Chapter 11 Adaptive Array Processing

11.1. Introduction

The emphasis in this chapter is on adaptive array processing. For this purpose, a top level overview of phased array antennas is first introduced. Phased array antennas are capable of forming multiple beams at the transmitting or receiving modes. Beamforming can be carried out at the Radio frequency (RF), Intermediate Frequency (IF), base band, or digital levels. RF beamforming is the simplest and most common technique. In this case, multiple narrow beams are formed through the use of phase shifters. IF and base band beamforming require complex coherent hardware. However, the system is operated at lower frequencies where tolerance is not as critical. Digital beamforming is more flexible than RF, IF, or base band techniques, but it requires a demanding level of processing hardware.

Adaptive arrays mostly employ phased arrays to automatically sense and eliminate unwanted signals entering the radar's Field of View (FOV) while enhancing reception about the desired target returns. For this purpose, adaptive arrays utilize a rather complicated combination of hardware and require demanding levels of software implementation. Through feedback networks, a proper set of complex weights is computed and applied to each channel of the array. A successful implementation of adaptive arrays depends heavily on two factors: first, a proper choice of the reference signal, which is used for comparison against the received target/jammer returns. A good estimate of the reference signal makes the computation of the weights systematic and effective. On the other hand, a bad estimate of the reference signal increases the array's adapting time and limits the system to impractical (non-real time) situations. Second, a fast (real time) computation of the optimum weights is essential. There have been many algorithms developed for this purpose. Nevertheless, they all share a common problem, that is, the computation of the inverse of a complex matrix. This drawback has limited the implementation of adaptive arrays to experimental systems or small arrays.

11.2. General Arrays

An array is a composite antenna formed from two or more basic radiators. Each radiator is denoted as an element. The elements forming an array could be dipoles, dish reflectors, slots in a wave guide, or any other type of radiator. Array antennas synthesize narrow directive beams that may be steered, mechanically or electronically, in many directions. Electronic steering is achieved by controlling the phase of the current feeding the array elements. Arrays with electronic beam steering capability are called phased arrays. Phased array antennas, when compared with other simple antennas such as dish reflectors, are costly and complicated to design. However, the inherent flexibility of phased array antennas to steer the beam electronically and also the need for specialized multifunction radar systems have made phased array antennas attractive for radar applications.

Figure 11.1 shows the geometrical fundamentals associated with this problem. Consider the radiation source located at (x_1, y_1, z_1) with respect to a phase reference at $(0, 0, 0)$. The electric field measured at far field point P is

$$E(\theta, \phi) = I_0 \frac{e^{-jkR_1}}{R_1} f(\theta, \phi) \tag{11.1}$$

where I_0 is the complex amplitude, $k = 2\pi/\lambda$ is the wave number, and $f(\theta, \phi)$ is the radiation pattern.

Now, consider the case where the radiation source is an array made of many elements, as shown in Fig. 11.2. The coordinates of each radiator with respect to the phase reference are (x_i, y_i, z_i), and the vector from the origin to the ith element is given by

$$\vec{r}_i = \hat{a}_x x_i + \hat{a}_y y_i + \hat{a}_z z_i \tag{11.2}$$

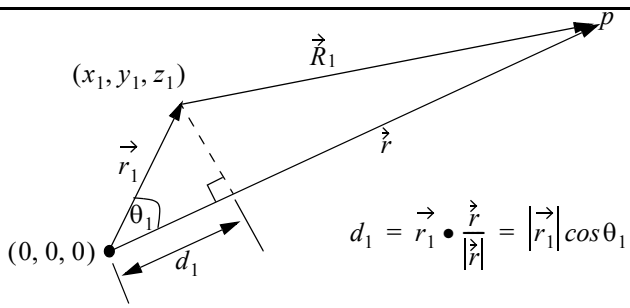

Figure 11.1. Geometry for an array antenna. Single element.

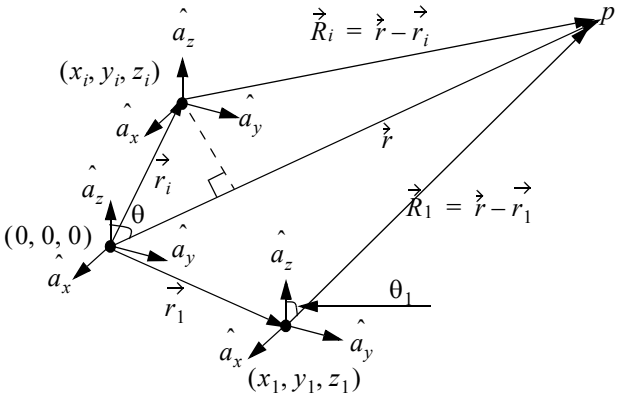

Figure 11.2. Geometry for an array antenna.

The far field components that constitute the total electric field are

$$E_i(\theta, \phi) = I_i \frac{e^{-jkR_i}}{R_i} f(\theta_i, \phi_i) \tag{11.3}$$

where

$$\begin{aligned} R_i &= \left|\vec{R}_i\right| = \left|\vec{r} - \vec{r}_i\right| = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} \\ &= r\sqrt{1 + (x_i^2 + y_i^2 + z_i^2)/r^2 - 2(xx_i + yy_i + zz_i)/r^2} \end{aligned} \tag{11.4}$$

Using spherical coordinates, where $x = r\sin\theta\cos\varphi$, $y = r\sin\theta\sin\varphi$, and $z = r\cos\theta$, yields

$$\frac{(x_i^2 + y_i^2 + z_i^2)}{r^2} = \frac{\left|\vec{r}_i\right|^2}{r^2} \ll 1 \tag{11.5}$$

Thus, a good approximation (using binomial expansion) for Eq. (11.4) is

$$R_i = r - r(x_i \sin\theta\cos\phi + y_i \sin\theta\sin\phi + z_i\cos\theta) \tag{11.6}$$

It follows that the phase contribution at the far field point from the ith radiator with respect to the phase reference is

$$e^{-jkR_i} = e^{-jkr}\ e^{jk(x_i \sin\theta\cos\phi + y_i \sin\theta\sin\phi + z_i\cos\theta)} \tag{11.7}$$

Remember, however, that the unit vector $\hat{r}_0$ along the vector $\vec{r}$ is

$$\hat{r}_0 = \frac{\vec{r}}{|\vec{r}|} = \hat{a}_x \sin\theta\cos\phi + \hat{a}_y \sin\theta\sin\phi + \hat{a}_z \cos\theta \tag{11.8}$$

Hence, we can rewrite Eq. (11.7) as

$$e^{-jkR_i} = e^{-jkr}\ e^{jk(\hat{r}_i \bullet \hat{r}_0)} = e^{-jkr} e^{j\Psi_i(\theta,\phi)} \tag{11.9}$$

Finally, by virtue of superposition, the total electric field is

$$E(\theta,\phi) = \sum_{i=1}^{N} I_i e^{j\Psi_i(\theta,\phi)} \tag{11.10}$$

which is known as the array factor for an array antenna where the complex current for the ith element is I_i.

In general, an array can be fully characterized by its array factor. This is true since knowing the array factor provides the designer with knowledge of the array's (1) 3-dB beamwidth, (2) null-to-null beamwidth, (3) distance from the main peak to the first side-lobe, (4) height of the first side-lobe as compared to the main beam, (5) location of the nulls, (6) rate of decrease of the side-lobes, and (7) grating lobes' locations.

11.3. Linear Arrays

Figure 11.3 shows a linear array antenna consisting of N identical elements. The element spacing is d (normally measured in wavelength units). Let element #1 serve as a phase reference for the array. From the geometry, it is clear that an outgoing wave at the nth element leads the phase at the $(n+1)th$ element by $kd\sin\theta$, where $k = 2\pi/\lambda$. The combined phase at the far field observation point P is independent of ϕ and can be written as

$$\Psi(\theta,\phi) = k(\hat{r}_n \bullet \hat{r}_0) = (n-1)kd\sin\theta \tag{11.11}$$

Thus, from Eq. (11.10), the electric field at a far field observation point with direction-sine equal to $\sin\theta$ (assuming isotropic elements) is

$$E(\sin\theta) = \sum_{n=1}^{N} e^{j(n-1)(kd\sin\theta)} \tag{11.12}$$

Expanding the summation in Eq. (11.12) yields

$$E(\sin\theta) = 1 + e^{jkd\sin\theta} + \ldots + e^{j(N-1)(kd\sin\theta)} \tag{11.13}$$

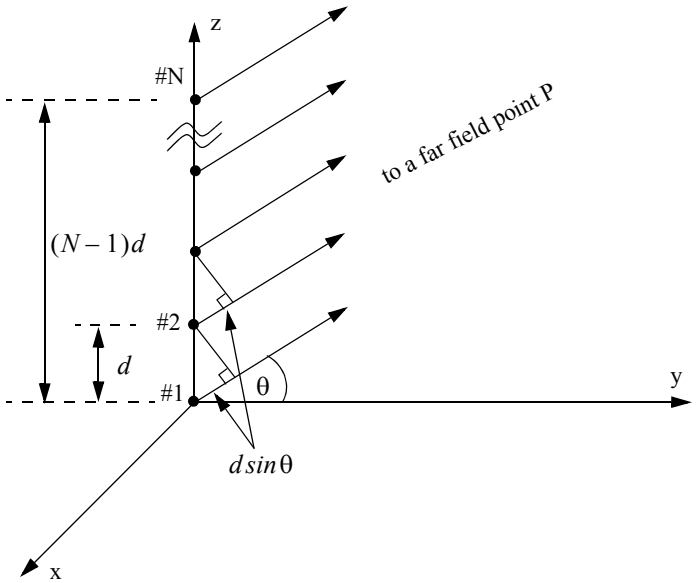

Figure 11.3. Linear array of equally spaced elements.

The right-hand side of Eq. (11.13) is a geometric series, which can be expressed in the form

$$1 + a + a^2 + a^3 + \ldots + a^{(N-1)} = \frac{1 - a^N}{1 - a} \tag{11.14}$$

Replacing a by $e^{jkd\sin\theta}$ yields

$$E(\sin\theta) = \frac{1 - e^{jNkd\sin\theta}}{1 - e^{jkd\sin\theta}} = \frac{1 - (\cos Nkd\sin\theta) - j(\sin Nkd\sin\theta)}{1 - (\cos kd\sin\theta) - j(\sin kd\sin\theta)} \tag{11.15}$$

The far field array intensity pattern is then given by

$$|E(\sin\theta)| = \sqrt{E(\sin\theta)E^*(\sin\theta)} \tag{11.16}$$

Substituting Eq. (11.15) into Eq. (11.16) and collecting terms yield

$$|E(\sin\theta)| = \sqrt{\frac{(1 - \cos Nkd\sin\theta)^2 + (\sin Nkd\sin\theta)^2}{(1 - \cos kd\sin\theta)^2 + (\sin kd\sin\theta)^2}} \tag{11.17}$$

which can be written as

$$|E(\sin\theta)| = \sqrt{\frac{1-\cos Nkd\sin\theta}{1-\cos kd\sin\theta}} \tag{11.18}$$

and using the trigonometric identity $1-\cos\theta = 2(\sin\theta/2)^2$ yields

$$|E(\sin\theta)| = \left|\frac{\sin(Nkd\sin\theta/2)}{\sin(kd\sin\theta/2)}\right| \tag{11.19}$$

which is a periodic function of $kd\sin\theta$, with a period equal to 2π.

The maximum value of $|E(\sin\theta)|$, which occurs at $\theta = 0$, is equal to N. It follows that the normalized intensity pattern is equal to

$$|E_n(\sin\theta)| = \frac{1}{N}\left|\frac{\sin((Nkd\sin\theta)/2)}{\sin((kd\sin\theta)/2)}\right| \tag{11.20}$$

The normalized two-way array pattern (radiation pattern) is given by

$$G(\sin\theta) = |E_n(\sin\theta)|^2 = \frac{1}{N^2}\left(\frac{\sin((Nkd\sin\theta)/2)}{\sin((kd\sin\theta)/2)}\right)^2 \tag{11.21}$$

Figure 11.4 shows a plot of Eq. (11.21) versus $\sin\theta$ for $N = 8$. This plot can be reproduced using the following MATLAB code.

```
% Use this code to produce figure 11.4a and 11.4b
clear all; close all;
eps = 0.00001;
k = 2*pi;
theta = -pi : pi / 10791 : pi;
var = sin(theta);
nelements = 8;
d = 1;        % d = 1;
num = sin((nelements * k * d * 0.5) .* var);
if(abs(num) <= eps)
  num = eps;
end
den = sin((k* d * 0.5) .* var);
if(abs(den) <= eps)
  den = eps;
end
pattern = abs(num ./ den);
maxval = max(pattern);
pattern = pattern ./ maxval;
figure(1)
plot(var,pattern)
xlabel('sine angle - dimensionless')
ylabel('Array pattern')
grid
```

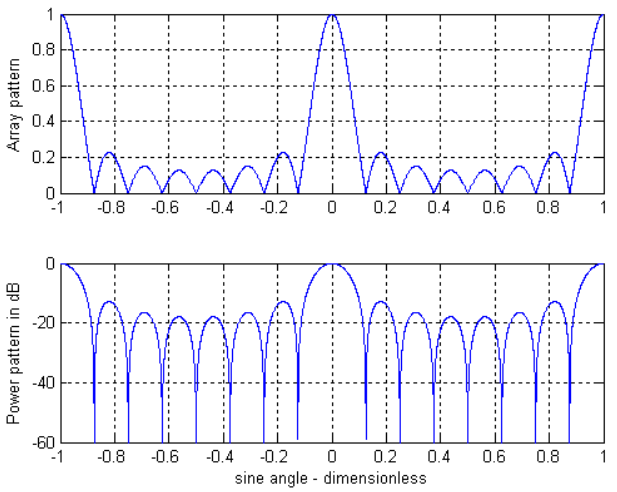

Figure 11.4a. Normalized radiation pattern for a linear array; $N = 8$; $d = \lambda$.

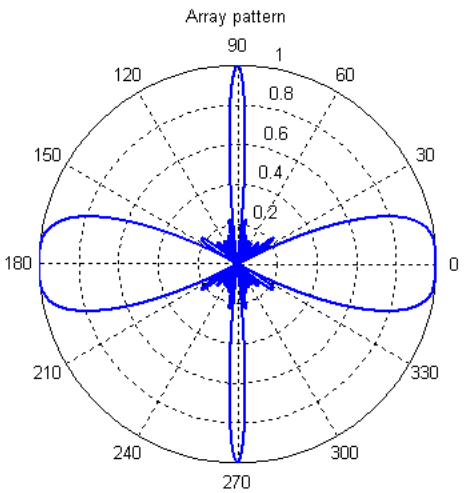

Figure 11.4b. Polar plot for the array pattern in Fig. 11.4a.

```
figure(2)
plot(var,20*log10(pattern))
axis ([-1 1 -60 0])
xlabel('sine angle - dimensionless')
ylabel('Power pattern in dB')
grid;
figure(3)
theta = theta +pi/2;
polar(theta,pattern)
title ('Array pattern')
```

The radiation pattern $G(sin\theta)$ has cylindrical symmetry about its axis $(sin\theta = 0)$ and is independent of the azimuth angle. Thus, it is completely determined by its values within the interval $(0 < \theta < \pi)$. The main beam of an array can be steered electronically by varying the phase of the current applied to each array element. Steering the main beam into the direction-sine $sin\theta_0$ is accomplished by making the phase difference between any two adjacent elements equal to $kdsin\theta_0$. In this case, the normalized radiation pattern can be written as

$$G(sin\theta) = \frac{1}{N^2}\left(\frac{sin[(Nkd/2)(sin\theta - sin\theta_0)]}{sin[(kd/2)(sin\theta - sin\theta_0)]}\right)^2 \quad \textbf{(11.22)}$$

If $\theta_0 = 0$, then the main beam is perpendicular to the array axis, and the array is said to be a broadside array. Alternatively, the array is called an endfire array when the main beam points along the array axis. The radiation pattern maxima are computed using L'Hopital's rule when both the denominator and numerator of Eq. (11.22) are zeros. More precisely,

$$\left(\frac{kdsin\theta}{2} = \pm m\pi\right);\ m = 0, 1, 2, \ldots \quad \textbf{(11.23)}$$

Solving for θ yields

$$\theta_m = asin\left(\pm\frac{\lambda m}{d}\right);\ m = 0, 1, 2, \ldots \quad \textbf{(11.24)}$$

where the subscript m is used as a maxima indicator. The first maximum occurs at $\theta_0 = 0$ and is denoted as the main beam (lobe). Other maxima occurring at $|m| \geq 1$ are called grating lobes. Grating lobes are undesirable and must be suppressed. The grating lobes occur at non-real angles when the absolute value of the arc-sine argument in Eq. (11.24) is greater than unity; it follows that $d < \lambda$. Under this condition, the main lobe is assumed to be at $\theta = 0$ (broadside array). Alternatively, when electronic beam steering is considered, the grating lobes occur at

$$\left|\sin\theta - \sin\theta_0\right| = \pm\frac{\lambda n}{d};\ \ n = 1, 2, \ldots \tag{11.25}$$

Thus, in order to prevent the grating lobes from occurring between $\pm 90°$, the element spacing should be $d < \lambda/2$.

The radiation pattern attains secondary maxima (side-lobes) when the numerator of Eq. (11.24) is maximum, or equivalently

$$\frac{Nkd\sin\theta}{2} = \pm(2l+1)\frac{\pi}{2};\ \ l = 1, 2, \ldots \tag{11.26}$$

Solving for θ yields

$$\theta_l = \text{asin}\left(\pm\frac{\lambda}{2d}\frac{2l+1}{N}\right);\ \ l = 1, 2, \ldots \tag{11.27}$$

where the subscript l is used as an indication of side-lobe maxima. The nulls of the radiation pattern occur when only the numerator of Eq. (11.24) is zero. More precisely,

$$\frac{N}{2}kd\sin\theta = \pm n\pi;\quad \begin{matrix} n = 1, 2, \ldots \\ n \neq N, 2N, \ldots \end{matrix} \tag{11.28}$$

Again solving for θ yields

$$\theta_n = \text{asin}\left(\pm\frac{\lambda}{d}\frac{n}{N}\right);\quad \begin{matrix} n = 1, 2, \ldots \\ n \neq N, 2N, \ldots \end{matrix} \tag{11.29}$$

where the subscript n is used as a null indicator. Define the angle that corresponds to the half power point as θ_h. It follows that the half power (3-dB) beamwidth is $2\left|\theta_m - \theta_h\right|$. This occurs when

$$\frac{N}{2}kd\sin\theta_h = 1.391\ \ radians \Rightarrow \theta_h = \text{asin}\left(\frac{\lambda}{2\pi d}\frac{2.782}{N}\right) \tag{11.30}$$

In order to reduce the side-lobe levels, the array must be designed to radiate more power toward the center and much less at the edges. This can be achieved through tapering (windowing) the current distribution over the face of the array. There are many possible tapering sequences that can be used for this purpose. However, as known from spectral analysis, windowing reduces side-lobe levels at the expense of widening the main beam. Thus, for a given radar application, the choice of the tapering sequence must be based on the trade-off between side-lobe reduction and main-beam widening.

Figures 11.5 through Fig. 11.13 show plots of the array gain pattern versus steering angle for a few. These plots can be reproduced using the following MATLAB code

```
% produce figures 11.5 through 11.13
clear all; close all; clc
win = hamming(19);
[theta,patternr,patterng] = linear_array(19, 0.5, 0, -1, -1, -3);
figure(5)
plot(theta, patterng,'linewidth',1.5)
xlabel('Steering angle in degrees'); ylabel('Antenna gain pattern in dB')
title('N = 19; d = 0.5\lambda; \theta = 0 degrees; Perfect phase shifters')
grid on; axis tight
[theta, patternr, patterng] = linear_array(19, 0.5, 0, 1, win, -3);
figure(6)
plot(theta, patterng,'linewidth',1.5)
xlabel('Steering angle - degrees')
ylabel('Antenna gain pattern - dB')
title('N = 19; d = 0.5\lambda; \theta = 0 degrees; Perfect phase shifters; Hamming window')
grid on; axis tight
[theta, patternr, patterng] = linear_array(19, 0.5, -15, -1, -1, 3);
figure(7)
plot(theta, patterng,'linewidth',1.5)
xlabel('Steering angle in degrees'); ylabel('Antenna gain pattern in dB')
title('N = 19; d = 0.5\lambda; \theta = -15 degrees; 3-bits phase shifters')
grid on; axis tight
[theta, patternr, patterng] = linear_array(19, 0.5, 5, 1, win, 3);
figure(8)
plot(theta, patterng,'linewidth',1.5)
xlabel('Steering angle - degrees')
ylabel('Antenna gain pattern - dB')
title('N = 19; d = 0.5\lambda; \theta = 5 degrees; 3-bits phase shifters; Hamming window')
grid on; axis tight
[theta, patternr, patterng] = linear_array(19, 0.5, 25, 1, win, 3);
figure(9)
plot(theta, patterng,'linewidth',1.5)
xlabel('Steering angle in degrees')
ylabel('Antenna gain pattern - dB')
title('N = 19; d = 0.5\lambda; \theta = 25 degrees; 3-bits phase shifters; Hamming window')
grid on; axis tight
[theta, patternr, patterng] = linear_array(19, 1.5, 48, -1, -1, -3);
figure(10)
plot(theta, patterng,'linewidth',1.5)
xlabel('Steering angle in degrees'); ylabel('Antenna gain pattern in dB')
title('N = 19; d = 1.5\lambda; \theta = 48 degrees; Perfect phase shifters')
```

```
grid on; axis tight
[theta, patternr, patterng] = linear_array(19, 1.5, 48, 1, win, -3);
figure(11)
plot(theta, patterng,'linewidth',1.5)
xlabel('Steering angle in degrees'); ylabel('Antenna gain pattern in dB')
title('N = 19; d = 1.5\lambda; \theta = 48 degrees; Perfect phase shifters; Hamming
window')
grid on; axis tight
[theta, patternr, patterng] = linear_array(19, 1.5, -53, -1, -1, 3);
figure(12)
plot(theta, patterng,'linewidth',1.5)
xlabel('Steering angle in degrees'); ylabel('Antenna gain pattern in dB')
title('N = 19; d = 1.5\lambda; \theta = -53 degrees; 3-bits phase shifters')
grid on; axis tight
[theta, patternr, patterng] = linear_array(19, 1.5, -33, 1, win, 3);
figure(13)
plot(theta, patterng,'linewidth',1.5)
xlabel('Steering angle in degrees')
ylabel('Antenna gain pattern - dB')
title('N = 19; d = 1.5\lambda; \theta = -33 degrees; 3-bits phase shifters; ...
Hamming window')
grid on;
axis tight
```

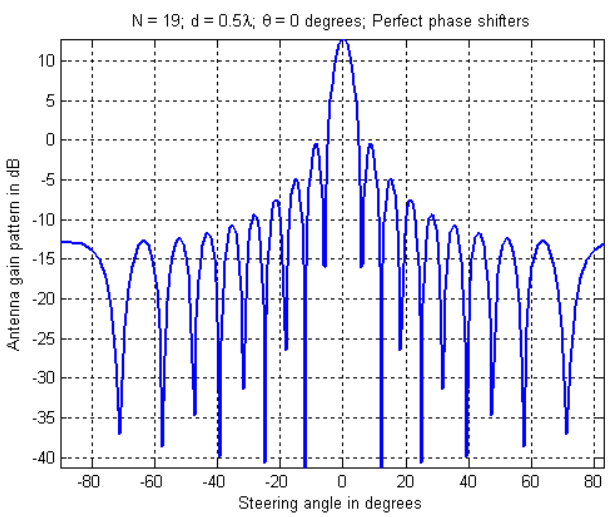

Figure 11.5. Array gain pattern.

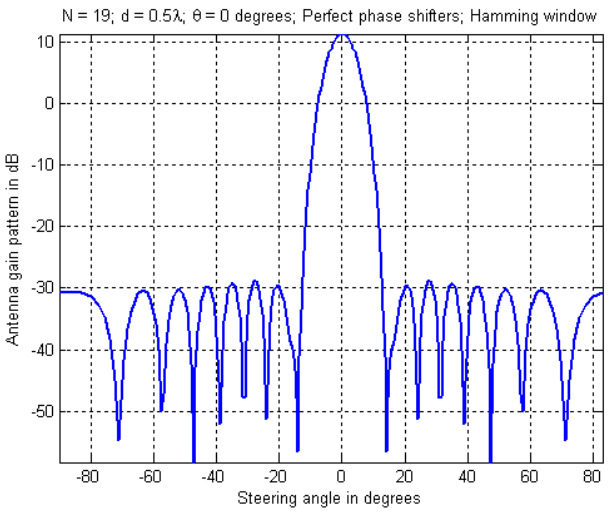

Figure 11.6. Array gain pattern.

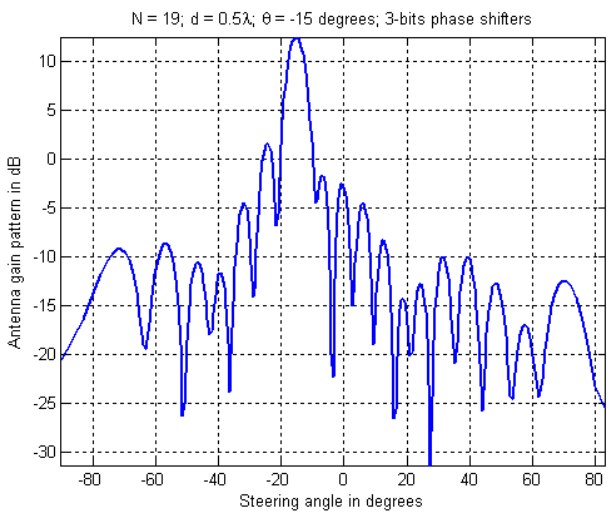

Figure 11.7. Array gain pattern.

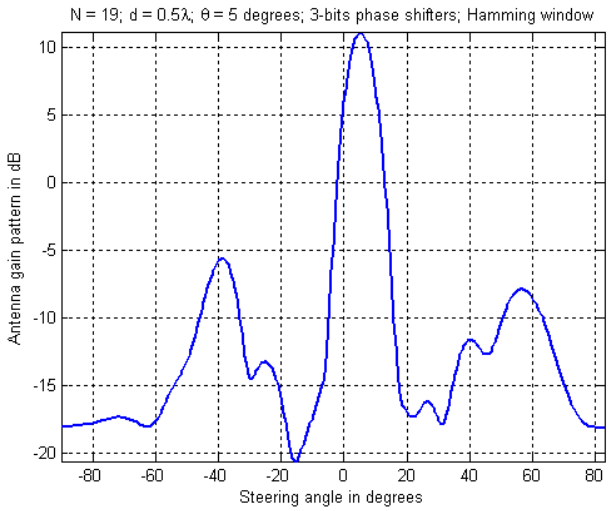

Figure 11.8. Array gain pattern.

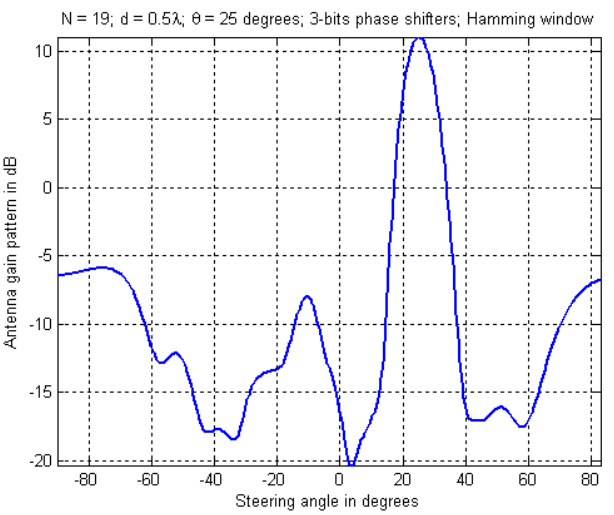

Figure 11.9. Array gain pattern.

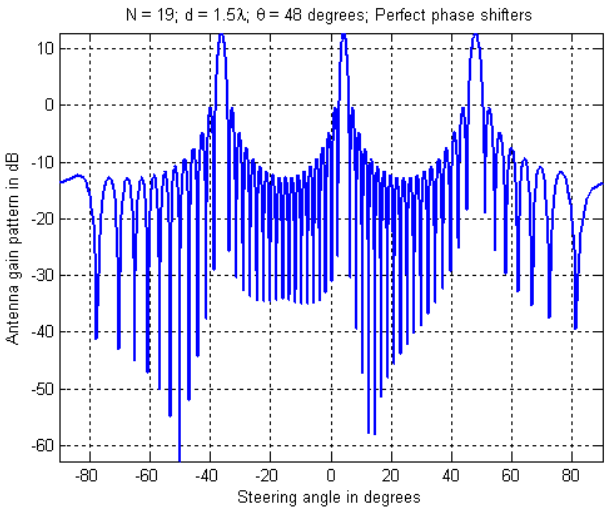

Figure 11.10. Array gain pattern.

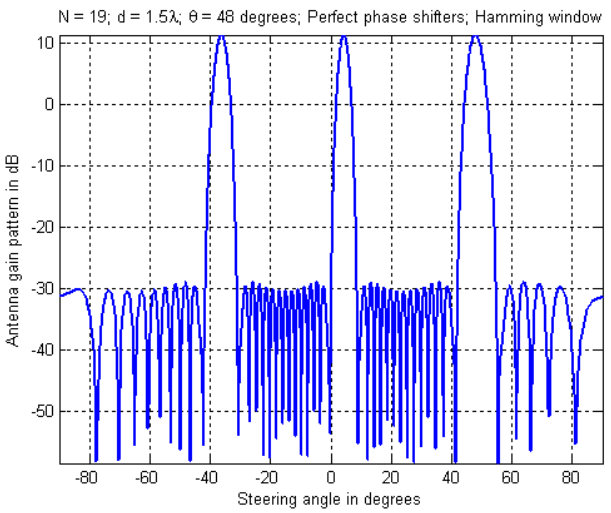

Figure 11.11. Array gain pattern.

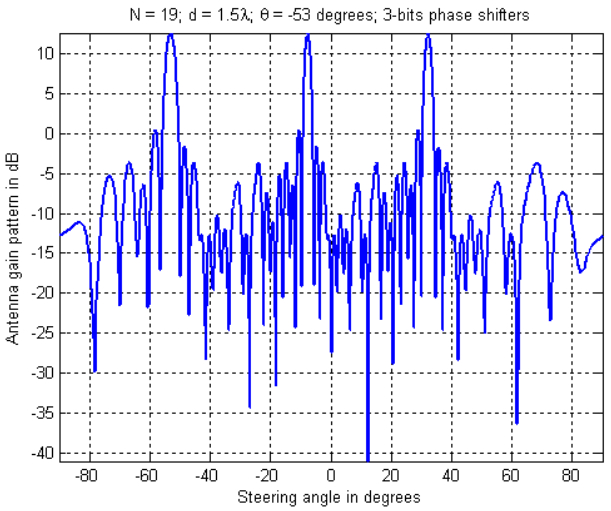

Figure 11.12. Array gain pattern.

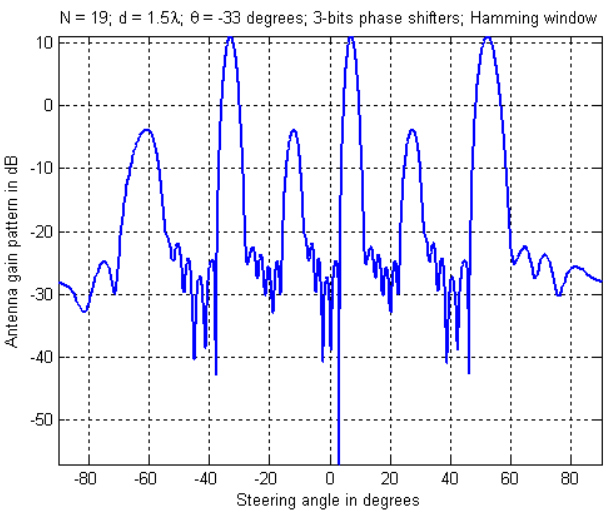

Figure 11.13. Array gain pattern.

11.4. Nonadaptive Beamforming

In adaptive beamforming the beam of interest is formed (generated) by continuously changing a set of weights through feedback circuits to minimize an output error signal. Nonadaptive or conventional beamformers do the same thing in the sense that the beam of interest is generated using a set of unique weights. Except in this case, these weights are determined a priori so that interference from a specific angle of arrival is minimized or eliminated. Different sets of weights will produce nulls in different directions in the array's field of view.

Consider a linear array of N equally spaced elements, and a plane wave $exp(j2\pi f_0 t))$ incident on the aperture with direction-sine $sin\theta$, as shown in Fig. 11.14. The weights $w_i,\ \ i = 0, 1, \ldots N-1$ are, in general, complex constants. The output of the beamformer is

$$y(t) = \sum_{n=0}^{N-1} w_n x_n(t-\tau_n) \tag{11.31}$$

$$\tau_n = n\frac{d}{c}sin\theta;\ \ n = 0, 1, \ldots, (N-1) \tag{11.32}$$

where d is the element spacing and c is the speed of light. Fourier transformation of Eq. (11.31) yields

$$Y(\omega) = \sum_{n=0}^{N-1} w_n X_n(\omega) exp(-j\omega\tau_n) = \sum_{n=0}^{N-1} w_n X_n(\omega) e^{-jn\Delta\theta} \tag{11.33}$$

The phase term $\Delta\theta$ is defined as

$$\Delta\theta = 2\pi f_0 \frac{d}{c} sin\theta = \frac{2\pi}{\lambda} d sin\theta \tag{11.34}$$

$\omega = 2\pi f_0$ and $f_0/c = 1/\lambda$. Eq. (11.33) can be written in vector form as

$$\mathbf{Y} = \mathbf{s}^{\dagger}\mathbf{x} \tag{11.35}$$

$$\mathbf{s}^{\dagger} = \begin{bmatrix} 1 & e^{j\Delta\theta} & \ldots & e^{j(N-1)\Delta\theta} \end{bmatrix} \tag{11.36}$$

$$\mathbf{x}^{\dagger} = \begin{bmatrix} w_0 X_o & w_1 X_1 & \ldots & \ldots w_{N-1} X_{N-1} \end{bmatrix}^{*} \tag{11.37}$$

where the superscripts * and †, respectively, indicate complex conjugate and complex conjugate transpose.

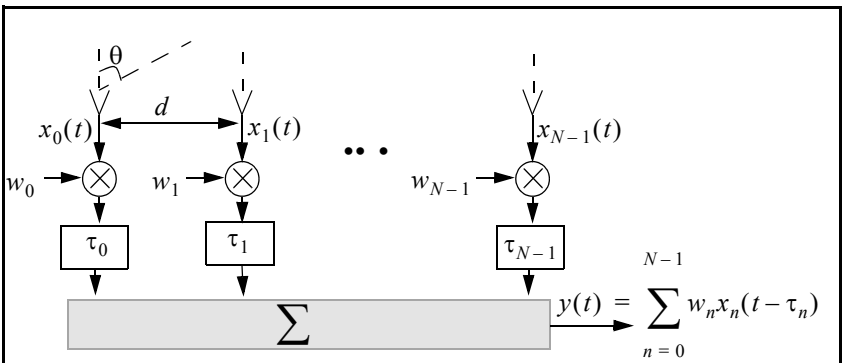

Figure 11.14. A linear array of size N, element spacing d, and an incident plane wave defined by $sin\,\theta$.

Let A_1 be the amplitude of the wavefront defined by $sin\,\theta_1$; it follows that the vector $\mathbf{x}$ is given by

$$\mathbf{x} = A_1 \mathbf{s}_1^* \tag{11.38}$$

where $\mathbf{s}_1$ is a steering vector can be written as,

$$\mathbf{s}^\dagger_1 = \left[w_0 \quad w_1 e^{-j\Delta\theta_1} \ldots w_{N-1} e^{-j(N-1)\Delta\theta_1} \right] \; ; \; \Delta\theta_1 = \frac{2\pi d}{\lambda} \cdot sin\,\theta_1 \tag{11.39}$$

Using this notation, Eq. (11.35) can be expressed in the form

$$\mathbf{Y} = \mathbf{s}^\dagger \mathbf{x} = A_1 \mathbf{s}^\dagger \mathbf{s}^*_1 \tag{11.40}$$

The array pattern of the beam steered at θ_1 is computed as the expected value of $\mathbf{Y}$. In other words, the power spectrum density for the beamformer output is given by

$$S(k) = E[\mathbf{Y}\mathbf{Y}^\dagger] = P_1 \mathbf{s}^\dagger \Re \mathbf{s} \tag{11.41}$$

where $P_1 = E[|A_1|^2]$ and $\Re$ is the correlation matrix given by

$$\Re = E\{\mathbf{s}_1 \mathbf{s}^\dagger_1\} \tag{11.42}$$

Consider L incident plane waves with directions of arrival defined by

$$\Delta\theta_i = \frac{2\pi d}{\lambda} sin\,\theta_i; \; i = 1, L \tag{11.43}$$

The nth sample at the output of the mth sensor is

$$y_m(n) = \upsilon(n) + \sum_{i=1}^{L} A_i(n) exp(-jm\Delta\theta_i);\ m = 0, N-1 \tag{11.44}$$

where $A_i(n)$ is the amplitude of the ith plane wave and $\upsilon(n)$ is white, zero-mean noise with variance σ_υ^2, and it is assumed to be uncorrelated with the signals. Equation (11.44) can be written in vector notation as

$$\mathbf{y}(n) = \upsilon(n) + \sum_{i=1}^{L} A_i(n)\mathbf{s}_i^* \tag{11.45}$$

A set of L steering vectors is needed to simultaneously form L beams. Define the steering matrix $\aleph$ as

$$\aleph = \left[\mathbf{s}_1\ \mathbf{s}_2\ \ldots\ \mathbf{s}_L\right] \tag{11.46}$$

Then the autocorrelation matrix of the field measured by the array is

$$\Re = E\{\mathbf{y}_m(n)\mathbf{y}_m^\dagger(n)\} = \sigma_\upsilon^2 \boldsymbol{I} + \aleph \boldsymbol{C} \aleph^\dagger \tag{11.47}$$

where $\boldsymbol{C} = dig\left[P_1\ P_2\ \ldots\ P_L\right]$, and $\boldsymbol{I}$ is the identity matrix.

For example, consider the case depicted in Fig. 11.15, where an interfering signal located at angle $\theta_i = \pi/6$ off the antenna boresight. The desired signal is at $\theta_t = 0°$. The desired output should contain only the signal $s(t)$. From Eq. (11.33) and Eq. (11.34) the desired output is

$$y_d(t) = \sum_{n=0}^{1} w_n x_n(t - \tau_{n_t}) = w_0 x_0 + w_1 x_1 e^{-j\frac{2\pi}{\lambda} d \sin\theta_t} \tag{11.48}$$

Since the angle $\theta_t = 0°$, it follows that

$$y_d(t) = \{A e^{j2\pi f_0 t}\}\{(w_{0R} + jw_{0I}) + (w_{1R} + jw_{1I})\} \tag{11.49}$$

$$\begin{aligned} w_0 &= w_{0R} + jw_{0I} \\ w_1 &= w_{1R} + jw_{1I} \end{aligned} \tag{11.50}$$

Thus, in order to produce the desired signal, $s(t)$, at the output of the beamformer, it is required that

$$\begin{aligned} w_{0R} + w_{1R} = 1 &\Rightarrow w_{0R} = 1 - w_{1R} \\ w_{0I} + w_{1I} = 0 &\Rightarrow w_{0I} = -w_{1I} \end{aligned} \tag{11.51}$$

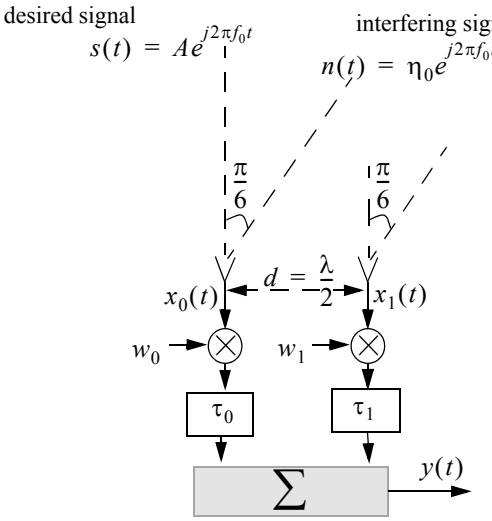

Figure 11.15. Two element array with an interfering signal at $\theta_i = \pi/6$.

Next, the output due to the interfering signal is

$$y_i(t) = \sum_{n=0}^{1} w_n x_n(t-\tau_{n_i}) = w_0 x_0 + w_1 x_1 e^{-j\frac{2\pi}{\lambda} d \sin\theta_i} \tag{11.52}$$

Since the angle $\theta_i = \pi/6$, it follows that

$$y_d(t) = \{\eta_0 e^{j2\pi f_0 t}\}\{(w_{0R} + jw_{0I}) - j(w_{1R} + jw_{1I})\} \tag{11.53}$$

and in order to eliminate the interference signal from the output of the beamformer, it is required that

$$\begin{aligned} w_{0R} + w_{1I} = 0 &\Rightarrow w_{0R} = -w_{1I} \\ w_{0I} - w_{1R} = 0 &\Rightarrow w_{0I} = w_{1R} \end{aligned} \tag{11.54}$$

Solving Eq. (11.51) and Eq. (11.54) yields

$$w_{0R} = \frac{1}{2}; w_{0I} = \frac{1}{2}; w_{1R} = \frac{1}{2}; w_{1I} = \frac{-1}{2} \tag{11.55}$$

Using the weights given in Eq. (11.55) will allow the desired signal to get through the beamformer unaffected; however, the interference signal will be completely eliminated from the output.

11.5. Adaptive Array Processing

11.5.1. Adaptive Signal Processing Using Least Mean Squares (LMS)

Adaptive signal processing evolved as a natural evolution from adaptive control techniques of time varying systems. Advances in digital processing computation techniques and associated hardware have facilitated maturing adaptive processing techniques and algorithms. Consider the basic adaptive digital system shown in Fig. 11.16. The system input is the sequence $x[k]$ and its output is the sequence $y[k]$. What differentiates adaptive from nonadaptive systems is that in adaptive systems the transfer function $H_k(z)$ is now time varying. The arrow through the transfer function box is used to indicate adaptive processing (or time varying transfer function). The sequence $d[k]$ is referred to as the *desired* response sequence. The error sequence is the difference between the desired response and the actual response. Remember that the desired sequence is not completely known; otherwise, if it were completely known, one would not need any adaptive processing to compute it. The definition of this desired response is dependent on the system specific requirements.

Many different techniques and algorithms have been developed to minimize the error sequence. Using one technique over another depends heavily on the operating environment under consideration. For example, if the input sequence is a stationary random process, then minimizing the error signal is nothing more than solving the least mean squares problem. However, in most adaptive processing systems the input signal is a nonstationary process. In this section the least mean squares technique is examined.

The least mean squares (LMS) algorithm is the most commonly utilized algorithm in adaptive processing, primary because of its simplicity. The time varying transfer function of order L can be written as a Finite Impulse Response (FIR) filter defined by

$$H_k(z) = b_0 + b_1 z^{-1} + \ldots + b_L z^{-L} \tag{11.56}$$

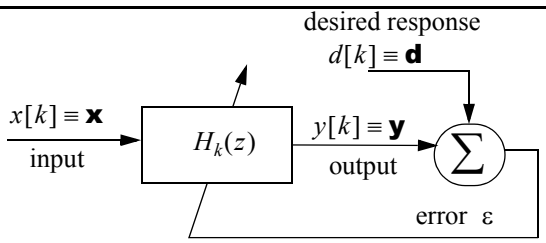

Figure 11.16. Basic adaptive system.

The input output relationship is given by the discrete convolution

$$y(k) = \sum_{n=0}^{L} b_n(k)x(k-n) \tag{11.57}$$

The goal of the adaptive LMS process is to adjust the filter coefficients toward an optimum minimum mean square error (MMSE). The most common approach to achieving this MMSE utilizes the method of steepest descent. For this purpose, define the filter coefficients in vector notation as

$$\mathbf{b}_k = \left[b_o(k)\ b_1(k)\ldots\quad b_L(k)\right]^{\dagger} \tag{11.58}$$

then

$$\mathbf{b}_{k+1} = \mathbf{b}_k - \mu\nabla_k \tag{11.59}$$

where μ is a parameter that controls how fast the error converges to the desired MMSE value, and the gradient vector ∇_k is defined by

$$\nabla_k = \frac{\partial}{\partial \mathbf{b}_k}E[\varepsilon_k^2] = \left[\frac{\partial}{\partial \mathbf{b}_0(k)}E[\varepsilon_k^2]\ \ldots\ \frac{\partial}{\partial \mathbf{b}_L(k)}E[\varepsilon_k^2]\right]^{\dagger} \tag{11.60}$$

As clearly indicated by Eq. (11.59) the adaptive filter coefficients update rate is proportional to the negative gradient; thus, if the gradient is known at each step of the adaptive process, then better computation of the coefficient is obtained. In other words, the MMSE decreases from step k to step $k+1$. Of course, once the solution is found the gradient becomes zero and the coefficient will not change any more.

When the gradient is not known, estimates of the gradient are used based only on the instantaneous squared error. These estimates are defined by

$$\hat{\nabla}_k = \frac{\partial}{\partial \mathbf{b}_k}[\varepsilon_k^2] = 2\varepsilon_k\frac{\partial}{\partial \mathbf{b}_k}(d_k - y_k) \tag{11.61}$$

Since the desired sequence $d[k]$ is independent from the output $y[k]$, Eq. (11.61) can be written as

$$\hat{\nabla}_k = -2\varepsilon_k\mathbf{x}_k \tag{11.62}$$

where the vector $\mathbf{x}_k$ is the input signal sequence. Substituting Eq. (11.62) into Eq. (11.59) yields

$$\mathbf{b}_{k+1} = \mathbf{b}_k + 2\varepsilon_k\mu\mathbf{x}_k \tag{11.63}$$

The choice of the convergence parameter μ plays a significant role in determining the system performance. This is clear because as indicated by Eq. (11.63), a successful implementation of the LMS algorithm depends on the input signal, the choice of the desired signal, and the convergence parameter. Much research and effort has been devoted toward selecting the optimal value for μ. Nonetheless, no universal value has been found. However, a range for this parameter has been determined to be $0 < \mu < 1$.

Often, a normalized value for the convergence parameter μ_N can be used instead of its absolute value. That is,

$$\mu_N = \frac{\mu}{(L+1)\sigma^2} \tag{11.64}$$

where L is the order of the adaptive FIR filter and σ^2 is the variance (power) of the input signal. When the input signal is not stationary and its variance is varying with time, a time varying estimate of σ^2 is used. That is

$$\hat{\sigma}_k^2 = \alpha x_k^2 + (1-\alpha)\hat{\sigma}_{k-1}^2 \tag{11.65}$$

where α is a factor selected such that $0 < \alpha < 1$. Finally, Eq. (11.63) can be written as

$$\mathbf{b}_{k+1} = \mathbf{b}_k + \frac{2\varepsilon_k \mu \mathbf{x}_k}{(L+1)\hat{\sigma}_k^2} \tag{11.66}$$

As an example and in reference to Fig. 11.15, let the input and desired signals be defined as

$$x[k] = \sqrt{2}\ \sin\left(\frac{2\pi k}{20}\right) + n[k] \quad ;\ k = 0, 1, \ldots, 500 \tag{11.67}$$

$$d[k] = \sqrt{2}\ \sin\left(\frac{2\pi k}{20}\right) \quad ;\ k = 0, 1, \ldots, 500 \tag{11.68}$$

where $n[k]$ is additive white noise with zero mean and variance $\sigma_n^2 = 2$. Figure 11.17 shows the output of the LMS algorithm defined in Eq. (11.66) when $\mu = 0.1$ and $\alpha = 0$. Figure 11.18 is similar to Fig. 11.17 except in this case, $\mu = 0.01$ and $\alpha = 0.1$. Note that in Fig. 11.18 the rate of convergences is reduced since μ is smaller than that used in Fig. 11.17; however, the filter's output is less noise because α is greater than zero which allows for more accurate updates of the noise variance as defined in Eq. (11.65). These plots can be reproduced using the following MATLAB code which utilizes the function *"LMS.m"* (see Section 11.6.2).

```
% Figures 11.17 and 11.18
close all; clear all
N = 501;
mu = 0.1; % convergence parameter
L = 20; % FIR filter order
B = zeros(1,L+1); % FIR coefficients
sigma = 2; %Initial estimate for noise power
alpha = .00; % forgetting factor
 k = 1:N;
noise = rand(1, length(k)) - .5; % Random noise
D = sqrt(2)*sin(2*pi*k/20);
X = D + sqrt(7)*noise;
Y = LMS(X, D, B, mu, sigma, alpha);
subplot(3,1,1)
plot(D,'linewidth',1); xlim([0 501]); grid on;
ylabel('Desired response'); title('\mu = 0.1; \alpha = 0.')
subplot(3,1,2)
plot(X,'linewidth',1); xlim([0 501]); grid on;
ylabel('Corrupted signal')
subplot(3,1,3)
plot(Y,'linewidth',1); xlim([0 501]); grid on;
xlabel('time in sec');
ylabel('LMS output')
```

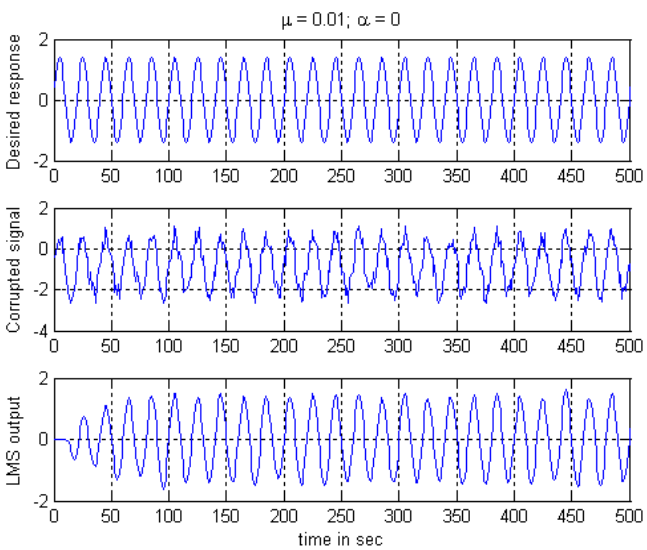

Figure 11.17. Input signal, desired response, and output response of an LMS filter.

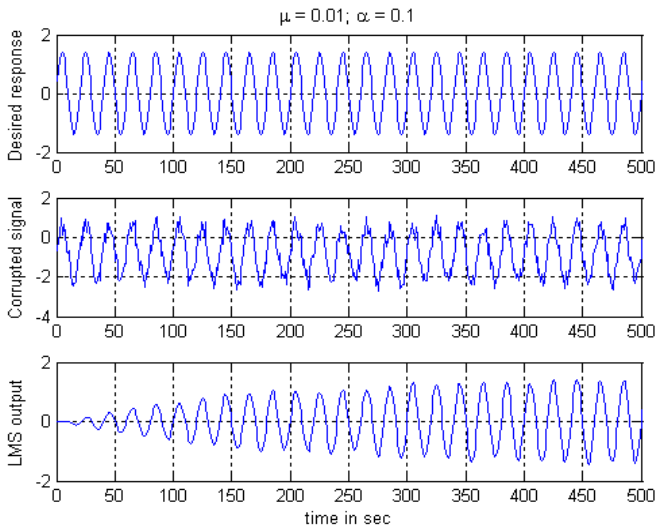

Figure 11.18. Input signal, desired response, and output response of an LMS filter.

11.5.2. The LMS Adaptive Array Processing

Consider the LMS adaptive array shown in Fig. 11.19. The difference between the reference signal and the array output constitutes an error signal. The error signal is then used to adaptively calculate the complex weights, using a predetermined convergence algorithm. The reference signal is assumed to be an accurate approximation of the desired signal (or desired array response). This reference signal can be computed using a training sequence or spreading code which is supposed to be known at the radar receiver. The format of this reference signal will vary from one application to another. But in all cases, the reference signal is assumed to be correlated with the desired signal. An increased amount of this correlation significantly enhances the accuracy and speed of the convergence algorithm being used. In this section, the LMS algorithm is assumed.

In general, the complex envelope of a bandpass signal and its corresponding analytical (pre-envelope) signal can be written using the quadrature components pair $(x_I(t), x_Q(t))$. Recall that the quadrature components are related using the Hilbert transform as follows:

$$x_Q(t) = \hat{x}_I(t) \tag{11.69}$$

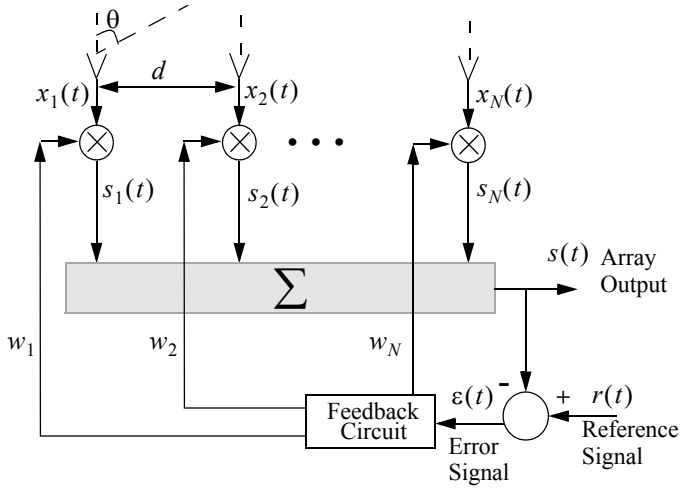

Figure 11.19. A linear adaptive array.

where $\hat{x}_I$ is the Hilbert transform of x_I. A bandpass signal $x(t)$ can be expressed as follows (see Chapter 2):

$$x(t) = x_I(t)\cos 2\pi f_0 t - x_Q(t)\sin 2\pi f_0 t \tag{11.70}$$

$$\psi(t) = x(t) + j\hat{x}(t) \equiv \tilde{x}(t)e^{j2\pi f_0 t} \tag{11.71}$$

$$\tilde{x}(t) = x_I(t) + jx_Q(t) \tag{11.72}$$

where $\psi(t)$ is the pre-envelope and $\tilde{x}(t)$ is the complex envelope. Equation (11.72) can be written using Eq. (11.69) as

$$\tilde{x}(t) = x_I(t) + jx_Q(t) = x_I(t) + j\hat{x}_I(t) \tag{11.73}$$

Using this notation, the adaptive array output signal, its reference signal, and the error signal can also be written using the same notation as

$$\tilde{s}(t) = s(t) + j\hat{s}(t) \tag{11.74}$$

$$\tilde{r}(t) = r(t) + j\hat{r}(t) \tag{11.75}$$

$$\tilde{\varepsilon}(t) = \varepsilon(t) + j\hat{\varepsilon}(t) \tag{11.76}$$

Referencing Fig. 11.19, denote the output of the n^{th} array input signal as $s_n(t)$ and assume complex weights given by

$$w_n = w_{nI} + jw'_{nQ} = w_{nI} - jw_{nQ} \tag{11.77}$$

It follows that

$$s_n(t) = w_{nI}\ x_{nI}(t) + w_{nQ}\ x_{nQ}(t) \tag{11.78}$$

Taking the Hilbert transform of Eq. (11.78) yields

$$\hat{s}_n(t) = w_{nI}\ \hat{x}_{nI}(t) + w_{nQ}\ \hat{x}_{nQ}(t) \tag{11.79}$$

By using Eq. (11.67) into Eq. (11.79), one gets

$$\hat{x}_n(t) = w_{nI}\ \hat{x}_{nQ}(t) - w_{nQ}\ \hat{x}_{nI}(t) \tag{11.80}$$

The n^{th} channel analytic signal is

$$\psi_n(t) = s_n(t) + j\hat{s}_n(t) \tag{11.81}$$

Substituting Eq. (11.78) and Eq. (11.79) into Eq. (11.80) gives

$$\psi_n(t) = w_{nI}\ x_{nI}(t) + w_{nQ}\ x_{nQ}(t) + j[w_{nI}\ \hat{x}_{nQ}(t) - w_{nQ}\ \hat{x}_{nI}(t)] \tag{11.82}$$

Collecting terms yields, using complex notation,

$$\psi_n(t) = w_n \tilde{x}_n(t) \tag{11.83}$$

Therefore, the output of the entire adaptive array is

$$\tilde{s}(t) = \sum_{n=1}^{N} \tilde{s}_n(t) = \sum_{n=1}^{N} w_n \tilde{x}_n(t) \tag{11.84}$$

which can be written using vector notation as

$$\tilde{\mathbf{s}} = \mathbf{w}^t \tilde{\mathbf{x}} = \tilde{\mathbf{x}}^t \mathbf{w} \tag{11.85}$$

where the vectors $\tilde{\mathbf{x}}$ and $\mathbf{w}$ are given by

$$\tilde{\mathbf{x}} = \left[\tilde{x}_1(t)\ \tilde{x}_2(t)\ \ldots \tilde{x}_N(t)\right]^t \tag{11.86}$$

$$\mathbf{w} = \left[w_1\ w_2\ \ldots w_N\right]^t \tag{11.87}$$

The superscript $\{\ ^t\}$ indicates the transpose operation.

As discussed earlier, one common technique to achieving the MMSE of an LMS algorithm is to use steepest descent. Thus, the complex weights in the LMS adaptive array are related as defined in Eq. (11.59). That is,

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu\nabla_k \tag{11.88}$$

where again μ is the convergence parameter. The subscript k indicates time samples. In this case, the gradient vector ∇_k is defined by

$$\nabla_k = \frac{\partial}{\partial \mathbf{w}_k} E[\tilde{\varepsilon}_k^2] = \left[\frac{\partial}{\partial \mathbf{w}_0(k)} E[\tilde{\varepsilon}_k^2] \;\ldots\; \frac{\partial}{\partial \mathbf{w}_N(k)} E[\tilde{\varepsilon}_k^2]\right]^t \tag{11.89}$$

Rearranging Eq. (11.88) so that the rate of change between consecutive estimates of the complex weights is on one side of the equation yields

$$\mathbf{w}_{k+1} - \mathbf{w}_k = -\mu \frac{\partial}{\partial \mathbf{w}_k}(E[\tilde{\varepsilon}_k^2]) \tag{11.90}$$

where the middle portion of Eq. (11.89) was also substituted for the gradient vector. In this format, the left hand side of Eq. (11.90) represents the rate of change of the complex weights with respect to time (i.e., the derivative of the weights with respect to time). It follows that

$$\frac{d}{dt}\mathbf{w} = -\mu \frac{\partial}{\partial \mathbf{w}}(E[\tilde{\varepsilon}^2(t)]) \tag{11.91}$$

However, see from Fig. 11.18, that the error signal complex envelope is

$$\tilde{\varepsilon}(t) = \tilde{r}(t) - \sum_{n=1}^{N} w_n \tilde{x}_n(t) \Rightarrow \tilde{\varepsilon}(t) = \tilde{r}(t) - \tilde{\mathbf{x}}^t \mathbf{w} \tag{11.92}$$

It can be shown (see Problem 11.6) that

$$\frac{\partial}{\partial \mathbf{w}} E[\tilde{\varepsilon}^2(t)] = -E[\tilde{\mathbf{x}}^* \tilde{\varepsilon}(t)] \tag{11.93}$$

Therefore, Eq. (11.91) can be written as

$$\frac{d}{dt}\mathbf{w} = \mu E[\tilde{\mathbf{x}}^* \tilde{\varepsilon}(t)] \tag{11.94}$$

substituting Eq. (11.92) into Eq. (11.94) gives

$$\frac{d}{dt}\mathbf{w} = \mu E[\tilde{\mathbf{x}}^* (\tilde{r}(t) - \tilde{\mathbf{x}}^t \mathbf{w})] \tag{11.95}$$

Equivalently,

$$\frac{d}{dt}\mathbf{w} + \mu E[\tilde{\mathbf{x}}^*\tilde{\mathbf{x}}^t]\mathbf{w} = \mu E[\tilde{\mathbf{x}}^*\tilde{r}(t)] \tag{11.96}$$

The covariance matrix is by definition

$$\boldsymbol{C} = E[\tilde{\mathbf{x}}^*\tilde{\mathbf{x}}^t] = \begin{bmatrix} \tilde{x}_1^*\tilde{x}_1 & \tilde{x}_1^*\tilde{x}_2 & \cdots \\ \tilde{x}_2^*\tilde{x}_1 & \tilde{x}_2^*\tilde{x}_2 & \cdots \\ \cdots & \cdots & \end{bmatrix} \tag{11.97}$$

and the reference signal correlation vector $\mathbf{s}$ is

$$\mathbf{s} = E[\tilde{\mathbf{x}}^*\tilde{d}(t)] = E\left[\tilde{x}_1^*\tilde{r} \;\; \tilde{x}_2^*\tilde{r} \;\; \cdots\right]^t \tag{11.98}$$

Using Eq. (11.98) and Eq. (11.97), one can rewrite the differential equation (DE) given Eq. (11.96) as

$$\frac{d}{dt}\mathbf{w} + \mu\boldsymbol{C}\mathbf{w} = \mu\mathbf{s} \tag{11.99}$$

The steady state solution for the DE defined in Eq. (11.99) (provided that the covariance matrix is not singular) is

$$\mathbf{w} = \boldsymbol{C}^{-1}\,\mathbf{S} \tag{11.100}$$

As the size of the covariance matrix increase (i.e., number of channels in the adaptive array) so does the complexity associated with computing the adaptive weights in real time. This is true because computing the inverse of large matrices in real time can be extremely challenging and demands significant amount of computing power. Consequently, the effectiveness of adaptive arrays has been limited to small-sized arrays, where only a few interfering signals can be eliminated (cancelled). Additionally, computing of a good estimate of the covariance matrix in real time is also difficult in practical applications. In order to mitigate that effect, a reasonable estimate for $E\{x_i x_j^*\}$ (the i,j element of the covariance matrix) is derived by averaging m independent samples of data from the same distribution. This approach can be extended to the entire covariance matrix by collecting M independent "snapshots" of data from N channels. Thus, the estimate of the covariance matrix can be given as,

$$\tilde{\boldsymbol{C}} \approx (\tilde{\mathbf{x}}^\dagger\tilde{\mathbf{x}})/M \tag{11.101}$$

The transient solution of Eq. (11.99) (see Problem 11.7) is

$$\mathbf{w}(t) = \sum_{n=1}^{N} \mathbf{p}_i e^{-\mu\lambda_i t} \tag{11.102}$$

where the vectors $\mathbf{p}_i$ are constants that depend on the initial value of $\mathbf{w}(t)$, and λ_i are the eigenvalues of the matrix $\boldsymbol{C}$. It follows that the complete solution of Eq. (11.99) is

$$\mathbf{w}(t) = \sum_{n=1}^{N} \mathbf{p}_i e^{-\mu\lambda_i t} + \boldsymbol{C}^{-1}\ \mathbf{s} \tag{11.103}$$

A very common measure of effectiveness of an adaptive array is the ratio of the total output interference power, S_o to the internal noise power, S_n.

Example:

Consider the two-element array in Section 11.4. Assume the desired signal is at directional-sine $sin(\theta_t)$ and the interference signal is at $sin(\theta_i)$. Calculate the adaptive weights so that the interference signal is cancelled.

Solution:

From Fig. 11.19

$$\tilde{x}_1(t) = \tilde{d}_1(t) + \tilde{n}_1(t) + \tilde{I}_1(t)$$

$$\tilde{x}_2(t) = \tilde{d}_2(t) + \tilde{n}_2(t) + \tilde{I}_2(t)$$

where d is the desired response, n is the noise, signal, and I is the interference signal. The noise signal is spatially incoherent, more specifically

$$E[\tilde{n}_i^*(t)\tilde{n}_j(t)] = \begin{cases} 0 & i \neq j \\ \sigma_n^2 & i = j \end{cases}$$

Also

$$E[\tilde{d}_i^*(t)\tilde{n}_i(t)] = 0 \qquad for\ all\ (i,j)$$

The desired signal is

$$\tilde{d}(t) = \tilde{d}_1(t) + \tilde{d}_2(t) = A_d e^{j2\pi f_0 t} e^{j\Theta_d} + A_d e^{j2\pi f_0 t} e^{j\Theta_d} e^{-j\pi \sin\theta_d}$$

where Θ_d is a uniform random variable. The interference signal is

$$\tilde{I}(t) = \tilde{I}_1(t) + \tilde{I}_2(t) = A_i e^{j2\pi f_0 t} e^{j\Theta_i} + A_i e^{j2\pi f_0 t} e^{j\Theta_i} e^{-j\pi \sin\theta_i}$$

where Θ_i is a uniform random variable. Of course the random variables Θ_d and Θ_i are assumed to be statistically independent. In vector format

$$\tilde{\mathbf{x}}_d = A_d e^{j2\pi f_0 t} e^{j\Theta_d} \begin{bmatrix} 1 \\ e^{-j\pi \sin\theta_d} \end{bmatrix}$$

$$\tilde{\mathbf{x}}_i = A_i e^{j2\pi f_0 t} e^{j\Theta_i} \begin{bmatrix} 1 \\ e^{-j\pi \sin\theta_i} \end{bmatrix}$$

Of course the noise vector is

$$\tilde{\mathbf{x}}_n = \begin{bmatrix} \tilde{n}_1(t) \\ \tilde{n}_2(t) \end{bmatrix}$$

and the reference signal is (this is an assumption so that the desired and reference signal are correlated)

$$\tilde{r}(t) = A_r e^{j2\pi f_0 t} e^{j\Theta_d}$$

Note that the input SNR is

$$SNR_d = A_d^2 / \sigma_n^2$$

and the interference to noise ratio is

$$SNR_i = A_i^2 / \sigma_n^2$$

The input signal can be written using vector notation as

$$\tilde{\mathbf{x}} = \tilde{\mathbf{x}}_d + \tilde{\mathbf{x}}_i + \tilde{\mathbf{x}}_n$$

The covariance matrix is computed from Eq. (11.97) as

$$\boldsymbol{C} = E[\tilde{\mathbf{x}}_d^* \tilde{\mathbf{x}}_d^t] = \begin{bmatrix} A_d^2 + A_i^2 + \sigma_n^2 & A_d^2 e^{-j\pi \sin\theta_d} + A_i^2 e^{-j\pi \sin\theta_i} \\ A_d^2 e^{j\pi \sin\theta_d} + A_i^2 e^{j\pi \sin\theta_i} & A_d^2 + A_i^2 + \sigma_n^2 \end{bmatrix}$$

In order to compute the covariance matrix eigenvalue, one needs to compute the determinant first

$$|\mathbf{C}| = 4A_d^2A_i^2\left(sin\left(\frac{\theta_d+\theta_i}{s}\right)\right)^2 + 2A_d^2\sigma_n^2 + 2A_i^2\sigma_n^2 + \sigma_n^4$$

Thus,

$$\mathbf{C}^{-1} = \frac{1}{|\mathbf{C}|}\begin{bmatrix} A_d^2 + A_i^2 + \sigma_n^2 & -A_d^2 e^{-j\pi sin\theta_d} - A_i^2 e^{-j\pi sin\theta_i} \\ -A_d^2 e^{j\pi sin\theta_d} - A_i^2 e^{j\pi sin\theta_i} & A_d^2 + A_i^2 + \sigma_n^2 \end{bmatrix}$$

The reference correlation vector is

$$\mathbf{s} = E[\tilde{\mathbf{x}}^*\tilde{r}(t)] = A_dA_r\begin{bmatrix} 1 \\ e^{j\pi sin\theta_d} \end{bmatrix}$$

It follows that the weights are

$$\mathbf{w} = \frac{A_dA_r}{|\mathbf{C}|}\begin{bmatrix} A_i^2 + \sigma_n^2 - A_i^2 e^{j\pi(sin\theta_d - sin\theta_i)} \\ e^{j\pi sin\theta_d}\{A_i^2 + \sigma_n^2 - A_i^2 e^{j\pi(sin\theta_i - sin\theta_d)}\} \end{bmatrix}$$

11.5.3. Sidelobe Cancelers (SLC)

Sidelobe cancelers typically consist of a main antenna (which can be a phased array or a single element) and one or more auxiliary antennas. The main antenna is referred to as the main channel; it is assumed to be highly directional and is pointed toward the desired signal angular location. The interfering signal is assumed to be located somewhere off the main antenna boresight (in the sidelobes). Because of this configuration the main channel receives returns from both the desired and the interfering signals. However, returns from the interfering signal in the main channel are weak because of the low main antenna sidelobe gain in the direction of the interfering signal. Also the auxiliary antenna returns are primarily from the interfering signal. This is illustrated in Fig. 11.20.

Referring to Fig. 11.20, $\tilde{s}(t)$ is the desired signal, $\tilde{n}(t)$ is the main channel noise signal which is primarily from the interfering signal, while $\tilde{n}'(t)$ is the interfering signal in the auxiliary array. It is assumed that the signals $\tilde{s}(t)$ and $\tilde{n}(t)$ are uncorrelated. It is also assumed that the interfering signal is highly correlated with the noise signal in the main channel. The basic idea behind SLC is to have the adaptive auxiliary channel produce an accurate estimate of the noise signal first, then to subtract that estimate from the main channel signal so that the output signal is mainly the desired signal.

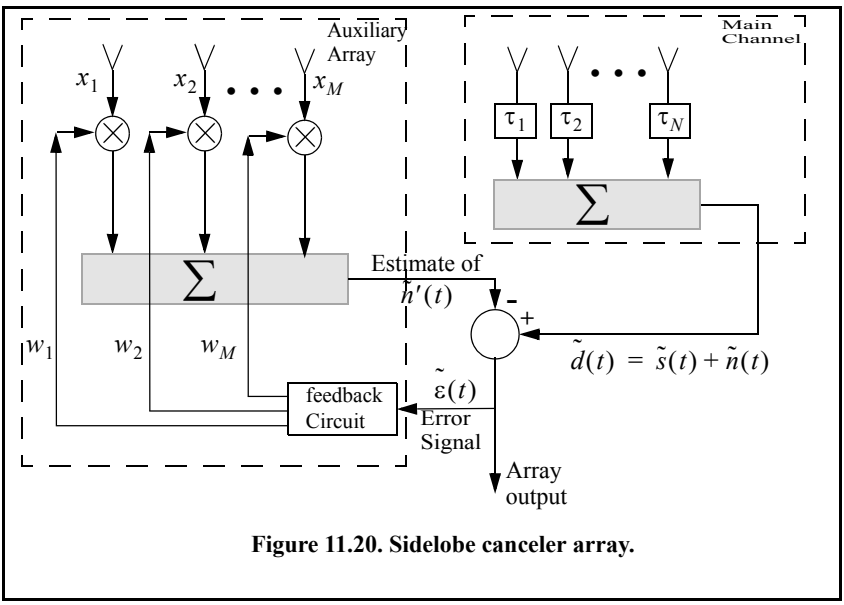

Figure 11.20. Sidelobe canceler array.

The error signal is

$$\tilde{\varepsilon} = \tilde{\mathbf{d}} - \mathbf{w}^t \tilde{\mathbf{x}} \tag{11.104}$$

where $\tilde{\mathbf{x}}$ is the vector of auxiliary array signal, $\mathbf{w}$ is the adapted weights. The vector $\mathbf{d}$ of size M. The residual power is

$$P_{res} = E[\tilde{\varepsilon}\tilde{\varepsilon}^{\dagger}] \tag{11.105}$$

$$P_{res} = E[(\tilde{\mathbf{d}} - \mathbf{w}^t \tilde{\mathbf{x}})(\tilde{\mathbf{d}}^* - \tilde{\mathbf{x}}^{\dagger}\mathbf{w}^*)] \tag{11.106}$$

It follows that

$$P_{res} = E[|\tilde{\mathbf{d}}|^2] - E[\tilde{\mathbf{d}}\tilde{\mathbf{x}}^{\dagger}\mathbf{w}^*] - E[\tilde{\mathbf{d}}^*\mathbf{w}^t\tilde{\mathbf{x}}] - \mathbf{w}^t E[\tilde{\mathbf{x}}\tilde{\mathbf{x}}^{\dagger}]\mathbf{w}^* \tag{11.107}$$

Differentiate the residual power with respect to $\mathbf{w}$ and setting the answer equal to zero (to compute the optimal weights that minimize the power residual) yields

$$\frac{\partial P_{res}}{\partial \mathbf{w}} = \mathbf{0} = -\tilde{\mathbf{x}}\tilde{\mathbf{d}} + \boldsymbol{C}_a \mathbf{w} \tag{11.108}$$

where $\boldsymbol{C}_a$ is the covariance matrix of the auxiliary channel. Finally, the optimal weights are given by

$$\mathbf{w} = \mathbf{C}_a^{-1}\ \tilde{\mathbf{x}}\tilde{\mathbf{d}} \tag{11.109}$$

Note that the vector $\tilde{\mathbf{x}}\tilde{\mathbf{d}}$ represents the components that are common to both main and auxiliary channels. Note that Eq. (11.109) makes intuitive sense where the objective is to isolate the components in the data which are common to the main and auxiliary channels and we then wish to give them some heavy attenuation (which comes from inverting $\mathbf{C}_a$).

11.6. MATLAB Program Listings

This section presents listings for all the MATLAB programs used in this chapter. They are listed in the same order they appear in the text.

11.6.1. MATLAB Function "linear_array.m"

The function *"linear_array.m"* computes and plots the linear array gain pattern as a function of real sine-space. The syntax is as follows:

[theta, patternr, patterng] = linear_array(Nr, dolr, theta0, winid, win, nbits)

where

Symbol	Description	Units	Status
Nr	*number of elements in array*	*none*	*input*
dolr	*element spacing in lambda units*	*wavelengths*	*input*
theta0	*steering angle*	*degrees*	*input*
winid	*-1: No weighting is used* *1: Use weighting defined in win*	*none*	*input*
win	*window for side-lobe control*	*none*	*input*
nbits	*negative #: perfect quantization* *positive #: use* 2^{nbits} *quantization levels*	*none*	*input*
theta	*real angle available for steering*	*degrees*	*output*
patternr	*array pattern*	*dB*	*output*
patterng	*gain pattern*	*dB*	*output*

MATLAB Function** "linear_array.m" **Listing

```
function [theta,patternr,patterng] = linear_array(Nr,dolr,theta0,winid,win,nbits);
% This function computes and returns the gain radiation pattern for a linear array
% It uses the FFT to computes the pattern
%%%% *INPUTS ********** %%%%%%%%%%%%%%
```

```
% Nr ==> number of elements; dolr ==> element spacing (d) in lambda units divided by lambda
% theta0 ==> steering angle in degrees; winid ==> use winid negative for no window, winid positive to enter your window of size(Nr)
% win is input window, NOTE that win must be an NrX1 row vector; nbits ==> number of bits used in the pahse shifters
% negative nbits mean no quantization is used
%%%% *OUTPUTS ********** %%%%%%%%%%%%%%%%
% theta ==> real-space angle; patternr ==> array radiation pattern in dBs
% patterng ==> array directive gain pattern in dBs
%%%%%%%%% ******************** %%%%%%%%%%%%
eps = 0.00001;
n = 0:Nr-1;
i = sqrt(-1);
%if dolr is > 0.5 then; choose dol = 0.25 and compute new N
if(dolr <=0.5)
  dol = dolr;
  N = Nr;
else
  ratio = ceil(dolr/.25);
  N = Nr * ratio;
  dol = 0.25;
end
% choose proper size fft, for minimum value choose 256
Nrx = 10 * N;
nfft = 2^(ceil(log(Nrx)/log(2)));
if nfft < 256
   nfft = 256;
end
% convert steering angle into radians; and compute the sine of angle
theta0 = theta0 *pi /180.;
sintheta0 = sin(theta0);
% detrmine and comput quantized steering angle
if nbits < 0
  phase0 = exp(i*2.0*pi .* n * dolr * sintheta0);
else
   % compute and add the phase shift terms (WITH nbits quantization)
   % Use formula thetal = (2*pi*n*dol) * sin(theta0) divided into 2^nbits
   % and rounded to the nearest quantization level
   levels = 2^nbits;
   qlevels = 2.0 * pi / levels; % compute quantization levels
% compute the phase level and round it to the closest quantization level
   angleq = round(dolr .* n * sintheta0 * levels) .* qlevels; % vector of possible angles
   phase0 = exp(i*angleq);
end
% generate array of elements with or without window
if winid < 0
   wr(1:Nr) = 1;
```

```
else
   wr = win';
end
% add the phase shift terms
 wr =  wr .* phase0;
 % determine if interpolation is needed (i.e N > Nr)
if N > Nr
   w(1:N) = 0;
   w(1:ratio:N) = wr(1:Nr);
else
   w = wr;
end
% compute the sine(theta) in real space that correspond to the FFT index
arg = [-nfft/2:(nfft/2)-1] ./ (nfft*dol);
idx = find(abs(arg) <= 1);
sinetheta = arg(idx);
theta = asin(sinetheta);
% convert angle into degrees
theta = theta .* (180.0 / pi);
% Compute fft of w (radiation pattern)
patternv = (abs(fftshift(fft(w,nfft)))).^2;
% convert radiation pattern to dBs
patternr = 10*log10(patternv(idx) ./Nr +  eps);
% Compute directive gain pattern
rbarr  = 0.5 *sum(patternv(idx)) ./ (nfft * dol);
patterng = 10*log10(patternv(idx) + eps) - 10*log10(rbarr + eps);
return
```

11.6.2. MATLAB Function "LMS.m"

The function *"LMS.m"* implements Eq. (11.66). Its syntax is as follows

$$Y = LMS(X, D, B, mu, sigma, alpha)$$

where X is the corrupted sequence, D is the desired response, B is a vector containing the FIR filter coefficients (its initial value can be set to zero), *mu* is the convergence parameter, *sigma* is the SNR, and *alpha* is the forgetting factor.

MATLAB Function *"LMS.m"* Listing

```
function X = LMS(X, D, B, mu, sigma, alpha)
%   This program was written by Stephen Robinson a senior radar
%   engineer at deciBel Research, Inc. in Huntsville, AL
%   X = data vector ; size = 1 x N
%   D = desired signal vector; size = 1 x N
%   N = number of data samples and of adaptive iterations
%   B = adaptive coefficients of Lht order fFIRfilter; size = 1 x L
%   L = order of adaptive system
%   mu = convergence parameter
```

```
% sigma = input signal power estimate
% alpha = exponential forgetting factor
N = size(X,2)
L = size(B,2)-1
px = B;
for k = 1:N
   px(1) = X(k);
   X(k) = sum(B.*px);
   E = D(k) - X(k);
   sigma = alpha*(px(1)^2) + (1 - alpha)*sigma;
   tmp = 2*mu/((L+1)*sigma);
   B = B + tmp*E*px;
   px(L+1:-1:2) = px(L:-1:1);
end
return
```

Problems

11.1. Consider an antenna whose diameter is $d = 3m$. What is the far field requirement for an X-band or an L-band radar that is using this antenna?

11.2. Consider an antenna with electric field intensity in the *xy*-plane $E(\varsigma)$. This electric field is generated by a current distribution $D(y)$ in the *yz*-plane. The electric field intensity is computed using the integral

$$E(\varsigma) = \int_{-r/2}^{r/2} D(y) exp\left(2\pi j \frac{y}{\lambda} sin\varsigma\right) dy$$

where λ is the wavelength and r is the aperture. (a) Write an expression for $E(\varsigma)$ when $D(Y) = d_0$ (a constant). (b) Write an expression for the normalized power radiation pattern and plot it in dB.

11.3. A linear phased array consists of 50 elements with $\lambda/2$ element spacing. (a) Compute the 3dB beam width when the main-beam steering angle is $0°$ and $45°$. (b) Compute the electronic phase difference for any two consecutive elements for steering angle $60°$.

11.4. A linear phased array antenna consists of eight elements spaced with $d = \lambda$ element spacing. (a) Give an expression for the antenna gain pattern (assume no steering and uniform aperture weighting). (b) Sketch the gain pattern versus sine of the off-boresight angle β. What problems do you see is using $d = \lambda$ rather than $d = \lambda/2$?

11.5. In Section 10.4.2 we showed how a DFT can be used to compute the radiation pattern of a linear phased array. Consider a linear of 64 elements at half wavelength spacing, where an FFT of size 512 is used to compute the pattern. What are the FFT bins that correspond to steering angles $\beta = 30°, 45°$?

11.6. Derive Eq. (11.93).

11.7. Compute the transient solution of the DE defined in Eq. (11.99).

11.8. Compute the interference power to the intput power ratio of the example in Section 11.5.3.

11.9. To generate the sum and difference patterns for a linear array of size N follow this algorithm: To form the difference pattern, multiply the first $N/2$ elements by -1 and the second $N/2$ elements by +1. Plot the sum and difference patterns for a linear array of size 60.

11.10. Generate the delta/sum patterns for a 21-element linear array using the form $\frac{\Delta}{\Sigma} = j\frac{V_\Delta}{\sqrt{|V_\Delta|^2 + |V_\Sigma|^2}}$ where V_Δ is the difference voltage pattern and V_Σ is the sum voltage pattern.