

Approximate Kernel Orthogonalization for Antenna Array Processing

Angel Navia-Vázquez, *Senior Member, IEEE*, Manel Martínez-Ramón, *Senior Member, IEEE*,
Luis Enrique García-Muñoz, *Member, IEEE*, and Christos G. Christodoulou, *Fellow, IEEE*

Abstract—We present a method for kernel antenna array processing using Gaussian kernels as basis functions. The method first identifies the data clusters by using a modified sparse greedy matrix approximation. Then, the algorithm performs model reduction in order to try to reduce the final size of the beamformer. The method is tested with simulations that include two arrays made of two and seven printed half wavelength thick dipoles, in scenarios with 4 and 5 users coming from different angles of arrival. The antenna parameters are simulated for all DOAs, and include the dipole radiation pattern and the mutual coupling effects of the array. The method is compared with other state-of-the-art nonlinear processing methods, to show that the presented algorithm has near optimal capabilities together with a low computational burden.

Index Terms—Antenna array processing, beamforming, DOA, kernel method, support vector machines.

I. INTRODUCTION

IT is well known that linear single user signal detection models are suboptimal in most array processing scenarios, since the optimal solution usually takes a nonlinear form (see e.g., [1], the classical survey [2] or the application to the multipath problem in array processing in [3]). Moreover, when the number of sensors is less than the number of users a linear model does not work at all, in this case the array cannot cancel all the unwanted users' signals to proceed with the detection of the signal from the desired one.

Since their introduction, machine learning algorithms have been used for signal processing applications. These algorithms are usually nonlinear in nature, which makes them more powerful than the linear ones as they can adapt better to the statistical properties of the given data. This feature decreases estimation errors or bit error rates. Neural networks [4] have been proposed for beamforming (e.g., [5]–[7]) and direction of arrival estimation (e.g., [8], [9]) among other array processing tasks. A com-

prehensive compilation of the use of neural networks in antenna array processing can be found in [10].

Nevertheless, machine learning algorithms like neural networks suffer from serious drawbacks such as overfitting or local minima, which leads to suboptimal solutions. These algorithms may also have a large number of free parameters to adjust, which often precludes its use in some signal processing applications.

Kernel methods are a paradigm for pattern analysis that arise from Mercer's theorem [11], [12]. They provide straightforward nonlinear extensions of well established linear methods. Kernel methods use a nonlinear mapping of the data into a higher dimensional Hilbert space (also called feature space), where one can use any linear algorithm that can be expressed in terms of dot products. When the solution is mapped back to the input space we obtain a nonlinear version of that algorithm, while other interesting properties are preserved: low number of free parameters, complexity control, or existence and uniqueness of the solution. The interested reader can refer to [13] for a good and comprehensive introduction to kernel methods.

In the past few years a number of linear algorithms have been *kernelized* to provide them with nonlinear properties, namely, Fisher's linear discriminant analysis (LDA) [14], partial least squares (PLS) [15], ridge regression (RR), principal component analysis (PCA) [16], spectral clustering (SC) [17], canonical correlation analysis (CCA) [18], independent component analysis (ICA) [19], ARMA models [20] or the gamma filter [21], among many other. Support vector machines (SVM) [22], both linear and kernelized, have become very popular due to their good performance in many real world problems.

Kernel methods have been used in antenna processing, because their nonlinear nature allows the user to detect signals in environments in which there are more users than array elements. Most of these applications are based on SVM classifiers [23]–[27] and SVM regression machines [28]–[33], both presenting important advantages. They are robust against non-gaussian noise in general and against high amplitude (outlier) samples in particular. When the number of available samples for training is small, the complexity control properties of SVMs minimize the risk of producing solutions biased from the optimal. That is, they minimize the overfitting phenomena. Moreover, SVMs can produce sparse solutions, which provide low complexity estimators. However, they are computationally costly because they deal with matrices with dimensions equal to the number of data (batch processing) and it is not possible to introduce prior information about the data when it is available, as in array processing for communications. In [34] the optimal solution for nonlinear beamforming is derived, and an asymptotic

A. Navia-Vázquez, M. Martínez-Ramón and L. E. García-Muñoz are with the Department of Signal Theory and Communications, Universidad Carlos III de Madrid, 28911 Leganés, Madrid, Spain, (e-mail: manel@tsc.uc3m.es; navia@tsc.uc3m.es).

C. G. Christodoulou is with the Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM 87131-1356 USA (e-mail: christos@ece.unm.edu).

totically kernel-based adaptive beamformer is proposed using an orthogonal forward selection procedure. Most of these algorithms are designed for real, binary signals, and the extension to general QAM or FSK signals is not straightforward.

In a beamforming scenario when the signal to noise ratio is high enough, data takes the form of Gaussian clusters with central symmetry, whose properties directly depend on the transmitted constellation. These characteristics can easily be exploited to construct a near optimal, computationally efficient algorithm based on kernels. In the present work we propose a method that first identifies the clusters by using a modified sparse greedy matrix approximation [13], [35], [36], which is an efficient incremental approximation to a kernel principal component analysis. Second, the algorithm performs model reduction in order to try to reduce the final size of the beamformer.

Here, we test the algorithm in two realistic environments. In the first case, we use two half wavelength thick dipoles at 3 GHz and 4 users using BPSK symbols, in order to compare the performance of the algorithm with other approaches. In the second case, we use 7 of such dipoles to form a linear array to detect the QPSK signals from four users. In all cases, coupling effects are considered.

II. PROBLEM STATEMENT

Let an M element array antenna receive signals from L users (the number of users is assumed not to change) with L unique directions of arrival (DOA) θ_k , $0 \leq k \leq L - 1$ (as shown in Fig. 1). The matrix form of 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, $\mathbf{n}[n]$ stands for the additive white Gaussian noise (AWGN) vector and \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

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

where λ is the wavelength, and d represents the distance between elements of the array.

Assume that the desired signal is element $s_1[n] = d[n]$ of the vector of incoming signals. A linear estimator for this signal can be expressed as

$$\hat{d}[n] = \mathbf{w}^H \mathbf{x}[n]. \quad (4)$$

This estimator must minimize the influence of the interference signals, which can be achieved by placing transmission zeros in the DOAs of these signals. Clearly, the number of zeros that can be placed is equal to $L - 1$, so when the total number

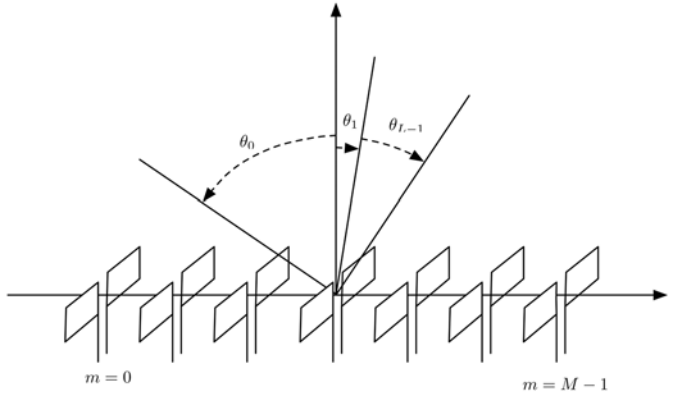


Fig. 1. Linear array of M elements with L incoming signals.

of signals is greater than the number of elements, then the estimator is not guaranteed to work properly because part of the interference will not be rejected by the array.

This situation can be modified by nonlinearly mapping the input data $\mathbf{x}[n]$ into a higher dimension Hilbert space, where the number of available dimensions is much higher than the number of interferers.

Parameter vector \mathbf{w} will also have a higher or infinite dimension, which, in principle, will increase the computational burden to an overwhelming amount. This problem is addressed as the “curse of dimensionality.” It, nevertheless, can be avoided by virtue of two facts. The first one is that some Hilbert spaces are provided with a dot product that can be expressed as a function of the input space (Kernels), this is $\varphi(u)^T \varphi(v) = k(u, v)$. These are the so-called reproducing Kernel Hilbert spaces (RKHS). The second one is that as the parameter vector lies in the subspace spanned by the given data, it can be expressed as a linear combination of these data.

These facts, whose application to nonlinear estimation is often called the *Kernel Trick*, give rise to the class of Kernel estimators. They have nonlinear properties, but the optimization is linear with respect to the parameters and its number is, at most, equal to the number of data available for training. They are developed in Section III.

III. ESTIMATORS IN REPRODUCING KERNEL HILBERT SUBSPACES

The class of linear algorithms that can be expressed as a linear combination of dot products between data can be nonlinearized through the Kernel Trick. The basic idea is to have a nonlinear transformation $\varphi(\cdot)$ of the data $\mathbf{x}[n]$ into a higher (possibly infinite) dimension Hilbert space for which the associated dot product is expressible as a function $k(\cdot, \cdot)$ of the input data, i.e.,

$$\varphi(\mathbf{x}[i])^T \varphi(\mathbf{x}[j]) = k(\mathbf{x}[i], \mathbf{x}[j]). \quad (5)$$

Such a function is called a Mercer Kernel. A Hilbert Space provided with a kernel is a RKHS. The Mercer theorem [11] states the conditions for a Kernel to be a dot product in a Hilbert Space. In particular, it says that a mapping function $\varphi(\cdot) : \mathbb{R}^n \mapsto \mathcal{H}$ and a function $k(\cdot, \cdot)$ as in (5) exist if and only

if $k(\cdot, \cdot)$ is an integral operator in a Hilbert Space, i.e., if the kernel satisfies that

$$\int g(\mathbf{x}_i)k(\mathbf{x}_i, \mathbf{x}_j)g(\mathbf{x}_j)d\mathbf{x}_i d\mathbf{x}_j \geq 0 \quad (6)$$

for any integrable squared function $g(\cdot)$. Following these concepts, a nonlinear estimator in the input space can be formulated as a linear one in \mathcal{H} :

$$\hat{d}[n] = \mathbf{w}^T \varphi(\mathbf{x}[n]) + b \quad (7)$$

where \mathbf{w} is the linear parameter set (whose dimension is now assumed to be greater than M) and $e[n] = d[n] - \hat{d}[n]$ is the error between the estimated and the desired outputs. Constant b is a scalar bias that needs to be added since the nonlinear estimator may be biased. Parameter vector \mathbf{w} always lies in the subspace spanned by the training data and, according to the representer theorem [37], it can be constructed as a linear combination of the given data

$$\mathbf{w} = \sum_i \alpha_i \varphi(\mathbf{x}[i]) \quad (8)$$

where $(\mathbf{x}[i], d[i])$ are the training data pairs. Then, estimator (7) can be rewritten as

$$\hat{d}[n] = \sum_i \alpha_i \varphi(\mathbf{x}[i])^T \varphi(\mathbf{x}[n]) + b. \quad (9)$$

In this context, \mathbf{w} 's are called primal parameters, and α_i 's are the dual ones. The problem of optimizing the estimator expressed as in (9) is called a dual problem, and it is useful for those cases where the primal problem is unsolvable in a direct way. As an example, defining $\Phi = \{\varphi(\mathbf{x}[1]) \dots \varphi(\mathbf{x}[N])\}$, the MMSE criterion to find \mathbf{w} from (7) will lead to the well known Wiener-Hopf solution $\mathbf{w}_{opt} = \mathbf{R}^{-1} \mathbf{p}$ where the autocorrelation matrix \mathbf{R} can be computed as $\mathbf{R} = \Phi \Phi^H$. Since vectors $\varphi(\mathbf{x}[n])$ have infinite dimension, then \mathbf{R} will also have infinite dimension, so its inverse does not exist, and the primal problem cannot be directly solved.

Nevertheless, the dual problem to find vector $\boldsymbol{\alpha}$ containing all parameters α_i using (9) and the MMSE criterion leads to the solution $\boldsymbol{\alpha} = \mathbf{K}^{-1} \mathbf{d}$, where $\mathbf{K} = \Phi^H \Phi$ is the matrix containing all dot products between data (Gramm matrix), whose dimension will be $N \times N$.

However, \mathbf{K} is usually rank-deficient and finding its inverse is still an ill-conditioned problem. In what follows we will describe how to proceed in the nonlinear beamforming particular case to obtain an efficient solution to this problem. We aim at building compact representations of patterns $\varphi(\mathbf{x}[n])$ in the feature space \mathcal{H} , such that a reduced complexity solution is possible, benefiting both computational cost and algorithmic stability.

In the beamforming case, the geometry of patterns $\mathbf{x}[n]$ in the input space usually takes the form of a constellation of Gaussian clusters. For instance, in the BPSK case, the constellation at the

transmitter has two Gaussian clusters, one for every different symbol. At the receiver, when signals from L users are present, we have a total of 2^L Gaussian clusters (S^L for the general case of L users and S different symbols). For the system to be usable, the signal to noise ratio (SNR) must be high enough such that we obtain bit error rates (BER) as low as 10^{-4} or 10^{-6} . In these situations, the Gaussian clusters are clearly separated in the input space, which induces a particular geometry in \mathcal{H} .

An optimal approach to reduce the complexity of the data, nonlinearly projected in \mathcal{H} , is to apply kernel principal component analysis (KPCA), identify the main directions (eigenvectors) and project patterns $\varphi(\mathbf{x}[n])$ along these main directions. These principal directions in \mathcal{H} have a direct interpretation as clusters in the input space [38]. However, we have to solve the so-called preimage problem, i.e., finding patterns in the input space whose projections are the best approximations to these principal directions in \mathcal{H} . Given the particular geometry of our problem, such procedure would identify S^L main directions, each one associated to a cluster of data in the input space. Applying KPCA techniques in a real-time environment is not feasible due to computational cost, and more efficient approximations are needed which do not need to compute preimages.

The objective is to identify an orthogonal base in \mathcal{H} that approximates patterns $\varphi(\mathbf{x}[n])_{n=1}^N$ with minimal least squares error. Efficient approximate methods to find such base exist, such as the sparse greedy matrix approximation method (SGMA) [13], [35], [36]. SGMA proposes to operate incrementally, selecting one by one the elements to be added to the base, such that every new added element maximally reduces the approximation error of the patterns. Therefore, every projected pattern $\varphi(\mathbf{x}[n])$ is approximated as a function of R base elements, where $\mathbf{z}[j]$ are vectors in \mathbb{R}^n selected using the procedure described at the end of this section

$$\varphi(\mathbf{x}[i]) = \sum_{j=1}^R a_{i,j} \varphi(\mathbf{z}[j]) \quad (10)$$

where the optimal weights $\mathbf{a}_i = [a_{i,1}, a_{i,2}, \dots, a_{i,R}]^T$ are obtained as

$$\mathbf{a}_i = \mathbf{K}_{zz}^{-1} \mathbf{k}_i \quad (11)$$

where $(\mathbf{K}_{zz})_{i,j} = k(\mathbf{z}[i], \mathbf{z}[j])$ and $(\mathbf{k}_i)_j = k(\mathbf{x}[i], \mathbf{z}[j])$. The $\varphi(\mathbf{z}[j])$ elements are selected such that we minimize the approximation error E_R at stage R , averaged for all patterns

$$E_R = \sum_{n=1}^N \left\| \varphi(\mathbf{x}[n]) - \sum_{i=1}^R a_{i,j} \varphi(\mathbf{z}[j]) \right\|^2. \quad (12)$$

By directly working with projections of already existing input patterns we avoid computing pre-images, unavailable in a real time environment, since their computation, as analyzed in [39], [40], is rather cumbersome and prone to fall in local minima. To identify the $\mathbf{z}[j]$ in (12), SGMA proposes to proceed greedily to identify the new base element to be added at every stage. The

error reduction if we add candidate $\mathbf{x}[m]$ to the already existing base is

$$\Delta E_m = \frac{(\mathbf{K}_{nz}\mathbf{K}_{zz}^{-1}\mathbf{k}_{mz} - \mathbf{k}_{mx})^T (\mathbf{K}_{nz}\mathbf{K}_{zz}^{-1}\mathbf{k}_{mz} - \mathbf{k}_{mx})}{(1 - \mathbf{k}_m^T \mathbf{K}_{zz}^{-1} \mathbf{k}_{mz})} \quad (13)$$

where $(\mathbf{k}_{mz})_i = k(\mathbf{x}[m], \mathbf{z}[i])$, $(\mathbf{k}_{mx})_i = k(\mathbf{x}[m], \mathbf{x}[i])$ and $(\mathbf{K}_{nz})_{i,j} = k(\mathbf{x}[i], \mathbf{z}[j])$. It is very inefficient to evaluate all the patterns in the training set as candidates, so a random sub-sample is selected at every stage. A total of 59 candidate patterns,¹ randomly selected among the training patterns, are evaluated at every stage and the one with the largest ΔE_m value is added to the base. In our case we have observed that the first candidates tend to be selected as close as possible as the center of every Gaussian cluster and, furthermore, since the Gaussians are well separated, $\varphi(\mathbf{z}[j])$ tend to form an orthonormal base or, equivalently, $\mathbf{K}_{zz} \simeq \mathbf{I}$, which greatly simplifies the SGMA operation. More precisely, at the beginning of the procedure, when the base is empty, the selection of the first element reduces to finding the candidate with the largest ΔE_m computed as

$$\Delta E_m = \mathbf{k}_{mx}^T \mathbf{k}_{mz}. \quad (14)$$

If we assume that $\mathbf{K}_{zz} \simeq \mathbf{I}$, we can proceed in every step independently of the other base elements. This approximated version of SGMA has been named as approximate kernel orthogonalization (AKO), and is valid for situations where clusters of data are well separated in the input space, as in our case. Once the first element has been added, we identify those patterns that are mainly projected onto that component by measuring the angle $\theta(\varphi(\mathbf{x}[n]), \varphi(\mathbf{z}[j]))$ between every projected pattern and the newly added base element

$$\cos(\theta(\varphi(\mathbf{x}[n]), \varphi(\mathbf{z}[j]))) = \frac{\varphi(\mathbf{x}[n])^T \varphi(\mathbf{z}[j])}{\|\varphi(\mathbf{x}[n])\| \cdot \|\varphi(\mathbf{z}[j])\|} \quad (15)$$

that reduces to a simple computation when Gaussian kernels are used

$$\begin{aligned} \cos(\theta(\varphi(\mathbf{x}[n]), \varphi(\mathbf{z}[j]))) &= \frac{k(\mathbf{x}[n], \mathbf{z}[j])}{\sqrt{k(\mathbf{x}[n], \mathbf{x}[n])k(\mathbf{z}[j], \mathbf{z}[j])}} \\ &= k(\mathbf{x}[n], \mathbf{z}[j]). \end{aligned} \quad (16)$$

We identify those patterns $\varphi(\mathbf{x}[n])$ ($n \in \mathcal{N}$) closer to $\varphi(\mathbf{z}[j])$, i.e., those $\varphi(\mathbf{x}[n])$ such that $\cos(\theta(\varphi(\mathbf{x}[n]), \varphi(\mathbf{z}[j]))) > \eta$ and we remove them from the training set. We repeat the candidate evaluation process using (14) and the removal of closer patterns as identified in (16) until the training dataset is empty.

¹The number of 59 candidates may seem strange, but the authors in [13], [36] have shown that, "to obtain an estimate that with probability 0.95 is among the best 0.05 of all estimates, the random subsample size of $\lceil \log 0.05 / \log 0.95 \rceil = 59$ will guarantee nearly as good performance as using the whole dataset."

To further improve the model, we redefine every base element as

$$\hat{\varphi}(\mathbf{z}[j]) = \frac{1}{N} \sum_{n \in \mathcal{N}} \varphi(\mathbf{x}[n]). \quad (17)$$

Once again, to avoid working with pre-images, we propose to approximate (17) by

$$\hat{\varphi}(\mathbf{z}[j]) \simeq \varphi\left(\frac{1}{N} \sum_{n \in \mathcal{N}} \mathbf{x}[n]\right). \quad (18)$$

Associated to base element $\hat{\varphi}(\mathbf{z}[j])$ we also annotate the symbol label $l[i]$ most frequently observed in the patterns in \mathcal{N} . Once the base elements are found, the model in (9) reduces to

$$\hat{d}[n] = \sum_{i=1}^R \beta_i \varphi(\mathbf{z}[i])^T \varphi(\mathbf{x}[n]) + b \quad (19)$$

where R is the base size and the weights β can now be computed² as

$$\beta = (\mathbf{K}_{xz}^T \mathbf{K}_{xz})^{-1} \mathbf{K}_{xz}^T \mathbf{d} \quad (20)$$

where $(\mathbf{K}_{xz})_{i,j} = k(\mathbf{x}[i], \mathbf{z}[j])$. Since this is a quadratic form, it can be said that the property of existence and uniqueness of solutions holds here. We can further simplify this computation by noting that $(\mathbf{K}_{xz}^T \mathbf{K}_{xz}) \simeq \mathbf{I}$ (apart from a scale factor), due to the implicit orthonormalization, and $\mathbf{K}_{xz}^T \mathbf{d} \simeq \mathbf{l}$ (apart from a scale factor), $(\mathbf{l})_i = l[i]$, the labels identified during the base construction. Therefore, it is not necessary to waste computational effort in estimating the weights of the model, we can simply use the labels stored in \mathbf{l} to decide which symbol was transmitted

$$\hat{s}[n] = l[i]; i = \operatorname{argmax}_i \{k(\mathbf{z}[i], \mathbf{x}[n])\}. \quad (21)$$

To initially evaluate the approximations assumed so far we have computed several quality measurements over the original and approximated kernel matrices using both SGMA and the proposed AKO method. We have computed the error in the approximation of the Gramm Matrix $\|\mathbf{K} - \mathbf{A}\mathbf{K}_{xz}^T\|^2$, where $(\mathbf{K}_{xz})_{(i,j)} = k(\mathbf{x}[i], \mathbf{z}[j])$, and $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N]^T$, where every \mathbf{a}_i is computed used (11). We have used the following conditions to generate the data: 2 sensors, a BPSK modulation, 15 frames and 6 users. In Fig. 2(a) we depict such approximation error. It can be observed that the AKO method produces less approximation error than SGMA approach for the same base size and for a wide range of SNR values in this particular scenario. We also observe in Fig. 2(b) that the CPU time for AKO is also smaller than SGMA, roughly half. Analogous results have been obtained for other beamforming scenarios with varying number of sensors, users and modulations.

We also explore the validity of the proposed approximation $\mathbf{K}_{xz}^T \mathbf{K}_{xz} \simeq \mathbf{I}$ in (20) for both AKO and SGMA. In Fig. 3 we

²For symmetry reasons, we will assume in what follows $b = 0$.

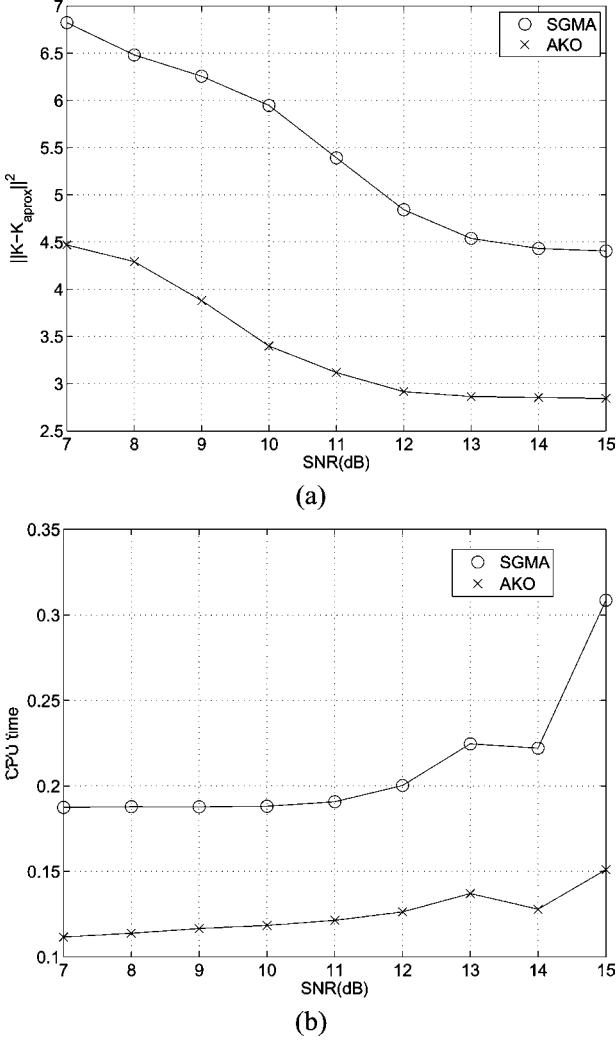


Fig. 2. Approximation error (a) and CPU time (b) for AKO and SGMA methods. Data corresponds to 2 sensors, a BPSK modulation, 15 frames, SNR = 15 dB and 6 users.

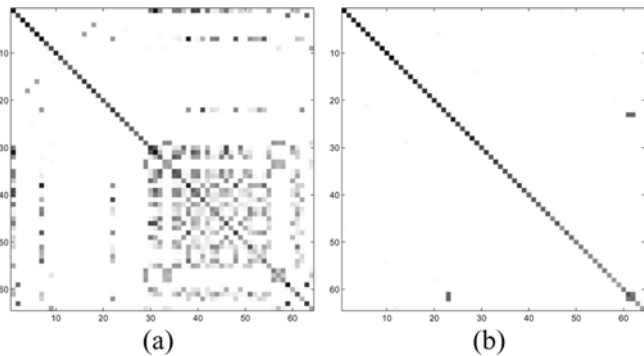


Fig. 3. Representation of matrix $\mathbf{K}_{zz}^T \mathbf{K}_{zz}$ (ideally equal to the identity matrix) for both SGMA (a) and AKO (b).

depict as an image the \mathbf{K}_{zz} matrix. We observe that the base elements selected using AKO (Fig. 3(b)) are much closer to achieving the approximation than those selected using SGMA (Fig. 3(a)).

TABLE I
AKO ALGORITHM

- **Step 1:** Process separately training patterns associated to every symbol class s_i :
 - **Step 1.0:** Select at random 59 patterns among the N patterns in the training set with label s_i : $\mathbf{x}[m]$, $m = 1, \dots, 59$
 - **Step 1.1:** Build the candidate kernel vectors:
 - \mathbf{k}_{mx} ; $(\mathbf{k}_{mx})_n = k(\mathbf{x}[m], \mathbf{x}[n])$
 - $m = 1, \dots, 59$; $n = 1, \dots, N$
 - **Step 1.2:** Compute error decrements and select the next base element $\mathbf{z}[r]$ as the \mathbf{x}_m with the largest error decrement: $\Delta E_m = \mathbf{k}_{mx}^T \mathbf{k}_{mx}$
 - **Step 1.3:** Update the training set by removing those patterns closer to the new base element, i.e., those that satisfy $\cos(\theta(\varphi(\mathbf{x}[n]), \varphi(\mathbf{z}[j]))) = k(\mathbf{x}[n], \mathbf{z}[j]) > \nu$
 - **Step 1.4:** Go back to step **Step 1.0** until there are no patterns left in the training set.
 - **Step 1.5:** Redefine the base elements: $\mathbf{z}[j] = \frac{1}{N} \sum_{n \in \mathcal{N}} \mathbf{x}[n]$, $\forall \mathbf{x}[n]$ s.t. $\cos(\theta(\varphi(\mathbf{x}[n]), \varphi(\mathbf{z}[j]))) = k(\mathbf{x}[n], \mathbf{z}[j]) > \eta$ (repeat 3 times)
 - **Step 1.6:** Build the classifier as $\hat{s}[n] = l[i]; i = \text{argmax}_i \{k(\mathbf{z}[i], \mathbf{x}[n])\}$
- **Step 2:** Model reduction (optional):
 - **Step 2.1:** Generate synthetic patterns $\tilde{\mathbf{x}}_{i,j,k}$
 $\tilde{\mathbf{x}}_{i,j,1} = (\eta)\mathbf{z}[i] + (1-\eta)\mathbf{z}[j]$ and $\tilde{\mathbf{x}}_{i,j,2} = (1-\eta)\mathbf{z}[i] + (\eta)\mathbf{z}[j]$, $\forall i, j$ s.t. $s_i \neq s_j$
 - **Step 2.2:** Classify them to obtain the synthetic labels $\tilde{s}_{i,j,k}$
 - **Step 2.3:** Evaluate new labels $\tilde{s}'_{i,j,k}$ when element $\mathbf{z}[i]$ is removed from the model.
 - **Step 2.4:** Prune element $\mathbf{z}[i]$ from the model if $\tilde{s}_{i,j,k} = \tilde{s}'_{i,j,k}$.
 - Repeat Steps 2.3-2.4 for all $\mathbf{z}[i]$ elements.

A. Model Reduction

The drawback of the AKO method as described above is that the resulting machine has size S^L , where L is the number of users and S is the size of the alphabet. For instance, for the QPSK case when 5 users are transmitting, the model is built using 1024 kernels. Such complexity could be unacceptable in some circumstances, forcing us to use smaller models. We propose here a postprocessing stage that will evaluate if it is possible to reduce the model by eliminating any superfluous elements, i.e., those that do not modify the decision boundary when removed from the model. To identify such nodes, we generate first a set of synthetic patterns, laying midway between patterns $\mathbf{z}[i]$ of different class. Next, for every pair $\mathbf{z}[i]$ and $\mathbf{z}[j]$ with different associated class, we generate two new points as $\tilde{\mathbf{x}}_{i,j,1} = (\eta)\mathbf{z}[i] + (1-\eta)\mathbf{z}[j]$ and $\tilde{\mathbf{x}}_{i,j,2} = (1-\eta)\mathbf{z}[i] + (\eta)\mathbf{z}[j]$. Once we have generated all the datapoints $\tilde{\mathbf{x}}_{i,j,k}$, we classify them with the full model to obtain their labels $\tilde{s}_{i,j,k}$. Finally, we evaluate the effect of removing each one of the base elements from the model, by simply classifying again the patterns without the contribution of that element (this computation can be done very efficiently by simply removing it from the computation in (21)). If any of the $\tilde{s}_{i,j,k}$ labels is modified, then the element cannot be removed, otherwise it can be pruned without affecting the model. This criterion could be relaxed, allowing some errors, to further compact the final classifier.

We have summarized in Table I the proposed approximated kernel orthogonalization (AKO) algorithm. In what follows we will benchmark the proposed AKO method using SNR-BER curves in different transmission scenarios against other state-of-the-art methods.

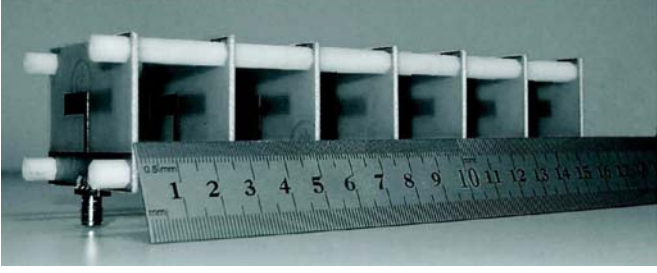


Fig. 4. 7 printed half wavelength thick dipoles array at 3 GHz, built over a FR 4 1.55 mm thick substrate, with spacing 24 mm ($\approx \lambda/4$).

IV. EXPERIMENTS

A. Array Models

Some array processing experiments have been performed using an antenna model array that consists of 2 and 7 printed half wavelength thick dipoles at 3 GHz. These antennas are separately built over a FR 4 1.55 mm thick substrate, with spacing 24 mm ($\approx \lambda/4$) (as shown in Fig. 4).

We compute the excitation voltage V_n of each element, $1 \leq n \leq N$ for 4 different angles of arrival. In order to compute these excitations, we assume that $V_n = V_n^+ + V_n^-$, where V_n^+ is the ideal excitation, and V_n^- is the one produced by the mutual coupling between antenna elements. If $\Gamma_n(\theta_0)$ is the active reflection coefficient, defined by [41]

$$\Gamma_n(\theta_0) = \frac{V_n^-}{V_n^+} = \frac{\sum_{m=1}^N S_{nm} e^{-j(m-1)u_0}}{e^{-j(n-1)u_0}} = \sum_{m=1}^N S_{nm} e^{-j(m-n)u_0} \quad (22)$$

where θ_0 is the direction of arrival, and S_{nm} the scattering parameters of the array, the excitation voltage can be written as

$$V_n = V_n^+ + V_n^- = (1 + \Gamma_n(\theta_0)) \cdot V_n^+. \quad (23)$$

Taking into account that $V_n^+ = V_0 e^{-j(n-1)u_0}$ with $u_0 = kd \sin(\theta_0)$ and using the expression of the electric field produced by the array in transmission

$$E_T(r, \theta) = E_0(r, \theta) \sum_{n=1}^N (1 + \Gamma_n(\theta_0)) e^{j(n-1)u} e^{j(n-1)u_0} \quad (24)$$

and knowing the array factor $F_0(\theta)$, the excitation voltage can be computed. We run 4 FDTD-based simulations using CST Microwave Studio 2008 (www.cst.com) to obtain the active reflection coefficients Γ_n , and then the excitation voltages V_n .

B. Experimental Setup

The aim of the experiments is to evaluate the BER performance of scenarios in which a desired signal is to be detected in

the presence of interferences. We assume that all transmitters are synchronous, which is a common situation in a cell phone environment. The central frequency of the signal is 3 GHz and the transmission rate is 1.77 Kbauds. Data is transmitted in frames containing 146 symbols, including a midamble of 26 training symbols. Under this general experimental setup, we consider the following different scenarios

- 1) Two element array, BPSK, 4 users at $\theta = -45^\circ, -30^\circ, 0^\circ$ and 30° . Desired user with amplitude 0.1, the rest with amplitude 1 (to simulate a near-far situation).
- 2) Two element array, BPSK, 4 users at $\theta = -45^\circ, -30^\circ, 0^\circ$ and 30° , equal amplitudes.
- 3) Two element array, BPSK, 4 users at $\theta = -45^\circ, -30^\circ, 0^\circ$ and 10° , equal amplitudes.
- 4) Two element array, BPSK, 5 users at $\theta = -45^\circ, -30^\circ, 0^\circ, 20^\circ$ and 30° , equal amplitudes.
- 5) Seven element array, QPSK, 4 users at angles of $\theta = -10^\circ, 0^\circ, 10^\circ$ and 20° .

In all cases, the desired user is along $\theta = 0^\circ$. The SNR is measured as the desired signal to thermal noise ratio. To further exploit the symmetries of the problem, we generate extra training points by applying rotations (in the BPSK case, training pairs $(\mathbf{x}[i], d[i])$ are multiplied by -1 to obtain extra training data $(-\mathbf{x}[i], -d[i])$, and in QPSK we obtain a training set four times larger if we multiply by $-1, j$ and $-j$).

C. BER Performance

In order to compare the performances of the SVM, the orthogonal forward selection (OFS) in [34], and the AKO algorithms, we run simulations with the different scenarios described in the previous section, and we measure the BER as a function of the number of received frames. We also measure the BER as a function of the SNR when the models have been trained with the training data contained in 15 frames (390 training symbols) for the BPSK modulations and 25 frames (650 symbols) for the QPSK modulations. In this and Section V, no comparisons between AKO and OFS have been presented for QPSK modulations because a complex version of OFS is not available.

Fig. 5 shows the BER performance of all systems in the scenarios with BPSK modulations (scenarios 1 to 4). The graphic includes the optimum Bayes estimation (theoretical performance limit). As it can be seen, the near-far phenomenon (scenario 1) does not affect the performance of the estimators, that show error rates close to the Bayes optimum. The performance of AKO and OFS are very close, and the SVM approach shows a degradation of about 1 dB.

Fig. 6 contains the performances of a complex version of the SVM and AKO algorithms compared to the Bayes estimation and the linear least squares estimator for scenario 5. Both SVM and AKO systems show a performance close to the optimum, and the differences between AKO and SVM are about 0.5 dB. The uniform linear array is not able to simultaneously fix a maximum at 0° and a minimum in any direction without constraints [42] using any linear procedure. In particular, the minimum possible first-null semi-beamwidth on a seven element array (without accounting for mutual coupling effects) is about 20° .

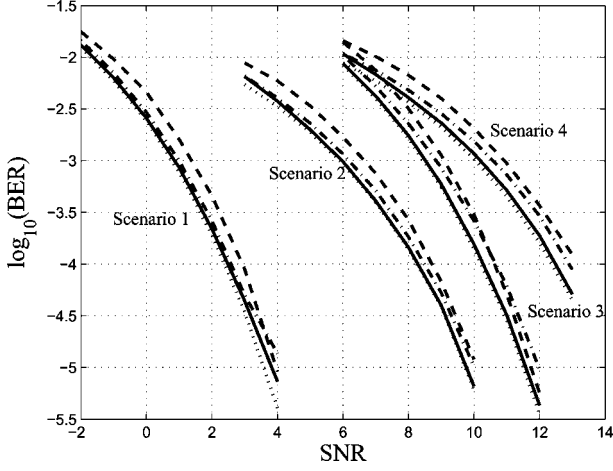


Fig. 5. Experiments with 4 and 5 users, BPSK. Dot: Bayes estimation; Solid: AKO; Dot-dash: OFS; Dash: SVM.

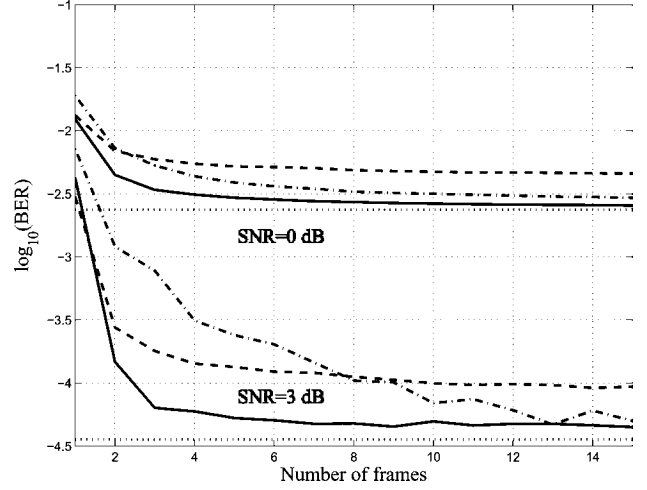


Fig. 7. BER performance (in logarithmic scale) as a function of the number of frames for the scenario 1. Dot: Bayes estimation; Solid: AKO; Dot-dash, OFS; Dash: SVM. Upper graphs produced with SNR = 0 dB, lower graphs with SNR = 3 dB.

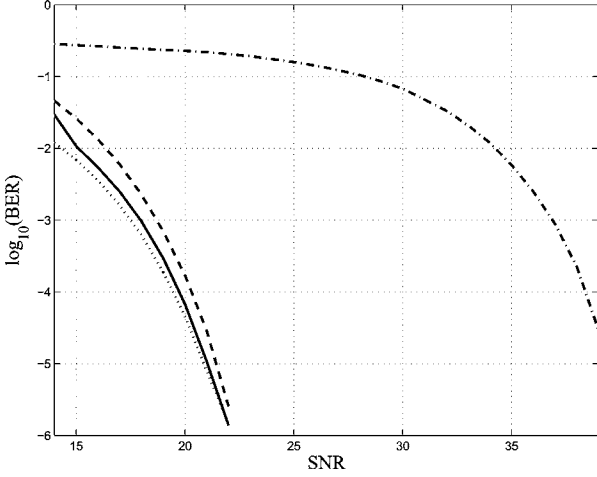


Fig. 6. Experiments with 4 users, QPSK. Dot: Bayes estimation; Solid: AKO; Dash: SVM; Dash-dot: LS.

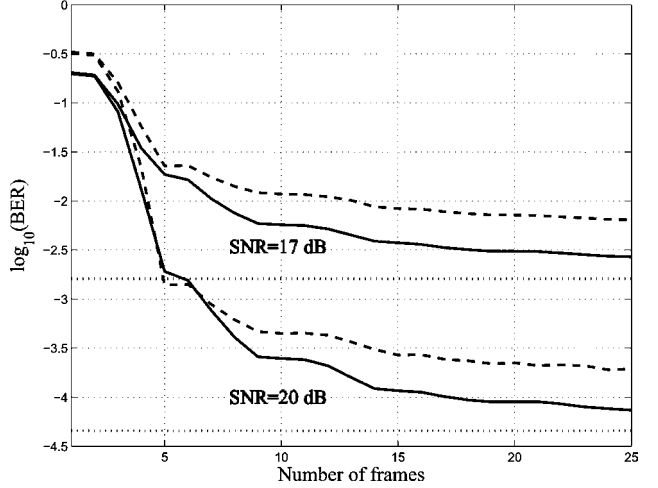


Fig. 8. BER performance (in logarithmic scale) as a function of the number of frames for the scenario 5. Dot: Bayes estimation; Solid: AKO; Dash: SVM. Upper graphs produced with SNR = 17 dB, lower graphs with SNR = 20 dB.

The first advantage of AKO over OFS or SVM can be seen in Fig. 7 (scenario 1) and 8 (scenario 5). In these figures, BER performances as a function of number of frames are shown for two different SNRs (0 and 3 dB in Fig. 7 and 17 and 20 dB in Fig. 8), and for 1 to 16 frames ($N = 26$ to 416 training symbols) in Fig. 7 and 1 to 25 frames ($N = 26$ to 650 training symbols) in Fig. 8).

These graphs show that the AKO method converges to the minimum error with a significantly smaller number of frames than the other methods. AKO needs 4 frames to reach $\text{BER} = 10^{-2.5}$, while OFS needs 10 frames in the SNR = 0 dB case, and in the SNR = 3 dB case, a $\text{BER} = 10^{-4.25}$ is also reached in 4 frames by AKO, while OFS needs 12 frames. SVM never reaches these values.

D. Computational Burden

In order to compare the feasibility of practical implementation of these methods, we measured the average training times over 1000 simulations (Table II) and the average size (number

of kernels) of every resulting model. We chose SNR values for every scenario such that the BERs are about $10^{-2.5}$ and 10^{-5} . Table II shows the superior performance of the AKO over the other methods. The AKO and SVM methods have a training computation time significantly smaller than the OFS method in all simulations. The improvement of the AKO over the SVM method is related to the size of the final model. In BPSK, the AKO method needed between 12 and 22 centroids to construct the model (the theoretical number of centroids is 16). SVM needed between 96 and 415. When tested in a QPSK environment, the AKO method used 256 centroids, which is the minimum number of centroids needed for this scenario (no reduction was possible in this case) while SVM needed more than 5000 centroids to produce acceptable solutions when trained with 10 frames (almost all samples). When trained with

TABLE II
COMPARISON OF TRAINING TIME (IN SECONDS) AND SIZE OF THE RESULTING
MACHINES FOR VARIOUS SNRS

Scenario	SNR	SIZE			Training Time		
		SVM	OFS	AKO	SVM	OFS	AKO
1	0	287.51	16	16	0.050	1.60	0.028
	3	300.33	16	16	0.045	1.60	0.027
2	4	96.97	21.69	12	0.029	2.72	0.027
	9	300.73	21.32	12	0.050	2.70	0.029
3	7	285.53	21.54	14	0.064	1.90	0.029
	11	122.13	21.72	14	0.047	1.95	0.028
4	9	415.50	35.18	22	0.084	2.70	0.063
	13	146.39	34.88	22	0.039	3.23	0.064
5	17	5235.3	—	256	3.02	—	1.04
	20	5242.3	—	256	3.19	—	1.12

20 frames, the computer needed more than one day to solve the SVM optimization.

V. CONCLUSIONS

We present a method for antenna array processing using Gaussian kernels as basis functions. In beamforming scenarios for communications, data is structured in Gaussian clusters. The method first identifies these clusters by using a modified sparse greedy matrix approximation [13], [35], [36], which is an efficient incremental approximation to a kernel principal component analysis. Second, the algorithm performs model reduction in order to try to reduce the final size of the beamformer. The centers of the identified clusters are labelled with their associated desired user symbols, and then a radial basis function classifier is constructed where each RBF output is simply weighted with the label associated to the center of its cluster, thus making unnecessary any parameter training step.

The method is tested in several scenarios with four or five users coming from different DOAs, with BPSK signaling, and using an antenna array that contains two printed thick dipoles. The method is also tested in an environment with four users and QPSK signaling using an array with 7 thick dipoles.

The antenna parameters are simulated for all DOAs, and include the dipole radiation pattern and the mutual coupling effects of the array.

The method is compared to the OFS approach presented in [34] for BPSK tests and to the SVM classifier in both BPSK and QPSK tests. Results show that the presented method is close to the optimum Bayes detector in all environments. The computational burden is reasonable and the estimator size after training is small. The previous methods are almost as close to the optimum as our method, but OFS presents a much higher computational burden and larger models. SVM show a reasonable computational burden—though the SVM LIB [43] software used for comparisons has been largely optimized—but the estimator models contain a much higher number of elements, which could preclude its practical implementation.

The algorithm presented in this paper can be modified to include adaptive properties in order to track changes in the channels, which is part of the future work related to this research.

REFERENCES

- [1] B. Goode, "Synthesis of a nonlinear Bayes detector for Gaussian signal and noise fields using Wiener filters," *IEEE Trans. Inf. Theory*, vol. 13, pp. 116–118, Jan. 1967.
- [2] R. Lucky, "A survey of the communication theory literature: 1968–1973," *IEEE Trans. Inf. Theory*, vol. 19, pp. 725–739, Nov. 1973.
- [3] J. Reilly and S. Haykin, "Nonlinear least squares techniques for solving array antenna processing problems," in *Proc. Antennas Propag. Society Int. Symp.*, Jun. 1980, vol. 18, pp. 362–365.
- [4] S. Haykin, *Neural Networks. A Comprehensive Foundation*. Englewood Cliffs, NJ: Prentice Hall, 1998.
- [5] P.-R. Chang, W.-H. Yang, and K.-K. Chan, "A neural network approach to MVDR beamforming problem," *IEEE Trans. Antennas Propag.*, vol. 40, pp. 313–322, Mar. 1992.
- [6] P. C. J. Hill and P. D. Wells, "Antenna beamforming for EW using adaptive layered networks," in *Proc. Inst. Elec. Eng. Colloq. on Signal Processing in Electronic Warfare*, Jan. 1994, pp. 2/1–2/5.
- [7] A. H. El Zooghby, C. G. Christodoulou, and M. Georgiopoulos, "Neural network-based adaptive beamforming for one- and two-dimensional antenna arrays," *IEEE Trans. Antennas Propag.*, vol. 46, pp. 1891–1893, Dec. 1998.
- [8] H. L. Southall, J. A. Simmers, and T. H. O'Donnell, "Direction finding in phased arrays with a neural network beamformer," *IEEE Trans. Antennas Propag.*, vol. 43, pp. 1369–1374, Dec. 1995.
- [9] A. H. El Zooghby, C. G. Christodoulou, and M. Georgiopoulos, "Performance of radial-basis function networks for direction of arrival estimation with antenna arrays," *IEEE Trans. Antennas Propag.*, vol. 45, pp. 1611–1617, Nov. 1997.
- [10] C. G. Christodoulou and M. Georgiopoulos, *Applications of Neural Networks in Electromagnetics*. Boston, MA: Artech House, 2001.
- [11] M. A. Aizerman, E. M. Braverman, and L. I. Rozoner, "Theoretical foundations of the potential function method in pattern recognition learning," *Auto. Remote Control*, vol. 25, pp. 821–837, 1964.
- [12] K. R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 12, pp. 181–202, Mar. 2001.
- [13] B. Schölkopf and A. Smola, *Learning With Kernels*. Cambridge, MA: MIT Press, 2002.
- [14] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K. R. Müller, "Fisher discriminant analysis with kernels," in *Proc. IEEE Signal Processing Society Workshop on Neural Networks for Signal Processing IX*, Aug. 1999, pp. 41–48.
- [15] M. Momma and K. Bennet, "Kernel partial least squares," in *Proc. Conf. on Learning Theory (COLT 2003)*, 2003, pp. 216–230.
- [16] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Max Planck Institut für biologische Kybernetik, Tübingen, Germany*, 1996, Tec. Rep. 44.
- [17] A. Szymkowiak-Have, M. A. Girolami, and J. Larsen, "Clustering via kernel decomposition," *IEEE Trans. Neural Netw.*, vol. 17, pp. 256–264, Jan. 2006.
- [18] P. L. Lai and C. Fyfe, "Kernel and nonlinear canonical correlation analysis," *Int. J. Neural Syst.*, vol. 10, no. 5, pp. 365–377, 2000.
- [19] F. R. Bach and M. I. Jordan, "Kernel independent component analysis," *J. Machine Learn.*, vol. 3, no. 1, pp. 1–48, Jan. 2003.
- [20] M. Martínez-Ramón, J. L. Rojo-Álvarez, G. Camps-Valls, 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, pp. 1617–1622, Nov. 2006.
- [21] G. Camps-Valls, M. Martínez-Ramón, J. L. Rojo-Álvarez, and E. Soria-Olivas, "Robust [gamma]-filter using support vector machines," *Neurocomputing*, vol. 62, pp. 493–499, 2004.
- [22] V. Vapnik, *Statistical Learning Theory, Adaptive and Learning Systems for Signal Processing, Communications, and Control*. New York: Wiley, 1998.
- [23] I. Jouny, "On svm for classification of real and synthetic radar signatures," in *Proc. IEEE Antennas Propag. Society Int. Symp.*, Washington, DC, 2005, vol. 1B, pp. 2–5.
- [24] N. Xu, C. G. Christodoulou, M. Martínez-Ramón, and T. Ozdemir, "Antenna array processing for radar applications using support vector machine," in *Proc. IEEE Antennas Propag. Society Int. Symp.*, Albuquerque, NM, 2006, pp. 1295–1298.

- [25] Q. Zhao and J. C. Principe, "Support vector machines for SAR automatic target recognition," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 37, no. 2, pp. 643–654, Apr. 2001.
- [26] S. Caorsi, D. Anguita, E. Bermiani, A. Boni, M. Donelli, and A. Massa, "A comparative study of NN and SVM-based electromagnetic inverse scattering approaches to on-line detection of buried objects," *Appl. Computat. Electromagn. Society J.*, vol. 18, no. 2, pp. 1–11, Jul. 2003.
- [27] C. G. Christodoulou, J. A. Rohwer, and C. T. Abdallah, "The use of machine learning in smart antennas," in *Proc. IEEE Antennas Propag. Society Int. Symp.*, Jun. 2004, vol. 1, pp. 321–324.
- [28] R. G. Avestaran and F. Las-Heras, "Support vector regression for the design of array antennas," *IEEE Antennas Wireless Propag. Lett.*, vol. 4, pp. 414–416, 2005.
- [29] M. Pastorino and A. Randazzo, "A smart antenna system for direction of arrival estimation based on a support vector regression," *IEEE Trans. Antennas Propag.*, vol. 53, pp. 2161–2168, 2005.
- [30] R. G. Avestaran, M. F. Campillo, and F. Las-Heras, "Multiple support vector regression for antenna array characterization and synthesis," *IEEE Trans. Antennas Propag.*, vol. 55, pp. 2495–2501, Sep. 2007.
- [31] M. Martínez-Ramón, J. L. Rojo-Álvarez, G. Camps-Valls, and C. G. Christodoulou, "Kernel antenna array processing," *IEEE Trans. Antennas Propag.*, vol. 55, no. 3, pt. I, pp. 642–650, Mar. 2007.
- [32] G. Angiulli, M. Cacciola, and M. Versaci, "Microwave devices and antennas modelling by support vector regression machines," *IEEE Trans. Magn.*, vol. 43, pp. 1589–1592, Apr. 2007.
- [33] A. Randazzo, M. A. Abou-Khousa, M. Pastorino, and R. Zoughi, "Direction of arrival estimation based on support vector regression: Experimental validation and comparison with music," *IEEE Antennas Wireless Propag. Lett.*, vol. 6, pp. 379–382, 2007.
- [34] S. Chen, L. Hanzo, and A. Wolfgang, "Kernel-based nonlinear beamforming construction using orthogonal forward selection with the fisher ratio class separability measure," *IEEE Signal Processing Lett.*, vol. 11, no. 5, pp. 478–481, May 2004.
- [35] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, 1998.
- [36] A. J. Smola and B. Schölkopf, P. Langley, Ed., "Sparse greedy matrix approximation for machine learning," in *Proc. 17th Int. Conf. on Machine Learning*, San Francisco, CA, 2000, pp. 911–918.
- [37] B. Schölkopf, R. Herbrich, and A. J. Smola, "A Generalized Representer Theorem," in *Computational Learning Theory*. Heidelberg: Springer Berlin, 2001, vol. 2111/2001, Lecture Notes in Computer Science, pp. 416–426.
- [38] B. Schölkopf, A. J. Smola, P. Knirsch, and C. Burges, "Fast approximation of support vector kernel expansions, and an interpretation of clustering as approximation in feature spaces," in *Proc. Mustererkennung DAGM-Symp.*, London, U.K., 1998, pp. 125–132.
- [39] G. H. Bakir, J. Weston, and B. Schölkopf, "Learning to find pre-images," *Adv. Neural Inf. Processing Syst.*, vol. 16, no. 7, pp. 449–456, 2004.
- [40] J. T.-Y. Kwok and I. W.-H. Tsang, "The pre-image problem in kernel methods," *IEEE Trans. Neural Netw.*, vol. 15, pp. 1517–1525, Nov. 2004.
- [41] D. M. Pozar, "A relation between the active input impedance and the active element pattern of a phased array," *IEEE Trans. Antennas Propag.*, vol. 51, pp. 2486–2489, Sep. 2003.
- [42] C. A. Balanis, *Antenna Theory. Analysis and Design*. Hoboken, NJ: Wiley-Interscience, 2005.
- [43] C.-C. Chang and C.-J. Lin, LIBSVM: A Library for Support Vector Machines 2001 [Online]. Available: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>



Angel Navia-Vázquez (M'99–SM'03) received the degree in Telecommunications Engineering and the Ph.D. degree in telecommunications engineering from the Universidad Politécnica de Madrid, Spain, in 1992 and 1997, respectively.

He is now an Associate Professor at the Department of Signal Theory and Communications, Universidad Carlos III de Madrid, Spain. His research interests are focused on new architectures and algorithms for nonlinear processing, as well as their application to multimedia processing, communications,

data mining, content management and E-learning. He has (co)authored 16 international refereed journal papers in these areas, several book chapters, more than 40 conference communications, and participated in more than 20 research projects.

Prof. Navia-Vázquez is an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS since January 2004.



Manel Martínez-Ramón (SM'04) received the degree in Telecommunications Engineering from the Universitat Politècnica de Catalunya, Spain, in 1994, and the Ph.D. degree in telecommunications engineering from the Universidad Carlos III de Madrid, Spain, in 1999.

He is with the Department of Signal Theory and Communications, Universidad Carlos III de Madrid. His research topics are in applications of the statistical learning to signal processing, with emphasis in communications and brain imaging. He has coauthored about 20 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.



Luis Enrique García-Muñoz (M'04) is an Associate Professor at the Universidad Carlos III de Madrid, Spain. He has managed or participated in several national and European research projects on areas such as antennas and array design. He has coauthored more than 50 papers in international journals and conferences and holds three patents. His current research interests include terahertz antennas, array design, truncation in antenna arrays and radioastronomy instrumentation.



Christos G. Christodoulou (F'02) received the Ph.D. degree in electrical engineering from North Carolina State University, Raleigh, 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, University of New Mexico, where he served as the Chair of the Electrical and Computer Engineering Department from 1999 to 2005. He has published over 350 papers in journals and conferences, has 12 book chapters and has coauthored four books. His research interests are in the areas of modeling of electromagnetic systems, reconfigurable systems, machine learning applications in electromagnetics, cognitive radio, and smart RF/photonics antennas.

He is a Fellow of the IEEE and a member of Commission B of USNC/URSI and Eta Kappa Nu and the Electromagnetics Academy. He served as the General Chair of the IEEE Antennas and Propagation Society/URSI 1999 Symposium in Orlando, FL, as the Co-Chair of the IEEE 2000 Symposium on Antennas and Propagation for Wireless Communications, 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 Antennas and Wireless Propagation Letters*, the *International Journal of RF and Microwave Computer-aided Engineering*, and the *IEEE Antennas and Propagation Magazine*. He was appointed as an IEEE AP-S Distinguished Lecturer (2007–now) and elected as the President for the Albuquerque IEEE Section in 2008. He served as an Associate Editor for the IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION for six years, was a Guest Editor for the *Electromagnetics in the Applied Computational Electromagnetics Society (ACES) Journal Special Issue on Applications of Neural Networks*, and was a Co-Guest Editor of the IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION Special Issue on Synthesis and Optimization Techniques in Electromagnetics and Antenna System Design (March 2007).