METU EE604 SENSOR ARRAY SIGNAL PROCESSING - LECTURE NOTES

ALPER YAZAR

These lecture notes are based on notes taken during EE604 Sensor Array Signal Processing course given at METU, Spring 2013 by Prof. T. Engin TUNCER. Also, I used some lecture notes taken at previous years. There are some missing figures. Initially, I started to prepare figures with computer softwares. However, since it took so long time, I switched to hand drawing figures, sorry for that :)

Most probably, that document has lots of mistakes. PLEASE ALWAYS BE SUSPICIOUS ABOUT CORRECTNESS.

All corrections and valuable comments are appreciated. Please feel free to contact me about corrections, comments, etc.

This document is served at following URL:

http://www.alperyazar.com/downloads/ee604_sensor_array_signal_processing_alper_yazar.pdf

If link doesn't work, please try the following permalink:

http://www.alperyazar.com/yonlendir/21/6730

Version:1, 11/08/2014

Contact: alperyazar@gmail.com

Contents

Date: Spring 2013.

Part 1. **18/02/14 Lecture Note**

1. Introduction

Sensor is a device that measures a physical quantity and converts it into a signal.

Physical aspects that sensors measures are light, motion, temperature, electromagnetic field, vibration, sound, gravity, etc.

Major sensor types are:

- Electromagnetic Sensor: Antenna
- Acoustic Sensor: Microphone
- Ultrasound Sensor: Sonobouy [1]
- Light Sensor: Camera, Photodiode

Sensor Array is a set of sensors placed in a certain array geometry to gather information in such a way that a single sensor con not.

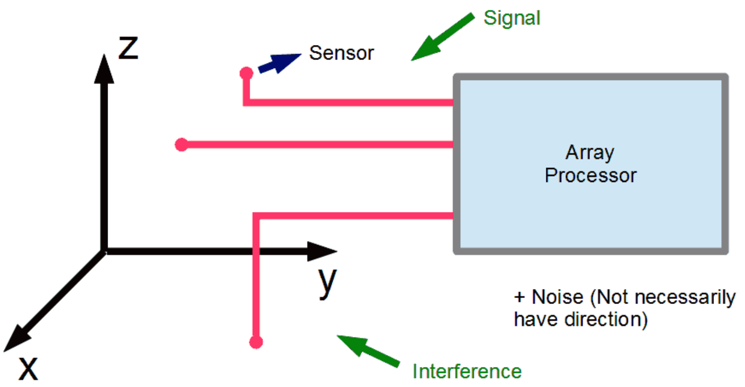

Figure 1.1. A Typical Array with a Single Signal and Interference Source

As a footnote, distances greater than 10 is a good assumption for far field.

Array Aperture is the space occupied by the array.

Generally speaking, as aperture size increases resolution increases. However, as intersensor (?) distance, d, exceeds =2 there is a spatial aliasing.

2. Array and Signal Categories

Arrays and array problems can be classified according to different parameters.

(1) Array Geometry
  (a) Linear (1D) Array
  (b) Planar (2D) Array
  (c) Volumetric (3D) Array
  When array geometry is fixed there is a Cramer-Rao Bound related with geometry.
(2) Sensor Placement
  (a) Uniform Spacing: Most robust placement

[1] http://en.wikipedia.org/wiki/Sonobuoy

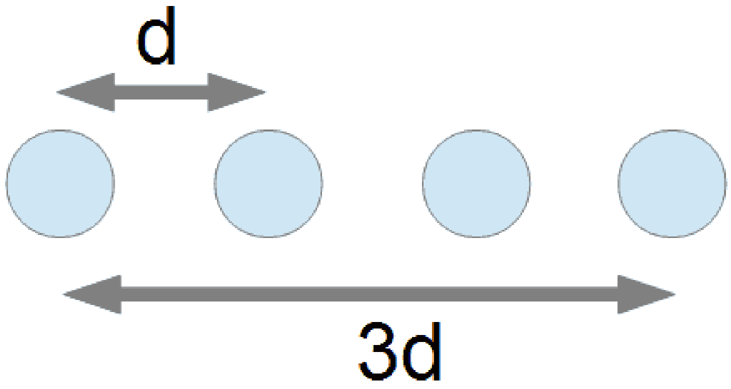

Figure 1.2. Sensors Separated By d, Array Aperture is 3d

   (b) Non-Uniform Spacing: Good to increase aperture size but spatial aliasing problem may occur.
   (c) Random Spacing: Throw very cheap array from plane for example.
(3) Temporal Characteristics (Related with Time) of Signal
   (a) Known Signal: Ex: Emergency Beacon Signal
   (b) Signals with Unknown Parameters
   (c) Signals with Known Structure: Ex: QPSK
   (d) Random Signal: Most of time it is.
(4) Spatial Characteristics (Related with Space) of Signal
   (a) Plane Wave Signals From Known Directions
   (b) Plane Wave Signals From Unknown Directions
   (c) Spatially Spread Signals

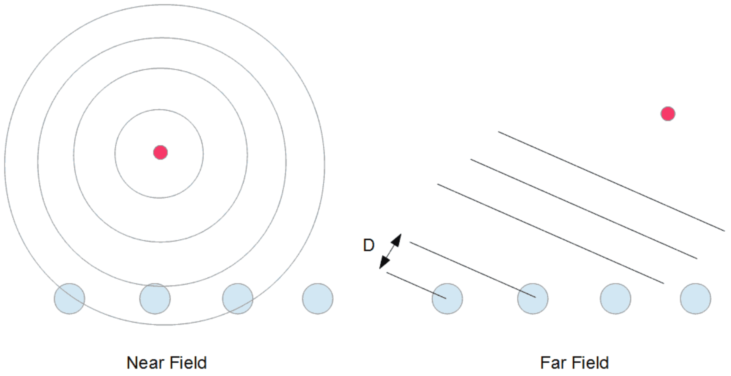

Figure 2.1. Near and Far Field Sources

As shown in Figure 2.1 sources can be classified into two main categories: Near and Far field sources. In near field case, time delay isn't a linear function. But from observed data it is possible to find both distance and angle information. There are devices that use that technique. In far field case, we assume that source is far away enough to consider waves as plane waves. There is an additional distance between two sensors D. And time delay is simply = D=c. However, it is only possible to find DOA in that case. If source distance is greater than 10 it can be assumed as far field source.

Similarly for interference we have the same spatial and temporal characteristics.

3. Objectives of Array Processing

(1) Passive Arrays
   (a) Detect
   (b) Estimate
   (c) Localize
   (d) Track
(2) Active Arrays (Transmit a Signal First)
   (a) Detect
   (b) Estimate
   (c) Localize
   (d) Track

- Detect the presence of a signal in the presence of noise and interference.
- Demodulate the signal and estimate the information from waveform in the presence of noise and interference.
- A binary communication signal arrives over multipath, detect the information sequence.
- Estimate the direction of arrival (DOA) of multiple plane wave signals in the presence of noise.
- Construct the temporal and spatial spectrum estimate of the incoming signal and noise field. (Spatial t-F Spectrum (?))
- Direct the transmitted signal to a specific spatial location.
- Find the location of the source signal (Localization)

Some applications:

- Radar
- Sonar
- Communications
- Acoustics
- Radio Astronomy
- Medical Diagnosis and Treatment

Array geometry establishes constraints on the array performance. Therefore it should be selected appropriately.

By designing complex weights for the sensor outputs, one can filter the signal such that signal coming from a particular angle is enhanced. (Beamforming)

4. Array Model

and is defined as *Elevation Angle* and *Azimuth Angle* respectively as shown in Figure

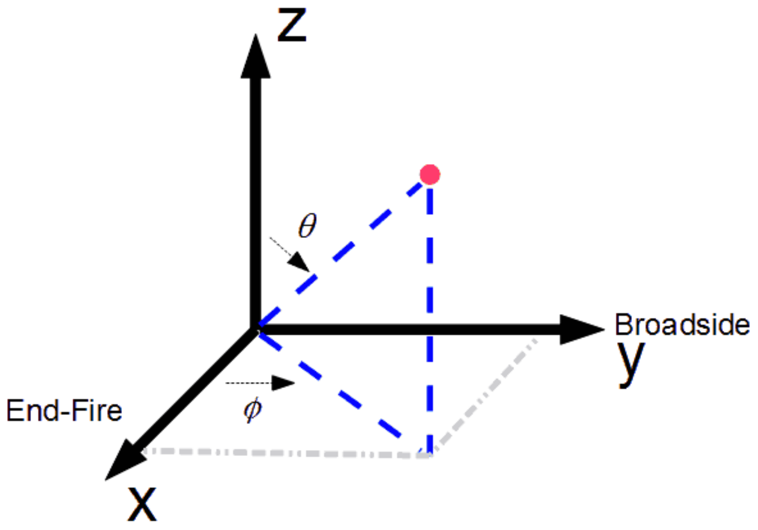

Figure 4.1. Spherical Coordinate System

(4.2)
$$\begin{aligned} x &= \ r\cos\ \ \sin \\ y &= \ r\sin\ \ \sin \\ z &= \ \ \ r\cos \end{aligned}$$

The direction of plane wave propagation in unit direction g is given as

(4.3)
$$g = \begin{bmatrix} \cos\ \ \sin \\ \sin\ \ \sin \\ \cos \end{bmatrix}$$

The signal arrives to each sensor with a time delay ${}_n$,

(4.4)
$${}_n = \frac{-g^T P_n}{c}$$

where c is the velocity of the propagation, P_n is the position vector for n^{th} sensor.

(4.5)
$$P_n \triangleq \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix}$$

Combining (4.4) and (4.5).

(4.6)
$${}_n = -\frac{1}{c}\left[P_{xn}\cos\ \ \sin\ \ + P_{yn}\sin\ \ \cos\ \ + P_{zn}\cos\ \ \right]$$

Let the frequency of the narrow band signal be $! = 2\ \ f$ rad=s. (Why narrowband ???) In that case,

$$! \;\; _n = -\frac{!}{c} g^T P_n = -2 \;\; \frac{f}{c} g^T P_n = -\frac{2}{\;} g^T P_n \tag{4.7}$$

where $\;\; = c{=}f$ m.

Also *wave number*, k, is defined as

$$k \triangleq \frac{2}{\;} g = \frac{2}{\;} \begin{bmatrix} cos \;\; sin \\ sin \;\; sin \\ cos \end{bmatrix} \tag{4.8}$$

5. Array Steering Vector (Array Manifold)

$$a(! ; \;\; ; \;\;) = \begin{bmatrix} e^{-j\omega\tau_0} \\ e^{-j\omega\tau_1} \\ e^{-j\omega\tau_2} \\ \vdots \\ e^{-j\omega\tau_{M-1}} \end{bmatrix}_{Mx1} \tag{5.1}$$

Combining (4.7), (4.8) and (5.1),

$$a(! ; \;\; ; \;\;) = \begin{bmatrix} e^{jk^T P_0} \\ e^{jk^T P_1} \\ e^{jk^T P_2} \\ \vdots \\ e^{jk^T P_{M-1}} \end{bmatrix} \tag{5.2}$$

Similarly, combining (4.5), (4.8) and (5.2)

$$a(! ; \;\; ; \;\;) = \begin{bmatrix} e^{j\frac{2\pi}{\lambda}[P_{x0}cos\phi sin\theta + P_{y0}sin\phi sin\theta + P_{z0}cos\theta]} \\ e^{j\frac{2\pi}{\lambda}[P_{x1}cos\phi sin\theta + P_{y1}sin\phi sin\theta + P_{z1}cos\theta]} \\ e^{j\frac{2\pi}{\lambda}[P_{x2}cos\phi sin\theta + P_{y2}sin\phi sin\theta + P_{z2}cos\theta]} \\ \vdots \\ e^{j\frac{2\pi}{\lambda}[P_{xM-1}cos\phi sin\theta + P_{yM-1}sin\phi sin\theta + P_{zM-1}cos\theta]} \end{bmatrix} \tag{5.3}$$

Note that array steering vector depends on frequency (equally $\;\;$). We will drop $!$ but it is always there, not forget!

Only DOA information in a vector, but location information isn't available (r is missing.). It seems that with a single array you can't find location. Well, how radar find? It is a single sensor. This is for passive scenario, but radar is an active device. Find time between transmission and reception. Radar is transmitting directional pulse.

Array steering vector incorporates all the spatial characteristics of array.

When there are multiple plane wave sources at directions ($_1$, $_1$) and ($_2$, $_2$) *Array Steering Matrix* $A(\;\; ; \;\;)$ can be constructed from steering vectors. [2]

[2] As mentioned, we drop $!$ here but it is there.

$$A(\; ; \;) = \left[a(\; _1; \; _1) \quad a(\; _2; \; _2) \right] \tag{5.4}$$

Notice that we are talking about *co-channel signals*. They occupy same band but they are at different locations.

6. Narrow Band Assumption

$$B \times T_{MAX} << 1 \tag{6.1}$$

where B is the bandwidth of the signal in Hz. And T_{MAX} is the maximum time to travel across the array. [3]

Given the above assumption array output in time can be written as

$$y(t)_{Mx1} = A(\; ; \;)_{Mxn} s(t)_{nx1} + e(t)_{Mx1} \tag{6.2}$$

where $e(t)$ is noise, $s(t)$ is source signal and $A(\; ; \;)$ is array steering matrix. N is the number of snapshots (observation). n is the number of sources.

Also $t = kT$ and $k = 1;2:::N$. T should be selected using Nyquist rate.

We know that t is used for continuous time signals. But we also use (6.2) in discrete time. If we place kT, it is discrete time. But we are going to keep t as it is but we think in discrete time. Abuse of notation! We take $t = 1;2:::N$. Because it is derived when everything is in analog. Wehen digital domain is used, t was thought as discrete time, who cares? In that case, dimension of matrices at (6.2) will change related with N. (Not sure?) Given dimensions are for analog case.

7. Questions

- For which equations narrow band signal is assumed?

8. Check

- Matrix sizes in (6.2). Especially N or n.

[3] Proof is later.

Part 2. 25/02/14 Lecture Note

9. Array Factor

Array factor is a function of antenna (sensor) positions and weights used for each antenna signal. In array factor definition we assume that each sensor has same beam shape and doesn't affect array factor. It doesn't depend on individual sensor characteristics. Elements are omni-directional.

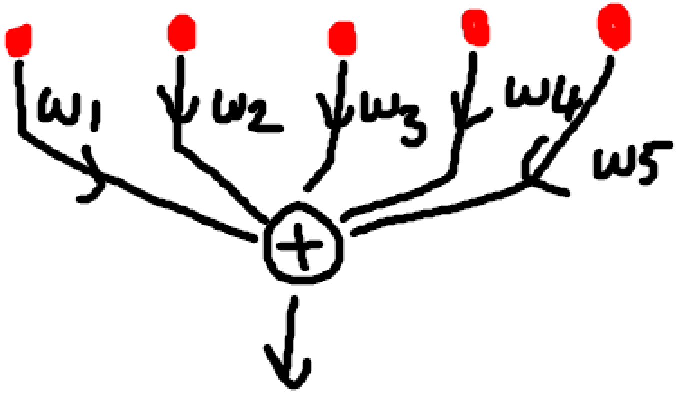

Figure 9.1. Sensor Weighting

$$B_{AF}(!;\ \ ;\ \ ) \triangleq w^H a(!;\ \ ;\ \ ) = \sum_{j=0}^{M-1} w_j^* e^{jk^T P_j} \tag{9.1}$$

where w is weight vector. Elements are complex numbers. Because the signals have assumed to be demodulated and they are in baseband. To have directional characteristics you should play with phase. If you use phase and amplitude you can control beam better. Only amplitude isn't sufficient.

$$w \triangleq \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ ::: \\ w_{M-1} \end{bmatrix}_{Mx1} \tag{9.2}$$

For a uniform linear array (ULA) with uniform weighting. In other words $w_j = 1{=}M$ where $j = 0; 1; ::: M - 1$. To look at broadside direction (forward direction) just sum up all sensor without no phase shift.

$$B_{AF}(\ \ ) = \frac{1}{M} \frac{sin\left(M \frac{d}{2}\frac{2}{\ \ }cos\ \ \right)}{sin\left(\frac{d}{2}\frac{2}{\ \ }cos\ \ \right)} \qquad 0 \leq \ \ \leq \tag{9.3}$$

Notice that in (9.3), $= 0$ for broadside direction. It is taken from Van Trees book. But, we take $= 90$ for broadside direction. One may change cos with sin our convention.

Also notice that you can't differentiate elevation angle. No in (9.3).

Plot of (9.3) is given in Figure 9.2. Code is given in the Appendix section.

Bw_{HP} is the *power beam width.* It is found using $3dB$ bandwidth. That is where $B_{AF}(\ )$ drops to $1{=}\sqrt{2}$.

Similarly, Bw_{NN} is the *null to null beam width.* Also, $Bw_{NN} \simeq 2Bw_{HP}$.

Nulls are at $=Md; 2 =Md; :::$. They are symmetric around y-axis.

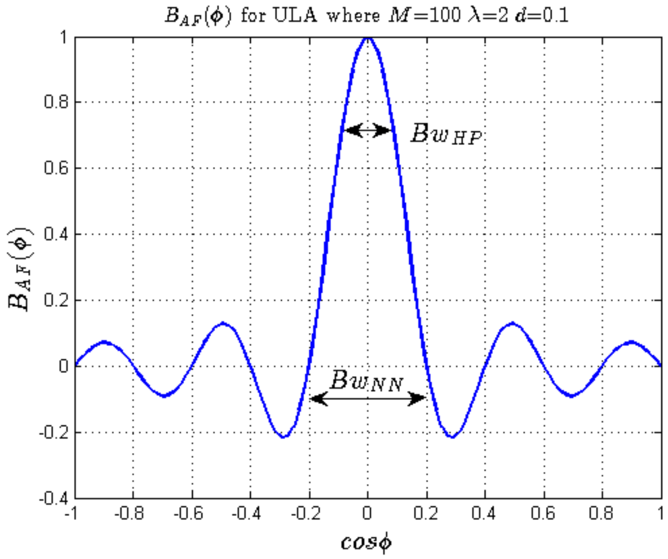

Figure 9.2. Array Factor for ULA with Uniform Weighting

10. Array Factor Parameters

(1) 3dB beamwidth (Bw_{HP})
(2) Null to null beamwidth (Bw_{NN})
(3) Distance to first sidelobe
(4) Height of the first sidelobe (Sidelobe level)
(5) Location of remaining nulls
(6) Rate of decrease of sidelobe
(7) Grating lobes. Grating lobes are very close to mainlobe in terms of amplitude. One reason is spatial aliasing. Violation of $=2$ rule.

Polar plot of Figure 9.2 is given in Figure 10.1. Code is given in the Appendix section.

As a note, narrowest beamwidth is obtained when $w_j = 1{=}M$ but in that case sidelobe level is largest ($\sim 13dB$). Decreasing sidelobe level increases beamwidth generally, filter design story...

Also notice that narrowest beamwidth is obtained at broadside direction (except for circular arrays maybe). If you turn your ULA to another direction electronically, beamwidth will increase.

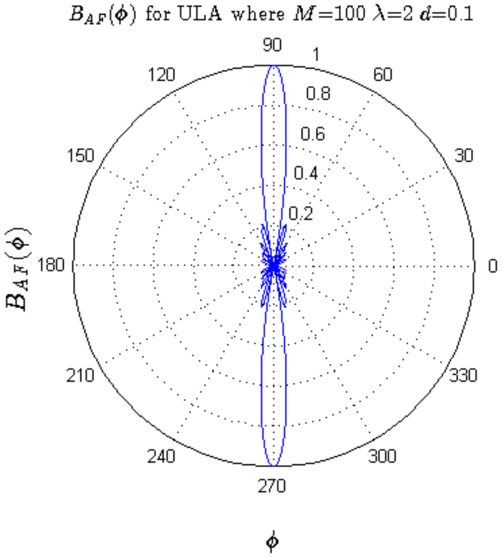

Figure 10.1. Polar Array Factor for ULA with Uniform Weighting

Bw_{NN}=2 provides a measure of the ability of the array to resolve two plane waves signals. This is referred to *Rayleigh Resoluiton* limit. Two plane wave signals are considered resolvable if the peak of the second beam pattern lies at/or outside the null of the first beam patters. (Separation $>$ Bw_{NN}=2).

Methods which can resolve sources closer that Bw_{NN}=2 which beyond Rayleigh resolution limit are called *super resolution methods.*[4].

Also, CRB is the ultimate limit for unbiased estimators. The connection between Rayleigh limit and CRB is done in D.N. Swingler, 1994.

The possible resolution of super resolution algorithms is approximately 1=10 of the array bandwidth under the ideal conditions.

Note that resolution is related with two signal sources overlapping in time and frequency (same signal). If there are two signals with different frequencies for example, they are already separable no need for resolution in spatial domain.

11. Approximate Formulas for Beamwidth

M element planar array with element positions given as follows:

$$P_k = \begin{bmatrix} X_k \\ Y_k \end{bmatrix} \qquad k = 1; 2 : : : ; M \tag{11.1}$$

Array center is the origin and following relation is valid.

$$\sum_{k=1}^{M} P_k = 0 \tag{11.2}$$

Equation (11.2) says that elements are symmetric with respect to origin.

[4] Chadwick, 2007

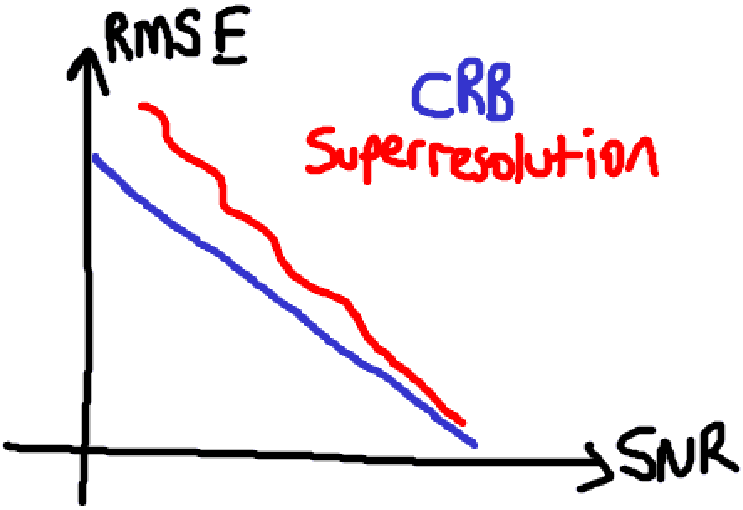

Figure 10.2. CRB and Superresolution Techniques

The array half power beamwidth for an elevation angle $\ = 90°$ is given as in (11.3).

$$Bw_{HP} \simeq 1{:}15\sqrt{2}\frac{\ }{2}\frac{1}{D(\ )} \tag{11.3}$$

where

$$D(\ ) \triangleq \sqrt{\frac{1}{M}\sum_{k=1}^{M} d_k^2(\ )} \tag{11.4}$$

where

$$dk(\ ) \triangleq x_k cos \ + y_k sin \tag{11.5}$$

For ULA, Equation (11.3) reduces to

$$Bw_{HP} \simeq \frac{\ }{(M-1)d|sin\ |} \tag{11.6}$$

Also as shown in Figure 11.1, narrowest beamwidth is obtained when $\ = 90°$.

12. (Beam) Array Pattern

Array output depends on the DOA of the incident plane wave hence array acts as a spatial filter. Array output is proportional to *Array Pattern* for a spatial direction. You can filter two sources in spatial domain even if they have same signals in time and frequency domain. Also it is possible to increase SNR with sensor arrays.

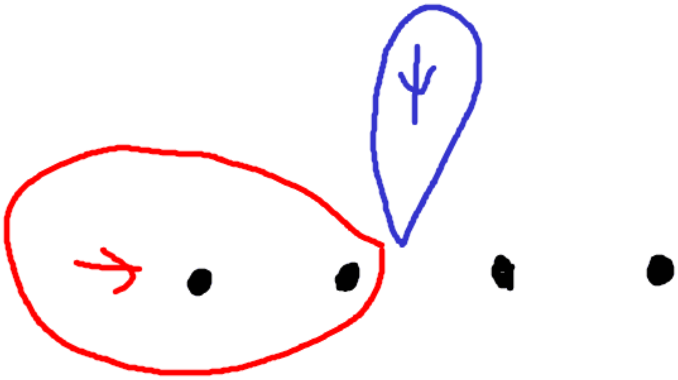

Figure 11.1. Narrowest Beamwidth at Broadside

$$B_{AP}(!;\ \ ;\ \ ) = R_0(\ \ ;\ \ )!_0^* e^{jk^T P_0} + ::: + R_{M-1}(\ \ ;\ \ )!_{M-1}^* e^{jk^T P_{M-1}} \tag{12.1}$$

where $Rj(\ \ ;\ \ )$ is element pattern for j^{th} sensor.

Also similar to array factor equations, $!$ in Equation (12.1) may drop in future, but it is there.

If we assume identical element patterns as follows:

$$R_0(\ \ ;\ \ ) = R_{M-1}(\ \ ;\ \ ) \tag{12.2}$$

then,

$$B_{AP}(\ \ ;\ \ ) = R_0(\ \ ;\ \ ) \times B_{AF}(\ \ ;\ \ ) \tag{12.3}$$

Array Pattern = Element Pattern x Array Factor

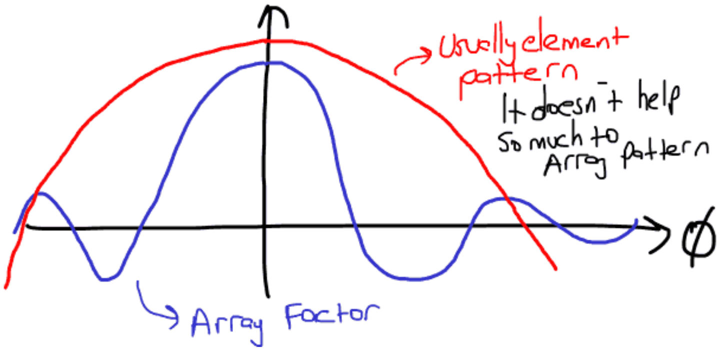

Figure 12.1. Effect of Element Pattern on Array Pattern

13. Array Performance Measures

(1) Directivity
(2) Array gain versus spatially white noise (Each element pick-ups uncorrelated white noise)(Kind of SNR)
(3) Sensitivity

13.1. Directivity.

Directivity of an ULA with $d = \;=2$ (inter-sensor distance) is given as follows:

$$D = \frac{1}{\sum_{j=0}^{M-1} |w_j|^2} = \frac{1}{w^H w} \tag{13.1}$$

where w is Mx1 weight vector. Also notice that uniform weighting maximizes the directivity (also Bw_{HP}) of ULA at expense of increased sidelobe level. ($w_j = 1{=}M$ then $D = M$)

For general case, directivity is proportional with number of sensors.

13.2. Array Gain.

Array processing improves the SNR by adding signals coherently and noise incoherently. The improvement is measured by array gain. Let the sensor output be for single source case:

$$y_{Mx1}(t) = a_{Mx1}(\;\;;\;\;)s_{1x1}(t) + e_{Mx1}(t) \tag{13.2}$$

Input SNR (at the sensor input)

$$SNR_{input} = \frac{{}^2_s}{{}^2_e} \tag{13.3}$$

After beamforming:

$$x(t) = w^H y(t) = w^H as(t) + w^H e(t) \tag{13.4}$$

For radar case $w = a$ is the optimum weight vector.
If $w^H a = 1$, then

$$SNR_{input} = \frac{{}^2_s}{\sum_{j=0}^{M-1} |w_j|^2 \;\; {}^2_e} \tag{13.5}$$

Array Gain is defined as

$$A_w = \frac{SNR_{output}}{SNR_{input}} = \frac{1}{\sum_{j=0}^{M-1} |w_j|^2} \tag{13.6}$$

In specific case, $A_w = M$. In general $A_w \leq M$.
Observations:

(1) Array gain under spatially white noise is valid for arbitrary arrays as long as $|w^H a|^2 = 1$.
(2) For ULA with $d = \;=2$ white noise array gain is identical to array directivity.
(3) For ULA with $d \neq \;=2$, $D \neq Aw$.
(4) Aw is maximum for uncorrelated noise when uniform weighting is used

13.3. **Sensitivity (Robustness).** Sensitivity to gain, phase and imprecise positioning of the sensors is important. For small variances and arbitrary array geometry.

$$TSE = A_w^{-1} \tag{13.7}$$

For ULA with uniform weighting $TSE = 1{=}M$.

14. Narrowband Model

We consider the problem of locating and radiating sources by using an array of M passive sensors. Source signals are sampled both in space and time by the sensor array.

Assumptions:

(1) The sources are assumed to be situated in the far field of the array.

$$distance > \frac{2D^2}{} \tag{14.1}$$

where D is the array aperture. This is limit distance but 10 is safer distance.

(2) Both sensors and sources are in the same plane.

(3) Sources are point emitters.

(4) Propagation medium is homogeneous. If medium is dispersive it is not true case. For example, sound waves in sea water.

(5) The number of sources, n, is assumed to be known. Otherwise it should be estimated.

(6) Array is calibrated. In other words, sensors can be assumed to be LTI systems. Their locations are known.

Let ${}_k$ denotes the time that wave travels from reference point to sensor k. Sensor output is

$$\bar{y}_k(t) = \bar{h}_k(t) * x(t - {}_k) + \bar{e}_k(t) \tag{14.2}$$

where $\bar{h}_k(t)$ (impulse response of sensor) is known and $x(t - {}_k)$, ${}_k$ and $\bar{e}_k(t)$ are unknown.

Write Equation (14.2) in Fourier domain using CTFT.

$$\bar{Y}_k(!) = \bar{H}_k(!)X(!)e^{-j\omega\tau_k} + \bar{E}_k(!) \tag{14.3}$$

Notice that Equation (14.3) or (14.2) (???) shows wideband model. It is a model which shows time information. Narrow band model doesn't show time information, it shows phase information. But phase information is ambiguous (2 periodicity). Time is unambiguous information. If you find time, you can find both angle and distance to the transmitter. In narrowband, you can estimate phase.

$x(t)$ is bandpass signal shown in Figure 14.1.

$s(t)$ is baseband signal shown in Figure 14.2.

Figure 14.1. Bandpass Spectrum

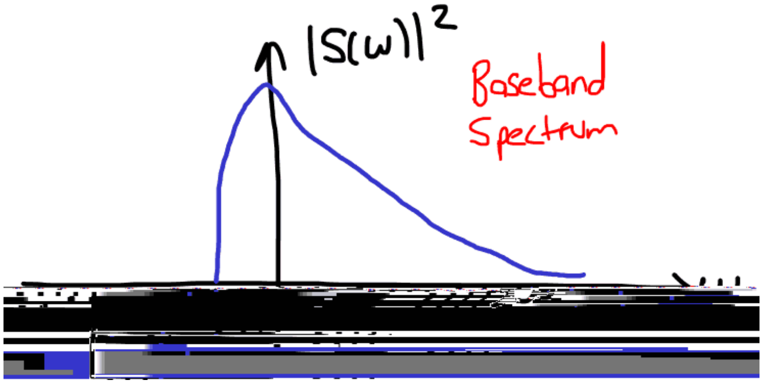

Figure 14.2. Baseband Spectrum

15. Modulation

Let us assume that the information signal is divided into two parts: $s_I(t)$, $s_Q(t)$; in phase and quadrature part respectively.

$$s(t) = s_I(t) + js_Q(t) \tag{15.1}$$

$s(t)$ is complex signal.

Also note the following relation.

$$s(t)e^{j\omega_c t} \overset{CTFT}{\longleftrightarrow} S(! - !_c) \tag{15.2}$$

Modulation is take signal and multiplies it with a complex exponential.

16. Questions

- Why bars in Equation (14.2)?

Part 3. 04/03/14 Lecture Note

17. Modulation

FIGURE HERE? FIGURE HERE?

Let,

$$x(t) \triangleq 2Re\{s(t)e^{j\omega_c t}\} \tag{17.1}$$

$$x(t) = s(t)e^{j\omega_c t} + s^*(t)e^{-j\omega_c t} \tag{17.2}$$

Taking CTFT of Equation (17.2).

$$X(\omega) = S(\omega - \omega_c) + S^*(-(\omega + \omega_c)) \tag{17.3}$$

Using (15.1)

$$x(t) = 2[s_I(t)cos\omega_c t - s_Q(t)sin\omega_c t] \tag{17.4}$$

$$x(t) = 2[s_I^2(t) + s_Q^2(t)]^{1/2}cos\left(\omega_c t + tan^{-1}\frac{s_Q(t)}{s_I(t)}\right) \tag{17.5}$$

$$x(t) = a(t)cos(\omega_c t + \phi(t)) \tag{17.6}$$

18. Demodulation

Let's demodulate $x(t)$,

$$x(t)e^{-j\omega_c t} \tag{18.1}$$

Taking CTFT,

$$X(\omega) = S(\omega) + S^*(-\omega - 2\omega_c) \tag{18.2}$$

Now put Equation (17.3) in Equation (14.3).

$$\bar{Y}_k(\omega) = \bar{H}_k(\omega)\left[S(\omega - \omega_c) + S^*(-\omega - \omega_c)\right]e^{-j\omega\tau_k} + \bar{E}_k(\omega) \tag{18.3}$$

In time domain, demodulated signal is

$$\tilde{y}_k(t) = \bar{y}_k(t)e^{-j\omega_c t} \tag{18.4}$$

Again in frequency domain,

$$\tilde{Y}_k(\omega) = \bar{H}_k(\omega + \omega_c)\left[S(\omega) + S^*(-\omega - 2\omega_c)\right]e^{-j(\omega+\omega_c)\tau_k} + \bar{E}_k(\omega + \omega_c) \tag{18.5}$$

After the low-pass filtering

$$Y_k(\omega) = H_k(\omega + \omega_c)S(\omega)e^{-j(\omega+\omega_c)\tau_k} + E_k(\omega + \omega_c) \tag{18.6}$$

19. Narrow Band Model

Now assume that the received signal is narrow band. Then $|S(\omega)|$ decreases rapidly with ω. Mathematically, $B\Delta Tmax < 1$.

Under that assumption, it means that impulse response only affect at carrier frequency.

$$Y_k(\omega) = H_k(\omega_c)S(\omega)e^{-j\omega_c\tau_k} + E_k(\omega + \omega_c) \tag{19.1}$$

Note that delay term is constant, not depends on signal.

Remark: If the signal is broadband but we use narrow band filters, we obtain (19.1) if

- Sensor (filter) frequency response if flat over the passband. $H_k(\omega + \omega_c) = H_k(\omega_c)$.
- Signal spectrum varies over passband.

(19.1) in time domain can be written as:

$$y_k(t) = H_k(\omega_c)s(t)e^{-j\omega_c\tau_k} + e_k(t) \tag{19.2}$$

Also $t = pT$ is possible in (19.2) where $p = 1, \ldots N$.

Now let's assume that sensors are identical, i.e., $Hj(\omega_c) = Hk(\omega_c)$.

In case of ULA, normalizing with respect to first sensor signal

FIGURE HERE

$$a(\ ) = \begin{bmatrix} 1 \\ e^{-j\omega_c\tau_1} \\ \\ \\ \end{bmatrix} \tag{19.3}$$

(19.6)
$$y(t) = \begin{bmatrix} 1 & 1 & ::: \\ e^{j2\pi/\lambda d cos\phi_1} & e^{j2\pi/\lambda d cos\phi_2} & ::: \\ e^{j2\pi/\lambda 2 d cos\phi_1} & e^{j2\pi/\lambda 2 d cos\phi_2} & ::: \\ \vdots & \vdots & \vdots \\ e^{j2\pi/\lambda (M-1) d cos\phi_1} & e^{j2\pi/\lambda (M-1) d cos\phi_2} & ::: \end{bmatrix}_{M\times n} \begin{bmatrix} s_1(t) \\ s_2(t) \\ s_3(t) \\ \vdots \\ s_n(t) \end{bmatrix}_{n\times 1} + \begin{bmatrix} e_1(t) \\ e_2(t) \\ e_3(t) \\ \vdots \\ e_n(t) \end{bmatrix}_{M\times 1}$$

where $t = 1; 2; ::: N$ and N is the number of observations.

Equation (19.6) is given for ULA with M sensors, n sources. Note that, in that equation steering vector has Vandermonde structure. This special for ULA. This is useful if we have coherent sources. In that case, covariance matrix is rank deficient. This is one of the major problems in array processing. ULA is the only (?) geometry for perfect solution of coherent signals due to Vandermonde structure. You can apply forward-backward spatial smoothing.

Problem is the given N observations of the array output, find $\ _i$ and $s_i(t)$ where $i = 1; 2; ::: n$.

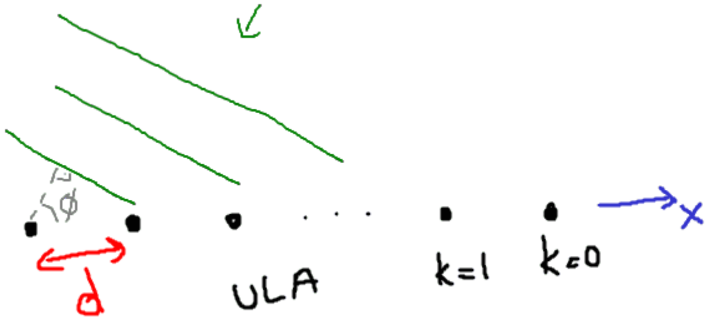

Figure 19.1. ULA Delay Relation

As mentioned,

(19.7)
$$_k = -\frac{g^T P_k}{c} = -\frac{d cos}{c}(k-1)$$

Then,

(19.8)
$$a(\ ) = \begin{bmatrix} 1 & e^{j\omega_c d cos\phi/c} & e^{j2\omega_c d cos\phi/c} & ::: \end{bmatrix}^T$$

where $!_c = 2\ f_c$ and $\ = c{=}f_c$.
Then,

(19.9)
$$a(\ ) = \begin{bmatrix} 1 & e^{j2\pi/\lambda d cos\phi} & e^{j2\pi/\lambda 2 d cos\phi} & ::: \end{bmatrix}^T$$

(19.10)
$$a(\ ) = \begin{bmatrix} 1 & e^{j\omega_p} & e^{j2\omega_p} & ::: \end{bmatrix}^T \text{(???)}$$

where $!_p$ is spatial frequency.

20. Sampling

20.1. **Sampling in Time.** Nyquist Theorem holds. $f_s > 2f_c$. (Assume narrowband, largest frequency is f_c)

20.2. **Sampling in Spatial Domain.** In (19.10), $a(\ )$ is uniquely specified if and only if $|w_p| < \ $. Any succeeding sensor phase difference should be less than . Then $|f_p| < 1{=}2$,

$$d|cos \ | < \ {=}2 \tag{20.1}$$

(20.1) holds for any if $d < \ {=}2$. This is the condition for no spatial aliasing (Spatial Nyquist Theorem).

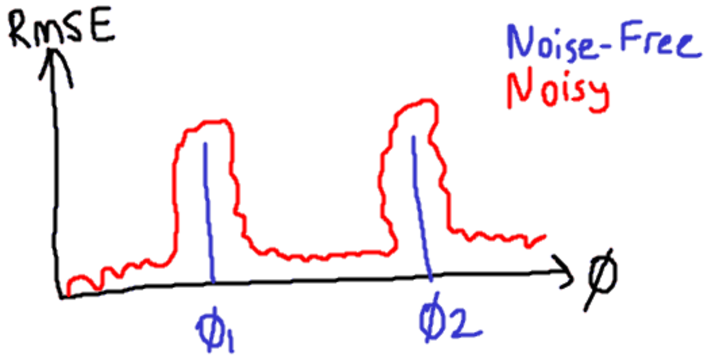

Figure 20.1. Angle Aliasing Under Noise-Free and Noisy Cases

Finding the aliased angles (20.2),

$$e^{j2\pi/\lambda d cos\phi_1} = e^{j2\pi/\lambda d(cos\phi_2+\lambda/dr} \tag{20.2}$$

where r is an integer.

As an example, $cos\ _1 = cos\ _2 + \ {=}dr$. Let take $\ = d$, $r = 1$.

$cos\ _1 - cos\ _2 = 1$, then $(\ _1 = 0°;\ _2 = 90°)$, $(\ _1 = 90°;\ _2 = 180°)$. Several angles could also be found taking $-2 \le r \le 2$. As you can see $d = \ $ is a problematic choice.

In theory, knowing exactly aliased angle pairs may be sufficient to eliminate problem. However, in practice, noisy environment case there will be a continuous range of aliasing angles instead of finite angle values due to noise etc. as shown in Figure 20.1.

21. Spaces of matrix, A

21.1. **Null Space, $N(A)$.** is the space spanned by x which satisfies $Ax = 0$. Denoted by $N(A)$. Let A be matrix represented by SVD.

$$A_{M\times N} = \begin{bmatrix} u_1 & u_2 \end{bmatrix} \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_1^H \\ v_2^H \end{bmatrix} = u_1 \Sigma v_1^H \tag{21.1}$$

where size of u_1 is r, u_1 is $M - r$, v_1^H is r and v_2^H is $N - r$.

Then

$$\text{Dim of N(A)} = N - \text{\# of singular values different than 0} \tag{21.2}$$

The basis vectors of N(A) are the eigenvectors in V that corresponds to zero singular values. Then v_2 is the orthonormal basis for N(A). N(A) = span(v_2).

21.2. **Row Space, $R(A^H)$.** is spanned by the singular vectors in V with singular values different than zero. Then v_1 is the orthonormal basis for $R(A^H)$. $R(A^H) =$ span(v_1). This is space spanned by rows of A.

Note that v_1 and v_2 are orthonormal (null space and row space is orthonormal).

21.3. **Column (Range) Space, $R(A)$.** is the subspace spanned by the columns of A. Then u_1 is the orthonormal basis for $R(A)$. $R(A) =$ span(u_1).

21.4. **Left-Null Space, $N(A^H)$.** u_2 is the orthonormal basis for $N(A^H)$. $N(A^H)$ = span(u_2).

The subspaces $R(A)$ and $N(A^H)$ are orthogonal and they span C^M. The subspaces $R(A^H)$ and $N(A)$ are orthogonal and they span C^N.

If $R = AA^H$, then it is positive semi-definite $M \times M$. Its eigenvectors are orthogonal.

$$R_{M\times M} = \begin{bmatrix} u_1 & u_2 \end{bmatrix} \begin{bmatrix} \Sigma\Sigma^H & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_1^H \\ v_2^H \end{bmatrix} \tag{21.3}$$

In that case, range space of R is equivalent to signal space of R which is spanned by vectors u_1. Similarly, null space of R is equivalent to noise space of R which is spanned by vectors u_2. Also these two spaces are orthogonal to each other (previously not for arbitrary rectangular matrix.). Also, $u_1^H u_2 = 0$.

11:40

22. Projection Matrices

Theorem 1

Let $\{a_1; a_2; :::; a_n\}$ be any basis for subspace W of C^M. Form $M \times n$ matrix.

$$A = \begin{bmatrix} a_1 & a_2 & ::: & a_n \end{bmatrix} \tag{22.1}$$

The projection matrix for W is $A(A^HA)^{-1}A^H = P$.

Theorem 2

Let $\{u_1; u_2; :::; u_n\}$ be a unitary basis for subspace W of C^M. Form the matrix $U = \begin{bmatrix} u_1 & u_2 & ::: & u_n \end{bmatrix}$. Then $UU^H = P$ is the orthogonal (???) projection matrix for the W.

Both theorem 1 and 2 projects onto signal space.

Any projection matrix satisfies:

- $P^2 = P$, Idempotent
- P is symmetric
- Eigenvalues of P is 0 or 1.

$$R = E\{y(t)y(t)^H\} = AE\{s(t)s(t)^H\}A^H + E\{e(t)e(t)^H\} \tag{22.2}$$

Note that in (22.2), noise and signal is assumed to be uncorrelated.

$$R = AR_sA^H + {}^2I = U\Lambda U^H \tag{22.3}$$

where

$$\Lambda \triangleq \begin{bmatrix} {}_1 & 0 & ::: & 0 \\ 0 & {}_2 & ::: & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & ::: & {}_M \end{bmatrix} \tag{22.4}$$

Also white noise is assumed at (22.3).
Also

$$R = U_s\Lambda_sU_s^H + U_e\Lambda_eU_e^H \tag{22.5}$$

where U_s corresponds to signal space eigenvectors and $U_s\Lambda_sU_s^H$ is signal space. Columns of U_s spans $R(A)$. Similarly, U_e corresponds to noise space eigenvectors and $U_e\Lambda_eU_e^H$ is noise space. Columns of U_e spans $N(A^H)$.

Now let's define,

$$\Pi \triangleq U_sU_s^H = A(A^HA)^{-1}A^H \tag{22.6}$$

Note that (22.6) is a projection matrix onto signal space. Also,

$$\Pi^\perp \triangleq U_eU_e^H = I - A(A^HA)^{-1}A^H \tag{22.7}$$

Note that (22.7) is a projection matrix onto noise space.
Notice that $\Pi + \Pi^\perp = I$.
For no noise, the array output is confined to n-dimensional subspace of complex M dimensional space(where n is the number of sources and M is the number of sensors) which is spanned by the steering vectors. $n < M$ should be satisfied for a solution. For no noise case, $rank(R)$ is n. R is rank deficient in that case (???).

23. Vandermonde Matrix

A matrix $A \in C^{M\times N}$ is called Vandermonde if it has the structure

$$A = \begin{bmatrix} 1 & 1 & ::: & 1 \\ z_1 & z_2 & ::: & z_N \\ z_1^2 & z_2^2 & ::: & z_N^2 \\ \vdots & \vdots & \vdots & \vdots \\ z_1^{M-1} & z_2^{M-1} & ::: & z_N^{M-1} \end{bmatrix} \tag{23.1}$$

If $z_k \neq z_p$ $\forall k; p$ $k \neq p$ and $M \geq N$, then rows of A are linearly independent and $rank(A) = N$. Note that ULA steering matrix has this form and forward-backward spatial smoothing (FBSS) algorithm can be applied in this case which solves multipath problems.

24. Least Squares Solution (LS)

$Ax = B$ where $A^{M\times N}$, $x^{N\times P}$, $B^{M\times P}$. Solution is pseudo-inverse.

$$x = A^{+}B = (A^H A)^{-1} A^H B = v_1 \Sigma^{-1} u_1^H B \tag{24.1}$$

where $A = U\Sigma V^H$.

(24.1) is called Moore-Penrose Pseudo-Inverse. It sets small eigenvalues to zero. By that way, $||Ax - B||^2$ is minimized. In other words, $Ax = B + \Delta B$ problem is solved in least-squares sense such that $min||\Delta B||^2$ is obtained.

25. Total Least Squares Solution (TLS)

Consider the minimum perturbations ΔA, Δb. Then minimize $||\Delta A \Delta B||^2$, s.t. $(A + \Delta A)x = B + \Delta B$.

In TLS, there is a an also error on A. But in LS, A is error free.

Let,

$$\begin{bmatrix} A & B \end{bmatrix} = \begin{bmatrix} \tilde{u}_1 & \tilde{u}_2 \end{bmatrix} \begin{bmatrix} \tilde{\Sigma}_1 & 0 \\ 0 & \tilde{\Sigma}_2 \end{bmatrix} \begin{bmatrix} \tilde{v}_1^H \\ \tilde{v}_2^H \end{bmatrix} \tag{25.1}$$

where A is $M \times N$, B is $M \times P$. Size of $\tilde{u}_1$ is N, $\tilde{u}_2$ is $M - N$, $\tilde{v}_1^H$ is N and $\tilde{v}_2^H$ is P.

Partition $\tilde{v}_2$ as

$$\tilde{v}_2^H = \begin{bmatrix} \tilde{v}_{21}^H & \tilde{v}_{22}^H \end{bmatrix} \tag{25.2}$$

where size of $\tilde{v}_{21}^H$ is N and size of $\tilde{v}_{22}^H$ is P.

Then,

$$x_{TLS} = -\tilde{v}_{21} \tilde{v}_{22}^{-1} \tag{25.3}$$

if $\tilde{v}_{22}^{-1}$ exists.

If you know A very well, use LS otherwise use TLS.

26. Quadratic Minimization

Let A be $N \times N$ Hermitian symmetric positive semi-definite matrix. $x^{N\times M}$, $B^{N\times K}$, $C^{M\times K}$. Then, minimize $x^H A x$ s.t. $B^H x = C^H$. It corresponds to beamforming. For example x may be weight vector. Minimize sidelobes, except at signal of interest (SOI). Optimum solution is,

$$x_o = A^{-1} B (B^H A^{-1} B)^{-1} C^H \tag{26.1}$$

Proof

$$\mathcal{L} = x^H A x + \ {}^H(B^H x - C^H) + \ {}^T(B^T x^* - C^T) \tag{26.2}$$

$$\frac{@\mathcal{L}}{@x} = x^H A + \ {}^H B^H = 0 \tag{26.3}$$

Then,

$$A^H x + B \quad = 0 \tag{26.4}$$

$$x = -(A^H)^{-1} B \tag{26.5}$$

If $A^H = A$,

$$x = -A^{-1} B \tag{26.6}$$

Apply the constraint

$$B^H x = C^H \tag{26.7}$$

$$- B^H A^{-1} B \quad = C^H \tag{26.8}$$

$$= -(B^H A^{-1} B)^{-1} C^H \tag{26.9}$$

$$x = A^{-1} B (B^H A^{-1} B)^{-1} C^H \tag{26.10}$$

27. Check

Equation (19.10): $!_p$?

Part 4. 11/03/14 Lecture Note

28. Cramer-Rao Lower Bound (CRB or CRLB)

CRLB is a lower bound on the accuracy of any unbiased estimator. CRLB provides an algorithm independent benchmark against which different algorithms can be compared. Performance is usually measured by root-mean-squared error (RMSE).

$$MSE = E\{(\hat{\ } - \ )^2\} = Var(\hat{\ }) + Bias^2(\hat{\ }) \tag{28.1}$$

where   is true and $\hat{\ }$ is estimated value.

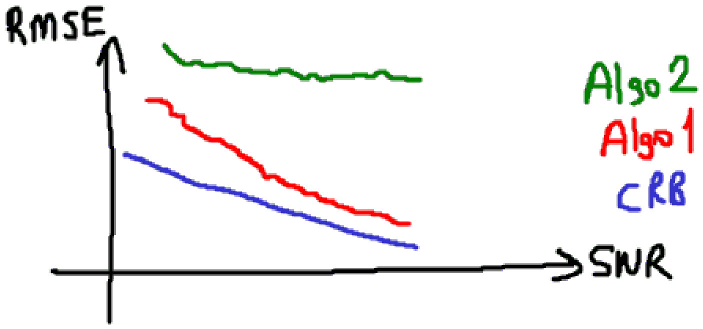

Figure 28.1. An Example RMSE Plot

Assume that $\hat{\ }$ is an unbiased estimate of   and let the covariance matrix of $\hat{\ }$, $C_{\hat{\theta}}$.

$$C_{\hat{\theta}} = E\{(\hat{\ } - \ )(\hat{\ } - \ )^T\} \tag{28.2}$$

Then there is a matrix such that $C_{\hat{\theta}} \geq C_{RB}$ in the sense that $C_{\hat{\theta}} - C_{RB}$ is a positive semi-definite matrix. (Diagonal values of this matrix is greater than diagonal values of that matrix. Individual elements variances are greater.)

29. Fisher Information Matrix

Let x be vector of observations,   be the vector of parameters.

$$J = FIM(\ ) = E\{g_{x,\theta} g_{x,\theta}^T\} \tag{29.1}$$

where

$$g_{x,\theta} \triangleq \nabla_\theta ln f_{x,\theta}(x;\ ) = \nabla_\theta P(x;\ ) \tag{29.2}$$

where $f_{x,\theta}(x;\ )$ is likelihood function. $g_{x,\theta}$ is called as gradient of the log likelihood function. Elements of J matrix

$$J_{ij} \triangleq E\left\{\frac{@P(x;\ )}{@\ _i}\frac{@P(x;\ )}{@\ _j}\right\} = -E\left\{\frac{@^2P(x;\ )}{@\ _i@\ _j}\right\} \tag{29.3}$$

Then,

$$[CRB(\quad)]_{ii} = [J^{-1}]_{ii} \tag{29.4}$$

Notice the indices in (29.4), we are only interested in the variances.

If $\hat{\ }_i$ is the i^{th} component of $\quad$, then its variance is lower bounded by:

$$Var\{\hat{\ }_i\} \geq FIM(\quad)^{-1}_{ii} \tag{29.5}$$

30. CRB Example

Let

$$y(t) = A(\quad)s(t) + e(t) \tag{30.1}$$

$$R_y = E\{y(t)y(t)^H\}) = AR_sA^H + R_e = AR_sA^H + \ {}^2_e I \tag{30.2}$$

In (30.2) it is assumed that signal and noise are uncorrelated and noise is white noise both temporally and spatially.

Then,

$$A = \begin{bmatrix} a(\ {}_1) & a(\ {}_2) & : : : & a(\ {}_n) \end{bmatrix} \tag{30.3}$$

Let define a vector

$$\triangleq \begin{bmatrix} \ {}^T & \ {}^T & \ {}^2_e \end{bmatrix} \tag{30.4}$$

where $\quad$ is vector composed of the elements of R_s.

$$\triangleq \begin{bmatrix} \ {}_1 & \ {}_2 & : : : & \ {}_n \end{bmatrix}^T \tag{30.5}$$

Then,

$$FIM_{p,k} = N \times Tr\left\{ \frac{@R_y}{@\ {}_p} R_y^{-1} \frac{@R_y}{@\ {}_k} R_y^{-1} \right\} \tag{30.6}$$

$$CRB(\quad) = \frac{\ {}^2}{2N}\left\{Re\left[(D^H\Pi_A^T D)\odot(R_sA^HR_y^{-1}AR_s)^T\right]\right\}^{-1} \tag{30.7}$$

where N is the number of samples (snapshots) and $\odot$ is Hadamard product operator.

And

$$D \triangleq \begin{bmatrix} d_1 & d_2 & : : : & d_n \end{bmatrix} \tag{30.8}$$

$$d_k \triangleq \frac{@a(\ {}_k)}{@\ {}_k} \tag{30.9}$$

$$\Pi_A \triangleq A(A^HA)^{-1}A^H \tag{30.10}$$

$$\Pi_A^T \triangleq I - \Pi_A \tag{30.11}$$

For a single source,

$$CRB \simeq \frac{1}{2 \times N \times SNR \times ||\dot{a}(\ )||^2} \tag{30.12}$$

where $\dot{a}(\ )$ is derivative of the steering vector. $\dot{a}(\ ) = d_1$.
For M element ULA,

$$CRB \simeq \frac{\ ^2}{8\ \ ^2 \times N \times SNR \times cos^2\ \ \times \bar{d}^2} \tag{30.13}$$

where

$$\bar{d}^2 \triangleq \sum_{m=1}^{M} d_m^2 \tag{30.14}$$

where d_m is the distance of the m^{th} sensor to the origin.

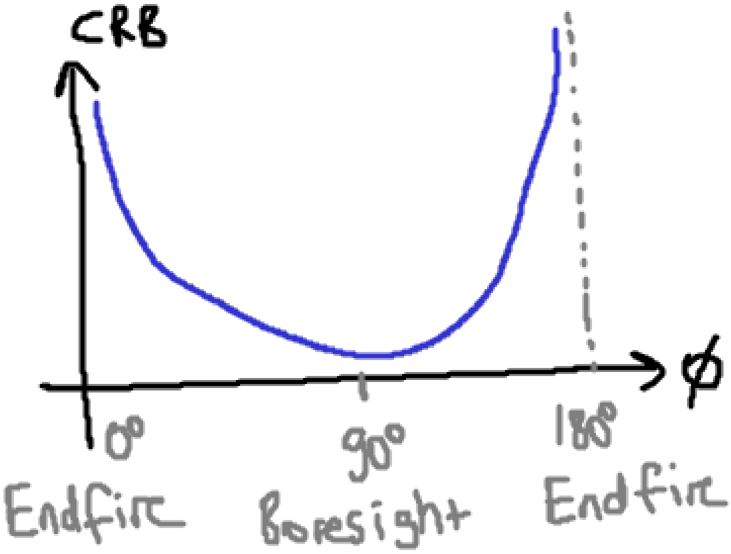

Figure 30.1. Approximate CRB of ULA with Single Source

When the measurements are (locally) loosely dependent on the parameter, FIM approaches to zero or has singular values as shown in Figure 30.1. There are singular points for CRB. = 180° (endfire) for a ULA is a singular point. CRB is not predicting correctly for this case. For example, MUSIC algorithm can find at those points with accuracy of approximately 10°. Therefore, approximation is invalid in that angles.

31. Classical Methods for DOA Estimation

- Amplitude Based Methods
- Phase Based Methods
- Frequency Based Methods (Doppler)
- Time Based Methods
- Hybrid Methods

32. Common Direction Finding Techniques

- Directional Antenna (Sensor)
- Wattson-Watt
- Pseudo-Doppler
- Interferometer

33. Directional Antenna (Sensor)

A single directional antenna is rotated in order to find the DOA angle.

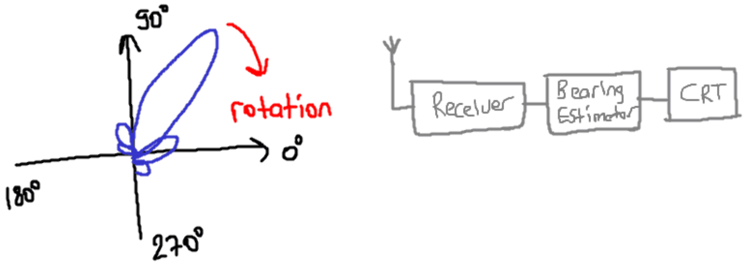

Figure 33.1. Directional Antenna

10:40
Advantages:

- High sensitivity due to antenna directivity
- Simple and cheap (Single channel system)
- Resolution of multipath signals
- Same antenna can be used for direction finding and monitoring.

Disadvantages:

- Probability of intercept is inversely proportional with directivity
- Fails for short duration signals
- Mechanical rotation is problematic.

34. Butter Array

Instead of a single antenna, an antenna array with different phase shifters and combiners are used to obtain directional patterns.

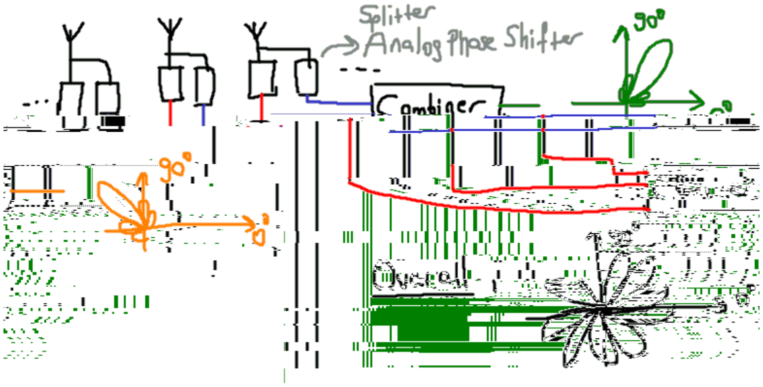

Figure 34.1. Butter Array

35. Monopulse

A variation of beamformer direction finding. Usually used in radar systems for tracking sources. Two overlapping antenna beams are formed which are steered slightly in different directions. This method is taking the difference between the output of two beams.

Figure 35.1. Two Beams

Response if the monopulse system is given as

$$b(\quad) = \frac{1}{\Delta}\left(\left|a^H(\quad + \frac{\Delta}{2})y\right|^2 - \left|a^H(\quad - \frac{\Delta}{2})y\right|^2\right) \tag{35.1}$$

where Δ is the offset angle.
Then,

$$b(\quad) = \frac{1}{\Delta}\left(B(\quad + \frac{\Delta}{2}) - B(\quad - \frac{\Delta}{2})\right) \tag{35.2}$$

where $B(\quad)$ is the beam pattern. And if Δ is small, then

$$b(\quad) \simeq \dot{B}(\quad) \tag{35.3}$$

Output is positive if the emitter is to the right of the boresight and negative otherwise.

This method is useful for tracking.

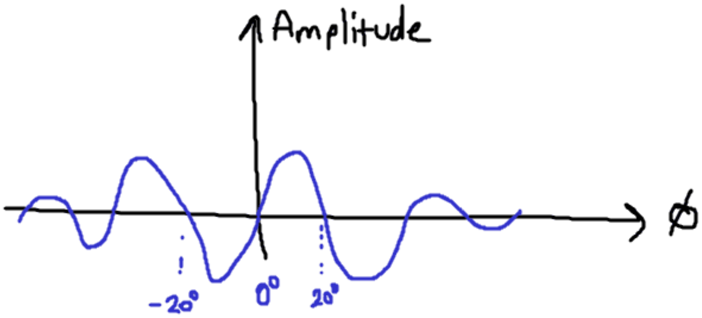

Figure 35.2. Monopulse Response

36. Sum-Difference Method

This is used to implement the monopulse idea with a sensor array. The array is divided into two parts: + and - parts. Let the array output be:

$$y(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_M(t) \end{bmatrix} \tag{36.1}$$

Sum response is obtained as follows

$$u_\Sigma = w_\Sigma^H y(t) \tag{36.2}$$

where

$$w_\Sigma \triangleq \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \tag{36.3}$$

Similarly difference response is obtained as follows

$$u_\Delta = w_\Delta^H y(t) \tag{36.4}$$

where

$$w_\Delta \triangleq \begin{bmatrix} 1 \\ 1 \\ \vdots \\ -1 \\ -1 \end{bmatrix} \tag{36.5}$$

where in (36.5), half is 1 and half is -1.

The sum and difference beam patterns are

$$\Sigma(\ ) = w_{\Sigma}^{H} a(\ ) \tag{36.6}$$

$$\Delta(\ ) = w_{\Delta}^{H} a(\ ) \tag{36.7}$$

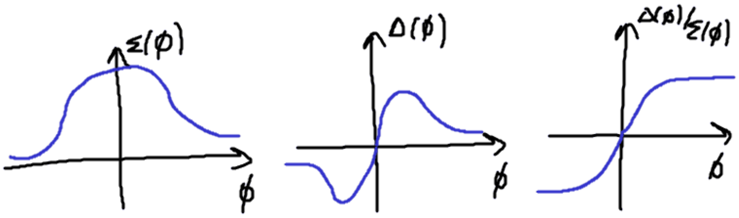

Figure 36.1. Sum-Difference Method

For ULA,

$$\Sigma(\ ) = \sum_{k=1}^{M} e^{j(k-1)\beta} = \frac{sin\left(M\frac{\ }{2}\right)}{sin\left(\frac{\ }{2}\right)} e^{j(M-1)\beta/2} \tag{36.8}$$

where

$$\triangleq \frac{2}{\ } dcos \tag{36.9}$$

Also,

$$\Delta(\ ) = \sum_{k=1}^{M/2} j e^{j(k-1)\beta} + \sum_{k=M/2+1}^{M} -j e^{j(k-1)\beta} = \frac{2sin^{2}\left(M\frac{\ }{4}\right)}{sin\left(\frac{\ }{2}\right)} e^{j(M-1)\beta/2} \tag{36.10}$$

Then,

$$f(\ ) \triangleq \frac{\Delta(\ )}{\Sigma(\ )} \tag{36.11}$$

$$f(\ ) = tan\left(\frac{M}{4}\right) = tan\left(\frac{M dcos}{2}\right) \tag{36.12}$$

In practice,

$$\simeq f^{-1}\left[Re\left\{\frac{u_{\Delta}}{u_{\Sigma}}\right\}\right] \tag{36.13}$$

It is an amplitude technique. It is used in radar. However, since it is not using the full information, it is not used in direction finding.

37. Wattson-Watt Method

Frequently used. It is small and accuracy is acceptable. This is also amplitude comparison technique. It has 180° ambiguity. It is solved by a sense antenna or by combining all antenna outputs ti get reference signal.

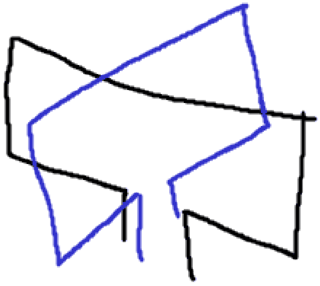

Figure 37.1. Crossed Loop Antenna

37.1. **Crossed Loop Antenna.** Advantages of decrease in size. But it is sensitive to sky waves (reflections from sky etc..).

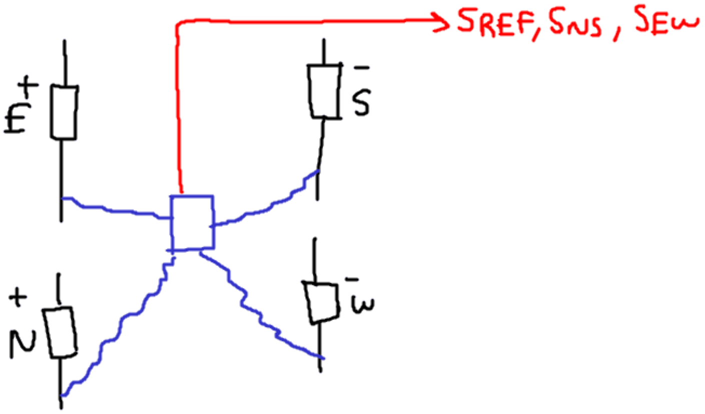

Figure 37.2. Adcock Antenna

37.2. **Adcock Antenna.** Then,

$$S_{REF}(t) \triangleq S_N(t) + S_S(t) + S_E(t) + S_W(t) \tag{37.1}$$

$$S_{NS}(t) \triangleq S_N(t) - S_S(t) \tag{37.2}$$

$$S_{EW}(t) \triangleq S_E(t) - S_W(t) \tag{37.3}$$

Then,

$$S_{NS}(t) = s(t)sin(\ )\left(e^{j\frac{2}{d}cos\phi sin\theta} - e^{-j\frac{2}{d}cos\phi sin\theta}\right) \tag{37.4}$$

$$S_{EW}(t) = s(t)sin(\ )\left(e^{j\frac{2}{d}sin\phi sin\theta} - e^{-j\frac{2}{d}sin\phi sin\theta}\right) \tag{37.5}$$

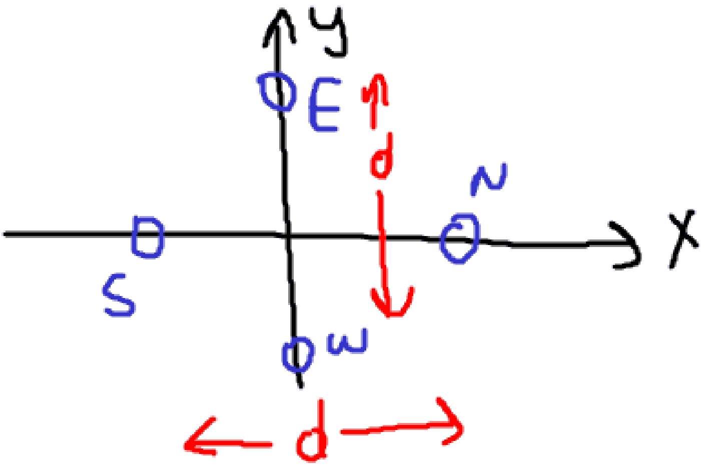

Figure 37.3. Sensor Placement

$$\simeq tan^{-1}\frac{S_{EW}(t)}{S_{NS}(t)} = tan^{-1}\left[\frac{sin\left(\frac{2}{}dsin\ \ sin\ \right)}{sin\left(\frac{2}{}dcos\ \ sin\ \right)}\right] \tag{37.6}$$

For (37.6), $d=$ should be small.

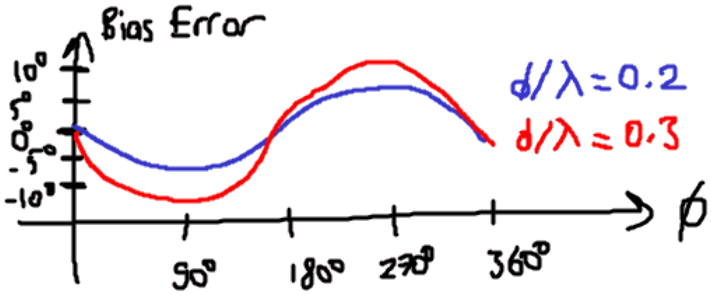

Figure 37.4. Bias Error

11:40
Advantages:

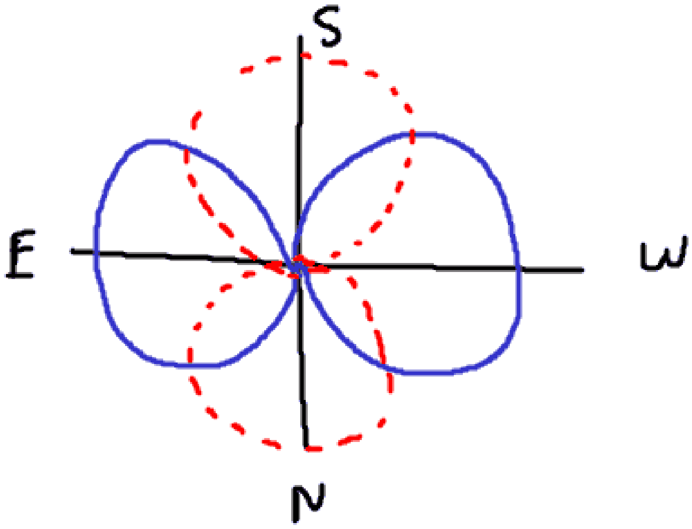

Figure 37.5. Figure of Eight

- Instantaneous bearing report
- Relatively small size for the operating frequency
- Acceptable bearing accuracy (5°)

Disadvantages:

- Better accuracy is possible
- No immunity to multipath signals, co-channel interference. (Indeed there is no algorithm immune to multipath except maximum likelihood which is a super-resolution method(???))

Crossed loop has problems especially for signals coming from elevated heights such as sky waves. This generates bearing errors. Adcock antennas are proposed to solve this problem.

$D=\ < 0{:}2$ is desired in order to have a figure of eight characteristics close to circle. (Really ???) But increase in separation is good thing to increase aperture.

38. Pseudo-Doppler Method

It is possible to mechanically rotate an antenna on a circle to find the DOA angle from Doppler effect. However, electronic switching is preferred in order to avoid mechanical operation.

Advantages:

- Simple operating principle
- Single channel, cheap, system

Disadvantages:

- Antenna sampling introduces signal distortions and DF error.
- DF accuracy decreases as elevation angle decreases.
- Accuracy is low, possibly worse than Wattson-Watt.
- Dwell time on signal is low due to switching.

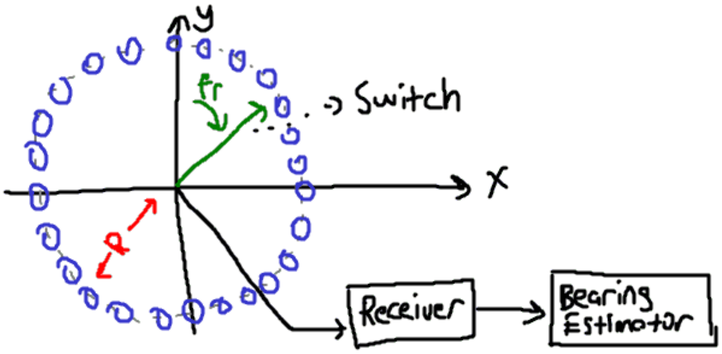

Figure 38.1. Pseudo-Doppler Method

$$s(t) = acos\left(!_0 t + \frac{2\ \ R}{\ }cos(!_r t - \ \ ) + '\right) \tag{38.1}$$

where $!_0$ is operating frequency, $!_r$ is rotation frequency, and $'$ is constant phase offset.

Define

$$\triangleq !_0 t + \frac{2\ \ R}{\ }cos(!_r t - \ \ ) + ' \tag{38.2}$$

$$!(t) = \frac{d}{dt} = !_0 - \frac{2\ \ R}{\ }sin(!_r t - \ \ ) \tag{38.3}$$

where $!(t)$ is instantaneous frequency.

If a high-pass filter is used, then

$$s_D(t) = -\frac{2\ \ R}{\ }sin(!_r t - \ \ ) \tag{38.4}$$

Then modulate the signal

$$s_E(t) = s_D(t)sin(!_r t) = -\frac{R}{\ }!_r\left[cos(\ \ ) - cos(2!_r t - \ \ )\right] \tag{38.5}$$

Apply low-pass filter

$$s_F(t) = -\frac{R}{\ }!_r cos(\ \ ) \tag{38.6}$$

Then,

$$= cos^{-1}\left[\frac{\ }{R!_r}s_F(t)\right] \tag{38.7}$$

Generally rotation frequency is around kHz.

This method is used especially in air planes. Accuracy is around 3 to 10° is sufficient for search and rescue operations for example.

Generally, accuracy: Interferometer > Wattson-Watt > Pseudo-Doppler

Part 5. 18/03/14 Lecture Note

39. Interferometer

This method uses the phase information. There are both phase dependent and phase and amplitude dependent implementation of interferometer technique.

39.1. **Correlative Interferometer.** This method is both amplitude and phase based technique. In that technique, you have sensors and transmitter turning around. You simply record signals for each angle. You calibrate sensor. At runtime, you correlate with table and find angle. Simple, effective and commercially used.

39.2. **Phase-Based Interferometer.**

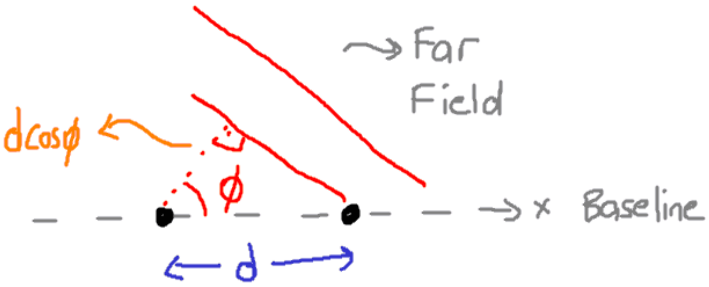

Figure 39.1. Far Field Source, 2-Channel Array

39.2.1. *2-Channel Case.*

$$= \frac{dcos}{c} \tag{39.1}$$

Note that in Figure 39.1, source is far field.

$$y_1(t) = s(t)e^{j\omega_c t} \tag{39.2}$$

is assumed to be received by the reference sensor. Then,

$$y_2(t) = y_1(t - \quad) = s(t - \quad)e^{j\omega_c(t-\tau)} \tag{39.3}$$

Using narrowband assumption:

$$y_2(t) = s(t)e^{j\omega_c(t-\tau)} = y_1(t)e^{-j\omega_c\tau} \tag{39.4}$$

Notice that we apply narrowband assumption immediately, not after demodulation.

Let,

$${}_1(t) \triangleq \angle y_1(t) \tag{39.5}$$

$${}_2(t) \triangleq \angle y_2(t) = {}_1(t) - \frac{!_c dcos}{c} = {}_1(t) - \frac{2}{\quad}dcos \tag{39.6}$$

Then,

$$(39.7) \qquad {}_1(t) - \ {}_2(t) = \frac{2}{\ }dcos$$

Finally

$$(39.8) \qquad = cos^{-1}\left[(\ {}_1(t) - \ {}_2(t))\frac{}{2\ \ d}\right]$$

Angle is found by cross-correlating two signals. Angle is found with respect to baseline. There is also a ambiguity for minus angles. Also elevation angle could not be identified.

39.2.2. *3-Channel Case.* It can also find elevation angle.

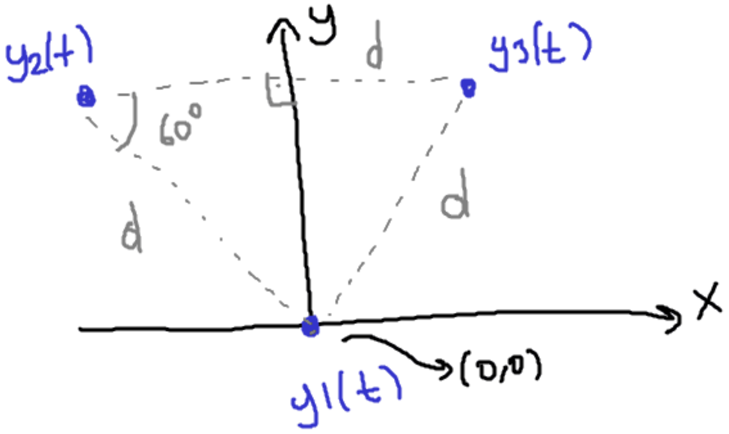

Figure 39.2. 3-Channel Array

Most robust design when geometry is equilateral triangle.
Take the first sensor as the reference

$$(39.9) \qquad y_2(t) = s(t)e^{j2\pi/\lambda}\,[-dsin(\ \ =6)cos\ \ sin\ \ + dcos(\ \ =6)sin\ \ sin\ \ ]$$

$$(39.10) \qquad y_3(t) = s(t)e^{j2\pi/\lambda}\,[dsin(\ \ =6)cos\ \ sin\ \ + dcos(\ \ =6)sin\ \ sin\ \ ]$$

Let

$$(39.11) \qquad {}_1(t) \triangleq \angle y_1(t) = 0$$

$$(39.12) \qquad {}_2(t) \triangleq \angle y_2(t) = \frac{2}{\ }\,[-dsin(\ \ =6)cos\ \ sin\ \ + dcos(\ \ =6)sin\ \ sin\ \ ]$$

$$(39.13) \qquad {}_3(t) \triangleq \angle y_3(t) = \frac{2}{\ }\,[dsin(\ \ =6)cos\ \ sin\ \ + dcos(\ \ =6)sin\ \ sin\ \ ]$$

$$\triangleq \frac{({}_2 - {}_1) + ({}_3 - {}_1)}{({}_3 - {}_1) - ({}_2 - {}_1)} \tag{39.14}$$

$$= \frac{cos(=6)sin}{sin(=6)cos} \tag{39.15}$$

As a note,

$$\hat{R}_y = \frac{1}{N}\sum_{t=1}^{N} y(t)y(t)^H = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{39.16}$$

then, ${}_2 = r_{12}$ and ${}_3 = r_{13}$.
Azimuth angle could be found as

$$= tan^{-1}[tan(=6) \ ] \tag{39.17}$$

Notice that it is exact contrast to Wattson-Watt. The only assumption is narrowband.

$$\triangleq [(({}_2 - {}_1) + ({}_3 - {}_1))\, sin(=6)]^2 + [(({}_3 - {}_1) - ({}_2 - {}_1))\, cos(=6)]^2 \tag{39.18}$$

$$= \left[\frac{4}{\ } dsin(\ )\frac{1}{2} sin(=3)\right] \tag{39.19}$$

Elevation angle is

$$= \frac{\sqrt{\ }}{\frac{2}{\ } dsin(=3)} \tag{39.20}$$

Those are non-linear expressions and generally linear expressions are more robust to errors.

Advantages:

- Azimuth and elevation angles are found
- No approximation is involved as in Wattson-Watt
- Good DOA accuracy

Disadvantages:

- Same as other algorithms including super-resolution
- Works for only one source

All classical methods work for one source case. Interferometer, Wattson-Watt and Pseudo-Doppler fails in other cases (multipath). But super-resolution works. But commercial systems use classical approaches generally.

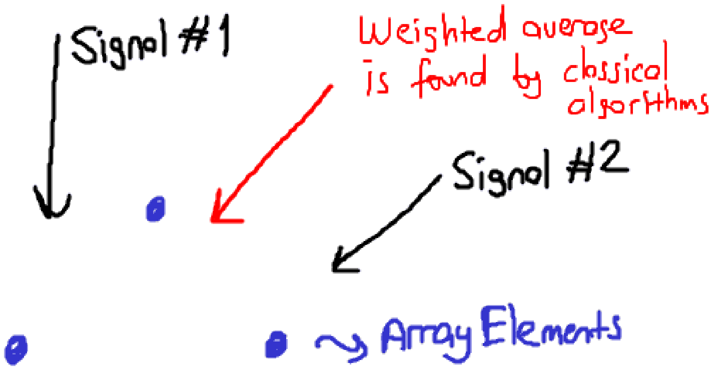

Figure 39.3. Multipath Case

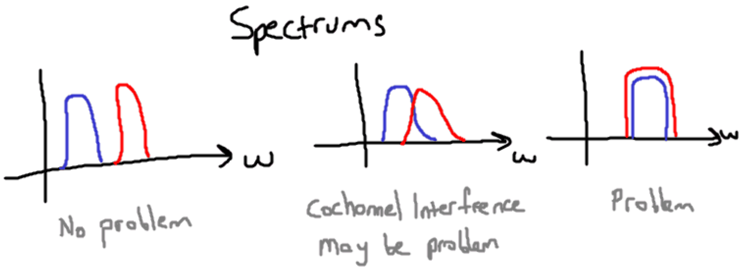

Figure 39.4. Different Sources with Different Spectrum or Time, No Problem

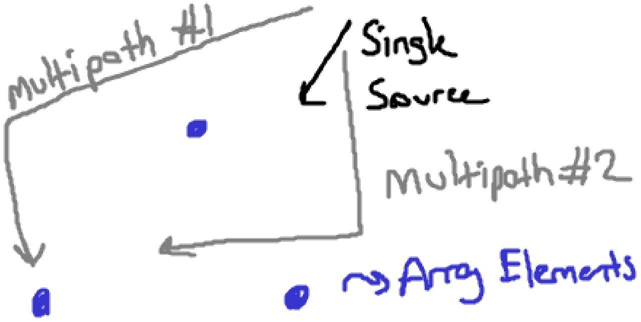

Figure 39.5. Multipath Problem, R is Rank Deficient

40. Optimum and Close to Optimum DOA Estimation

Maximum likelihood is optimum algorithm. MUSIC is also reaches CRB in ideal conditions. Ideal means for example there is no multi-path.

When there is only one source signal there is not much difference between super-resolution and classical techniques. Especially, interferometer is as good as super-resolution techniques at suitable conditions.

When there are more than one source signals, the subspace (super-resolution) methods perform significantly better. Classical methods may fail completely. They can identify and perform beyond Rayleigh limit. Therefore, they are called super-resolution methods.

There are several methods for DOA estimation based on subspace techniques (noise and signal subspace as mentioned which are orthogonal).

- MUSIC (Very Good, Not Only for DOA)
- ESPRIT (Very Good, Not Only for DOA)
- Min-Norm (Load Efficient, Inferior)
- Maximum Likelihood (The Best, Heavy Load and Convergence Problem)

10:40

41. MUSIC (Multiple Signal Classification)

One of the most powerful methods in DOA estimation. There are two versions of algorithm.

- Spectral MUSIC: can be applied to any sensor geometry, computationally intense.
- Root MUSIC: valid for only linear array, fast algorithm.

$$y(t) = As(t) + e(t) \tag{41.1}$$

$$R_y = E\{y(t)y(t)^H\} = AR_sA^H + R_e = AR_sA^H + \ {}^2 I \tag{41.2}$$

$$\hat{R}_y = \frac{1}{N}\sum_{t=1}^{N} y(t)y(t)^H \tag{41.3}$$

(41.3) is called sample covariance matrix. It is best estimator under AWGN. After SVD

$$R_y = V\Lambda V^H \tag{41.4}$$

where

$$\Lambda = \begin{bmatrix} {}_1 + \ {}^2 & & & & & \\ & \ddots & & & & \\ & & {}_n + \ {}^2 & & & \\ & & & {}^2 & & \\ & & & & \ddots & \\ & & & & & {}^2 \end{bmatrix} \tag{41.5}$$

and

$$V = \begin{bmatrix} v_1 & v_2 & ::: & v_n & v_{n+1} & ::: & v_M \end{bmatrix} \tag{41.6}$$

n is the number of sources.

$$rank(AR_sA^H) = n \text{ if } M > n \text{ noise free case (???)} \tag{41.7}$$

$$S_{M \times N} \triangleq \begin{bmatrix} v_1 & v_2 & ::: & v_n \end{bmatrix} \tag{41.8}$$

where S in (41.8) contains eigenvectors of signal subspace vectors. Similarly,

$$G_{M \times (M-N)} \triangleq \begin{bmatrix} v_{n+1} & v_{n+2} & ::: & v_M \end{bmatrix} \tag{41.9}$$

where G in (41.9) contains eigenvectors of noise subspace vectors.
Note that vectors in (41.8) and (41.9) are orthogonal to each other.

$$R_yG = AR_sA^HG + \ {}^2G = \ {}^2G \tag{41.10}$$

Note that AR_s is a full-column rank matrix. A represents the signal subspace. True DOA angles $\{\ _k\}_{k=1}^n$ are the only solutions of the equation.

$$a^H(\ _k)GG^Ha(\ _k) = 0 \text{ for any } M > n \tag{41.11}$$

Note that GG^H is orthogonal projector on $R(G)$ which is noise space.

$$p(\ ) \triangleq \frac{1}{a^H(\ )GG^Ha(\ )} \tag{41.12}$$

(41.12) is called as MUSIC pseudo spectrum.

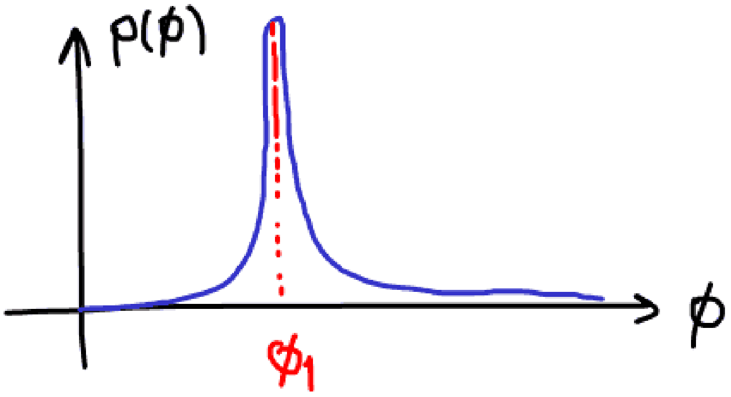

Figure 41.1. MUSIC Pseudo Spectrum

42. MUSIC Algorithm

Compute the sample covariance matrix

$$\hat{R}_y = \frac{1}{N}\sum_{t=1}^{N} y(t)y(t)^H \tag{42.1}$$

Find

$$\hat{G} = \begin{bmatrix} v_{n+1} & v_{n+2} & : : : & v_M \end{bmatrix} \tag{42.2}$$

42.1. Spectral MUSIC.

Find the DOA angle estimates as the locations of the n-highest peaks of the function

$$p(\;) = \frac{1}{a^H(\;)\hat{G}\hat{G}^H a(\;)} \tag{42.3}$$

where $\in [- \;;\;]$
This algorithm requires search.

42.2. Root MUSIC.

Let

$$a(z) = \begin{bmatrix} 1 & z^{-1} & : : : & z^{-(M-1)} \end{bmatrix} \tag{42.4}$$

for ULA.

Consider the equation $a^T(z^{-1})\hat{G}\hat{G}^H a(z) = 0$ polynomial roots. Angular positions if the n-roots (inside the unit circle) which are closest to the unit circle are the DOA angles. It is a fast algorithm. However, it can be applied to only linear array.

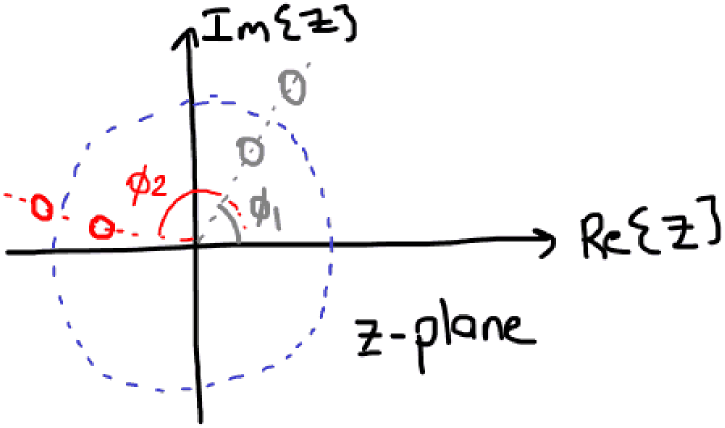

Figure 42.1. z-Plane

For MUSIC algorithm, $n < M$ should be satisfied and sensor position should be known. Performance of MUSIC is good. It is a sub-optimum subspace algorithm.

43. MUSIC Algorithm Application

Let $\ = 60°$, $M = 2$ and $d = \ =2$.

$$a(\ ) = \begin{bmatrix}1\\ e^{j2\pi/\lambda dcos\phi}\end{bmatrix} = \begin{bmatrix}1\\ e^{j\pi/2}\end{bmatrix} = \begin{bmatrix}1\\ j\end{bmatrix} \tag{43.1}$$

$$y(t) = a(\ )s(t) + e(t) \tag{43.2}$$

$$R_y = aa^H\ {}^2_s + {}^2_e I = \begin{bmatrix}1 & -j\\ j & 1\end{bmatrix}\ {}^2_s + {}^2_e I \tag{43.3}$$

Let ${}^2_s = 1$ and ${}^2_e = 0$.

$$R_y = \begin{bmatrix}1 & -j\\ j & 1\end{bmatrix} = \begin{bmatrix}-\sqrt{2}=2 & j\sqrt{2}=2\\ -j\sqrt{2}=2 & \sqrt{2}=2\end{bmatrix}\begin{bmatrix}2 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}-\sqrt{2}=2 & j\sqrt{2}=2\\ -j\sqrt{2}=2 & \sqrt{2}=2\end{bmatrix}^H \tag{43.4}$$

There is one source, then

$$S = \begin{bmatrix}-\sqrt{2}=2\\ -j\sqrt{2}=2\end{bmatrix} \tag{43.5}$$

$$G = \begin{bmatrix}j\sqrt{2}=2\\ \sqrt{2}=2\end{bmatrix} \tag{43.6}$$

11:40

$$p(\ ) = \frac{1}{a^H(\ )GG^H aH(\ )} \tag{43.7}$$

$$GG^H = \begin{bmatrix}0{:}5 & 0{:}5j\\ -0{:}5j & 0{:}5\end{bmatrix} \tag{43.8}$$

$$a(\ ) = \begin{bmatrix}1\\ e^{j2\pi/\lambda dcos\phi}\end{bmatrix} = \begin{bmatrix}1\\ e^{j\pi cos\phi}\end{bmatrix} \tag{43.9}$$

$$Q \triangleq a^H(\ )GG^H aH(\ ) = 1 - sin(\ cos\ ) \tag{43.10}$$

$Q_{min} = 0$ (noise-free case), then $\ = \pm 60°$.

44. Min-Norm Algorithm

Inferior algorithm comparing to MUSIC, seldom used. MUSIC uses (M-n) linearly independent vectors in G. Min=Norm uses a single vector which is a good candidate to represent those vectors in G. By doing so some computational savings is achieved with a certain loss in accuracy.

Let $[1\hat{g}]^T$ be the vector $R(G)$ (range space) with first element equal to 1 that has minimum Euclidean norm.

44.1. **Spectral Min-Norm.** The locations of the n highest peaks in the pseudo-spectrum given in (44.1) corresponds to the DOA angle.

$$\frac{1}{\left|a^{H}(\quad)\begin{bmatrix}1\\ \hat{g}\end{bmatrix}\right|^{2}} \tag{44.1}$$

44.2. **Root Min-Norm.** The angular positions of the n roots of the polynomial given in (44.2) that are closest to the unit circle are DOA angles.

$$a^{T}(z^{-1})\begin{bmatrix}1\\ \hat{g}\end{bmatrix} \tag{44.2}$$

OK But how can I find $\hat{g}$?

$$\hat{s}_{M\times n} = \begin{bmatrix} \quad{}^{H} \\ \bar{s} \end{bmatrix} \tag{44.3}$$

In (44.3), $\hat{s}$ is signal space eigenvectors. $\quad{}^{H}$ is 1xn row vector and $\bar{s}$ is (M-1)xn matrix.

Then,

$$\hat{s}^{H}\begin{bmatrix}1\\ \hat{g}\end{bmatrix} = 0 \tag{44.4}$$

Then,

$$\bar{s}^{H}\hat{g} = - \tag{44.5}$$

$$\hat{g} = -\bar{s}\left(\bar{s}^{H}\bar{s}\right)^{-1} \tag{44.6}$$

(44.6) is LS solution.

$$\hat{s}^{H}\hat{s} = I = \quad{}^{H} + \bar{s}^{H}\bar{s} \tag{44.7}$$

Then,

$$\bar{s}^{H}\bar{s} = I - \quad{}^{H} \tag{44.8}$$

Using (44.8) in (44.6)

$$\hat{g} = -\frac{\bar{s}}{I - \quad{}^{H}} \tag{44.9}$$

Performance loss is significant.

45. Forward-Backward Spatial Smoothing (FBSS)

For one way spatial smoothing $M > 2n$ is required (???). For two way spatial smoothing $M > 1{:}5n$ is required. Two-way reduces the array size.

Multipath os an important error source for DOA estimation. FBSS can be used to solve this problem. FBSS can only be used in linear arrays due to Vandermonde matrix structure.

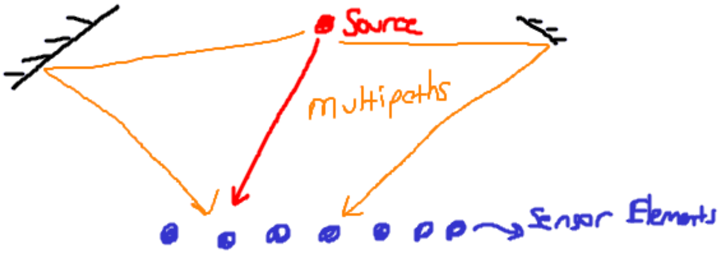

Figure 45.1. Multipath Problem

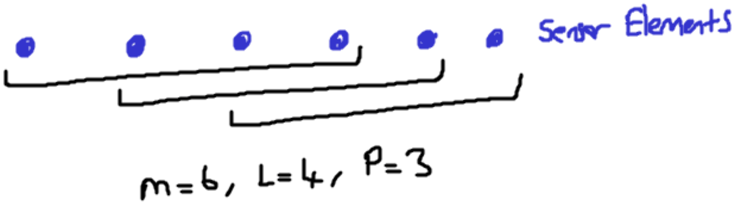

Figure 45.2. Overlapping Sub-Arrays

Obtain

$$\hat{R}_y = \frac{1}{N}\sum_{t=1}^{N} y(t)y(t)^H \tag{45.1}$$

Define

$$J \triangleq \begin{bmatrix} 0 & ::: & 0 & 1 \\ 0 & ::: & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & ::: & 0 \end{bmatrix} \tag{45.2}$$

(45.2) is off-diagonal matrix.
Obtain

$$\tilde{R}_{M\times M} = \frac{1}{2}\left(\hat{R}_y + J\hat{R}_y^H J\right) \tag{45.3}$$

$$R^{fb} = \frac{1}{K}\sum_{k=1}^{K} z_k^T \tilde{R} z_k \tag{45.4}$$

where

$$z_{k_{M\times L}} \triangleq \begin{bmatrix} \bar{0} & I_{L\times L} & \bar{0} \end{bmatrix} \tag{45.5}$$

where identity matrix starts from k^{th} column in (45.5).

Last step is simply summing ‘diagonal matrices‘ of $\tilde{R}$. For example, let

$$\tilde{R} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix} \tag{45.6}$$

Then for example,

$$R^{fb} = \begin{bmatrix} a & b \\ e & f \end{bmatrix} + \begin{bmatrix} f & g \\ j & k \end{bmatrix} + \begin{bmatrix} k & l \\ o & p \end{bmatrix} \tag{45.7}$$

M is number of sensors, L is subarray size and P is the number of subarrays.

$L > n$ and $M \geq \left\lceil \frac{3}{2} n \right\rceil$ should be satisfied. (round or ceil ???) $\lceil\rceil$ is ceiling operator.

For example

n	M	L	P
2	3	3	1
3	5	4	2
4	6	5	2

If n increases but P stays there, you may have problems.

46. What does coherent source mean?

Suppose we have $s_1(t) = s(t)$ and we have also $s_2(t) = \quad e^{j\beta} s(t)$ in narrow-band case. Then,

$$s = \begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix} \tag{46.1}$$

And

$$R_s = \begin{bmatrix} r_{11} & e^{-j\beta} r_{11} \\ e^{j\beta} r_{11} & | \quad |^2 r_{11} \end{bmatrix} \tag{46.2}$$

Matrix in 46.2 is rank deficient. Rank of this matrix is 1 whereas it should be 2. Coherent sources mean fully correlated sources. In that case you can't separate.

For wide-band case $s_1(t) = s(t)$ and we have also $s_2(t) = \quad s(t - \quad)$

Notice,

$$R_s = \begin{bmatrix} 1 & \\ & 1 \end{bmatrix} \tag{46.3}$$

$|\quad| < 1$ should be in (46.3). If it is 0, you have uncorrelated sources and perfect. But if its absolute value is close to 1, you have problems. For fully-correlated case it is 1.

In smoothing, price is that: You start with $R_{M\times M}$ but ends with $R^{fb}_{L\times L}$. $L < M$ is always true. Win rank but loose aperture.

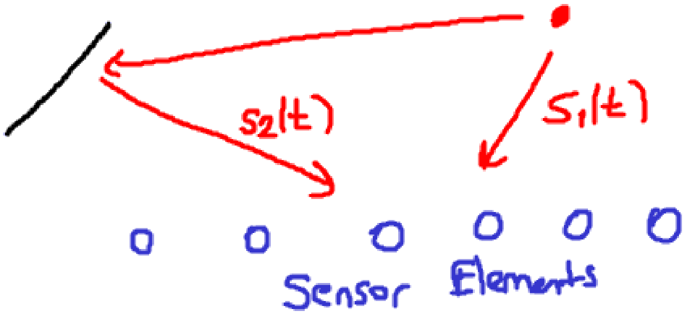

Figure 46.1. Multipath Causes Fully Correlated Source Signals

Part 6. 25/03/14 Lecture Note

47. ESPRIT method

It uses signal space (a subspace algorithm). It can be applied on the only certain geometries. When I move subarray by Δ distance I need to get the second subarray. Therefore, there is a baseline and it should repeat itself for every doublet as shown in 47.1.

It is a powerful method for parameter estimation.

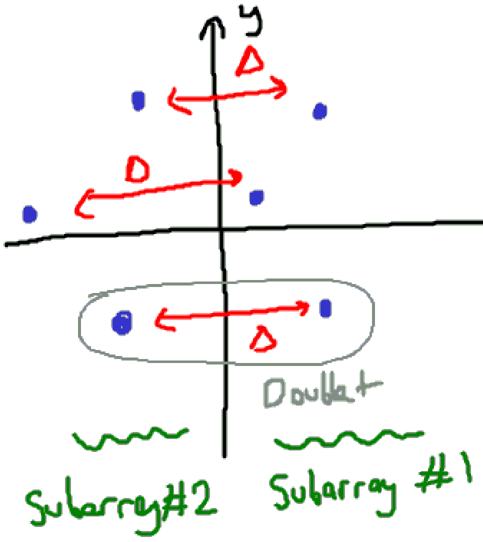

Figure 47.1. ESPRIT Method Array Partitioning

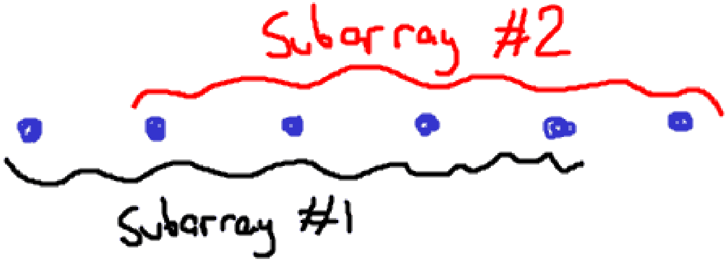

Figure 47.2. Subarrays for ULA

M is the number of sensors. $M_2 = M_1 = M{=}2$ is the number of sensors in the subarrays. $M{=}2 \geq n$ or $M \geq 2n$ where n is number of sources.

Unambiguous DOA estimation is possible if $\Delta < \quad =2$.

$$y(t) = As(t) + e(t) \tag{47.1}$$

Let

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \tag{47.2}$$

where A_1 stands for subarray 1 and A_2 stands for subarray 2. Due to required geometry

$$A_2 = A_1 D \tag{47.3}$$

where

$$D \triangleq diag\{e^{-j\omega_0\Delta cos(\phi_1)/c}; e^{-j\omega_0\Delta cos(\phi_2)/c}; ::::; e^{-j\omega_0\Delta cos(\phi_n)/c}\} \tag{47.4}$$

D has n eigenvalues $\bar{\ }_1; \bar{\ }_2; ::::\bar{\ }_n$. And

$$arg\{\bar{\ }_j\} = -\frac{!_0\Delta cos(\ _j)}{c} \tag{47.5}$$

$_j$ can be found from $\bar{\ }_j$.

Let the signal subspace eigenvectors organized as follows

$$S = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} \tag{47.6}$$

where S_1 stands for subarray 1 and S_2 stands for subarray 2.

Assume

$$R = AR_sA^H + \ ^2I \tag{47.7}$$

Then,

$$RS = S \begin{bmatrix} _1 & & & \\ & _2 & & \\ & & \ddots & \\ & & & _n \end{bmatrix} = AR_sA^HS + \ _2S \tag{47.8}$$

Then,

$$S = AR_sA^HS\bar{\Lambda}^{-1} \tag{47.9}$$

where

$$\bar{\Lambda} \triangleq \begin{bmatrix} _1 - \ ^2 & & & \\ & _2 - \ ^2 & & \\ & & \ddots & \\ & & & _n - \ ^2 \end{bmatrix} \tag{47.10}$$

Define

$$c \triangleq R_sA^HS\bar{\Lambda}^{-1} \tag{47.11}$$

Then,

$$\begin{bmatrix} S_1 \\ S_2 \end{bmatrix} = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} c \tag{47.12}$$

$$S_2 = A_2C = A_1DC = S_1C^{-1}DC = S_1\Phi \tag{47.13}$$

where

$$\Phi \triangleq 1C^{-1}DC \tag{47.14}$$

In (47.13), C is full-rank square matrix then its inverse is exist. D is a diagonal matrix. $C^{-1}DC$ is a similarity transformation over D.

Note that eigenvalues of Φ are same as eigenvalues of D.

A_1, A_2, S_1 and S_2 are full-column rank.

From (47.13)

$$\Phi = (S_1^H S_1)^{-1} S_1^H S_2 \tag{47.15}$$

Φ is related with D over a similarity transformation and Φ and D have same eigenvalues.

ESPRIT estimates for $\{\Phi_k\}_{k=1}^N$ are obtained from $-arg\{\bar{\ }_k\}$ where $\bar{\ }_k\}$ are the eigenvalues of

$$\Phi_{n\times n} = (S_1^H S_1)^{-1} S_1^H S_2 \tag{47.16}$$

Advantages:

- No need to know steering matrix or sensor positions except a doublet
- Sensors in the array do not need to be match except the ones in the doublets
- DOA angle is estimated without search. It is a fast algorithm.
- The solution returns only number of sources not more or less. We assume that we know number of sources.

Disadvantages:

- ESPRIT computes DOA angles using a base line. Hence, it has 180° ambiguity.
- $\Delta <$ =2 to avoid spatial aliasing.
- When the array is placed in a platform, doublets may not be affected similarly.
- ESPRIT uses the information in sub-arrays. Therefore it is not as effective as MUSIC.

48. Maximum Likelihood Methods

Previous methods are not optimum. ML methods are optimum but not computationally efficient, they are usually search based.

Coherent Signals: Two signals are coherent if one is scaled and delayed version of the other. Sub-space methods can not resolve coherent signals except FBSS with ULA. But ML algorithms solves.

Consistency: An estimate is consistent if it converges to the true value when the number of data tends to infinity.

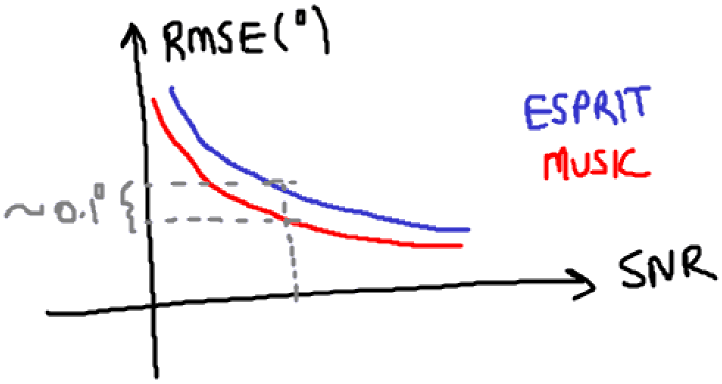

Figure 47.3. ESPRIT vs MUSIC in General

Statistical Efficiency: An estimator is statistically efficient if it asymptotically attains the CRB which is a lower bound on the covariance matrix of any unbiased estimator. The difference between consistency is that even if data is not infinity statistical efficient algorithm tends to close CRB. For example, MUSIC is statistical efficient is under ideal conditions (AWGN, $n < M$, well-known sensor positions etc...).

Subspace methods (MUSIC, ESPRIT) are sub-optimum. ML algorithms are optimum. They can solve coherent signals.

Disadvantages of ML algorithms are computational expense and local minima problem. Two ML algorithms exist: Deterministic and Stochastic ML algorithm.

10:40

49. Deterministic ML Method (DML)

Noise is modelled as stationary AWGN random process. It is also spatially white and circularly symmetric. The signals are deterministic(discard statistical information even if exists) and unknown.

$$y(t) = As(t) + e(t) \tag{49.1}$$

Complex random process is circularly symmetric:

$$E\{x[n_1]x[n_0]\} = 0 \tag{49.2}$$

This implies that

$$R_{x_r}[n_1; n_0] = R_{x_i}[n_1; n_0] \tag{49.3}$$

Basically

$$x[n] = x_r[n] + jx_i[n] \tag{49.4}$$

Then

$$R_{x_i x_r}[n_1; n_0] = -R_{x_r x_i}[n_1; n_0] \tag{49.5}$$

What is the significance of circular symmetricity? In order to define covariance matrices for complex case it is required.(???)

Then,

$$E\{e(t_1)e(t_2)^H\} = \ {}^2 I \ \ {}_{t_1,t_2} \tag{49.6}$$

$$E\{e(t_1)e(t_2)^T\} = 0 \tag{49.7}$$

Observation vector $y(t)$ is also circularly symmetric and temporally white Gaussian random process with mean $As(t)$.

Let's write PDF of y.

$$f_y(y) = \frac{1}{(\ \ {}^2)^M} e^{-\frac{||y(t)-As(t)||^2}{{}^2}} \tag{49.8}$$

Then, likelihood function

$$\mathcal{L}_{DML}(\ \ ; s(t);\ \ {}^2) = \prod_{t=1}^{N} (\ \ {}^2)^{-M} e^{-\frac{||y(t)-As(t)||^2}{{}^2}} \tag{49.9}$$

N is the number of observations.

Write negative log-likelihood function ignoring constants normalized by $1{=}N$.

$$l_{DML} = M \log \ {}^2 + \frac{1}{{}^2 N} \sum_{t=1}^{N} ||y(t)-As(t)||^2 \tag{49.10}$$

Note that non-linear least-squares and DML are the same for Gaussian noise.

$$f = \frac{1}{N} \sum_{t=1}^{N} ||y(t)-As(t)||^2 \tag{49.11}$$

(49.11) is non-linear least-squares.

$$\hat{s}(t) = (A^H A)^{-1} A^H y(t) = A^+ y(t) \tag{49.12}$$

(49.12) is LS solution.

Put (49.12) in (49.10) ignoring some terms.

$$\hat{s}(t) = \frac{1}{{}^2 N} \sum_{t=1}^{N} ||y(t) - A(A^H A)^{-1} A^H y(t)||^2 \tag{49.13}$$

$$\hat{s}(t) = \frac{1}{{}^2 N} \sum_{t=1}^{N} ||[I - A(A^H A)^{-1} A^H] y(t)||^2 \tag{49.14}$$

Let's define

$$\Pi_A \triangleq A(A^H A)^{-1} A^H \tag{49.15}$$

is a projection onto signal space.

$$\Pi_A^{\perp} \triangleq I - \Pi_A \tag{49.16}$$

Then open norm expression

$$\hat{s}(t) = \frac{1}{{}^2N}\sum_{t=1}^{N} y(t)^H \Pi_A^T \Pi_A^T y(t) \tag{49.17}$$

Notice that from properties of projection matrices $\Pi_A^T \Pi_A^T = \Pi_A^T$.

$$\hat{s}(t) = \frac{1}{{}^2N}\sum_{t=1}^{N} tr\{y(t)^H \Pi_A^T y(t)\} \tag{49.18}$$

Notice that in (49.18) $y(t)^H \Pi_A^T y(t)$ is a scalar value and trace of a scalar value is itself.

Remember

$$tr(AB) = tr(BA) \tag{49.19}$$

in general. Then,

$$\hat{s}(t) = \frac{1}{{}^2N}\sum_{t=1}^{N} tr\{\Pi_A^T y(t) y(t)^H\} \tag{49.20}$$

Also remember,

$$tr(A) + tr(B) = tr(A + B) \tag{49.21}$$

Then,

$$\hat{s}(t) = \frac{1}{{}_2}tr\{\Pi_A^T \frac{1}{N}\sum_{t=1}^{N} y(t) y(t)^H\} \tag{49.22}$$

Notice that

$$R_y = \frac{1}{N}\sum_{t=1}^{N} y(t) y(t)^H \tag{49.23}$$

Then

$$\hat{s}(t) = \frac{1}{{}_2}tr\{\Pi_A^T R_y\} \tag{49.24}$$

$$\frac{@l_{DML}}{@\ {}^2} = \frac{M}{{}_2} - \frac{1}{{}_4}tr\{\Pi_A^T R_y\} = 0 \tag{49.25}$$

Then,

(49.26)
$$^{2} = \frac{1}{M} tr\{\Pi_A^T R_y\}$$

Finally

(49.27)
$$\hat{}_{DML} = \text{argmin } tr\{\Pi_A^T R_y\}$$

Measurement $y(t)$ are projected onto a model subspace orthogonal to all anticipates signal components and a power measurement like

(49.28)
$$\frac{1}{N}\sum_{t=1}^{N} ||\Pi_A^{\perp} y(t)||^2 = tr\{\Pi_A^{\perp} R\}$$

is evaluated. The power should be smallest when the projector removes all the signal components.

In implementation, you select and generate $\Pi_A^{\perp}$ and search. For n source n-dimensional search is required. steps vary.

50. Stochastic ML Method (SML)

The signal waveforms are modelled as Gaussian random process.

(50.1)
$$E\{s(t_1)s(t_2)^H\} = R_s \quad {}_{t_1,t_2}$$

(50.2)
$$E\{s(t_1)s(t_2)^T\} = 0$$

(50.3)
$$R_y = AR_sA^H + \ {}^{2} I$$

unknowns are $\ , R_s, \ ^2$.

Negative log-likelihood is proportional to (ignoring constants)

(50.4)
$$l = \frac{1}{N}\sum_{t=1}^{N} ||\Pi_A^T y(t)||^2 = tr\{\Pi_A^T R_y\}$$

For fixed minimum with respect to $\ ^2$ and R_s as follows:

(50.5)
$$\hat{}^{2}_{SML}(\ ) = \frac{1}{M-n} tr\{\Pi_A^T R_y\}$$

(50.6)
$$\hat{R}_s(\ ) = A^+(R_y - \ {}^{2}_{SML} I)(A^+)^H$$

(50.7)
$$\hat{}_{SML} = \text{argmin } log|A\hat{R}_sA^H + \ {}^{2}_{SML} I|$$

In (50.7), || is determinant. It is a generalized variance measures the volume of confidence interval for data.

Both algorithm are nearly same for uncorrelated signal and noise case. But difference is observed under correlated case, low SNR, etc. Deterministic ML is easier to compute.

11:40

51. Beamspace Processing and Beamforming

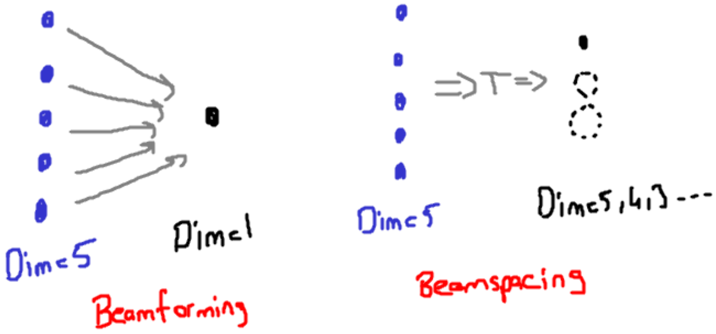

Figure 51.1. Beamforming and Beamspacing

Array perform spatial sampling of the wavefront similar ti temporal sampling. $a(\ )$ characterizes the array as a spatial sampling device. If it is known, array is said to be *calibrated.*

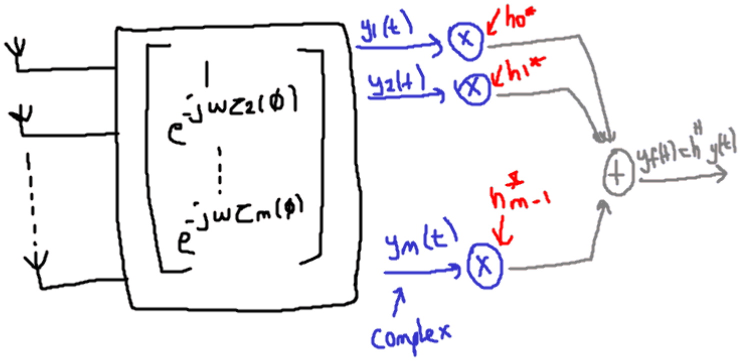

Figure 51.2. Narrowband Beamforming

Spatial filter (coefficient vector) h can be selected to enhance SOI (signal of interest) and suppress interference. SOI and interference may cover the same time-frequency domain. They can be separated in spatial domain.

For h we desired to have

- It passes the SOI undistorted
- It attenuates all the other signal, coming from different directions.

The power of at the beamformer output is found as:

(51.1) $$\text{Output Power at Beamformer} = E\{y_f(t)^2\} = E\{h^H R_y h\}$$

where

$$R_y \triangleq E\{y(t)y(t)^H\} \tag{51.2}$$

$h^H R_y h$ should peak at the DOA of the source signal. This can be used for DOA estimation.

For a single source $n = 1$ DOA estimation by beamforming is consistent. For $n > 1$ it is inconsistent and bias can be large if sources are correlated or closely spaced.

Beamformers can be classified in different ways:

- Narrowband Beamformer
- Wideband Beamformer (Underwater Acoustics)

Also in other way

- Data Independent Beamformer
- Statistically Optimum Beamformer
- Adaptive Beamformer
- Partially Adaptive Beamformer

Also

- Transmit Beamforming
- Receive Beamforming (Our focus, closed for expressions available)

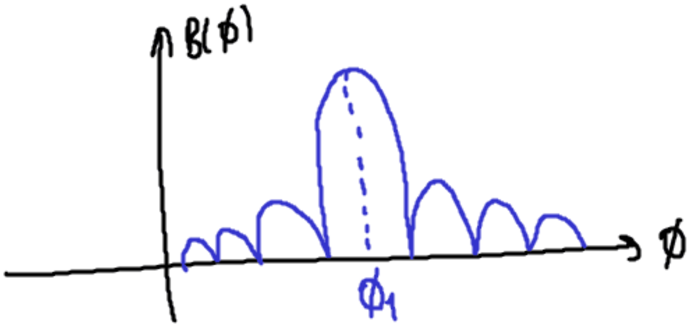

Figure 51.3. General Beamforming Situation

52. Data Independent Beamformer

The weight vector, h, is designed such that the beamformer response approximates a desired response independent of the array data or statistics.

Desired:

$$h^H a(\ ) = 1 \tag{52.1}$$

where is SOI angle.

$$a^H(\ )a(\ ) = P \tag{52.2}$$

by normalization.

Let

$$R_y = I \tag{52.3}$$

then

$$\text{Output Power at Beamformer} = E\{|y_f(t)|^2\} = E\{h^H h\} \tag{52.4}$$

The objective is minimize $h^H h$ over h, subject to $h^H a(\ \ ) = 1$.
Note that,

$$h = \frac{a(\ \ )}{a(\ \ )^H a(\ \ )} = \frac{a(\ \ )}{M} \tag{52.5}$$

where $M\ = P$. Power in this case

$$E\{|y_f(t)|^2\} = \frac{a(\ \ )^H R_y a(\ \ )}{M^2} \tag{52.6}$$

For one signal it is the optimum beamformer.
You can have this from Lagrangian.

$$\mathcal{L} = h^H h + \ \ (1 - h^H a(\ \ )) + \ \ ^*(1 - h^T a^*(\ \ )) \tag{52.7}$$

$$\frac{@\mathcal{L}}{@h} = h - \ \ a(\ \ ) = 0 \tag{52.8}$$

$$h = \ \ a(\ \ ) \tag{52.9}$$

$$h^H a(\ \ ) = 1 \tag{52.10}$$

$$a^H(\ \ )a(\ \ ) = 1 \tag{52.11}$$

$$= \frac{1}{a^H(\ \ )a(\ \ )} = \frac{a(\ \ )}{M} \tag{52.12}$$

In receive case this type of problems may be solved by Lagrangian.
Note that, suppose there is only one signal.

$$y_f(t) = h^H y(t) = h^H a(\ \ )s(t) + h^H p(t) \tag{52.13}$$

Note that $h^H a(\ \ ) = 1$

53. Statistically Optimum Beamformer

We will talk about: MVDR (Minimum Variance Distortionless Beamformer)

Part 7. 01/04/14 Lecture Note

Spatial filter (or beamformer weight vector) is designed based on the statistics of the data received by the array. The goal is the optimize the beamformer response so that the output containing minimum contribution due to noise and signals arriving from directions other that the SOI. But computational cost is a question. We will concentrate on narrow-band case.

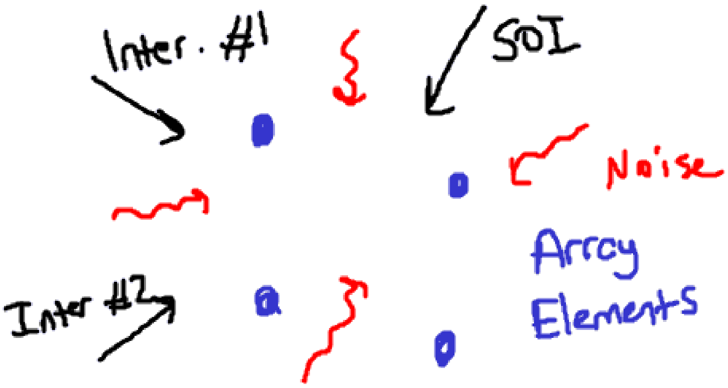

Figure 53.1. An Example Situation

Idea is maximization SINR (Signal To Interference + Noise Ratio). MVDR: Not distort SOI is the optimum beamformer. It requires the signal (R_s) and noise + interference (R_e) covariances. Our signal model is:

$$y(t) = a_0 s_0(t) + a_1 s_1(t) + a_2 s_2(t) + v(t) \tag{53.1}$$

$$i(t) \triangleq a_1 s_1(t) + a_2 s_2(t) \tag{53.2}$$

$$e(t) \triangleq a_1 s_1(t) + a_2 s_2(t) + v(t) \tag{53.3}$$

where $s_0(t)$ is desired signal, $i(t)$ is interference, $e(t)$ is interference + noise and $v(t)$ is white Gaussian noise.

$$R_e = E\{(i(t) + e(t))(i(t) + e(t))^H\} \tag{53.4}$$

$$s_s(t) \triangleq a_0(\quad)s(t) \tag{53.5}$$

$$R_s = E\{s_s(t)s_s(t)^H\} \tag{53.6}$$

Note that

$$1 \geq Rank(R_s) \geq M \tag{53.7}$$

$Rank(R_s) \geq 1$ for incoherently scattered sources or signals with randomly scattering waveforms such as in radar, sonar and wireless communication. As an example in practice

$$R_s = \int_{-\pi}^{\pi} P(\ )a(\ )a(\ )^H d \tag{53.8}$$

This course we usually assume that signals are point sources. Our problem is

$$\frac{h^H R_s h}{h^H R_e h} = SINR \tag{53.9}$$

by playing h where $h^H R_s h^H$ is SOI power and $h^H R_e h^H$ is signal + interference power. We will have two cases:

- $Rank > 1$
  Minimize $h^H R_e h$ over h such that $h^H R_s h = 1$.
- $Rank = 1$
  Minimize $h^H R_e h$ over h such that $h^H a(\ ) = 1$.
  Because $R_s = \ {}^2_s a(\ )a(\ )^H$

53.1. **$Rank > 1$.**

$$\mathcal{L} = h^H R_e h + \ \ (1 - h^H R_s h) \tag{53.10}$$

$$\frac{@\mathcal{L}}{@h^H} = R_e h - \ \ R_s h = 0 \tag{53.11}$$

Then,

$$R_e h = \ \ R_s h \tag{53.12}$$

$$R_e^{-1} R_s h = \frac{1}{\ } h \tag{53.13}$$

(53.13) is a generalized eigenvalue problem.
Then h_{opt} becomes eigenvector corresponding to largest eigenvalue of $R_e^{-1} R_s$.

53.2. **$Rank = 1$.** If $R_s = \ {}^2_s a(\ )a(\ )^H$ then,

$$SINR = \frac{{}^2_s h^H a(\ )a(\ )^H h}{h^H R_e h} = \frac{{}^2_s |h^H a(\ )|^2}{h^H R_e h} \tag{53.14}$$

Then minimize $h^H R_e h$ such that $h^H a(\ ) = 1$ (distortionless response)

$$\mathcal{L} = h^H R_e h + \ \ (1 - h^H a(\ )) + \ {}^*(1 - h^T a^*(\ )) \tag{53.15}$$

$$\frac{@\mathcal{L}}{@h^H} = R_e h - \ \ a(\ ) = 0 \tag{53.16}$$

$$R_e h = \quad a(\quad) \tag{53.17}$$

$$h = \quad R_e^{-1} a(\quad) \tag{53.18}$$

Putting constraint

$$h^H a(\quad) = 1 \tag{53.19}$$

$$a^H(\quad) R_e^{-1} a(\quad) = 1 \tag{53.20}$$

$$= \frac{1}{a^H(\quad) R_e^{-1} a(\quad)} \tag{53.21}$$

$$h_{opt} = \frac{R_e^{-1} a(\quad)}{a^H(\quad) R_e^{-1} a(\quad)} \tag{53.22}$$

You need to know R_e also   .

54. CAPON Beamformer

CAPON beamformer and MVDR turn out to be same for $Rank = 1$ case ($R_s = \ {}_s^2 a(\quad) a^H(\quad)$) and noise and signal is uncorrelated. In general they perform differently.

Consider the typical array model:

$$y(t) = A(\quad) s(t) + e(t) \tag{54.1}$$

CAPON tries to minimize $h^H R_y h$ (array output power) over h such that $h^H a(\quad) = 1$. Suppress power everywhere except SOI.

$$\mathcal{L} = h^H R_y h + \quad (1 - h^H a(\quad)) + \quad ^*(1 - h^T a^*(\quad)) \tag{54.2}$$

$$\frac{@\mathcal{L}}{@h^H} = R_y h - \quad a(\quad) = 0 \tag{54.3}$$

$$R_y h = \quad a(\quad) \tag{54.4}$$

$$h = \quad R_y^{-1} a(\quad) \tag{54.5}$$

Considering constraint

$$h^H a(\quad) = 1 \tag{54.6}$$

$$a^H(\quad) R_y^{-1} a(\quad) = 1 \tag{54.7}$$

$$= \frac{1}{a^H(\ )R_y^{-1}a(\ )} \tag{54.8}$$

$$h_{CAPON} = \frac{R_y^{-1}a(\ )}{a^H(\ )R_y^{-1}a(\ )} \tag{54.9}$$

where R_y is true (not estimate) of covariance matrix if the array output. Different from MVDR R_e is switched by R_y

10:40

55. Proof of the Equivalence Between CAPON and MVDR Beamformer

Assumptions:

$$R_s = \ {}_s^2 a(\ )a(\ )^H \tag{55.1}$$

$$R_y = \ {}_s^2 a(\ )a(\ )^H + R_e \tag{55.2}$$

In (55.1), $Rank = 1$ is assumed at (55.2) uncorrelated source and interference + noise is assumed.

Matrix inversion lemma:

$$[A - CB^{-1}D]^{-1} = A^{-1} + A^{-1}C[B - DA^{-1}C]^{-1}DA^{-1} \tag{55.3}$$

Set $A = R_e$, $B = 1$, $C = a(\ )$ and $D = -a^H(\ )$.

$$R_y^{-1} = R_e^{-1} - R_e^{-1}a[1 + a^H R_e^{-1}a]^{-1}a^H R_e^{-1} \tag{55.4}$$

$$R_y^{-1} = R_e^{-1} - \frac{R_e^{-1}aa^H R_e^{-1}}{1 + a^H R_e^{-1}a} \tag{55.5}$$

$$h = \frac{R_y^{-1}a}{a^H R_y^{-1}a} \tag{55.6}$$

$$h = \frac{R_e^{-1}a(1 - \ )}{a^H R_e^{-1}a(1 - \ )} = \frac{R_e^{-1}a}{a^H R_e^{-1}a} \tag{55.7}$$

where

$$\ \triangleq \frac{a^H R_e^{-1}a}{1 + a^H R_e^{-1}a} \tag{55.8}$$

Then MVDR and CAPON are same under assumptions. Notice that we are talking about R_y which is not estimate, known perfectly.

Beamformer output:

$$y_f(t) = h^H y(t) \tag{55.9}$$

Array output power:

$$E\{|y_f(t)|^2\} = E\{h^H y(t) y^H(t) h\} = h^H R_h \tag{55.10}$$

$$h^H R_h = \frac{a^H R_y^{-1} R_y R_y^{-1} a}{(a^H R_y^{-1} a)^2} = \frac{1}{a^H R_y^{-1} a} \tag{55.11}$$

Then output spectrum of CAPON is:

$$f_{CB} = \frac{1}{a^H(\ \ )R_y^{-1}a(\ \ )} \tag{55.12}$$

56. Linearly Constrained Minimum Variance Beamformer (LCMVB)

CAPON beamformer can also be seen as linearly constrained minimum variance beamformer. In case of multiple constraints minimize $h^H R_y h$ over h such that

$$h^H \left[a(\ _1) \quad a(\ _2)\right] = \left[1 \quad g\right] \tag{56.1}$$

In other words,

$$h^H c_{M\times K} = f^H \tag{56.2}$$

where f is a vector.

$$\mathcal{L} = h^H R_y h + (h^H c - f^H)\ \ _{K\times 1} + (h^T c^* - f^T)\ \ ^* \tag{56.3}$$

$$\frac{@\mathcal{L}}{@h^H} = R_y\ \ + c\ \ = 0 \tag{56.4}$$

$$h = -R_y^{-1} c \tag{56.5}$$

Using the constraint

$$-\ \ ^H c^H R_y^{-1} c = f^H \tag{56.6}$$

$$^H = -f^H (c^H R_y^{-1} c)^{-1} \tag{56.7}$$

$$^H = -(c^H R_y^{-1} c)^{-1} f \tag{56.8}$$

$$h = R_y^{-1} c (c^H R_y^{-1} c)^{-1} f \tag{56.9}$$

Original CAPON expression is similar to 56.9.
Given

$$h = \frac{R_e^{-1} a(\ \ )}{a^H(\ \ ) R_e^{-1} a(\ \ )} \tag{56.10}$$

Notice that (56.10) is MVDR expression. The beamformer output becomes

$$y_f(t) = h^H y(t) = \frac{R_e^{-1}a}{a^H R_e^{-1} a}[as_0(t) + e(t)] \tag{56.11}$$

$$y_f(t) = s_0(t) + \frac{R_e^{-1}a}{a^H R_e^{-1} a} e(t) \tag{56.12}$$

Then,

$$\text{Signal Power} \sim \; {}^{2}_{s_0} \tag{56.13}$$

$$\text{Noise and Interference Power} \sim \frac{a^H R_e^{-1} R_e R_e^{-1} a}{(a^H R_e^{-1} a)^2} = \frac{1}{a^H R_e^{-1} a} \tag{56.14}$$

Then,

$$\text{SINR} = \frac{{}^{2}_{s_0}}{\dfrac{1}{a^H R_e^{-1} a}} = \; {}^{2}_{s_0} a^H R_e^{-1} a \tag{56.15}$$

In practice R_e, R_y and R_e are usually not available. Sample covariance matrix must be used.

$$\hat{R}_y = \frac{1}{N} \sum_{t=1}^{N} y(t) y(t)^H \tag{56.16}$$

(56.16) is best under AWGN case.

The beamformers which use sample covariance are called as *Sample Matrix Inverse (SMI)* beamformers.

$$h_{SMI} = \text{Eigenvectors of } \{\hat{R}_y^{-1} R_s\} \text{ for } Rank > 1 \tag{56.17}$$

$$h_{SMI} = \frac{\hat{R}_y^{-1} a(\;\;)}{a^H(\;\;)\hat{R}_y^{-1} a(\;\;)} \text{ for } Rank = 1 \tag{56.18}$$

This is SMI using CAPON expression, not CAPON. Substantial performance loss is observed.

57. Loaded SMI Beamformer

One of the most popular approach to robust adaptive (sample or block adaptive) beamforming in the presence of array response error and small training sample size is the diagonal loading technique. The idea is to regularize the problem by adding a quadratic penalty term to the objective function.

Minimize $h^H \hat{R} h + \;\; h^H h$ over h such that $h^H R_s h = 1$ where   is penalty weight.

Solution is called as LSMI beamformer.

$$\mathcal{L} = h^H \hat{R} h + \;\; h^H h + \;\; (1 - h^H R_s h) \tag{57.1}$$

$$\frac{@\mathcal{L}}{@h^H} = \hat{R}h + \quad h - \quad R_s h = 0 \tag{57.2}$$

$$(\hat{R} + \quad I)h = \quad R_s h \tag{57.3}$$

$$(\hat{R} + \quad I)^{-1} R_s h = \frac{1}{\ }h \tag{57.4}$$

where h is the eigenvector corresponding to the largest eigenvalue of $(\hat{R} + \quad I)^{-1}R_s$. Similar to MVDR but we have $\hat{R}$ instead of R_e.

For $Rank = 1$ case, $R_s = \quad {}^2_s aa^H$.

$$h = \frac{(\hat{R} + \quad I)^{-1}a(\quad)}{a^H(\quad)(\hat{R} + \quad I)^{-1}a(\quad)} \tag{57.5}$$

(57.5) very similar to MVDR where R_e is switched by $\hat{R} + \quad I$. Performance is good.

What is ?

Usually around 0.1. You can find it in optimum manner.

58. An Equivalent Formulation for CAPON Beamformer

It is called as Constrained Covariance Fitting.

Our goal is to maximize signal power while fitting $\ {}^2_s a(\quad)a(\quad)^H$ to R imposing $R - \quad {}^2_s a(\quad)a(\quad)^H \geq 0$ which is called as positive semi-definite matrix where eigenvalues are not negative and eigenvectors are orthogonal.

New formulation is maximize $\ {}^2_s$ such that $R - \quad {}^2_s a(\quad)a(\quad)^H \geq 0$. It is same as CAPON beamformer expression which is minimize $h^H R h$ over h such that $h^H a(\quad) = 1$. We will show that they are equivalent.

11:40

$$h^H[R - \quad {}^2_s a(\quad)a(\quad)^H]h \geq 0 \text{ for any } h \in C^{M\times 1} \tag{58.1}$$

Positive semidefinetness is used in (58.1). Rewrite using a constraint.

$$h^H[R - \quad {}^2_s a(\quad)a(\quad)^H]h \geq 0 \text{ such that } h^H a(\quad) = 1 \tag{58.2}$$

Net result

$$h^H R h \geq \quad {}^2_s \text{ such that } h^H a(\quad) = 1 \tag{58.3}$$

Finally,

$$\ {}^2_s = \min h^H R h \text{ such that } h^H a(\quad) = 1 \tag{58.4}$$

$$R - \quad {}^2_s a^H a \geq 0 \tag{58.5}$$

By multiplying 58.5 $R^{-1/2}$ both left and right side, (58.6) $$I - \ {}^2_s R^{-1/2} a^H a R^{-1/2} \geq 0$$

Note that $Rank(R^{-1/2} a^H a R^{-1/2}) = 1$ it has one non-zero eigenvalue. $I-$ that matrix is $1-$ non-zero eigenvalue. Since $Trace\{\}$ of a matrix is sum of eigenvalues (in general ?), using $Trace(AB) = Trace(BA)$ then

(58.7) $$1 - \ {}^2_s a^H R^{-1} a \geq 0$$

Then,

(58.8) $${}^2_s \leq \frac{1}{a^H R^{-1} a}$$

$1{=}a^H R^{-1} a$ is the output signal power hence it is the CAPON estimate of the signal power. (58.8) is equivalent to (58.1)

59. Robust CAPON Beamformer with Single Constraint

CAPON beamformer performs very well when the steering vector $a(\ )$ is known accurately and R_y (theoretical) is used, However, if $a(\ )$ knowledge is imprecise its performance is not good.

Robust CAPON is extension of CAPON method that assumes $a(\ )$ belongs to uncertainty ellipsoid.

(59.1) $$(a - \bar{a})^* C^{-1} (a - \bar{a}) \leq$$

where $\bar{a}$ is known(measured) steering vector. C is positive definite error covariance.

When there is a little information about C, $C = \ I$ chosen and 59.1 becomes as follows

(59.2) $$||a - \bar{a}||^2 <$$

59.2 is error ball.

The choice of does not change the performance much.

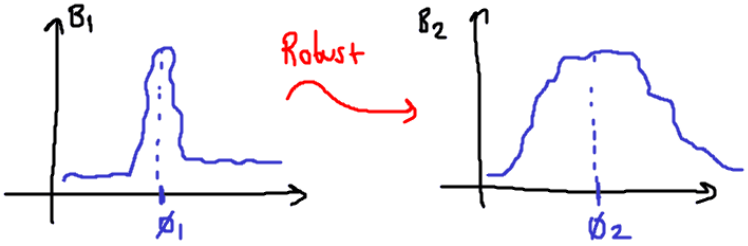

Figure 59.1. Robust Design Example

The robust CAPON beamformer (RCB) solves the following problem:
Maximize ${}^2_s$ over a and ${}^2_s$ subject to $R - \ {}^2_s a a^H \geq 0$ and $(a - \bar{a})^H C^{-1} (a - \bar{a}) \leq 1$.
To avoid from trivial solution $a^H a = M$ is assumed.

For a fixed a (if you know it) maximizing ^2_s is equivalent to

$$\tilde{}^2_s = \frac{1}{a^H R^{-1} a} \tag{59.3}$$

Then the problem can be simplified as minimize $a^H R^{-1} a$ over a such that $||a - \bar{a}||^2 \leq$ by assuming $C = \ I$. This is RCB problem.

The steps of RCB algorithm:

First compute eigen decomposition.

$$R = U \Lambda U^H \tag{59.4}$$

and set

$$b = U^H \bar{a} \tag{59.5}$$

where

$$\Lambda \triangleq \begin{bmatrix} _1 & & \\ & \ddots & \\ & & _M \end{bmatrix} \tag{59.6}$$

Solve

$$\sum_{k=1}^{M} \frac{|b_k|^2}{(1 + \ \ _k)^2} = \tag{59.7}$$

where in (59.7) is our . Solve it using Newton Method assuming the solution is in $[\ _L; \ _u]$.

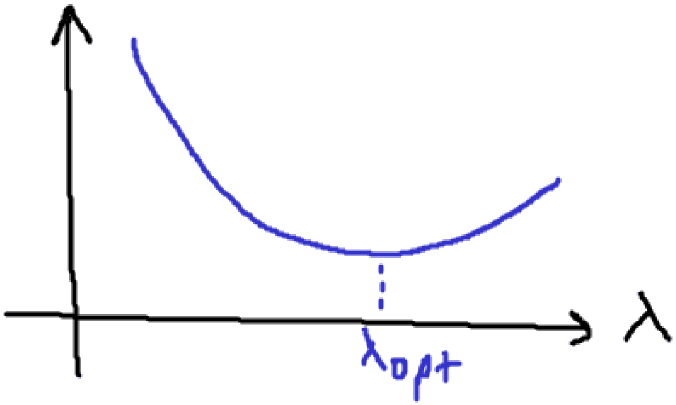

Figure 59.2. Example Plot for Equation (59.7)

Then compute

$$\tilde{a} = \bar{a} - U(I + \Lambda\ \ )^{-1} b = (I + \frac{1}{\ }R^{-1})\bar{a} \tag{59.8}$$

$$h_{RCB} = \frac{\left(R + \frac{1}{\ }I\right)^{-1} \bar{a}}{\bar{a}^H \left[\left(R + \frac{1}{\ }I\right) R^{-1} \left(R + \frac{1}{\ }I\right)\right]^{-1} \bar{a}} \tag{59.9}$$

Compare with LSMI.

$$h = \frac{(R + \ \ I)^{-1}\bar{a}}{\bar{a}^H (R + \ \ I)^{-1}\bar{a}} \tag{59.10}$$

They are very similar. In fact if you use $\ = 1= \ $, it will be a very good beamformer.

In practice, RCB is the best. MVDR is a benchmark for us.

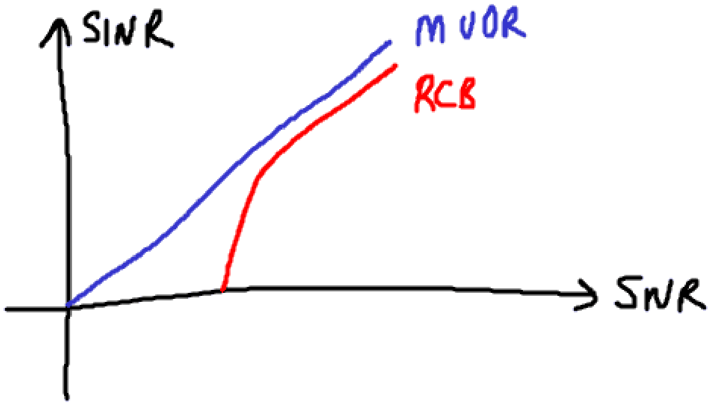

Figure 59.3. Typical Performance Curve

Part 8. 08/04/14 Lecture Note

60. Generalized Sidelobe Canceller (GSC)

Until now we saw block based adaptive algorithms. But GSC is sample based adaptive. It is practical.

GSC is an alternative form of LCMV beamformer (CAPON).

Minimize $h^H R_y h$ such that $h^H c = f^H$.

It is the implementation of LCMV.

Let h_0 be the optimum beamformer given as (previously known)

$$h_0 = R_y^{-1} c (c^H R_y^{-1} c)^{-1} f \tag{60.1}$$

h_0 is decomposed into two orthogonal components.

$$h_0 = h_c - h_p \tag{60.2}$$

h_c is defined to be the projection of h_0 onto the constraint subspace.

h_p is defined to be the projection of h_0 onto the subspace which is orthogonal to constraint subspace. This space is represented by B matrix.

$$C^H_{M\times K} B_{M\times K} = 0 \tag{60.3}$$

The projection onto the constraint subspace.

$$P_c = C(C^H C)^{-1} C^H \tag{60.4}$$

And

$$h_c = P_c h_0 \tag{60.5}$$

$$h_c = C(C^H C)^{-1} f \tag{60.6}$$

This beamformer has closed form expression which is fixed.

$$h_p = -B(B^H B)^{-1} B^H h_0 = -P_c^{\perp} h_0 \tag{60.7}$$

where

$$P_c^{\perp} \triangleq I - P_c \tag{60.8}$$

$$h_p = -B(B^H B)^{-1} R_y^{-1} C (C^H R_y^{-1} C)^{-1} f \tag{60.9}$$

The constraint

$$h_0^H C = f^H \tag{60.10}$$

$$(h_c^H - h_p^H) C = h_c^H C - h_p^H C \tag{60.11}$$

(60.12)
$$(h_c^H - h_p^H)C = f^H(C^HC)^{-1}C^HC + f^H(C^HR_y^{-1}C)^{-1}C^HR_y^{-1}(B^HB)^{-1}B^HC$$

Notice at second term $B^HC = 0$.

$$(h_c^H - h_p^H)C = f^H \tag{60.13}$$

Therefore constraint is satisfied. Since the constraint is satisfied, GSC converts the constrained problem to an unconstrained one.

Let

$$h_p = Bh_a \tag{60.14}$$

Minimize $(h_c - Bh_a)^HR_y(h_c - Bh_a)$ over h_a.

Take the derivative with respect to h_a and equate to 0.

$$\hat{h}_a = (B^HR_yB)^{-1}B^HR_yh_c \tag{60.15}$$

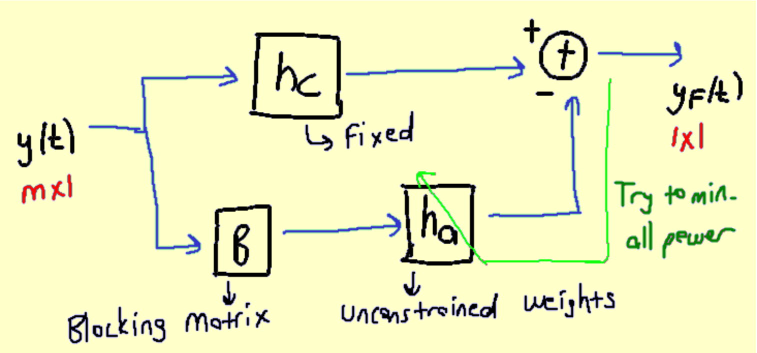

Figure 60.1. Implementation of GSC

Constrained problem is converted to unconstrained one. A data independent beamformer, h_c is used. h_a is unconstrained and adaptive algorithms can be used for it.

B is not unique. One method of finding B is

$$P_c^{\perp} = I - C(C^HC)^{-1}C^H \tag{60.16}$$

Orthogonalize $P_c^{\perp}$ by Gram-Schmidt algorithm and choose the first $(M - K)$ columns as the B matrix.

In general, if the constraints are designed to present a specific response to signals for certain directions and frequencies then columns of B will block these directions and frequencies. Since h_c processes these according to constraints desired response is achieved.

Example

Let $c = a(\ \ )$ a single constraint and $h^Ha(\ \ ) = 1$

$$h_c = \frac{a(\ )}{a(\ )^H a(\ )} \tag{60.17}$$

$$a(\ )^H B = 0 \tag{60.18}$$

Each column of B can be seen as data independent beamformer with a null direction .

61. Beamspace Processing

The main advantage of beamspace processing is to reduce data and increase computational efficiency.

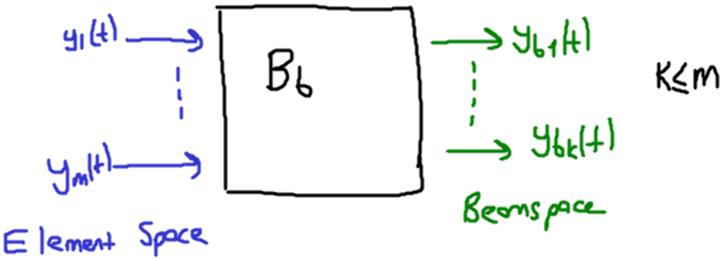

Figure 61.1. Beamspace Processing

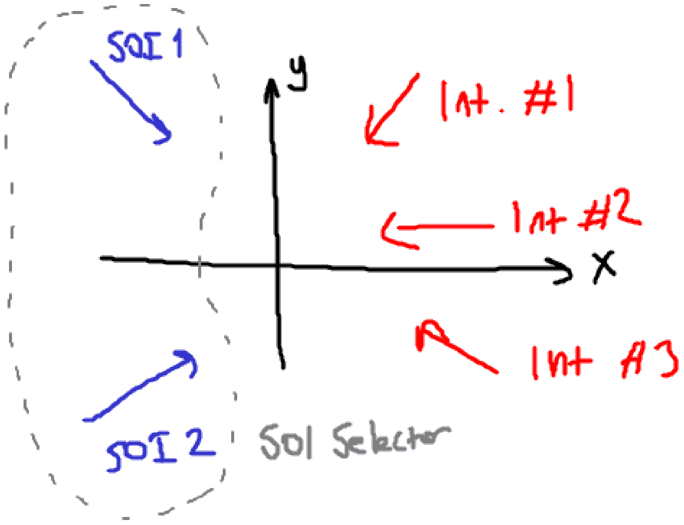

Figure 61.2. SOI Selector

The variance of DOA estimates in beamspace is usually larger than the element space. If you are estimate DOA it is generally better to it in element space.

Increase in probability of resolution is possible if there is some information regarding the sector where the SOI lie. If two signal are close to each other you may probably separate them in beam space easily.

10:40

(61.1)
$$y(t) = As(t) + e(t)$$

(61.2)
$$y_b(t) = B_b^H y(t) = B_b As(t) + B_b e(t)$$

$B_b^H B_b = I$ is desired in order to have white noise assumption valid in beamspace. Hence it has no noise amplification.

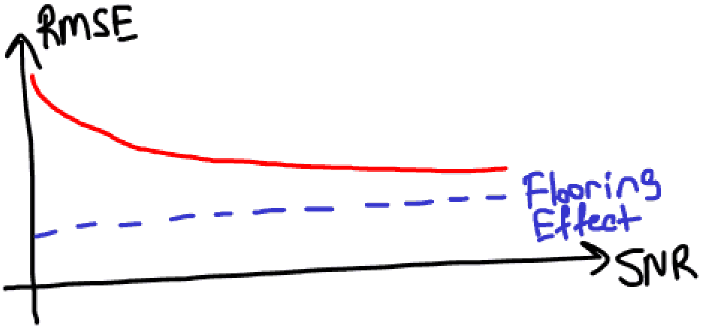

Figure 61.3. Flooring Effect Due to Matrix Multiplication and Coloured Noise

Let A_b represents a beamspace matrix whose columns consist of a set of steering vectors representing the sector of interest.

Taking the QR decomposition

(61.3)
$$A_b = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 & 0 \end{bmatrix}$$

(61.4)
$$B_b = Q_1$$

(61.5)
$$y_b(t) = B_b^H y(t)$$

(61.6)
$$R_{y_b} = B_b^H R_y B_b = \begin{bmatrix} \hat{u}_1 & \hat{u}_2 \end{bmatrix} \begin{bmatrix} \hat{}_1 & \\ & \hat{}_2 \end{bmatrix} \begin{bmatrix} \hat{u}_1 & \hat{u}_2 \end{bmatrix}^H$$

62. Beamspace MUSIC

(62.1)
$$p_{(}\ ) = \frac{1}{a^H(\ )B_b\hat{u}_2\hat{u}_2^H B_b^H a(\ )}$$

Note that $a(\ )$ is still generated in element space.

63. Beamspace Beamforming

LCMV (CAPON):

$$h_b = \frac{R_{y_b}^{-1} B_b^H a(\ )}{a(\ )^H B_b R_{y_b}^{-1} B_b^H a(\ )} \tag{63.1}$$

For multiple constraints in the element space

$$h^H c = f^H \tag{63.2}$$

$$h = B_b h_b \tag{63.3}$$

$$h_b^H (B_b^H C) = f^H \tag{63.4}$$

$$C_b = B_b^H C \tag{63.5}$$

$$h_b^H C_b = f^H \tag{63.6}$$

$$h_b = R_{y_b}^{-1} C_b (C_b^H R_{y_b}^{-1} C_b)^{-1} f \tag{63.7}$$

64. Adaptive Algorithms for Beamforming

Two basic approaches
Block Adaptation

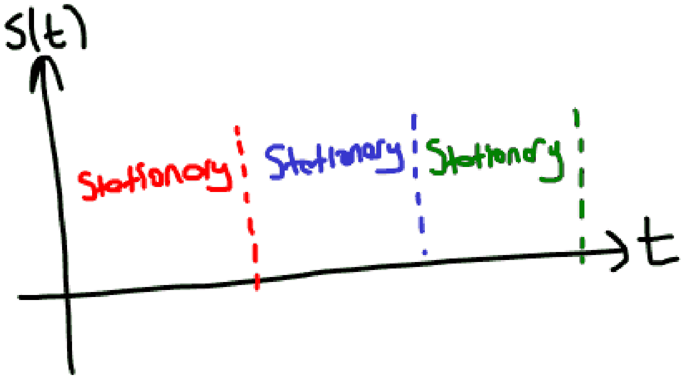

Figure 64.1. Block Adaptation

Statistics are estimated from a temporal block of array data and is used an optimum weight equation.

Continuous Adaptation Weight vector is adjusted as data is sampled such that the resulting weight vector sequence converges to the optimum solution.

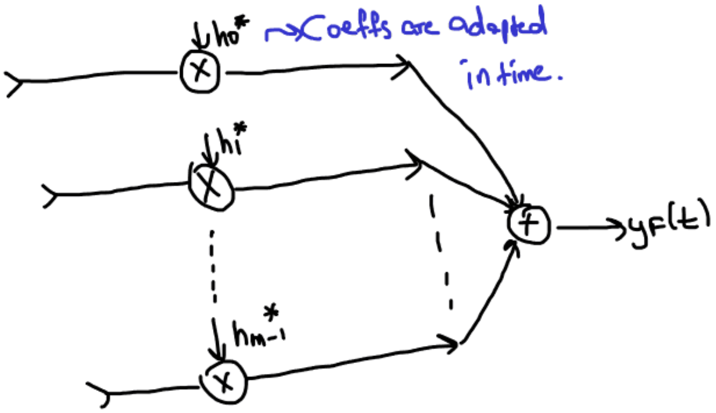

Figure 64.2. General Adaptive Structure

Consider

$$y(t) = a(\quad)s(t) + v(t) + e(t) \tag{64.1}$$

where $s(t)$ is SOI $v(t)$ is noise and $e(t)$ is interference.

$$SINR = \frac{h^H a R_s a^H h}{h^H R_{v+e} h}(???) \tag{64.2}$$

$$R_{v+e} \triangleq E\{(v(t)+e(t))(v(t)+e(t))^*\} \tag{64.3}$$

We know that h_{op} is the eigenvector corresponding to maximum eigenvalue of $R_{v+e}^{-1} a R_s a^H$ for $rank > 1$ case. And $\dfrac{R_{v+e}^{-1} a}{a^H R_{v+e}^{-1} a}$ for $rank = 1$ case.

R_{v+e} is usually replaced with the training data covariance matrix which is

$$R_y = E\{y(t)y(t)^H\} \tag{64.4}$$

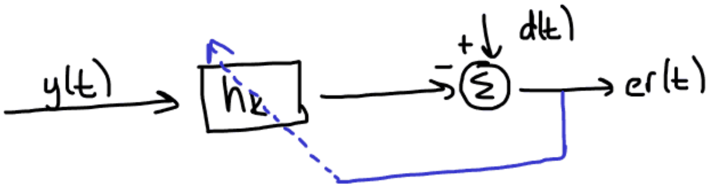

Figure 64.3. LMS Algorithm Implementation

$$y(t) = h(t-1) + \quad y(t)er^*(t) \tag{64.5}$$

Convergence depends on the eigenvalue spread. If the spread is large convergence is slow.

65. Sidelobe Canceller

A simple structure of an adaptive canceller. It is shown in Figure 65.1.

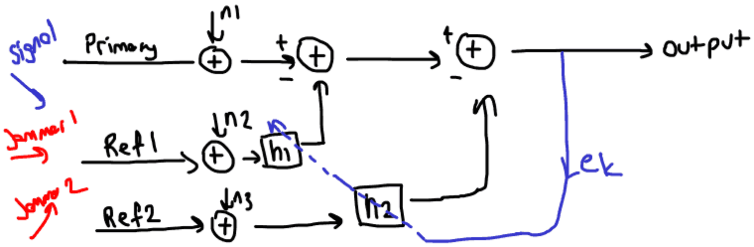

Figure 65.1. Sidelobe Canceller

The main assumption is that the jammer signals are much stronger than the SOI. Therefore adaptive weights are mostly controlled by the interference. Output contains interference component close to primary.

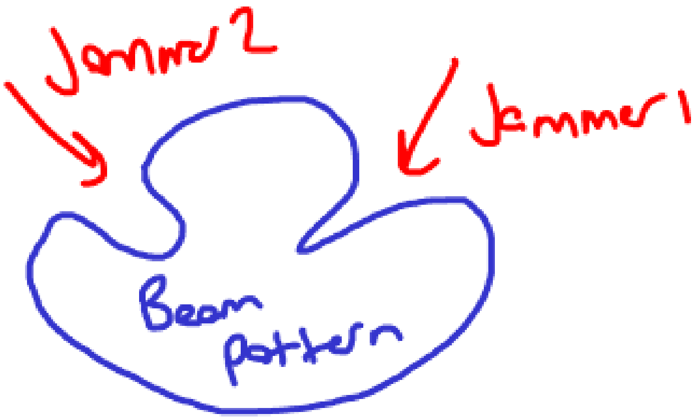

Figure 65.2. Jammer Signal Suppression with Sidelobe Canceller

11:40

66. Beamforming with a Pilot Signal

66.1. **Two-mode Case.** Delay $_k$ is adjusted to simulate a signal coming from a certain direction. In P mode, input is due to pilot signal. Desired signal is also the the pilot. A beam is formed towards direction. In mode A, input is due to sensors. Adaptation is done to eliminate all received signals since desired signal is zero. If we continue in mode A, weights send to zero. Therefore, frequent switching between mode P and mode A should be done.

Note that as shown in from Figure 66.1, it uses delay terms. Any beamformer uses delay terms is wideband beamformer (?).

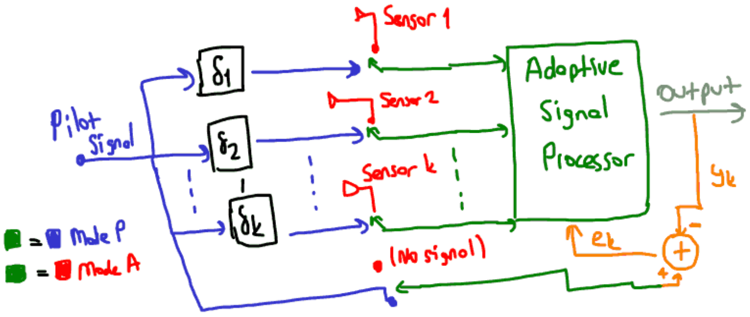

Figure 66.1. Two-mode Case

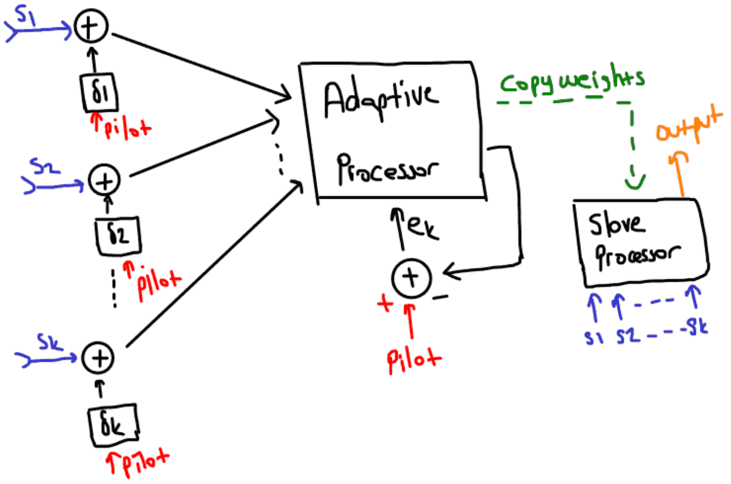

Figure 66.2. Single Mode Case

66.2. **Single Mode Case.** Signal reception on P-mode is possible. Adaptation tries to reproduce pilot and eliminate all signals coming from the sensors. Therefore a beam is formed toward direction where pilot is pointing.

As a note, GSC (Generalized Sidelobe Canceller) is developed after them and it is the current state of the art. But roboust beamformers may be better than GSC. However, GSC performs very well. These were usually in early analog systems. You can do much better them.

67. Narrowband and Wideband Beamforming

67.1. **Narrowband Beamforming.** Previous discussions were about the narrowband beamformers. The common structure as discussed was this

67.2. **Wideband Beamforming.** Figure 67.2 is time domain implementation.

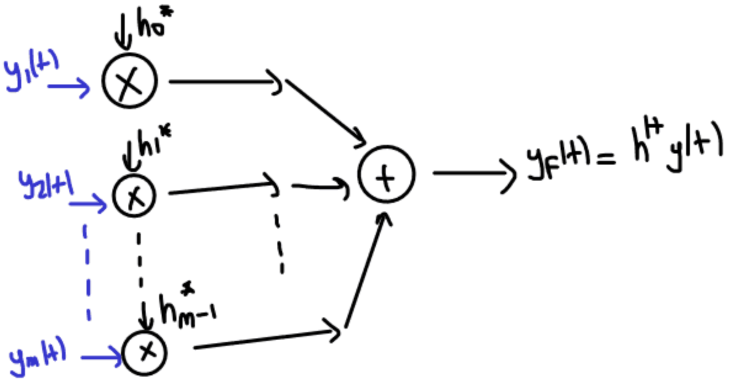

Figure 67.1. General Narrowband Beamformer

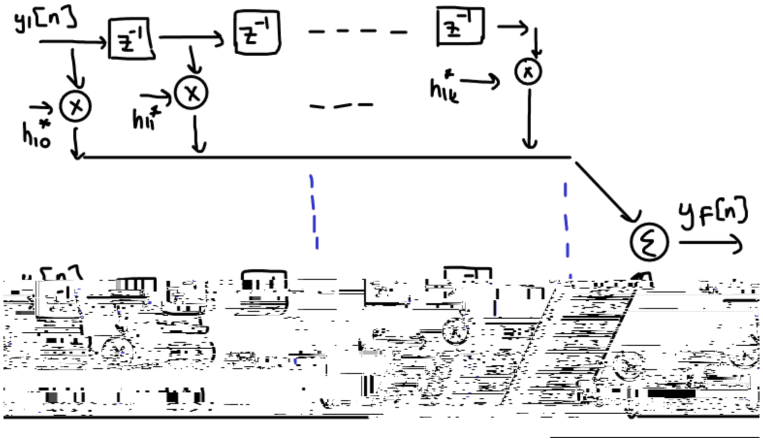

Figure 67.2. General Wideband Beamformer (Filter and Sum)

$$y_F[n] = \sum_{i=1}^{M} \sum_{m=0}^{K-1} h_{i,m}^* y[n-m] \tag{67.1}$$

68. DFT Domain Wideband Beamformer

DFT of each channel signal is taken. Each frequency has its own beamformer. DFT and time-domain (filter and sum) based beamformers can be made equivalent.

In Figure 68.1, P is the DFT size.

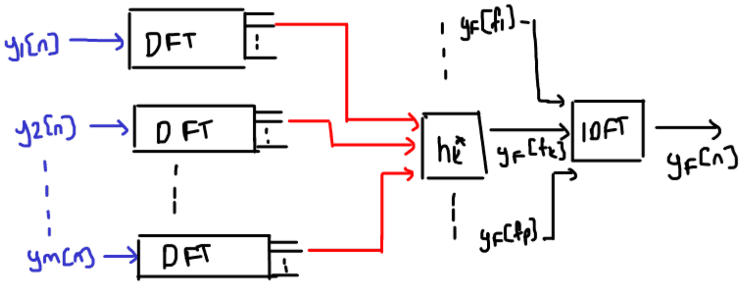

Figure 68.1. DFT Based Wideband Beamformer

Part 9. 15/04/14 Lecture Note

69. Delay and Sum Beamformer

Consider the k^{th} channel.

$$y_k(t) = \bar{h}_k(t) * x(t - {}_k) + \bar{e}_k(t) \qquad k = 1; : : : ; M \tag{69.1}$$

Assume that

$$\bar{h}_k(t) = \ (t) \qquad \forall k \tag{69.2}$$

Then,

$$y_k(t) = x(t - {}_k) + \bar{e}_k(t) \tag{69.3}$$

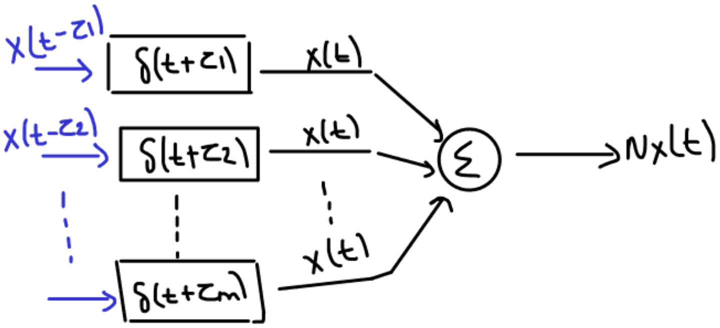

Figure 69.1. Delay and Sum Beamformer

70. Wideband Processing

Signals in practical applications (such as acoustics and some RF signals) are wideband in nature. Signal power varies over the frequency band and it is advantageous to process such signals with wideband techniques.

There are two types of wideband processing methods:

- Coherent Methods Use focusing or mapping matrices in order to to transform covariance matrices and obtain a single covariance matrix. Then known techniques are applied over this covariance matrix. It is a computationally efficient method.

$$\bar{R}_k = T_k R_k \tag{70.1}$$

where in Equation (69.1) $k = 1; : : : ; 128$ for example.

$$\bar{R} = \sum_k \bar{R}_k \tag{70.2}$$

- Non-coherent Methods

  Process each frequently separately and then combine the result. Computationally complex method. It performs as good as non-coherent methods (If not better) with significantly less computation.

70.1. Non-coherent Processing, MUSIC Example.

$$p(\ ) = \sum_{k=1}^{K} \frac{1}{a^H(!_k;\ )G(!_k)G^H(!_k)a(!_k;\ )} \tag{70.3}$$

$G(!_k)$ is found from (70.4)

$$\hat{R}(!_k) = \frac{1}{N}\sum_{p=1}^{N} Y(!_k;p)Y^H(!_k;p) \tag{70.4}$$

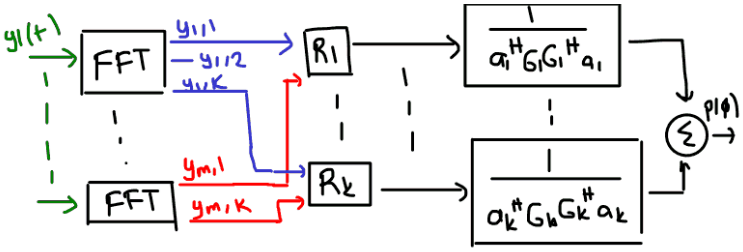

Figure 70.1. Non-coherent MUSIC Example

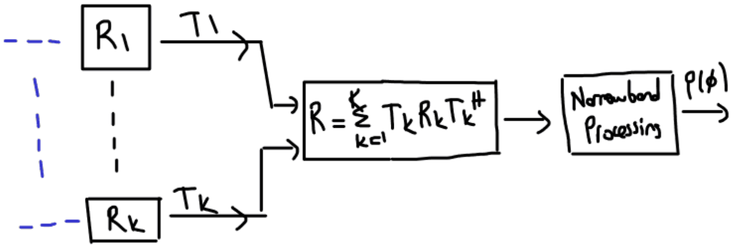

Figure 70.2. Coherent Processing Example

70.2. Coherent Processing Example. In Figure 70.2, T_k is called as focusing (matrix) matrix. It is find by array interpolation. Generally wide processing gain $\leq$ K.

71. Wideband Model

Assume identical sensors, m^{th} sensor signal can be written as

$$y_m(t) = \sum_{k=1}^{n} s_k(t - \ _{mn}) + v_m(t) \qquad m = 1;2;\dots;M \tag{71.1}$$

where M is the number of sensors and n is the number of sources, $s_k(t)$ is k^{th} source signal, $v_m(t)$ is noise for m^{th} sensor, $\ _{mn}$ is delay for the signal propagation from the n^{th} source to the m^{th} sensor. Example case is shown in Figure 71.1.

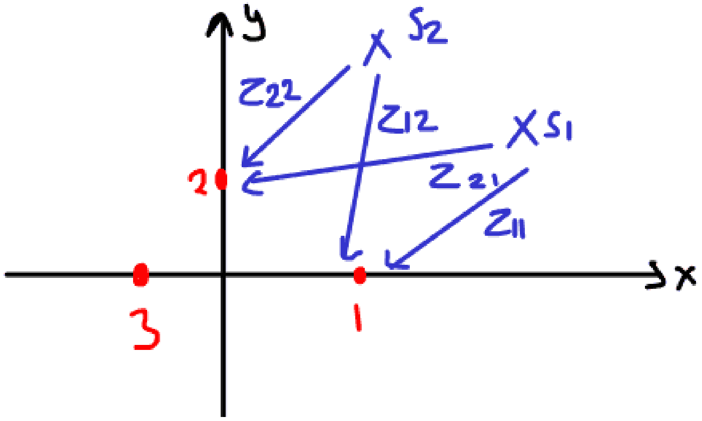

Figure 71.1. Example Wideband Case

Take DFT of 71.1,

$$Y_m(!) = \sum_{k=1}^{n} S_k(!)e^{-j\omega\tau_{mn}} + V_m(!) \qquad m = 1;2;:::;M \tag{71.2}$$

In matrix-vector form

$$Y(!) = A(!)S(!) + V(!) \tag{71.3}$$

where

$$S(!) \triangleq \begin{bmatrix} s_1(!) \\ s_2(!) \\ \vdots \\ s_n(!) \end{bmatrix} \tag{71.4}$$

$$A(!)_{mk} \triangleq e^{-j\omega\tau_{mk}} \tag{71.5}$$

Let $\ _{1k}$ be the time delay for the k^{th} source to the reference sensor 1.

$$\ _{mk} - \ _{1k} = \frac{1}{c}[x_m cos\ _k sin\ _k + y_m sin\ _k sin\ _k + z_m cos\ _k] \tag{71.6}$$

Reference sensor 1 is positioned at $(0;0;0)$.

$$R_s(!) \triangleq E\{S(w)S^H(w)\} \tag{71.7}$$

$$R_v(!) \triangleq E\{V(w)V^H(w)\} \tag{71.8}$$

$$R(!) \triangleq E\{Y(w)Y^H(w)\} = A(!)R_s(!)A^H(!) + R_v(!) \tag{71.9}$$

To estimate

$$\hat{R}(!) = \frac{1}{N_s}\sum_{j=1}^{N_s} Y(w;j)Y^H(w;j) = \frac{1}{N_s}Y(!)Y^H(!) \tag{71.10}$$

72. Virtual Array Processing

Usually the number of sensors is limited and sensors are preferred to be positioned in a large aperture. Therefore gaps between sensors occur. Furthermore conversion from a certain array geometry to another may be required (Ex: UCA to ULA). In virtual array processing, given the sensor signals of a real array sensor signals of a virtual array are obtained. Example situation is shown in Figure 72.1.

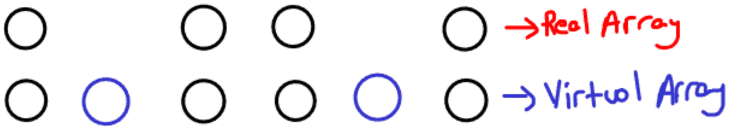

Figure 72.1. Virtual Array Example with ULA

There are different methods for virtual array processing: Array interpolation, manifold separation, HOS (Higher Order Statistics) are some examples.

73. Array Interpolation

73.1. **Co-array.** Co-array is a function which gives the number of times each spatial correlation lag is contained in an array. Consider the non-redundant array shown in Figure 73.1.

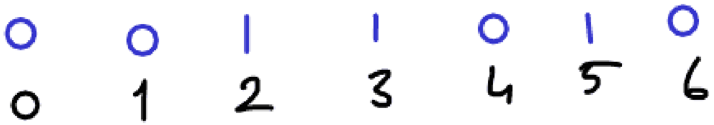

Figure 73.1. Non-redundant Array

Let $h[n]$ denotes the sensor displacement, $h(d_i) = 1$.

$$h = \begin{bmatrix}1 & 1 & 0 & 0 & 1 & 0 & 1\end{bmatrix} \tag{73.1}$$

Co-array is found as

$$c[n] = h[n] * h[-n] \tag{73.2}$$

$$c[n] = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 4 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}^T \tag{73.3}$$

In (73.3) 4 is called as *center point.*
Also,

$$R = \begin{bmatrix} r_{00} & r_{01} & r_{02} & r_{03} & r_{04} \\ r_{10} & r_{11} & r_{12} & r_{13} & r_{14} \end{bmatrix} \tag{73.4}$$

For example,

$$r_{13} = E\{y(1)y^*(3)\} \tag{73.5}$$

Lag is $3 - 1 = 2$.

$$r_{02} = E\{y(0)y^*(2)\} \tag{73.6}$$

Lag is $2 - 0 = 2$.
For ULA

$$c[n] = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \end{bmatrix}^T \tag{73.7}$$

Advantage of no redundancy is they are placed well but disadvantage is SNR is low.

From a standpoint of efficient spatial sampling we would like the co-array equal to one except at the origin. If we find such a array (called as *perfect array*). There are $N_a = M(M - 1)$=2 number of different off-diagonal elements in a $M \times M$ covariance matrix.

Perfect arrays do not exist for $M > 4$. After $M = 4$, holes are seen in the co-array.

Perfect arrays are shown in Figure 73.2.

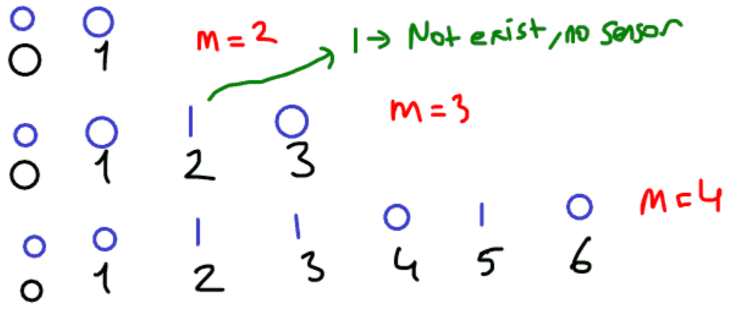

Figure 73.2. Perfect Arrays

For larger arrays we consider two options

73.2. **Non-redundant Arrays.** We consider the array such that $c[n]$ is either zero or one except the origin.

Example-1 $M = 5$ is shown in Figure

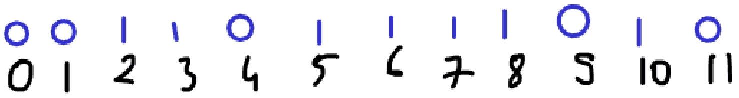

Figure 73.3. Non-redundant Array with $M = 5$

$$c[n] = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 5 \\ s \\ y \\ m \\ m \\ e \\ t \\ r \\ i \\ c \end{bmatrix}^T \tag{73.8}$$

73.3. **Minimum Redundant Arrays.** We choose the sensor positions to make array aperture, N_a, as large as possible without having any gaps. One can write

$$N_a = \frac{M(M-1)}{2} - N_R + N_H \tag{73.9}$$

where N_R is number of redundancies and N_H is number of holes in (73.9). For minimum redundant arrays, $N_H = 0$.
Example-2

[redacted]

Figure 73.4. Minimum Redundant Array Example

For that case,

$$c[n] = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 2 \\ 1 \\ 1 \\ 5 \\ s \\ y \\ m \\ m \\ e \\ t \\ r \\ i \\ c \end{bmatrix}^T \tag{73.10}$$

73.4. **Covariance Matrix Augmentation.** Given the ULA in 5 × 1, with $M = 5$ elements we have all the covariance lags up to 11 except lag = 6 (What ?). In covariance matrix augmentation, we would like to obtain $M_d \times M_d$ covariance matrix ($M_d > M$) from M sensor data. We can do this by using fully augmentable NLA (minimum redundant arrays like last Example-2) or by using partially augmentable NLA (non-redundant arrays like last Example-1).

Consider Example-2. $M = 5$, the covariance matrix augmentation for 10 × 10 covariance matrix can be applied as follows.

Consider (73.10), using Toeplitz completion.

$$R = \begin{bmatrix} \frac{a_0}{5} & a_1 & a_2 & \frac{a_3}{2} & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 \\ & \ddots & a_1 & a_2 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix} \tag{73.11}$$

$$r = \begin{bmatrix} y_0 \\ y_1 \\ 0 \\ 0 \\ y_4 \\ 0 \\ 0 \\ y_7 \\ 0 \\ y_9 \end{bmatrix} \begin{bmatrix} y_0^* & y_1^* & 0 & 0 & y_4^* & 0 & 0 & y_7^* & 0 & y_9^* \end{bmatrix} \tag{73.12}$$

$$a_0 = r_{00} + r_{11} + r_{44} + r_{77} + r_{99} \tag{73.13}$$

$$a_1 = r_{01} \tag{73.14}$$

$$a_2 = r_{79} \tag{73.15}$$

$$a_3 = r_{14} + r_{47} \tag{73.16}$$

74. Array Interpolation

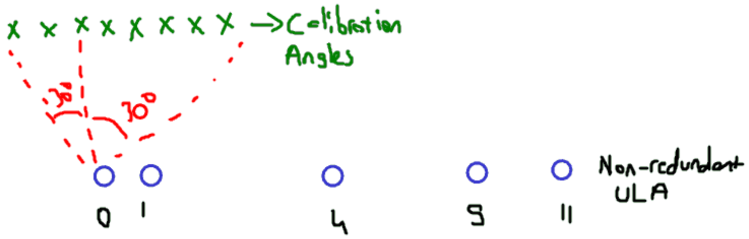

Figure 74.1. Array Interpolation

There is a sector of interest where the source DOAs are assumed to exist. Violation of this assumption may result in significant losses.

Angular sector is diveded into angles composed of calibration sources. These angles are $\tilde{\ }_i$.

$$\tilde{\ }_i = i\Delta \qquad i = 1;2;:::;\frac{{}_f - {}_b}{\Delta}; \qquad \tilde{\ }_i \in [\ {}_b;\ {}_f] \tag{74.1}$$

where also

$$P \triangleq \frac{{}_f - {}_b}{\Delta} \tag{74.2}$$

and ${}_f$ is final and ${}_b$ is beginning angle.

Let signal of real array is:

$$y_r(t) = A_r(\ )s(t) + e(t) \tag{74.3}$$

and for virtual array

$$y_v(t) = A_v(\ )s(t) + e(t) \tag{74.4}$$

Then *Array Interpolation Marrix* $T_{M\times M}$ is obtained.

$$T_{M\times M}A_{r_{M\times P}}(\tilde{\ }) = A_{v_{M\times P}}(\tilde{\ }) \tag{74.5}$$

where P (number of calibratin angles) is greater or equal than M. Then,

$$T = A_v A_r^H (A_v A_r^H)^{-1} \tag{74.6}$$

(74.6) is a classical array interpolation (AI) mapping. It has limited sector width, biased and ill-conditioned matrix (for inversion) (?).

75. Wiener Array Interpolation

$$y(t) = A_r s(t) + e_r(t) \tag{75.1}$$

is given.

$$\hat{y}(t) = A_v s(t) \tag{75.2}$$

is desired.

$$E \triangleq E\{(T_y - \hat{y})(T_y - \hat{y})^H\} \tag{75.3}$$

$$\frac{@E}{@T^H} = 0 \tag{75.4}$$

then,

$$T = A_v R_s A_r^H \left(A_r R_s A_r^H + R_e\right)^{-1} \tag{75.5}$$

Notice that in (75.5), R_e term provides robustness for matrix inversion.

This method as better mapping accuracy especially at low SNR. Ill conditioning in matrix inversion is less likely. A larger sector width can be used. Bias effect is also comparably low.

If $R_s = {}^2_s I$ and $R_e = {}^e_s I$ then

$$T = {}^2_s A_v A_r^H \left(A_r \; {}^2_s A_r^H + \; {}^2_e\right)^{-1} \tag{75.6}$$

Signal and noise variances can be estimated using SNR estimate.

One problem with array mapping (or AI) is $TT^H = I$ in general. Therefore transform domain noise is not white. Noise flooring effect shown in Figure 61.3 occurs. So, whitening transformation should be used.

Steps for AI in DOA Estimation

First, obtain the covariance matrix estimate from the real array, $\hat{R}$.

Second, compute the mapped covariance matrix using AI.

$$\hat{R} = T\bar{R}T^H \tag{75.7}$$

Let,

$$B = TT^H \tag{75.8}$$

Define,

$$\tilde{R} = B^{-1/2}\hat{R}(B^{-1/2})^H \tag{75.9}$$

$$\tilde{R} = B^{-1/2}T\bar{A}R_s\bar{A}^H T^H (B^{-1/2})^H + \; {}^2_e I \tag{75.10}$$

Finally find the noise space eigenvectors matrix $\hat{G}$ (found from $\tilde{R}$) and compute the MUSIC spectra; $||a^H(\ )B^{-1/2}\hat{G}||^2$.

76. Noise Whitening

$$y(t) = As(t) + v(t) \tag{76.1}$$

where $v(t)$ is coloured noise.

$$R_y = AR_sA^H + R_v \tag{76.2}$$

Let

$$\bar{y}(t) \triangleq R_v^{-1/2}y(t) = R_v^{-1/2}As(t) + R_v^{-1/2}v(t) \tag{76.3}$$

$$R_{\bar{y}} = R_v^{-1/2}AR_sA^HR_v^{-1/2} + R_v^{-1/2}R_vR_v^{-1/2} \tag{76.4}$$

Finally,

$$R_{\bar{y}} = R_v^{-1/2}AR_sA^HR_v^{-1/2} + I \tag{76.5}$$

then $R_v^{-1/2}$ is whitening transformation.
MUSIC when the model is modified.

$$y(t) = CAs(t) + v(t) \tag{76.6}$$

$$p(\ ) = \frac{1}{a^H(\ )C^HGG^HCa(\ )} \tag{76.7}$$

where G is obtained from R_y (noise space eigenvectors).
For noise whitening case,

$$p(\ ) = \frac{1}{a^H(\ )R_v^{-1/2}GG^HR_v^{-1/2}a(\ )} \tag{76.8}$$

Part 10. 22/04/14 Lecture Note

Array Mapping Methods:

- AI (Array Interpolation)
- WAI (Wiener AI)
- RSS (Rotational Signal Subspace) No noise flooring especially for ULA
- Manifold Seperation

77. RSS (Rotational Signal Subspace)

In AI, T is designed to map A_r (Real Array Manifold) to A_v (Virtual Array Manifold).

In RSS, T is a unitary matrix ($T^H T = TT^H = I$) and therefore does not lead to noise amplification. No noise flooring effect. (See Figure 61.3).

Consider

$$y(t) = As(t) + v(t) \tag{77.1}$$

$$Ty(t) = TAs(t) + Tv(t) \tag{77.2}$$

$$T = \arg\min_{T} ||TA_{r_{M\times K}} - A_{v_{M\times K}}||^2 \qquad s.t: \quad T^H T = I \tag{77.3}$$

The solution is

$$T_{M\times M} = V\hat{\Sigma}U^H \tag{77.4}$$

T is called as *Focusing Matrix*.

where

$$A_r A_v^H = V\Sigma U^H \qquad \text{SVD} \tag{77.5}$$

and

$$\hat{\Sigma} = \begin{bmatrix} I_{M\times M} & 0 \end{bmatrix}^T \tag{77.6}$$

$$y(t) = A_r(\quad)s(t) + v(t) \tag{77.7}$$

$$y_v(t) = Ty(t) = TA_r(\quad)s(t) + Tv(t) \tag{77.8}$$

$$R_{y_v} = TA_r R_s A_r T^H + TT^H \ {}_v^2 \tag{77.9}$$

in (77.9), it is assumed that $R_v = \ {}_v^2 I$, uncorrelated noise and signal.

$$TT^H = V\hat{\Sigma}U^H U\hat{\Sigma}^H V^H \tag{77.10}$$

$$TT^H = V\hat{\Sigma}I\hat{\Sigma}^H V^H \tag{77.11}$$

$$TT^H = VV^H \tag{77.12}$$

$$TT^H = I \tag{77.13}$$

No noise amplification. But for circular arrays it has problems.

78. Manifold Separation

The idea in MS is to map the real array to a virtual array with a Vandermonde steering vector.

There is no limitation on sector width.

This is being done with an approximate mapping as

$$a(\ \ ) \simeq Tg(\ \ ) \tag{78.1}$$

where

$$g(\ \ ) = \begin{bmatrix} e^{-j\frac{P-1}{2}\phi} \\ e^{-j\frac{P-3}{2}\phi} \\ \vdots \\ e^{j\frac{P-3}{2}\phi} \\ e^{j\frac{P-1}{2}\phi} \end{bmatrix} \tag{78.2}$$

where $g(\ \ )$ is $P \times 1$, T is $M \times P$ and $a(\ \ )$ is $M \times 1$ matrix.

Elements of T are:

$$T_{mn} = \sqrt{2\ \ }(j)^n J_n\left(\frac{2\ }{\ }r_m\right)e^{jn\phi_m} \tag{78.3}$$

where $(r_m; \ \ _m)$ are polar coordinates of the m^{th} sensor. J_n is the Bessel function of the first kind, order n.

T is depending only ti (real) sensor positions.

$g(\ \ )$ is a Fourier basis and P determines the approximation accuracy. As $P \to \infty$, (78.1) becomes exact.

This technique does not require a sector of interest.

A suggested value for P:

$$P = \frac{8\ \ r}{\ } \tag{78.4}$$

$$y(t) = a(\ \ )s(t) + v(t) \tag{78.5}$$

$$y(t) = Ta(\ \ )s(t) + v(t) \tag{78.6}$$

$$p(\ ) = \frac{1}{g^H(\ )T^H GG^H T g(\ )} \tag{78.7}$$

where (78.7) is MUSIC pseudo-spectra. G is regular noise space eigenvectors.

79. Wideband Processing (Explanation of the Covariance Matrix Mapping)

79.1. **Coherent WBP.** First,you may review (71.1) and succeeding equations.

$$Y(!_k) = A(!_k)S(!_k) + E(!_k) \tag{79.1}$$

where

$$Y(!_k) \triangleq \begin{bmatrix} Y_1(!_k) \\ Y_2(!_k) \\ \vdots \\ Y_M(!_k) \end{bmatrix} \tag{79.2}$$

$$S(!_k) \triangleq \begin{bmatrix} S_1(!_k) \\ S_2(!_k) \\ \vdots \\ S_M(!_k) \end{bmatrix} \tag{79.3}$$

$$A(!_k)_{ji} \triangleq e^{-j\omega\tau_{ji}} \qquad j = 1;2;:::;M \quad i = 1;2;:::;n \tag{79.4}$$

$$R_k \triangleq E\{Y(!_k)Y^H(!_k)\} \tag{79.5}$$

Using sample covariance matrix:

$$\hat{R}_k \triangleq \frac{1}{N_s}\sum_{p=1}^{N_s} Y(!_k;p)Y^H(!_k;p) \tag{79.6}$$

Consider the steering vector at w_k for ULA

$$a_k(\ ) = \begin{bmatrix} 1 \\ e^{jd\omega_k/ccos\phi} \\ \vdots \\ e^{j(M-1)d\omega_k/ccos\phi} \end{bmatrix} \tag{79.7}$$

We would like to have

$$d!_k = \text{constant} = d_0!_0 \tag{79.8}$$

where d is element spacing at operating frequency and $!$ is operating frequency.

$$A(!_0) = T_k A(!_k) \qquad \forall k \tag{79.9}$$

Using classical array interpolation for simplicity

$$T_k = A(!_0)A^H(!_k)\left[A(!_k)A^H(!_k)\right]^{-1} \tag{79.10}$$

You can use other previously learned techniques to generate T as well.

$$\bar{R} = \sum_{k=1}^{K} T_k R_k T_k^H \tag{79.11}$$

(79.11) is the final covariance matrix carry all the information in different frequency bands. This is called as *focusing* operation.

DOA estimation methods can be applied on $\bar{R}$ for coherent wideband processing.

79.2. Non-coherent WBP.

$$s(\ \ ) \triangleq \sum_{k=1}^{K} a^H(!_k;\ \ )G_k G_k^H a^H(!_k;\ \ ) \tag{79.12}$$

G_k is obtained from R_k.

$$p_{incoherent}(\ \ ) = \frac{1}{s(\ \ )} \tag{79.13}$$

(79.13) is MUSIC pseudo-spectra.

Notice that for each frequency bin, SVD operation should be done and it has computational load.

Coherent methods works closely well as non-coherent methods with significant computational advantage.

10:40

80. Source Localization

The problem with passive source localization is to determine the location of an emitting target. This is also called as position fixing (PF).

The physical quantities if localization are:

- **Time of Arrival (TOA)**

  Time that the signal from the transmitter reaches to receiver. It requires synchronisation.
- **Time Difference of Arrival (TDOA)**

  Difference of the arrival times between the sensors is used. No need to sync to transmitter but sensors should be synced.
- **Phase**

  Direction of arrival estimates ate each sensor (array) are used (DF).
- **Amplitude, RSS (Received Signal Strength)**

  Used especially for wireless sensors in close proximity.
- **Frequency Difference of Arrival (FDOA)**
- **Doppler Frequency**

  Two types of localization problems: Navigation and Localization. They have some common points.

  In navigation transmitter positions are known and the receiver has to be located.

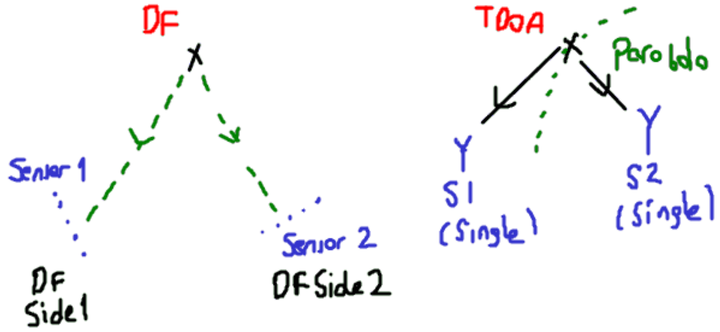

Figure 80.1. Sensor Array and Single Sensor Case

In localization, receivers have known positions and the transmitter has to be located.

	TOA	TDOA
Navigation ($Tx > 1$, $Rx = 1$)	GPS	LORAN, DECCA, mobile positioning
Localization ($Rx > 1$, $Tx = 1$)	Active Radar, Sonar	Passive Radar, Sonar, Seismic

Some common techniques for localization:

- DOA ant triangulation
- TDOA (Requires wideband signal) (?)
- TOA
- Hybrid techniques: DOA with TDOA, etc ...

Localization accuracy depends on

- TDOA and DOA accuracy
- Sensor positions with respect to target
- The method used for estimation

81. Triangulation

Two or more LOB (line of bearings) assumed to be measured on the target at the same time can be intersected for position finding (PF). This technique is called as triangulation.

$$sin \quad _1 = \frac{d_1}{d} \tag{81.1}$$

$$d_1 = dsin \quad _1 \tag{81.2}$$

(81.3)

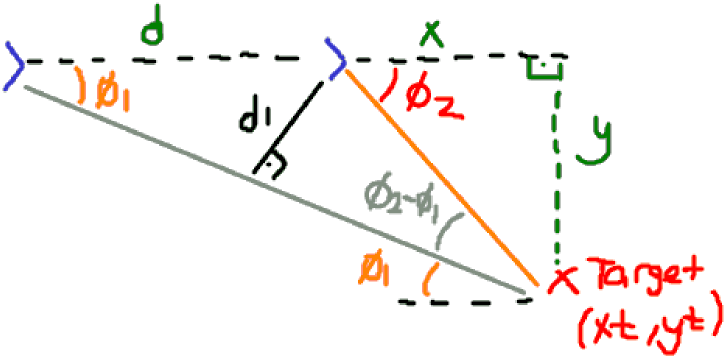

Figure 81.1. Triangulation

$$r = \frac{dsin\ \ _{1}}{sin(\ \ _{2} - \ \ _{1})} \tag{81.5}$$

$$x = rcos\ \ _{2} \tag{81.6}$$

$$y = rsin\ \ _{2} \tag{81.7}$$

It can be generalized to 3-D case.

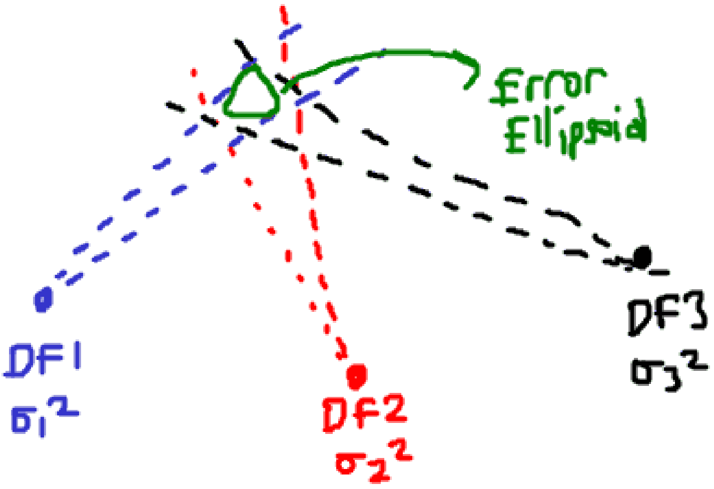

Figure 81.2. Error Ellipsoid

11:40

There are three deterministic methods for location given the triangle of LOBs.

- a) Intersection of medians
- b) Intersection of angle bisectors

- c) Steiner point (defined as the point where angles between lines from the corners are 120 degree)

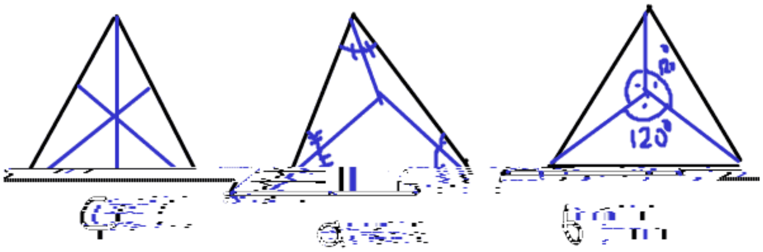

Figure 81.3. 3 Deterministic Methods

82. Least-Squares Location Estimation

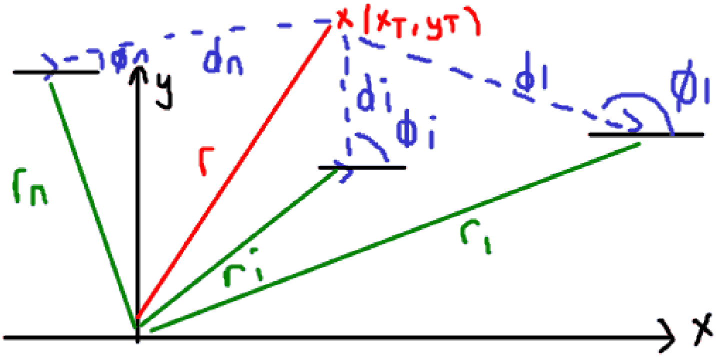

Figure 82.1. Least-Squares Location Estimation

Let define a position vector, r:

$$r = \begin{bmatrix} x_T \\ y_T \end{bmatrix} \tag{82.1}$$

$$r = r_i + d_i u_i \tag{82.2}$$

where r_i is position of the i^{th} DF sensor, d_i is distance from i^{th} DF sensor to target u_i is unit direction vector where

$$u_i \triangleq \begin{bmatrix} cos \quad _i \\ sin \quad _i \end{bmatrix} \tag{82.3}$$

$$r = \begin{bmatrix} x_T \\ y_T \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} + d_i \begin{bmatrix} cos \quad _i \\ sin \quad _i \end{bmatrix} \tag{82.4}$$

$$x_T = x_i + d_i cos \quad _i \tag{82.5}$$

(82.6) $$y_T = y_i + d_i sin \;\; {}_i$$

From previous equations,

(82.7) $$- x_i sin \;\; {}_i + y_i cos \;\; {}_i = -x_T sin \;\; {}_i + y_T cos \;\; {}_i$$

(82.8) $$\begin{bmatrix} -x_1 sin \;\; {}_1 + y_1 cos \;\; {}_1 \\ \vdots \\ -x_N sin \;\; {}_1 + y_N cos \;\; {}_N \end{bmatrix} = \begin{bmatrix} -sin \;\; {}_1 & cos \;\; {}_1 \\ \vdots & \vdots \\ -sin \;\; {}_N & cos \;\; {}_N \end{bmatrix} \begin{bmatrix} x_T \\ y_T \end{bmatrix}$$

In matrix form,

(82.9) $$b(\;\;) = H(\;\;)x$$

(82.10) $$\hat{x} = \left[H^H H\right]^{-1} H^H b$$

Note $H(\;\;)$ is overdetermined if $N > 2$. It is a good thing.

83. Maximum Likelihood Algorithm

Assume that noise is Gaussian and zero-mean.

(83.1) $$\hat{x} = \arg \min_x F(x; \;\; ; \;\;)$$

(83.2) $$F(x; \;\;) = \frac{1}{2}[g(x) - \;\;]^T S^{-1}[g(x) - \;\;]$$

(83.3) $$g(x) = \begin{bmatrix} g_1(x) \\ g_2(x) \\ \vdots \\ g_N(x) \end{bmatrix}$$

(83.4) $$g_i(x) = tan^{-1}\left(\frac{\Delta y_i}{\Delta x_i}\right)$$

(83.5) $$\Delta x_i = x_T - x_i$$

(83.6) $$\Delta y_i = y_T - y_i$$

where $(x_i; y_i)$ is the i^{th} DF site position and $(x_T; y_T)$ is target position.

(83.7) $$= \begin{bmatrix} {}_1 \\ {}_2 \\ \vdots \\ {}_N \end{bmatrix}$$

$$S = diag(\ {}_1^2;\ \ {}_2^2; :::;\ \ {}_N^2) \tag{83.8}$$

where N is the number of DF sites and ${}_i^2$ is variance of azimuth estimates. The solution can be found by using Gauss-Newton iterative method.

$$\hat{x}_{M+1} = \hat{x}_M + [g_x^T S^{-1} g_x]^{-1} g_x^T S^{-1} [\quad - g(\hat{x}_M)] \tag{83.9}$$

$$g_x \triangleq \begin{bmatrix} \frac{-\Delta y_1}{r_1^2} & \frac{\Delta x_1}{r_1^2} \\ \frac{-\Delta y_2}{r_2^2} & \frac{\Delta x_2}{r_2^2} \\ \vdots & \vdots \\ \frac{-\Delta y_N}{r_N^2} & \frac{\Delta x_N}{r_N^2} \end{bmatrix} \tag{83.10}$$

g_x is called as *Jacobian Matrix*.

$$r_i^2 = x_i^2 + y_i^2 \tag{83.11}$$

To start search, result of LS solution may used as initial point.

Part 11. 13/05/14 Lecture Note

84. Sources of Error in Triangulation

- Noise
- Measurement Noise
- Geometric Dilution of Precision in Triangulation

  If the target is farther away from the sensor baseline, the error in position fixing increases.

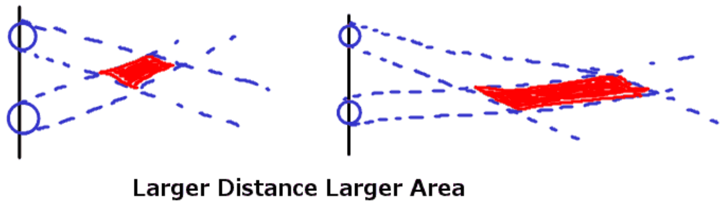

Figure 84.1. Geometric Dilution of Precision in Triangulation

- Line of Bearing Error
- Effect of Navigation Error

  The errors in sensor position affects the PF accuracy. Usually the error is due to GPS measurements.

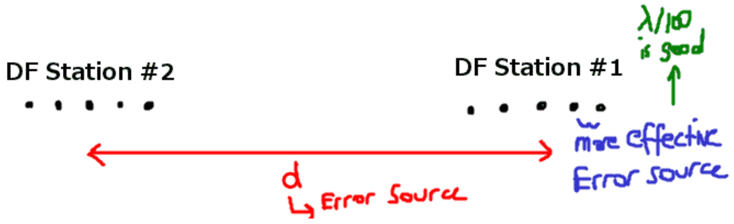

Figure 84.2. Placement Errors

85. Single-Site Location Estimation (SSL)

RF case, HF frequency (3 - 30 MHz)

HF signals are long-range signals that are reflected (refracted) in the ionosphere back to earth.

Measuring DOA (azimuth and elevation) and the height of the point of the reflection allows us to calculate the position of the emitter.

Ionosphere height may no be needed when there are two ore more ray paths arriving at the sensor side.

10:40

Ionospheric sounders are used to measure the height of ionosphere. The time of flight of pulse is used to find the height of ionosphere.

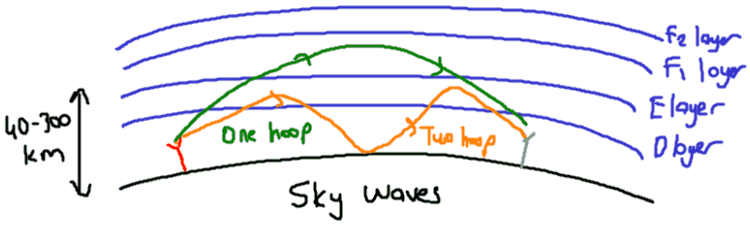

Figure 85.1. Reflections from Ionosphere

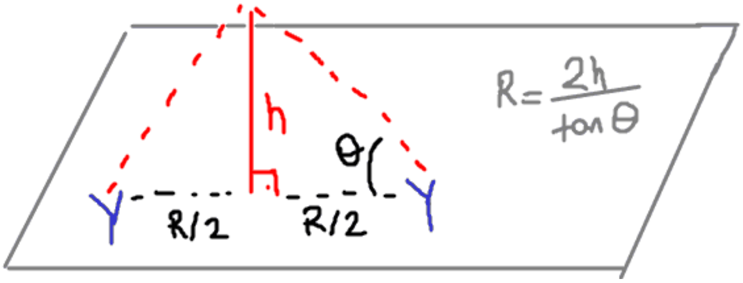

Figure 85.2. SSL Assumption

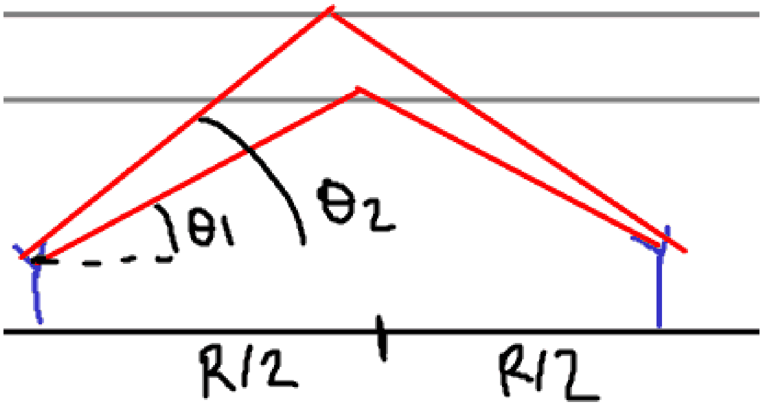

Figure 85.3. Passive SSL

85.1. **Passive SSL.** When the signals arrive from two directions, there is no need to know height.

$$R = \frac{c}{tan \;_2 - tan \;_1} = \frac{cos\frac{_2 - \;_1}{2}}{sin\frac{_2 + \;_1}{2}} \tag{85.1}$$

where is the time difference between two paths.

86. TDOA-FDOA Location Estimation

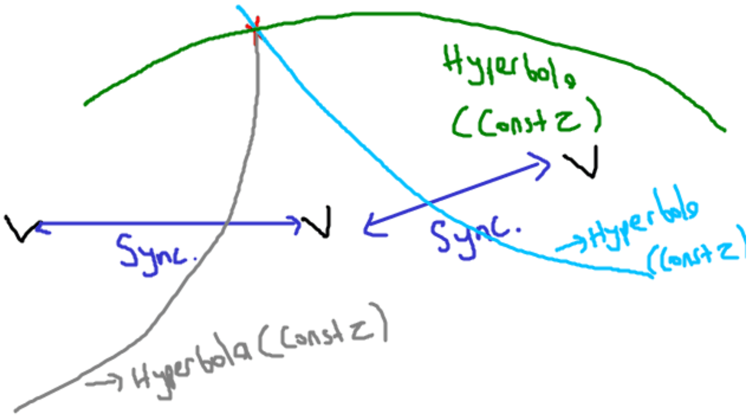

Figure 86.1. TDOA Scenario

$$\hat{s}_i(t) = e^{j\alpha_i} e^{-j\omega d_i t} \hat{s}(t - \ {}_i) \tag{86.1}$$

where ${}_i$ is time delay.

TDOA requires at least 4 sensors.

Narrow Band Received Signal

$$\hat{s}_i(t) = e^{j\alpha_i} e^{-j\omega d_i t} \hat{s}(t - \ {}_i) \tag{86.2}$$

where

$${}_i \triangleq \, !_c \ {}_i \tag{86.3}$$

$$!d_i \triangleq \frac{!cvi}{c} \tag{86.4}$$

${}_i$ is the time delay.

To estimate time of differences radars (planes) should have perfect synchronization.

Delay Estimation

Cross correlation of time signals

$$R_{s_1 s_2}(\ ) = \int_0^T s_1(t) s_2^*(t - \ ) dt \tag{86.5}$$

find peak of $|R_{s_1 s_2}(\ )|$.

Doppler Shift

$$R_{s_1 s_2}(!) = \int_0^T s_1(t) s_2^*(t) e^{-j\omega t} dt \tag{86.6}$$

find peak of $|R_{s_1 s_2}(!)|$.

TDOA/FDOA

Ambiguity function

$$A(!;\ ) = \int_0^T s_1(t)s_2^*(t-\ )e^{-j\omega t}dt \tag{86.7}$$

find peak of $|A(!;\ )|$.

87. Quadratic Localization Methods

TDOA is an example of quadratic methods. LOP (Lines of position) curves are intersected to estimate the emitter location.

Advantages

- In triangulation an array is used for DOA measurement. At least two DF site is required for PF (position finding).

  In TDOA, a single antenna/sensor and at least 4 sensors are required for 3-D PF.
- Usually higher precision and accuracy can be obtained in TDOA (especially for radar signals)

Disadvantages

- Accurate and synchronized clocks are needed for each sensor.

Simple Case Example

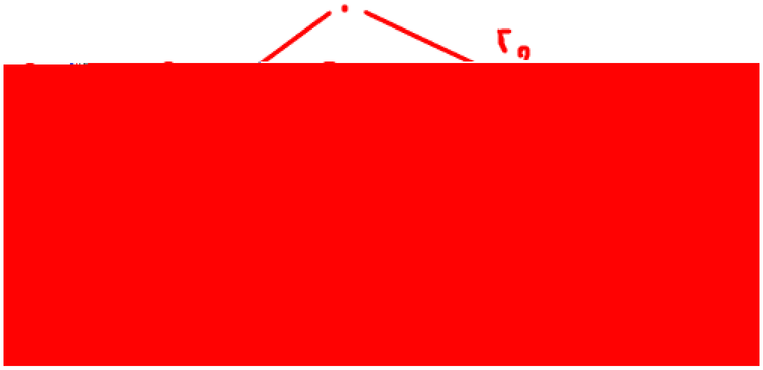

Figure 87.1. Simple Example

c is speed of flight, t_i is the time of flight between the target and sensor.

$$r_i = ct_i \tag{87.1}$$

$$= t_2 - t_1 = \frac{1}{c}(r_2 - r_1) \tag{87.2}$$

$$r_i = \sqrt{(x_T - x_i)^2 + y_T^2} \qquad i = 1;2 \tag{87.3}$$

$$\Delta r = r_2 - r_1 = \sqrt{(x_T - x_1)^2 + y_T^2} - \sqrt{(x_T - x_2)^2 + y_T^2} \tag{87.4}$$

$$d = x_1 + x_2 \tag{87.5}$$

$$\frac{x^2}{a} - \frac{y^2}{b} = 1 \tag{87.6}$$

(87.6) is hyperbola.

$$a = \frac{\Delta r^2}{4} \tag{87.7}$$

$$b = \frac{d^2}{4} - \frac{\Delta r^2}{4} \tag{87.8}$$

$$y = \pm \frac{b}{a} x \tag{87.9}$$

(87.9) are the asymptotes which define DOA for far field sources.
Line of positions (LOP) are isochrones.

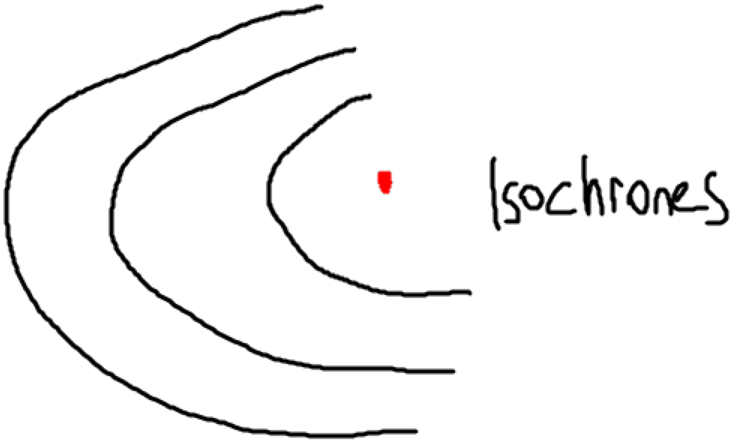

Figure 87.2. Line of Positions Isochrones

88. Location Estimation by TDOA

$${}_{i} = t_i - t_1 \tag{88.1}$$

t_1 is the time of flight to the reference sensor.
$(x_T; y_T; z_T)$ is the target position.

$$(ct_1)^2 = d^2 = (x_1 - x_T)^2 + (y_1 - y_T)^2 + (z_1 - z_T)^2 \tag{88.2}$$

(88.3) $$(ct_i)^2 = (d + c\ {}_i)^2 = (x_i - x_T)^2 + (y_i - y_T)^2 + (z_i - z_T)^2 \qquad i = 1; 2; 3; 4$$

Linearization

- Addition of a new variable
- Taylor series expension

11:40

88.1. **Addition of New Variable.**

(88.4) $$d^2 = (x_1 - x_T)^2 + (y_1 - y_T)^2 + (z_1 - z_T)^2$$

(88.5) $$d^2 = x_1^2 + y_1^2 + z_1^2 - 2x_1x_T - 2y_1y_T - 2z_1z_T + x_T^2 + y_T^2 + z_T^2$$

(88.6) $$d^2 + 2dc\ {}_2 + c^2\ {}_2^2 = x_2^2 + y_2^2 + z_2^2 - 2x_2x_T - 2y_2y_T - 2z_2z_T + x_T^2 + y_T^2 + z_T^2$$

Let's subtract (88.5) from (88.6)

(88.7) $$-2dc\ {}_2 - c^2\ {}_2^2 = (x_1^2 + y_1^2 + z_1^2) - (x_2^2 + y_2^2 + z_2^2) + 2(x_2 - x_1)x_T + 2(y_2 - y_1)y_T + 2(z_2 - z_1)z_T$$

where $x_1; x_2; y_1; y_2; z_1; z_2$ is known in (88.7).
Similarly obtain (88.6) for other is and subtract (88.5) from them.
Then,

(88.8) $$2\begin{bmatrix} x_1 - x_2 & y_1 - y_2 & z_1 - z_2 & -c\ {}_2 \\ x_1 - x_3 & y_1 - y_3 & z_1 - z_3 & -c\ {}_3 \\ \vdots & \vdots & \vdots & \vdots \\ x_1 - x_N & y_1 - y_N & z_1 - z_N & -c\ {}_N \end{bmatrix} \begin{bmatrix} x_T \\ y_T \\ z_T \\ d \end{bmatrix} = \begin{bmatrix} c^2\ {}_2^2 + x_1^2 + y_1^2 + z_1^2 - x_2^2 - y_2^2 - z_2^2 \\ \vdots \\ c^2\ {}_N^2 + x_1^2 + y_1^2 + z_1^2 - x_N^2 - y_N^2 - z_N^2 \end{bmatrix}$$

(88.9) $$Ax = b$$

Then,

(88.10) $$x = (A^H A)^{-1} A^H b$$

3D coordinates can be found by using at least 5 sensors. $(N \geq 5)$
It is desired to have more sensors and an overdetermined set of equations to improve the accuracy.
Sensors should be sufficiently further away from each other in order to have different $\ {}_i$.

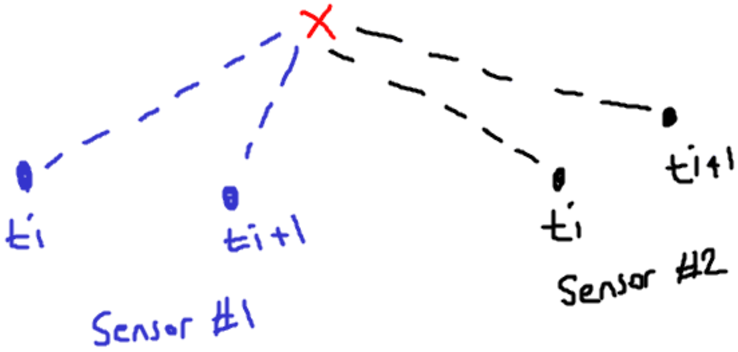

Figure 88.1. Taylor Series Approach Assumption

88.2. **Taylor Series Approach.** The equation can be linearized by a Taylor series expansion about an initial estimate of target location $(x_{T_O}; y_{T_O}; z_{T_O})$

$$f(x) = f(x_{T_O}) + \frac{f'(x_{T_O})}{1!}(x_T - x_{T_O}) + f''(x_{T_O}) \tag{88.11}$$

$$\begin{bmatrix} P_1 - f_1 \\ P_2 - f_2 \\ \vdots \\ P_k - f_k \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_T} & \frac{\partial f_1}{\partial y_T} & \frac{\partial f_1}{\partial z_T} \\ \frac{\partial f_2}{\partial x_T} & \frac{\partial f_2}{\partial y_T} & \frac{\partial f_2}{\partial z_T} \\ \vdots & & \\ \frac{\partial f_k}{\partial x_T} & \frac{\partial f_k}{\partial y_T} & \frac{\partial f_k}{\partial z_T} \end{bmatrix} \begin{bmatrix} x_T - x_{T_0} \\ y_T - y_{T_0} \\ z_T - z_{T_0} \end{bmatrix} \tag{88.20}$$

$$Ax = P \tag{88.21}$$

$$\hat{x} = (A^H A)^{-1} A^H P \tag{88.22}$$

$$\hat{x}_T = \hat{x} + \begin{bmatrix} x_{T_0} \\ y_{T_0} \\ z_{T_0} \end{bmatrix} \tag{88.23}$$

Accuracy depends on:

- Sensor geometry with respect to target
- Timing accuracy of the receiver
- Multipath
- Inaccuracy in sensor positions
- Frequency synchronization between transmitter and receiver

89. ML Source Localization

Noise is zero-mean Gaussian.
The cost function for ML

$$F_\tau(p) = \frac{1}{2}(\hat{h} - h(p))^T C_\tau^{-1}(\hat{h} - h(p)) \tag{89.1}$$

where C_τ is the covariance matrix of range differences.

$$\hat{h} = c \begin{bmatrix} {}_{2} \\ {}_{3} \\ \vdots \\ {}_{M} \end{bmatrix} \tag{89.2}$$

where ${}_i = t_i - t_1$ is the time difference with respect to first sensor.

$$h(p) = \begin{bmatrix} h_2(p) \\ h_3(p) \\ \vdots \\ h_M(p) \end{bmatrix} \tag{89.3}$$

$$h_i(p) = ||p_i - p_T|| - ||p_1 - p_T|| \tag{89.4}$$

where p_T is target and p_i is sensor position.

Assume that TDOA measurements are independent from each other.

$$R_\tau = c^2 \;\; {}^{2}_{TDOA} \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} & \cdots \frac{1}{2} \\ \frac{1}{2} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \frac{1}{2} & \cdots & \frac{1}{2} & 1 \end{bmatrix} \tag{89.5}$$

If a reference sensor is used where each TDOA is computed with respect to this sensor, there is a correlation between measurements and covariance matrix becomes as this.

Then MLE algorithm is:

- Estimate the TDOA for sensor pairs.
- Obtain $\hat{h}$ and C_τ
- Perform a grid search at each point
  - Calculate the range difference between the first sensor and others with respect to the target, $h_i(p)$
  - Calculate the cost function, $F_\tau(p)$.
- Find the target position which minimizes the cost function.

Part 12. 20/05/14 Lecture Note

90. Received Signal Strength (RSS) Location

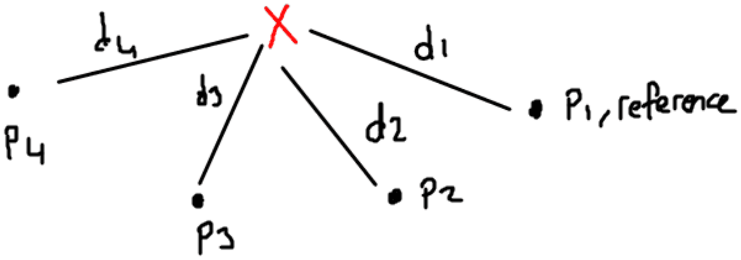

Figure 90.1. RSS Example

We will consider differential RSS method which is suitable for passive incorporating localization. Otherwise transmitter power should be known.

Advantages

- Simplicity
- Size
- Power
- Complexity
- Cost

Disadvantages

- Noisy range estimates
- Multipath
- Attenuation due to environmental factors.

90.1. Signal Model.

$$P_1 = P_T - 10 \ \ log_{10} d_1 + n_1 \tag{90.1}$$

$$P_i = P_T - 10 \ \ log_{10} d_i + n_i \qquad i = 1; 2; : : : ; N \quad N \geq 4 \tag{90.2}$$

where P_i received signal power in dB and P_T is transmitter power in dB. is path loss factor ($2 \leq \ \ \leq 6$). d_i is distance between sensor and transmitter. n_i is the noise (shadow fading)

$$d_i = \sqrt{(x_T - x_i)^2 + (y_T - y_i)^2} \tag{90.3}$$

In order to eliminate P_T, subtract (90.1) from (90.2).

$$P_{i1} = 0{:}1 ln(10)[P_i - P_1] = - \ \ ln\left(\frac{d_i}{d_1}\right) + m_{i1} \qquad i = 2; : : : ; N \tag{90.4}$$

where

$$m_{i1} \triangleq 0{:}1 ln(10)(n_i - n_1) \tag{90.5}$$

In order to linearize (90.3), d_i^2 is required.

$$e^{-2/\gamma P_{i1}} = \left(\frac{d_i}{d_1}\right)^2 e^{-2/\gamma m_{i1}} \tag{90.6}$$

$$E\{e^{-2/\gamma P_{i1}}\} = \left(\frac{d_i}{d_1}\right)^2 e^{-2/\gamma^2(\lambda_i^2+\lambda_1^2)} \qquad i = 2;:::;N \tag{90.7}$$

(or ^{2} at exponent?)

$$_i^2 \triangleq 0{:}01[ln(10)]^2 \;\; _i^2 \tag{90.8}$$

$$_i^2 \triangleq Var\{n_i\} \tag{90.9}$$

Unbiased estimate of $\dfrac{d_i^2}{d_1^2}$

$$\frac{d_i^2}{d_1^2} = r_{i1} = e^{-\frac{2}{\;}P_{i1}-\frac{2}{\;^2}(\lambda_i^2+\lambda_1^2)} \tag{90.10}$$

Ignoring noise

$$r_{i1}d_1^2 = d_i^2 \qquad i = 2;:::;N \tag{90.11}$$

From (90.3)

$$d_i^2 = x_T^2 + y_T^2 - 2x_i x_T - 2y_i y_T + x_i^2 + y_i^2 \qquad i = 2;:::;N \tag{90.12}$$

$$R \triangleq x_T^2 + y_T^2 \tag{90.13}$$

$$r_{i1}d_1^2 = r_{i1}\left[(x_T - x_1)^2 + (y_T - y_1)^2\right] \tag{90.14}$$

$$r_{i1}d_1^2 = r_{i1}\left[R - 2x_1x_T - 2y_1y_T + x_1^2 + y_1^2\right] \tag{90.15}$$

From (90.12) and (90.15),
For $i = 2$,

$$R(r_{21}-1) + (2x_2 + 2x_1r_{21})x_T + (2y_2 - 2y_1r_{21})y_T = x_2^2 + y_2^2 - r_{21}(x_1^2 + y_1^2) \tag{90.16}$$

Let,

$$A \triangleq \begin{bmatrix} 2x_2 - 2r_{21}x_1 & 2y_2 - 2r_{21}y_1 & r_{21} - 1 \\ \vdots & \vdots & \vdots \\ 2x_N - 2r_{N1}x_1 & 2y_N - 2r_{N1}y_1 & r_{N1} - 1 \end{bmatrix} \tag{90.17}$$

$$\triangleq \begin{bmatrix} x_T \\ y_T \\ R \end{bmatrix} \tag{90.18}$$

$$b \triangleq \begin{bmatrix} x_2^2+y_2^2-r_{21}(x_1^2+y_1^2) \\ \vdots \\ x_N^2+y_N^2-r_{N1}(x_1^2+y_1^2) \end{bmatrix} \tag{90.19}$$

$$A \quad = b \tag{90.20}$$

LS solution is:

$$\hat{\ } = \begin{bmatrix} x_T \\ y_T \\ R \end{bmatrix} = (A^H A)^{-1} A^H b \tag{90.21}$$

Define error vector,

$$W \triangleq \begin{bmatrix} W_2 \\ W_3 \\ \vdots \\ W_N \end{bmatrix} \tag{90.22}$$

$$w_i = d_1^2 r_{i1} = d_i^2 \qquad i = 2;:::;N \tag{90.23}$$

$$E\{w\} = 0 \tag{90.24}$$

$$C_w = \Sigma\Lambda\Sigma^H \tag{90.25}$$

$$\Sigma = diag(r_{21};:::;r_{N1}) \tag{90.26}$$

$$\Lambda = \begin{bmatrix} e^{\frac{4}{\ 2}(\lambda_2^2+\lambda_1^2)}-1 & e^{\frac{4}{\ 2}\lambda_1^2}-1 & ::: & e^{\frac{4}{\ 2}\lambda_1^2}-1 \\ \vdots & \ddots & & \\ e^{\frac{4}{\ 2}\lambda_1^2}-1 & ::: & ::: & e^{\frac{4}{\ 2}(\lambda_N^2+\lambda_1^2)}-1 \end{bmatrix} \tag{90.27}$$

90.2. **BLUE (Best Linear Unbiased Example) Solution.**

$$\hat{\ } = (A^H C_w^{-1} A)^{-1} A^H C_w^{-1} b \tag{90.28}$$

91. Calibration

Calibration corresponds to correction due to unknowns or uncertainties of the system parameters.

In practical systems, it is not possible to operate the system without appropriate calibration.

91.1. **Bench Test and Calibration.** It is shown in Figure 90.1.

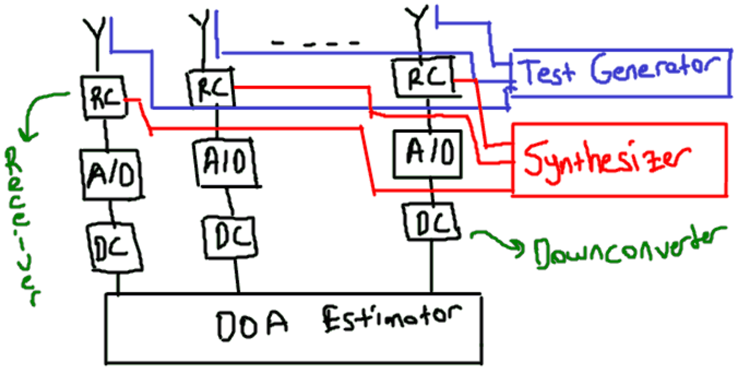

Figure 91.1. Built-in Calibration Example

91.2. **Anechoic Chamber Test and Calibration.** It is an special shielded room with RF absorbers on the walls as well as ground and ceiling. Turn table is used to rotate DF system. A calibrated antenna and transmitter is used as source to test the system for different frequency (*), azimuth (**), elevation (*), modulation, polarization. (*: Importance)

91.3. **Calibrated Site Test and Calibration.** A calibrated open field test facility bridges the the gap between production and operational use. Test area should have 15 to 10 dimensions at the lowest frequency.

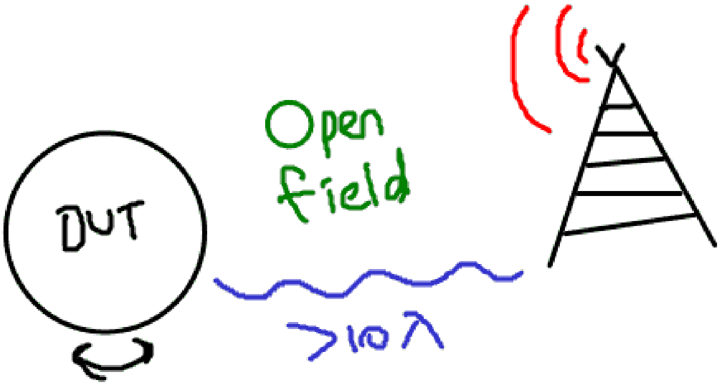

Figure 91.2. Open Field Calibration

For each frequency, azimuth. elevation, test equipment is rotated around a circle and an emitter is used to collect the calibration data. Calibration data can be interpolated to obtain better angular coverage.

91.4. **Installed Test and Calibration.** DF system is installed on its platform and tested in open test field.

91.5. **Operational Test and Calibration.** If your system is fixed at specific field, this is an important method.

Nothing to say about !

11:40

92. How to Calibrate?

Consider M sensor outputs, baseband signals (RF situation).

$$y(t) = \sum_{k=1}^{n} a(\ _k)s_k(t) + v(t) = A(\ )s(t) + v(t) \qquad t = 1; : : : ; N \tag{92.1}$$

(92.1) shows ideal case. Practical case shown in (88.13).

$$y(t) = CGA(\ )s(t) + v(t) \qquad t = 1; : : : ; N \tag{92.2}$$

where C is mutual coupling matrix and G is gain-phase mismatch matrix.

Error sources:

- Mutual coupling
- Uncertainty for antenna positions and orientations
- Gain-phase imbalance of receivers.
- IQ imbalances of receivers
- Near-field scattering due to platform or terrain
- Non-linear components
- Quantization in phase shifters, A/D converters

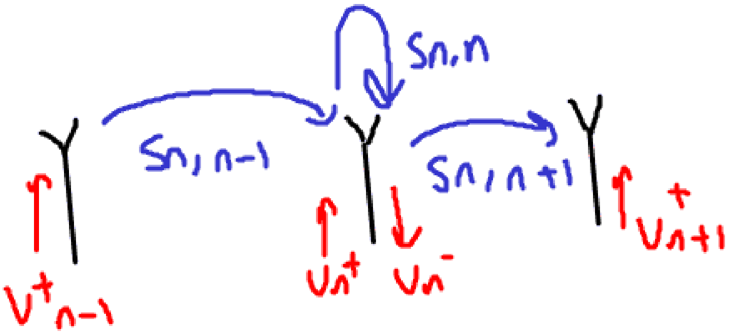

Figure 92.1. Mutual Coupling

$$V_n^- = S_{n,n+1}V_{n-1}^+ + S_{n,n}V_n^+ + S_{n,n+1}V_{n+1}^+ \tag{92.3}$$

(92.3) gives reflected wave at antenna n.

$$V^- = SV^+ \tag{92.4}$$

S is scattering matrix and S_{ij} are scattering parameters.

For 2 antenna case:

$$\begin{bmatrix} V_1^- \\ V_2^- \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} V_1^+ \\ V_2^+ \end{bmatrix} \tag{92.5}$$

In (92.5), S_{11} is input port reflection coefficient, S_{12} is reverse voltage gain, S_{21} is forward voltage gain, S_{22} is output port reflection coefficient.

$$\Gamma = \frac{V_r}{V_f} \tag{92.6}$$

where V_r is reflected wave and V_f is forward wave. When $\Gamma = \pm 1$ corresponds to maximum reflection and $\Gamma = 0$ corresponds to perfect match, no reflection.

$$VSWR = \frac{1+}{1-} = \frac{V_{max}}{V_{min}} \geq 1 \tag{92.7}$$

$$Z = (I - S)^{-1}(I + S) \tag{92.8}$$

$$C = Z^{-1} \tag{92.9}$$

where Z is mutual *impedance* matrix and C is mutual *coupling* matrix.

92.1. Auto-Calibration (Online Calibration) (Self Calibration).

$$y(t) = CGA(\quad)s(t) + v(t) \tag{92.10}$$

C has a structure for uniform arrays like circular, linear arrays, etc.

$$C = \begin{bmatrix} c_0 & c_1 & c_2 & 0 \\ c_1 & c_0 & c_1 & c_2 \\ c_2 & c_1 & c_0 & c_1 \\ 0 & c_2 & c_1 & c_0 \end{bmatrix} \tag{92.11}$$

(92.11) is called as *Bended Toeplitz Matrix*, it is for linear arrays.

$$G = diag(g_1; g_2; : : : ; g_m) \tag{92.12}$$

For structured C matrix

$$\bar{a}_1 = Ca_1 \tag{92.13}$$

$$\bar{a}_1 = T_{a_1}c = T_{a_1} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} \tag{92.14}$$

$$\tilde{P} = (C^H T_{a_1} G G^H T_{a_1} C)^{-1} \tag{92.15}$$

(92.16) $$\hat{C} = \arg\max_{C} \tilde{P}$$

where G is obtained from R_y as in MUSIC algorithm.
In realistic scenarios self calibration doesn't have very much chance.

92.2. **Offline Calibration.** Calibration data is collected in a controlled manner.

(92.17) $$x_c(t) = a_c s_c(t) + v_c(t) \qquad t = 1;:::;N_c \quad c = 1;:::;C$$

where N_c is number of snapshots and C is number of emitter positions and a_c is true array steering vector in (92.17).

92.2.1. *Coherent Calibration.* In this case, $s_c(t)$ is known and true array steering vector estimated as

(92.18) $$\hat{a}_c = \frac{\sum_{t=1}^{N_c} x_c(t)s_c^*(t)}{\sum_{t=1}^{N_c} |s_c(t)|^2}$$

(92.19) $$\hat{a}_c = Ta$$

where T is calibration matrix and a is nominal and $\hat{a}_c$ is true steering vector in (92.18).
This result perfects calibration vectors as $N_c \to \infty$.
There is a synchronization between transmitter and receiver in this structure.

92.2.2. *Non-Coherent Calibration.* When $s_c(t)$ is not known $\hat{a}_c$ is estimated from the principle eigenvector of the sample covariance.

(92.20) $$\hat{R}_c = \frac{1}{N_c}\sum_{t=1}^{N_c} x_c(t)x_c(t)^H = \sum_{k=1}^{M} \hat{x}_k \hat{e}_k \hat{e}_k^H$$

(92.21) $$\hat{a}_c = \quad \hat{e}_1$$

How to apply MUSIC?

(92.22) $$x_c(t) = a_c(\quad)s_c(t) + v_c(t)$$

(92.23) $$T^{-1}x_c(t) = a(\quad)s_c(t) + e_c(t)$$

then use MUSIC.
A better approach is:
$R_c \to G_c$

(92.24) $$p(\quad) = \frac{1}{a^H T^H G_c G_c^H T a}$$

Part 13. Code Appendix

. MATLAB Code Figure 9.2

```matlab
%Alper Yazar
close all;
clear all;

M = 100;
cos_phi = linspace(-1,1,100);
lambda = 2;
d = 0.1;
num = sin(M/2 * d * 2 * pi /lambda * cos_phi);
denum = sin(1/2 * d * 2 * pi /lambda * cos_phi);
fn = 1 / M * num ./ denum;

plot (cos_phi, fn, 'linewidth', 2)
grid on;
xlabel('$cos \phi$','interpreter','latex','Fontsize',15)
ylabel('$B_{AF} (\phi)$','interpreter','latex','Fontsize',15)

annotation('doublearrow',[0.4875 0.548125],...
    [0.758767454350159 0.758767454350159]);

annotation('textbox',...
    [0.560 0.728767454350159 0.058125 0.0510204081632653],'Interpreter','latex',...
    'String',{'$Bw_{HP}$'},...
    'FontSize',15,...
    'FontName','Agency FB',...
    'LineStyle','none');

annotation('doublearrow',[0.440625 0.594375],...
    [0.284714285714286 0.285714285714286]);

annotation('textbox',...
    [0.452875 0.295918367346939 0.038375 0.0484693877551015],'String',{'$Bw_{NN}$'},...
    'FontSize',15,...
    'FontName','Agency FB',...
    'LineStyle','none', ...
    'Interpreter','latex');

title(['$B_{AF} (\phi)$ for ULA where $M$=' num2str(M,'%g') ' $\lambda$=' num2str(lambda,'%g') ' $d$=' num2str(d,'%g')],'interpreter','latex','Fontsize',12)
```

MATLAB Code Figure 10.1

```
%Alper Yazar
close all;
clear all;

M = 100;
theta = linspace(0,2*pi,1e5);
cos_phi = cos(theta);
lambda = 2;
d = 0.1;
num = sin(M/2 * d * 2 * pi /lambda * cos_phi);
denum = sin(1/2 * d * 2 * pi /lambda * cos_phi);
fn = 1 / M * num ./ denum;

%fn = 20*log10(abs(fn));

polar (theta, fn)
grid on;
xlabel('$\phi$','interpreter','latex','Fontsize',15)
ylabel('$B_{AF} (\phi)$','interpreter','latex','Fontsize',15)
title(['$B_{AF} (\phi)$ for ULA where $M$=' num2str(M,'%g') ' $\lambda$=' num2str(lambda,'%g') ' $d$=' num2str(d,'%g')],'interpreter','latex','Fontsize',12)
```

E-mail address: alperyazar@gmail.com