16th European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland, August 25-29, 2008, copyright by EURASIP

ROBUST ADAPTIVE NONLINEAR BEAMFORMING BY KERNELS AND PROJECTION MAPPINGS

Konstantinos Slavakis[1], *Sergios Theodoridis*[2], *and Isao Yamada*[3]

[1] University of Peloponnese, Dept. Telecommunications Science and Technology, Karaiskaki St., Tripoli 22100, Greece. Email: slavakis@uop.gr.

[2] University of Athens, Dept. Informatics and Telecommunications, Ilissia, Athens 15784, Greece. Email: stheodor@di.uoa.gr.

[3] Tokyo Institute of Technology, Dept. Communications and Integrated Systems, S3-60, Tokyo 152-8552, Japan. Email: isao@comm.ss.titech.ac.jp.

ABSTRACT

This paper introduces a novel adaptive nonlinear beamforming design by using the wide frame of Reproducing Kernel Hilbert Spaces (RKHS). The task is cast in the framework of convex optimization. A collection of closed convex constraints is developed that describe: (a) the information dictated by the training data and, (b) the required robustness against steering vector errors. Since a time recursive solution is sought, the task is equivalent with the problem of finding a point, in a Hilbert space, that satisfies an infinite number of closed convex constraints. An algorithm is derived using projection mappings. Numerical results show the increased resolution offered by the proposed approach, even with a few antenna elements, as opposed to the classical Linearly Constrained Minimum Variance (LCMV) beamformer, and to a nonlinear regression approach realized by the Kernel Recursive Least Squares (KRLS) method.

1. INTRODUCTION — PROBLEM FORMULATION

We consider the system of Fig. 1. The *steering vector* associated with a planar wave of wavelength λ, arriving to a Uniform Linear Array (ULA) with an angle $\theta \in [0, \pi]$, called the *Direction Of Arrival (DOA)*, is defined as [1, 2] $\boldsymbol{s} := \boldsymbol{s}(\theta, d, \lambda) := [1, e^{2\pi i \frac{d}{\lambda}\cos\theta}, \ldots, e^{2\pi i (N-1)\frac{d}{\lambda}\cos\theta}]^t \in \mathbb{C}^N$, where $\mathbb{C}$ represents the complex numbers, $i := \sqrt{-1}$, and the superscript $(\cdot)^t$ stands for vector transposition. The steering vector corresponding to the *Signal Of Interest (SOI)* will be denoted by $\boldsymbol{s}_0$, while $\boldsymbol{s}_j$ associates to the j-th jammer, $\forall j \in \overline{1, J}$, for $J \in \mathbb{Z}_{>0}$ (given integers $j_1 \leq j_2$, define $\overline{j_1, j_2} := \{j_1, j_1 + 1, \ldots, j_2\}$).

The signal $(\boldsymbol{r}(k))_{k \in \mathbb{Z}_{\geq 0}}$ (k represents time and $\mathbb{Z}_{\geq 0}$ stands for the nonnegative integers) received by the ULA is given by

$$\boldsymbol{r}(k) := \sum_{j=0}^{J} \alpha_j b_j(k) \boldsymbol{s}_j + \boldsymbol{n}(k), \ \forall k \in \mathbb{Z}_{\geq 0}.$$

The complex scalar random processes $(b_j(k))_{k \in \mathbb{Z}_{>0}}$, $j \in \overline{0, J}$, denote the symbols carried by the SOI and the jammers. To keep the discussion simple, we assume that the BPSK modulation scheme is used so that $b_j(k) = \pm 1$, $\forall k \in \mathbb{Z}_{\geq 0}$, $\forall j \in \overline{0, J}$. The complex vector random process $(\boldsymbol{n}(k))_{k \in \mathbb{Z}_{\geq 0}} \subseteq \mathbb{C}^N$ stands for the additive noise. The complex coefficients $\alpha_j \in \mathbb{C}$, $j \in \overline{0, J}$, comprise a variety of parameters like the signal power, channel attenuation, etc. In order to work with real vectors, a complex vector in $\mathbb{C}^N$ is mapped to a vector in $\mathbb{R}^{2N}$ ($\mathbb{R}$ denotes the set of all real numbers), by using its real $\Re(\cdot)$ and its imaginary $\Im(\cdot)$ part.

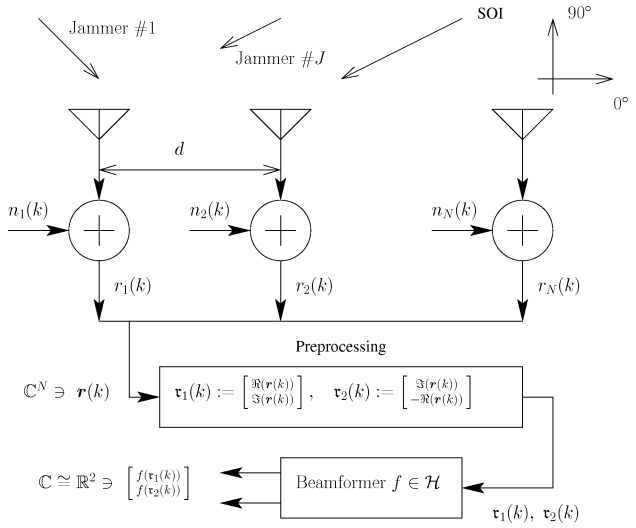

Figure 1: A Uniform Linear Antenna (ULA) of $N \in \mathbb{Z}_{>0}$ elements ($\mathbb{Z}_{>0}$ stands for the positive integers), and with an interelement distance $d > 0$. The beamformer is the function f which belongs to a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$. To work with real vectors, a preprocessing step is applied prior the signal enters the beamformer. To accommodate general complex input signals, the output of the beamformer is a two-dimensional real vector such that its first component corresponds to the real part $\Re(\cdot)$ and the second component to the imaginary part $\Im(\cdot)$ of a complex number.

Here, beamforming is performed by a function $f : \mathbb{R}^{2N} \to \mathbb{R}$. To show that Fig. 1 strictly covers the classical linear beamforming scenario, we consider here the special case where f is linear (more precisely bounded linear). Then, it is well-known that there exists a unique vector $\boldsymbol{u} \in \mathbb{R}^{2N}$ such that $f(\boldsymbol{x}) = \boldsymbol{u}^t \boldsymbol{x}$, $\forall \boldsymbol{x} \in \mathbb{R}^{2N}$ (Riesz representation theorem). If we make the partition $\boldsymbol{u} =: [\boldsymbol{u}_1^t, \boldsymbol{u}_2^t]^t$, where $\boldsymbol{u}_1, \boldsymbol{u}_2 \in \mathbb{R}^N$, and if we define $\boldsymbol{w} := \boldsymbol{u}_1 + i\boldsymbol{u}_2 \in \mathbb{C}^N$, then one can easily see from Fig. 1 that $f(\mathfrak{r}_1(k)) = \boldsymbol{u}^t \mathfrak{r}_1(k) = \boldsymbol{u}^t [\Re(\boldsymbol{r}(k))^t, \Im(\boldsymbol{r}(k))^t]^t = \Re(\boldsymbol{w}^* \boldsymbol{r}(k))$, and similarly $f(\mathfrak{r}_2(k)) = \Im(\boldsymbol{w}^* \boldsymbol{r}(k))$, where the superscript $*$ stands for complex conjugate vector transposition. Hence, the output in Fig. 1 is $(\boldsymbol{w}^* \boldsymbol{r}(k))_k$ which is nothing but the classical linear beamforming scenario [1, 2].

In this paper, a more general beamformer is studied. The function f is assumed to belong to a *Reproducing Kernel Hilbert Space (RKHS)* $\mathcal{H}$ [3] (see Section 2.1), which offers a wide variety of options for f: linear, Gaussian, polynomial, etc. [3].

The beamforming problem to be considered here is as follows. We assume that, due to incorrect a-priori information or poor array calibration, we are given an erroneous knowledge $\tilde{\boldsymbol{s}}_0$ of the SOI steering vector, the received $(\boldsymbol{r}(k))_k$, as well as the training sequence $(b_0(k))_k$. We seek an $f \in \mathcal{H}$ such that the output sequence of the beamformer in Fig. 1 stays close to the training sequence $(b_0(k))_k$ (see Section 3.2), while at the same time beam-pattern and robustness constraints are satisfied (Section 3.1).

Our procedure evolves along three major steps: (a) develop the constraints, (b) form the cost functions, and (c) optimize via projections. The proofs, which clarify various theoretical points of the paper, are omitted due to lack of space.

2. MATHEMATICAL PRELIMINARIES

2.1 Reproducing Kernel Hilbert Space (RKHS).

Henceforth, the symbol $\mathcal{H}$ will stand for a generally infinite dimensional Hilbert space equipped with an inner product denoted by $\langle \cdot, \cdot \rangle$. The induced norm becomes $\|\cdot\| := \langle \cdot, \cdot \rangle^{1/2}$.

Assume a real Hilbert space $\mathcal{H}$ consisting of functions defined on $\mathbb{R}^m$, i.e., $f : \mathbb{R}^m \to \mathbb{R}$, for some $m \in \mathbb{Z}_{>0}$. Let's say $m := 2N$. The function $\kappa(\cdot, \cdot) : \mathbb{R}^{2N} \times \mathbb{R}^{2N} \to \mathbb{R}$ is called a *reproducing kernel* of $\mathcal{H}$ if $\kappa(\boldsymbol{x}, \cdot) \in \mathcal{H}$, $\forall \boldsymbol{x} \in \mathbb{R}^{2N}$, and if $\forall \boldsymbol{x} \in \mathbb{R}^{2N}$ and $\forall f \in \mathcal{H}$, $f(\boldsymbol{x}) = \langle f, \kappa(\boldsymbol{x}, \cdot) \rangle$. In this case, $\mathcal{H}$ is called a *Reproducing Kernel Hilbert Space (RKHS)* [3].

Celebrated examples of reproducing kernels are i) the linear kernel (here the associated RKHS is the space $\mathbb{R}^{2N}$ itself [3]), and ii) the Gaussian kernel $\kappa(\boldsymbol{x}, \boldsymbol{y}) := \exp(-\frac{(\boldsymbol{x}-\boldsymbol{y})^t(\boldsymbol{x}-\boldsymbol{y})}{2\sigma^2})$, $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{2N}$, where $\sigma > 0$ (here the RKHS is of infinite dimension [3]).

2.2 Closed convex sets and metric projection mappings.

A subset C of $\mathcal{H}$ will be called convex if every line segment $\{\lambda f_1 + (1-\lambda) f_2 : \lambda \in [0,1]\}$, with endpoints any $f_1, f_2 \in C$, lies in C.

Given a point $f \in \mathcal{H}$ and a closed convex set $C \subsetneq \mathcal{H}$, a way to move from f to a point in C is by means of the *metric projection mapping P_C onto* C, which is defined as the mapping that takes f to the *uniquely* existing point $P_C(f)$ of C such that $\|f - P_C(f)\| = \inf\{\|f - f'\| : f' \in C\}$ [4].

3. PROPOSED METHOD

3.1 The set of constraints.

We assume the erroneous knowledge $\tilde{\boldsymbol{s}}_0$ of the actual SOI steering vector $\boldsymbol{s}_0$. It is natural to assume that the actual $\boldsymbol{s}_0$ belongs to a neighborhood of $\tilde{\boldsymbol{s}}_0$. An assumption like this is often met in the robust Capon (linear) beamforming scenario [2]. Obviously, such an *ambiguity neighborhood* will be transfered to our working RKHS $\mathcal{H}$.

Following the preprocessing step of Fig. 1, define the vectors $\tilde{\boldsymbol{\mathfrak{s}}}_{01} := [\Re(\tilde{\boldsymbol{s}}_0)^t, \Im(\tilde{\boldsymbol{s}}_0)^t]^t$, $\tilde{\boldsymbol{\mathfrak{s}}}_{02} := [\Im(\tilde{\boldsymbol{s}}_0)^t, -\Re(\tilde{\boldsymbol{s}}_0)^t]^t$. The mapping which transfers data from the space $\mathbb{R}^{2N}$ to $\mathcal{H}$ is $\phi(\tilde{\boldsymbol{\mathfrak{s}}}) := \kappa(\tilde{\boldsymbol{\mathfrak{s}}}, \cdot)$. We assume that all the ambiguity, coming from the erroneous knowledge of the SOI steering vector, can be contained in closed balls with centers located at $\kappa(\pm\tilde{\boldsymbol{\mathfrak{s}}}_{01}, \cdot)$, $\kappa(\pm\tilde{\boldsymbol{\mathfrak{s}}}_{02}, \cdot)$, and with radii fixed to a predefined δ. The $\pm$ signs, which multiply the vectors $\tilde{\boldsymbol{\mathfrak{s}}}_{01}$, $\tilde{\boldsymbol{\mathfrak{s}}}_{02}$, are due to the BPSK symbols.

Our robust approach is to force the beamformer to behave in a uniform way over the introduced ambiguity neighborhoods in $\mathcal{H}$. It should produce an output close to the desired information, i.e., the transmitted by the SOI BPSK symbols. Recall, now, the fundamental reproducing property of the RKHS $\mathcal{H}$, stated in Section 2.1, and easily verify that $f(\tilde{\boldsymbol{\mathfrak{s}}}_{01}) = \langle f, \kappa(\tilde{\boldsymbol{\mathfrak{s}}}_{01}, \cdot) \rangle$ and $f(\tilde{\boldsymbol{\mathfrak{s}}}_{02}) = \langle f, \kappa(\tilde{\boldsymbol{\mathfrak{s}}}_{02}, \cdot) \rangle$. If we denote by h any point inside the ambiguity neighborhoods, we impose the following constraints on the desired beamformer f:

$$\begin{aligned} 1-\epsilon \le \langle f, h \rangle \le 1+\epsilon, \quad \forall h \in B[\kappa(\tilde{\boldsymbol{\mathfrak{s}}}_{01}, \cdot), \delta], \\ -1-\epsilon \le \langle f, h \rangle \le -1+\epsilon, \quad \forall h \in B[\kappa(-\tilde{\boldsymbol{\mathfrak{s}}}_{01}, \cdot), \delta], \\ -\epsilon \le \langle f, h \rangle \le \epsilon, \quad \forall h \in B[\kappa(\tilde{\boldsymbol{\mathfrak{s}}}_{02}, \cdot), \delta], \\ -\epsilon \le \langle f, h \rangle \le \epsilon, \quad \forall h \in B[\kappa(-\tilde{\boldsymbol{\mathfrak{s}}}_{02}, \cdot), \delta], \end{aligned} \tag{1}$$

where $B[h_0, \delta]$ stands for a closed ball with center h_0 and radius some $\delta > 0$. The paramater ϵ is some predefined nonnegative real number. The first two constraints in (1) refer to the real part of the beamformer's output, while the last two constraints refer to the imaginary part.

For example, let us refer to the first equation of (1), and let us consider the special case where the point h, inside that specific ambiguity area, becomes its center, i.e., $h := \kappa(\tilde{\boldsymbol{\mathfrak{s}}}_{01}, \cdot)$. Then the desired beamformer f should produce an output that satisfies $1-\epsilon \le f(\tilde{\boldsymbol{\mathfrak{s}}}_{01}) \le 1+\epsilon$. In other words, whenever the received signal is $\boldsymbol{r}(k) = (+1) \cdot \tilde{\boldsymbol{s}}_0$, our beamformer produces an output whose real part is close to 1 and the imaginary part, by the third equation of (1), is close to 0. Hence, the beamformer produces an output which is close to the transmitted BPSK symbol $[+1, 0]^t$. Constraints like this are required for any value h in the ambiguity area. They are often met in the robust linear beamforming scenarios [2], and are usually called *distortionless constraints*. In a similar way, the second and fourth relation of (1) refer to the beamformer's output whenever the transmitted BPSK symbol is -1.

The constraints in (1) can be written in a compact form as follows:

$$\langle f, h \rangle \le \gamma_m, \ \forall h \in B[h_m, \delta], \forall m \in \overline{1,8}, \tag{2}$$

where the symbols $\{\gamma_m, h_m\}_{m \in \overline{1,8}}$ are appropriately defined. For example, $\gamma_1 := 1+\epsilon$, $h_1 := \kappa(\tilde{\boldsymbol{\mathfrak{s}}}_{01}, \cdot)$, $\gamma_2 := -1+\epsilon$, $h_2 := -\kappa(\tilde{\boldsymbol{\mathfrak{s}}}_{01}, \cdot)$, etc. The following fact gives a clear geometric representation of the above.

Fact 1 *([5, 6]) The following holds: $\hat{f} \in \Gamma := \{f \in \mathcal{H} : \langle f, h \rangle \le \gamma, \ \forall h \in B[h_0, \delta]\}$ if and only if $\exists \hat{\tau} \in \mathbb{R}$ such that $(\hat{f}, \hat{\tau}) \in K \cap \Pi$, where $K := \{(f, \tau) \in \mathcal{H} \times \mathbb{R} : \|f\| \le \tau\}$, and $\Pi := \{(f, \tau) \in \mathcal{H} \times \mathbb{R} : \langle (f, \tau), (h_0, \delta) \rangle = \gamma\}$.*

Both of the sets K and Π are special cases of closed convex sets, with K being an *icecream cone* and Π a *hyperplane* [6]. *In other words, each one of the constraints in* (2) *can be dealt as the intersection of an icecream cone and a hyperplane in the product space $\mathcal{H} \times \mathbb{R}$.* The solution has to lie in the intersection of the K and Π. We will achieve that via projections, whose formulae are given by the following simple rules.

First, we notice that Fact 1 stands for each one of the eight constraints in (2). As such, we construct the Hilbert space $\mathcal{H} \times \mathbb{R}^8$, by the product space of $\mathcal{H}$ and $\mathbb{R}^8$ (for each one of the constraints in (2) add a real component), and by the inner product $\langle (f_1, \boldsymbol{\tau}_1), (f_2, \boldsymbol{\tau}_2) \rangle_{\mathcal{H} \times \mathbb{R}^8} := \langle f_1, f_2 \rangle_{\mathcal{H}} + \boldsymbol{\tau}_1^t \boldsymbol{\tau}_2$, $\forall f_1, f_2 \in \mathcal{H}$, $\forall \boldsymbol{\tau}_1, \boldsymbol{\tau}_2 \in \mathbb{R}^8$. We define also the following sets: the icecream cones and the hyperplanes $\forall m \in \overline{1,8}$,

$$K_m := \{(f, \boldsymbol{\tau}) \in \mathcal{H} \times \mathbb{R}^8 : \|f\| \le \tau_m\}, \tag{3}$$

$$\Pi_m := \{(f, \boldsymbol{\tau}) \in \mathcal{H} \times \mathbb{R}^8 : \langle (f, \boldsymbol{\tau}), (h_m, [\boldsymbol{0}^t, \delta, \boldsymbol{0}^t]^t) \rangle = \gamma_m\}, \tag{4}$$

where $\boldsymbol{\tau} := [\tau_1, \ldots, \tau_m, \ldots, \tau_8]^t \in \mathbb{R}^8$, and δ appears in the m-th position of such a vector in (4). Thus, in order to

achieve robust distortionless constraints, a point $(f,\boldsymbol{\tau})$ that belongs to $\bigcap_{m=1}^{8}(K_m\cap\Pi_m)$ is sought. For all these cases, the metric projection mappings are analytically (and in linear complexity) given by: $\forall(f,\boldsymbol{\tau})\in\mathcal{H}\times\mathbb{R}^8$, $\forall m\in\overline{1,8}$,

$$P_{K_m}((f,\boldsymbol{\tau}))=\begin{cases}(f,\boldsymbol{\tau}), & \text{if } \|f\|\leq\tau_m,\\ (0,[\tau_1,\ldots,\tau_{m-1},0,\\ \quad\tau_{m+1},\ldots,\tau_8]^t), & \text{if } \|f\|\leq-\tau_m,\\ (\frac{\|f\|+\tau_m}{2\|f\|}f,[\tau_1,\ldots,\tau_{m-1},\\ \quad\frac{\|f\|+\tau_m}{2},\tau_{m+1},\ldots,\tau_8]^t), & \text{otherwise,}\end{cases}$$

$$P_{\Pi_m}((f,\boldsymbol{\tau}))=(f,\boldsymbol{\tau})+\frac{\gamma_m-\langle f,h_m\rangle-\tau_m\delta}{\|h_m\|^2+\delta^2}(h_m,[\mathbf{0}^t,\delta,\mathbf{0}^t]^t).$$

These metric projection mappings are used to "push" the solution in $\bigcap_{m=1}^{8}(K_m\cap\Pi_m)$.

3.2 Cost functions: The effect of the training data.

We will use now the information carried by the training sequence $(b_0(k))_k$ and the received signal $(\boldsymbol{r}(k))_k$ to form a sequence of cost objectives. It turns out that this is equivalent to an infinite sequence of certain closed convex constraints. In order to deal with real vectors, we refer to Fig. 1 for the definition of $\mathfrak{r}_1(k)$ and $\mathfrak{r}_2(k)$. Since we adopt the BPSK modulation scheme in this paper, the training symbols take values $b_0(k)\in\{\pm1\}$, $\forall k$. However, to provide with a compact formulation (see (5)) also for more general modulation schemes, where complex symbols are used, we define here $\forall k\in\mathbb{Z}_{\geq0}$, $b_{01}(k):=\Re(b_0(k))$, and $b_{02}(k):=\Im(b_0(k))$.

Our goal is to keep the distance of the beamformer's output to the SOI symbols as small as possible. To bound such distances, we fix an $\hat{\epsilon}$, and we impose the following constraints on the distances defined by the output of Fig. 1 and the SOI symbols: $\{f\in\mathcal{H}:|f(\mathfrak{r}_l(k))-b_{0l}(k)|\leq\hat{\epsilon}\}$, $\forall l\in\overline{1,2}$, $\forall k\in\mathbb{Z}_{\geq0}$.

We have already seen in the previous section that in order to deal with robust distortionless constraints, an augmented Hilbert $\mathcal{H}\times\mathbb{R}^8$ is adopted. Since such a Hilbert space is our stage of discussion, we have to modify the above constraints on the training data accordingly. As such, we notice by the fundamental reproducing property, given in Section 2.1, that $\forall\boldsymbol{\tau}\in\mathbb{R}^8$, $f(\mathfrak{r}_l(k))=\langle f,\kappa(\mathfrak{r}_l(k),\cdot)\rangle=\langle f,\kappa(\mathfrak{r}_l(k),\cdot)\rangle+\boldsymbol{\tau}^t\mathbf{0}=\langle(f,\boldsymbol{\tau}),(\kappa(\mathfrak{r}_l(k),\cdot),\mathbf{0})\rangle_{\mathcal{H}\times\mathbb{R}^8}$. Thus, we obtain a sequence of closed convex constraints $\forall l\in\overline{1,2}$, $\forall k\in\mathbb{Z}_{\geq0}$,

$$S_l(k):=\{(f,\boldsymbol{\tau})\in\mathcal{H}\times\mathbb{R}^8:\ |\langle(f,\boldsymbol{\tau}),(\kappa(\mathfrak{r}_l(k),\cdot),\mathbf{0})\rangle_{\mathcal{H}\times\mathbb{R}^8}-b_{0l}(k)|\leq\hat{\epsilon}\}.\qquad(5)$$

A closed convex constraint, as in (5), is usually called a *hyperslab*. Our wish is, once more, to "push" our solution in the intersection of these regions via a series of projections. To express the associated projection mappings in a compact, analytic (and in linear complexity) form, we introduce the following coefficients (the proof is omitted due to lack of space): $\forall k\in\mathbb{Z}_{\geq0}$, and $\forall l\in\overline{1,2}$:

$$\beta_{k,l}:=\begin{cases}\frac{b_{0l}(k)-\hat{\epsilon}-\langle f,\kappa(\mathfrak{r}_l(k),\cdot)\rangle}{\kappa(\mathfrak{r}_l(k),\mathfrak{r}_l(k))},\\ \qquad\text{if } b_{0l}(k)-\hat{\epsilon}>\langle f,\kappa(\mathfrak{r}_l(k),\cdot)\rangle,\\ 0,\quad\text{if } |\langle f,\kappa(\mathfrak{r}_l(k),\cdot)\rangle-b_{0l}(k)|\leq\hat{\epsilon},\\ \frac{b_{0l}(k)+\hat{\epsilon}-\langle f,\kappa(\mathfrak{r}_l(k),\cdot)\rangle}{\kappa(\mathfrak{r}_l(k),\mathfrak{r}_l(k))},\\ \qquad\text{if } b_{0l}(k)+\hat{\epsilon}<\langle f,\kappa(\mathfrak{r}_l(k),\cdot)\rangle.\end{cases}\qquad(6)$$

Then, the metric projection mapping $P_{S_l(k)}$ takes the following form; $\forall k\in\mathbb{Z}_{\geq0}$, $\forall l\in\overline{1,2}$, and $\forall(f,\boldsymbol{\tau})\in\mathcal{H}\times\mathbb{R}^8$,

$$P_{S_l(k)}((f,\boldsymbol{\tau}))=(f,\boldsymbol{\tau})+\beta_{k,l}(\kappa(\mathfrak{r}_l(k),\cdot),\mathbf{0}).\qquad(7)$$

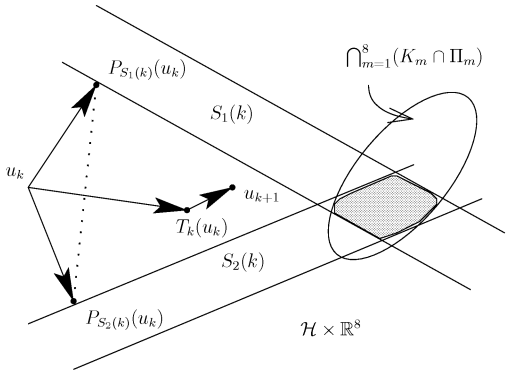

Figure 2: Illustration of the proposed algorithm (10). For simplicity, we consider here only two hyperslabs: $S_1(k)$, and $S_2(k)$. Concurrent processing on these two hyperslabs is achieved by the associated projections mappings $P_{S_1(k)}$ and $P_{S_2(k)}$. The set of all convex combinations $\sum_{(j,l)\in\mathcal{J}_k}\omega_{j,l}^{(k)}P_{S_l(j)}(u)$ in (8) is denoted by the dotted line. However, the extrapolation given by the mapping T_k takes us closer to the shaded region which denotes the desired beamformers for the time instant k. As time goes by, the hyperslabs change according to the incoming sequence of data, while the distortionless constraints $\bigcap_{m=1}^{8}(K_m\cap\Pi_m)$ remain fixed.

To summarize, *we seek for a point $(f,\boldsymbol{\tau})$ in the intersection of the following infinite collection of closed convex sets:* $\bigcap_{m=1}^{8}(K_m\cap\Pi_m)\cap(\bigcap_{k=k_0}^{\infty}\bigcap_{l=1}^{2}S_l(k))$, *for let's say some nonnegative integer k_0.*

It is well-known, as in the celebrated Affine Projection Algorithm [7], that concurrent processing can increase the speed of convergence of an algorithm. As such, we allow ourselves the freedom to process also hyperslabs (5) that correspond to data that have been received in time instants previous than the current k one. To do so, we introduce the following index set, which indicates the concurrently processed hyperslabs at each time instant k:

$$\mathcal{J}_k:=\begin{cases}\overline{0,k}\times\overline{1,2}, & \text{if } k<q-1,\\ \overline{k-q+1,k}\times\overline{1,2}, & \text{if } k\geq q-1,\end{cases}$$

where the symbol $\overline{j_1,j_2}:=\{j_1,j_1+1,\ldots,j_2\}$, for any integers j_1,j_2, where q is a predefined positive integer, and where the sumbol $\times$ stands for the (Cartesian) product operation. To quantify the contribution of each hyperlab to such concurrent processing, we assign a convex weight $\omega_{j,l}^{(k)}$ to each pair $(j,l)\in\mathcal{J}_k$. By convex weight, we mean that the set $\{\omega_{j,l}^{(k)}\}$ satisfies the following constraints: for any pair $(j,l)\in\mathcal{J}_k$ we have $\omega_{j,l}^{(k)}\in[0,1)$, and $\sum_{(j,l)\in\mathcal{J}_k}\omega_{j,l}^{(k)}=1$.

For any time instant $k\in\mathbb{Z}_{\geq0}$, concurrent processing is expressed by the following mapping:

$$T_k(u):=u+\mu_k\left(\sum_{(j,l)\in\mathcal{J}_k}\omega_{j,l}^{(k)}P_{S_l(j)}(u)-u\right),\ \forall u\in\mathcal{H}\times\mathbb{R}^8.\qquad(8)$$

Due to the concurrency, i.e., the multiplicity of the constraints that are processed at every time instant k, the relation (8) extrapolates the projections mappings $\{P_{S_l(j)}\}_{(j,l)\in\mathcal{J}_k}$. The range of the *extrapolation parameter* μ_k, which affects the speed of convergence, is calculated recursively within an iterative scheme as follows: given the current estimate $u_k\in\mathcal{H}\times\mathbb{R}^8$, μ_k takes values inside the interval $[0,2\mathcal{M}_k]$, where $\mathcal{M}_k$ is calculated by the closed form

(the proof is omitted due to lack of space):

$$\mathcal{M}_k := \begin{cases} \frac{\sum_{(j,l)\in\mathcal{J}_k} \omega_{j,l}^{(k)} \|P_{S_l(j)}(u_k)-u_k\|^2}{\|\sum_{(j,l)\in\mathcal{J}_k} \omega_{j,l}^{(k)} P_{S_l(j)}(u_k)-u_k\|^2}, \\ \qquad \text{if } u_k \notin \bigcap_{(j,l)\in\mathcal{J}_k} S_l(j), \\ 1, \qquad \text{otherwise.} \end{cases} \quad (9)$$

By the convexity of the function $\|\cdot\|^2$ and the definition of $\mathcal{M}_k$, we can derive that $\mathcal{M}_k \geq 1$ (see Fig. 2). In general, the larger the μ_k, but with $\mu_k \leq 2\mathcal{M}_k$, the larger the extrapolation step in (8), and thus the faster the convergence speed of the algorithm. For the special case where $\mathcal{J}_k$ contains only a single element, i.e., $\mathcal{J}_k := \{(j_0, l_0)\}$, then we no longer have concurrency since $T_k = P_{S_{l_0}(j_0)}$ if $\mu_k := 1$ in (8).

3.3 The Algorithm.

Assume an arbitrary initial point $u_0 := (f_0, \boldsymbol{\tau}_0)$ in $\mathcal{H} \times \mathbb{R}^8$, and generate the point sequence $(u_k)_{k\in\mathbb{Z}_{\geq 0}} := (f_k, \boldsymbol{\tau}_k)_{k\in\mathbb{Z}_{\geq 0}}$ by using the mappings of the previous sections as follows:

$$u_{k+1} := P_{K_1} \cdots P_{K_8} P_{\Pi_1} \cdots P_{\Pi_8} T_k(u_k), \ \forall k \in \mathbb{Z}_{\geq 0}. \quad (10)$$

The relation (10) is a special case of a very recently developed version of the *Adaptive Projected Subgradient Method (APSM)* [4]. Although APSM relies on the same principles of the celebrated *Projections Onto Convex Sets (POCS)* method [8], it is a generalization, since POCS deals only with a finite number of closed convex constraints.

Under mild conditions, the algorithm (10) strongly converges to a point $u_* := (f_*, \boldsymbol{\tau}_*)$ in $\mathcal{H} \times \mathbb{R}^8$ such that f_* is the desired beamformer. It can also be proved that the sequence of beamformers $(f_k)_k$ takes the following form $\forall k \in \mathbb{Z}_{>0}$:

$$f_k = \sum_{m=1}^{8} \alpha_m^{(k)} h_m + \sum_{(j,l)\in\overline{0,k-1}\times\overline{1,2}} \alpha_{j,l}^{(k)} \kappa(\mathfrak{r}_l(j), \cdot),$$

where $\{\alpha_m^{(k)}\}_{m,k}$ and $\{\alpha_{j,l}^{(k)}\}_{j,l,k}$ are properly defined real coefficients, and the points h_m are given in (2). The above algorithm enjoys also several theoretical properties like monotone approximation, asymptotic optimality with respect to a certain sequence of cost functions, as well as a characterization of the limit point u_*. Due to lack of space, these properties and the associated proofs will be presented elsewhere.

3.4 Sparsification.

Looking at the sequence of the estimates produced by the proposed algorithm in (3.3), it becomes clear that as the time index k advances, the collections $(\mathfrak{r}_l(i))$, $i \in \overline{0,k}, l \in \overline{1,2}$, and $(\alpha_{j,l}^{(i)})$, $j \in \mathcal{J}_k, l \in \overline{1,2}, i \in \overline{0,k}$, must be saved in memory for keeping track of the functional representation in (3.3). This leads to an increase, at a linear rate, of the computational load, which comes in conflict with the limited resources of real time online settings.

Recent research [9] on such kernel series expansions focuses on sparsification of the functional representations (3.3), i.e., on constructing estimates where the number of terms used in (3.3) is finite. In [10], sparsification was achieved by forcing every estimate f_k, in a classification task, to lie within a sphere of given radius $\Delta > 0$ in the RKHS $\mathcal{H}$. In other words, the estimates were forced to satisfy $\|f_k\| \leq \Delta$, $\forall k \in \mathbb{Z}_{\geq 0}$. Following this line, and since our working space is $\mathcal{H} \times \mathbb{R}^8$, we define the following constraint

$$\mathcal{B} := \{(f, \boldsymbol{\tau}) \in \mathcal{H} \times \mathbb{R}^8 : \|f\| \leq \Delta\}, \quad (11)$$

which is a closed convex set. Its metric projection mapping $P_{\mathcal{B}}$ takes a very simple (with linear complexity) form: $\forall (f, \boldsymbol{\tau}) \in \mathcal{H} \times \mathbb{R}^8$,

$$P_{\mathcal{B}}((f, \boldsymbol{\tau})) = \begin{cases} (f, \boldsymbol{\tau}), & \text{if } \|f\| \leq \Delta, \\ (\frac{\Delta}{\|f\|} f, \boldsymbol{\tau}), & \text{if } \|f\| > \Delta. \end{cases} \quad (12)$$

It can be shown, in a way similar to the one exhibited in [10], that as time goes by, and as more and more data enter the functional representation in (3.3), the bound Δ on the norm of the estimates f_k potentially forces the coefficients corresponding to old data to degrade to zero. In this way, one can discard old samples, and can introduce a buffer of finite length L_b that keeps only the L_b most recent data:

$$\tilde{f}_k = \sum_{m=1}^{8} \tilde{\alpha}_m^{(k)} h_m + \sum_{(j,l)\in\overline{k-\frac{L_b}{2}, k-1}\times\overline{1,2}} \tilde{\alpha}_{j,l}^{(k)} \kappa(\mathfrak{r}_l(j), \cdot), \quad (13)$$

where $\{\tilde{\alpha}_m^{(k)}\}_{m,k}$ and $\{\tilde{\alpha}_{j,l}^{(k)}\}_{j,l,k}$ are some properly defined real coefficients. Thus, one can add also the projection mapping $P_{\mathcal{B}}$ to (10), and form the algorithm $u_{k+1} := P_{K_1} \cdots P_{K_8} P_{\Pi_1} \cdots P_{\Pi_8} P_{\mathcal{B}} T_k(u_k)$, $\forall k \in \mathbb{Z}_{\geq 0}$, for an arbitrary u_0, in order to introduce sparsification to the design.

Considering now all the convex constraints, our goal is to reach a point

$$(f_*, \boldsymbol{\tau}_*) \in \mathcal{B} \cap \left(\bigcap_{m=1}^{8} (K_m \cap \Pi_m) \right) \cap \left(\bigcap_{k=k_0}^{\infty} \bigcap_{l=1}^{2} S_l(k) \right), \quad (14)$$

where f_* is the desired beamformer.

The main reason for introducing a beamforming problem in a very high dimensional (possibly infinite) RKHS is to increase the probability of having the intersection in (14) nonempty, as opposed to standard methodologies that pose the same problem in finite dimensional Euclidean spaces. A justification for this rationale will be given in the next section, where by only a few array elements we shall be able to compute effective beamformers in infinite dimensional RKHS that cannot be obtained in the classical finite dimensional space framework.

4. NUMERICAL EXAMPLES

In Fig. 1, we let $N := 3$ with $d/\lambda := 0.5$. The SOI's DOA is 90°, and the DOAs of five jammers are 30°, 80°, 100°, 130°, 160°. Gaussian i.i.d. noise is used to form the process $(\boldsymbol{n}(k))_k$. The SNRs are given by 10, 10, 30, 20, 10, and 30 dB for the SOI and the jammers respectively. Alltogether, the input SINR is −23.26dB. In addition, we assume the erroneous information of 93° for the SOI's DOA. We used the Gaussian kernel function for working in an infinite dimensional Hilbert space (see Section 2.1), with a variance of $\sigma^2 := 0.5$. We choose $q := 100$ for the index set $\mathcal{J}_k$, and all the convex weights $\{\omega_{j,l}^{(k)}\}$ are set equal to each other for every time instant k (see Section 3.3). The extrapolation parameter $\mu_k := 1.95\mathcal{M}_k$, $\forall k \in \mathbb{Z}_{\geq 0}$.

We let also $\epsilon := 0.2$, $\hat{\epsilon} := 0.2$, $\delta := 0.1$, and $\Delta := 10$. The larger the values of ϵ, $\hat{\epsilon}$, the larger the set of solutions for our beamformers, while the opposite conclusion holds for δ. We noticed that for smaller values of ϵ, $\hat{\epsilon}$, faster convergence results can be obtained. This becomes a natural consequence, if we consider the fact that the smaller the solution set becomes, the faster we move forward via projections to the desired region. The parameter Δ affects the sparsification of the algorithm. We noticed that the concurrent processing of

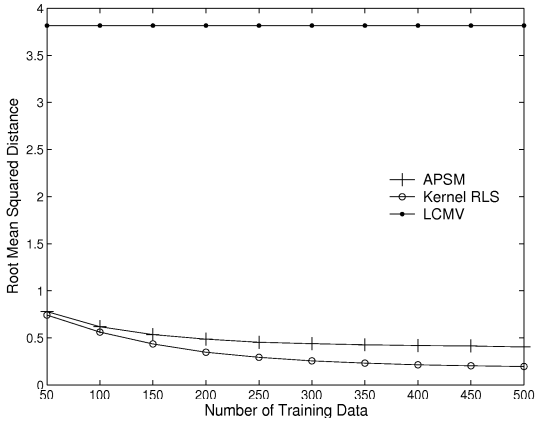

Figure 3: The KRLS shows quadratic complexity $\mathcal{O}(m_b^2)$, where m_b is the dimension of a basis (here the average value of m_b is 764.26 with a sparsification criterion $\nu_{\text{KRLS}} = 0.5$ [9]). The only quadratic complexity burden for the proposed method, but with respect to the cardinality $\text{card}(\mathcal{J}_k)$ (here $\text{card}(\mathcal{J}_k) = 200$), is due to the calculation of the quadratic form appearing in the denumerator of $\mathcal{M}_k$ in (9). All the introduced projection mappings show a linear complexity with respect to the number of kernel fuctions in (13), i.e., of order $\mathcal{O}(L_b)$.

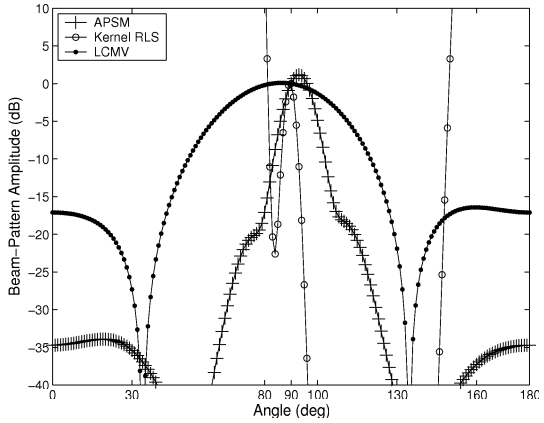

Figure 4: The KRLS offers unconstrained nonlinear regression, so it shows inability to control the beam-pattern of the ULA as in the convexly constrained optimization approach of the APSM. The linear approach of LCMV needs more antenna elements to produce a narrower main lobe.

the proposed method behaves in a stable and uniform way for a wide interval of Δ values.

The total number of training data to be used was set to 500 and a separate 100 test data were used to validate the obtained results and to calculate the corresponding SINRs. The buffer length L_b was set to 500, i.e., half the number of the training data since at each time instant k the received $\boldsymbol{r}(k)$ is mapped to $\mathfrak{r}_l(k)$, $l \in \overline{1,2}$. For each realization of the experiment, we calculate the root mean squared distance of the beamformer's output to the test data, and the array beam-pattern. We performed 100 realizations and uniformly averaged the results. The proposed method (denoted by APSM) is compared to the classical *Linearly Constrained Minimum Variance (LCMV)* beamformer [1, 2], where perfect knowledge of all of the DOAs is available, and the nonlinear regression approach realized by the *Kernel Recursive Least Squares (KRLS)* algorithm [9].

In Fig. 3, the average root mean squared distance of the beamformer's output to the test data is depicted. Recall that in the BPSK constellation, the desired symbols are placed at points $[\pm 1, 0]^t$ in $\mathbb{R}^2$. The linear approach of LCMV performs poorly since it needs more array elements ($N = 3$ here) to deal with the given number J ($= 5$) of jammers. In Fig. 4, we design the corresponding beam-patterns. The LCMV approach exhibits a very wide main lobe so that it cannot separate the jammers located close to SOI's DOA of 90°, i.e., at the angles 80°, 100°. Indeed, it shows only an average output of $\text{SINR}_{\text{LCMV}} = -20.21\text{dB}$ as compared to the large improvement of $\text{SINR}_{\text{APSM}} = 18.65\text{dB}$ offered by the APSM approach. Moreover, since the KRLS performs only nonlinear regression, and does not incorporate any other design constraints, it shows inability to control the beam-pattern, which can lead to unpleasant effects, e.g., amplification of the jammer located at 160°. As a result it exhibits a very low negative SINR value.

5. CONCLUSIONS

This paper presented a novel adaptive nonlinear beamforming method in Reproducing Kernel Hilbert Spaces (RKHS). To overcome the limitations and to enhance the capabilities of the array against noise and multiple jamming signals, we use a reproducing kernel function to implicitly transfer the beamforming problem from the standard Euclidean space to a very high dimensional RKHS. Convex analysis is used to formulate a convexly constrained minimization problem in that implicit space. A solution is given by a very recently developed projection-based minimization tool. The exhibited numerical results show that, without increasing the array elements, a low complexity convexly constrained design in the implicitly defined RKHS gives solution in cases where the classical linear solution does not, and outperforms a recently developed kernel Recursive Least Squares technique.

REFERENCES

[1] H. L. Van Trees, *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory*, John Wiley & Sons, New York, 2002.

[2] J. Li and P. Stoica, *Robust Adaptive Beamforming*, Wiley, 2005.

[3] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, Amsterdam, 3rd edition, 2006.

[4] K. Slavakis, I. Yamada, and N. Ogura, "The Adaptive Projected Subgradient Method over the fixed point set of strongly attracting nonexpansive mappings," *Numerical Functional Analysis and Optimization*, vol. 27, no. 7&8, pp. 905–930, 2006.

[5] K. Slavakis and I. Yamada, "Robust wideband beamforming by the Hybrid Steepest Descent Method," *IEEE Trans. Signal Processing*, vol. 55, no. 9, pp. 4511–4522, Sept. 2007.

[6] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[7] A. H. Sayed, *Fundamentals of Adaptive Filtering*, John Wiley & Sons, New Jersey, 2003.

[8] L. G. Gubin, B. T. Polyak, and E. V. Raik, "The method of projections for finding the common point of convex sets," *USSR Comput. Math. Phys.*, vol. 7, pp. 1–24, 1967.

[9] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Trans. Signal Processing*, vol. 52, no. 8, pp. 2275–2285, Aug. 2004.

[10] K. Slavakis, S. Theodoridis, and I. Yamada, "Online sparse kernel-based classification by projections," in *Proc. IEEE Machine Learning for Signal Processing (MLSP)*, Thessaloniki: Greece, Aug. 2007, pp. 294–299.