Adaptive RLS Filters with Linear and Quadratic Constraints

J. Tuthill, Y. H. Leung and I. Thng

Australian Telecommunications Research Institute, Curtin University of Technology, Bentley, WA 6102, Australia

ABSTRACT

In many adaptive signal processing applications, it is often desired to impose linear and quadratic constraints on the adaptive filter weights in order to meet certain performance criteria. This paper presents a modification of the well known adaptive RLS or FLS algorithm to achieve this. By way of illustration, the paper considers an adaptive narrowband beamformer with first and second order spatial derivative constraints. The performance of the algorithm is studied via computer simulations.

1. INTRODUCTION

In antenna array processing it is often required to minimise the array processor mean output power subject to a fixed response in the array look direction [1]. The look direction requirement can be met by imposing a set of linear constraints on the processor weights to yield what is known as the *Linearly Constrained Minimum Variance* (LCMV) processor. It has been found, however, that LCMV processors are susceptible to errors in the assumed direction of arrival of the desired signal. To achieve robustness against directional mismatch, additional constraints known as *derivative constraints* can be introduced. These constraints force the first and second order spatial derivatives of the array power response in the look direction to zero [2]. However derivative constraints corresponding to necessary and sufficient conditions for these spatial derivatives to be zero are in general quadratic, and the resulting weight vector solution space is non-convex [3]. The past approach to this complex problem has been to consider conditions which are only sufficient for the spatial derivatives to be zero. Whilst this results in linear constraints, it nevertheless exhibits certain anomalous behaviour, e.g. dependence on the choice of array phase centre [4].

In [3], a method for solving the non-convex output power minimisation problem with quadratic derivative constraints is presented. However, the problem solved in [3] is an optimisation problem which assumes that the signal autocorrelation matrix $\boldsymbol{R}$ is known. In practice $\boldsymbol{R}$ must be estimated from the receiver data. In this paper, we present an adaptive algorithm for determining the optimum weight, in the least squares sense, from measured data. This algorithm is essentially the well-known adaptive Recursive Least Squares (RLS) filter [5] but satisfying also linear and quadratic constraints.

Although this paper deals only with a narrowband planar array operating in 2D space, the technique developed herein can be easily extended to 3D space operation as well as broadband array processor applications.

2. SECOND ORDER OPTIMUM DERIVATIVE CONSTRAINED ARRAY PROCESSORS

In order that a narrowband array processor have a first and second order maximally flat spatial power response in the look direction θ_o, it can be shown that [6], in a 2D scenario, the following non-linear optimisation problem results.

$$\min_{\boldsymbol{w}_c} P(\boldsymbol{w}_c) = \boldsymbol{w}_c^H \boldsymbol{R}_c \boldsymbol{w}_c \tag{1}$$

subject to
$$\boldsymbol{s}^H(\theta_o)\boldsymbol{w}_c = 1 \tag{2}$$

$$\mathrm{Re}\left(\boldsymbol{s}_1^H(\theta_o)\boldsymbol{w}_c\right) = 0 \tag{3}$$

and
$$\mathrm{Re}\left(\boldsymbol{s}_2^H(\theta_o)\boldsymbol{w}_c\right) + \left[\mathrm{Im}\left(\boldsymbol{s}_1^H(\theta_o)\boldsymbol{w}_c\right)\right]^2 = 0 \tag{4}$$

where the array consists of L elements, $P(\boldsymbol{w}_c)$ is the mean output power, $\boldsymbol{w}_c \in C^L$ is the complex weight vector, $\boldsymbol{s}(\theta_o) \in C^L$ is the steering vector for the look direction $\theta = \theta_o$, $\boldsymbol{s}_1(\theta_o) \in C^L$ and $\boldsymbol{s}_2(\theta_o) \in C^L$ are, respectively, the first and second order derivatives of the steering vector wrt θ evaluated at $\theta = \theta_o$, and $\boldsymbol{R}_c \in C^{L \times L}$ is the complex input signal correlation matrix.

Equation (2) corresponds to the look direction constraint while (3) and (4) correspond to the additional first and second order derivative constraints.

3. TRANSFORMATION TO A TWO STAGE MINIMISATION PROBLEM

Noting that the terms $\left[\mathrm{Re}\left(\boldsymbol{s}_2^H(\theta_o)\boldsymbol{w}_c\right)\right]$ and $\left[\mathrm{Im}\left(\boldsymbol{s}_1^H(\theta_o)\boldsymbol{w}_c\right)\right]$ in (4) are scalars and making the assignment

$$\left[\mathrm{Im}\left(\boldsymbol{s}_1^H(\theta_o)\boldsymbol{w}_c\right)\right] = \alpha \text{ and } \left[\mathrm{Re}\left(\boldsymbol{s}_2^H(\theta_o)\boldsymbol{w}_c\right)\right] = -\alpha^2 \tag{5}$$

where $\alpha \in \mathcal{R}$ is some unspecified scalar, it can be shown [3], [6] that the quadratically constrained optimisation problem of (1) to (4) can be re-formulated as a two-stage minimisation problem where the first stage is a linearly constrained optimisation problem, parameterised by α, and the second-stage is an unconstrained optimisation of α:

$$\min_{\alpha} \begin{bmatrix} \min_{\boldsymbol{w}} \ \boldsymbol{w}^T \boldsymbol{R} \boldsymbol{w} \\ \text{subject to} \quad \boldsymbol{C}^T \boldsymbol{w} = \boldsymbol{h}(\alpha) \end{bmatrix} \tag{6}$$

where $\boldsymbol{C} \in \mathcal{R}^{L \times M}$ is a stacked constraint matrix formed from (2), (3) and (5), $\boldsymbol{h}(\alpha) \in \mathcal{R}^M$ is the corresponding constraint vector, For time-varying polynomial coefficients, as encountered in many adaptive signal processing applications, the coefficients and roots often do not change by large amounts from one sampling instant to the next. Since global methods such as (14) exhibit fast convergence in the vicinity of the actual roots it is an obvious advantage to initialise the root-finding algorithm each time with the previous root estimates. However to ensure convergence of the algorithm to the actual roots a small, artificial complex component is added to one of the previous root estimates before the algorithm is started. This serves to destroy any symmetry about the real axis that may exist in the initial guesses [11], [12]. The existence of such symmetry will prevent the algorithm from finding the actual roots in the following two situations: (i) the initial root estimates are all real whereas two of the actual roots occur in complex conjugate pairs; and (ii) two of the initial root estimates occur in a complex conjugate pair whereas all of the actual roots are real.

The adaptive RLS filter with linear and quadratic constraints is summarised in Table 1.

Note that the algorithm requires $R(n)$ to be updated (see (15.2)). Note also that in Step 3, we use the stabilised RLS algorithm of [13]. We could have also use the stabilised FLS algorithm of [14]. The salient feature of the algorithm presented in [8] is that it incorporates a self-correction for $Q(n)$ which ensures that the constraints are satisfied at each iteration. This is important in a finite precision environment where round-off error can cause the weight vector solution to drift out of the constraint space.

5. NUMERICAL STUDIES

The following simulation results were obtained using the algorithm as it is presented in equations (15.1) to (15.9).

The simulation scenario assumes a narrowband, uniformly distributed 5-element circular array with the phase centre located at the centre of the circle and one element positioned on the positive y-axis. The circle radius was set at two wavelengths. A 6dB random Gaussian source was located at 30° from the x-axis and the look direction was set at 105°. The element self-noise is -30 dB. The data correlation matrix was initialised with $\delta = 10$ in equation (15.1) and the forgetting factor was set to $\lambda = 0.99$.

The learning curve for the derivative constrained adaptive array processor averaged over 100 separate runs is displayed in Fig. 1. Note that the output power converges to the optimum value of -33.26 dB which can be found from the corresponding optimum processor. Note also that the algorithm does not exhibit numerical instabilities.

For comparison, the learning curve of the array processor subject to the "sufficient" linear derivative constraints is also shown in Fig. 1 (upper plot). These results were obtained using the same simulation scenario and input data as above. Note that this processor converges to a higher output power (-23.219 dB).

Fig. 2 shows the trajectories taken by the roots of the polynomial

Initialise

$$
\begin{aligned}
&R(0) = \delta \times I_{2L} \\
&R^{-1}(0) = \delta^{-1} \times I_{2L} \\
&\delta << 0.01\sigma_x^2 \text{ and } I_{2L} \text{ is a } 2L \times 2L \text{ Identity Matrix} \\
&Q(0) = R^{-1}(0)C\left(C^T R^{-1}(0)C\right)^{-1}
\end{aligned}
\quad (15.1)
$$

For each new data sample $x(n)$ at time n:

1. Update R:

$$R(n) = \lambda R(n-1) + (1-\lambda)x(n)x^T(n) \quad (15.2)$$

2. Compute the adaptation gain vector:

$$g(n) = \frac{R^{-1}(n-1)x(n)}{\left[\lambda/(1-\lambda)\right] + x^T(n)R^{-1}(n-1)x(n)} \quad (15.3)$$

3. Update $R^{-1}(n)$ with the stabilised RLS algorithm of [13].
4. Update the matrix $Q(n)$:

$$u(n) = C^T g(n) \quad (15.4)$$

$$v^T(n) = x^T(n)Q(n-1) \quad (15.5)$$

$$Q'(n) = \left[Q'(n-1) - g(n)v^T(n)\right]\left[I_M + \frac{u(n)v^T(n)}{1 - v^T(n)u(n)}\right] \quad (15.6)$$

$$Q(n) = Q'(n-1) + C\left(C^T C\right)^{-1}\left[I_M - C^T Q'(n-1)\right] \quad (15.7)$$

5. Compute the matrix $\Gamma(n)$:

$$\Gamma(n) = \left(C^T C\right)^{-1} C^T R(n) Q(n) \quad (15.8)$$

6. Compute the coefficients of the polynomial $P'(\alpha) = \frac{d}{d\alpha}P(\alpha)$ from the elements of $\Gamma(n)$.

7. Update the roots of $P'(\alpha)$ using (14) to identify the α that minimises $P(\alpha)$. Denote this α as $\alpha_{opt}(n)$

8. Compute the weight vector

$$w(n) = Q(n)h\left(\alpha_{opt}(n)\right) \quad (15.9)$$

Table 1. Quadratically Constrained RLS Filter

$P'(\alpha)$ as the root-finding algorithm of (14) converges. The plot was obtained from one of the simulation runs at n = 1 where no previous root estimates were available. The initial guesses in this case were chosen to be

$w \in \mathcal{R}^{2L}$ is the weight vector, $R \in \mathcal{R}^{2L \times 2L}$ is the real signal correlation matrix and M is the number of resulting linear constraints.

Note that in the minimisation problem (6), we have made use of the real signal representation of [7].

Using the method of Lagrange multipliers, the inner stage of the two-stage minimisation problem (6) has solution

$$w_{opt}(\alpha) = R^{-1}C\left(C^T R^{-1} C\right)^{-1} h(\alpha) \tag{7}$$

Note the optimum α is yet to be determined.

The outer stage of the two-stage minimisation problem of (6) is now given by

$$\min_{\alpha} P(\alpha) = h^T(\alpha)\left(C^T R^{-1} C\right)^{-1} h(\alpha) \tag{8}$$

It can be verified that $P(\alpha)$ is a quartic polynomial in α [6]. Thus given R^{-1} and C the optimum α can be found by computing the roots of a cubic polynomial.

Substituting the optimum α into (7) and (8) gives finally the optimum weight vector w_{opt} and output power P_{opt}.

4. ADAPTIVE ALGORITHM

Clearly in a real-time implementation, R^{-1} is not known *a priori* and must be estimated. In [8] an adaptive algorithm is presented for solving a linearly constrained power minimisation problem. A recursive update was given for estimating the matrix $R^{-1}C\left(C^T R^{-1} C\right)^{-1}$ (see (7) above) which was referred to as $Q(n)$.

The key to solving the two-stage minimisation problem, however, is in obtaining an estimate of the matrix $\left(C^T R^{-1} C\right)^{-1}$. It is from this estimate the optimum α is found from (8).

Pre-multiplying $Q(n)$ by $R(n)$ and $\left(C^T C\right)^{-1} C^T$ gives

$$\begin{aligned} \Gamma(n) &= \left(C^T C\right)^{-1} C^T R(n) Q(n) \\ &= \left(C^T C\right)^{-1} C^T R(n)\left[R^{-1}(n) C\left(C^T R^{-1}(n) C\right)^{-1}\right] \\ &= \left[\left(C^T R^{-1}(n) C\right)^{-1}\right] \end{aligned} \tag{9}$$

The matrix $\left(C^T C\right)^{-1} C^T$ is known *a priori* and can be pre-computed. In contrast, the correlation matrix $R(n) \in \mathcal{R}^{2L \times 2L}$ is unknown *a priori* and can be estimated using the standard exponentially weighted update [5]

$$R(n) = \lambda R(n-1) + (1-\lambda) x(n) x^T(n) \tag{10}$$

where $x(n) \in \mathcal{R}^{2L}$ is the real input signal vector [7] and λ is the forgetting factor, usually chosen so that $0 << \lambda < 1$.

The matrix $\Gamma(n)$ thus determined can now be used in (8) to find the coefficients of the quartic polynomial in α. To find the optimum alpha $\alpha_{opt}(n)$ at time n the zeros of the first derivative of $P(\alpha)$ wrt α need to be found. This can be done using conventional factorisation algorithms, eg. Mullers method [9], or, since $P'(\alpha)$ is only a third order polynomial an exact closed-form solution may be used. However, for higher order polynomials which may be encountered for example in broadband array processors, the conventional techniques have some shortcomings [10]: (i) each zero of the polynomial must be estimated to the required accuracy using a separate iterative process, (ii) the process of deflation used in these techniques is prone to numerical inaccuracies from round-off error accumulation, and (iii) for slight changes in the polynomial coefficients the entire factorisation process needs to be re-started.

Consequently, techniques known as global methods [10], [11] which determine all the zeros of a polynomial simultaneously using an iterative process have been developed. There are a number of global methods available offering different convergence properties and degrees of computational complexity. The third order global method, described briefly below, was used in the computer studies which follow in Section 5.

Consider the general rth order complex valued polynomial in z,

$$f(z) = z^r + a_1 z^{r-1} + \cdots + a_{r-1} z + a_r \tag{11}$$

where $a_i, z \in C$. Let $\lambda_j \in C, j = 1,2,\ldots,r$, be the actual zeroes of $f(z)$ and successive approximations to these zeros be represented by $\lambda_j(k), \lambda_j(k+1), \ldots$ with initial guesses $\lambda_j(0), j = 1,2,\ldots,r$.

Define the functions

$$u\left(\lambda_j(k)\right) = \frac{f\left(\lambda_j(k)\right)}{f'\left(\lambda_j(k)\right)} \tag{12}$$

and

$$T\left(\lambda_j(k)\right) = \sum_{i=1; i \neq j}^{r} \frac{1}{\left(\lambda_j(k) - \lambda_i(k)\right)} \tag{13}$$

The third order global method updates the root estimates at time n by

$$\lambda_j(k) = \lambda_j(k-1) - \frac{u\left(\lambda_j(k-1)\right)}{1 - T\left(\lambda_j(k-1)\right) u\left(\lambda_j(k-1)\right)} \tag{14}$$

$$j = 1,2,\ldots,r$$

The determination of the roots of $P'(\alpha)$ using this algorithm, then, is a two stage process: (i) as new input data arrives update the coefficients of $P'(\alpha)$ using the matrix $\Gamma(n)$, then (ii) iterate (14) until the desired accuracy is reached.

$$\lambda_1(0) = 1 + 0.1i; \;\; \lambda_2(0) = 1 + i; \;\; \lambda_3(0) = 1 - i \tag{16}$$

Note that the algorithm converges to the values $\{-0.00085, -7.65946, 7.65501\}$ in 8 iterations. If good initial guesses are provided, as is the case for $n > 1$ when a set of previous root estimates is available, then the algorithm typically converges in only two or three iterations.

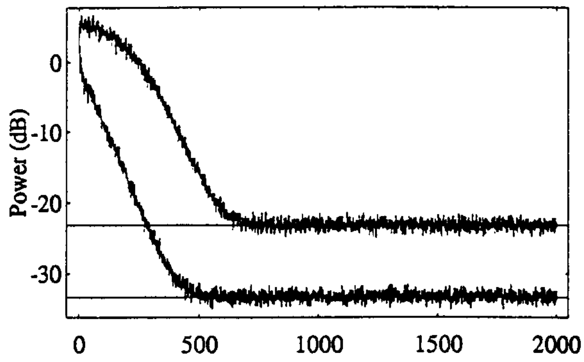

Figure 1. Ensemble averaged array output power.

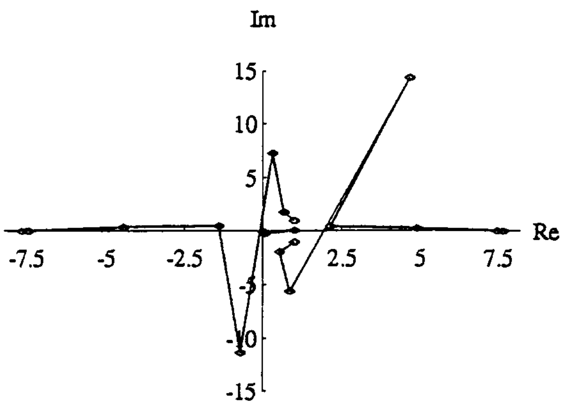

Figure 2. Trajectories taken by the roots on the complex plane as the global root finding algorithm converges.

6. CONCLUSIONS

This paper has formulated an adaptive RLS algorithm with linear and quadratic constraints. The algorithm allows an adaptive array processor with 1st and 2nd order NS derivative constraints to be implemented. This array processor overcomes some of the shortcomings evident in array processors constrained by the "sufficient" linear derivative constraints.

The authors note that as the update for $Q(n)$ is still essentially a variant of the RLS algorithm, it can suffer from numerical instability problems when implemented in finite precision arithmetic [5], [13]. The authors are currently investigating techniques for improving the numerical stability of $Q(n)$.

REFERENCES

[1] O. L. Frost, III, "An Algorithm for Linearly Constrained Adaptive Array Processing", *Proc. IEEE*, vol. 60, no. 8, pp. 926-935, August 1972.

[2] M. H. Er and A. Cantoni, "Derivative Constraints for Broad-Band Element Space Antenna Array Processors", *IEEE Trans. Acoust. Speech Signal Processing*, vol. 31, no. 6, pp. 1378-1393, December 1983.

[3] I. Thng, A. Cantoni and Y. H. Leung, "Derivative Constrained Optimum Broadband Antenna Arrays", *IEEE Trans. Signal Processing*, vol. 41, no. 7, pp. 2367-2377, July 1993.

[4] K. M. Buckley and L. J. Griffiths, "An Adaptive Sidelobe Canceller with Derivative Constraints", *IEEE Trans. Antennas Propagat.*, vol. 34, no. 3, pp. 311-319, March 1986.

[5] S. Haykin, *Adaptive Filter Theory*, 2nd Ed., Prentice Hall, Englewood Cliffs, NJ, 1991.

[6] C. Y. Tseng, "Minimum Variance Beamforming with Phase-Independent Derivative Constraints", *IEEE Trans. Antennas Propagat.*, vol. 40, no. 3, pp. 285-294, March 1992.

[7] R. T. Compton, Jr., *Adaptive Antennas*, Prentice Hall, Englewood Cliffs, NJ, 1988.

[8] L. S. Resende, J. M. T. Romano and M. G. Bellanger, "A Robust Algorithm For Linearly-Constrained Adaptive Filtering", *Proc. ICASSP*, Adelaide, Australia, pp. III 381- III 384, April 1994.

[9] S. D. Conte and C. de Boor, *Elementary Numerical Analysis: An Algorithmic Approach*, McGraw-Hill, New York, 1972.

[10] D. Starer and A. Nehorai, "Adaptive Polynomial Factorisation by Coefficient Matching", *IEEE Trans. Signal Processing*, vol. 39, no. 2, pp. 527 - 530, February 1991.

[11] M. Igarashi, "Some Remarks for the Methods to Find All the Zeros of a Polynomial Simultaneously", *Topics in Polynomials of One and Several Variables and their Applications*, T. M. Rassias, H. M. Srivastava and A. Yanushauskas (Ed.), World Scientific Publ. Co., Singapore, pp. 273 - 285, 1993.

[12] J. Tuthill, "On Global Methods for finding the Roots of a Polynomial Simultaneously", ASPL Report, Curtin University of Technology, Bentley, Australia.

[13] G E Bottomley, S T Alexander, "A Novel Approach for Stabilising Recursive Least Squares Filters", *IEEE Trans. Signal Processing*, vol. 39, no. 8, pp. 1770-1779, August 1991.

[14] D. T. M. Slock and T. Kailath, "Numerically Stable Fast Transversal Filters for Recursive Least Squares Adaptive Filtering", *IEEE Trans. Signal Processing*, vol. 39, no. 1, pp. 92-114, January 1991.