Optimal Toeplitz Completion of Covariance Matrix for Robust DOA Estimation

Carl-Inge C. Nilsen, *Member, IEEE,* Ines Hafizovic, *Student Member, IEEE*

***Abstract*—Redundancy averaging has been suggested as a method for robust covariance matrix estimation, but it has certain disadvantages. It results in biased DOA estimates, false sources, and is only applicable to uniform linear and rectangular arrays. In addition, its current derivation is somewhat ad-hoc and does not generalize easily. In this paper we introduce a generalized redundancy averaging method from an alternative derivation.**

I. Generalized Redundancy Averaging

A. Signal Model

Assume a 2D wavefield in the (x, y)-plane consisting of P monochromatic planewaves. The plane waves have identical wavelengths λ, random complex amplitude-and-phase values $A_p \in \mathbb{C}$, and distinctly different directions of arrival θ_p. The wavenumber vector of a wave with DOA θ is defined as:

$$\vec{k}(\theta) = [k_x(\theta), k_y(\theta)]^T = \frac{2\pi}{\lambda}[\sin\theta, \cos\theta]^T \tag{1}$$

As there is only one real parameter of interest, θ, we can discard either of the wavenumber vector components. We will therefore choose k_x, and simply refer to it as $k(\theta)$.The actual observable wavefield along the x-axis of the coordinate system is given as:

$$s(x) = \sum_{p=0}^{P-1} A_p e^{ik(\theta_p)x}. \tag{2}$$

B. Array Model and Beamforming

In reality, we do not know the wavefield along the entire x-axis. This wavefield is sampled by an array of finite length at M discrete points x_m in space, creating the array data vector:

$$\vec{a} = [a_0, \cdots, a_{M-1}]^T \in \mathbb{C}^M. \tag{3}$$

For simplicity, we will assume that the array is a ULA with higher-than-Nyquist sampling. This means that the elements are placed at:

$$x_m = d_x m \text{ for } d_x = \kappa\frac{\lambda}{2}, \kappa \in (0, 1] \tag{4}$$

The signal observed by an array element is:

$$a_m = s_m h_m + n_m = s(x_m)h_m + n_m = h_m \sum_{p=0}^{P-1} A_p e^{ik(\theta_p)d_x m} + n_m, \tag{5}$$

where h_m defines the finite extent of the array:

$$h_m = \begin{cases} 1 & \text{for} \quad 0 \le m < M \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

The sequence n_m represents spatially white noise (element self-noise), which is defined through:

$$E\{n_m\} = 0, E\{n_m n_{m+l}^*\} = \sigma_n^2 \delta_l, \tag{7}$$

where δ_l is the Kronecker delta function for index l. We will define the normalized wavenumber (for notational simplicity) as:

$$\tilde{k}(\theta) = k(\theta)d_x \in [-\kappa\pi, \kappa\pi]. \tag{8}$$

Note that we will sometimes suppress the θ parameter. The DTFT across the array is given as:

$$A(\tilde{k}) = \sum_{m=0}^{M-1} a_m e^{-i\tilde{k}m} = S(\tilde{k}) * H(\tilde{k}), \tag{9}$$

where $H(\tilde{k})$ is the DTFT of h_m. Note that $S(\tilde{k})$ is band-limited according to (8) while $H(\tilde{k})$.

Our goal is to extract information about the wavefield $S(k(\theta))$ using only the available information in the array data vector. The information of interest is usually one of the following two possibilities:

1) one or more of the amplitudes A_p when the angles θ_p are known (in which case the processing is usually referred to as *beamforming*).
2) the entire set of angles $\{\theta_p\}_{p=0}^{P-1}$ (in which case the processing is usually referred to as *DOA estimation* or *tracking*).

Delay-and-sum beamforming is an attempt at estimating $S(k(\theta))$ from a_m:

$$\hat{S}(\tilde{k}(\theta)) = \sum_{m=0}^{M-1} \tilde{w}_m^* a_m e^{-i\tilde{k}(\theta)m} = \vec{w}^H(\tilde{k}(\theta))\vec{a}, \tag{10}$$

where the weight-and-steering vector $\vec{w}$ is given as

$$\vec{w}(\tilde{k}(\theta)) = \left[\tilde{w}_0 e^{i\tilde{k}(\theta)0}, \tilde{w}_1 e^{i\tilde{k}(\theta)1}, \cdots, \tilde{w}_{M-1} e^{i\tilde{k}(\theta)(M-1)}\right]^T. \tag{11}$$

We observe that the output of the delay-and-sum beamformer is the (possibly weighted) DTFT of the array data vector.

C. Estimating the Covariance Matrix

In most adaptive beamforming methods, the goal is to estimate the covariance matrix of the array:

$$\mathbf{R} = E\{\vec{a}\vec{a}^H\} \tag{12}$$

C.-I.C.Nilsen is with the Department of Informatics, University of Oslo. Ines Hafizovic is with Squarehead Technology AS.

Manuscript received XX XX, 2012.

This is usually done with the Sample Covariance Matrix (SCM):

$$\hat{\mathbf{R}}_{SCM} = \frac{1}{N}\sum_{n=0}^{N-1}\vec{x}[n]\vec{x}^H[n] \tag{13}$$

However, a low number of samples, N, or effects such as correlation between signals and interferers, can result in a poor estimate of $\mathbf{R}$. Another approach can be formulated if we view the covariance matrix as an estimate of the spatial correlation of the wavefield instead of the correlation between physical array elements:

$$\begin{aligned}[\mathbf{R}]_{m,n} &= E\left\{a_m a_n^*\right\} = E\left\{s(x_m)s^*(x_n)\right\} \\ &= E\left\{s(x_m)s^*(x_m+\delta_x[m,n])\right\},\end{aligned} \tag{14}$$

where $[\mathbf{R}]_{m,n}$ means the element in row m and column n of the matrix $\mathbf{R}$, and $\delta_x[m,n]$ is the spatial lag (i.e. distance) between array elements m and n:

$$\delta_x[m,n] = x_m - x_n. \tag{15}$$

For ULAs, the lag function is very simple:

$$\delta_x[m,n] = d_x l \text{ for } l = m-n. \tag{16}$$

A wavefield of monochromatic plane waves is *stationary*, meaning that the spatial correlation depends on lag only:

$$E\left\{s(x_m)s^*(x_m+\delta_x[m,n])\right\} = E\left\{s(x)s^*(x+\delta_x[m,n])\right\} \forall x \tag{17}$$

Therefore, we can define the spatial correlation function of a ULA with the single parameter $l = m-n$:

$$[R]_{m,n} = r_s[l] = E\left\{s(0)s^*(d_x l)\right\} \tag{18}$$

Since $s(x)$ can be written using the inverse DTFT of $S(\tilde{k})$, we can rewrite the spatial correlation function as:

$$r_s[l] = \frac{1}{4\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi} E\left\{S(\tilde{k})S^*(\tilde{k}')\right\} e^{i(\tilde{k}-\tilde{k}')l}\, d\tilde{k}'\, d\tilde{k}. \tag{19}$$

Because $S(\tilde{k})$ is bandlimited as specified in (8), we can rewrite the integration limits:

$$r_s[l] = \frac{1}{4\pi^2}\int_{-\kappa\pi}^{\kappa\pi}\int_{-\kappa\pi}^{\kappa\pi} E\left\{S(\tilde{k})S^*(\tilde{k}')\right\} e^{i(\tilde{k}-\tilde{k}')l}\, d\tilde{k}'\, d\tilde{k}. \tag{20}$$

The expectation in the above expression depends on the actual statistics of the plane wave amplitudes A_p. The basis for most algorithms such as MVDR, MUSIC, ESPRIT, etc. is the assumption that they are independent:

$$E\left\{A_p A_{p'}^*\right\} = |A_p|^2\,\delta_{p-p'}. \tag{21}$$

In reality, this may not be the case. However, when they are independent the inverse DTFT in (20) reduces to:

$$r_s[l] = \frac{1}{2\pi}\int_{-\kappa\pi}^{\kappa\pi}\left|S(\tilde{k})\right|^2 e^{i\tilde{k}l}\, d\tilde{k}. \tag{22}$$

We will use this expression for our covariance matrix estimator. Since we do not know $S(\tilde{k})$, we will first need to estimate it. An obvious estimator is DAS, as given in (10) with $\tilde{w} = M^{-1}$. Inserting (10) into (22) and carrying out the integration yields:

$$r_s[l] = M^{-2}\sum_{m=0}^{M-1}\sum_{n=0}^{M-1} a_m a_n \frac{\sin(\kappa\pi(m-n))}{m-n} \tag{23}$$

For the special case of $\kappa = 1$, this reduces to:

$$r_s[l] = M^{-1}\sum_{m=0}^{M-l-1} a_m a_{m+l}, \tag{24}$$

which is the sum along the l^{th} diagonal of $\hat{\mathbf{R}}_{SCM}$ divided by the number of elements in the array.

D. Additional points

1) Spatially white noise is not covered by the current method. Introduce via diagonal loading?
2) Importance of including κ in integration limits, cf. (20).