

Kernel Antenna Array Processing

Manel Martínez-Ramón, *Senior Member, IEEE*, Jose Luis Rojo-Álvarez, *Member, IEEE*, Gustavo Camps-Valls, *Member, IEEE*, and Christos G. Christodoulou, *Fellow, IEEE*.

Abstract— We introduce two support vector machine (SVM) based approaches for solving antenna problems such as beamforming, sidelobe suppression, and maximization the signal-to-noise ratio. A basic introduction to SVM optimization is provided and a complex non-linear SVM formulation is developed to handle antenna array processing in space and time. The new optimization formulation is compared with both the Minimum Mean Square Error and the Minimum Variance Distortionless Response methods. Several examples are included to show the performances of the new approaches.

Index Terms— Antenna Arrays, Support Vector Machines, Beamforming, Mercer's Kernels, Kernel Method.

I. INTRODUCTION

Support Vector Machines (SVM) have shown several advantages in prediction, regression and estimation over some of the classical approaches in a wide range of applications due to its improved generalization capabilities (see, e. g., [1]). SVMs are good candidates for tackling antenna problems such as beamforming, direction of arrival estimation, siledobe suppression, and antenna array processing in general. They can also be used in classification problems with radar and remote sensing [2], [3]. An SVM is a supervised computer learning method which exploits prior knowledge of similar scenarios and functions to identify unknown (never experienced before) cases or similar functions. Once the SVM has been trained then all future unknown cases that the SVM sees can be identified in “real time”.

In this paper we introduce the basic framework of the SVM approach as applied to antenna arrays. Array signal processing involves complex signals, for which a complex-valued formulation of the SVM is needed. We introduce this formulation by taking into account the real and imaginary parts of the error in the primal optimization and then proceeding as usual in SVM methodology [1], to solve a complex valued constrained optimization problem. The resulting algorithm is a natural counterpart of the real valued Support Vector Regression (SVR), which can be immediately applied to array signal processing. Additionally, we introduce the e-Huber cost function in order to make the regression more robust in the presence of different noise sources [4].

M. Martínez-Ramón, is with the Department of Signal Theory and Communications, Universidad Carlos III de Madrid, Avda Universidad, 30, 28911 Leganés, Madrid, Spain, (manel@ieec.org). J. L. Rojo-Álvarez is with Dept. of Signal Theory and Communications, Universidad Rey Juan Carlos, Camino del Molino s/n, 28947 Fuenlabrada, Madrid, Spain (joseluis.rojo@urjc.es). G. Camps-Valls is with Departament d'Enginyeria Electrònica, Universitat de València, Spain (gcamps@uv.es). C. G. Christodoulou is with the department of Electrical and Computer Engineering, University of New Mexico, 1 University of New Mexico, Albuquerque, New Mexico 87131-1356 (christos@ece.unm.edu)

The non-linear formulation of SVM is derived for both space and time processing purposes. We call these methods SVM with Temporal Reference (SVM-TR) and SVM with Spatial Reference (SVM-SR). Furthermore, we develop approximations that result in the nonlinear counterparts of the Minimum Mean Square Error (MMSE) and the Minimum Variance Distortionless (MVDR) methods. We call these methods Kernel Temporal Reference (Kernel-TR) and Kernel Spatial Reference (Kernel-SR) processors. The newly developed formulation is applied to optimizing the beamforming from an array antenna and maximizing its S/N ratio.

II. OPTIMUM LINEAR BEAMFORMERS

Let an M element array antenna receive L signals from different directions of arrival (DOA) θ_k , $0 \leq k \leq L-1$. The matrix form of the the array snapshot is

$$\mathbf{x}[n] = \mathbf{A}\mathbf{s}[n] + \mathbf{n}[n] \quad (1)$$

where $\mathbf{s}[n]$ is the vector of incoming signals, n represents the time instant, \mathbf{A} is the matrix of steering vectors of the incoming signals,

$$\mathbf{A} = [\mathbf{a}(\theta_0), \mathbf{a}(\theta_1), \dots, \mathbf{a}(\theta_{L-1})] \quad (2)$$

with

$$\mathbf{a}(\theta_k) = [1, e^{j2\pi\frac{d}{\lambda}\sin(\theta_k)}, \dots, e^{j2(M-1)\pi\frac{d}{\lambda}\sin(\theta_k)}]^T \quad (3)$$

where λ is the wavelength, d represents the distance between elements of the array and $\mathbf{n}[n]$ stands for the additive white Gaussian noise (AWGN) vector.

The spatial correlation matrix of the received noisy signals is

$$\begin{aligned} \mathbf{R} &= \mathbb{E}[\mathbf{x}[n]\mathbf{x}^H[n]] = \mathbf{A}\mathbb{E}[\mathbf{s}[n]\mathbf{s}^H[n]]\mathbf{A}^H + \mathbb{E}[\mathbf{n}[n]\mathbf{n}^H[n]] \\ &= \mathbf{A}\mathbf{P}\mathbf{A}^H + \sigma^2\mathbf{I} = \sum_{k=1}^M \lambda_k \mathbf{v}_k \mathbf{v}_k^H \end{aligned} \quad (4)$$

where \mathbf{P} is the autocorrelation matrix of the incoming signals and σ^2 is the noise power, and λ_k and \mathbf{v}_k are the eigenvalues and eigenvectors of \mathbf{R} respectively.

The output of the beamformer is a linear combination of the outputs of the elements that can be expressed as the dot product

$$d[n] = \mathbf{w}^H \mathbf{x}[n] + e[n] \quad (5)$$

where $d[n]$ is the estimation of the desired signal at the output of the beamformer, and $e[n]$ is the output error, which is in practice due to the errors on the parameter estimation and to the input signal noise.

Provided that certain transmitted data is known for training purposes, one may apply the MMSE criterion to the received signal, which leads to the well-known (and widely used) Wiener solution:

$$\mathbf{w} = \mathbf{R}^{-1}\mathbf{p} \quad (6)$$

where \mathbf{p} is the cross correlation between the desired output and the incoming signal.

If the DOA of the desired signal rather than a training burst is known, then a spatial reference procedure must be applied. For beamforming purposes one must minimize the output energy

$$\mathcal{E} = \mathbf{w}^H \mathbf{R} \mathbf{w} \quad (7)$$

while, simultaneously, keeping constant the amplitude of the output corresponding to the desired angle of arrival, this is $\mathbf{w}^H \mathbf{a}_d = r$ where \mathbf{a}_d is the steering vector corresponding to the desired direction of arrival, and r is the desired response to \mathbf{a}_d .

The optimization of this problem leads to the solution

$$\mathbf{w} = \frac{r \mathbf{R}^{-1} \mathbf{a}_d}{\mathbf{a}_d^H \mathbf{R}^{-1} \mathbf{a}_d} \quad (8)$$

which is the well known MVDR method [5].

The most obvious advantage of nonlinear approaches over the previous linear ones is that there is a higher chance of detecting the desired signals when there are more signals (desired plus interferences) than elements. The only constraint for the separation of signals to be possible is that two different signals corresponding to different symbols must have a low correlation, that is, they must be in different regions in the space. This fact is illustrated in Fig. 1. In this figure, the horizontal and vertical axes represent the real part of the outputs $x[n]$ and $x[n-1]$ of a two element array receiving a BPSK signal from a desired source corrupted by Gaussian noise and two interfering signals. The different clusters show signals of class +1 (circles) and -1 (dots). The signal is applied to a nonlinear receiver which constructs a boundary (solid lines) to classify the data. As the nonlinear processor has the ability of constructing a nonlinear boundary, the data can be separated. The dotted line represents the boundary constructed by a standard linear processor using the LS criterion. As it can be seen in the figure, this boundary is useless in classifying the data. Next we will show how it is possible to use SVMs to construct nonlinear equivalents of algorithms (6) and (8).

III. ROBUST COST SUPPORT VECTOR REGRESSION IN THE COMPLEX PLANE

In order to present our approach, we first develop the linear SVM algorithm, which is presented here in its complex format (see [6] for more details). This is necessary in order to deal with the electromagnetic signals when there is information present simultaneously in the in-phase and quadrature-phase parts of the incoming signals. The nonlinear extension is achieved through a projection of the incoming snapshots to a higher dimensional Hilbert space using a nonlinear transformation. Next, we introduce some of the basics of SVM.

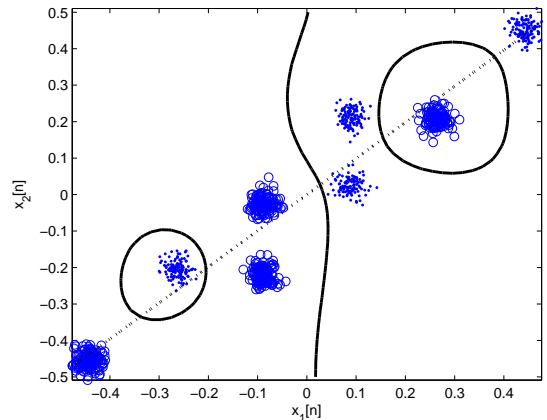


Fig. 1. Real part of the separating boundaries of the SVM-TR (continuous line) and the linear MMSE (dash) algorithms applied to an array of 2 elements with a desired DOA of 0 degrees and two interfering signals at -10 and 20 degrees. Dots and circles correspond to +1 and -1 symbols.

A. Support Vector Regression

The data model for the antenna beamformer can be written as

$$d[n] = \mathbf{w}^H \mathbf{x}[n] + e[n] \quad (9)$$

The SVM regression approach [7] can be applied to solving this model. The main idea of SVMs is to obtain a solution that jointly minimizes the norm of \mathbf{w} and a cost function $\ell(\cdot)$ of the errors. Due to the minimization of the weight vector norm, the solution will be regularized in the sense of Tikhonov [8], hence improving the generalization performance.

In antenna array processing, the received data is bandpass and not necessarily hermitic with respect to the central frequency. Thus, its low pass equivalent is a complex number, and hence, there is a need for the formulation of the SVM in the complex plane. A simple solution consists of expressing the error in real and imaginary parts and inserting them in the SVM formulation [9].

Let the set $\mathbf{x}[i]$, $1 \leq i \leq N$ be a set of snapshots and $d[i]$ the corresponding desired signals, all of them available for training purposes. The SVM regressor is, thus, the one which minimizes the functional

$$L_p = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_i \ell_\varepsilon(\xi[i], \xi'[i]) + C \sum_i \ell_\varepsilon(\zeta[i], \zeta'[i]) \quad (10)$$

subject to constraints

$$\begin{aligned} \operatorname{Re}(d[i] - \mathbf{w}^T \mathbf{x}[i]) &\leq \varepsilon + \xi[i] \\ -\operatorname{Re}(d[i] - \mathbf{w}^T \mathbf{x}[i]) &\leq \varepsilon + \xi'[i] \\ \operatorname{Im}(d[i] - \mathbf{w}^T \mathbf{x}[i]) &\leq \varepsilon + \zeta[i] \\ -\operatorname{Im}(d[i] - \mathbf{w}^T \mathbf{x}[i]) &\leq \varepsilon + \zeta'[i] \\ \xi[i], \xi'[i], \zeta[i], \zeta'[i] &\geq 0 \end{aligned} \quad (11)$$

where C is the trade-off parameter between the minimization of the norm (to improve generalization ability) and the minimization of the errors [10]. The quantities $\xi[i]$, $\xi'[i]$, $\zeta[i]$

and $\zeta'[i]$ are the *slack variables* or losses associated with each sample $\mathbf{x}[n]$. Slack variables are used to convert the constraints in equation (11) to equalities. The optimization is intended to minimize a cost function over these variables. The parameter ε is used to allow those slack variables for which the error is less than ε to be zero. This is equivalent to the minimization of the so-called ε -insensitive (or Vapnik's) Loss Function [10], whose expression is:

$$\ell_\varepsilon(u) = \begin{cases} |u| - \varepsilon & |u| > \varepsilon \\ 0 & |u| \leq \varepsilon \end{cases} \quad (12)$$

The optimization of the above constrained problem is solved by constructing a Lagrange functional with multipliers α_i, α'_i and β_i, β'_i applied to the real and imaginary parts of constraints (11). It leads to the explicit SVM solution, given by

$$\mathbf{w} = \sum_{n=1}^N \psi_n \mathbf{x}[n] \quad (13)$$

If one applies this solution to the Lagrange functional, the following *dual* functional is obtained (see details in [6]).

$$L_d = -\boldsymbol{\psi}^H \mathbf{K} \boldsymbol{\psi} + \mathbb{R}e(\boldsymbol{\psi}^T \mathbf{d}^*) - (\boldsymbol{\alpha} + \boldsymbol{\alpha}' + \boldsymbol{\beta} + \boldsymbol{\beta}') \mathbf{1}\varepsilon \quad (14)$$

where \mathbf{K} is the Gram matrix whose element (i, k) is given by the dot product

$$K[i, k] = \mathbf{x}^H[i] \mathbf{x}[k] \quad (15)$$

and $\boldsymbol{\psi} = \boldsymbol{\alpha} - \boldsymbol{\alpha}' - j(\boldsymbol{\beta} - \boldsymbol{\beta}')$, where $\boldsymbol{\alpha}, \boldsymbol{\alpha}', \boldsymbol{\beta}$ and $\boldsymbol{\beta}'$ are column vectors containing the Lagrange multipliers. The dual is minimized with respect to these Lagrange multipliers.

Dual (14) involves the Gram matrix \mathbf{K} of the dot products of the data vectors $\mathbf{x}[i]$. This matrix may be almost singular. To avoid this numerical inconvenience, a small diagonal $\gamma \mathbf{I}$ is added to the matrix prior to the numerical optimization. Thus, the dual that is always used in practice is

$$L_d = -\boldsymbol{\psi}^H (\mathbf{K} + \gamma \mathbf{I}) \boldsymbol{\psi} + \mathbb{R}e(\boldsymbol{\psi}^T \mathbf{d}^*) - (\boldsymbol{\alpha} + \boldsymbol{\alpha}' + \boldsymbol{\beta} + \boldsymbol{\beta}') \mathbf{1}\varepsilon \quad (16)$$

Functional (16) can be derived by applying a modified cost function that combines both ε -insensitive and Huber [11] cost function, and that was introduced in [12] for SVM applied to system identification. The cost function has the expression:

$$\ell_R(u) = \begin{cases} 0 & |u| < \varepsilon \\ \frac{1}{2\gamma C} (|u| - \varepsilon)^2 & \varepsilon \leq |u| \leq e_C \\ (|u| - \varepsilon) - \frac{1}{2}\gamma C & e_C \leq |u| \end{cases} \quad (17)$$

where $e_C = \varepsilon + \gamma C$. Using this cost function leads to the Lagrange multipliers [9]

$$\psi_n = \begin{cases} 0 & |e| < \varepsilon \\ \frac{1}{\gamma} (e[n] - \varepsilon) & \varepsilon \leq |e[n]| \leq e_C \\ C & e_C \leq |e[n]| \end{cases} \quad (18)$$

It can be shown that for $\varepsilon \neq 0$, only a subset of Lagrange multipliers ψ_i will be nonzero. The samples $\mathbf{x}[i]$ associated

with the nonzero Lagrange multipliers are the so-called Support Vectors (SV).

The cost function in (17) provides a functional that is numerically regularized by the matrix $\gamma \mathbf{I}$. The cost function is quadratic for the data which produce errors between ε and e_C , and linear for errors above e_C . Thus, one can adjust the parameter e_C to apply a quadratic cost for the samples which are mainly affected by thermal noise (i.e., for which the quadratic cost is Maximum Likelihood). The linear cost is then applied to the those samples corrupted by impulse or other nongaussian, or high amplitude noises. Using a linear cost function, the contribution of these samples to the solution will not depend on its error value, but only on its sign, thus avoiding the bias that a quadratic cost function produces.

B. The Mercer Theorem and the nonlinear SVM

A theorem provided by Mercer [13] in the early 1900's is of extreme relevance because it extends the principle of linear SVM (minimization of a regularized cost with linear constraints) to the nonlinear case. The basic idea is that vectors \mathbf{x} in a finite dimension space (called input space) can be mapped to a higher (possibly infinite) dimension in Hilbert space \mathcal{H} provided with a dot product, through a nonlinear transformation $\varphi(\cdot)$. A linear machine can be constructed in a higher dimensional space [14], [15] (often called the feature space) which will be nonlinear from the point of view of the input space.

The Mercer's theorem shows that there exist a function $\varphi : \mathcal{R}^n \rightarrow \mathcal{H}$ and a dot product

$$K(\mathbf{x}_i, \mathbf{x}_k) = \varphi(\mathbf{x}_i)^T \varphi(\mathbf{x}_k) \quad (19)$$

if and only if $K(\cdot, \cdot)$ is a positive integral operator on a Hilbert space, i.e, if only if for any function $g(\mathbf{x})$ for which

$$\int g(\mathbf{x}) d\mathbf{x} < \infty \quad (20)$$

the inequality

$$\int K(\mathbf{x}, \mathbf{y}) g(\mathbf{x}) g(\mathbf{y}) d\mathbf{x} d\mathbf{y} \geq 0 \quad (21)$$

holds. Hilbert spaces provided with dot products that fit the Mercer's theorem are often called Reproducing Kernel Hilbert Spaces (RKHS).

The first kernels to be proven to fit the Mercer's theorem were the homogeneous polynomial kernel

$$K(\mathbf{x}_i, \mathbf{x}_k) = (\mathbf{x}_i^T \mathbf{x}_k)^p \quad (22)$$

and the inhomogeneous polynomial kernel

$$K(\mathbf{x}_i, \mathbf{x}_k) = (\mathbf{x}_i^T \mathbf{x}_k + 1)^p \quad (23)$$

Another important kernel type is the Gaussian kernel, because it is widely used in many practical applications. Its expression is

$$K(\mathbf{x}_i, \mathbf{x}_k) = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_k\|^2}{2\delta^2}} \quad (24)$$

In this case, the corresponding nonlinear mapping is not explicit and the dimension of its Hilbert space is infinite.

C. Construction of a nonlinear Support Vector Regressor

A nonlinear SVM can be constructed by mapping the incoming data $\mathbf{x}[n]$ into a RKHS using a nonlinear transformation $\varphi(\cdot)$. The dot product of the corresponding spaces can be expressed as a function of the input vectors as in (19).

The Gram matrix of the dot products can be expressed as

$$\mathbf{K} = \Phi^H \Phi \quad (25)$$

where $\Phi = [\varphi(\mathbf{x}[1]), \dots, \varphi(\mathbf{x}[N])]$ and represent the mapped input vectors to a (possibly infinite dimension) RKHS. Note that the nonlinear transformation does not need to be explicitly known to construct the nonlinear SVM, because the only needed operator is the dot product or, equivalently, the kernel.

The solution for the linear SVR is given by a linear combination of a subset of the training data mapped into the RKHS

$$\mathbf{w} = \sum_{i=1}^N \psi_i \varphi(\mathbf{x}_i) \quad (26)$$

The expression of the regressor is now

$$d[n] = \mathbf{w}^H \varphi(\mathbf{x}[n]) + b \quad (27)$$

where the bias b is added because the data is not necessarily centered around the origin in the Hilbert space. Plugging (26) into (27), yields

$$\begin{aligned} d[n] &= \sum_{i=1}^N \psi_i \varphi(\mathbf{x}[i])^H \varphi(\mathbf{x}[n]) + b \\ &= \sum_{i=1}^N \psi_i K(\mathbf{x}[i], \mathbf{x}[n]) + b \end{aligned} \quad (28)$$

The resulting machine can now be expressed directly in terms of the Lagrange multipliers and the kernel dot products. In order to solve the dual functional which determines the Lagrange multipliers, the vectors are not required either, but only the Gram matrix \mathbf{K} of the dot products between them. Here the kernel is used to compute this matrix, as in (19).

Once this matrix has been computed, solving for a nonlinear SVM is as easy as solving for a linear one. It can be shown that if the kernel fits the Mercer theorem, the matrix will be positive definite [13].

In the following sections we introduce four alternatives in using the above mentioned techniques for nonlinear antenna array signal processing.

IV. NONLINEAR ARRAY PROCESSORS WITH TEMPORAL REFERENCE

Let us assume that no knowledge about the desired direction of arrival is available, but that there is a sequence of data available for training purposes. The output of the nonlinear beamformer given the training data is then

$$d[i] = \mathbf{w}^H \varphi(\mathbf{x}[i]) + b + e[i] \quad (29)$$

One can apply the procedure in Section III and obtain a solution for the beamformer of the form (28) by solving the dual functional (16) where the elements of matrix \mathbf{K} are the

kernel dot products $\mathbf{K}_{ik} = K(\mathbf{x}_i, \mathbf{x}_k)$ (see also [2], [4]). This is the simplest solution and, as there is a training sequence, it constitutes an SVM array processor with temporal reference (SVM-TR).

In this case, the following property holds.

Property 1: The SVM-TR processor approaches the Wiener (temporal reference) processor as $C \rightarrow \infty$ and $\varepsilon = 0$.

Proof: If we choose $\varepsilon = 0$ and $C \rightarrow \infty$ in (18), the Lagrange multipliers are equal to the estimation errors, this is, $\psi_i = \frac{e^*[i]}{\gamma}$. Under these conditions, the dual functional (16) can be rewritten as

$$L_d = -\frac{1}{\gamma^2} \mathbf{e}^T (\mathbf{K} + \gamma \mathbf{I}) \mathbf{e}^* + \frac{1}{\gamma} \Re e (\mathbf{e}^T \mathbf{d}^*)$$

\mathbf{e} being a column vector containing the errors $e[i]$. If these functional is minimized with respect the errors, the following expression holds

$$-\frac{1}{\gamma^2} (\mathbf{K} + \gamma \mathbf{I}) \mathbf{e}^* + \frac{1}{\gamma} \mathbf{d}^* = 0 \quad (30)$$

and, taking into account that $\mathbf{e}^* = \mathbf{d}^* - \Phi \mathbf{w}$ and that $\mathbf{K} = \Phi^H \Phi$, the weight vector \mathbf{w} can be straightforwardly isolated as

$$\mathbf{w} = (\mathbf{R} + \gamma \mathbf{I})^{-1} \mathbf{p} \quad (31)$$

where $\mathbf{R} = \Phi \Phi^H$ and $\mathbf{p} = \Phi \mathbf{d}^*$. ■

Under these conditions, and assuming that parameters \mathbf{w} are necessarily a combination of the data of the form $\mathbf{w} = \Phi \psi$, the solution for the multipliers of (28) is

$$\psi = (\mathbf{K} + \gamma \mathbf{I})^{-1} \mathbf{d}^* \quad (32)$$

which is a kernelized version of the Wiener equation (Kernel-TR).

V. SVM ARRAY PROCESSOR WITH SPATIAL REFERENCE

A. Formulation of the Processor

A kernel array processor with a spatial reference must include a minimization of the output power similar to those of the MVDR in (8). A simple solution to include the power minimization in a linear Support Vector Beamformer has been introduced in [16]. Here we present a different approach based on the same idea, with a direct complex formulation and a nonlinear solution (SVM-SR).

Let us assume now that the direction of arrival, rather than a training sequence, is known by the receiver. Then, one can write the following primal functional similar to (7):

$$L_p = \frac{1}{2} \mathbf{w}^H \mathbf{R} \mathbf{w} + C \sum_i \ell_R(\xi_i + \xi_i') + C \sum_i \ell_R(\zeta_i + \zeta_i') \quad (33)$$

where here the autocorrelation matrix has the expression

$$\mathbf{R} = \frac{1}{N} \Phi \Phi^H \quad (34)$$

We apply a kernelized version of the constraints of the standard MVDM but adapted to the SVM formulation of Section III. Let us assume a quadrature amplitude modulation (QAM) and

let r_i , $1 \leq i \leq M$ be all the possible transmitted symbols. Then, the set of constraints is

$$\begin{aligned} \Re(r_i - \mathbf{w}^H \boldsymbol{\varphi}(r_i \mathbf{a}_d) - b) &\leq \varepsilon + \xi_i \\ -\Re(r_i - \mathbf{w}^H \boldsymbol{\varphi}(r_i \mathbf{a}_d) - b) &\leq \varepsilon + \xi'_i \\ \Im(r_i - \mathbf{w}^H \boldsymbol{\varphi}(r_i \mathbf{a}_d) - b) &\leq \varepsilon + \zeta_i \\ -\Im(r_i - \mathbf{w}^H \boldsymbol{\varphi}(r_i \mathbf{a}_d) - b) &\leq \varepsilon + \zeta'_i \end{aligned} \quad (35)$$

The difference between these constraints and the ones of the linear MVDR is that in the linear case we use constant r as the required output to input \mathbf{a}_d . If the input is multiplied by a complex constant, then the output will be equally scaled. This is not the case here because we deal with a nonlinear transformation. Thus, we must specify in the constraints all possible complex desired outputs r_i .

Applying Lagrange analysis to primal functional (33) gives the result

$$\mathbf{w} = \mathbf{R}^{-1} \boldsymbol{\Phi}_d \boldsymbol{\psi} \quad (36)$$

where $\boldsymbol{\Phi}_d = [\boldsymbol{\varphi}(r_1 \mathbf{a}_d), \dots, \boldsymbol{\varphi}(r_M \mathbf{a}_d)]^T$. Applied to the primal, the previous result leads to the dual

$$\begin{aligned} L_d = -\frac{1}{2} \boldsymbol{\psi}^H \left[\boldsymbol{\Phi}_d^H \mathbf{R}^{-1} \boldsymbol{\Phi}_d + \gamma \mathbf{I} \right] \boldsymbol{\psi} + \Re(\boldsymbol{\psi}^T \mathbf{r}^*) - \\ - \varepsilon \mathbf{1}(\boldsymbol{\alpha} + \boldsymbol{\beta} + \boldsymbol{\alpha}' + \boldsymbol{\beta}') \end{aligned} \quad (37)$$

A regularization term naturally appears from the application of the ε -Huber cost function (17).

B. Eigenanalysis in the Feature Space

The algorithm as explained above is not solvable due to the fact that we do not have access to the data into the feature space, but only to the original Gram matrix \mathbf{K} . But we still can indirectly solve the problem applying Kernel Principal Component Analysis (KPCA) techniques [17]. Let the autocorrelation matrix in the feature space be defined as in (34). Expressing the inverse of the autocorrelation matrix as $\mathbf{R}^{-1} = \mathbf{U} \mathbf{D}^{-1} \mathbf{U}^H$, one can rewrite the dual as

$$\begin{aligned} L_d = -\frac{1}{2} \boldsymbol{\psi}^H \left[\boldsymbol{\Phi}_d^H \mathbf{U} \mathbf{D}^{-1} \mathbf{U}^H \boldsymbol{\Phi}_d + \gamma \mathbf{I} \right] \boldsymbol{\psi} + \Re(\boldsymbol{\psi}^T \mathbf{r}^*) - \\ - \varepsilon \mathbf{1}(\boldsymbol{\alpha} + \boldsymbol{\beta} + \boldsymbol{\alpha}' + \boldsymbol{\beta}') \end{aligned} \quad (38)$$

The optimization of the dual (38) gives us the Lagrange multipliers $\boldsymbol{\psi}$ from which one can compute the optimal weight vector (36).

The eigenvalues \mathbf{D} and eigenvectors \mathbf{U} of \mathbf{R} satisfy

$$\mathbf{D} \mathbf{U} = \mathbf{R} \mathbf{U} \quad (39)$$

The eigenvectors can be expressed as a linear combination of the data set as

$$\mathbf{U} = \boldsymbol{\Phi} \mathbf{V} \quad (40)$$

and plugging (40) in (39) and premultiplying by $\boldsymbol{\Phi}^H$ we get

$$\mathbf{D} \boldsymbol{\Phi}^H \boldsymbol{\Phi} \mathbf{V} = \boldsymbol{\Phi}^H \frac{1}{N} \boldsymbol{\Phi} \boldsymbol{\Phi}^H \boldsymbol{\Phi} \mathbf{V} \quad (41)$$

Using the definition (25) of the Gram matrix and simplifying we obtain

$$N \mathbf{D} \mathbf{V} = \mathbf{K} \mathbf{V} \quad (42)$$

The first implication of this equation is that if λ is an eigenvalue of \mathbf{R} then $N\lambda$ is an eigenvalue of \mathbf{K} and that the matrix \mathbf{V} of coefficients are the corresponding eigenvectors of \mathbf{K} . Thus

$$\mathbf{K} = N \mathbf{V} \mathbf{D} \mathbf{V}^H \quad (43)$$

The fact that the eigenvectors of \mathbf{R} must be normalized yields to the normalization condition $1 = \lambda_i \mathbf{v}_i^T \mathbf{v}_i$. Also, in order to compute the eigenvectors of \mathbf{R} , it is assumed that the data is centered around the origin, which is, in general, not true. Therefore, for the sake of simplicity, we will assume that the data is centered and, at the end of the analysis, we will force that situation as it is done in [17].

Putting expression (40) in (38) gives the result

$$\begin{aligned} L_d = \frac{1}{2} \boldsymbol{\psi}^H \left[\boldsymbol{\Phi}_d^H \boldsymbol{\Phi} \mathbf{V} \mathbf{D}^{-1} \mathbf{V}^H \boldsymbol{\Phi}_d + \gamma \mathbf{I} \right] \boldsymbol{\psi} - \\ - \Re(\boldsymbol{\psi}^T \mathbf{d}^*) + \varepsilon \mathbf{1}(\boldsymbol{\alpha} + \boldsymbol{\beta} + \boldsymbol{\alpha}' + \boldsymbol{\beta}') \end{aligned} \quad (44)$$

Using expression (43) yields to

$$\begin{aligned} L_d = \frac{1}{2} \boldsymbol{\psi}^H \left[N \mathbf{K}_d^H \mathbf{K}^{-1} \mathbf{K}_d + \gamma \mathbf{I} \right] \boldsymbol{\psi} - \\ - \Re(\boldsymbol{\psi}^T \mathbf{r}^*) + \varepsilon \mathbf{1}(\boldsymbol{\alpha} + \boldsymbol{\beta} + \boldsymbol{\alpha}' + \boldsymbol{\beta}') \end{aligned} \quad (45)$$

This expression contains now two matrices that can be computed. The first one is the Gram matrix of kernel products whose elements are defined in (19). The second one is the matrix $\mathbf{K}_d = \boldsymbol{\Phi}_d^H \boldsymbol{\Phi}_d$ whose elements are $K(\mathbf{x}[n], r_i \mathbf{a}_d)$. This dual functional can be optimized using a quadratic programming procedure (see, e.g. [18]).

Putting (40) into (36) gives the expression of the weights as a function of the dual parameters

$$\begin{aligned} \mathbf{w} = \mathbf{R}^{-1} \boldsymbol{\Phi}_d \boldsymbol{\psi} = \boldsymbol{\Phi} \mathbf{V} \mathbf{D}^{-1} \mathbf{V}^H \boldsymbol{\Phi}_d \boldsymbol{\psi} = \\ = N \boldsymbol{\Phi} \mathbf{K}^{-1} \mathbf{K}_d \boldsymbol{\psi} \end{aligned} \quad (46)$$

and then the SVM output for a snapshot $\mathbf{x}[n]$ can be expressed as

$$\begin{aligned} d[n] = \mathbf{w}^H \boldsymbol{\varphi}(\mathbf{x}[n]) + b = \\ = N \boldsymbol{\psi}^H \mathbf{K}_d \mathbf{K}^{-1} \boldsymbol{\Phi}_d \boldsymbol{\varphi}(\mathbf{x}[n]) + b = \\ = N \boldsymbol{\psi}^H \mathbf{K}_d \mathbf{K}^{-1} \mathbf{k}[n] + b \end{aligned} \quad (47)$$

where $\mathbf{k}[n] = [K(\mathbf{x}[1], \mathbf{x}[n]), \dots, K(\mathbf{x}[N], \mathbf{x}[n])]^T$ is the vector of dot products of the vector $\boldsymbol{\varphi}(\mathbf{x}[n])$ with all the training vectors $\boldsymbol{\varphi}(\mathbf{x}[i])$, $1 \leq i \leq N$.

C. Centering the Data in a Hilbert Space

In order to be able to find the autocorrelation matrix, the above algorithm, as it is explained, assumes that data is centered in the origin in the feature space, which can be done by transforming all samples as

$$\tilde{\boldsymbol{\varphi}}(\mathbf{x}[i]) = \boldsymbol{\varphi}(\mathbf{x}[i]) - \frac{1}{N} \sum_k \boldsymbol{\varphi}(\mathbf{x}[k]) \quad (48)$$

and computing its Gram matrix gives

$$\tilde{\mathbf{K}} = \mathbf{K} - \mathbf{B} \mathbf{K} - \mathbf{K} \mathbf{B} + \mathbf{B} \mathbf{K} \mathbf{B} \quad (49)$$

where \mathbf{B} is an $N \times N$ matrix whose elements are equal to $\frac{1}{N}$.

We also need an expression to apply the same transformation to the new data during the performance phase. For each new snapshot that arrives to the receiver, the dot products with all the support vectors must be computed in order to obtain the output (47). The vector of dot products of the matrix Φ of N centered training vectors and a new centered snapshot $\mathbf{x}[m]$ in the feature space can be expressed as [17]

$$\tilde{\mathbf{k}}[n] = \mathbf{k}[n] - \mathbf{b}\mathbf{K} - \mathbf{k}[n]\mathbf{B} + \mathbf{b}\mathbf{K}\mathbf{B} \quad (50)$$

where \mathbf{b} is a row vector whose elements are equal to $1/N$.

D. Approximation to Nonlinear MVDM

The above algorithm combines the generalization properties of the SVM with the interference rejection ability of the MVDM. The main drawback of that algorithm is its computational burden. In addition to a number of matrix operations, a quadratic optimization procedure is needed to optimize the dual functional (38). Nevertheless, an alternative solution can be used in order to avoid the quadratic optimization.

We can reformulate the above as follows:

Property 2: The SVM-SR approaches the MVDR in the Hilbert Space as $C \rightarrow \infty$ and $\varepsilon = 0$

Proof: If we choose $\varepsilon = 0$ and $C \rightarrow \infty$, then the Lagrange multipliers are equal to the estimation errors, this is, $\psi_i = \frac{e^*[i]}{\gamma}$. Under these conditions, one can rewrite the dual (37) as

$$L = \frac{1}{2\gamma^2} \mathbf{e}^H \left[\Phi_d^H \mathbf{R}^{-1} \Phi_d + \gamma \mathbf{I} \right] \mathbf{e} - \frac{1}{\gamma} \Re \{ \mathbf{e}^H \mathbf{r} \} \quad (51)$$

The optimization of the functional with respect to the errors is solved by computing its derivative and equalling it to zero. This gives the result

$$\mathbf{0} = \frac{1}{\gamma^2} \Phi_d^H \mathbf{R}^{-1} \Phi_d \mathbf{e} - \frac{1}{\gamma} \mathbf{r} + \frac{1}{\gamma} \mathbf{e} \quad (52)$$

Taking into account that $\mathbf{e} = \mathbf{r} - \mathbf{w}^H \Phi_d$ and isolating the weight vector, we obtain the following result

$$\mathbf{w} = \mathbf{R}^{-1} \Phi_d (\Phi_d^H \mathbf{R}^{-1} \Phi_d + \gamma \mathbf{I})^{-1} \mathbf{r} \quad (53)$$

which is the kernel counterpart of the MVDR of (8), except for a numerical regularization term (Kernel-SR processor). ■

Putting that equation and result (53) together, the following approximation can be made

$$\mathbf{w} = \mathbf{R}^{-1} \Phi_d \psi \approx \mathbf{R}^{-1} \Phi_d (\Phi_d^H \mathbf{R}^{-1} \Phi_d^{-1} + \gamma \mathbf{I})^{-1} \mathbf{r} \quad (54)$$

From (54), the following expression can be derived

$$\psi \approx (\Phi_d^H \mathbf{R}^{-1} \Phi_d + \gamma \mathbf{I})^{-1} \mathbf{r} \quad (55)$$

which is an approximate solution of the optimization for $\varepsilon = 0$, $C \rightarrow \infty$, and $\gamma \rightarrow \infty$.

Setting ε to zero is justified by the fact that, for the case of data corrupted by Gaussian noise, the optimum value of ε is proportional to the noise standard deviation [19], [20]. Thus, in many situations this noise deviation is small enough to make ε negligible. Also, if the noise is Gaussian, it is reasonable to make C big enough to consider cost function (17) only quadratic, as then it will be the optimal cost function from a Maximum Likelihood viewpoint.

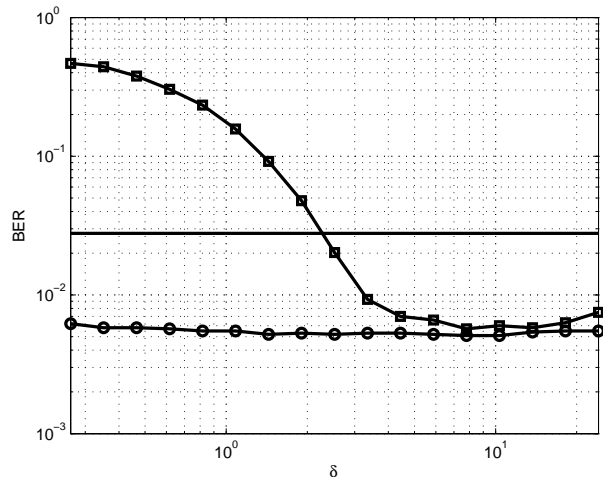


Fig. 2. BER performance, as a function of Gaussian RBF kernel parameter δ , of the SVM-TR (squares) and the SVM-SR (circles) in an array of 7 elements and with three interferent signals. Continuous line corresponds to the performance of the linear algorithms.

VI. RESULTS

We compare the temporal reference and the spatial reference array processors to the linear MMSE and MVDM in three experiments. The kernel that has been used in all simulations is the Gaussian RBF of (24). Parameters δ and γ have been chosen by sweeping in the first experiment, and assessing that there is not a dramatic sensitivity of the algorithms to these parameters. Parameters ε and C have been chosen for the nonlinear approaches using the criteria introduced in [19], [20]. The scenario consists of a multiuser environment with four users, one of which is the desired user and the rest are considered interferences. Each one transmits a statistically independent QPSK signal. The signal is corrupted by AWGN. The desired signal is a train of bursts with a preamble of 100 training bits plus 1000 test bits.

In the first experiment, we measure the Bit Error Rate (BER) of the algorithms as a function of the kernel parameter δ for arrays of 5 and 7 elements. Interferences have DOAs of 10° , 20° and -10° , and unitary amplitudes. The desired signal comes from a DOA of 0° and it has the same amplitude.

Figure 2 depicts the BER performance as a function of the RBF kernel width δ for the temporal and spatial reference algorithms optimized using SVM. Fig. 3 compares the temporal and spatial approximate algorithms (i.e., Kernel-TR and SR). For these figures the array had 7 elements, and the noise power was of -1 dB. Figures 4 and 5 show the same comparisons for an array of 5 elements and a noise power of -6 dB.

When the number of array elements is high enough, the non-linear spatial reference algorithms have a good performance and they are less sensitive than the time reference algorithms with respect to δ . Using less elements, the performance of the spatial reference algorithms degrades, while the temporal reference algorithms are able to solve the problem, due to the fact that they use more information to construct the separating hyperplane.

In the second experiment, we compute approximate beam

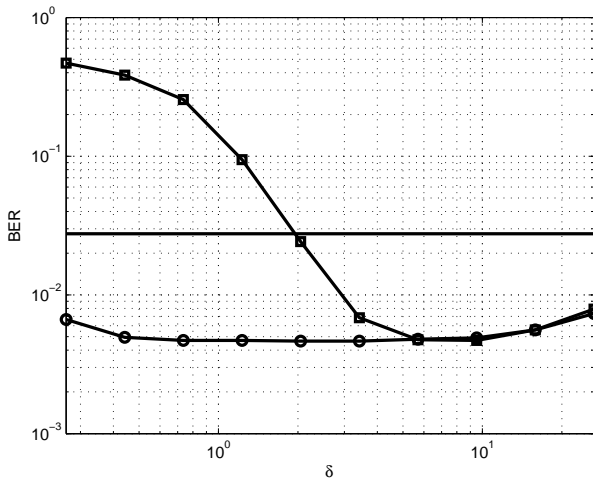


Fig. 3. BER performance, as a function of Gaussian RBF kernel parameter δ , of the Kernel-TR (squares) and the Kernel-SR (circles) in an array of 7 elements and with three interferent signals. Continuous line corresponds to the performance of the linear algorithms.

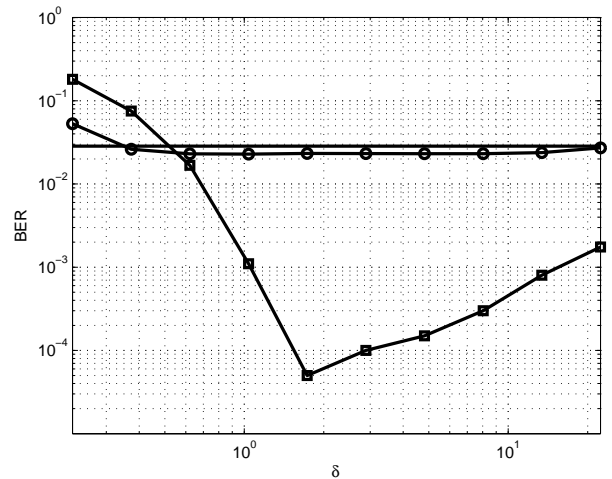


Fig. 5. BER performance with respect to the parameter δ of the Kernel-TR (squares) and the Kernel-SR (circles) with Gaussian RBF kernels in an array of 5 elements and with three interferent signals. Continuous line corresponds to the performance of the linear algorithms.

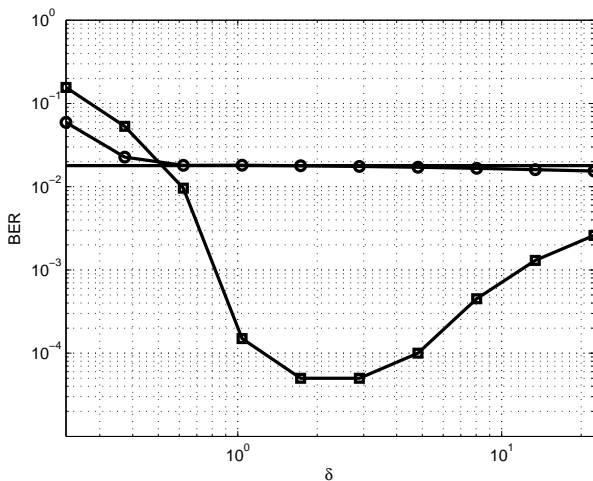


Fig. 4. BER performance with respect to the parameter δ of the SVM-TR (squares) and the SVM-SR (circles) with Gaussian RBF kernels in an array of 5 elements and with three interferent signals. Continuous line corresponds to the performance of the linear algorithms.

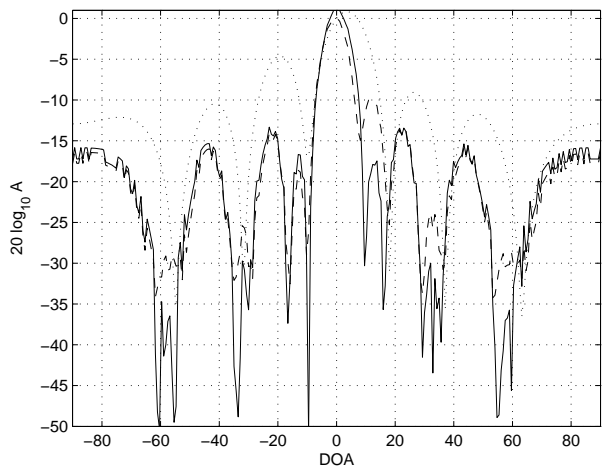


Fig. 6. Radiation pattern of the SVM-TR (cont. line) and the SVM-SR (dash) with Gaussian RBF kernels in an array of 7 elements and with one interferent signal at a DOA of -10° . Dots corresponds to the beam of the linear MVDM.

shapes of the TR and SR algorithms. They have been computed by measuring the response of the systems to different DOAs. The number of elements for this experiment was 7, the desired signal has a DOA of 0° , and there is an interfere at -10° .

Figure 6 shows the results for the SVM-TR and SVM-SR, and Fig. 7 shows the results for the Kernel-TR and Kernel-SR algorithms, compared to the beam shape of the linear MVDM. These figures show an improvement of the nonlinear methods in sidelobe amplitude reduction. Also, their interference rejection is deeper than the one of the linear algorithm, and the main lobes have their maxima at a DOA much closer to 0° than the one of the linear algorithm.

The third experiment measures the BER of the four nonlinear processors as a function of the thermal noise power

σ^2 . The performances are compared to those of the linear MVDR and MMSE algorithms. The array had 7 elements. The interferences had DOAs of -10° , 10° and 20° . The desired signal had a DOA of 0° . All amplitudes were unitary. In all simulations, parameters were the following: Gaussian kernel with width $\delta = 10$, $\gamma = 1e-6$, $C = 1$, $\varepsilon = \sigma$. Figure 8 shows the results for all processors. All nonlinear approaches show similar performance, and an improvement of several decibels with respect to the linear algorithms.

VII. CONCLUSIONS

We have introduced two nonlinear SVM algorithms and their approximations to kernel MMSE and kernel MVDM that can be used in antenna array beamforming, DOA estimation and sidelobe suppression of beam shaping. The introduced

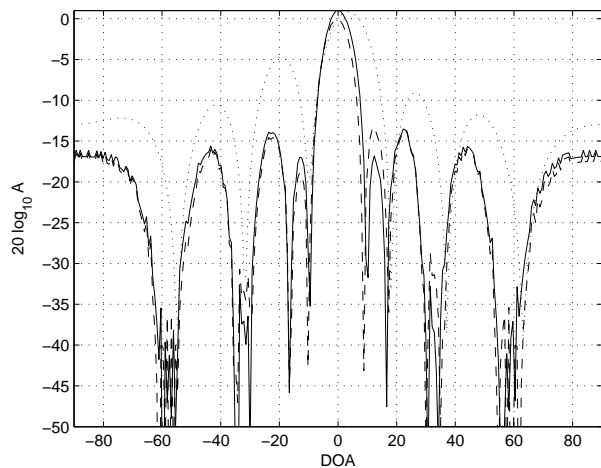


Fig. 7. Radiation pattern of the Kernel-TR (cont. line) and the Kernel-SR (dash) with Gaussian RBF kernels in an array of 7 elements and with one interferent signal at a DOA of -10° . Dots corresponds to the beam of the linear MVDM.

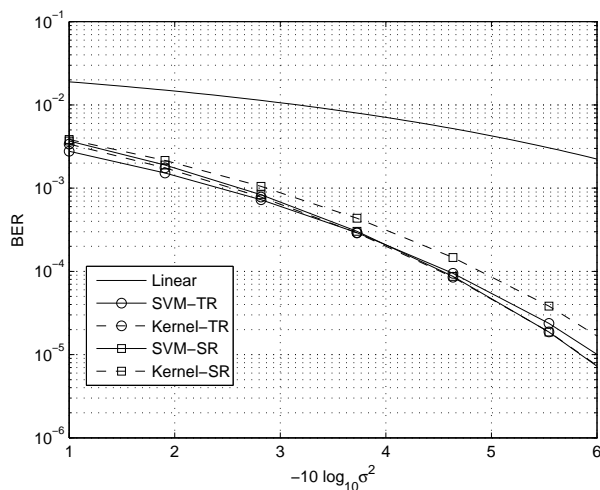


Fig. 8. BER performance as a function of thermal noise power for linear algorithms, SVM SVM-TR, SVM-SR, Kernel-TR and Kernel-SR.

methods have a clear advantage over linear MMSE and MVDM based algorithms in those cases where small data sets are available for training and where non Gaussian noise is present. In order to make the algorithm adequate to array processing purposes, we first apply an alternative cost function which is suitable in problems in which there are gaussian noise and other non Gaussian sources, and interference in a multiuser environment. Compared to other classical nonlinear methods, the introduced algorithms yield a single solution, thus avoiding the local minima problem of other approaches such as Neural Networks. We provided several examples for a number of linear array antennas to demonstrate the validity of the algorithms and how they compare with the linear algorithms.

ACKNOWLEDGEMENTS

We want to thank the useful discussions and comments of Ignacio Santamaría, Universidad de Cantabria, César C. Gaudés, Universidad de Zaragoza, and M. A. Lagunas, Centre Tecnològic de Telecomunicacions de Catalunya.

REFERENCES

- [1] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, 2004.
- [2] M. Martínez-Ramón and C. G. Christodoulou, *Support Vector Machines for Antenna Array Processing and Electromagnetics*, Synthesis Lectures on Computational Electromagnetics, Morgan & Claypool Publishers, CA, USA, 2006.
- [3] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 6, pp. 1351–1362, 2005.
- [4] M. Martínez-Ramón, N. Xu, and C. Christodoulou, "Beamforming using support vector machines," *IEEE Antennas and Wireless Propagation Letters*, vol. 4, pp. 439–442, 2005.
- [5] J. Capon, "High resolution frequency-wavenumber spectrum analysis," *Proceedings of the IEEE*, vol. 57, no. 8, pp. 1408–1418, Aug 1969.
- [6] G. Camps-Valls, J. L. Rojo-Álvarez, and M. Martínez-Ramón, Eds., *Kernel Methods in Bioengineering, Signal and Image Processing*, Idea Group Inc., Harshey, PA (USA), Nov. 2006.
- [7] A. J. Smola and Schölkopf B, "A tutorial on support vector regression," *Statistics and Computing*, vol. 4, no. 3, pp. 199–222, Aug. 2004.
- [8] A.N. Tikhonov and V.Y. Arsenen, *Solution to Ill-Posed Problems*, V.H. Winston & Sons, 1977.
- [9] J. L. Rojo-Álvarez, G. Camps-Valls, M. Martínez-Ramón, A. Navia-Vázquez, and A. R. Figueiras-Vidal, "Support Vector Machines Framework for Linear Signal Processing," *Signal Processing*, vol. 85, no. 12, pp. 2316–2326, dec 2005.
- [10] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, 1998.
- [11] P. J. Huber, *Robust Statistics: a review*, vol. 43, Ann. Statistics, 1972.
- [12] J. L. Rojo-Álvarez, M. Martínez-Ramón, M. dePrado Cumplido, A. Artés-Rodríguez, and A. R. Figueiras-Vidal, "Support Vector Method for Robust ARMA System Identification," *IEEE Trans. on Signal Processing*, vol. 52, no. 1, pp. 155–164, Jan 2004.
- [13] M. A. Aizerman, E. M. Braverman, and L.I. Rozoner, "Theoretical foundations of the potential function method in pattern recognition learning," *Automation and remote Control*, vol. 25, pp. 821–837, 1964.
- [14] V. Vapnik, *Statistical Learning Theory, Adaptive and Learning Systems for Signal Processing, Communications, and Control*, John Wiley & Sons, 1998.
- [15] C.J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 1–32, 1998.
- [16] C. C. Gaudes, J. Via, and I. Santamaría, "Robust array beamforming with sidelobe control using support vector machines," in *IEEE 5th Workshop on Signal Processing Advances in Wireless Communications*, jul 2004, pp. 258–262.
- [17] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," Technical Report 44, Max Planck Institut für biologische Kybernetik, Tübingen, Germany, dec 1996.
- [18] J. Platt, *Advances in Kernel Methods: Support Vector Learning (B. Schölkopf, C. J. C. Burges and A. J. Smola, Eds.)*, chapter Fast Training of Support Vector Machines Using Sequential Minimal Optimization, pp. 185–208, MIT Press, 1999.
- [19] J. T. Kwok and I. W. Tsang, "Linear dependency between ϵ and the input noise in ϵ -support vector regression," *IEEE Transactions in Neural Networks*, vol. 14, no. 3, pp. 544–553, May 2003.
- [20] V. Cherkassky and Y. Ma, "Practical Selection of SVM Parameters and Noise Estimation for SVM Regression," *Neural Networks*, vol. 17, no. 1, pp. 113–126, Jan 2004.



Manel Martínez-Ramón received his Degree in Telecommunications Engineering in 1994 (Universitat Politècnica de Catalunya, Spain), and finished his PhD, also in Telecommunications Engineering in 1999 (Universidad Carlos III de Madrid, Spain). He is with the Department of Signal Th. and Comms., Universidad Carlos III de Madrid, Spain. His research topics are in applications of the statistical learning to signal processing, with emphasis in communications and brain imaging. He has coauthored 14 papers in international journals and 30 conference papers on these topics. He has written a book on applications of SVMs to antennas and electromagnetics and coauthored several book chapters. He is a senior member of the IEEE since 2004.



Christos G. Christodoulou Received his Ph.D. degree in Electrical Engineering from North Carolina State University in 1985. He served as a faculty member in the University of Central Florida, Orlando, from 1985 to 1998. In 1999, he joined the faculty of the Electrical and Computer Engineering Department of the University of New Mexico, where he served as the Chair of the Department from 1999 to 2005. He is a Fellow member of IEEE and he served as the general Chair of the IEEE Antennas and Propagation Society/URSI 1999 Symposium in Orlando, Florida, as the co-chair of the IEEE 2000 Symposium on Antennas and Propagation for wireless communications, in Waltham, MA, and the co-technical chair for the IEEE Antennas and Propagation Society/URSI 2006 Symposium in Albuquerque. Currently, he is an associate editor for the IEEE Transactions on Antennas and Propagation, the International Journal of RF and Microwave Computer-aided Engineering, and the IEEE Antennas and Propagation Magazine. He served as a guest editor for a special issue on "Applications of Neural Networks in Electromagnetics" in the Applied Computational Electromagnetics Society(ACES) journal. He has published over 250 papers in journals, conferences, and book chapters. He has also co-authored 4 books. His research interests are in the areas of modeling of electromagnetic systems, reconfigurable systems, machine learning applications in electromagnetics, and smart antennas.



José Luis Rojo-Álvarez (M'01) received the Telecommunication Engineer degree from University of Vigo, Spain, in 1996 and the Ph.D. degree in Telecommunication from the Polytechnical University of Madrid, Spain, in 2000. He is an Associate Professor with the Department of Signal Th. and Comms., University Rey Juan Carlos, Madrid. His main research interests include statistical learning theory, digital signal processing, and complex system modeling, with applications both to cardiac signal and image processing, and to digital communications. He has published more than 20 papers and more than 50 international conference communications, on support vector machines and neural networks, robust analysis of time series and images, cardiac arrhythmia mechanisms, and Doppler echocardiographic image for hemodynamic function evaluation.



Gustavo Camps-Valls (M'04) was born in València, Spain in 1972, and received a B.Sc. degree in Physics (1996), a B.Sc. degree in Electronics Engineering (1998), and a PhD degree in Physics (2002) from the Universitat de València. He is currently an assistant professor in the Dept. of Electronics Eng. at the Universitat de València, where teaches electronics, advanced time series processing, and digital signal processing. His research interests are neural networks and kernel methods for hyperspectral data analysis, health sciences, and safety-related areas. He is the author (or co-author) of 30 journal papers, several book chapters, and more than 50 international conference papers. He is a referee of several international journals and has served on the Scientific Committees of several international conferences.