A complex version of the LASSO algorithm and its application to beamforming

José F. de Andrade Jr. and Marcello L. R. de Campos
Program of Electrical Engineering
Federal University of Rio de Janeiro (UFRJ)
Rio de Janeiro, Brazil 21.941–972, P. O. Box 68504
e-mails: andrade@dctim.mar.mil.br, campos@lps.ufrj.br

José A. Apolinário Jr.
Department of Electrical Engineering (SE/3)
Military Institute of Engineering (IME)
Rio de Janeiro, Brazil 22.290–270
e-mail: apolin@ime.eb.br

***Abstract*— Least Absolute Shrinkage and Selection Operator (LASSO) is a useful method to achieve coefficient shrinkage and data selection simultaneously. The central idea behind LASSO is to use the L_1-norm constraint in the regularization step. In this paper, we propose an alternative complex version of the LASSO algorithm applied to beamforming aiming to decrease the overall computational complexity by zeroing some weights. The results are compared to those of the Constrained Least Squares and the Subset Selection Solution algorithms. The performance of nulling coefficients is compared to results from an existing complex version named the Gradient LASSO method. The results of simulations for various values of coefficient vector L_1-norm are presented such that distinct amounts of null values appear in the coefficient vector. In this supervised beamforming simulation, the LASSO algorithm is initially fed with the optimum LS weight vector.**

Keywords*— *Beamforming; LASSO algorithm; complex-valued version; optimum constrained optimization.

I. Introduction

The goal of the Least Absolute Shrinkage and Selection Operator (LASSO) algorithm [1] is to find a Least-Squares (LS) solution such that the L_1-norm (also known as "taxi-cab" or "Manhattan" norm) of its coefficient vector is not greater than a given value, that is, $L_1\{\mathbf{w}\} = \sum_{k=1}^{N} |w_k| \leq t$. Starting from linear models, the LASSO method has been applied to solve various problems such as wavelets, smoothing splines, logistic models, etc [2]. The problem may be solved using quadratic programming, general convex optimization methods [3], or by means of the Least Angle Regression algorithm [4]. In real-valued applications, the L_1 regularized formulation is applicable to a number of contexts due to its tendency to select solution vectors with fewer nonzero components, resulting in an effective reduction of the number of variables upon which the given solution is dependent [1]. Therefore, the LASSO algorithm and its variants could be of great interest to the fields of compressed sensing and statistical natural selection.

This work introduces a novel version of the LASSO algorithm which, as opposed to an existing complex version, viz, the recently published Gradient LASSO [5], brings to the field of complex variables the ability to produce solutions with a large number of null coefficients. The paper is organized as follows: Section II presents an overview of the LASSO problem formulation. Section III-A starts by providing a version of the new Complex LASSO (C-LASSO) while the Gradient LASSO (G-LASSO) method and Subset Selection Solution (SSS) [6] are described in the sequence. In Section IV, simulation results comparing the proposed C-LASSO to other schemes are presented. Finally, conclusions are summarized in Section V.

II. Overview of the LASSO Regression

The real-valued version of the LASSO algorithm is equivalent to solving a minimization problem stated as:

$$\min_{(w_1,\cdots,w_N)} \frac{1}{2}\sum_{i=1}^{M}(y_i - \sum_{j=1}^{N} x_{ij}w_j)^2 \text{ s.t. } \sum_{k=1}^{N}|w_k| \leq t \quad (1)$$

or, equivalently,

$$\min_{(\mathbf{w}\in\mathbb{R}^N)} \frac{1}{2}(\mathbf{y}-\mathbf{X}\mathbf{w})^T(\mathbf{y}-\mathbf{X}\mathbf{w}) \text{ s.t. } g(\mathbf{w}) = t - \|\mathbf{w}\|_1 \geq 0. \quad (2)$$

The LASSO regressor, although quite similar to the shrinking procedure used in the ridge regressor [7], will cause some of the coefficients to become zero. Several algorithms have been proposed to solve the LASSO [3]- [4]. Tibshirani [1] suggests, for the case of real-valued signals, a simple, albeit not efficient, way to solve the LASSO regression problem. Let $\mathrm{s}(\mathbf{w})$ be defined as

$$\mathrm{s}(\mathbf{w}) = \mathrm{sign}(\mathbf{w}). \quad (3)$$

Therefore, $\mathrm{s}(\mathbf{w})$ is a member of set

$$\mathcal{S} = \{\mathbf{s}_j\}\ ,\ \ j = 1, \ldots, 2^N, \quad (4)$$

whose elements are of the form

$$\mathbf{s}_j = [\pm 1\ \ \pm 1\ \ \ldots\ \ \pm 1]^T. \quad (5)$$

The LASSO algorithm in [1] is based on the fact that the constraint $\sum_{k=1}^{N}|w_k| \leq t$ is satisfied if

$$\mathbf{s}_j^T\mathbf{w} \leq t = \alpha t_{LS},\ \text{ for all } j, \quad (6)$$

with $t_{LS} = \|\mathbf{w}_{LS}\|_1$ (L_1-norm of the unconstrained LS solution) and $\alpha \in (0, 1]$.

However, we do not know in advance the correct index j corresponding to $\mathrm{sign}(\mathbf{w}_{LASSO})$. At best, we can project a candidate coefficient vector onto the hyperplane defined by $\mathbf{s}_j^T\mathbf{w} = t$. After testing if the L_1-norm of the projected solution is a valid solution, we can accept the solution or move further, choosing another member of $\mathcal{S}$ such that, after a number of trials, we obtain a valid LASSO solution.

III. Complex Shrinkage Regression Methods

A. A Complex LASSO Algorithm

The original version of the LASSO algorithm, as proposed in [1], is not suitable to be used in complex-valued applications, such as beamforming. Aiming to null a large portion of the coefficient vector, we suggest to solve (1) in two steps, treating separately real and imaginary quantities. The algorithm, herein called C-LASSO for Complex LASSO, is summarized in Algorithm 1. In order to overcome some of the difficulties brought by the complex constraints, we propose that

$$\sum_{k=1}^{N} |w_k| \le t \tag{7}$$

be changed to

$$\sum_{k=1}^{N} |\mathcal{R}e\{w_k\}| \le t_R \;\text{ and} \tag{8}$$

$$\sum_{k=1}^{N} |\mathcal{I}m\{w_k\}| \le t_I. \tag{9}$$

Considering the equality condition in (7)

$$\sum_{k=1}^{N} |w_k| = t \le \sum_{k=1}^{N}\Big\{|\mathcal{R}e\{w_k\}| + |\mathcal{I}m\{w_k\}|\Big\} \le t_R + t_I, \tag{10}$$

choosing $t_R = t_I$ and the lower bound in (10), we have

$$t_R = t_I = \frac{\alpha t_{LS}}{2}. \tag{11}$$

For an array containing N sensors, where all coefficients are nonzero, consider the total number of multiplications required for calculating a beamforming output equal to $3N$. For each coefficient having null real part (or imaginary), the computational complexity is reduced from three to two multiplications. For each null coefficient, the number of multiplications is reduced from three to zero. Thus, the resulting number of multiplication operation is:

$$N_R = 3N - (N_p + 3N_z) \tag{12}$$

where N_z and N_p are the amounts of null coefficients and null coefficient parts (real or imaginary), respectively. The count of N_p excludes the null parts which had already been taken into account for N_z.

Algorithm 1 : The Complex LASSO (C-LASSO) Algorithm

Initialization
$\alpha \in (0, 1]$;
$\mathbf{X}$; % ($\mathbf{X}$ is a matrix with $N \times M$ elements)
$\mathbf{Y}$; % ($\mathbf{Y}$ is a vector with $N \times 1$ elements)

$\mathbf{w}_{LS} \leftarrow (\mathbf{X}^H\mathbf{X})^{-1}\mathbf{X}^H\mathbf{Y}$;
$t_{LS} \leftarrow \sum_{k=1}^{N} |w_k^{LS}|$;
$t_R \leftarrow \alpha t_{LS}/2$;
$t_I \leftarrow t_R$;
$\mathbf{w}^o_{Real} \leftarrow \mathcal{R}e[\mathbf{w}_{LS}]$;
$\mathbf{w}^o_{Imag} \leftarrow \mathcal{I}m[\mathbf{w}_{LS}]$;
$\mathbf{C}^c_{Real} \leftarrow \text{sign}[\mathbf{w}^o_{Real}]$;
$\mathbf{C}^c_{Imag} \leftarrow \text{sign}[\mathbf{w}^o_{Imag}]$;
$\mathbf{X}_{Real} \leftarrow \mathcal{R}e[\mathbf{X}]$;
$\mathbf{Y}_{Real} \leftarrow \mathcal{I}m[\mathbf{Y}]$;
$\mathbf{R}^{-1}_{Real} \leftarrow \frac{1}{2}(\mathbf{X}_{Real}\mathbf{X}^H_{Real})^{-1}$;
$\mathbf{R}^{-1}_{Imag} \leftarrow \frac{1}{2}(\mathbf{X}_{Imag}\mathbf{X}^H_{Imag})^{-1}$;

while $(\|\mathbf{w}^c_{Real}\|_1 > t_{Real})$ **do**
  $Col \leftarrow$ Numbers of columns of $\mathbf{C}^c_{Real}$;
  $\mathbf{f} \leftarrow \mathbf{1}_{(Col\times 1)} \times t_R, \quad \mathbf{1}_{(Col\times 1)} = [1\ 1 \ldots 1]^T_{(Col\times 1)}$;
  $\mathbf{w}^c_{Real} \leftarrow \mathbf{w}^o_{Real} - \mathbf{R}^{-1}_{Real}\mathbf{C}^c_{Real}\Big((\mathbf{C}^c_{Real})^H\mathbf{R}^{-1}_{Real}\mathbf{C}^c_{Real}\Big)^{-1}\Big((\mathbf{C}^c_{Real})^H\mathbf{w}^o_{Real} - \mathbf{f}\Big)$;
  $\mathbf{C}_{Real} \leftarrow [\mathbf{C}_{Real} \;\; \text{sign}[\mathbf{w}^c_{Real}]]$;
end while

while $(\|\mathbf{w}^c_{Imag}\|_1 > t_{Imag})$ **do**
  $Col \leftarrow$ Numbers of columns of $\mathbf{C}^c_{Imag}$;
  $\mathbf{f} \leftarrow \mathbf{1}_{(Col\times 1)} \times t_I, \quad \mathbf{1}_{(Col\times 1)} = [1\ 1 \ldots 1]^T_{(Col\times 1)}$;
  $\mathbf{w}^c_{Imag} \leftarrow \mathbf{w}^o_{Imag} - \mathbf{R}^{-1}_{Imag}\mathbf{C}^c_{Imag}\Big((\mathbf{C}^c_{Imag})^H\mathbf{R}^{-1}_{Imag}\mathbf{C}^c_{Imag}\Big)^{-1}\Big((\mathbf{C}^c_{Imag})^H\mathbf{w}^o_{Imag} - \mathbf{f}\Big)$;
  $\mathbf{C}_{Imag} \leftarrow [\mathbf{C}_{Imag} \;\; \text{sign}[\mathbf{w}^c_{Imag}]]$;
end while

$\mathbf{w}_{LASSO} \leftarrow \mathbf{w}^c_{Real} + j\mathbf{w}^c_{Imag}$.

B. The Subset Selection Solution Procedure

The Subset Selection (SS) solution [6] is obtained from the ordinary least squares solution (LS) after forcing the smallest coefficients (in magnitude) to be equal to zero. In order to compare the beam patterns produced by the SS and C-LASSO algorithms, the number of zero real and imaginary parts obtained by C-LASSO algorithm are taken into account to calculate the coefficients using the SS algorithm. The SS algorithm is summarized in Algorithm 2.

Algorithm 2 : The Subset Selection (SS) Solution

Initialization: $\mathbf{w}_{LS}$, $N_{Z_{Real}}$, $N_{Z_{Imag}}$;
$[\mathbf{w}_{SSS_{Real}}, \mathbf{pos}_{Real}] \leftarrow$ Sort($|\mathbf{w}_{LS_{Real}}|$);
$[\mathbf{w}_{SSS_{Imag}}, \mathbf{pos}_{Imag}] \leftarrow$ Sort($|\mathbf{w}_{LS_{Imag}}|$);
for $i = 1$ to $N_{Z_{Real}}$ **do**
  $w_{SSS_{Real}}(i) \leftarrow 0$
end for
for $k = 1$ to $N_{Z_{Imag}}$ **do**
  $w_{SSS_{Imag}}(k) \leftarrow 0$;
end for
$\mathbf{w}_{SSS_{Real}}(\mathbf{pos}_{Real}) \leftarrow \mathbf{w}_{SSS_{Real}}$;
$\mathbf{w}_{SSS_{Imag}}(\mathbf{pos}_{Imag}) \leftarrow \mathbf{w}_{SSS_{Imag}}$;
$\mathbf{w}_{SSS_{Real}} \leftarrow \mathbf{w}_{SSS_{Real}} .* sign(\mathbf{w}_{SSS_{Real}})$;
$\mathbf{w}_{SSS_{Imag}} \leftarrow \mathbf{w}_{SSS_{Imag}} .* sign(\mathbf{w}_{SSS_{Imag}})$;
$\mathbf{w}_{SSS} \leftarrow \mathbf{w}_{SSS_{Real}} + j\mathbf{w}_{SSS_{Imag}}$;

C. The Gradient LASSO Algorithm

The gradient LASSO (G-LASSO) algorithm, summarized in Algorithm 3, is computationally more stable than quadratic programming (QP) schemes because it does not require matrix inversions; thus, it can be more easily applied to higher-dimensional data. Moreover, the G-LASSO algorithm is guaranteed to converge to the optimum under mild regularity conditions [2].

Algorithm 3 : The G-LASSO Algorithm

Initialization: $\mathbf{w} = \mathbf{0}$ and $m = 0$;
while $(\mathbf{not}(\mathbf{coverge}))$ **do**
  $m \leftarrow m + 1$;
  Compute the gradient $\nabla(\mathbf{w}) \leftarrow (\nabla(\mathbf{w})^1, ..., \nabla(\mathbf{w})^d)$;
  Find the $(\hat{l}, \hat{k}, \hat{\gamma})$ which minimizes $\gamma\nabla(\mathbf{w})_k^l$ for $l = 1, \ldots, d$, $k = 1, \ldots, p_l$, $\gamma = \pm 1$;
  Let $v^{\hat{l}}$ be the $p^{\hat{l}}$ dimensional vector such that the $\hat{k}$-th element is $\hat{\gamma}$ and the other column elements are 0;
  Find $\hat{\beta} = argmin_{\beta\in[0,1]} C(\mathbf{w}[\beta, v^{\hat{l}}])$;
  Update $\mathbf{w}$:

$$w_k^l \leftarrow \begin{cases} (1-\hat{\beta})w_k^l + \hat{\gamma}\hat{\beta} & l = \hat{l}, k = \hat{k} \\ (1-\hat{\beta})w_k^l & l = \hat{l}, k \neq \hat{k} \\ w_k^l & otherwise \end{cases}$$

end while

return $\mathbf{w}$.

IV. Simulation Results

In this section, we present the results of simulations for various values of α such that distinct numbers of null coefficients arise in the coefficient vector $\mathbf{w}$. The simulations were performed from a supervised modeling beamforming and the LASSO algorithm was initially fed with the optimum weight vector $\mathbf{w}_{opt}$. Subsection IV-A describes, briefly, the signal model used to calculate $\mathbf{w}_{opt}$ and $\mathbf{w}_{LASSO}$. The results are presented in Subsection IV-B

A. Signal Model

Consider a uniform linear array (ULA) composed by N receiving antennas (sensors) and q receiving narrowband signals coming from different directions $\phi_1, \cdots, \phi_q$. The output signal observed from N sensors during M snapshots can be denoted as $\mathbf{x}(t_1), \mathbf{x}(t_2), \cdots, \mathbf{x}(t_M)$. The $N \times 1$ signal vector is then written as:

$$\mathbf{x}(t) = \sum_{k=1}^{q} \mathbf{a}(\phi_k)\mathbf{s}_k(t) + \mathbf{n}(t) \ , \ t = t_1, t_2, \cdots, t_M, \quad (13)$$

or, using matrix notation,

$$\mathbf{X} = [\mathbf{x}(t_1) \ \mathbf{x}(t_2) \ \cdots \ \mathbf{x}(t_M)] = \mathbf{AS} + \boldsymbol{\eta}, \quad (14)$$

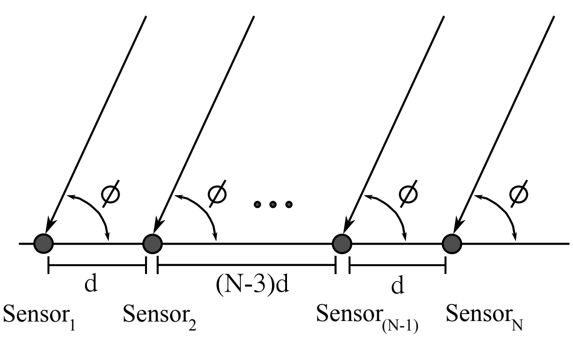

Figure 1. Narrowband Signal Model for Uniform Linear Array (ULA) for snapshot at instant t.

where matrix $\mathbf{X}$ is the input signal matrix of dimension $N \times M$. Matrix $\mathbf{A} = [\mathbf{a}(\phi_1), \cdots, \mathbf{a}(\phi_q)]$ is the steering matrix of dimension $N \times q$, whose columns are denoted by

$$\boldsymbol{a}(\phi) = [1, e^{-j(2\pi/\lambda)d\cos(\phi)}, \cdots, e^{-j(2\pi/\lambda)d(N-1)\cos(\phi)}]^T. \quad (15)$$

$\mathbf{S}$, in (14), is a $q \times M$ signal matrix, whose lines refer to snapshots. $\boldsymbol{\eta}$ is the noise matrix of dimension $N \times M$; λ and d are the wavelength of the signal and the distance between antenna elements (sensors), respectively.

Based on above definition, the covariance matrix $\mathbf{R}$, defined as E$[\mathbf{x}(t)\mathbf{x}^H(t)]$, can be estimated as

$$\hat{\mathbf{R}} = \mathbf{XX}^H = \mathbf{ASS}^H\mathbf{A}^H + \boldsymbol{\eta}\boldsymbol{\eta}^H, \quad (16)$$

which, except for a multiplicative constant, corresponds to the time average

$$\hat{\mathbf{R}} = \frac{1}{M}\sum_{k=1}^{M}\mathbf{x}(t_k)\mathbf{x}^H(t_k). \tag{17}$$

From the formulation above and imposing linear constraints, it is possible to obtain a closed-form expression to $\mathbf{w}_{opt}$ [8], the LCMV solution

$$\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{C}\left(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C}\right)^{-1}\mathbf{f}, \tag{18}$$

such that $\mathbf{C}^H\mathbf{w}_{opt} = \mathbf{f}$, $\mathbf{C}$ and $\mathbf{f}$ are given by [9]

$$\mathbf{C} = [1, e^{-j\pi\cos(\phi)}, \cdots, e^{-j\pi(N-1)\cos(\phi)}]^T \tag{19}$$

and

$$\mathbf{f} = 1. \tag{20}$$

B. The Results

Experiments were conducted where we designed ULA beamformers with 50 and 100 sensors. C-LASSO beam patterns were compared to SSS and CLS beam patterns for α values equal to 0.05, 0.10, 0.250, and 0.50. We also compared the capacity of shrinking of the C-LASSO to the G-LASSO algorithms.

The simulations presented in this section were carried out considering the direction of arrival of the desired signal $\theta = 120°$. The four interfering signals used in the simulations employed power levels such that Interference-to-Noise Ratios (INR) were $30, 20, 30, 30$ dB. The number of snapshots was $6,000$.

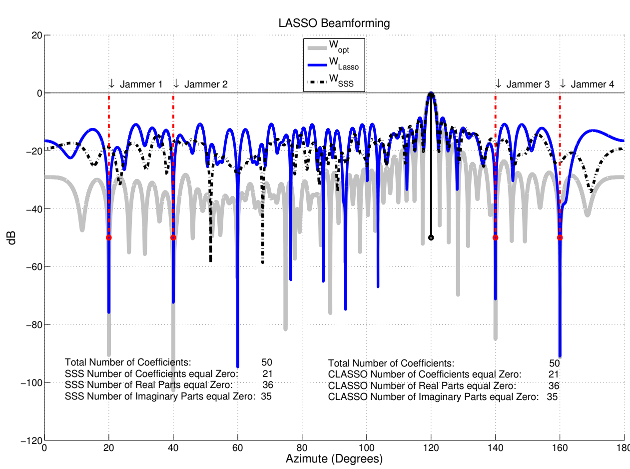

Figure 2. Beam pattern calculated for $\theta = 120°$, $N = 50$, and $\alpha = 0.05$.

Figure 2 compares the beam patterns from the CLS, the SS, and the C-LASSO algorithms, where it is possible to realize that the constraints relating to interference are satisfied by the C-LASSO coefficients. In this experiment, about 42% of the coefficients are zero and the percentage for the real and imaginary parts is around 70%.

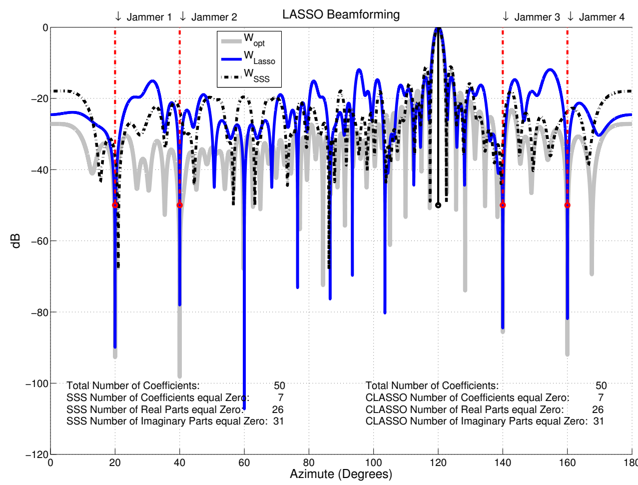

Figure 3. Beam pattern calculated for $\theta = 120°$, $N = 50$, and $\alpha = 0.1$.

In the simulation presented in Figure 3, only about 14% of the coefficients are zero, although the percentage for the real and imaginary parts is around 50%.

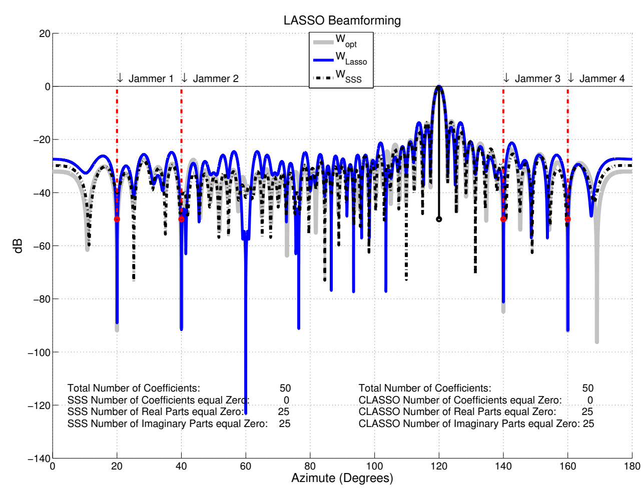

Figure 4. Beam pattern calculated for $\theta = 120°$, $N = 50$, and $\alpha = 0.25$.

In the simulations whose results are presented in Figures 4 and 5, no coefficient is forced to zero (both real and imaginary parts) although real or imaginary parts of some coefficients were zeroed, representing 50% of the total.

Figure 5 shows that, for $\alpha = 0.50$ and $N = 100$, the C-LASSO and the SS solution beam patterns approximate the optimal case (CLS), even with 50% of real and imaginary parts equal to zero.

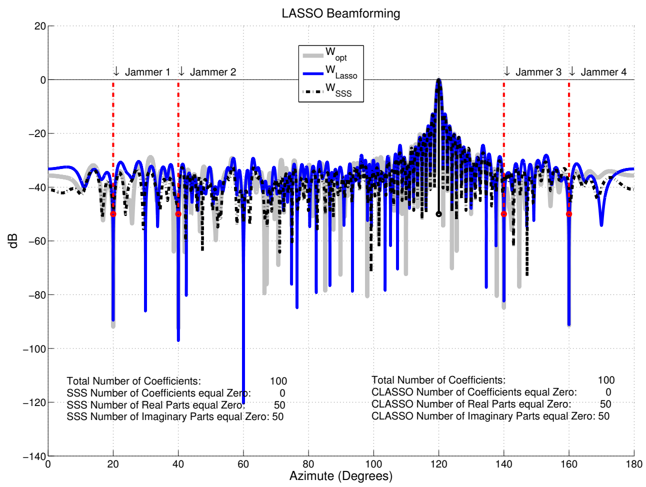

Figure 5. Beam pattern calculated for $\theta = 120°$, $N = 100$, and $\alpha = 0.50$.

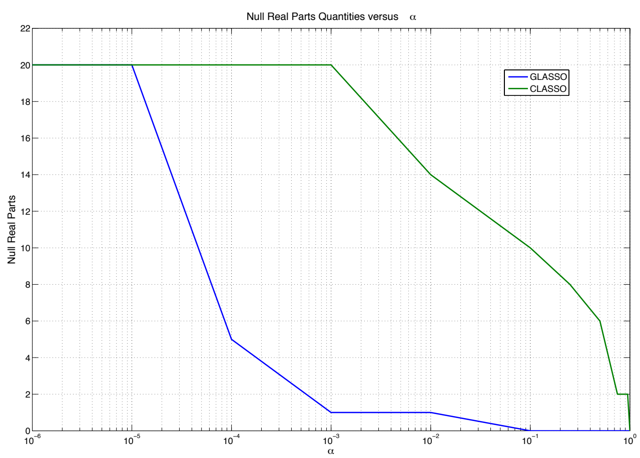

Figure 6. Null Real Parts Quantities versus α for $N = 20$.

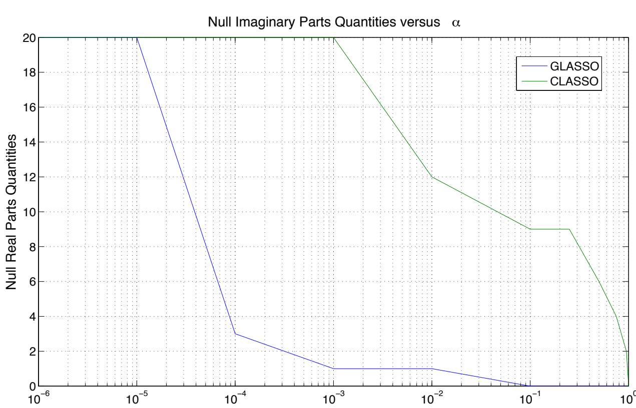

Figure 7. Null Imaginary Parts Quantities versus α for $N = 20$.

Based on the results of a second experiment, whose simulations are presented in Figures 6 and 7, we can say that the algorithm C-LASSO is more efficient than the algorithm G-LASSO in terms of coefficient shrinkage, with regards to forcing coefficients (or their real or imaginary parts) to become zero.

V. Conclusions

In this article, we proposed a novel complex version for the shrinkage operator named C-LASSO and explored shrinkage techniques in computer simulations. We compared beam patterns obtained using the algorithms C-LASSO, SS, and CLS. Based on the results presented in the previous section, it is clear that the C-LASSO algorithm can be employed to solve problems related to beamforming with reduced computational complexity.

Although, for very low values of α, the radiation pattern presents extra secondary lobes, which decrease the SNR in the receiver, it is noted that the constraints concerning jammers continue to be satisfied by the C-LASSO algorithm, making this technique ideal for anti-jamming systems [10].

One important area to apply and test this kind of statistical selection algorithm would be in the field of electronic warfare [10], where miniaturization, short delay, and low power circuits are required.

Acknowledgments

The authors thank the Brazilian Army, the Brazilian Navy, and the Brazilian National Research Council (CNPq), under contracts 306445/2007-7 and 478282/2007-9, for partially funding of this work.

References

[1] R. Tibshirani,"Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society (Series B)*, vol. 58, no. 1, pp. 267–288, May 1996.

[2] K. Jinseog, K. Yuwon and K. Yongdai, "Gradient LASSO algorithm," *Technical report, Seoul National University*, URL http://idea.snu.ac.kr/Research/papers/GLASSO.pdf., 2005.

[3] M. R. Osborne, B. Presnell and B. A. Turlach,"On the LASSO and Its Dual," *Journal of Computational and Graphical Statistics*, vol. 9, pp. 319–337, May 1999.

[4] B. Efron, T. Hastie, I. Johnstone and R. Tibshirani, "Least angle regression," *Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[5] E. van den Berg and M. P. Friedlander, "Probing the Pareto frontier for basis pursuit solutions," *SIAM Journal on Scientific Computing*, vol. 31, no. 2, pp. 890–912, 2008.

[6] M. L. R. de Campos and J. A. Apolinário Jr, "Shrinkage methods applied to adaptive filters," *Proceedings of the International Conference on Green Circuits and Systems*, June 2010.

[7] A. E. Hoerl and R. W. Kennard,"Ridge Regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, February 1970.

[8] H. L. Van Trees, *Optimum array processing*, 1st Ed., vol. 4, Wiley-Interscience, March 2002.

[9] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementations*, 3rd Ed., Springer, 2008.

[10] M. R. Frater, *Electronic Warfare for Digitized Battlefield*, 1st Ed., Artech Printer on Demand, October 2001.