

Article

A Nonlinear Adaptive Beamforming Algorithm Based on Least Squares Support Vector Regression

Lutao Wang ^{1,*}, Gang Jin ², Zhengzhou Li ³ and Hongbin Xu ¹

¹ School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China; E-Mail: hbxu@uestc.edu.cn

² China Aerodynamics Research & Development Center, Mianyang 621000, China; E-Mail: gjin@ioe.ac.cn

³ School of Communication, Chongqing University, Chongqing 400044, China; E-Mail: lizhengzhou@cqu.edu.cn

* Author to whom correspondence should be addressed; E-Mail: wltuestc@163.com; Tel./Fax: +86-28-6183-1819.

Received: 28 July 2012; in revised form: 21 August 2012 / Accepted: 30 August 2012 /

Published: 12 September 2012

Abstract: To overcome the performance degradation in the presence of steering vector mismatches, strict restrictions on the number of available snapshots, and numerous interferences, a novel beamforming approach based on nonlinear least-square support vector regression machine (LS-SVR) is derived in this paper. In this approach, the conventional linearly constrained minimum variance cost function used by minimum variance distortionless response (MVDR) beamformer is replaced by a squared-loss function to increase robustness in complex scenarios and provide additional control over the sidelobe level. Gaussian kernels are also used to obtain better generalization capacity. This novel approach has two highlights, one is a recursive regression procedure to estimate the weight vectors on real-time, the other is a sparse model with novelty criterion to reduce the final size of the beamformer. The analysis and simulation tests show that the proposed approach offers better noise suppression capability and achieve near optimal signal-to-interference-and-noise ratio (SINR) with a low computational burden, as compared to other recently proposed robust beamforming techniques.

Keywords: adaptive beamforming; least-squares support vector regression (LS-SVR); sparsification; kernel function

1. Introduction

As one important branch of modern array signal processing, the beamforming technique has been widely studied and applied in the radar, wireless communication, sonar, medical imaging, as well as astronomy domains. The standard beamforming approach, such as the minimum variance distortionless response (MVDR) beamformer [1], was usually established based on an ideal antenna array with exactly known array manifold. Thus, it is very sensitive to practical circumstances, and its performance would be seriously degraded by diverse factors, such as the steering vector mismatch, array calibration errors and snapshot number restrictions.

During the last decades, in order to resist the model mismatches and possible environment changes, the robust beamforming approach have been largely studied [2–5]. Among others, by introducing a penalty term into the objective function, the diagonal loading (DL) algorithm could effectively reduce the eigenvalue spread of the noise and prevent the distortion of beampattern [6]. Nevertheless, how to get the optimal loading factor for DL is still a serious issue when the desired steering vector and/or the available snapshot numbers are uncertain [7]. A robust adaptive beamforming, based on the worst-case performance optimization, would delimit the uncertainty set of steering vectors by upper bounding the norm of the steering vector mismatch [8]. However, neither the mismatch vector nor its upper bound is known in practice. To overcome this model defect in standard DL algorithm, an adaptive beamforming method was developed, which estimates iteratively the difference between the actual and presumed steering vectors in order to maximize the output signal-to-noise plus interference ratio (SINR) [9–11]. But this adaptive beamforming algorithm is not sufficiently reliable in the case when the snapshots are small.

In order to reject jamming signals, poor array calibration, signal wave-front distortions, the minimum-variance-distortionless-response (MVDR) beamforming is modified by the means of incorporating multiple linear constrains [12–14]. Whereas, the augmentation of constrains would reduce the array freedom degrees in the linear beamforming framework. Nonlinear beamforming approaches provide a novel idea to address this issue for they can adapt better to the statistical properties of the given data than linear ones [15]. Neural network has been applied to beamforming among other nonlinear array processing tasks. But this approach suffers from serious drawbacks such as over-fitting or local minima, which leads to suboptimal solutions [16].

Support Vector Machines (SVM), introduced by Vapnik [17], is an important new methodology for pattern classification and nonlinear function approximation. This method addresses the beamforming problem by means of incorporating additional inequality constrains to penalize sidelobe levels and allowing a certain error in the desired signal direction [18]. Thus the MVDR beamforming method is reformulated and the cost function turns out to be equivalent to SVM for regression. However, the time consumed to train SVM beamformer scales super linearly to the number of observations, and it leads to an insurmountable computational burden in online operation modes [19]. The least-squares support vector machine (LS-SVM) inherits the SVM's generalization capacity. By solving linear equations instead of a quadratic programming (QP) problem in the standard SVM, the training procedure and the computational complexity of the standard SVM would be effectively simplified [20]. The main drawback of LS-SVM is that it works in batch mode. Thus, it is difficult to be used in large-scale

applications. Recent researches about LS-SVM continuously focus on the improvement of the training algorithms, model selection and sparseness [21,22].

This paper presents a new LS-SVR-based approach to address the robust beamforming issue. This approach alleviates the array output SINR degradation in the presence of steering vector mismatches, strict restrictions on the number of available snapshots, and numerous interferences by replacing the conventional linearly constrained minimum variance cost function with a squared-loss function, and achieves better generalization capacity by applying Gaussian kernels to the array observations. We also present a fast recursive procedure to estimate the weight vectors on real-time, and a novelty criterion to perform model reduction. The paper is organized as follows. The signal model, also the minimum mean square error (MMSE) and the MVDR-beamformer solutions are presented in Section 2. The basic principle of LS-SVR-based beamforming method is introduced in Section 3. In Section 4, a recursive procedure to calculate the regression parameters is provided. And a sparse mode is presented in Section 5. The simulation tests under different mismatch scenarios are illustrated in Section 6. A summary conclusion is given at the last of this paper.

2. Sensor Signal Model

Consider a linear array of M sensors receives signals from D narrowband source. The vector of array observations $\mathbf{x}(t) \in \mathbf{C}^{M \times 1}$ at time t could be modeled as:

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

where, $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_D]^T \in \mathbf{R}^{D \times 1}$ is the vector with the directions of arrival (DOA) and $(\cdot)^T$ stands for transpose, $\mathbf{A} = [\mathbf{a}(\theta_1), \mathbf{a}(\theta_2) \dots \mathbf{a}(\theta_D)] \in \mathbf{C}^{M \times D}$ is the matrix containing the array steering vectors $\mathbf{a}(\theta_i) = [1, e^{-j2\pi \sin(\theta_i)d/\lambda} \dots e^{-j2\pi \sin(\theta_i)d(M-1)/\lambda}]^T \in \mathbf{C}^{M \times 1}$. The uncorrelated sources are represented by the vector $\mathbf{s}(t) = [s_1(k), s_2(k) \dots s_D(k)]^T \in \mathbf{C}^{D \times 1}$. The vector $\mathbf{n}(t) \in \mathbf{C}^{M \times 1}$ is the sensor noise, and it is assumed as complex Gaussian with zero-mean:

$$\mathbf{A} = [\mathbf{a}(\theta_1), \mathbf{a}(\theta_2) \dots \mathbf{a}(\theta_D)] \quad (2)$$

The output of the beamformer is defined as:

$$y(t) = \mathbf{w}^H \mathbf{x}(t) \quad (3)$$

where, $\mathbf{w} = [w_1, \dots, w_M]^T \in \mathbf{C}^{M \times 1}$ is the complex vector of beamformer weights, $(\cdot)^T$ denotes transpose, and $(\cdot)^H$ denotes Hermitian transpose.

If certain observations are known during the procedure of training parameters, then, according to the MMSE criterion, the complex vector of beamformer weights \mathbf{w} can be described as:

$$\mathbf{w} = \mathbf{R}^{-1} \mathbf{p} \quad (4)$$

where, \mathbf{R} is $M \times M$ covariance matrix, and \mathbf{p} is the cross-correlation between the desired output and the received signal.

The classical MVDR beamformer minimizes the array output energy, and the weights subject to a constraint of unity array response on the desired array steering vectors, that is:

$$\min_{\mathbf{w}} \mathbf{w}^H \mathbf{R} \mathbf{w} \quad \text{s.t.} \quad \mathbf{w}^H \mathbf{a}(\theta_1) = 1 \quad (5)$$

The constraint $\mathbf{w}^H \mathbf{a}(\theta_1) = 1$ prevents the gain at the look direction from being reduced, and the solution of Equation (5) can be easily estimated by means of using Lagrange multiplier method:

$$\mathbf{w} = \frac{\mathbf{R}^{-1} \mathbf{a}(\theta_1)}{\mathbf{a}(\theta_1)^H \mathbf{R}^{-1} \mathbf{a}(\theta_1)} \quad (6)$$

In practice, it is not feasible to calculate the exact covariance matrix \mathbf{R} and it would be estimated by the sample covariance matrix $\hat{\mathbf{R}} = \frac{1}{K} \sum_{k=1}^K \mathbf{x}(k) \mathbf{x}^H(k)$ where K is the number of observed snapshots.

The performance of MVDR beamformer in Equation (5) is sensitive to mismatch between the presumed and actual steering vectors due to the uncertainty of the desired signal DOA, strict restrictions on the number of available snapshots, and numerous interferences.

3. LS-SVR-Based Beamforming Method

3.1. Nonlinear SVM-Based Beamforming

Consider a set of snapshots $\mathbf{x}_i, i = 1, \dots, N$ at time t from an array and the corresponding set of desired symbols $y_i, i = 1, \dots, N$, are available for training purpose. The basic idea of nonlinear beamforming is to transform the data set $\mathbf{x}_i, i = 1, \dots, N$ into a higher (possibly infinite) dimension feature space H by a nonlinear transformation $\phi(\cdot)$. Thus, the beamformer's output can be formulated as a linear regression in H . It could be expressed as:

$$y_i = \mathbf{w}^H \phi(\mathbf{x}_i) + e_i \quad (7)$$

where, $\mathbf{w} \in H$ is the linear parameter set and e_i is the output error.

The parameter set \mathbf{w} can be estimated by minimizing a certain cost function on output error e_i . For SVM regression, the parameter set \mathbf{w} and the ε -insensitive loss function could be estimated by the minimum risk criterion, *i.e.*,

$$\min J(\mathbf{w}, \varepsilon) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N L_\varepsilon(\xi_i, \zeta_i) \quad (8)$$

subject to $\xi_i, \zeta_i \geq 0$. Where, $C \geq 0$ is the tradeoff term between the minimization of the weight norm and the output error. The ε -insensitive loss function is given by:

$$L(e) = \begin{cases} 0, & |e| < \varepsilon \\ |e| - \varepsilon, & |e| \geq \varepsilon \end{cases} \quad (9)$$

where ε is a positive parameter which is used as an error threshold.

The weight vector \mathbf{w} is regularized by solving Equation (8), Thus, the generalization capacity of the beamformer will be remarkably improved.

3.2. Nonlinear LS-SVR Beamforming

Instead of the inequality constraints in standard SVM algorithm, the equality ones are taken in LS-SVR, and the linear equation of the ε -insensitive loss function is replaced by a quadratic equation. Therefore, The LS-SVR beamformer can be described as the following quadratic optimization problem [20]:

$$\begin{aligned} \min J(\bar{\mathbf{w}}_t, \bar{e}_t) &= \frac{1}{2} \|\bar{\mathbf{w}}_t\|^2 + C \sum_{i=1}^N e_{i,t}^2 \\ \text{s.t. } e_{i,t} &= \bar{y}_{i,t} - \bar{\mathbf{w}}_t^T \phi(\bar{\mathbf{x}}_{i,t}) - b_t, i = 1, 2, \dots, N \end{aligned} \quad (10)$$

where, e_t^i is the error at time t . The sum of squared errors in Equation (10) represents the ε -insensitive loss function under the linear constraint. This treatment would greatly reduce the computation complexity since only the linear equation, instead of the QP problem in SVM, is solved.

The array observations of the beamformer are complex, whereas the variables in the objective function of SVM are real. So, it is necessary to rewrite the complex variables as real variables. For this reason, the array observations \mathbf{x}_i , the beamformer outputs y_i and the weight vectors \mathbf{w}_t are rewritten as:

$$\begin{aligned} \bar{\mathbf{x}}_{i,t} &= \begin{cases} [\text{Re}(\mathbf{x}_{i,t}^T) \ \text{Im}(\mathbf{x}_{i,t}^T)]^T \in R^{2M}, & i = 1, \dots, N \\ [\text{Im}(\mathbf{x}_{i-t,t}^T) \ -\text{Re}(\mathbf{x}_{i-t,t}^T)]^T \in R^{2M}, & i = t+1, \dots, 2N \end{cases} \\ \bar{\mathbf{w}}_t &= [\text{Re}(\mathbf{w}_t^T) \ \text{Im}(\mathbf{w}_t^T)]^T \in R^{2M} \\ \bar{y}_{i,t} &= \begin{cases} \text{Re}(y_{i,t}), & i = 1, \dots, N \\ \text{Im}(y_{i-t,t}), & i = t+1, \dots, 2N \end{cases} \end{aligned} \quad (11)$$

The result of the quadratic optimization problem of Equation (10) is the saddle point of the following Lagrange function:

$$\begin{aligned} L(\bar{\mathbf{x}}_{i,t}, b_t, \bar{e}_t, \boldsymbol{\alpha}_t) &= J(\bar{\mathbf{w}}_t, e_t) - \\ &\sum_{i=1}^{2N} \alpha_{i,t} \{ \bar{\mathbf{w}}_t^T \phi(\bar{\mathbf{x}}_{i,t}) + b_t + e_{i,t} - \bar{y}_{i,t} \} \end{aligned} \quad (12)$$

where, $\boldsymbol{\alpha}_t = (\alpha_1, \alpha_2, \dots, \alpha_{2N})^T$, $\alpha_i > 0$ is Lagrange multipliers, defined as regression parameters in this paper.

According to the Karush-Kuhn-Tucker (KKT) conditions, differentiating the above function with respect to the Lagrange multipliers $\boldsymbol{\alpha}_t$ and $\bar{\mathbf{x}}_{i,t}, b_t, e_{i,t}$ yields:

$$\begin{cases} \frac{\partial L}{\partial \bar{\mathbf{w}}_t} = 0 \Rightarrow \bar{\mathbf{w}}_t = \sum_{i=1}^{2N} \alpha_i \phi(\bar{\mathbf{x}}_{i,t}) \\ \frac{\partial L}{\partial b_t} = 0 \Rightarrow \sum_{i=1}^{2N} \alpha_i = 0 \\ \frac{\partial L}{\partial \bar{e}_{i,t}} = 0 \Rightarrow \alpha_i = C e_{i,t} \\ \frac{\partial L}{\partial \alpha_{i,t}} = 0 \Rightarrow \bar{\mathbf{w}}_t^T \phi(\bar{\mathbf{x}}_{i,t}) + b_t + e_{i,t} - y_{i,t} \end{cases} \quad (13)$$

The system obtained from the KKT conditions is linear. Its result is obtained by solving the linear system which is expressed as following matrix:

$$\begin{bmatrix} 0 & \bar{\mathbf{e}}^T \\ \bar{\mathbf{e}} & \mathbf{Q}_t + C^{-1}\mathbf{I} \end{bmatrix} \begin{bmatrix} b_t \\ \mathbf{a}_t \end{bmatrix} = \begin{bmatrix} 0 \\ \bar{\mathbf{y}}_t \end{bmatrix} \quad (14)$$

where, $\bar{\mathbf{y}}_t = (\bar{y}_1, \bar{y}_2, \dots, \bar{y}_{2N})^T$, $\bar{\mathbf{e}} = (1, 1, \dots, 1)^T$, \mathbf{Q}_t is Gram matrix and the element of \mathbf{Q}_t is $\mathbf{Q}_{i,j} = \langle \phi(\bar{\mathbf{x}}_{i,t}), \phi(\bar{\mathbf{x}}_{j,t}) \rangle = k_t(\bar{\mathbf{x}}_i, \bar{\mathbf{x}}_j)$, $i, j = 1, 2, \dots, 2N$, $k_t(\bar{\mathbf{x}}_i, \bar{\mathbf{x}}_j)$ denotes kernel function responsible for the nonlinear mapping $\phi(\cdot)$, which greatly simplify the inner product calculation in the feature space. Thus, linear methods can be applied on the transformed data, and it is not necessary to perform computations in the high-dimensional feature space. As the most widely used kernel function in many practical applications, Gaussian kernel is taken here:

$$k(\bar{\mathbf{x}}_i, \bar{\mathbf{x}}_j) = \exp\left(-\frac{\|\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\|^2}{2\sigma^2}\right) \quad (15)$$

where $\sigma > 0$ is the kernel radius.

The outputs of the nonlinear LS-SVR beamformer are:

$$\bar{y}_{t+1} = \sum_{i=1}^{2N} \alpha_i k(\bar{\mathbf{x}}_{t+1}, \bar{\mathbf{x}}_{i,t}) + b_t \quad (16)$$

4. Recursive Algorithms

From Equation (16), it could be known that once the regression parameters \mathbf{a}_t and b_t are computed, the beamformer outputs can be obtained. Denoting $\mathbf{U}_t = \mathbf{H}_t^{-1} = (\mathbf{Q}_t + C^{-1}\mathbf{I})^{-1}$, the result of LS-SVR (Equation (14)) can be represented as:

$$\begin{bmatrix} 0 & \bar{\mathbf{e}}^T \\ \bar{\mathbf{e}} & \mathbf{U}_t \end{bmatrix} \begin{bmatrix} b_t \\ \mathbf{a}_t \end{bmatrix} = \begin{bmatrix} 0 \\ \bar{\mathbf{y}}_t \end{bmatrix} \quad (17)$$

Then, we have:

$$\begin{aligned} b_t &= \frac{\bar{\mathbf{e}}^T \mathbf{U}_t \bar{\mathbf{y}}_t}{\bar{\mathbf{e}}^T \mathbf{U}_t \bar{\mathbf{e}}} \\ \mathbf{a}_t &= \mathbf{U}_t \left(\bar{\mathbf{y}}_t - \frac{\bar{\mathbf{e}} \bar{\mathbf{e}}^T \mathbf{U}_t \bar{\mathbf{y}}_t}{\bar{\mathbf{e}}^T \mathbf{U}_t \bar{\mathbf{e}}} \right) \end{aligned} \quad (18)$$

As the number of snapshots increases, the dimension of Gram matrix \mathbf{Q}_t will be increasing because it is in proportional to the number of snapshots. Therefore, the computation for the regression parameters \mathbf{a}_t and b_t would be very intensive as the snapshots increase, and it is key issue for LS-SVR beamformer to find out a fast algorithm to improve the computation efficiency of \mathbf{U}_t .

At time step t , \mathbf{Q}_t and \mathbf{H}_t are the matrixes with dimension of $2N \times 2N$:

$$\mathbf{Q}_t = \begin{pmatrix} k(\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_1) & \cdots & k(\bar{\mathbf{x}}_t, \bar{\mathbf{x}}_1) \\ \vdots & \ddots & \vdots \\ k(\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_t) & \cdots & k(\bar{\mathbf{x}}_t, \bar{\mathbf{x}}_t) \end{pmatrix}$$

$$\mathbf{H}_t = (\mathbf{Q}_t + C^{-1}\mathbf{I})$$

$$= \begin{pmatrix} k(\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_1) + 1/C & \cdots & k(\bar{\mathbf{x}}_{2N}, \bar{\mathbf{x}}_1) \\ \vdots & \ddots & \vdots \\ k(\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_{2N}) & \cdots & k(\bar{\mathbf{x}}_{2N}, \bar{\mathbf{x}}_{2N}) + 1/C \end{pmatrix}$$
(19)

As time run to $t + 1$, new input snapshots \mathbf{x}_{t+1} and the corresponding desired array output y_{t+1} are added to the current training set. So \mathbf{Q}_{t+1} and \mathbf{H}_{t+1} can be represented as:

$$\mathbf{Q}_{t+1} = \begin{pmatrix} k(\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_1) & \cdots & k(\bar{\mathbf{x}}_{2N}, \bar{\mathbf{x}}_1) & k(\bar{\mathbf{x}}_{2(N+1)}, \bar{\mathbf{x}}_1) \\ \vdots & \ddots & \vdots & \vdots \\ k(\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_{2N}) & \cdots & k(\bar{\mathbf{x}}_{2N}, \bar{\mathbf{x}}_{2N}) & k(\bar{\mathbf{x}}_{2(N+1)}, \bar{\mathbf{x}}_1) \\ k(\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_{2(N+1)}) & \cdots & k(\bar{\mathbf{x}}_{2N}, \bar{\mathbf{x}}_{2(N+1)}) & k(\bar{\mathbf{x}}_{2(N+1)}, \bar{\mathbf{x}}_{2(N+1)}) \end{pmatrix}$$

$$\mathbf{H}_{t+1} = \mathbf{Q}_{t+1} + C^{-1}\mathbf{I} =$$

$$\begin{pmatrix} k(\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_1) + 1/C & \cdots & k(\bar{\mathbf{x}}_{2N}, \bar{\mathbf{x}}_1) & k(\bar{\mathbf{x}}_{2(N+1)}, \bar{\mathbf{x}}_1) \\ \vdots & \ddots & \vdots & \vdots \\ k(\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_{2N}) & \cdots & k(\bar{\mathbf{x}}_{2N}, \bar{\mathbf{x}}_{2N}) + 1/C & k(\bar{\mathbf{x}}_{2(N+1)}, \bar{\mathbf{x}}_1) \\ k(\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_{2(N+1)}) & \cdots & k(\bar{\mathbf{x}}_{2N}, \bar{\mathbf{x}}_{2(N+1)}) & k(\bar{\mathbf{x}}_{2(N+1)}, \bar{\mathbf{x}}_{2(N+1)}) + 1/C \end{pmatrix}$$
(20)

Comparing the elements of \mathbf{H}_t and \mathbf{H}_{t+1} , the matrix \mathbf{H}_{t+1} could be reconstructed by the matrix \mathbf{H}_t plus an additional row and column, *i.e.*,

$$\mathbf{H}_{t+1} = \begin{pmatrix} \mathbf{H}_t & \mathbf{v}_{t+1} \\ \mathbf{v}_{t+1}^T & v_{t+1} \end{pmatrix}$$
(21)

where, $\mathbf{v}_{t+1} = [k(\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_{2(N+1)}), \dots, k(\bar{\mathbf{x}}_{2N}, \bar{\mathbf{x}}_{2(N+1)})]^T$, $v_{t+1} = k(\bar{\mathbf{x}}_{2(N+1)}, \bar{\mathbf{x}}_{2(N+1)}) + 1/C$.

According to the theorem of inverting block matrix, the inverse of \mathbf{H}_{t+1} can be expressed by the inverse of \mathbf{H}_t and the new column \mathbf{v}_{t+1} as:

$$\mathbf{H}_{t+1}^{-1} = \begin{pmatrix} \mathbf{H}_t^{-1} + \beta \mathbf{H}_t^{-1} \mathbf{v}_{t+1} \mathbf{v}_{t+1}^T \mathbf{H}_t^{-1} & -\beta \mathbf{H}_t^{-1} \mathbf{v}_{t+1} \\ -\beta \mathbf{v}_{t+1}^T \mathbf{H}_t^{-1} & \beta \end{pmatrix}$$
(22)

where, $\beta = (v_{t+1} - \mathbf{v}_{t+1}^T \mathbf{H}_t^{-1} \mathbf{v}_{t+1})^{-1}$. Thus the inverse of \mathbf{H}_{t+1} , which is equal to \mathbf{U}_{t+1} , can be calculated from the inverse of \mathbf{H}_t , and it is not necessary to calculate the inverse of \mathbf{H}_t when it has high dimension, so the computation complexity would be greatly reduced and the numerical stability problem arising from inverse matrix would be also avoided. When the set of snapshots is small, the \mathbf{U}_t can be computed directly by matrix inverse theory.

5. Sparsification

The crucial drawback of LS-SVR beamformer is that it deals with high-dimension matrix, which is equal to the number of the snapshots due to the use of a quadratic constraint function. This would bring

a big implementation problem to the proposed beamforming method since it is required to increase memory and computational resources as time evolves. Several methods have been proposed to cope with these problems [23,24]. The sliding-window approach [25] fixes the size of LS-SVR beamformer and allows it to be operated online in time-varying environments by keeping only the last N input snapshots in the sliding-window and simply abandoning those out of it. In [26], an exponential forgetting mechanism is introduced to describe the influence, which is imposed on the present situation by the past data [26]. This paper employs the novelty criterion, presented by Platt [27,28], to reduce the final size of the proposed beamformer, keep the algorithm complexity bounded and realize online sparsification. The basic idea of this approach is to construct a dictionary with center set C and update it appropriately according to the novelty criterion. The stages of the proposed specification are given as follows:

Step 1: Initializing an empty center set C_0 ;

Step 2: Calculating the distance between the new snapshot \mathbf{x}_t and the present dictionary $\text{dis} = \min_{\mathbf{c}_k \in C_i} \|\mathbf{x}_t - \mathbf{c}_k\|$;

Step 3: If the distance obtained from Step 2 is smaller than the preset threshold δ_1 , \mathbf{x}_t is not added into the dictionary, otherwise the prediction error $\mathbf{e}_t = y_t - \hat{y}_t$ is calculated;

Step 4: if $|\mathbf{e}_t|$ is larger than another preset threshold δ_2 , \mathbf{x}_t is accepted as a new center and C_i is updated to C_{i+1} , otherwise go to Step 2.

Increasing δ_1 and δ_2 , the final size of the LS-SVR beamformer will be decreased. But this will result to performance degradation. In practical applications, δ_1 is set to around one tenth of the kernel bandwidth, and δ_2 is around the square root of the steady-state mean square error (MSE). Cross-validation also can be used to select these appropriate thresholds.

Applying the above sparsification procedure, the computation complexity of the proposed beamformer will be reduced from $O(N^2)$ to $O(K^2)$, where K is the effective number of centers in the network at time t . As K is finite, the online real-time beamforming will be practical.

6. Simulation Tests

To evaluate the performance of the proposed LS-SVR-based beamformer, simulation tests are carried out. A 10 elements uniform linear array with half-wavelength spacing is taken into account. The desired signal comes from a presumed direction $\theta = 3^\circ$ and two irrelevant interferences, with interference-to-noise ratio (INR) of 20 dB, impinge on the array from $\theta_2 = -32^\circ$ and $\theta_3 = 17^\circ$ respectively. The additive noise is assumed to be a 0-dB complex white Gaussian distributed random variable. For comparison purpose, the conventional MVDR, the diagonal loading MVDR (MVDR-DL), the ES [29], the SQP [9] and the RR [30] method are considered. The parameters of the proposed beamformer, σ , δ_1 and δ_2 , are chosen as 1.0, 0.1 and 0.08 respectively. The load value of MVDR-DL beamformer is set to $(P_e + 10 \text{ dB})$, where P_e denotes the power of desired signal. All results are obtained from 100 independent simulation runs.

The first simulation aims to compare the performance of these beamformers when steering vector mismatch is presented. From Figure 1(a), we observe that the proposed LS-SVR beamformer consistently improves its output SINR as SNR increases and performs much closer as the idea one

when the input SNR is varied from -20 dB to 30 dB. Due to the DOA mismatch, the interested signal is considered as interference and a null is allocated in the desired signal direction by the MVDR beamformer. As a result, the output SINR is decreased. When input SNR is larger than -5 dB, the output SINR of MVDR beamformer degrades seriously. In comparing with the MVDR beamformer, the MVDR-DL, ES, SQP and RR methods get more robustness against DOA mismatch. But they still suffer from a degradation of performance while the input SNR becomes higher.

Figure 1. Scenario with only DOA mismatch (a) Output SINR versus SNR; (b) Beam-patterns, SNR = 10 dB, 2 interference.

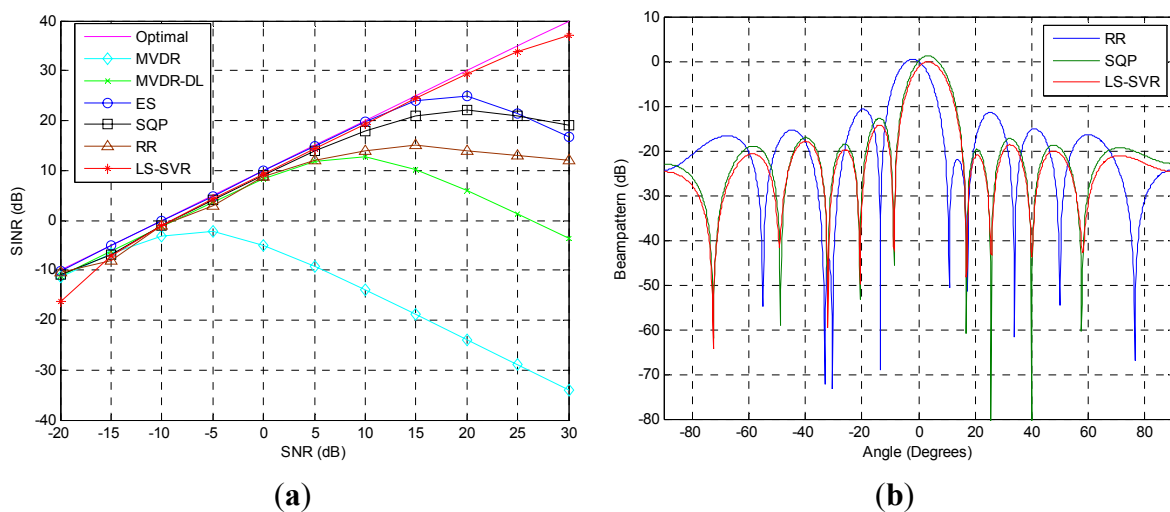
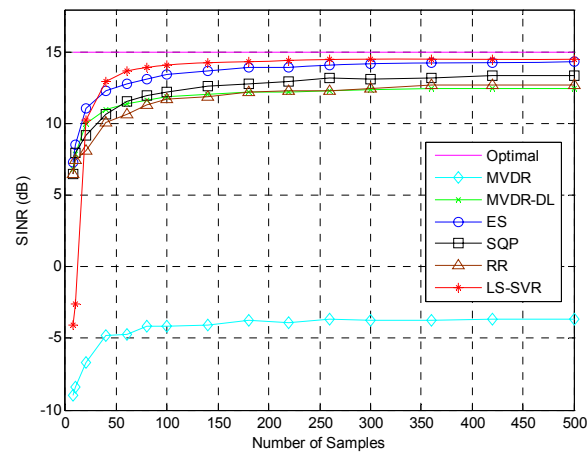
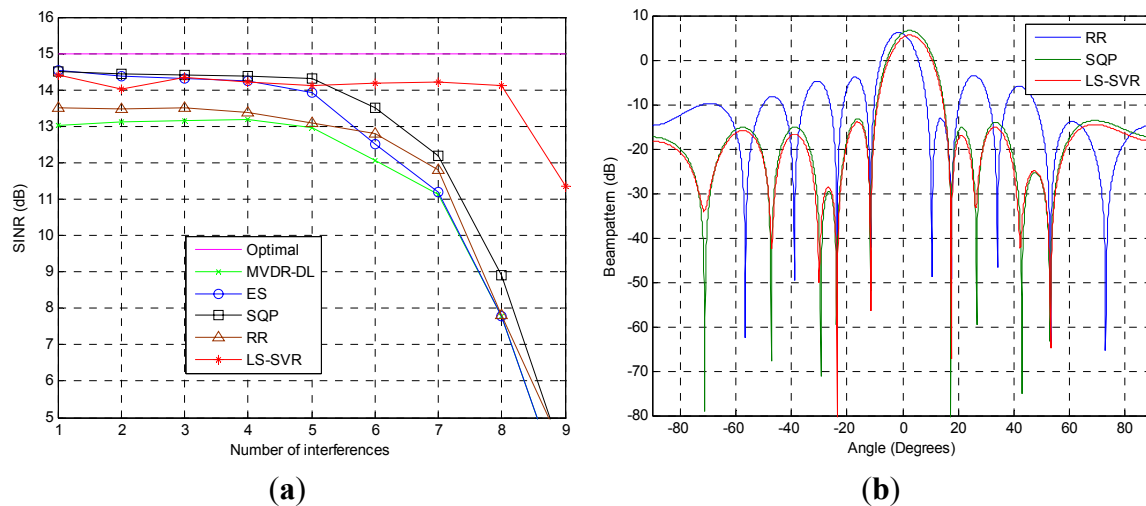


Figure 1(b) shows the normalized beampattern plots when the input SNR is equal to 10 dB. As it is illustrated, all beam-patterns of the robust beamformers have nulls at the DOAs of the interferences. But the proposed LS-SVR still outperforms others by markedly lower sidelobe level, and maintaining distortionless response for the desired signal.

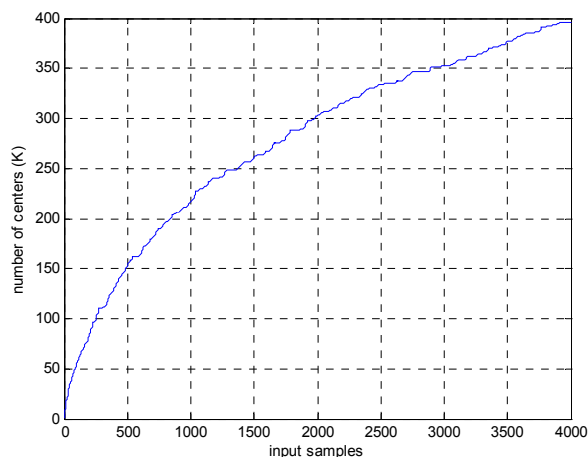
The covariance matrix would be inaccurately estimated owing to insufficient snapshots, DOA mismatch of desired signal and array calibration errors. This kind of inaccuracy may result in the degradation of array response. Hence, both the errors of insufficient snapshots and DOA mismatch are considered to verify the proposed beamformer in our second simulation tests. Figure 2 shows the resulting output SINRs versus the snapshot number K . When snapshots are over 20, the LS-SVR clearly outperforms other beamformers tested. Owing to the steering vector mismatch, the MVDR beamformer see the desired signal as interference and fails in its operation.

The performance of the proposed beamformer in the scenario with multiple interferences is demonstrated in the third test. The steering vector mismatch is also presented. As it can be seen from Figure 3(a), the proposed algorithm performs equally well as ES and SQP when the number of interferences less than 5. When the interference numbers is increased to 8, the output SINR of the proposed LS-SVR beamformer is only 1 dB lower than that of idea beamformer. In contrast, the output SINRs of other beamformers tested are dramatically decreased due to the decrease of the available freedom degrees which are devoted to suppress the interference.

Figure 2. Scenario with limited snapshots and two interferences.**Figure 3.** Scenario with DOA mismatch and multiple interferences (a) SINR versus Number of interferences; (b) Beam-patterns, SNR = 10 dB, 4 interference.

The corresponding beampatterns of the beamformers are demonstrated in Figure 3(b), where the four interferences with DOAs of $\theta_i = [17.4^\circ, -11.5^\circ, 53.1^\circ, -23.5^\circ]$ are taken into account. It can be seen that the LS-SVR beamformer not only presents deep nulls at the DOAs of interference, but also achieves better sidelobe suppression than other beamformers tested. Thus, the proposed LS-SVR method can get better SINR performance than the usual robust linear beamforming algorithms in the case of numerous interferences.

To show the computation complexity of the novel approach, the dictionary size growth with the input samples is given in Figure 4. As it can be seen in Figure 4, only 396 center numbers are needed to calculate the beamformed output for 4,000 input samples. In comparison with the original LS-SVR algorithm, in which 4,000 centers are needed for the same case. Thus, the computation cost is largely reduced.

Figure 4. Dictionary size vs. input samples for the novel approach.

7. Conclusions

We present a novel nonlinear LS-SVR-based beamforming approach in this paper. This approach first uses a squared-loss function to replace the conventional linearly constrained minimum variance cost function, which can significantly increase robustness against mismatch problems and provide additional control over the sidelobe level. The method also applies Gaussian kernels to the array observations to improve the generalization capacity. Finally, the method uses a recursive regression procedure to estimate the weight vectors on real-time and performs mode reduction to reduce the final size of the beamformer.

The simulation tests, with steering vector mismatch, numerous interferences and limited available snapshots, are carried out to verify the performance of the proposed beamforming algorithm in comparison with other recently proposed ones. The test results show that the proposed beamforming method significantly outperforms many other recently proposed linear robust beamforming techniques in terms of signal distortion in the desired signal and noise reduction in scenarios with DOA mismatch, limited observation samples, and numerous interferences.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (Grant No.61071191) and Natural Science Foundation of Chongqing (CSTC 2011BB2048).

References

1. Van Trees, H.L. Part IV Detection, Estimation and Modulation Theory; In *Optimum Array Processing*; John Wiley & Sons, Inc.: New York, NY, USA, 2002; pp. 439–440.
2. Liu, J.; Gershman, A.B.; Luo, Z.Q.; Wong, K.M. Adaptive beamforming with sidelobe control: A second-order cone programming approach. *IEEE Signal Process. Lett.* **2003**, *10*, 331–334.
3. Li, J.; Stoica, P.; Wang, Z. Doubly constrained robust Capon beamformer. *IEEE Trans. Signal Process.* **2004**, *52*, 2407–2423.
4. Li, J.; Stoica, P. *Robust Adaptive Beamforming*; John Wiley & Sons, Inc.: New York, NY, USA, 2006.

5. Liu, C.; Liao, G. Robust capon beamformer under norm constraint. *Signal Process.* **2010**, *90*, 1573–1581.
6. Li, J.; Stoica, P.; Wang, Z. On robust capon beamforming and diagonal loading. *IEEE Trans. Signal Process.* **2003**, *51*, 1702–1715.
7. Lorenz, R.G.; Boyd, S.R. Robust minimum variance beamforming. *IEEE Trans. Signal Process.* **2005**, *53*, 1684–1696.
8. Vorobyov, S.A.; Gershman, A.B.; Luo, Z.Q. Robust adaptive beamforming using worst-case performance optimization: A solution to the signal mismatch problem. *IEEE Trans. Signal Process.* **2003**, *51*, 313–324.
9. Hassanien, A.; Vorobyov, S.A.; Wong, K.M. Robust adaptive beamforming using sequential quadratic programming: An iterative solution to the mismatch problem. *IEEE Signal Process. Lett.* **2008**, *15*, 733–736.
10. Lie, J.P.; Ser, W.; See, C.M. Adaptive uncertainty based iterative robust capon beamformer using steering vector mismatch estimation. *IEEE Trans. Signal Process.* **2011**, *59*, 4483–4488.
11. Landau, L.; de, Lamare, R.C.; Haardt, M. Robust Adaptive Beamforming Algorithms Using Low-Complexity Mismatch Estimation. In *Proceedings of the IEEE Statistical Signal Processing Workshop*, Jachranka, Poland, 28–30 June 2011; pp. 445–448.
12. Yu, Z.L.; Er, M.H. A robust minimum variance beamformer with new constraint on uncertainty of steering vector. *Signal Process.* **2006**, *86*, 2243–2254.
13. Nai, S.E.; Ser, W.; Yu, Z.L.; Rahardja, S. A robust adaptive beamforming framework with beampattern shaping constraints. *IEEE Trans. Antennas Propagat.* **2009**, *57*, 2198–2203.
14. Yu, Z.L.; Ser, W.; Er, M.H.; Gu, Z.; Li, Y. Robust adaptive beamformers based on worst-case optimization and constraints on magnitude response. *IEEE Trans. Signal Process.* **2009**, *57*, 2615–2628.
15. Martinez-Ramon, M.; Rojo-Alvarez, J.L.; Camps-Valls, G.; Christodoulou, C.G. Kernel antenna array processing. *IEEE Trans. Antennas Propagat.* **2007**, *55*, 642–650.
16. Chang, P.R.; Yang, W.H.; Chan, K.K. A neural network approach to MVDR beamforming problem. *IEEE Trans. Antennas Propagat.* **1992**, *40*, 313–322.
17. Vapnik, V.N. *The Nature of Statistical Learning Theory*; Springer-Verlag Inc.: New York, NY, USA, 1995.
18. Gaudes, C.C.; Santamaria, I.; Via, J.; Gomez, E.M.M.; Paules, T.S. Robust array beamforming with sidelobe control using support vector machines. *IEEE Trans. Signal Process.* **2007**, *55*, 574–584.
19. Ramon, M.M.; Xu, N.; Christodoulou, C.G. Beamforming using support vector machines. *IEEE Antenn. Wirel. Propag. Lett.* **2005**, *4*, 439–442.
20. Suykens, J.A.K.; De Brabanter, J.; Lukas, L.; Vandewalle, J. Weighted least squares support vector machines: Robustness and sparse approximation. *Neurocomputing* **2002**, *48*, 85–105.
21. Chu, W.; Ong, C.J.; Keerthi, S.S. An improved conjugate gradient scheme to the solution of least squares svm. *IEEE Trans. Neural Networks* **2005**, *16*, 498–501.
22. Adankon, M.M.; Cheriet, M. Model selection for LSSVM application to handwriting recognition. *Pattern Recognit.* **2009**, *42*, 3264–3270.
23. Hoegaerts, L.; Suykens, J.A.K.; Vandewalle, J.; de Moor, B. Subset based least squares subspace regression in RKHS. *Neurocomputing* **2005**, *63*, 293–323.

24. Jiao, L.; Bo, L.; Wang, L. Fast sparse approximation for least square support vector machine. *IEEE Trans. Neural Networks* **2007**, *18*, 685–697.
25. Vaerenbergh, S.V.; Via, J.; Santamaria, I. A Sliding-Window Kernel RLS Algorithm and Its Application to Nonlinear Channel Identification. In *Proceedings of the Acoustics, Speech and Signal Processing*, Toulouse, France, 14–19 May 2006; pp. 789–792.
26. Slavakis, K.; Theodoridis, S.; Yamada, I. On line classification using kernels and projection based adaptive algorithm. *IEEE Trans. Signal Process.* **2008**, *56*, 2781–2797.
27. Platt, J. A resource allocating network for function interpolation. *Neural Comput.* **1991**, *3*, 213–225.
28. Liu, W.F.; Jose, C.P.; Simon, H. *Kernel Adaptive Filtering*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2010.
29. Yu, J.L.; Yeh, C.C. Generalized eigenspace-based beamformers. *IEEE Trans. Signal Process.* **1995**, *43*, 2453–2461.
30. Selen, Y.; Abrahamsson, R.; Stoica, P. Automatic robust adaptive beamforming via ridge regression. *Signal Process.* **2008**, *88*, 33–49.

© 2012 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).