Optimal Beamforming

1 Introduction

In the previous section we looked at how *fixed* beamforming yields significant gains in communication system performance. In that case the beamforming was fixed in the sense that the weights that multiplied the signals at each element were fixed (they did not depend on the received data). We now allow these weights to change or *adapt, depending on the received data* to achieve a certain goal. In particular, we will try to adapt these weights to *suppress interference*. The interference arises due to the fact that the antenna might be serving *multiple* users.

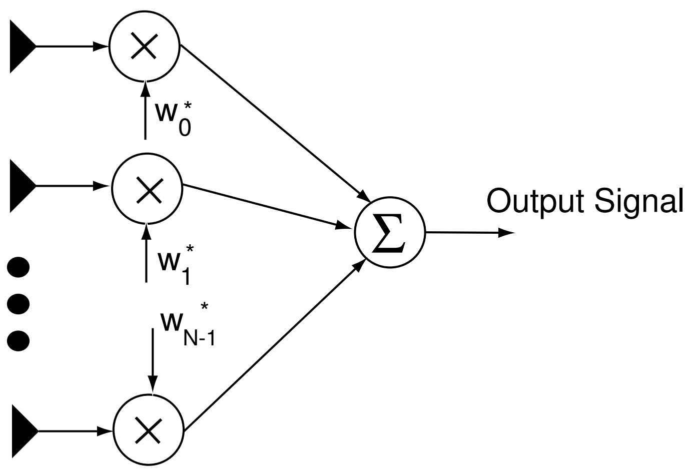

Figure 1: A general beamforming system.

2 Array Weights and the Weighted Response

Figure 1 illustrates the receive beamforming concept. The signal from each element (x_n) is multiplied with a *weight* w_n^*, where the superscript * represents the complex conjugate. Note that the conjugate of the weight multiplies the signal, not the weight itself. The weighted signals are added together to form the output signal. The output signal r is therefore given by

$$\begin{aligned} r &= \sum_{n=0}^{N-1} w_n^* x_n, \\ &= \mathbf{w}^H \mathbf{x}, \end{aligned} \tag{1}$$

where $\mathbf{w}$ represents the length N vector of weights, $\mathbf{x}$ represents the length N vector of received signals and the superscript ^{H} represents the Hermitian of a vector (the conjugate transpose), i.e., $\mathbf{w}^H = \left[w_0^*, \;\; w_1^*, \;\; \ldots \;\; w_{N-2}^*, \;\; w_{N-1}^*\right] = \left[\mathbf{w}^T\right]^*$.

Using these weights, we can define a *weighted response*. If the received data is from a single signal arriving from angle ϕ, the received signal is $\mathbf{x} = \mathbf{s}(\phi)$. The output signal, therefore is,

$$r(\phi) = \mathbf{w}^H \mathbf{s}(\phi). \tag{2}$$

Plotting this function $r(\phi)$ versus ϕ results in the weighted response or the *weighted beampattern*.

The array, of N elements, receives message signals from $M+1$ users. In addition, the signal at each element is corrupted by thermal noise, modelled as additive white Gaussian noise (AWGN). The received signals are multiplied by the *conjugates* of the weights and then summed. The system set up is illustrated in Fig. 1. The weights are allowed to change, depending on the received data, to achieve a purpose. If the signals at the N elements are written as a length-N vector $\mathbf{x}$ and the weights as a length-N vector $\mathbf{w}$, the output signal y is given by

$$y = \sum_{n=0}^{N-1} w_n^* x_n = \mathbf{w}^H \mathbf{x}. \tag{3}$$

The signal received is a sum over the signals from multiple users, *one of which we will designate the "desired" signal*. The received data is a sum of signal, interference and AWGN.

$$\begin{aligned} \mathbf{x} &= \alpha \mathbf{h}_0 + \mathbf{n}, & (4) \\ \mathbf{n} &= \sum_{m=1}^{M} \alpha_m \mathbf{h}_m + \text{noise}. & (5) \end{aligned}$$

The goal of beamforming or interference cancellation is to isolate the signal of the desired user, contained in the term α, from the interference and noise. The vectors $\mathbf{h}_m$ are the *spatial signatures* of the m^{th} user. Note that, unlike in direction of arrival estimation, we are **not** making any assumptions as to the structure of this spatial signature. In perfect line-of-sight conditions, this vector is the steering vector defined earlier. However, in more realistic setting, this vector is a single realization of a random fading process. The model above is valid because are assuming the fading is slow (fading is constant over the symbol period or several symbol periods) and flat (the channel impulse response is a δ-function).

3 Optimal Beamforming

We begin by developing the theory for optimal beamforming. We will investigate three techniques, each for a different definition of optimality. Define the interference and data covariance matrices as $\mathbf{R}_n = \mathrm{E}\left[\mathbf{n}\mathbf{n}^H\right]$ and $\mathrm{R} = \mathrm{E}\left[\mathbf{x}\mathbf{x}^H\right]$, respectively.

3.1 Minimum Mean Squared Error

The minimum mean squared error (MMSE) algorithm minimizes the error *with respect to a reference signal* $d(t)$. In this model, the desired user is assumed to transmit this reference signal, i.e., $\alpha = \beta d(t)$, where β is the signal amplitude and $d(t)$ is known to the receiving base station. The output $y(t)$ is required to track this reference signal. The MMSE finds the weights $\mathbf{w}$ that minimize the average power in the error signal, the difference between the reference signal and the output signal obtained using Eqn. (3)

$$\mathbf{w}_{\text{MMSE}} = \arg\min_{\mathbf{w}} \mathrm{E}\left\{|e(t)|^2\right\}, \tag{6}$$

where

$$\begin{aligned} \mathrm{E}\left\{|e(t)|^2\right\} &= \mathrm{E}\left\{\left|\mathbf{w}^H\mathbf{x}(t) - d(t)\right|^2\right\}, \\ &= \mathrm{E}\left\{\mathbf{w}^H\mathbf{x}\mathbf{x}^H\mathbf{w} - \mathbf{w}^H\mathbf{x}d^* - \mathbf{x}^H\mathbf{w}d + dd^*\right\}, \\ &= \mathbf{w}^H\mathbf{R}\mathbf{w} - \mathbf{w}^H\mathbf{r}_{xd} - \mathbf{r}_{xd}^H\mathbf{w} + dd^*, \end{aligned} \tag{7}$$

where

$$\mathbf{r}_{xd} = \mathrm{E}\left\{\mathbf{x}d^*\right\}. \tag{8}$$

To find the minimum of this functional, we take its derivative with respect to $\mathbf{w}^H$ (we have seen before that we can treat $\mathbf{w}$ and $\mathbf{w}^H$ as independent variables).

$$\begin{aligned} \frac{\partial \mathrm{E}\left\{|e(t)|^2\right\}}{\partial \mathbf{w}^H} &= \mathbf{R}\mathbf{w} - \mathbf{r}_{xd} = 0, \\ \Rightarrow \mathbf{w}_{\text{MMSE}} &= \mathbf{R}^{-1}\mathbf{r}_{xd}. \end{aligned} \tag{9}$$

This solution is also commonly known as the *Wiener filter*.

We emphasize that the MMSE technique minimizes the error with respect to a reference signal. This technique, therefore, does not require knowledge of the spatial signature $\mathbf{h}_0$, but does require knowledge of the transmitted signal. This is an example of a *training based scheme*: the reference signal acts to train the beamformer weights.

3.2 Minimum Output Energy

The minimum output energy (MOE) beamformer defines a different optimality criterion: we minimize the total output energy while *simultaneously* keeping the gain of the array on the desired signal fixed. Because the gain on the signal is fixed, any reduction in the output energy is obtained by suppressing interference. Mathematically, this can be written as

$$\begin{aligned}
\mathbf{w}_{\mathrm{MOE}} &= \arg\min_{\mathbf{w}} \mathrm{E}\left\{|y|^2\right\}, \qquad \mathbf{w}^H\mathbf{h}_0 = c, \\
&\equiv \arg\min_{\mathbf{w}} \mathrm{E}\left\{\left|\mathbf{w}^H\mathbf{x}\right|^2\right\}, \qquad \mathbf{w}^H\mathbf{h}_0 = c. \qquad (10)
\end{aligned}$$

This final minimization can be solved using the method of Lagrange multipliers, finding $\min_{\mathbf{w}}\left[\mathcal{L}(\mathbf{w};\lambda)\right]$, where

$$\begin{aligned}
\mathcal{L}(\mathbf{w};\lambda) &= \mathrm{E}\left\{\left|\mathbf{w}^H\mathbf{x}\right|^2\right\} + \lambda\left[\mathbf{w}^H\mathbf{h}_0 - c\right], \\
&= \mathrm{E}\left\{\mathbf{w}^H\mathbf{x}\mathbf{x}^H\mathbf{w}\right\} + \lambda\left[\mathbf{w}^H\mathbf{h}_0 - c\right], \\
&= \mathbf{w}^H\mathbf{R}\mathbf{w} + \lambda\left(\mathbf{w}^H\mathbf{h}_0 - c\right), \qquad (11) \\
\Rightarrow \frac{\partial\mathcal{L}}{\partial\mathbf{w}^H} &= \mathbf{R}\mathbf{w} + \lambda\mathbf{h}_0 \\
\Rightarrow \mathbf{w}_{\mathrm{MOE}} &= -\lambda\mathbf{R}^{-1}\mathbf{h}_0 \qquad (12)
\end{aligned}$$

Using the constraint on the weight vector, the Lagrange parameter λ can be easily obtained by solving the gain constraint, setting the final weights to be

$$\mathbf{w}_{\mathrm{MOE}} = c\frac{\mathbf{R}^{-1}\mathbf{h}_0}{\mathbf{h}_0^H\mathbf{R}^{-1}\mathbf{h}_0}. \qquad (13)$$

Setting the arbitrary constant $c = 1$ gives us the minimum variance distortionless response (MVDR), so called because the output signal y has minimum variance (energy) and the desired signal is not distorted (the gain on the signal direction is unity).

Note the MOE technique does not require a reference signal. This is an example of a *blind* scheme. The scheme, on the other hand, does require knowledge of the spatial signature $\mathbf{h}_0$. The technique "trains" on this knowledge.

3.3 Maximum Output Signal to Interference Plus Noise Ratio - Max SINR

Given a weight vector $\mathbf{w}$, the output signal $y = \mathbf{w}^H\mathbf{x} = \alpha\mathbf{w}^H\mathbf{h}_0 + \mathbf{w}^H\mathbf{n}$, where $\mathbf{n}$ contains both interference and noise terms. Therefore, the output signal to interference plus noise ratio (SINR) is given by

$$\mathrm{SINR} = \mathrm{E}\left\{|\alpha|^2\right\}\frac{\left|\mathbf{w}^H\mathbf{h}_0\right|^2}{\mathrm{E}\left\{\left|\mathbf{w}^H\mathbf{n}\right|^2\right\}} = A^2\frac{\left|\mathbf{w}^H\mathbf{h}_0\right|^2}{\mathrm{E}\left\{\left|\mathbf{w}^H\mathbf{n}\right|^2\right\}}, \qquad (14)$$

where $A^2 = \mathrm{E}\left\{|\alpha|^2\right\}$ is the average signal power. Another (reasonable) optimality criterion is to maximize this output SINR with respect to the output weights.

$$\mathbf{w}_{\text{Max SINR}} = \arg\max_{\mathbf{w}} \{\text{SINR}\}. \tag{15}$$

We begin by recognizing that multiplying the weights by a constant does not change the output SINR. Therefore, since the spatial signature $\mathbf{h}_0$ is fixed, we can choose a set of weights such that $\mathbf{w}^H\mathbf{h}_0 = c$. The maximization of SINR is then equivalent to minimizing the interference power, i.e.

$$\mathbf{w}_{\text{Max SINR}} = \min_{\mathbf{w}} \mathrm{E}\left\{\left|\mathbf{w}^H\mathbf{n}\right|^2\right\} = \mathbf{w}^H\mathbf{R}_n\mathbf{w}, \qquad \mathbf{w}^H\mathbf{h}_0 = c. \tag{16}$$

We again have a case of applying Lagrange multipliers, as in Section 3.2, with $\mathbf{R}$ replaced with $\mathbf{R}_n$. Therefore,

$$\mathbf{w} = c\frac{\mathbf{R}_n^{-1}\mathbf{h}_0}{\mathbf{h}_0^H\mathbf{R}_n^{-1}\mathbf{h}_0}. \tag{17}$$

3.4 Equivalence of the Optimal Weights

We have derived three different weight vectors using three different optimality criteria. Is there a "best" amongst these three? No! We now show that, in theory, all three schemes are equivalent. We begin by demonstrating the equivalence of the MOE and Max SINR criteria. Using the definitions of the correlation matrices $\mathbf{R}$ and $\mathbf{R}_n$,

$$\mathbf{R} = \mathbf{R}_n + A^2\mathbf{h}_0\mathbf{h}_0^H \tag{18}$$

Using the Matrix Inversion Lemma[1]

$$\begin{aligned}
\mathbf{R}^{-1} &= \left[\mathbf{R}_n + A^2\mathbf{h}_0\,\mathbf{h}_0^H\right]^{-1}, \\
\Rightarrow \mathbf{R}^{-1} &= \mathbf{R}_n^{-1} - \frac{\mathbf{R}_n^{-1}\mathbf{h}_0\,\mathbf{h}_0^H\mathbf{R}_n^{-1}}{\mathbf{h}_0^H\mathbf{R}_n^{-1}\mathbf{h}_0 + A^{-2}}, \qquad (19)\\
\Rightarrow \mathbf{R}^{-1}\mathbf{h}_0 &= \mathbf{R}_n^{-1}\mathbf{h}_0 - \frac{\mathbf{R}_n^{-1}\mathbf{h}_0\,\mathbf{h}_0^H\mathbf{R}_n^{-1}\mathbf{h}_0}{\mathbf{h}_0^H\mathbf{R}_n^{-1}\mathbf{h}_0 + A^{-2}}, \\
&= \mathbf{R}_n^{-1}\mathbf{h}_0 - \frac{\left(\mathbf{h}_0^H\mathbf{R}_n^{-1}\mathbf{h}_0\right)\mathbf{R}_n^{-1}\mathbf{h}_0}{\mathbf{h}_0^H\mathbf{R}_n^{-1}\mathbf{h}_0 + A^{-2}}, \\
&= \left(\frac{A^{-2}}{\mathbf{h}_0^H\mathbf{R}_n^{-1}\mathbf{h}_0 + A^{-2}}\right)\mathbf{R}_n^{-1}\mathbf{h}_0, \\
&= c\mathbf{R}_n^{-1}\mathbf{h}_0, \qquad (20)
\end{aligned}$$

[1] $[\mathbf{A} + \mathbf{BCD}]^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}\left[\mathbf{DA}^{-1}\mathbf{B} + \mathbf{C}^{-1}\right]^{-1}\mathbf{DA}^{-1}$

i.e., the adaptive weights obtained using the MOE and Max SINR criteria are proportional to each other. Since multiplicative constants in the adaptive weights do not matter, these two techniques are therefore equivalent.

To show that the MMSE weights (Wiener filter) and the MOE weights are equivalent, we start with the definition of $\mathbf{r}_{xd}$,

$$\begin{aligned}
\mathbf{r}_{xd} &= \mathrm{E}\left\{\mathbf{x}d^*(t)\right\}, \\
&= \mathrm{E}\left\{\left[\alpha\mathbf{h}_0 + \mathbf{n}\right]d^*(t)\right\}.
\end{aligned} \tag{21}$$

Note that the term $\alpha = \beta d(t)$, where β is some amplitude term and $d(t)$ is the reference signal. Therefore,

$$\begin{aligned}
\mathbf{r}_{xd} &= \beta|d|^2\mathbf{h}_0 + \mathrm{E}\left\{\mathbf{n}\right\}d^*(t) \\
&= \beta|d|^2\mathbf{h}_0 \\
&\propto \mathbf{h}_0 \\
\Rightarrow \mathbf{w}_{\mathrm{MMSE}} &\propto \mathbf{w}_{\mathrm{MOE}}
\end{aligned} \tag{22}$$

i.e., the MMSE weights and the MOE weights are also equivalent. Therefore, theoretically, all three approaches yield the same weights starting from different criteria for optimality. We will see that these criteria are very different *in practice.*

3.5 Suppression of Interference

We can use the matrix inversion lemma to determine the response of the optimal beamformer to a particular interference source. Since all three beamformers are equivalent, we can choose any one of the three. We choose the MOE. Within a constant, the optimal weights are given by $\mathbf{w} = \mathbf{R}^{-1}\mathbf{h}_0$. Denote as $\mathbf{Q}$ the interference correlation matrix without the particular interference source (assumed to have amplitude α_i, with $\mathrm{E}\left\{|\alpha_i|^2\right\} = A_i$, and spatial signature $\mathbf{h}_i$. In this case,

$$\begin{aligned}
\mathbf{R} &= \mathbf{Q} + \mathrm{E}\left\{|\alpha|^2\right\}\mathbf{h}_i\mathbf{h}_i^H \\
&= \mathbf{Q} + A_i\mathbf{h}_i\mathbf{h}_i^H
\end{aligned}$$

$$\Rightarrow \mathbf{R}^{-1} = \mathbf{Q}^{-1} - \frac{A_i^2\mathbf{Q}^{-1}\mathbf{h}_i\mathbf{h}_i^H\mathbf{Q}^{-1}}{1 + A_i^2\mathbf{h}_i^H\mathbf{Q}^{-1}\mathbf{h}_i}$$

The gain of the beamformer (with the interference) on the interfering source is given by $\mathbf{w}^H\mathbf{h}_i$. Using the definition of the weights,

$$
\begin{aligned}
\mathbf{w}^H\mathbf{h}_i &= \mathbf{h}_0^H\mathbf{R}^{-1}\mathbf{h}_i \\
&= \mathbf{h}_0\mathbf{Q}^{-1}\mathbf{h}_i - \frac{A_i^2\left[\mathbf{h}_0^H\mathbf{Q}^{-1}\mathbf{h}_i\right]\left[\mathbf{h}_i^H\mathbf{Q}^{-1}\mathbf{h}_i\right]}{1+A_i^2\mathbf{h}_i^H\mathbf{Q}^{-1}\mathbf{h}_i} \\
&= \frac{\mathbf{h}_0^H\mathbf{Q}^{-1}\mathbf{h}_i}{1+A_i^2\mathbf{h}_i^H\mathbf{Q}^{-1}\mathbf{h}_i}
\end{aligned} \tag{23}
$$

The numerator in the final equation is the response of the optimal beamformer on the interference *if the interferer were not present.* The denominator represents the amount by which this response is reduced due to the presence of the interferer. Note that the amount by which the gain is reduced is dependent on the power of the interference (corresponding to A_i^2). The stronger the interference source, the greater is it suppressed. An optimal beamformer may “ignore” a weak interfering source.

Figure 2 illustrates the performance of the optimal beamformer. This example uses line of sight conditions, which allows us to use “beampatterns”. However, it must be emphasized that this is for convenience only. In line of sight conditions, the direction of arrival, ϕ, determines the spatial signature to be the steering vector, $\mathbf{s}(\phi)$, as developed earlier in this course. In the first case, there are two interferers at angles $\phi = 30^o$, 150^o and in the other, there two interferers at $\phi = 40^o$, 150^o. The desired signal is at $\phi = 0$. Note how the null moves over from 30^o to 40^o. It is important to realize that the optimal beamformer creates these nulls (suppresses the interfering sources) *without a-priori knowledge as to the location of the interference sources.*

If there are more interfering sources than elements, all sources cannot be nulled (a N element array can only null $N-1$ interfering signals). In this case, the beamformer sets the weights to minimize the *overall* output interference power. Figure 3 illustrates this situation. The solid line is the case of seven equally strong interfering sources received by a five element antenna array ($N = 5$). Note that the array, apparently, forms only four nulls. These nulls do not correspond directly to the locations of the interference as it is the *overall* interference that is being minimized. The dashed line plots the beampattern in the case that one of the seven interfering sources, arriving from $\phi = 40^o$, is extremely strong. Note that the beamformer automatically moves a null in direction of this strong interferer (to the detriment of the suppression of the other interfering sources - again, it is the *overall* interference power that is minimized).

3.6 Relation of Beampattern and Noise

In this section we focus on the impact the weights have on the noise component. Again, the term ”beampattern” has a connotation that beamforming is for line of sight conditions only. In point of fact, a beampattern can always be defined. However, in multipath fading channels, the desired

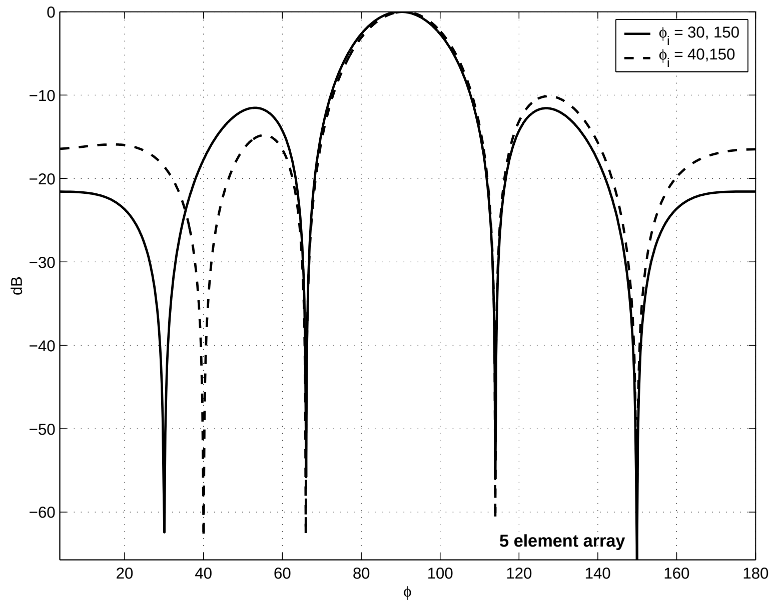

Figure 2: Illustration of nulling using the optimal beamformer.

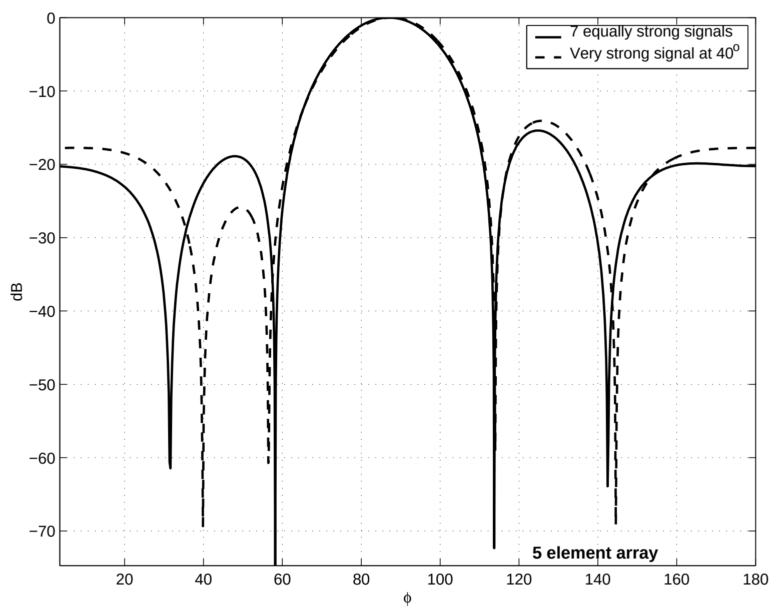

Figure 3: Nulling with number of interferers greater than number of elements.

signal does not arrive from a single direction. We assume that the noise at each element is white, independent, zero mean and Gaussian, with noise power σ^2, i.e., $\mathrm{E}\left[\mathbf{n}\mathbf{n}^H\right] = \sigma^2\mathbf{I}$. Here, in an abuse of notation we use $\mathbf{n}$ to represent noise (which was noise and interference in the earlier sections).

The noise component in the output signal is $\mathbf{w}^H\mathbf{n}$, i.e. the noise power in the output (σ_o^2) is given by

$$\sigma_o^2 = \mathrm{E}\left[\left|\mathbf{w}^H\mathbf{n}\right|^2\right] = \mathrm{E}\left[\left|\sum_{i=0}^{N-1} w_i^* n_i\right|^2\right], \tag{24}$$

$$= \sigma^2 \sum_{i=0}^{N-1} |w_i|^2 . \tag{25}$$

The noise is, therefore, enhanced by the "total energy" in the weights. Now, consider the adapted beam pattern. In direction ϕ the gain of the array due to the weights is given by $\mathbf{w}^H\mathbf{s}(\phi)$,where $\mathbf{s}(\phi)$ is the steering vector associated with direction ϕ. In fact, due to the structure of the steering vector, setting $\psi = kd\cos\phi$,

$$G(\psi) = \sum_{n=0}^{N-1} w_n e^{jn\psi}, \tag{26}$$

i.e., in ψ space, the beam pattern is the DFT of the weights. Therefore, due to Parseval's theorem,

$$\sum_{i=0}^{N-1} |w_i|^2 = \int_0^{2\pi} |G(\psi)|^2 \, d\psi, \tag{27}$$

i.e., the enhancement of the noise power is the same as total area under the beampattern. Clearly, if the adapted pattern had high sidelobes, the noise would be enhanced. Therefore, in trying to suppress interference, the beamforming algorithm (MOE/MMSE/Max SINR) also attempts to keep sidelobes low, thereby maximizing the overall SINR.

Figure 4 plots the adapted beampattern in the case of signals with noise and signals without noise. Clearly, is noise were absent the minimum interference power is achieved by a set of weights that result in high sidelobes. If these weights were used in a real system, they would minimize interference power, but enhance the noise power. The solid line plots the adapted beam pattern with exactly the same signal and interference data, but now noise is also added in. Note both plots pas through 0dB at the direction of the desired signal (0^o). Clearly to minimize *SINR* (not SIR), the sidelobes must come down.

This issue is of relevance when trying to use algorithms such as *zero-forcing* in the spatial case. Zero-forcing is a technique that focuses on the interference only and is best known when implementing *transmit* beamforming (as opposed to the receive beamforming we have been doing) [1]. It is well known that zero-forcing enhances noise. This is because in attempting to cancel interference only, it lets the sidelobes rise, resulting in higher noise power.

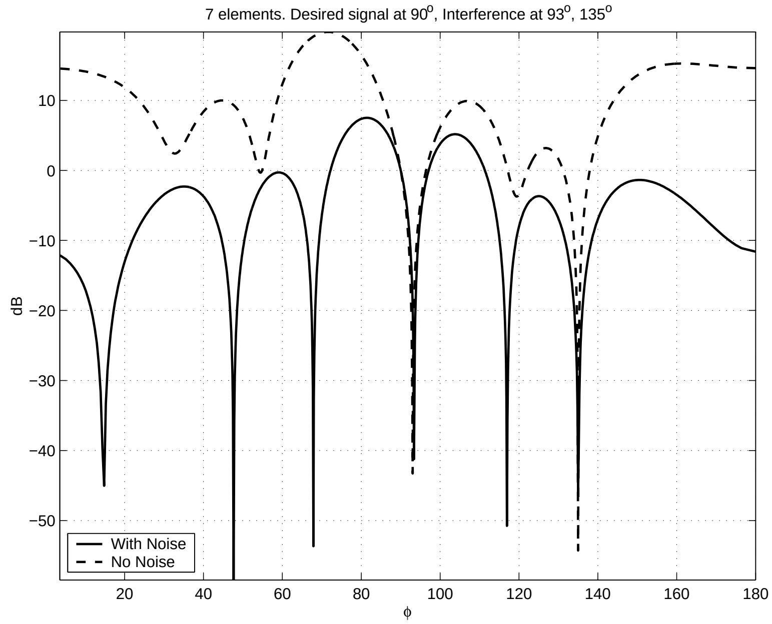

Figure 4: Adapted pattern with and without noise.

4 Beamforming in Practice

In practice it is impossible to implement the optimal beamformer, because the correlation matrix $\mathbf{R}$ is unknown. In this section, we will investigate five techniques to implement beamforming in practice. Four of these are iterative, though we start with the only non-iterative technique: direct matrix inversion.

4.1 Direct Matrix Inversion

In direct matrix inversion (DMI), the correlation matrix $\mathbf{R}$ is estimated by averaging over multiple snapshots (time samples of the received data). If the interference does not change significantly over the time length used in the averaging process, one can show that (in Gaussian interference), the following average is a maximum likelihood estimate of the correlation matrix.

$$\tilde{\mathbf{R}} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{x}_k\mathbf{x}_k^H \tag{28}$$

where $\mathbf{x}_k$ is the data received at the N elements at time index k. In [2], the authors prove that we need $K > 2N$ so that the output SINR is within 3dB of the optimum.

DMI can be applied using either the MMSE or Max SNR formulations. In using the MMSE

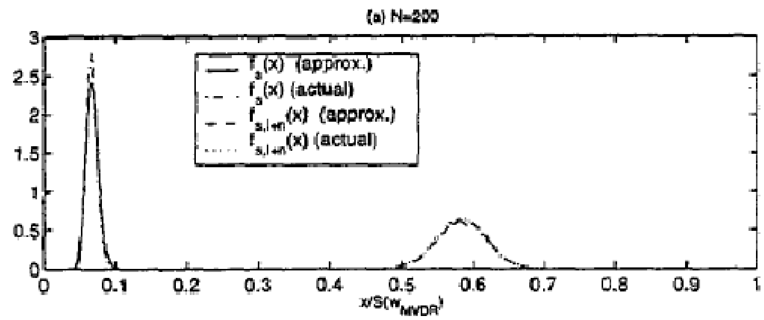

Figure 5: PDF of the output SINR when the signal is and is not present in the training data when using the blind MOE technique

criterion, the cross correlation vector $\mathbf{r}_{xd}$ is estimated by

$$\tilde{\mathbf{r}}_{xd} = \frac{1}{K}\sum_{k=1}^{K} d_k^* \mathbf{x}_k, \tag{29}$$

$$\tilde{\mathbf{w}}_{\text{MMSE}} = \tilde{\mathbf{R}}^{-1}\tilde{\mathbf{r}}_{xd} \tag{30}$$

i.e., using DMI in this fashion requires a sequence of K *known* transmitted symbols, also known as *a training sequence*. This is our first example of a training based beamforming procedure. On the other hand, a *blind* procedure does not require a training sequence. DMI may also be applied in this fashion using the MOE criterion.

$$\tilde{\mathbf{w}}_{\text{MOE}} = \tilde{\mathbf{R}}^{-1}\mathbf{h}_0. \tag{31}$$

The problem with this procedure is that, as shown in [3], the data used to estimate the correlation matrix $\mathbf{x}_k$ must be *signal-free*, i.e., it is only possible to use the Max SINR criterion. Figure 5 plots the output pdf normalized output SINR (normalized with respect to the theoretical optimal). The two plots are for $K = 200$. Clearly, in both cases, when the signal is present in the training data, the pdf of the output SINR is significantly to the left (the mean is much lower) than when the signal is absent. The physical interpretation appears to be that the MOE process assumes all the signals in the training data is interference. So, the MOE approach tries to place *a null and a maximum* simultaneously in the direction of the desired signal.

This problem is well established in radar systems, where a guard range is placed to eliminate any of the desired signal. In a communication system, on the other hand, this would imply that the desired user must stay silent in the training phase. Of course, if the desired signal is silent, this is equivalent to the Max SINR criterion. This is in direct contrast to the MMSE criterion where the user must transmit a known signal. It must be emphasized that this problem only occurs when the correlation matrix is *estimated*. We have already shown that when the true matrix is known, the MMSE and Max SNR techniques are equivalent.

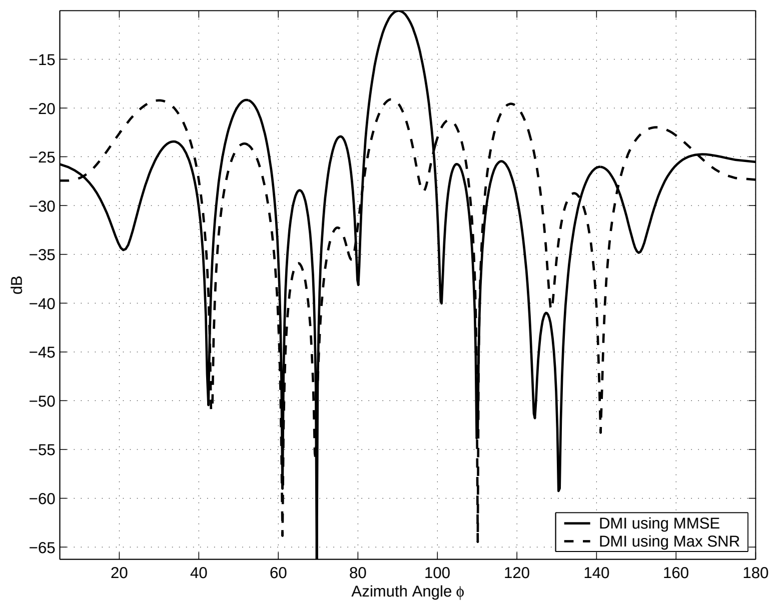

Figure 6: Adapted beampattern using the MMSE and Max SNR techniques.

Figure 6 illustrates the problem of having the desired signal in the training data. The example uses an 11-element array ($N = 11$) with five signals at $\phi = 90^o$, 61^o, 69.5^o, 110^o, 130^o, . The signal at $\phi = 90^o$ is the desired signal while the other signals are interference. Using the DMI technique with the MMSE as in Eqn. (30) the adapted beampattern using the MMSE version of DMI clearly shows nulls in direction of the interference. However, when using the MOE version of DMI (using Eqn. (31)) the beampattern retains the nulls, but the overall pattern is significantly corrupted. We know that high sidelobes implies worse SNR, i.e., the MOE approach has severely degraded performance if the desired signal is present in the training data.

We next look at iterative techniques for beamforming in practice. Before we explore these techniques we need to understand iterative techniques in general. We begin the discussion with the Steepest Descent algorithm which is an iterative technique to find the minimum of a functional when the *functional is known exactly.*

4.2 Steepest Descent

The steepest descent technique is an iterative process to find the minimum of a functional, in our case, a functional of the weight vector $\mathbf{w}$. We want to find the solution to a matrix equation $\mathbf{R}\mathbf{w} = \mathbf{r}_{xd}$, however the functional being minimized is $f(\mathbf{w}) = \mathrm{E}\left[\left|\mathbf{w}^H\mathbf{x} - d\right|^2\right]$. This functional can be thought of being a function of N variables, $[w_0, w_1, \ldots, w_{N-1}]$.

Assume that at the k-th iteration we have an estimate of the weights $\mathbf{w}_k$. We look to find a better estimate by modifying $\mathbf{w}_k$ in the form $\mathbf{w}_k + \Delta\mathbf{w}$. We know any function *increases fastest* in direction of its gradient, and so it decreases fastest in the opposite direction. The steepest descent therefore finds the correction factor $\Delta\mathbf{w}$ in the direction opposite to that of the gradient, i.e. $\Delta\mathbf{w} = -\mu\nabla f$, $\mu > 0$. Here, μ is called the step-size. Therefore, at the $(k+1)$-th step,

$$\begin{aligned}
\mathbf{w}_{k+1} &= \mathbf{w} + \Delta\mathbf{w}, \\
&= \mathbf{w} - \mu\nabla f, && (32)\\
f(\mathbf{w}) &= \mathrm{E}\left[\left|\mathbf{w}^H\mathbf{x} - d\right|^2\right], \\
&= \mathbf{w}^H\mathbf{R}\mathbf{w} - \mathbf{w}^H\mathbf{r}_{xd} - \mathbf{r}_{xd}^H\mathbf{w} + |d|^2 && (33)\\
\Rightarrow \nabla f(\mathbf{w}) &= 2\frac{\partial f}{\partial \mathbf{w}^H} = \mathbf{R}\mathbf{w} - \mathbf{r}_{xd}. && (34)
\end{aligned}$$

This results in the relationship

$$\begin{aligned}
\mathbf{w}_{k+1} &= \mathbf{w}_k - \mu\nabla f \\
&= \mathbf{w}_k + \mu\left(\mathbf{r}_{xd} - \mathbf{R}\mathbf{w}_k\right). && (35)
\end{aligned}$$

Note that, other than a factor of the stepsize μ, the change in the weight vector is determined by the error term, at the k^{th} step, in the matrix equation $\mathbf{R}\mathbf{w} = \mathbf{r}_{xd}$.

4.2.1 The range of μ

An immediate question that arises is how to choose μ. Conceivably, a large value of μ would make the solution converge faster. However, as in any iteration process, an extremely large step size could make the iterations unstable. We, therefore, need to determine the range of μ that guarantees convergence. The iterative process must converge to the solution of the above matrix equation

$$\mathbf{R}\mathbf{w} = \mathbf{r}_{xd}.$$

Let $\mathbf{w}_o$ (o for optimal, do not confuse with $k = 0$, the initial guess) be the solution to this equation. The error at the k-th step is $\mathbf{e}_k = \mathbf{w}_k - \mathbf{w}_o$.

Subtracting the optimal solution from both sides of Eqn. (35), we get

$$\begin{aligned}
\mathbf{w}_{k+1} - \mathbf{w}_o &= \mathbf{w}_k - \mathbf{w}_o + \mu\left(\mathbf{r}_{xd} - \mathbf{R}\mathbf{w}_k\right) \\
\Rightarrow \mathbf{e}_{k+1} &= \mathbf{e}_k + \mu\left(\mathbf{r}_{xd} - \mathbf{R}\mathbf{w}_k\right). && (36)
\end{aligned}$$

However, since $\mathbf{r}_{xd} = \mathbf{R}\mathbf{w}_o$,

$$\begin{aligned}
\mathbf{e}_{k+1} &= \mathbf{e}_k + \mu\left(\mathbf{R}\mathbf{w}_o - \mathbf{R}\mathbf{w}_k\right), \\
\mathbf{e}_{k+1} &= \mathbf{e}_k - \mu\mathbf{R}\mathbf{e}_k, \\
&= \left(\mathbf{I} - \mu\mathbf{R}\right)\mathbf{e}_k && (37)
\end{aligned}$$

Writing the correlation matrix $\mathbf{R}$ in its eigenvalue decomposition, $\mathbf{R} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^H$, where $\mathbf{Q}^H\mathbf{Q} = \mathbf{Q}\mathbf{Q}^H = \mathbf{I}$, we are write the final equation as

$$
\begin{aligned}
\mathbf{e}_{k+1} &= \mathbf{Q}\left[\mathbf{I} - \mu\mathbf{\Lambda}\right]\mathbf{Q}^H\mathbf{e}_k, && (38)\\
\Rightarrow \mathbf{Q}^H\mathbf{e}_{k+1} &= \left[\mathbf{I} - \mu\mathbf{\Lambda}\right]\mathbf{Q}^H\mathbf{e}_k. && (39)
\end{aligned}
$$

Let $\tilde{\mathbf{e}}_k = \mathbf{Q}^H\mathbf{e}_k$. Therefore,

$$
\tilde{\mathbf{e}}_{k+1} = \left[\mathbf{I} - \mu\mathbf{\Lambda}\right]\tilde{\mathbf{e}}_k. \tag{40}
$$

Since $\mathbf{\Lambda}$ is a diagonal matrix of the eigenvalues, this equation represents an *uncoupled* system of equations and each individual term can be written separately. The n-th term can be written as

$$
\begin{aligned}
\tilde{\mathbf{e}}_{k+1,n} &= (1 - \mu\lambda_n)\tilde{\mathbf{e}}_{k,n},\\
&= (1 - \mu\lambda_n)^{k+1}\tilde{\mathbf{e}}_{0,n}. && (41)
\end{aligned}
$$

Therefore, if the solution is to converge, the exponent term must tend to zero, i.e.

$$
\begin{aligned}
&|1 - \mu\lambda_n| < 1,\\
&\Rightarrow -1 < 1 - \mu\lambda_n < 1,\\
&\Rightarrow 0 < \mu < \frac{2}{\lambda_n}, && (42)
\end{aligned}
$$

Since this inequality must be true for all values of n, $0 \leq n \leq N-1$, the inequality that μ must satisfy is

$$
0 \leq \mu \leq \frac{2}{\lambda_{\max}}. \tag{43}
$$

Therefore, the highest possible value of the step size is set by the largest eigenvalue of the correlation matrix.

Finally, there are a few points about the steepest descent approach that must be emphasized:

- This form of eigendecomposition based analysis is common in iterative techniques based on a matrix.
- The steepest descent algorithm is *not* adaptive. So far, all we have seen is an iterative method to solve for the optimal weights.
- Choosing a "good" value of μ is essential for the algorithm to converge without numerical instabilities.

5 Least Mean Squares Algorithm

The Least Mean Squares (LMS) algorithm is an adaptive implementation of the steepest descent algorithm. This is probably the most popular adaptive algorithm as it is simple and easy to implement. However, a significant drawback is poor convergence.

In the LMS algorithm, the correlation matrix ($\mathbf{R}$) and the cross correlation vector ($\mathbf{r}_{xd}$) are replaced by their instantaneous values, i.e., in Eqn.(35), $\mathbf{R}$ is replaced by $\mathbf{R}_n = \mathbf{x}_n\mathbf{x}_n^H$ and $\mathbf{r}_{xd}$ is replaced by $\mathbf{x}_n d_n^*$. Remember that $\mathbf{x}_n$ is the received data at the N elements at the n-th time index. *Note that in implementing the LMS algorithm we have switched from using the index k (which denotes iteration number) to index n which denotes the n-th time sample.* The LMS algorithm, therefore, requires a *training sequence of known symbols*, d_n, to train the adaptive weights. Enough training data symbols must be available to ensure convergence. It is important to realize that this training sequence represents wasted energy and time of transmission (there cannot be any information transmitted in the training sequence). The LMS algorithm can be summarized as

$$\begin{aligned} \mathbf{w}_{n+1} &= \mathbf{w}_n - \mu\left(\mathbf{x}_n\mathbf{x}_n^H\mathbf{w}_n - \mathbf{x}_n d_n^*\right), \\ &= \mathbf{w}_n - \mu\mathbf{x}_n\left(\mathbf{x}_n^H\mathbf{w}_n - d_n^*\right), \\ \Rightarrow \mathbf{w}_{n+1} &= \mathbf{w}_n - \mu\mathbf{x}_n e_n^* \end{aligned} \tag{44}$$

where $e_n = \mathbf{w}_n^H\mathbf{x}_n - d_n$, a scalar, is the error at the n-th training step.

In implementing the LMS algorithm, due to the random nature of the received data, Eqn.(44) is a stochastic difference equation. If the solution converges, it will *only converge in the mean.* In general, to ensure convergence, μ must be significantly smaller than the bound obtained in Eqn. (43).

6 Recursive Least Squares

The recursive least squares (RLS) technique approaches the adaptive problem from a different angle. At a time index time index n the algorithm minimizes not the error at that time index only, but the accumulated errors over all previous time instances, i.e., RLS minimizes the following error

$$E(n) = \sum_{j=0}^{n}\lambda^{n-j}|e(j)|^2 = \sum_{j=0}^{n}\lambda^{n-j}\left|d(j) - \mathbf{w}_n^H\mathbf{x}_j\right|^2. \tag{45}$$

Here, the parameter λ is the *forgetting parameter* which sets the memory of the solution system. Generally, $\lambda < 1$ and so the error term is exponentially weighted toward the recent data samples. The case of $\lambda = 1$ is called a growing memory implementation (as all the samples are retained). This results in what is effectively a sliding window implementation of the MMSE approach. Using our standard practice as done for the MMSE and Max SNR cases, we can show that the results weights, at the n-th time index are the solution to the matrix equation

$$\mathbf{R}_n \mathbf{w}_n = \mathbf{r}_{\mathrm{RLS}}, \tag{46}$$

where,

$$\mathbf{R}_n = \sum_{j=0}^{n} \lambda^{n-j} \mathbf{x}_j \mathbf{x}_j^H, \tag{47}$$

$$\mathbf{r}_{\mathrm{RLS}} = \sum_{j=0}^{n} \lambda^{n-j} \mathbf{x}_j d_j^*. \tag{48}$$

Note that we are basically using a new definition of an averaged correlation matrix and vector that is exponentially weighted toward the "recent" values.

In the present form, the RLS algorithm is extremely computationally intensive as it requires the solution to a matrix equation at each time index. However, we can use the matrix inversion lemma to significantly reduce the computation load. We note that at time index n, the averaged correlation matrix can be written in terms of the same matrix at time index $n-1$.

$$\mathbf{R}_n = \lambda \mathbf{R}_{n-1} + \mathbf{x}_n \mathbf{x}_n^H \tag{49}$$

$$\Rightarrow \mathbf{R}_n^{-1} = \lambda^{-1}\mathbf{R}_{n-1}^{-1} - \frac{\left[\lambda^{-1}\mathbf{R}_{n-1}^{-1}\mathbf{x}_n\right]\left[\lambda^{-1}\mathbf{R}_{n-1}^{-1}\mathbf{x}_n\right]^H}{1+\lambda^{-1}\mathbf{x}_n^H \mathbf{R}_{n-1}^H \mathbf{x}_n}, \tag{50}$$

$$= \lambda^{-1}\mathbf{R}_{n-1}^{-1} - \mathbf{g}_n \mathbf{g}^H \tag{51}$$

where

$$\mathbf{g}_n = \frac{\lambda^{-1}\mathbf{R}_{n-1}^{-1}\mathbf{x}_n}{\sqrt{1+\lambda^{-1}\mathbf{x}_n^H \mathbf{R}_{n-1}^{-1}\mathbf{x}_n}}. \tag{52}$$

A new matrix inverse is therefore not needed at each step and inverting the new correlation matrix is not the usual $\mathcal{O}(N^3)$ problem, but just a $\mathcal{O}(N^2)$ problem at each iteration step.

A comparison of the performance of the LMS and RLS algorithms is shown in Fig. 9 where the residual error term is plotted as a function of the number of training symbols. As can be seen, the RLS algorithm converges faster than the LMS algorithm. The significant drawback with RLS, in comparison to LMS, is the larger computation load. Even with the matrix inversion lemma, the RLS method requires computation of order of N^2, while the computation load of LMS remains of order N.

Note that all the RLS algorithm does is solve the original matrix equation $\mathbf{R}\mathbf{w} = \mathbf{r}_{xd}$ in a *distributed* manner. While the overall computation load may actually be higher than in solving this matrix equation directly, *for each symbol* the computation load is of order N^2. This is in contrast to the DMI technique that requires an order-N^3 matrix solution after "doing nothing" for K time instants.

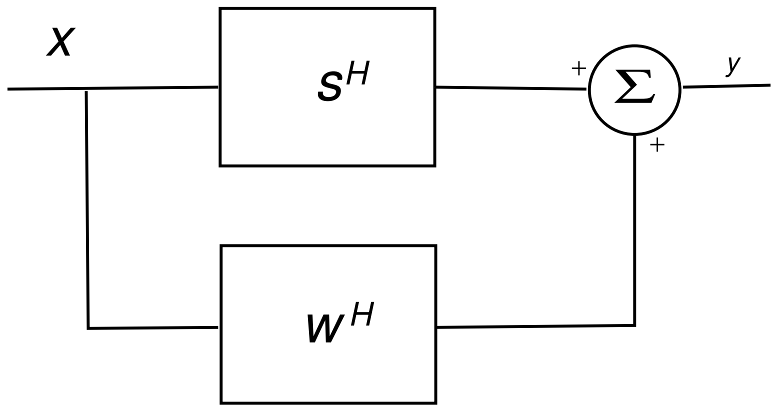

Figure 7: Block diagram of the minimum output energy formulation

7 Minimum Output Energy

The LMS and RLS algorithms described above are examples of training based techniques that require a known sequence of transmitted symbols. These training symbols represent wasted time and energy. We investigate here a *blind* technique based on the minimum output energy (MOE) formulation. As with the idea case, the idea is to maintain gain on the signal while minimizing total output energy. Blind techniques do not require a training sequence and hence accrue significant savings in terms of energy. However, as can be expected, they have slow convergence to the optimal weights.

The MOE structure is shown in Fig. 7. The top branch is the matched filter set such that $\mathbf{h}_0^H\mathbf{h}_0 = 1$. The bottom branch represents the adaptive weights that are chosen such that $\mathbf{w} \perp \mathbf{h}_0$ always. The algorithm then chooses that set of weights $\mathbf{w}$ that minimizes the output energy, $|y|^2$. Note that the overall weights are given by $(\mathbf{h}_0 + \mathbf{w})$ since

$$y = (\mathbf{h}_0 + \mathbf{w})^H\mathbf{x}. \tag{53}$$

Further note that since $\mathbf{w} \perp \mathbf{h}_0$, the gain on the desired signal is fixed. This can be easily proved: $(\mathbf{h}_0 + \mathbf{w})^H\,\mathbf{h}_0 = \mathbf{h}_0^H\mathbf{h}_0 = 1$. Therefore, any reduction in the energy in the output signal y, must come from reductions in the output interference power. In this sense, this MOE formulation is the same as the optimal MOE criterion used in Section 3.2.

In the MOE algorithm, the functional we are minimizing is

$$\begin{aligned} f(\mathbf{w}) &= \mathrm{E}\left\{\left|(\mathbf{h}_0 + \mathbf{w})^H\,\mathbf{x}\right|^2\right\}, & (54)\\ \Rightarrow \nabla f(\mathbf{w}) &= 2\left[\mathbf{x}^H\,(\mathbf{h}_0 + \mathbf{w})\right]\mathbf{x}. & (55)\end{aligned}$$

However, since $\mathbf{w} \perp \mathbf{h}_0$, in the update equation for the weights, we only take the component of ∇f

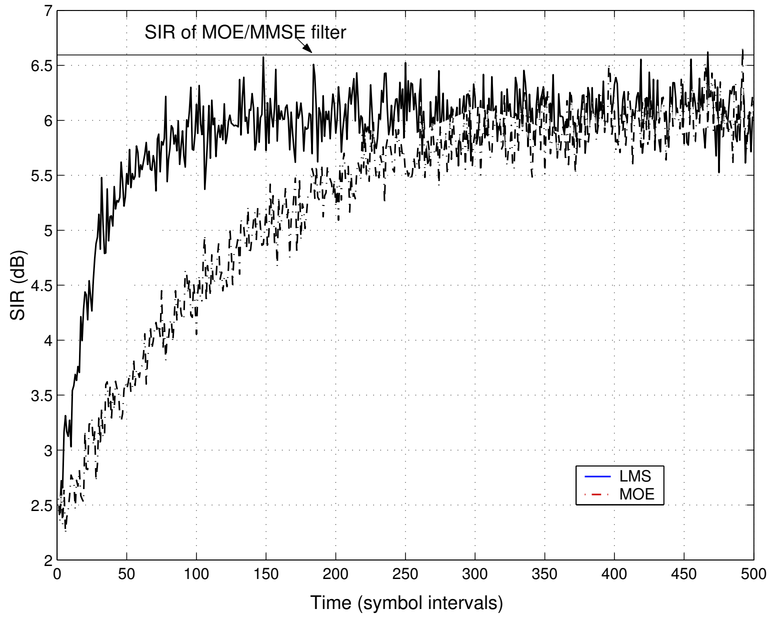

Figure 8: Comparison of the convergence of the LMS and MOE algorithms

that is orthogonal to the channel vector $\mathbf{h}_0$. Using Eqn. (55),

$$\begin{aligned} \nabla f_{\perp} &= 2\left[\mathbf{x}^H\left(\mathbf{h}_0 + \mathbf{w}\right)\right]\left[\mathbf{x} - \left(\mathbf{h}_0^H \mathbf{x}\right)\mathbf{h}_0\right], \\ &= 2y^*\left[\mathbf{x} - y_{MF}\mathbf{h}_0\right], \qquad (56) \\ \Rightarrow \mathbf{w}_{n+1} &= \mathbf{w}_n - \mu y_n^*\left[\mathbf{x}_n - y_{MF_n}\mathbf{h}_0\right], \qquad (57) \end{aligned}$$

where y_{MF_n} is the output from the top branch only, the matched filter at the n-th time instant and y_n is the output of the overall filter at the n-th time instant. Note that the term in the brackets is the *non-adaptive estimate* of the interference vector at the n^{th} symbol.

This algorithm does not require any training information. At no point is the symbol value required to arrive at the set of adaptive weights. Also, even if the interference were cancelled exactly, the output signal y would not be the transmitted symbols because the MOE method cannot account for the unknown complex signal amplitude (α).

As mentioned earlier, the drawback of this algorithm is slow convergence. Figure 8 compares the convergence of the LMS and MOE algorithm for a sample interference scenario, in terms of the output signal to interference plus noise ratio[2]. As can be seen the SINR converges to nearly the same value, however the MOE requires significantly more training. A rule of thumb is that LMS requires about 10 times the number of weights to converge while MOE at least twice that. For this reason, in practice, if only a limited number of training symbols were available, one could use the LMS or RLS algorithms in this training phase for relatively fast convergence and then switch

[2]My thanks to Prof. T.J. Lim for use of this plot

to the blind MOE algorithm after the training sequence is extinguished. Such a scheme, however, requires a training sequence as well as the channel vector (possibly estimated using the training data).

In addition, note that the LMS algorithm does not require the channel vector. The MOE algorithm, on the other hand, "trains" on the steering vector.

8 Constant Modulus Algorithm

The final iterative algorithm we will consider is a constant modulus algorithm which requires neither the steering vector nor training data. This algorithm looks for a signal with a constant magnitude (modulus) within the received data vector (which comprises signal, interference and noise). This algorithm is only applicable within the class of modulation schemes that use symbols of equal power, e.g., BPSK, QPSK).

We are essentially looking for a signal of a particular magnitude (say α) within the received data vector $\mathbf{x}$. The functional we are trying to minimize is therefore

$$f(\mathbf{w}) = \mathrm{E}\left\{ \left| \left|\mathbf{w}^H\mathbf{x}\right|^p - |\alpha|^p \right|^q \right\} \tag{58}$$

One can choose the parameters p and q arbitrarily, though usually they are chosen to be small integers. To illustrate the development of the CMA algorithm, consider the case of $p = 1$ and $q = 2$, and setting $\alpha = 1$, the functional we are minimizing is

$$\begin{aligned} f(\mathbf{w}) &= \mathrm{E}\left\{ \left| \left|\mathbf{w}^H\mathbf{x}\right| - 1 \right|^2 \right\} \\ &= \mathrm{E}\left\{ \left|\, |y| - 1\, \right|^2 \right\} \\ \Rightarrow \nabla f &= 2\frac{\partial f}{\partial \mathbf{w}^*} = 2\left(|y| - 1\right)\mathbf{x}\frac{y}{|y|} \end{aligned} \tag{59}$$

$$\Rightarrow \mathbf{w}_{n+1} = \mathbf{w}_n - 2\mu\left(y - \frac{y}{|y|}\right)\mathbf{x}_n \tag{60}$$

A significant problem with the CMA algorithm is that without any external information, it can identify only one signal. Usually, this is the signal with the greatest power. There are variants of the CMA algorithm that alleviate this problem somewhat.

9 Impact of Digitization

Finally, as a practical matter, the adaptive algorithms presented here would be implemented after digitizing the received signals with an A/D. However, by digitizing the signals, we introduce

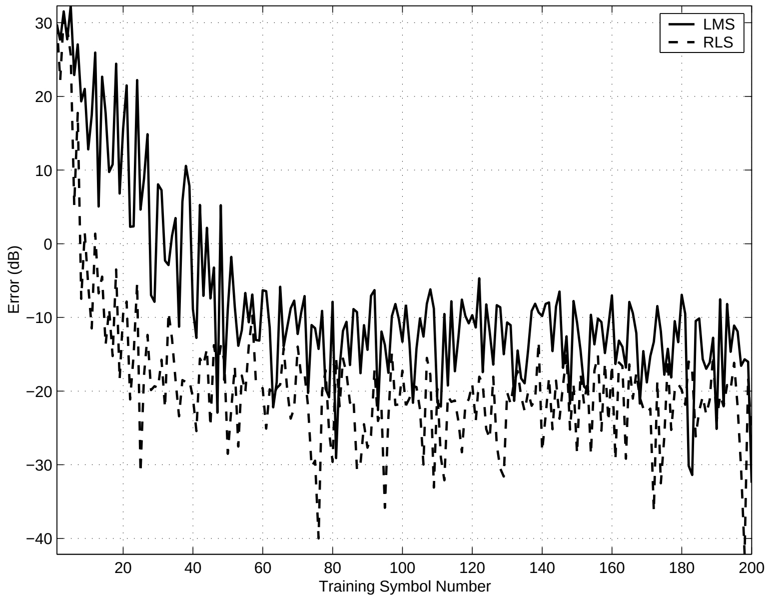

Figure 9: Residual error in the LMS and RLS techniques

quantization noise which has not been accounted for in our analysis. Consider a system where the signal (x) at each element is sampled and digitized using b_x bits and the weight is represented using b_z bits. Due to digitization, the signal that will be processed ($\bar{x}$) is the true signal (x) plus some quantization noise (n_x). Similarly for the weights. Therefore,

$$\bar{x} = x + n_x, \tag{61}$$

$$\bar{w} = w + n_w. \tag{62}$$

In the beamforming approach we are multiplying these two quantized values. The output $\bar{y} = \bar{w}^* \bar{x}$.

$$\begin{aligned} \bar{y} &= \bar{w}^* \bar{x} & (63) \\ &= (w + n_w)^* (x + n_x) \\ &= w^* x + n_w^* x + w^* n_x + n_w^* n_x & (64) \end{aligned}$$

Representing as σ_x^2 the average power in the signal x and $\sigma_w^2 = E(|w|^2)$, the useful power in the output signal y is $\sigma_x^2 \sigma_w^2$. Similarly, if σ_{nx}^2 and σ_{nw} are the powers in the quantization noise terms n_x and n_w, assuming independence between x and w, the average quantization noise power is $\sigma_x^2 \sigma_{nw}^2 + \sigma_w^2 \sigma_{nx}^2 + \sigma_{nw}^2 \sigma_{nw}^2$. The signal-to-quantization-noise ration (SQNR) is therefore

$$\begin{aligned} \frac{1}{\text{SQNR}} &= \frac{\sigma_x^2 \sigma_{nw}^2 + \sigma_w^2 \sigma_{nx}^2 + \sigma_{nw}^2 \sigma_{nw}^2}{\sigma_x^2 \sigma_w^2} & (65) \\ &= \frac{\sigma_{nw}^2}{\sigma_x^2} + \frac{\sigma_{nx}^2}{\sigma_w^2} + \frac{\sigma_{nw}^2 \sigma_{nx}^2}{\sigma_w^2 \sigma_x^2} & (66) \end{aligned}$$

Now, if the dynamic range of x (w) is R_x (R_w), and one were using b_x (b_z) bits to discretize the real and imaginary parts x (w), the real and imaginary parts of the quantization error n_x is uniformly distributed between $(-\Delta/2, \Delta/2)$ where $\Delta = \frac{R_x}{2^{b_x}}$. The error power each the real and imaginary components is then $\Delta^2/12$, i.e., the total error power is $\Delta^2/6$. The SQNR is therefore,

$$\frac{1}{\text{SQNR}} = \frac{R_x^2}{6 \times 2^{2b_x} \times \sigma_x^2} + \frac{R_w^2}{6 \times 2^{2b_w} \times \sigma_w^2} + \frac{R_x^2 R_w^2}{36 \times 2^{2b_x+2b_w} \times \sigma_w^2 \sigma_x^2} \tag{67}$$

The crucial conclusion is that each additional bit of discretization results in an extra 6dB of available SQNR. Without an adequate SQNR, there is *no point* in claiming a large gain in SINR from beamforming (or digital signal processing of any kind).

References

[1] A. Bourdoux and N. Khaled, "Joint TX-RX optimisation for MIMO-SDMA based on a null-space constraint," in *Proc. of IEEE Vehicular Technology Conference (VTC-02 Fall)*, (Vancouver, Canada), pp. 171–174, Sept. 2002.

[2] I. S. Reed, J. Mallett, and L. Brennan, "Rapid convergence rate in adaptive arrays," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 10, No. 6, pp. 853–863, Nov. 1974.

[3] I. Psaromiligkos and S. Batalama, "Interference-plus-noise covariance matrix estimation for adaptive space-time processing of DS/CDMA signals," in *IEEE Fall Vehicular Technology Conference*, pp. 2197–2204, Oct. 2000.