# Nonlinear System Identification With Composite Relevance Vector Machines

Gustavo Camps-Valls, *Member, IEEE*, Manel Martínez-Ramón, *Senior Member, IEEE*, José Luis Rojo-Álvarez, *Member, IEEE*, and Jordi Muñoz-Marí

***Abstract*—Nonlinear system identification based on relevance vector machines (RVMs) has been traditionally addressed by stacking the input and/or output regressors and then performing standard RVM regression. This letter introduces a full family of composite kernels in order to integrate the input and output information in the mapping function efficiently and hence generalize the standard approach. An improved trade-off between accuracy and sparsity is obtained in several benchmark problems. Also, the RVM yields confidence intervals for the predictions, and it is less sensitive to free parameter selection.**

***Index Terms*—Composite kernels, nonlinear system identification, relevance vector machine (RVM).**

## I. Introduction

AUTO-REGRESSIVE and moving average (ARMA) digital filter structures are commonly used to build functional relationships between related discrete time processes (DTPs) when the function that relates both processes is linear and time invariant [1]. However, nonlinear behavior can be observed in many practical situations, and general nonlinear models, such as artificial neural networks or fuzzy algorithms, are alternatively used [2]. A powerful nonlinear technique for learning-from-samples problems is the support vector machine (SVM) [3], which was originally presented as an effective method for pattern classification [3]. The support vector regression (SVR) is the SVM implementation for regression and function approximation [4], and it has also been previously used for nonlinear system identification [5][6], but the time series structure of the data was not scrutinized. In [7], SVM was explicitly formulated for modeling linear time-invariant systems fulfilling and ARMA difference equations (linear SVM-ARX), and then it was extended to a general framework for linear signal processing problems [8] and for nonlinear SVM-based modeling [9].

Despite the good performance yielded by SVM schemes, some limitations still remain: 1) by assuming an explicit loss function (usually, the $\varepsilon$-insensitive loss function), one assumes a fixed distribution of the residuals; 2) several free parameters must be tuned, usually with cross-validation methods, which result in time-consuming tasks; and 3) very importantly, sparsity is not always achieved, and a high number of support vectors is eventually obtained.

Manuscript received June 7, 2006; revised August 17, 2006. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Dominic K. C. Ho.

G. Camps-Valls and J. Muñoz-Marí are with the Dep. d'Enginyeria Electrònica, Escola Tècnica Superior d'Enginyeria, Universitat de València, 46100 Burjassot (València), Spain (e-mail: gcamps@uv.es; jordi@uv.es).

M. Martínez-Ramón and J. L. Rojo-Álvarez are with the Dep. Teoría de la Señal y Comunicaciones, Univ. Carlos III de Madrid, Leganés, Madrid, Spain. (e-mail: manel@tsc.uc3m.es; jlrojo@tsc.uc3m.es).

Digital Object Identifier 10.1109/LSP.2006.885290

These problems of SVR and SVM are efficiently alleviated by the relevance vector machine (RVM), introduced by Tipping [10]. The RVM constitutes a Bayesian approximation for solving nonlinear models. The RVM follows a different inference principle from the one followed by SVM, and it has yielded good trade-off between accuracy and sparsity of the solution. In addition, RVM can produce probabilistic outputs (and hence they theoretically capture the uncertainty in the predictions), and they are less sensitive than SVM to setting of the free parameters, which is particularly interesting when working with multiple kernel machines as the number of free parameters increases. However, to the authors' knowledge, the use of RVM for nonlinear system identification and time series prediction is limited to a few works [11], and in all these cases, the approach consisted in stacking the input and output DTP into a training vector and then applying the traditional RVM formulation. This approach, although powerful, does not consider the input-output DTP relationships in the modeling, which may lead to suboptimal results.

In this letter, we introduce a general class of RVM-based system identification algorithms. Our proposal includes the use of *composite kernels* and shows that the previous stacked approach is a particular case. Several algorithms for nonlinear system identification are presented, which account for the input and output time processes either separately, jointly, or both, thus allowing different levels of flexibility and sophistication for model development. The proposed composite kernels have been presented in [9] for SVM-based modeling. Here, their suitability for the sparse Bayesian framework is analyzed from theoretical and simulation considerations.

## II. System Identification With the Relevance Vector Machine (RVM)

Assume a nonlinear system whose input and output are DTP $\{x_n\}$ and $\{y_n\}$. Let vectors $\boldsymbol{y}_{n-1} = [y_{n-1}, y_{n-2}, \ldots, y_{n-P}]^\top$ and $\boldsymbol{x}_n = [x_n, x_{n-1}, \ldots, x_{n-Q+1}]^\top$ denote the states of input and output DTP at time instant $n$. The nonlinear RVM-based system identification procedure is traditionally conducted by using the standard RVM regression algorithm with the concatenation of input and output states, $\boldsymbol{z}_n = [\boldsymbol{y}_{n-1}^\top, \boldsymbol{x}_n^\top]^\top$.

In the RVM formulation, given a *corpus* of training samples $\{\boldsymbol{z}_i\}_{i=1}^N \in \mathbb{R}^d$, with $d = P + Q - 1$, and given their corresponding output targets $\{y_i\}_{i=1}^N \in \mathbb{R}$, the outputs of an extended linear model are a linear combination of the response of a set of $M$ basis functions (or kernels), as follows:

$$\hat{y}_i = \sum_{j=1}^{M} w_j K(\boldsymbol{z}_i, \boldsymbol{z}_j) + w_o = \boldsymbol{\omega}^\top \mathbf{k}(\boldsymbol{z}_i) + w_o \qquad (1)$$

where $\boldsymbol{\omega} = [w_o, w_1, \ldots, w_M]^\top$ are the weights in the model, $w_o$ represents the bias in the regression function, $K(\boldsymbol{z}_i, \boldsymbol{z}_j)$ is the response of the $j$th basis function to input sample $\boldsymbol{z}_i$, and

$\mathbf{k}(\boldsymbol{z}_i) = [K(\boldsymbol{z}_i, \boldsymbol{z}_1), K(\boldsymbol{z}_i, \boldsymbol{z}_2), \ldots, K(\boldsymbol{z}_i, \boldsymbol{z}_M)]^\top$ are the kernel vectors. Let $\mathbf{K} \equiv \{K(\boldsymbol{z}_i, \boldsymbol{z}_j)\}$ be the $N \times (M+1)$ mapping (or kernel) matrix among all training samples. The obtained error (or residual) signal is expressed as $e_i = y_i - \hat{y}_i \sim \mathcal{N}(0, \sigma^2)$. By also assuming that the targets are independent, the likelihood of the target vector $\mathbf{y}$ can be written as

$$p(\mathbf{y} \,|\, \boldsymbol{\omega}, \sigma^2) = \prod_{i=1}^{N} p(y_i \,|\, \boldsymbol{\omega}, \sigma^2) = \frac{e^{-\|\mathbf{y} - \mathbf{K}\boldsymbol{\omega}\|^2/(2\sigma^2)}}{\sqrt{(2\pi\sigma^2)^N}}. \quad (2)$$

Once the basis functions of the model described in (1) are defined, a maximum likelihood approach could be used for estimating model weights $\boldsymbol{\omega}$. However, risk of overfitting arises, and *a priori* models on weight distribution are introduced in the Bayesian framework [12]. In the RVM learning scheme [13], a Gaussian *prior* distribution of zero mean and variance $\alpha_j \equiv 1/\sigma_{w_j}^2$ is defined over each weight

$$p(\boldsymbol{\omega} \,|\, \boldsymbol{\alpha}) = \prod_{j=1}^{M} \mathcal{N}(w_j \,|\, 0, \alpha_j^{-1}) = \prod_{j=1}^{M} \sqrt{\frac{\alpha_j}{2\pi}} \exp\left\{-\frac{1}{2}\alpha_j w_j^2\right\} \quad (3)$$

where the key to obtain sparsity is the use of $M$ independent hyper-parameters $\boldsymbol{\alpha} = [\alpha_o, \alpha_1, \ldots, \alpha_M]^\top$, one *per* weight (or basis function), which moderate the strength of the *prior*. After defining the *prior* over the weights, we must define the hyper-priors over $\boldsymbol{\alpha}$ and the noise variance $\sigma^2$.

Now, with *prior* (3) and likelihood distribution given by (2), the posterior distribution over the weights is Gaussian, and it can be computed by using Bayes' rule

$$p(\boldsymbol{\omega} \,|\, \mathbf{y}, \boldsymbol{\alpha}, \sigma^2) = \frac{p(\mathbf{y} \,|\, \boldsymbol{\omega}, \sigma^2) p(\boldsymbol{\omega} \,|\, \boldsymbol{\alpha})}{p(\mathbf{y} \,|\, \boldsymbol{\alpha}, \sigma^2)} \sim \mathcal{N}(\boldsymbol{\omega} \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad (4)$$

where the covariance and the mean are, respectively, given by

$$\boldsymbol{\Sigma} = (\sigma^{-2}\mathbf{K}^\top\mathbf{K} + \mathbf{A})^{-1} \quad \text{and} \quad \boldsymbol{\mu} = \sigma^{-2}\boldsymbol{\Sigma}\mathbf{K}^\top\mathbf{y} \quad (5)$$

with $\mathbf{A} = \mathrm{diag}(\boldsymbol{\alpha})$. Hence, the likelihood distribution over the training targets, given by (2), can be "marginalized" by integrating out the weights to obtain the *marginal likelihood* for the hyperparameters

$$p(\mathbf{y} \,|\, \boldsymbol{\alpha}, \sigma^2) = \int p(\mathbf{y} \,|\, \boldsymbol{\omega}, \sigma^2) p(\boldsymbol{\omega} \,|\, \boldsymbol{\alpha}) d\boldsymbol{\omega} \sim \mathcal{N}(0, \mathbf{C}) \quad (6)$$

where the covariance is given by $\mathbf{C} = \sigma^2\mathbf{I} + \mathbf{K}\mathbf{A}^{-1}\mathbf{K}^\top$. For computational efficiency, the logarithm of the evidence is maximized

$$\log p(\mathbf{y} \,|\, \boldsymbol{\alpha}, \sigma^2) = -\frac{1}{2}\left(M \log 2\pi + \log|\mathbf{C}| + \mathbf{y}^\top\mathbf{C}^{-1}\mathbf{y}\right) \quad (7)$$

which is commonly done using the standard *type-II maximum likelihood* (ML) procedure [12].

In the RVM learning scheme, the estimated value of the model weights is given by the mean of the posterior distribution in (4), which is also the *maximum a posteriori* (MAP) estimate of the weights. The MAP estimate of the weights depends on the value of the hyper-parameters $\boldsymbol{\alpha}$ and of the noise $\sigma^2$. The estimation of these two variables ($\hat{\boldsymbol{\alpha}}$ and $\hat{\sigma}^2$) is obtained by maximizing the marginal likelihood in (7). The uncertainty about the optimal value of the weights, given by (4), is used to express uncertainty about the predictions made by the model, i.e., given an input $\boldsymbol{z}_*$, the probability distribution of the corresponding output $y_*$ is given by the (Gaussian) predictive distribution

$$p(\mathbf{y}_* \,|\, \boldsymbol{z}_*, \hat{\boldsymbol{\alpha}}, \hat{\sigma}^2) = \int p(\mathbf{y}_* \,|\, \boldsymbol{z}_*, \boldsymbol{\omega}, \hat{\sigma}^2) p(\boldsymbol{\omega} \,|\, \mathbf{y}, \hat{\boldsymbol{\alpha}}, \hat{\sigma}^2) d\boldsymbol{\omega} \quad (8)$$

which has a Gaussian form, $p(\mathbf{y}_* \,|\, \boldsymbol{z}_*, \hat{\boldsymbol{\alpha}}, \hat{\sigma}^2) = \mathcal{N}(y_*, \sigma_*^2)$, where the mean and the variance (uncertainty) of the prediction are, respectively

$$y_* = \mathbf{k}^\top(\boldsymbol{z}_*)\boldsymbol{\mu} \quad \text{and} \quad \sigma_*^2 = \hat{\sigma}^2 + \mathbf{k}^\top(\boldsymbol{z}_*)\,\boldsymbol{\Sigma}\,\mathbf{k}(\boldsymbol{z}_*). \quad (9)$$

In the iterative maximization of marginal $\mathcal{L}(\boldsymbol{\alpha})$ in (7), many of hyper-parameters $\alpha_j$ tend to infinity, yielding *a posterior* distribution (4) of the corresponding weight $w_j$ that tends to be a delta function centered around zero. The corresponding weight is thus deleted from the model, along with its associated basis function. The remaining examples with nonzero associated weight are called the relevance vectors (RVs), and they resemble the support vectors in the SVM framework. However, RVM are typically more sparse than SVR.

## III. Composite Kernels for RVM System Identification

The RVM algorithm for nonlinear identification presented before does not represent an ARX model in a feature space, since the used data are the transformed concatenation of the input and output states. In this section, we exploit the *direct sum of Hilbert spaces* [19] to introduce a family of composite kernels in the RVM formulation that will allow us to analyze the explicit form of the ARX process in feature spaces. These composite kernels have demonstrated good capabilities in the context of SVM applied to speech processing [14] and image classification [15]. In [9], the SVM-based system identification problem with kernels was addressed. Here we extend it to the sparse Bayesian framework, in which some attractive additional properties are available, such as the improved sparsity of the solution and the confidence intervals provided for the predictions. The idea underlying this approach consists of splitting the information content of kernel matrix into AR and MA components. This approach not only will yield separate forms for the point predictions and uncertainties but also will allow different degrees of sophistication in kernel engineering and analysis. Also, the curse of dimensionality of stacking vectors is alleviated with our proposal.

### A. Explicit ARX in the Feature Spaces

Both the input and the output DTP state vectors can be separately mapped to $\mathcal{H}_x, \mathcal{H}_y$, by using two possibly different nonlinear mappings, $\mathbf{k}_x(\boldsymbol{x}_n) : \mathbb{R}^Q \to \mathcal{H}_x$ and $\mathbf{k}_y(\boldsymbol{y}_n) : \mathbb{R}^P \to \mathcal{H}_y$, respectively. Two linear models in $\mathcal{H}_x$ and $\mathcal{H}_y$ can be summed, yielding the difference equation

$$y_n = \boldsymbol{a}^\top\mathbf{k}_y(\boldsymbol{y}_{n-1}) + \boldsymbol{b}^\top\mathbf{k}_x(\boldsymbol{x}_n) + e_n \quad (10)$$

where $\boldsymbol{b} = [b_1, \ldots, b_{H_x}]^\top$ is the moving average (MA) component of the digital filter in the RKHS, which yields the eXogenous (X) component of the model, and $\boldsymbol{a} = [a_1, \ldots, a_{H_y}]^\top$ is the autorregressive (AR) component of the model in the mapped space, where $H_x$ and $H_y$ are the corresponding feature space dimensions. Note that defining the kernel vectors $\mathbf{k}_x(\boldsymbol{x}_n) = [K(\boldsymbol{x}_n, \boldsymbol{x}_1), K(\boldsymbol{x}_n, \boldsymbol{x}_2), \ldots, K(\boldsymbol{x}_n, \boldsymbol{x}_M)]^\top$ and $\mathbf{k}_y(\boldsymbol{y}_{n-1})$

$= [K(\boldsymbol{y}_{n-1}, \boldsymbol{y}_1), K(\boldsymbol{y}_{n-1}, \boldsymbol{y}_2), \ldots, K(\boldsymbol{y}_{n-1}, \boldsymbol{y}_M)]^\top$, an explicit mapping matrix can be built by adding the AR and X mapping matrices

$$\mathbf{K} = \mathbf{K}_{xx} + \mathbf{K}_{yy} \tag{11}$$

where $\mathbf{k}_{xx} \equiv \{K(\boldsymbol{x}_i, \boldsymbol{x}_j)\}$ and $\mathbf{k}_{yy} \equiv \{K(\boldsymbol{y}_{i-1}, \boldsymbol{y}_{j-1})\}$ are the $N \times (M+1)$ design matrices for the input and output DTP, respectively. This kernel matrix $\mathbf{K}$ can be included in the standard RVM formulation (9), leading to the so-called RVM$_{2K}$.

### B. Cross-Information Composite Kernel

Note that (11) produces 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. Therefore, the RVM$_{2k}$ model could be limited in some of the cases when strong cross information is present. In these cases, an ARX model considering the input and output components simultaneously could improve the results. By making use of the sum of Hilbert spaces property (see the Appendix), the kernel components are

$$K(\boldsymbol{z}_i, \boldsymbol{z}_j) = K_{xx}(\mathbf{x}_i, \mathbf{x}_j) + K_{yy}(\mathbf{y}_{i-1}, \mathbf{y}_{j-1}) + K_{xy}(\mathbf{x}'_i, \mathbf{y}'_{j-1}) + K_{yx}(\mathbf{y}'_{j-1}, \mathbf{x}'_i) \tag{12}$$

which can be notationally simplified as

$$\mathbf{K} = \mathbf{K}_{xx} + \mathbf{K}_{yy} + \mathbf{K}_{xy} + \mathbf{K}_{yx}. \tag{13}$$

As before, the use of this mapping function in the generic RVM system identification algorithm (9) produces the so-called RVM$_{4k}$ estimation algorithm (see the Appendix for the proof).

### C. Extended Composite Kernels

Collaborative combinations of the mapping strategies presented before can be used by just considering the combination between the RVM and the RVM$_{2K}$ structures

$$K(\boldsymbol{z}_i, \boldsymbol{z}_j) = K_{yy}(\boldsymbol{y}_{i-1}, \boldsymbol{y}_{j-1}) + K_{xx}(\boldsymbol{x}_i, \boldsymbol{x}_j) + K_z(\boldsymbol{z}_i, \boldsymbol{z}_j) \tag{14}$$

which we call RRVM$_{2K}$ algorithm, or the combination between the RVM and the RVM$_{4K}$ structures

$$K(\boldsymbol{z}'_i, \boldsymbol{z}'_j) = K_{yy}(\boldsymbol{y}_{i-1}, \boldsymbol{y}_{j-1}) + K_{xx}(\boldsymbol{x}_i, \boldsymbol{x}_j) + K_{xy}(\boldsymbol{x}'_i, \boldsymbol{y}'_{j-1}) + K_z(\boldsymbol{z}_i, \boldsymbol{z}_j) \tag{15}$$

which we call RRVM$_{4K}$ algorithm (see the Appendix).

## IV. Experimental Results

In this section, we compare the performance of the standard RVM and the RVM-ARX formulations in several examples. In order to obtain a model, the form of the mapping matrix must be defined. In the RVM framework, $\mathbf{K}$ must not necessarily fulfil Mercer's condition (as it occurs in the SVR case). Nevertheless, in this letter, we focus on the radial basis function (RBF) kernel is defined as $K(\boldsymbol{z}_i, \boldsymbol{z}_j) = \exp(-\|\boldsymbol{z}_i - \boldsymbol{z}_j\|^2/(2\sigma^2_{\text{ker}}))$, where $\sigma^2_{\text{ker}} \in \mathbb{R}^+$ represents the variance (length scale or width) of the kernel and constitutes the free parameter to be tuned for each kernel. Additionally, model order $P$ must be tuned. An exhaustive search among all free parameters is computationally unfeasible. Therefore, a non-exhaustive iterative search strategy ($T$ iterations) was used here. At each iteration, a sequential search of the minimum cross-validation error on each parameter domain is performed by splitting the range of the parameter in $K$ points. Values of $T = 3$ and $K = 20$ exhibited good performance in our simulations. All MATLAB source code of this letter is available in http://www.uv.es/gcamps/arx_rvm/ for the interested reader.

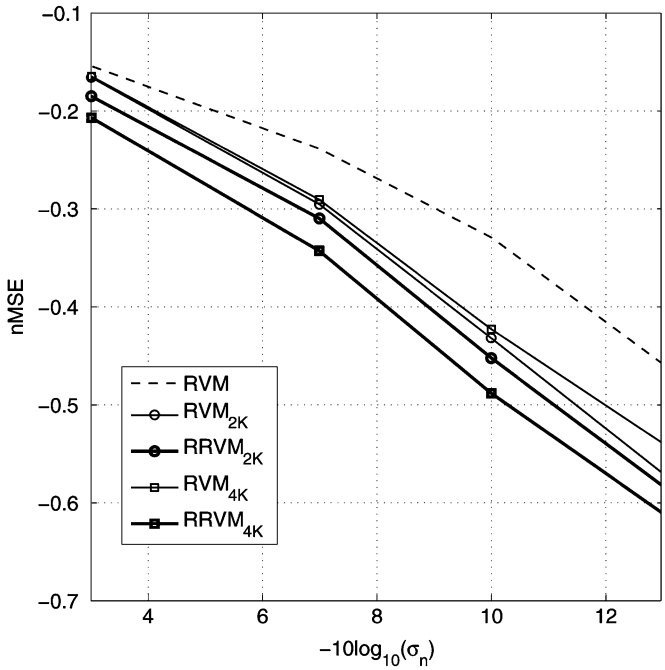


Fig. 1. nMSE as a function of additive noise of power $\sigma_n$ for all models.

### A. Nonlinear Robust System Identification

We consider the single-input single-output system originally proposed in [16], given by $y_n = (0.8 - 0.5\exp(-y^2_{n-1}))\, y_{n-1} - (0.3 + 0.9\exp(-y^2_{n-1}))\, y_{n-2} + 0.1\sin(\pi y_{n-1}) + e_n$, where $e_n$ is a Gaussian distributed random signal of zero-mean and tunable variance, $\sigma_n$, which was changed between 0 and 1. We generated a reduced training set containing 50 samples, and the following 1000 samples were used for testing. This scenario is intended to illustrate robustness to a low number of training samples and noise simultaneously. The five-fold cross-validation method was used in the training set. Averaged results over 100 realizations for the test set are shown in Fig. 1. Both RRVM models show the best performance (the most accurate being RRVM$_{4K}$), and a clear difference is observed with respect the standard RVM model, especially significant in high signal-to-noise ratios (SNRs). A certain trade-off between order and sparsity was observed for all methods but without dramatic differences (results not shown). In [17], the same system was considered, and a restriction was imposed to work with $\sigma_n < 0.2$, which is equivalent to SNR $> 7$ dB. In these cases, the proposed methods yield their best performance.

### B. Mackey–Glass Time Series

We test the presented models performance in the standard Mackey–Glass time series prediction problem, which is well known because of its strong nonlinearity. This classical high-dimensional chaotic system is generated by the delay differential equation: $dx/dt = -0.1x_n + 0.2x_{n-\Delta}/(1 + x^{10}_{n-\Delta})$, with delays $\Delta = 17$ and $\Delta = 30$, thus yielding the time series `MG17` and `MG30`, respectively. We considered 500 training samples and used the next 1000 for free parameter selection (validation set). Results are shown in Table I.

The methods proposed here outperform standard RVM, especially significant for the `MG17` time series. In the case of `MG30`,

TABLE I
RESULTS FOR THE MACKEY–GLASS TIME SERIES PREDICTION PROBLEM

| | MG17 | | | | |
|---|---|---|---|---|---|
| | **RVM** | **RVM**$_{2K}$ | **RRVM**$_{2K}$ | **RVM**$_{4K}$ | **RRVM**$_{4K}$ |
| **nMSE** | -1.687 | -2.055 | -1.976 | *-2.080* | **-2.088** |
| $P$ | 8 | 14 | 12 | 13 | 13 |
| $\hat{\sigma}_n$ | 0.0010 | 0.0008 | 0.0007 | 0.0006 | 0.0007 |
| **%RVs** | 37.88 | 24.32 | 29.97 | 25.51 | 24.07 |
| | **MG30** | | | | |
| | **RVM** | **RVM**$_{2K}$ | **RRVM**$_{2K}$ | **RVM**$_{4K}$ | **RRVM**$_{4K}$ |
| **nMSE** | -1.257 | -1.293 | -1.296 | *-1.298* | **-1.299** |
| $P$ | 6 | 6 | 6 | 6 | 6 |
| $\hat{\sigma}_n$ | 0.0092 | 0.0075 | 0.0065 | 0.0071 | 0.0064 |
| **%RVs** | 10.34 | 13.38 | 20.28 | 15.01 | 20.28 |

differences are not significant but still show a preference for RVM$_{4K}$-based models, and it suggests that this is a more complicated system. Several interesting issues can be noticed. First, the fact that this data set has virtually no output noise is better detected with all composite methods, which yields much lower estimated noise variance $\hat{\sigma}_n$ than that provided by the standard RVM. Second, the expected zero noise variance along with the chaotic nature of the time series prevent sparsity from arising in a trivial way. We observe, however, that the number of RVs retained by the proposed methods is smaller than the standard RVM in MG17, but in a scenario of increased dynamics complexity (i.e., MG30), the higher number of RVs are needed to attain competitive results.

Finally, it is worth commenting that unlike SVM-based methods, RVMs yield predictive uncertainties. This good characteristic of the method, however, has been recently related to uncontrolled sparseness, and thus, a certain trade-off between this and accuracy typically emerges [18].

## V. CONCLUSION

This letter presented a full family of RVM-based methods for nonlinear system identification. This technique not only results in improved performance, but it also opens the field to the development of other RVM-based algorithms by including *a priori* knowledge about the problem in the model. Further work will consider extension of this framework to other related kernel methods, such as Gaussian processes.

## APPENDIX
## COMPOSITE KERNELS MAPPINGS

*Cross-information kernel:* Assume a nonlinear mapping $\boldsymbol{\varphi}(\cdot)$ into $\mathcal{H}_{\varphi}$ and three linear transformations $\boldsymbol{A}_i$ from $\mathcal{H}_{\varphi}$ to $\mathcal{H}_i, i = 1, 2, 3$. Note, however, that in this case, $\boldsymbol{x}_n$ and $\boldsymbol{y}_n$ need to have the same dimension for the formulation to be valid, which can be forced by considering $P' = Q' = \max(P, Q)$. Now suppose the following composite transformation corresponding to the direct sum of Hilbert spaces:

$$\mathbf{k}(\boldsymbol{z}') = [\boldsymbol{A}_1\varphi(\boldsymbol{x}')^\top, \boldsymbol{A}_2\boldsymbol{\varphi}(\boldsymbol{y}')^\top, \boldsymbol{A}_3(\boldsymbol{\varphi}(\boldsymbol{x}') + \boldsymbol{\varphi}(\boldsymbol{y}'))^\top]^\top. \quad (16)$$

The obtained mapping matrix is

$$\mathbf{K} = \boldsymbol{\varphi}^\top(\boldsymbol{y}'_i)\boldsymbol{R}_1\boldsymbol{\varphi}(\boldsymbol{y}'_j) + \boldsymbol{\varphi}^\top(\boldsymbol{x}'_i)\boldsymbol{R}_2\boldsymbol{\varphi}(\boldsymbol{x}'_j) + \boldsymbol{\varphi}^\top(\boldsymbol{y}'_{i-1})\boldsymbol{R}_3\boldsymbol{\varphi}(\boldsymbol{x}'_j) + \boldsymbol{\varphi}^\top(\boldsymbol{x}'_i)\boldsymbol{R}_3\boldsymbol{\varphi}(\boldsymbol{y}'_{j-1}) \quad (17)$$

where $\boldsymbol{R}_1 = \boldsymbol{A}_1^\top\boldsymbol{A}_1 + \boldsymbol{A}_3^\top\boldsymbol{A}_3$, $\boldsymbol{R}_2 = \boldsymbol{A}_2^\top\boldsymbol{A}_2 + \boldsymbol{A}_3^\top\boldsymbol{A}_3$, and $\boldsymbol{R}_3 = \boldsymbol{A}_3^\top\boldsymbol{A}_3$ are three (independent) definite positive matrices.

*Extended composite mappings:* Let us define the mapping

$$\mathbf{k}(z_n) = [\boldsymbol{\phi}_y(\boldsymbol{y}_{n-1})^\top, \boldsymbol{\phi}_x(\boldsymbol{x}_n)^\top, \boldsymbol{\phi}_z(z_n)^\top]^\top. \quad (18)$$

Then, exploiting the direct sum of Hilbert spaces [19], it is straightforward to demonstrate that the induced kernel matrix is given in (14). Similarly, the mapping

$$\mathbf{k}(\boldsymbol{z}') = [\boldsymbol{A}_1\varphi(\boldsymbol{x}')^\top, \boldsymbol{A}_2\boldsymbol{\varphi}(\boldsymbol{y}')^\top, \boldsymbol{A}_3(\boldsymbol{\varphi}(\boldsymbol{x}') + \boldsymbol{\varphi}(\boldsymbol{y}'))^\top, \boldsymbol{\phi}_z(\boldsymbol{z})^T]^\top \quad (19)$$

leads to the RRVM$_{4K}$ kernel given in (15).

## REFERENCES


[1] L. Ljung, *System Identification. Theory for the User*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1999.

[2] O. Nelles, *Nonlinear System Identification. From Classical Approaches to Neural Networks and Fuzzy Models*. New York: Springer, 2001.

[3] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[4] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statist. Comput.*, vol. 14, no. 3, pp. 199–222, 2004.

[5] K. Pelckmans, I. Goethals, J. De Brabanter, J. A. K. Suykens, and B. De Moor, "Componentwise least squares support vector machines," in *Support Vector Machines: Theory and Applications*. New York: Springer, 2005.

[6] M. Espinoza, J. A. K. Suykens, and B. De Moor, "Kernel based partially linear models and nonlinear identification," *IEEE Trans Autom. Control*, vol. 50, no. 10, pp. 1602–6, Oct. 2005.

[7] J. L. Rojo-Álvarez, M. Martínez-Ramón, A. R. Figueiras-Vidal, M. de-Prado Cumplido, and A. Artés-Rodríguez, "Support vector method for ARMA system identification," *IEEE Trans. Signal Process.*, vol. 52, no. 1, pp. 155–64, Jan. 2004.

[8] 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," *Signal Process.*, vol. 85, no. 12, pp. 2316–26, 2005.

[9] M. Martínez-Ramón, J. L. Rojo-Álvarez, G. Camps-Valls, J. Muñoz-Marí, A. Navia-Vázquez, E. Soria-Olivas, and A. R. Figueiras-Vidal, "Support vector machines for nonlinear kernel ARMA system identification," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1617–1622, Nov. 2006.

[10] M. E. Tipping, , S. A. Solla, T. K. Leen, and K.-R. Müller, Eds., "The Relevance Vector Machine," in *Advances in Neural Information Processing Systems 12*. Cambridge, MA: MIT Press, 2000.

[11] D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," *IEEE Trans Signal Process.*, vol. 52, no. 8, pp. 2153–2164, Aug. 2004.

[12] A. O'Hagan, *Bayesian Inference, Volume 2B of Kendall's Advanced Theory of Statistics*. London, U.K.: Arnold, 1994.

[13] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, 2001.

[14] B. Mak, J. T. Kwok, and S. Ho, "A study of various composite kernels for kernel eigenvoice speaker adaptation," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, May 2004, vol. 1, pp. 325–328.

[15] G. Camps-Valls, L. Gomez-Chova, J. Muñoz-Marí, J. Vila-Francés, and J. Calpe-Maravilla, "Composite kernels for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 3, no. 1, pp. 93–97, Jan. 2006.

[16] S. Chen and S. A. Billings, "Neural networks for non-linear dynamic system modelling and identification," *Int. J. Control*, vol. 56, no. 2, pp. 319–346, 1992.

[17] Q. Song, L. Yin, and Y. C. Soh, "Robust adaptive identification of nonlinear system using neural network," in *Proc. IEEE Signal Process. Soc. Workshop Neural Networks Signal Processing*, Sydney, Australia, 2000, vol. 1, pp. 95–104.

[18] C. E. Rasmussen and J. Quiñonero-Candela, L. De Raedt and S. Wrobel, Eds., "Healing the Relevance Vector Machine through augmentation," in *Proc. ICML*, June 2005, pp. 689–696.

[19] M. C. Reed and B. Simon, *Functional Analysis*, ser. Methods of Modern Mathematical Physics. New York: Academic, 1980, vol. 1.