varphi(\mathbf{y}')^T, \mathbf{A}_3(\varphi(\mathbf{u}') + \varphi(\mathbf{y}'))^T, \phi_z(\mathbf{z})^T]^T \quad (29)$$

gives the following composite kernel:

$$K(\mathbf{z}'_i, \mathbf{z}'_j) = K_y(\mathbf{y}_{i-1}, \mathbf{y}_{j-1}) + K_u(\mathbf{u}_i, \mathbf{u}_j) + K_{uy}(\mathbf{u}'_i, \mathbf{y}'_{j-1}) + K_z(\mathbf{z}_i, \mathbf{z}_j), \quad (30)$$

which produces the SVR-ARMA $_{4k}$  algorithm.

Note that SVR-ARMA $_{2k}$  and SVR-ARMA $_{4k}$  have not a straightforward interpretation in terms of normal equations in the RKHS, but rather they can contain all the relevant model information that can be extracted from the data by each component kernel. Therefore, despite that SVM-ARMA and SVR nonlinear system identification are different problem statements, both underlying models can be combined and embedded into a more general SVM signal processing framework for non-linear system identification.

In conclusion, we can say that composite kernels can be used to provide us with model flexibility in terms of: (1) emphasized consideration, if necessary, of the input-output cross information; and (2) straightforward and natural combinations of implicit and explicit ARMA models.

TABLE I

SYSTEM IDENTIFICATION SIMULATION RESULTS. BOLD AND ITALICS INDICATE THE TWO BEST MODELS FOR EACH MERIT FIGURE.

	Eq.	ME	MSE	MAE	$r$	nMSE
<b>SVR</b>	(9)	0.29	3.08	1.08	0.991	-0.82
<b>SVM-ARMA<sub>2K</sub></b>	(19)	0.14	<b>2.08</b>	<b>0.88</b>	<b>0.992</b>	<b>-0.92</b>
<i>SVR-ARMA<sub>2K</sub></i>	(26)	<i>0.13</i>	<i>2.13</i>	<i>0.90</i>	<i>0.991</i>	<i>-0.88</i>
<b>SVM-ARMA<sub>4K</sub></b>	(26)	0.16	2.50	0.95	0.991	-0.85
<b>SVR-ARMA<sub>4K</sub></b>	(30)	<b>0.03</b>	3.76	1.11	0.976	-0.77

## V. EXPERIMENTAL RESULTS

In this section, we compare the performance of the SVR and the SVM-ARMA formulations in the previous sections. In all kernel computations, we used the Gaussian kernel, which provides universal non-linear mapping capabilities and computational convenience [16], [17]. Different types of kernels (linear, polynomial, etc.) could be considered for the input, output or cross-information kernels, according to *a priori* knowledge of the system.

**Example 1. Non-linear feedback system.** We first consider the following system. The input DTP is generated with Lorenz equations, given by  $du/dt = -\rho u + \rho y$ ,  $dy/dt = -uz + ru - y$ , and  $dz/dt = uy - bz$ , and using  $\rho = 10$ ,  $r = 28$ , and  $b = 8/3$ . Only the  $u$  component is used as input signal to the system, and it goes forward through an 8th-order low-pass FIR filter,  $H(z)$ , with cutoff frequency  $\omega_n = 0.5$  and normalized gain of -6dB at  $\omega_n$ . The output signal goes through a feedback loop consisting of a high-pass minimum-phase channel,  $G(z) = (1.00 + 2.01z^{-1} + 1.46z^{-2} + 0.39z^{-3})^{-1}$ , and then distorted with non-linearity  $f(\cdot) = \log(\cdot)$ .

This system was used to generate 10,000 input-output DTP samples, that were split into a training set (50) and a test set (following 500). The experiment was repeated 100 times with randomly selected starting points in the DTP. Free parameters were selected through 8-fold cross-validation using the training set, and average results for the test set are shown in Table I for mean error (ME), mean-squared error (MSE), mean absolute error (MAE), correlation coefficient ( $r$ ), and the normalized MSE ( $\text{nMSE} = \log_{10}(\sqrt{\text{MSE}/\text{var}(y)})$ ) of models in the test set. It is worth noting that the SVM-ARMA<sub>2K</sub> is the best model for this example, and that also SVR-ARMA<sub>2K</sub> and SVM-ARMA<sub>4K</sub> outperform the SVR. However, the composite SVR-ARMA<sub>4K</sub> turns to be here a worse model specification. SVR-ARMA<sub>4K</sub> is a more complex model, which can be appropriate for much more complex dynamics. Otherwise, the complexity of the model may degrade the generalization performance.

**Example 2. The Mackey-Glass time series.** We also compared the performance of SVM models in the standard Mackey-Glass time series prediction problem following the same approach as in [19], where the use of the standard SVR was originally presented for time series prediction. This classical high-dimensional chaotic system is generated by the following delay differential equation:

$$\frac{dx(t)}{dt} = -0.1x(t) + \frac{0.2x(t - t_\Delta)}{1 + x(t - t_\Delta)^{10}}, \quad (31)$$

TABLE II  
 nMSE IN VALIDATION SET FOR THE SVM MODELS AND METHODS IN [18].

	Poly	Rat	Loc $d = 1$	Loc $d = 2$	RBF	MLP	SVR	SVM -ARMA <sub>2K</sub>	SVR -ARMA <sub>2K</sub>	SVM -ARMA <sub>4K</sub>	SVR -ARMA <sub>4K</sub>
<b>MG17</b>	-1.95	-1.14	-1.48	-1.89	-1.97	-2.00	-2.36	-2.83	-2.87	<b>-2.88</b>	-2.86
<b>MG30</b>	-1.40	-1.33	-1.24	-1.42	-1.60	-1.5	<b>-1.87</b>	-1.73	-1.76	-1.73	-1.76

with delays  $t_{\Delta}=17$  and 30, thus yielding the time series MG17 and MG30, respectively. For comparison with [19], we considered 500 training samples and used the next 1000 for free parameter selection (validation set). This procedure also allows us direct comparison with previous results in the literature [18], [20], [21], in terms of nMSE.

Results are shown in Table II. The SVR algorithm outperformed the preceding methods for both time series. The methods proposed here widely outperform SVR in MG17; note that a difference between SVR and SVM-ARMA<sub>4K</sub> in nMSE is equivalent to almost one order of magnitude in MSE. Nevertheless, they did not outperform SVR in MG30, which could be due to differences in the kernel choice (RBF instead of trigonometric kernel [19]) or the considered embedding.

## VI. CONCLUSIONS

This paper presented the explicit formulation of nonlinear SVM-ARMA models in RKHS, which makes possible to scrutinize the statistical properties and the time series structure in system identification problems. In addition, a full family of methods for non-linear system identification have been proposed, by taking advantage of composite kernels, in which dedicated mappings are used for input, output and cross terms. Simulation results illustrated the potential capabilities of this framework, as demonstrated in the field of image processing [22] recently. This framework also allows a successful integration and combination of non-linear SVR and SVM-ARMA models.

## APPENDIX

### I. NONLINEAR SVM-ARMA AND PREDICTION ERROR METHODS

*Property 7:* Let us denote *quadratic cost conditions* (QCC) as  $\varepsilon = 0$  and  $C = \infty$  in problem (12). Then, for QCC we have that model weights in (19) are proportional to the residuals, this is,  $\beta_n = \frac{1}{\gamma}e_n$ . The proof is immediate by making  $\varepsilon = 0$  and  $C = \infty$  in (10).

*Property 8:* For QCC, the linear SVM-ARMA model can be viewed as a regularization of the normal equations of the Wiener filter for system identification. See [10] for proof.

**Theorem 1:** For QCC, nonlinear SVM-ARMA<sub>2k</sub> algorithm solves the regularized normal equations in the RKHS.

*Proof.* Let  $\phi_u(\mathbf{x}_i) \in \mathcal{H}_u$ , and  $\phi_y(\mathbf{y}_i) \in \mathcal{H}_y$ , vectors of size  $(H_u \times 1)$  and  $(H_y \times 1)$ , respectively, for  $i = 1, \dots, N$ . Let the data matrix in each RKHS be given by  $\Phi_u = [\phi_u(\mathbf{u}_1), \dots, \phi_u(\mathbf{u}_N)]$  and  $\Phi_y = [\phi_y(\mathbf{y}_1), \dots, \phi_y(\mathbf{y}_N)]$ .

Then, the matrix-form equation of the model for the observed data is  $\mathbf{y} = \Phi_u^T \mathbf{b} + \Phi_y^T \mathbf{a} + \mathbf{e}$ . Dual problem (18) can now be expressed (in matrix form) as the maximization of:

$$L_{QCC}^{2k}(\mathbf{e}) = \frac{1}{2\gamma^2} \mathbf{e}^T \left( \Phi_y^T \Phi_y + \Phi_u^T \Phi_u \right) \mathbf{e} - \frac{1}{\gamma} \mathbf{y}^T \mathbf{e} + \frac{1}{2\gamma} \mathbf{e}^T \mathbf{e} \quad (32)$$

By making zero the gradient of  $L_{QCC}^{2k}$ , and after some manipulations, the following expression is obtained:

$$\Phi_y^T \Phi_y \mathbf{y} - \Phi_y^T \Phi_y \Phi_u^T \mathbf{b} - \Phi_y^T \Phi_y \Phi_y^T \mathbf{a} + \Phi_u \Phi_u^T \mathbf{y} - \Phi_u^T \Phi_u \Phi_u^T \mathbf{b} - \Phi_u^T \Phi_u \Phi_y^T \mathbf{a} - \gamma \Phi_u^T \mathbf{b} - \gamma \Phi_y^T \mathbf{a} = 0 \quad (33)$$

By denoting  $\Phi_y \Phi_y^T = \mathbf{R}_{yy}$ ,  $\Phi_u \Phi_u^T = \mathbf{R}_{uu}$ ,  $\Phi_u \Phi_y^T = \mathbf{R}_{uy}$ ,  $\Phi_y \Phi_u^T = \mathbf{R}_{yu}$ , and

$$\mathbf{r}_{zy}^{2k} = \begin{bmatrix} \Phi_y \mathbf{y} \\ \Phi_u \mathbf{y} \end{bmatrix}, \quad \mathbf{R}^{2k} = \begin{bmatrix} \mathbf{R}_{yy} & \mathbf{R}_{yu} \\ \mathbf{R}_{uy} & \mathbf{R}_{uu} \end{bmatrix} \quad (34)$$

the following term grouping can be done:

$$\begin{bmatrix} \Phi_y^T & \Phi_u^T \end{bmatrix} \mathbf{r}_{zy}^{2k} = \begin{bmatrix} \Phi_y^T & \Phi_u^T \end{bmatrix} \left( \mathbf{R}^{2k} + \gamma \mathbf{I} \right) \begin{bmatrix} \mathbf{a}^T & \mathbf{b}^T \end{bmatrix}^T \quad (35)$$

which holds if and only if  $\mathbf{r}_{zy}^{2k} = \left( \mathbf{R}^{2k} + \gamma \mathbf{I} \right) \begin{bmatrix} \mathbf{a}^T & \mathbf{b}^T \end{bmatrix}^T$  that are the regularized Wiener equations in RKHS.  $\square$

Therefore, the SVM-ARMA<sub>2k</sub> formulation for nonlinear kernels and QCC leads naturally to the statement of the equations in the joint RKHS. Note that for some Mercer's kernels, the dimension of these equations can be infinite; however, as far as we are not solving them explicitly, but rather implicitly in (18) and by means of the kernel trick, we will still be able to scrutinize the statistical properties of the time series problem, specially if these properties can be conveniently expressed with dot products in the RKHS, and subsequently analyzed using Mercer's kernels.

## II. NONLINEAR SVM-ARMA AND CORRELATION METHODS

CM for system identification are based on the assumption that a good model produces residuals that are uncorrelated with past data, and consequently, these methods minimize the cross correlation between a function of the residuals and a transformation of the data, both of them being possibly nonlinear. Different approaches in the literature [1] are based on different methods for determining suitable residual functions and data transformations.

*Property 9:* For QCC, the nonlinear SVM-ARMA<sub>2k</sub> system identification model minimizes the cross-correlation between the data transformed to RKHS and the residuals.

*Proof.* Taking into account (15) and Property 2, we have  $\mathbf{a} = \Phi_y \mathbf{e}$  and  $\mathbf{b} = \Phi_u \mathbf{e}$ , and according to the primal problem statement (12), we are minimizing the  $L_2$  norm of the coefficients, even though they are in the RKHS and they can be not explicitly known (see [10] for details on the similar property for linear SVM-ARMA).

*Property 10:* Under the set of nonzero and finite possible values for the free parameters of the  $\varepsilon$ -Huber cost function ( $0 < \varepsilon, C, \gamma < \infty$ ), the nonlinear SVM-ARMA<sub>2k</sub> system identification algorithm minimizes the correlation between the data and a nonlinear transformation of the residuals.

*Proof.* For each given fixed subset of the free parameters  $0 < \varepsilon, C, \gamma < \infty$ , the nonlinear relationship between the model coefficients and the residuals is given by (10). According to (15) and (10), we have  $\mathbf{a} = \Phi_y g_{nl}(\mathbf{e})$  and  $\mathbf{b} = \Phi_u g_{nl}(\mathbf{e})$ , which stand for the model coefficients being the (uncoupled) cross correlation between the

nonlinearly transformed residuals and the data in the RKHS. Furthermore, we are minimizing the norm of these coefficients in (12).  $\square$

## REFERENCES

- [1] Lennart Ljung, *System Identification. Theory for the User*, Prentice Hall Inc., Englewood Cliffs, NJ, 2 edition, 1999.
- [2] O. Nelles, *Nonlinear System Identification. From classical approaches to Neural Networks and Fuzzy Models*, Springer-Verlag, Berlin Heidelberg New York, 2001.
- [3] V. Vapnik, *Statistical Learning Theory (Adaptive and Learning Systems for Signal Processing, Communications, and Control)*, John Wiley & Sons, 1998.
- [4] A. J. Smola and B. Schölkopf, "A tutorial on Support Vector Regression," Tech. Rep. NC-TR-98-030, Royal Holloway College, University of London, UK, 1998.
- [5] R.F. Drezet, P.M.L. Harrison, "Support vector machines for system identification," in *UKACC Intl. Conf. on Control '98*, Swansea, UK, Sept 1998, vol. 1, pp. 688–692.
- [6] A. Gretton, A. Doucet, R. Herbrich, P.J.W. Rayner, and B. Schölkopf, "Support Vector Regression for Black-Box System Identification," in *Proc. 11th IEEE Signal Processing Workshop on Statistical Signal Processing*, 2001, pp. 341–4.
- [7] J.A.K. Suykens, "Support Vector Machines : a nonlinear modelling and control perspective," *European Journal of Control, Special Issue on fundamental issues in control*, vol. 7, no. 2-3, pp. 311–327, Aug. 2001.
- [8] Ivan Goethals, Kristiaan Pelckmans, Johan A K Suykens, and Bart De Moor, "Subspace Identification of Hammerstein Systems Using Least Squares Support Vector Machines," *IEEE Trans. Automat. Control*, vol. 50, no. 10, pp. 1509–19, 2005.
- [9] Marcelo Espinoza, Johan A K Suykens, and Bart De Moor, "Kernel Based Partially Linear Models and Nonlinear Identification," *IEEE Trans Automat Control*, vol. 50, no. 10, pp. 1602–6, 2005.
- [10] J. L. Rojo-Álvarez, M. Martínez-Ramón, A. R. Figueiras-Vidal, M. dePrado Cumplido, and A. Artés-Rodríguez, "Support vector method for ARMA system identification," *IEEE Trans Sig Proc*, vol. 52, no. 1, pp. 155–64, 2004.
- [11] J L Rojo-Álvarez, G Camps-Valls, M Martínez-Ramón, E Soria-Olivas, A. Navia Vázquez, and A R Figueiras-Vidal, "Support vector machines framework for linear signal processing," *Sig Proc*, vol. 85, no. 12, pp. 2316–26, 2005.
- [12] Nello Cristianini and John Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge Univ. Press, Cambridge, UK, 2000.
- [13] D.G. Luenberger, *Linear and Nonlinear Programming*, Addison–Wesley Pub Co, Reading, MA, 1984.
- [14] A. Papoulis, *Probability Random Variables and Stochastic Processes*, McGraw-Hill, New York, USA, 3 edition, 1991.
- [15] M. C. Reed and B. Simon, *Functional Analysis*, vol. I of *Methods of Modern Mathematical Physics*, Academic Press, 1980.
- [16] B. Schölkopf and A. Smola, *Learning with Kernels. Support Vector Machines, Regularization, Optimization and Beyond*, MIT Press, 2002.
- [17] S. S. Keerthi and C.-J. Lin, "Asymptotic behaviors of support vector machines with Gaussian kernel," *Neural Computation*, vol. 15, no. 7, pp. 1667–1689, 2003.
- [18] M. Casdagli and S. Eubank, "Nonlinear modeling and forecasting," in *Proc. Santa Fe Institute Studies in the Science of Complexity*. November 1992, vol. XII, Addison–Wesley, Reading, MA.
- [19] S. Mukherjee, E. Osuna, and F. Girosi, "Nonlinear prediction of chaotic time series using a support vector machine," in *Neural Networks for Signal Processing VII — Proc. 1997 IEEE Workshop*, J. Principe, L. Gile, N. Morgan, and E. Wilson, Eds., New York, 1997, IEEE.
- [20] A. Lapedes and R. Farber, "Nonlinear signal processing using neural networks: prediction and system modeling," Tech. Rep., 1987.
- [21] J. D. Farmer and J. J. Sidorowich, "Predicting chaotic time series," *Phys. Rev. Lett.*, vol. 59, pp. 543, 1987.
- [22] G. Camps-Valls, L. Gomez-Chova, J. Muñoz-Marí, J. Vila-Francés, and J. Calpe-Maravilla, "Composite kernels for hyperspectral image classification," *IEEE Geosc. Remote Sensing Letters*, vol. 3, no. 1, pp. 93–97, Jan 2006.