Microphone-Array Signal Processing

José A. Apolinário Jr. and Marcello L. R. de Campos

`{apolin},{mcampos}@ieee.org`

LPS
COPPE/Poli/UFRJ

IME – Lab. Processamento Digital de Sinais

Outline

Outline

1. Introduction and Fundamentals

Outline

1. Introduction and Fundamentals
2. Sensor Arrays and Spatial Filtering

Outline

1. Introduction and Fundamentals
2. Sensor Arrays and Spatial Filtering
3. Optimal Beamforming

Outline

1. Introduction and Fundamentals
2. Sensor Arrays and Spatial Filtering
3. Optimal Beamforming
4. Adaptive Beamforming

Outline

1. Introduction and Fundamentals
2. Sensor Arrays and Spatial Filtering
3. Optimal Beamforming
4. Adaptive Beamforming
5. DoA Estimation with Microphone Arrays

1. Introduction and Fundamentals

General concepts

In this course, signals and noise...

General concepts

In this course, signals and noise...

- have spatial dependence

General concepts

In this course, signals and noise...

- have spatial dependence
- must be characterized as space-time processes

General concepts

In this course, signals and noise...

- have spatial dependence
- must be characterized as space-time processes

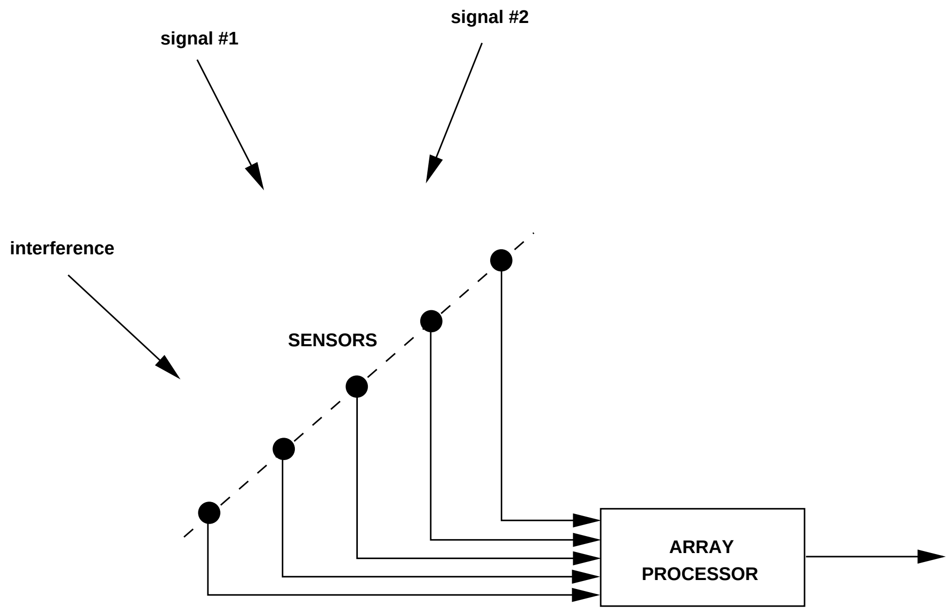
An array of sensors is represented in the figure below.

General concepts

In this course, signals and noise...

- have spatial dependence
- must be characterized as space-time processes

An array of sensors is represented in the figure below.



1.2 Signals in Space and Time

Defining operators

Let $\nabla_{\mathbf{x}}(\cdot)$ and $\nabla^2_{\mathbf{x}}(\cdot)$ be the gradient and Laplacian operators, i.e.,

Defining operators

Let $\nabla_{\mathbf{x}}(\cdot)$ and $\nabla^2_{\mathbf{x}}(\cdot)$ be the gradient and Laplacian operators, i.e.,

$$\begin{aligned}\nabla_{\mathbf{x}}(\cdot) &= \frac{\partial(\cdot)}{\partial x}\overrightarrow{\imath}_x + \frac{\partial(\cdot)}{\partial y}\overrightarrow{\imath}_y + \frac{\partial(\cdot)}{\partial z}\overrightarrow{\imath}_z \\ &= \left[\frac{\partial(\cdot)}{\partial x} \quad \frac{\partial(\cdot)}{\partial y} \quad \frac{\partial(\cdot)}{\partial z}\right]^T\end{aligned}$$

Defining operators

Let $\nabla_{\mathbf{x}}(\cdot)$ and $\nabla^2_{\mathbf{x}}(\cdot)$ be the gradient and Laplacian operators, i.e.,

$$\begin{aligned}\nabla_{\mathbf{x}}(\cdot) &= \frac{\partial(\cdot)}{\partial x}\overrightarrow{\imath}_x + \frac{\partial(\cdot)}{\partial y}\overrightarrow{\imath}_y + \frac{\partial(\cdot)}{\partial z}\overrightarrow{\imath}_z \\ &= \left[\frac{\partial(\cdot)}{\partial x} \quad \frac{\partial(\cdot)}{\partial y} \quad \frac{\partial(\cdot)}{\partial z}\right]^T\end{aligned}$$

and

$$\nabla^2_{\mathbf{x}}(\cdot) = \frac{\partial^2(\cdot)}{\partial x^2} + \frac{\partial^2(\cdot)}{\partial y^2} + \frac{\partial^2(\cdot)}{\partial z^2}$$

respectively.

Wave equation

From Maxwell’s equations,

$$\nabla^2 \overrightarrow{E} = \frac{1}{c^2} \frac{\partial^2 \overrightarrow{E}}{\partial t^2}$$

Wave equation

From Maxwell's equations,

$$\nabla^2 \overrightarrow{E} = \frac{1}{c^2} \frac{\partial^2 \overrightarrow{E}}{\partial t^2}$$

or, for $s(\mathbf{x}, t)$ a general scalar field,

$$\frac{\partial^2 s}{\partial x^2} + \frac{\partial^2 s}{\partial y^2} + \frac{\partial^2 s}{\partial z^2} = \frac{1}{c^2} \frac{\partial^2 s}{\partial t^2}$$

where: c is the propagation speed, $\overrightarrow{E}$ is the electric field intensity, and $\mathbf{x} = [x\ y\ z]^T$ is a position vector.

Wave equation

From Maxwell’s equations,

$$\boxed{\nabla^2 \overrightarrow{E} = \frac{1}{c^2}\frac{\partial^2 \overrightarrow{E}}{\partial t^2}}$$

“vector” wave equation

or, for $s(\mathbf{x}, t)$ a general scalar field,

$$\frac{\partial^2 s}{\partial x^2} + \frac{\partial^2 s}{\partial y^2} + \frac{\partial^2 s}{\partial z^2} = \frac{1}{c^2}\frac{\partial^2 s}{\partial t^2}$$

where: c is the propagation speed, $\overrightarrow{E}$ is the electric field intensity, and $\mathbf{x} = [x\ y\ z]^T$ is a position vector.

Wave equation

From Maxwell's equations,

$$\boxed{\nabla^2 \overrightarrow{E} = \frac{1}{c^2}\frac{\partial^2 \overrightarrow{E}}{\partial t^2}}$$

"vector" wave equation

or, for $s(\mathbf{x}, t)$ a general scalar field,

$$\boxed{\frac{\partial^2 s}{\partial x^2} + \frac{\partial^2 s}{\partial y^2} + \frac{\partial^2 s}{\partial z^2} = \frac{1}{c^2}\frac{\partial^2 s}{\partial t^2}}$$

"scalar" wave equation

where: c is the propagation speed, $\overrightarrow{E}$ is the electric field intensity, and $\mathbf{x} = [x\ y\ z]^T$ is a position vector.

Note: From this point onwards the terms *wave* and *field* will be used interchangeably.

Monochromatic plane wave

Now assume $s(\mathbf{x}, t)$ has a complex exponential form,

Monochromatic plane wave

Now assume $s(\mathbf{x}, t)$ has a complex exponential form,

$$s(\mathbf{x}, t) = Ae^{j(\omega t - k_x x - k_y y - k_z z)}$$

where A is a complex constant and k_x, k_y, k_z, and $\omega \geq 0$ are real constants.

Monochromatic plane wave

Substituting the complex exponential form of $s(\mathbf{x}, t)$ into the wave equation, we have

Monochromatic plane wave

Substituting the complex exponential form of $s(\mathbf{x}, t)$ into the wave equation, we have

$$k_x^2 s(\mathbf{x}, t) + k_y^2 s(\mathbf{x}, t) + k_z^2 s(\mathbf{x}, t) = \frac{1}{c^2}\omega^2 s(\mathbf{x}, t)$$

Monochromatic plane wave

Substituting the complex exponential form of $s(\mathbf{x}, t)$ into the wave equation, we have

$$k_x^2 s(\mathbf{x}, t) + k_y^2 s(\mathbf{x}, t) + k_z^2 s(\mathbf{x}, t) = \frac{1}{c^2}\omega^2 s(\mathbf{x}, t)$$

or, after canceling $s(\mathbf{x}, t)$,

Monochromatic plane wave

Substituting the complex exponential form of $s(\mathbf{x}, t)$ into the wave equation, we have

$$k_x^2 s(\mathbf{x}, t) + k_y^2 s(\mathbf{x}, t) + k_z^2 s(\mathbf{x}, t) = \frac{1}{c^2}\omega^2 s(\mathbf{x}, t)$$

or, after canceling $s(\mathbf{x}, t)$,

$$k_x^2 + k_y^2 + k_z^2 = \frac{1}{c^2}\omega^2$$

Monochromatic plane wave

Substituting the complex exponential form of $s(\mathbf{x}, t)$ into the wave equation, we have

$$k_x^2 s(\mathbf{x}, t) + k_y^2 s(\mathbf{x}, t) + k_z^2 s(\mathbf{x}, t) = \frac{1}{c^2}\omega^2 s(\mathbf{x}, t)$$

or, after canceling $s(\mathbf{x}, t)$,

$$\boxed{k_x^2 + k_y^2 + k_z^2 = \frac{1}{c^2}\omega^2}$$

constraints to be satisfied
by the parameters
of the scalar field

Monochromatic plane wave

From the constraints imposed by the complex exponential form, $s(\mathbf{x}, t) = Ae^{j(\omega t - k_x x - k_y y - k_z z)}$ is

Monochromatic plane wave

From the constraints imposed by the complex exponential form, $s(\mathbf{x}, t) = Ae^{j(\omega t - k_x x - k_y y - k_z z)}$ is

- monochromatic

Monochromatic plane wave

From the constraints imposed by the complex exponential form, $s(\mathbf{x}, t) = Ae^{j(\omega t - k_x x - k_y y - k_z z)}$ is

- monochromatic
- plane

Monochromatic plane wave

From the constraints imposed by the complex exponential form, $s(\mathbf{x}, t) = Ae^{j(\omega t - k_x x - k_y y - k_z z)}$ is

- monochromatic
- plane

Monochromatic plane wave

From the constraints imposed by the complex exponential form, $s(\mathbf{x}, t) = Ae^{j(\omega t - k_x x - k_y y - k_z z)}$ is

- monochromatic
- plane

For example, take the position at the origin of the coordinate space:

Monochromatic plane wave

From the constraints imposed by the complex exponential form, $s(\mathbf{x},t) = Ae^{j(\omega t - k_x x - k_y y - k_z z)}$ is

- monochromatic
- plane

For example, take the position at the origin of the coordinate space:

$$\mathbf{x} = [0 \;\; 0 \;\; 0]^T$$

Monochromatic plane wave

From the constraints imposed by the complex exponential form, $s(\mathbf{x}, t) = Ae^{j(\omega t - k_x x - k_y y - k_z z)}$ is

- monochromatic
- plane

For example, take the position at the origin of the coordinate space:

$$\mathbf{x} = [0 \;\; 0 \;\; 0]^T$$

$$s(\mathbf{0}, t) = Ae^{j\omega t}$$

Monochromatic plane wave

From the constraints imposed by the complex exponential form, $s(\mathbf{x}, t) = Ae^{j(\omega t - k_x x - k_y y - k_z z)}$ is

- monochromatic
- plane

Monochromatic plane wave

From the constraints imposed by the complex exponential form, $s(\mathbf{x}, t) = Ae^{j(\omega t - k_x x - k_y y - k_z z)}$ is

- monochromatic
- plane

The value of $s(\mathbf{x}, t)$ is the same for all points lying on the plane

Monochromatic plane wave

From the constraints imposed by the complex exponential form, $s(\mathbf{x}, t) = Ae^{j(\omega t - k_x x - k_y y - k_z z)}$ is

- monochromatic
- plane

The value of $s(\mathbf{x}, t)$ is the same for all points lying on the plane

$$k_x x + k_y y + k_z z = C$$

where C is a constant.

Monochromatic plane wave

Defining the *wavenumber vector* $\mathbf{k}$ as

$$\mathbf{k} = [k_x \;\; k_y \;\; k_z]^T$$

Monochromatic plane wave

Defining the *wavenumber vector* $\mathbf{k}$ as

$$\mathbf{k} = [k_x \;\; k_y \;\; k_z]^T$$

we can rewrite the equation for the monochromatic plane wave as

$$s(\mathbf{x}, t) = Ae^{j(\omega t - \mathbf{k}^T \mathbf{x})}$$

Monochromatic plane wave

Defining the *wavenumber vector* $\mathbf{k}$ as

$$\mathbf{k} = [k_x \;\; k_y \;\; k_z]^T$$

we can rewrite the equation for the monochromatic plane wave as

$$\boxed{s(\mathbf{x}, t) = Ae^{j(\omega t - \mathbf{k}^T \mathbf{x})}}$$

The planes where $s(\mathbf{x}, t)$ is constant are perpendicular to the wavenumber vector $\mathbf{k}$

Monochromatic plane wave

As the plane wave propagates, it advances a distance $\delta\mathbf{x}$ in δt seconds.

Monochromatic plane wave

As the plane wave propagates, it advances a distance $\delta\mathbf{x}$ in δt seconds.

Therefore,

$$s(\mathbf{x}, t) = s(\mathbf{x} + \delta\mathbf{x}, t + \delta t)$$

$$\Longleftrightarrow Ae^{j(\omega t - \mathbf{k}^T\mathbf{x})} = Ae^{j\left[\omega(t+\delta t) - \mathbf{k}^T(\mathbf{x}+\delta\mathbf{x})\right]}$$

Monochromatic plane wave

As the plane wave propagates, it advances a distance $\delta\mathbf{x}$ in δt seconds.

Therefore,

$$s(\mathbf{x}, t) = s(\mathbf{x} + \delta\mathbf{x}, t + \delta t)$$

$$\Longleftrightarrow Ae^{j(\omega t - \mathbf{k}^T\mathbf{x})} = Ae^{j\left[\omega(t+\delta t) - \mathbf{k}^T(\mathbf{x}+\delta\mathbf{x})\right]}$$

$$\Longrightarrow \omega\delta t - \mathbf{k}^T\delta\mathbf{x} = 0$$

Monochromatic plane wave

Naturally the plane wave propagates in the direction of the wavenumber vector, i.e.,

Monochromatic plane wave

Naturally the plane wave propagates in the direction of the wavenumber vector, i.e.,

$\mathbf{k}$ and $\delta\mathbf{x}$ point in the same direction.

Monochromatic plane wave

Naturally the plane wave propagates in the direction of the wavenumber vector, i.e.,

$\mathbf{k}$ and $\delta\mathbf{x}$ point in the same direction.

Therefore,

$$\mathbf{k}^T\delta\mathbf{x} = \|\mathbf{k}\|\|\delta\mathbf{x}\|$$

Monochromatic plane wave

Naturally the plane wave propagates in the direction of the wavenumber vector, i.e.,

$\mathbf{k}$ and $\delta\mathbf{x}$ point in the same direction.

Therefore,

$$\mathbf{k}^T\delta\mathbf{x} = \|\mathbf{k}\|\|\delta\mathbf{x}\|$$

$$\Longrightarrow \omega\delta t = \|\mathbf{k}\|\|\delta\mathbf{x}\|$$

Monochromatic plane wave

Naturally the plane wave propagates in the direction of the wavenumber vector, i.e.,

$\mathbf{k}$ and $\delta\mathbf{x}$ point in the same direction.

Therefore,

$$\mathbf{k}^T\delta\mathbf{x} = \|\mathbf{k}\|\|\delta\mathbf{x}\|$$

$$\Longrightarrow \omega\delta t = \|\mathbf{k}\|\|\delta\mathbf{x}\|$$

or, equivalently,

Monochromatic plane wave

Naturally the plane wave propagates in the direction of the wavenumber vector, i.e.,

$\mathbf{k}$ and $\delta\mathbf{x}$ point in the same direction.

Therefore,

$$\mathbf{k}^T\delta\mathbf{x} = \|\mathbf{k}\|\|\delta\mathbf{x}\|$$

$$\Longrightarrow \omega\delta t = \|\mathbf{k}\|\|\delta\mathbf{x}\|$$

or, equivalently,

$$\frac{\|\delta\mathbf{x}\|}{\delta t} = \frac{\omega}{\|\mathbf{k}\|}$$

Monochromatic plane wave

Naturally the plane wave propagates in the direction of the wavenumber vector, i.e.,

$\mathbf{k}$ and $\delta\mathbf{x}$ point in the same direction.

Therefore,

$$\mathbf{k}^T\delta\mathbf{x} = \|\mathbf{k}\|\|\delta\mathbf{x}\|$$

$$\Longrightarrow \omega\delta t = \|\mathbf{k}\|\|\delta\mathbf{x}\|$$

or, equivalently,

$$\frac{\|\delta\mathbf{x}\|}{\delta t} = \frac{\omega}{\|\mathbf{k}\|}$$

Remember the constraints?
$\|\mathbf{k}\|^2 = \omega^2/c^2$

Monochromatic plane wave

After $T = 2\pi/\omega$ seconds, the plane wave has completed one cycle and it appears as it did before, but its *wavefront* has advanced a distance of one *wavelength*, λ.

Monochromatic plane wave

After $T = 2\pi/\omega$ seconds, the plane wave has completed one cycle and it appears as it did before, but its *wavefront* has advanced a distance of one *wavelength*, λ.

For $\|\delta\mathbf{x}\| = \lambda$ and $\delta t = T = \frac{2\pi}{\omega}$,

$$T = \frac{\lambda\|\mathbf{k}\|}{\omega} \implies \lambda = \frac{2\pi}{\|\mathbf{k}\|}$$

Monochromatic plane wave

After $T = 2\pi/\omega$ seconds, the plane wave has completed one cycle and it appears as it did before, but its *wavefront* has advanced a distance of one *wavelength*, λ.

For $\|\delta\mathbf{x}\| = \lambda$ and $\delta t = \boxed{T = \frac{2\pi}{\omega}}$,

$$T = \frac{\lambda\|\mathbf{k}\|}{\omega} \Longrightarrow \boxed{\lambda = \frac{2\pi}{\|\mathbf{k}\|}}$$

The wavenumber vector, $\mathbf{k}$, may be considered a *spatial frequency* variable, just as ω is a *temporal frequency* variable.

Monochromatic plane wave

We may rewrite the wave equation as

$$\begin{aligned} s(\mathbf{x}, t) &= Ae^{j(\omega t - \mathbf{k}^T \mathbf{x})} \\ &= Ae^{j\omega(t - \boldsymbol{\alpha}^T \mathbf{x})} \end{aligned}$$

where $\boldsymbol{\alpha} = \mathbf{k}/\omega$ is the *slowness vector*.

Monochromatic plane wave

We may rewrite the wave equation as

$$\begin{aligned} s(\mathbf{x}, t) &= Ae^{j(\omega t - \mathbf{k}^T \mathbf{x})} \\ &= Ae^{j\omega(t - \boldsymbol{\alpha}^T \mathbf{x})} \end{aligned}$$

where $\boldsymbol{\alpha} = \mathbf{k}/\omega$ is the *slowness vector*.

As $c = \omega/\|\mathbf{k}\|$, vector $\boldsymbol{\alpha}$ has a magnitude which is the reciprocal of c.

Periodic propagating periodic waves

Any arbitrary periodic waveform $s(\mathbf{x}, t) = s(t - \boldsymbol{\alpha}^T \mathbf{x})$ with fundamental period ω_0 can be represented as a sum:

$$s(\mathbf{x}, t) = s(t - \boldsymbol{\alpha}^T \mathbf{x}) = \sum_{n=-\infty}^{\infty} S_n e^{jn\omega_0 (t - \boldsymbol{\alpha}^T \mathbf{x})}$$

Periodic propagating periodic waves

Any arbitrary periodic waveform $s(\mathbf{x},t) = s(t - \boldsymbol{\alpha}^T\mathbf{x})$ with fundamental period ω_0 can be represented as a sum:

$$s(\mathbf{x},t) = s(t - \boldsymbol{\alpha}^T\mathbf{x}) = \sum_{n=-\infty}^{\infty} S_n e^{jn\omega_0(t-\boldsymbol{\alpha}^T\mathbf{x})}$$

The coefficients are given by

$$S_n = \frac{1}{T}\int_0^T s(u)e^{-jn\omega_0 u}du$$

Periodic propagating periodic waves

Based on the previous derivations, we observe that:

Periodic propagating periodic waves

Based on the previous derivations, we observe that:

- The various components of $s(\mathbf{x}, t)$ have different frequencies $\omega = n\omega_0$ and different wavenumber vectors, $\mathbf{k}$.

Periodic propagating periodic waves

Based on the previous derivations, we observe that:

- The various components of $s(\mathbf{x}, t)$ have different frequencies $\omega = n\omega_0$ and different wavenumber vectors, $\mathbf{k}$.
- The waveform propagates in the direction of the slowness vector $\boldsymbol{\alpha} = \mathbf{k}/\omega$.

Nonperiodic propagating waves

More generally, any function constructed as the integral of complex exponentials who also have a defined and converged Fourier transform can represent a waveform

Nonperiodic propagating waves

More generally, any function constructed as the integral of complex exponentials who also have a defined and converged Fourier transform can represent a waveform

$$s(\mathbf{x}, t) = s(t - \boldsymbol{\alpha}^T \mathbf{x}) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S(\omega) e^{j\omega(t - \boldsymbol{\alpha}^T \mathbf{x})} d\omega$$

Nonperiodic propagating waves

More generally, any function constructed as the integral of complex exponentials who also have a defined and converged Fourier transform can represent a waveform

$$s(\mathbf{x},t) = s(t - \boldsymbol{\alpha}^T\mathbf{x}) = \frac{1}{2\pi}\int_{-\infty}^{\infty} S(\omega)e^{j\omega(t-\boldsymbol{\alpha}^T\mathbf{x})}d\omega$$

where

$$S(\omega) = \int_{-\infty}^{\infty} s(u)e^{-j\omega u}du$$

Nonperiodic propagating waves

More generally, any function constructed as the integral of complex exponentials who also have a defined and converged Fourier transform can represent a waveform

$$s(\mathbf{x},t) = s(t - \boldsymbol{\alpha}^T\mathbf{x}) = \frac{1}{2\pi}\int_{-\infty}^{\infty} S(\omega)e^{j\omega(t-\boldsymbol{\alpha}^T\mathbf{x})}d\omega$$

where

$$S(\omega) = \int_{-\infty}^{\infty} s(u)e^{-j\omega u}du$$

We will come back to this later...

Outline

2. Sensor Arrays and Spatial Filtering

2.1 Wavenumber-Frequency Space

Space-time Fourier Transform

The four-dimensional Fourier transform of the space-time signal $s(\mathbf{x}, t)$ is given by

Space-time Fourier Transform

The four-dimensional Fourier transform of the space-time signal $s(\mathbf{x}, t)$ is given by

$$S(\mathbf{k}, \omega) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} s(\mathbf{x}, t) e^{-j(\omega t - \mathbf{k}^T \mathbf{x})} d\mathbf{x} dt$$

Space-time Fourier Transform

The four-dimensional Fourier transform of the space-time signal $s(\mathbf{x}, t)$ is given by

$$S(\mathbf{k}, \omega) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} s(\mathbf{x}, t) e^{-j(\omega t - \mathbf{k}^T \mathbf{x})} d\mathbf{x} dt$$

temporal frequency

Space-time Fourier Transform

The four-dimensional Fourier transform of the space-time signal $s(\mathbf{x}, t)$ is given by

$$S(\mathbf{k}, \omega) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} s(\mathbf{x}, t) e^{-j(\omega t - \mathbf{k}^T \mathbf{x})} d\mathbf{x} dt$$

temporal frequency

spatial frequency: wavenumber

Space-time Fourier Transform

The four-dimensional Fourier transform of the space-time signal $s(\mathbf{x}, t)$ is given by

$$S(\mathbf{k}, \omega) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} s(\mathbf{x}, t) e^{-j(\omega t - \mathbf{k}^T \mathbf{x})} d\mathbf{x}\, dt$$

$$s(\mathbf{x}, t) = \frac{1}{(2\pi)^4} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} S(\mathbf{k}, \omega) e^{j(\omega t - \mathbf{k}^T \mathbf{x})} d\mathbf{k}\, d\omega$$

Space-time Fourier Transform

We have already concluded that if the space-time signal is a propagating waveform such that $s(\mathbf{x}, t) = s(t - \boldsymbol{\alpha}_0^T \mathbf{x})$, then its Fourier transform is equal to

$$S(\mathbf{k}, \omega) = S(\omega)\delta(\mathbf{k} - \omega\boldsymbol{\alpha}_0)$$

Space-time Fourier Transform

We have already concluded that if the space-time signal is a propagating waveform such that $s(\mathbf{x},t)=s(t-\boldsymbol{\alpha}_0^T\mathbf{x})$, then its Fourier transform is equal to

$$S(\mathbf{k},\omega)=S(\omega)\delta(\mathbf{k}-\omega\boldsymbol{\alpha}_0)$$

Remember the nonperiodic propagating wave Fourier transform?

Space-time Fourier Transform

We have already concluded that if the space-time signal is a propagating waveform such that $s(\mathbf{x}, t) = s(t - \boldsymbol{\alpha}_0^T \mathbf{x})$, then its Fourier transform is equal to

$$S(\mathbf{k}, \omega) = S(\omega)\delta(\mathbf{k} - \omega\boldsymbol{\alpha}_0)$$

Remember the nonperiodic propagating wave Fourier transform?

This means that $s(\mathbf{x}, t)$ only has energy along the direction of $\mathbf{k} = \mathbf{k}_0 = \omega\boldsymbol{\alpha}_0$ in the wavenumber-frequency space.

2.2 Frequency-Wavenumber (WN) Response and Beam patterns (BP)

Signals at the sensors

- An array is a set of N (isotropic) sensors located at positions $\boldsymbol{p}_n, n = 0, 1, \cdots, N-1$

Signals at the sensors

- An array is a set of N (isotropic) sensors located at positions $\boldsymbol{p}_n, n = 0, 1, \cdots, N-1$
- The sensors spatially sample the signal field at locations $\boldsymbol{p}_n$

Signals at the sensors

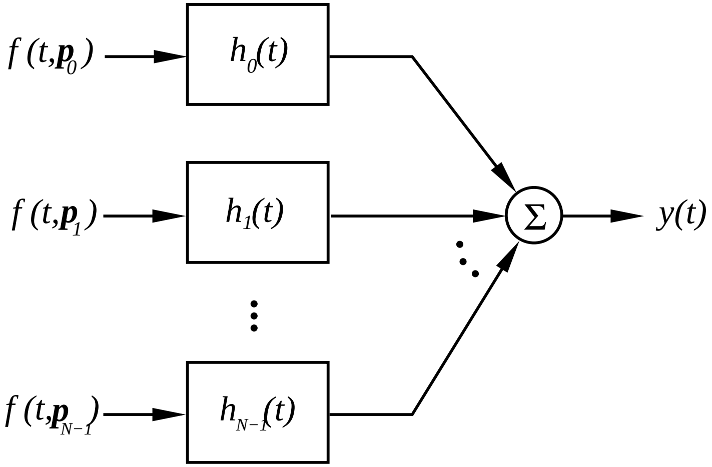
- An array is a set of N (isotropic) sensors located at positions $\boldsymbol{p}_n, n = 0, 1, \cdots, N-1$
- The sensors spatially sample the signal field at locations $\boldsymbol{p}_n$
- At the sensors, the set of N signals are denoted by

Signals at the sensors

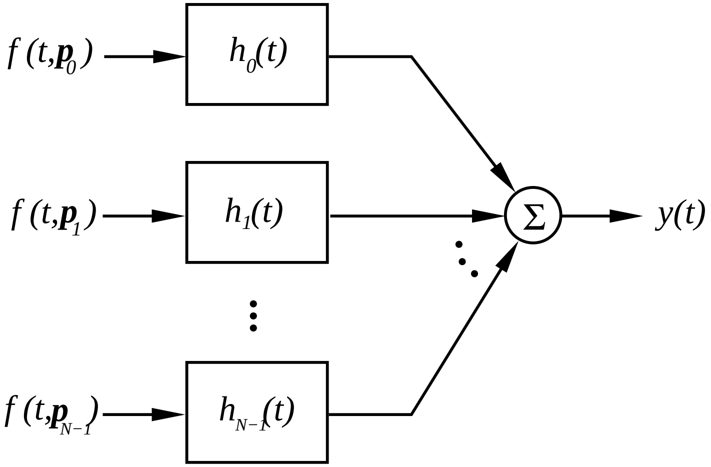
- An array is a set of N (isotropic) sensors located at positions $\boldsymbol{p}_n, n = 0, 1, \cdots, N-1$
- The sensors spatially sample the signal field at locations $\boldsymbol{p}_n$
- At the sensors, the set of N signals are denoted by

$$\boldsymbol{f}(t, \boldsymbol{p}) = \begin{bmatrix} f(t, \boldsymbol{p}_0) \\ f(t, \boldsymbol{p}_1) \\ \vdots \\ f(t, \boldsymbol{p}_{N-1}) \end{bmatrix}$$

Array output

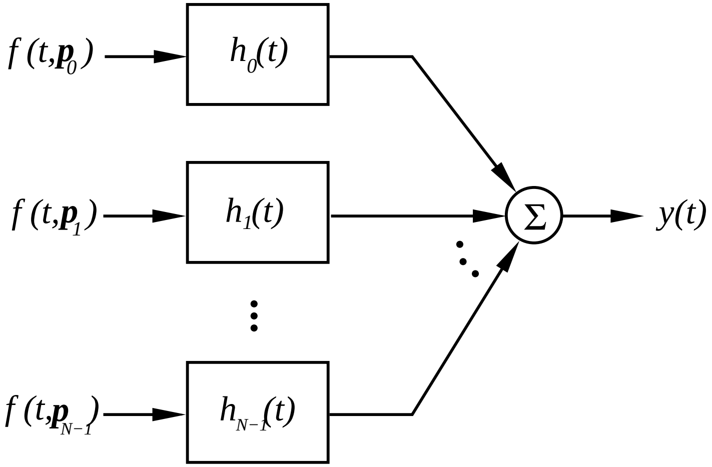


Array output



$$y(t) = \sum_{n=0}^{N-1} \int_{-\infty}^{\infty} h_n(t-\tau) f_n(\tau, \boldsymbol{p}_n) d\tau$$
$$= \int_{-\infty}^{\infty} \boldsymbol{h}^T(t-\tau) \boldsymbol{f}(\tau, \boldsymbol{p}) d\tau$$

Array output



$$y(t) = \sum_{n=0}^{N-1} \int_{-\infty}^{\infty} h_n(t-\tau) f_n(\tau, \boldsymbol{p}_n) d\tau$$

$$= \int_{-\infty}^{\infty} \boldsymbol{h}^T(t-\tau) \boldsymbol{f}(\tau, \boldsymbol{p}) d\tau$$

where $\boldsymbol{h}(t) = [h_o(t) \;\; h_1(t) \;\; \cdots \;\; h_{N-1}(t)]^T$

In the frequency domain, $\cdots$

$$\begin{aligned} Y(\omega) &= \int_{-\infty}^{\infty} y(t) e^{-j\omega t} dt \\ &= \boldsymbol{H}^T(\omega) \boldsymbol{F}(\omega) \end{aligned}$$

In the frequency domain, $\cdots$

$$\begin{aligned} Y(\omega) &= \int_{-\infty}^{\infty} y(t) e^{-j\omega t} dt \\ &= \boldsymbol{H}^T(\omega) \boldsymbol{F}(\omega) \end{aligned}$$

where

$$\boldsymbol{H}(\omega) = \int_{-\infty}^{\infty} \boldsymbol{h}(t) e^{-j\omega t} dt$$

$$\boldsymbol{F}(\omega) = \boldsymbol{F}(\omega, \boldsymbol{p}) = \int_{-\infty}^{\infty} \boldsymbol{f}(t, \boldsymbol{p}) e^{-j\omega t} dt$$

Plane wave propagating $\cdots$

- Consider a plane wave propagating in the direction of vector $\boldsymbol{a}$:

$$\boldsymbol{a} = \begin{bmatrix} -sin\theta cos\phi \\ -sin\theta sin\phi \\ -cos\theta \end{bmatrix}$$

Plane wave propagating $\cdots$

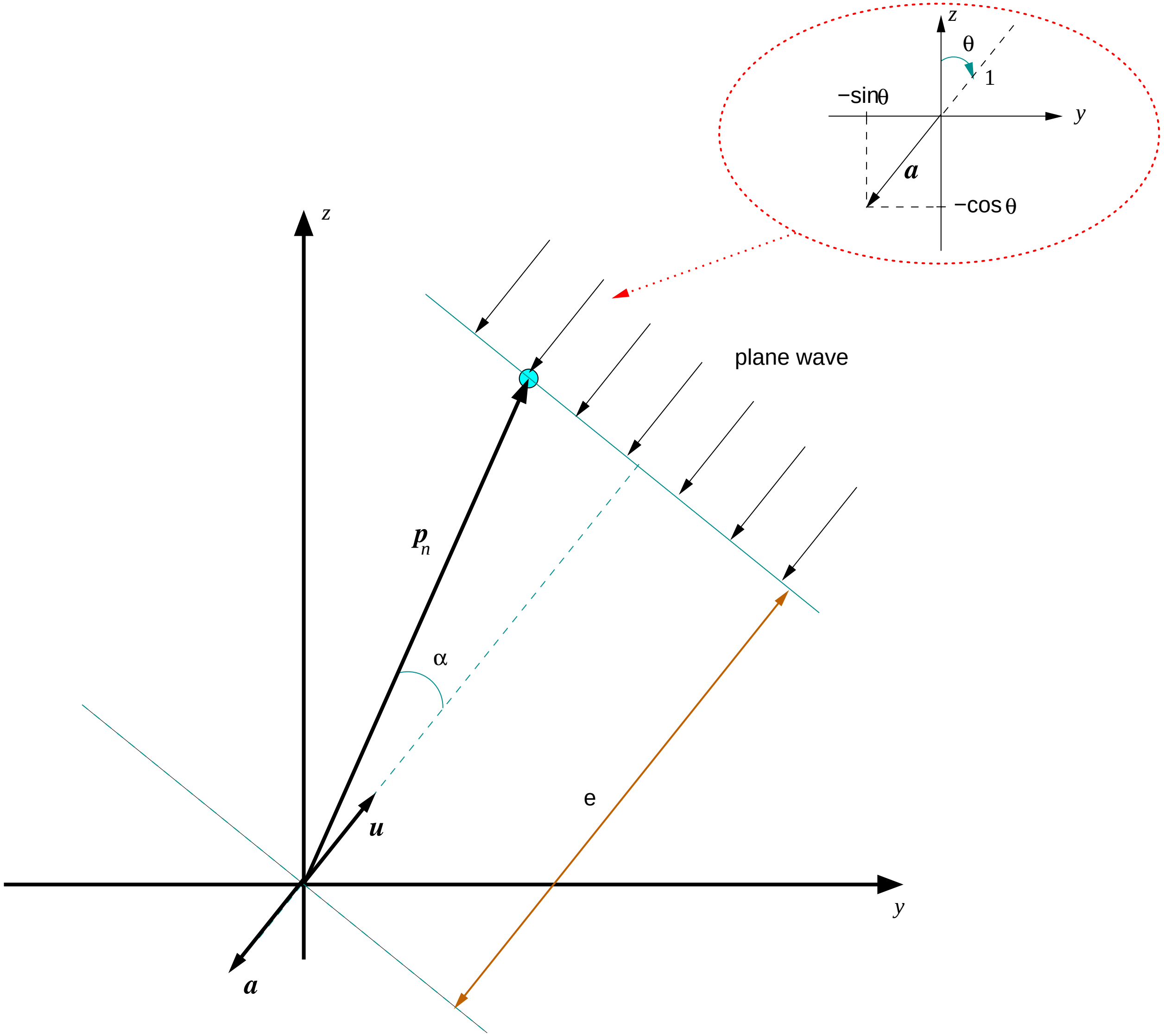
- Consider a plane wave propagating in the direction of vector $\boldsymbol{a}$:

$$\boldsymbol{a} = \begin{bmatrix} -sin\theta cos\phi \\ -sin\theta sin\phi \\ -cos\theta \end{bmatrix}$$

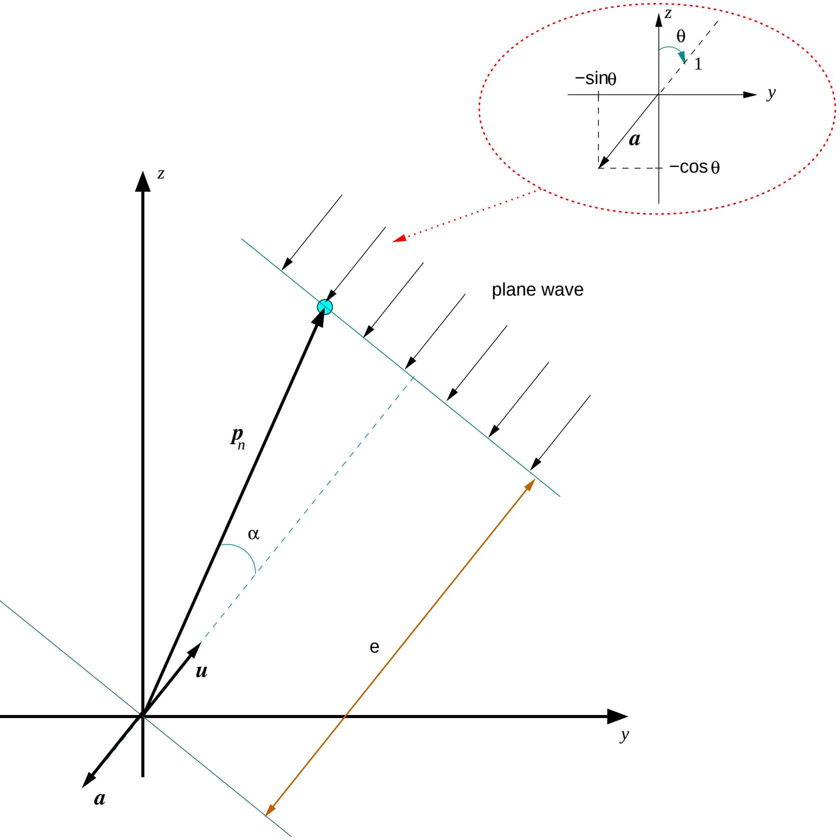
- If $f(t)$ is the signal that would be received at the origin, then:

$$\boldsymbol{f}(t, \boldsymbol{p}) = \begin{bmatrix} f(t-\tau_0) \\ f(t-\tau_1) \\ \vdots \\ f(t-\tau_{N-1}) \end{bmatrix}$$

Plane wave (assuming $\phi = 90^o$)

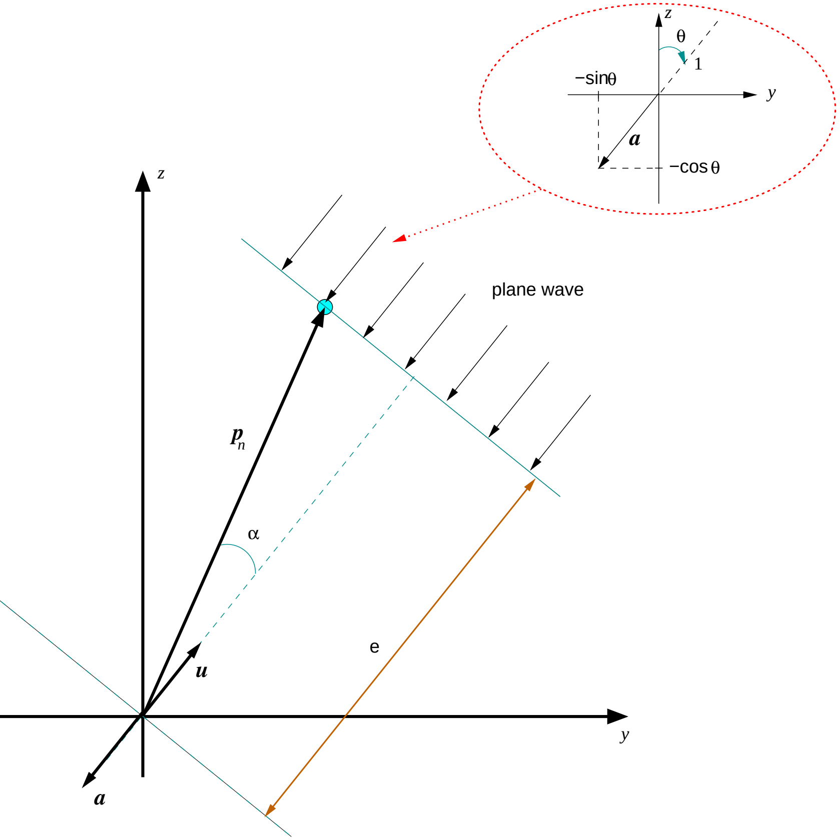


Plane wave (assuming $\phi = 90^o$)



$e = ?$

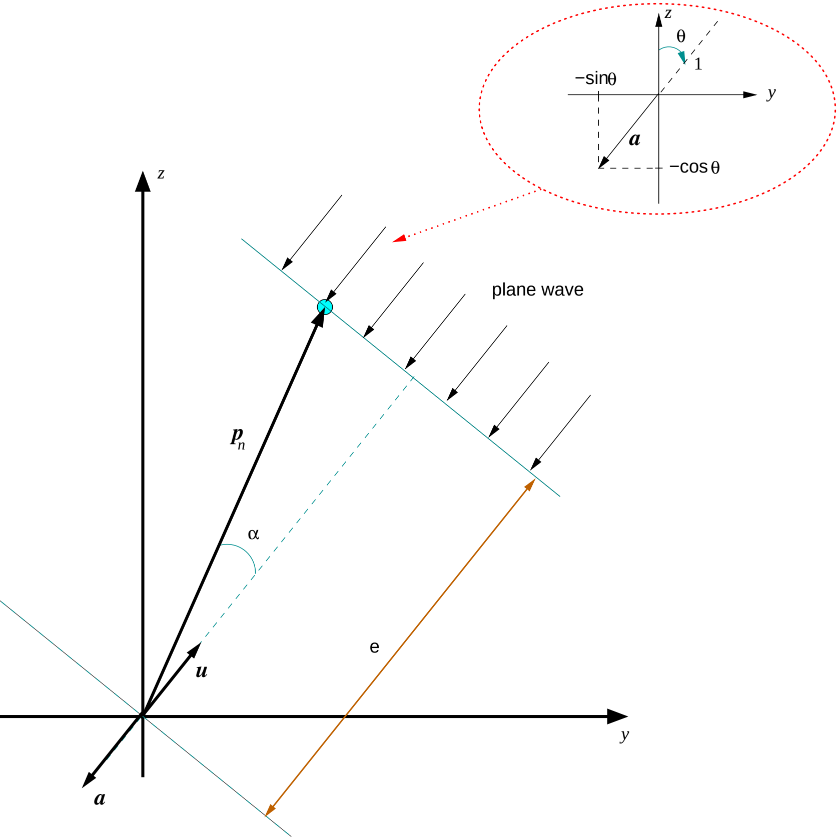
Plane wave (assuming $\phi = 90^o$)



$e = ?$

$e = c\tau_n$

Plane wave (assuming $\phi = 90^o$)

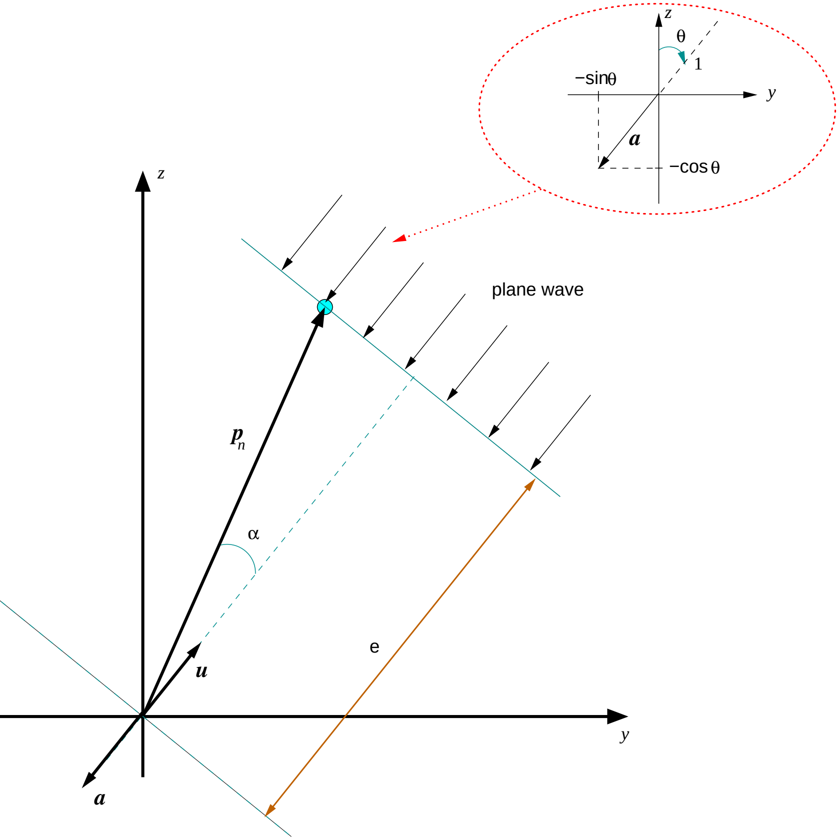


$$e = ?$$

$$e = c\tau_n$$

$$\Rightarrow \tau_n = \frac{e}{c}$$

Plane wave (assuming $\phi = 90^o$)



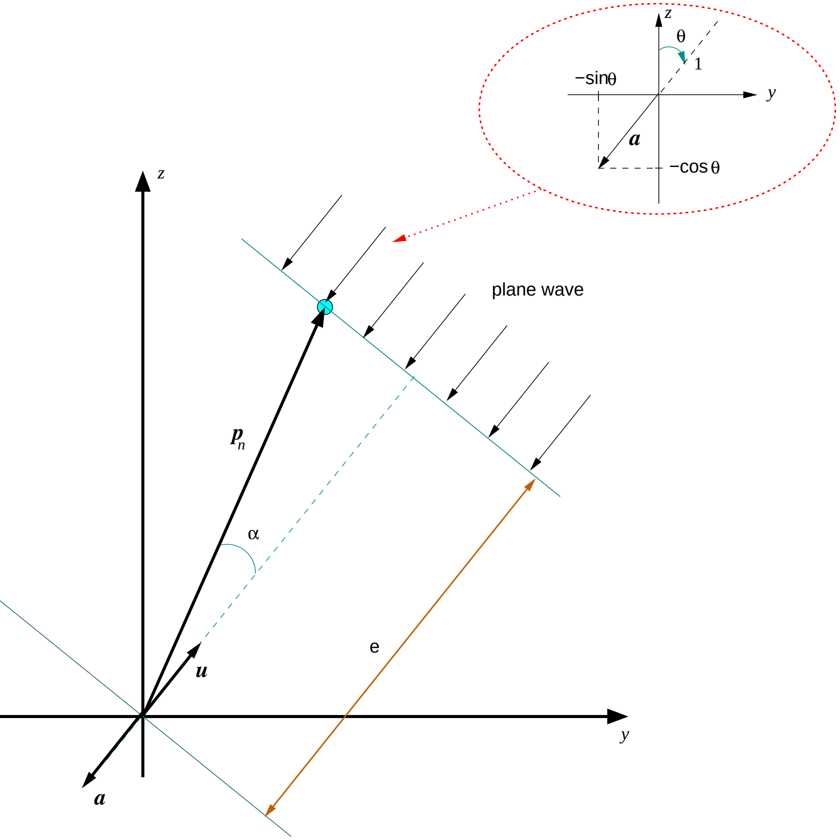
$$e = ?$$

$$e = c\tau_n$$

$$\Rightarrow \tau_n = \frac{e}{c}$$

BUT

Plane wave (assuming $\phi = 90^o$)



$$e = ?$$

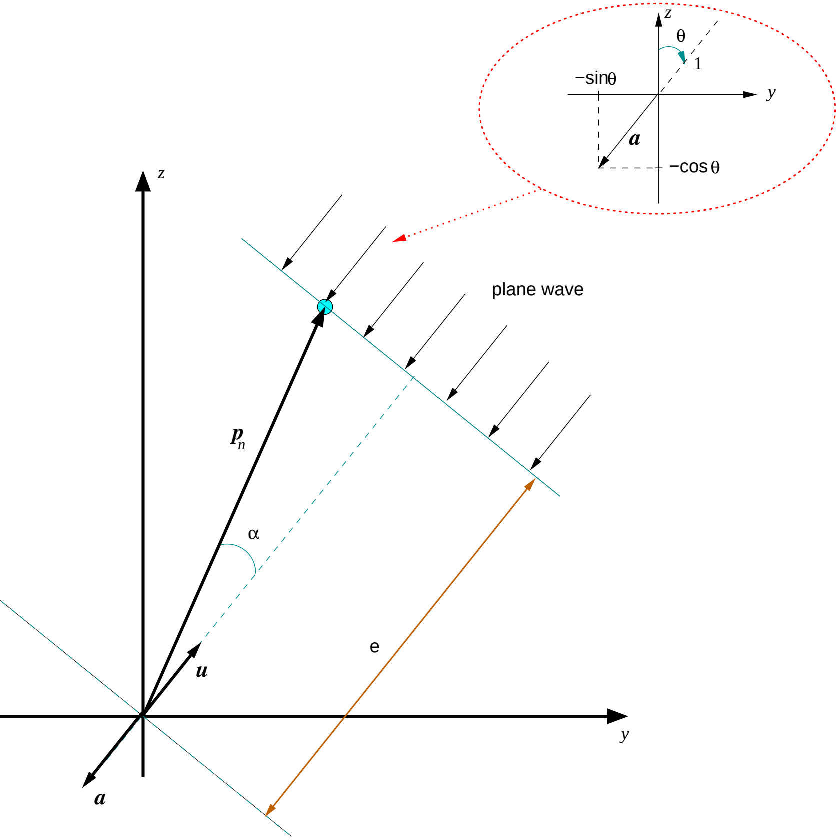
$$e = c\tau_n$$

$$\Rightarrow \tau_n = \frac{e}{c}$$

BUT

$$e = \|\mathbf{p}_n\| cos(\alpha) = \underbrace{\|\mathbf{u}\|}_{=1} \|\mathbf{p}_n\| cos(\alpha)$$

Plane wave (assuming $\phi = 90^o$)



$$e = ?$$

$$e = c\tau_n$$

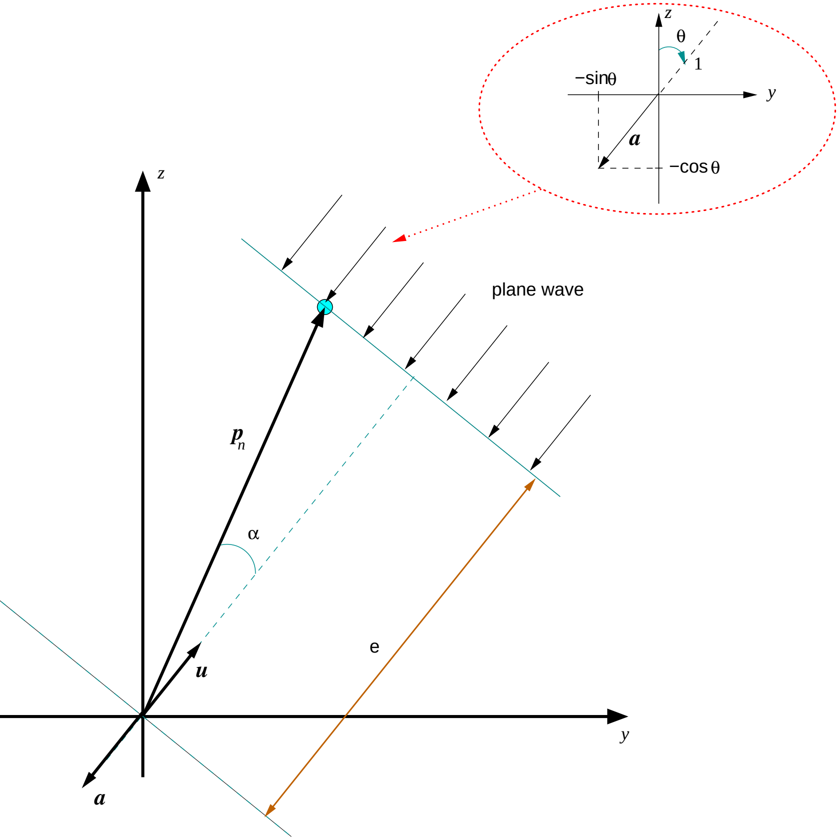
$$\Rightarrow \tau_n = \frac{e}{c}$$

BUT

$$e = \|\mathbf{p}_n\| cos(\alpha) = \underbrace{\|\mathbf{u}\|}_{=1} \|\mathbf{p}_n\| cos(\alpha)$$

$$\therefore \tau_n = -\frac{\mathbf{u}^T\mathbf{p}_n}{c} = \frac{\mathbf{a}^T\mathbf{p}_n}{c}$$

Plane wave (assuming $\phi = 90^o$)



$e = ?$

$e = c\tau_n$

$\Rightarrow \tau_n = \frac{e}{c}$

BUT

$$e = \|\mathbf{p}_n\| cos(\alpha) = \underbrace{\|\mathbf{u}\|}_{=1} \|\mathbf{p}_n\| cos(\alpha)$$

$$\therefore \tau_n = -\frac{\mathbf{u}^T\mathbf{p}_n}{c} = \frac{\mathbf{a}^T\mathbf{p}_n}{c}$$

$_n$ is the time since the plane wave hits the sensor at location $\mathbf{p}_n$ until it reaches point $(0,0)$.

Back to the frequency domain

- Then, we have:

$$\boldsymbol{F}(\omega) = \begin{bmatrix} \int_{-\infty}^{\infty} e^{-j\omega t} f(t-\tau_0)dt \\ \int_{-\infty}^{\infty} e^{-j\omega t} f(t-\tau_1)dt \\ \vdots \\ \int_{-\infty}^{\infty} e^{-j\omega t} f(t-\tau_{N-1})dt \end{bmatrix}$$

$$= \begin{bmatrix} e^{-j\omega\tau_0} \\ e^{-j\omega\tau_1} \\ \vdots \\ e^{-j\omega\tau_{N-1}} \end{bmatrix} F(\omega)$$

Definition of Wavenumber

- For plane waves propagating in a locally homogeneous medium:

$$\boldsymbol{k} = \frac{\omega}{c}\boldsymbol{a} = \frac{2\pi}{c/f}\boldsymbol{a} = \frac{2\pi}{\lambda}\boldsymbol{a} = -\frac{2\pi}{\lambda}\boldsymbol{u}$$

Definition of Wavenumber

- For plane waves propagating in a locally homogeneous medium:

$$\boldsymbol{k} = \frac{\omega}{c}\boldsymbol{a} = \frac{2\pi}{c/f}\boldsymbol{a} = \frac{2\pi}{\lambda}\boldsymbol{a} = -\frac{2\pi}{\lambda}\boldsymbol{u}$$

Wavenumber Vector ("spatial frequency")

Definition of Wavenumber

- For plane waves propagating in a locally homogeneous medium:

$$\boldsymbol{k} = \frac{\omega}{c}\boldsymbol{a} = \frac{2\pi}{c/f}\boldsymbol{a} = \frac{2\pi}{\lambda}\boldsymbol{a} = -\frac{2\pi}{\lambda}\boldsymbol{u}$$

Wavenumber Vector ("spatial frequency")

- Note that $|\boldsymbol{k}| = \frac{2\pi}{\lambda}$

Definition of Wavenumber

- For plane waves propagating in a locally homogeneous medium:

$$\boldsymbol{k} = \frac{\omega}{c}\boldsymbol{a} = \frac{2\pi}{c/f}\boldsymbol{a} = \frac{2\pi}{\lambda}\boldsymbol{a} = -\frac{2\pi}{\lambda}\boldsymbol{u}$$

Wavenumber Vector ("spatial frequency")

- Note that $|\boldsymbol{k}| = \frac{2\pi}{\lambda}$
- Therefore

$$\omega\tau_n = \frac{\omega}{c}\boldsymbol{a}^T\boldsymbol{p}_n = \boldsymbol{k}^T\boldsymbol{p}_n$$

Array Manifold Vector

- And we have

$$\boldsymbol{F}(\omega) = \begin{bmatrix} e^{-j\boldsymbol{k}^T\boldsymbol{p}_0} \\ e^{-j\boldsymbol{k}^T\boldsymbol{p}_1} \\ \vdots \\ e^{-j\boldsymbol{k}^T\boldsymbol{p}_{N-1}} \end{bmatrix} F(\omega) = F(\omega)\boldsymbol{v}_{\boldsymbol{k}}(\boldsymbol{k})$$

Array Manifold Vector

- And we have

$$\boldsymbol{F}(\omega) = \begin{bmatrix} e^{-j\boldsymbol{k}^T\boldsymbol{p}_0} \\ e^{-j\boldsymbol{k}^T\boldsymbol{p}_1} \\ \vdots \\ e^{-j\boldsymbol{k}^T\boldsymbol{p}_{N-1}} \end{bmatrix} F(\omega) = F(\omega)\underbrace{\boldsymbol{v}_{\boldsymbol{k}}(\boldsymbol{k})}_{\text{Array Manifold Vector}}$$

Array Manifold Vector

- And we have

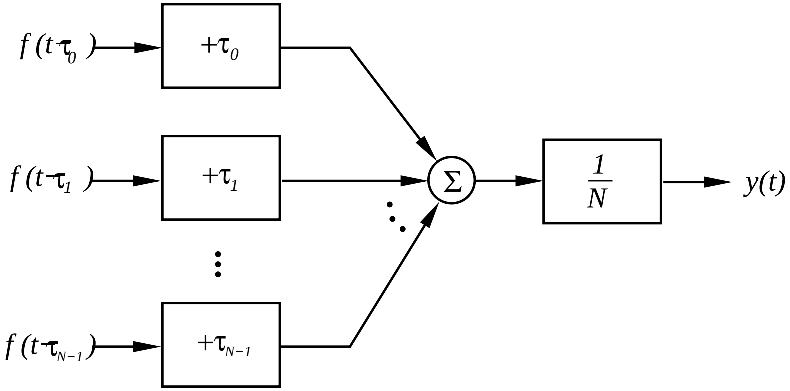
$$\boldsymbol{F}(\omega) = \begin{bmatrix} e^{-j\boldsymbol{k}^T\boldsymbol{p}_0} \\ e^{-j\boldsymbol{k}^T\boldsymbol{p}_1} \\ \vdots \\ e^{-j\boldsymbol{k}^T\boldsymbol{p}_{N-1}} \end{bmatrix} F(\omega) = F(\omega)\underbrace{\boldsymbol{v}_{\boldsymbol{k}}(\boldsymbol{k})}_{\text{Array Manifold Vector}}$$

- In this particular example, we can use $h_n(t) = \frac{1}{N}\delta(t + \tau_n)$ such that

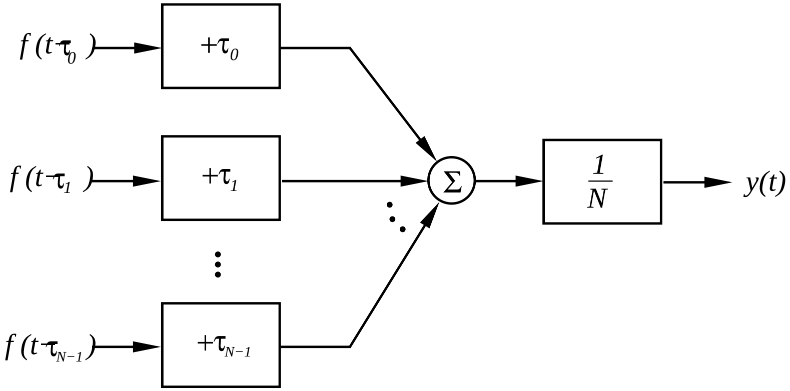
$$y(t) = f(t)$$

  Following, we have the delay-and-sum beamformer.

Delay-and-sum Beamformer

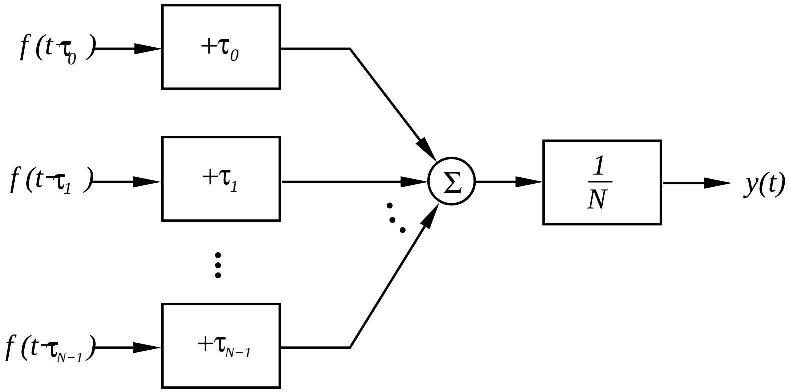


Delay-and-sum Beamformer



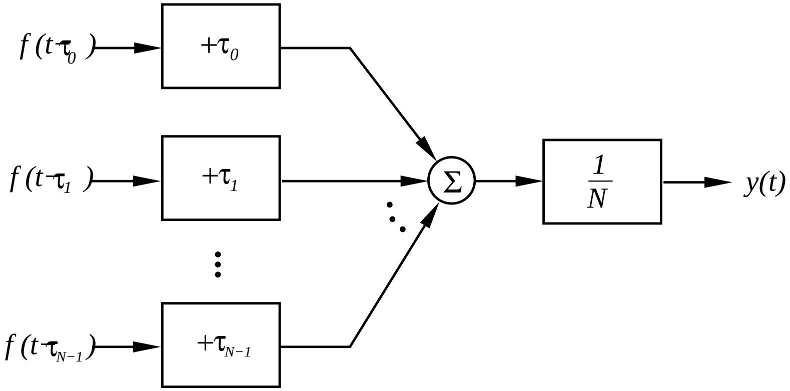
- A common delay is added in each channel to make the operations physically realizable

Delay-and-sum Beamformer



- A common delay is added in each channel to make the operations physically realizable
- Since $\mathcal{F}\left\{h_n(t)\right\} = \mathcal{F}\left\{\frac{1}{N}\delta(t+\tau_n)\right\} = e^{j\omega\tau_n}$

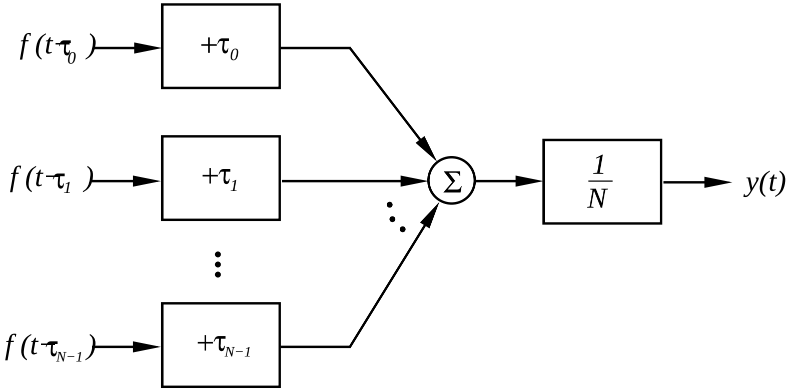
Delay-and-sum Beamformer



- A common delay is added in each channel to make the operations physically realizable
- Since $\mathcal{F}\left\{h_n(t)\right\} = \mathcal{F}\left\{\frac{1}{N}\delta(t+\tau_n)\right\} = e^{j\omega\tau_n}$
- We can write

$$\boldsymbol{H}^T(\omega) = \frac{1}{N}\boldsymbol{v}_{\boldsymbol{k}}^H(\boldsymbol{k})$$

Delay-and-sum Beamformer

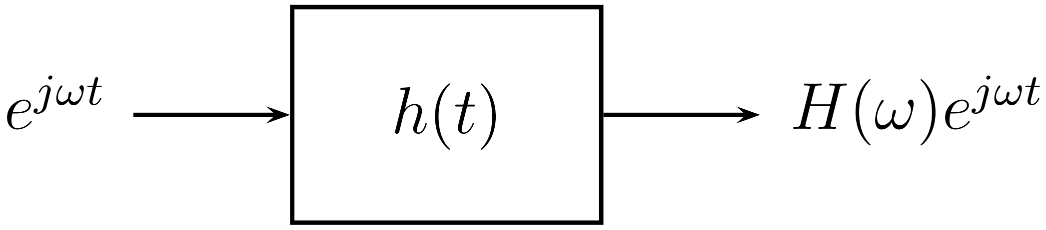


- A common delay is added in each channel to make the operations physically realizable
- Since $\mathcal{F}\left\{h_n(t)\right\} = \mathcal{F}\left\{\frac{1}{N}\delta(t+\tau_n)\right\} = e^{j\omega\tau_n}$
- We can write

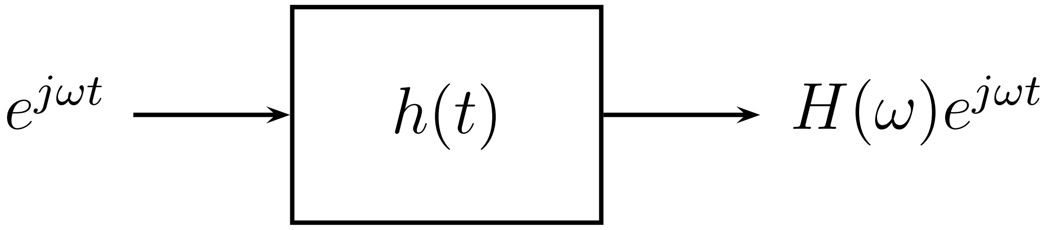
Array Manifold Vector

$$\boldsymbol{H}^T(\omega) = \frac{1}{N}\boxed{\boldsymbol{v}_{\boldsymbol{k}}^H(\boldsymbol{k})}$$

LTI System



LTI System



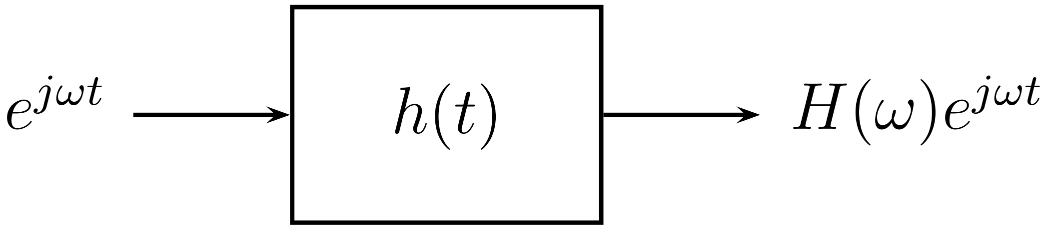
$$e^{j\omega t} \longrightarrow \boxed{h(t)} \longrightarrow H(\omega)e^{j\omega t}$$

- Space-time signals (base functions):

$$f_n(t, \boldsymbol{p}) = e^{j\omega(t-\tau_n)} = e^{j(\omega t - \boldsymbol{k}^T\boldsymbol{p}_n)}$$

Note that $\omega\tau_n = \boldsymbol{k}^T\boldsymbol{p}_n$

LTI System



- Space-time signals (base functions):

$$f_n(t,\boldsymbol{p}) = e^{j\omega(t-\tau_n)} = e^{j(\omega t - \boldsymbol{k}^T\boldsymbol{p}_n)}$$

Note that $\omega\tau_n = \boldsymbol{k}^T\boldsymbol{p}_n$

- $\therefore\ \boldsymbol{f}(t,\boldsymbol{p}) = e^{j\omega t}\boldsymbol{v}_{\boldsymbol{k}}(\boldsymbol{k})$

Frequency-Wavenumber Response Function

- The response of the array to this plane wave is:

$$y(t, \boldsymbol{k}) = \boldsymbol{H}^T(\omega)\boldsymbol{v}_{\boldsymbol{k}}(\boldsymbol{k})e^{j\omega t}$$

Frequency-Wavenumber Response Function

- The response of the array to this plane wave is:

$$y(t, \boldsymbol{k}) = \boldsymbol{H}^T(\omega)\boldsymbol{v}_{\boldsymbol{k}}(\boldsymbol{k})e^{j\omega t}$$

- After taking the Fourier transform, we have:

$$Y(\omega, \boldsymbol{k}) = \boldsymbol{H}^T(\omega)\boldsymbol{v}_{\boldsymbol{k}}(\boldsymbol{k})$$

Frequency-Wavenumber Response Function

- The response of the array to this plane wave is:

$$y(t, \boldsymbol{k}) = \boldsymbol{H}^T(\omega)\boldsymbol{v}_{\boldsymbol{k}}(\boldsymbol{k})e^{j\omega t}$$

- After taking the Fourier transform, we have:

$$Y(\omega, \boldsymbol{k}) = \boldsymbol{H}^T(\omega)\boldsymbol{v}_{\boldsymbol{k}}(\boldsymbol{k})$$

- And we define the Frequency-Wavenumber Response Function:

Upsilon

$$\Upsilon(\omega, \boldsymbol{k}) \triangleq \boldsymbol{H}^T(\omega)\boldsymbol{v}_{\boldsymbol{k}}(\boldsymbol{k})$$

$\Upsilon(\omega, \boldsymbol{k})$ describes the complex gain of an array to an input plane wave with wavenumber $\boldsymbol{k}$ and temporal frequency ω.

Beam Pattern and Bandpass Signal

- BEAM PATTERN is the Frequency Wavenumber Response Function evaluated versus the direction:

$$B(\omega : \theta, \phi) = \Upsilon(\omega, \boldsymbol{k})$$

Note that $\boldsymbol{k} = \frac{2\pi}{\lambda}\boldsymbol{a}(\theta, \phi)$, and $\boldsymbol{a}$ is the unit vector with spherical coordinates angles θ and ϕ

Beam Pattern and Bandpass Signal

- BEAM PATTERN is the Frequency Wavenumber Response Function evaluated versus the direction:

$$B(\omega : \theta, \phi) = \Upsilon(\omega, \boldsymbol{k})$$

Note that $\boldsymbol{k} = \frac{2\pi}{\lambda}\boldsymbol{a}(\theta, \phi)$, and $\boldsymbol{a}$ is the unit vector with spherical coordinates angles θ and ϕ

- Let's write a bandpass signal:

$$f(t, \boldsymbol{p}_n) = \sqrt{2} Re\{\tilde{f}(t, \boldsymbol{p}_n) e^{j\omega_c t}\}, n = 0, 1, \cdots, N-1$$

Beam Pattern and Bandpass Signal

- BEAM PATTERN is the Frequency Wavenumber Response Function evaluated versus the direction:

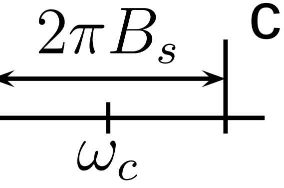
$$B(\omega : \theta, \phi) = \Upsilon(\omega, \boldsymbol{k})$$

  Note that $\boldsymbol{k} = \frac{2\pi}{\lambda}\boldsymbol{a}(\theta, \phi)$, and $\boldsymbol{a}$ is the unit vector with spherical coordinates angles θ and ϕ

- Let's write a bandpass signal:

$$f(t, \boldsymbol{p}_n) = \sqrt{2}Re\{\tilde{f}(t, \boldsymbol{p}_n)e^{j\omega_c t}\}, n = 0, 1, \cdots, N-1$$

- ω_c corresponds to the carrier frequency and the complex envelope $\tilde{f}(t, \boldsymbol{p}_n)$ is bandlimited to the region



$$|\underbrace{\omega - \omega_c}_{\omega_L}| \leq 2\pi B_s/2$$

Bandlimited and Narrowband Signals

- Bandlimited plane wave:

$$f(t, \boldsymbol{p}_n) = \sqrt{2} Re\{\tilde{f}(t - \tau_n) e^{j\omega_c (t - \tau_n)}\}, n = 0, 1, \cdots, N - 1$$

Bandlimited and Narrowband Signals

- Bandlimited plane wave:
  $f(t, \boldsymbol{p}_n) = \sqrt{2} Re\{\tilde{f}(t - \tau_n) e^{j\omega_c (t - \tau_n)}\}, n = 0, 1, \cdots, N-1$

- Maximum travel time (ΔT_{max}) across the (linear) array: travel time between the two sensors at the extremities (signal arriving along the end-fire)

Bandlimited and Narrowband Signals

- Bandlimited plane wave:
  $f(t, \boldsymbol{p}_n) = \sqrt{2} Re\{\tilde{f}(t-\tau_n)e^{j\omega_c(t-\tau_n)}\}, n = 0, 1, \cdots, N-1$

- Maximum travel time (ΔT_{max}) across the (linear) array: travel time between the two sensors at the extremities (signal arriving along the end-fire)

- Assuming the origin is at the array's center of gravity:
  $\sum_{n=0}^{N-1} \boldsymbol{p}_n = 0 \Rightarrow \tau_n \leq \Delta T_{max}$

Bandlimited and Narrowband Signals

- Bandlimited plane wave:
  $f(t, \boldsymbol{p}_n) = \sqrt{2} Re\{\tilde{f}(t - \tau_n) e^{j\omega_c (t-\tau_n)}\}, n = 0, 1, \cdots, N-1$

- Maximum travel time (ΔT_{max}) across the (linear) array: travel time between the two sensors at the extremities (signal arriving along the end-fire)

- Assuming the origin is at the array's center of gravity:
  $\sum_{n=0}^{N-1} \boldsymbol{p}_n = 0 \Rightarrow \tau_n \leq \Delta T_{max}$

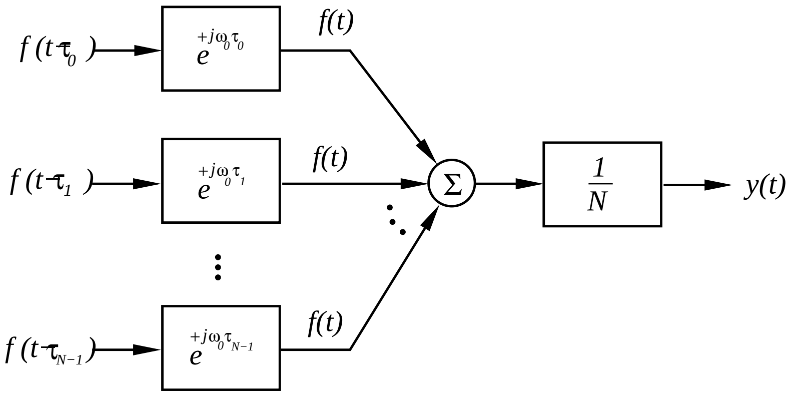
- In Narrowband (NB) signals, $B_s \Delta T_{max} \ll 1$
  $\Rightarrow \tilde{f}(t - \tau_n) \simeq \tilde{f}(t)$ and
  $f(t, \boldsymbol{p}_n) = \sqrt{2} Re\{\tilde{f}(t) e^{-j\omega_c \tau_n} e^{j\omega_c t}\}$

Phased-Array

- For NB signals, the delay is approximated by a phase-shift:
  $\Rightarrow$ delay&sum beamformer $\equiv$ PHASED ARRAY

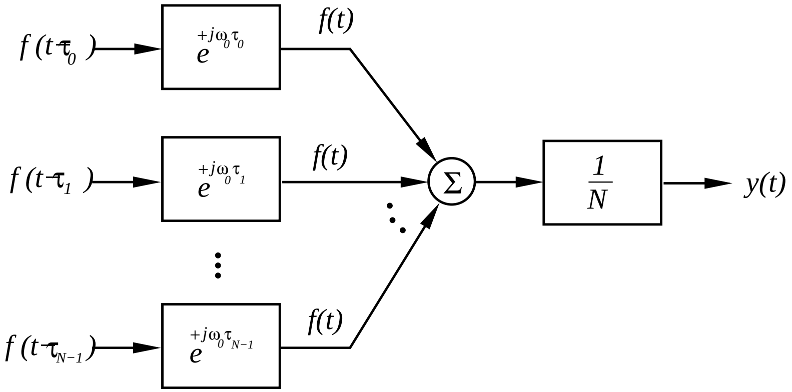
Phased-Array

- For NB signals, the delay is approximated by a phase-shift:
  $\Rightarrow$ delay&sum beamformer $\equiv$ PHASED ARRAY



Phased-Array

- For NB signals, the delay is approximated by a phase-shift:
  $\Rightarrow$ delay&sum beamformer $\equiv$ PHASED ARRAY



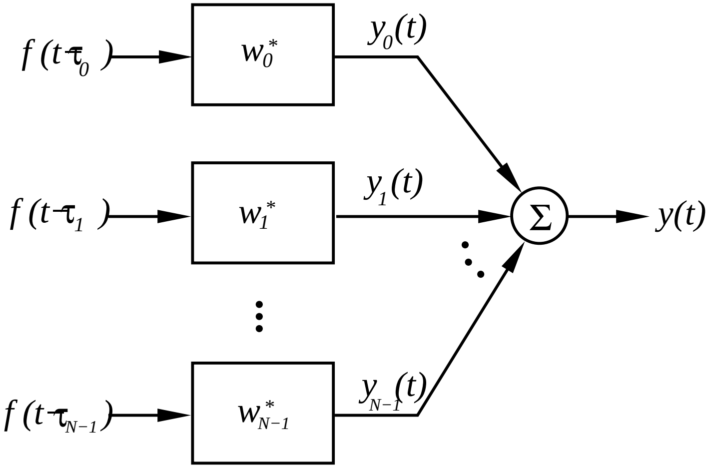
- The phased array can be implemented adjusting the gain and phase to achieve a desired beam pattern

NB Beamformers

- In narrowband beamformers: $y(t, \boldsymbol{k}) = \boldsymbol{w}^H \boldsymbol{v}_{\boldsymbol{k}}(\boldsymbol{k}) e^{j\omega t}$

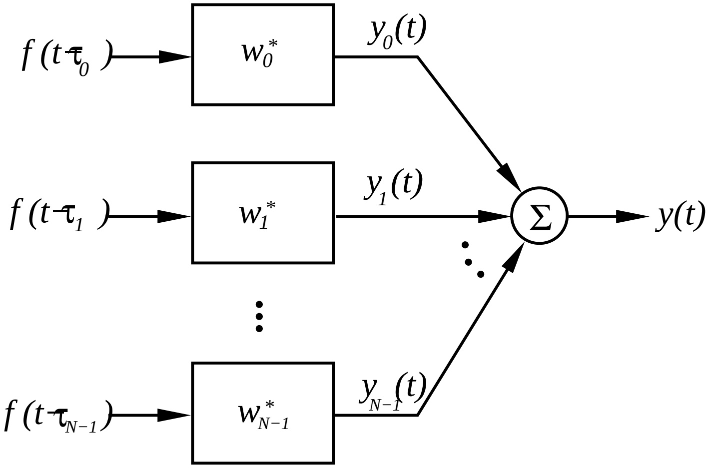
NB Beamformers

- In narrowband beamformers: $y(t, \boldsymbol{k}) = \boldsymbol{w}^H \boldsymbol{v}_{\boldsymbol{k}}(\boldsymbol{k}) e^{j\omega t}$



NB Beamformers

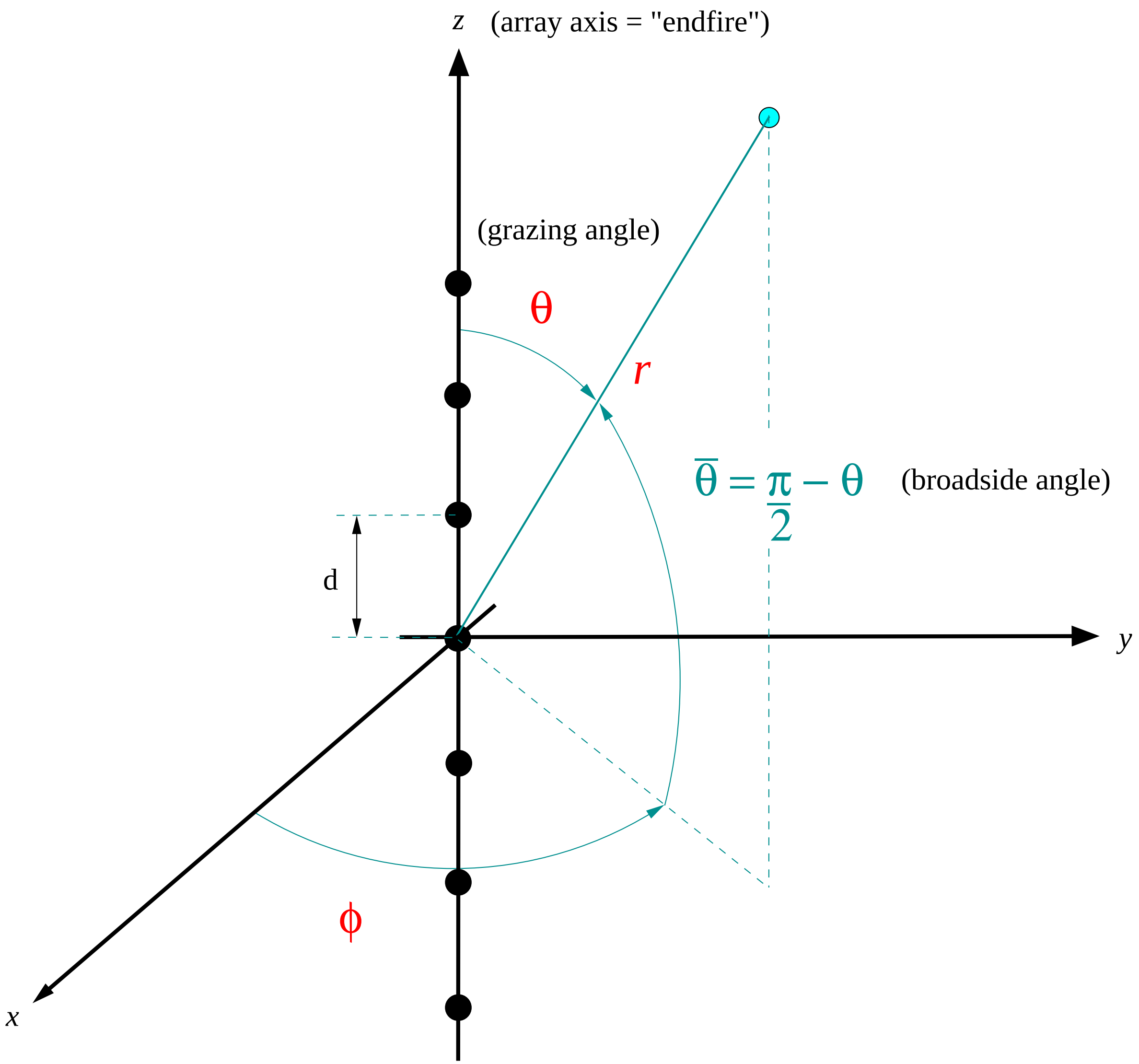
- In narrowband beamformers: $y(t, \boldsymbol{k}) = \boldsymbol{w}^H \boldsymbol{v}_{\boldsymbol{k}}(\boldsymbol{k}) e^{j\omega t}$



- $\Upsilon(\omega, \boldsymbol{k}) = \underbrace{\boldsymbol{w}^H}_{\boldsymbol{H}^T(\omega)} \boldsymbol{v}_{\boldsymbol{k}}(\boldsymbol{k})$

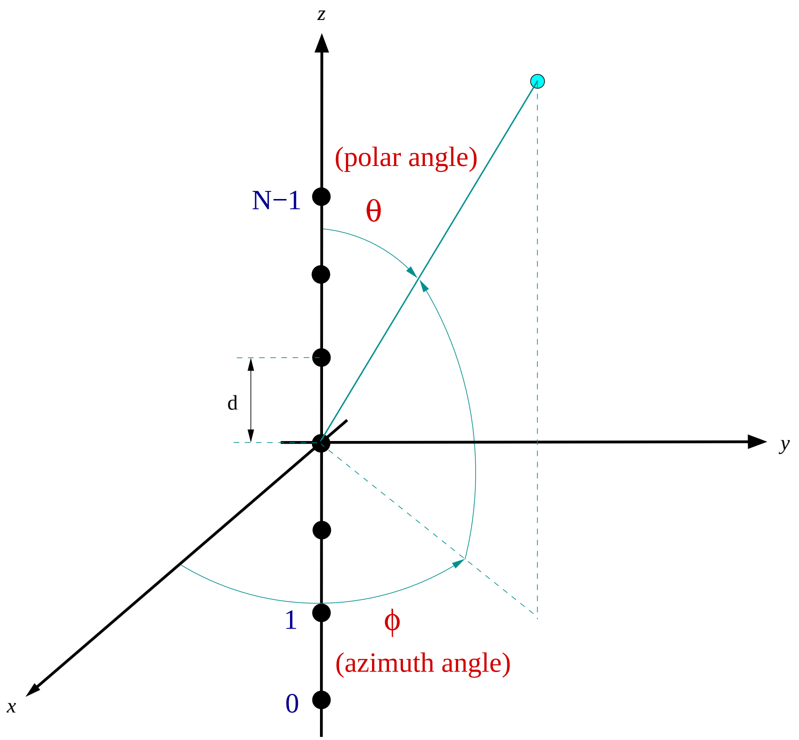
2.3 Uniform Linear Arrays (ULA)

Uniformly Spaced Linear Arrays



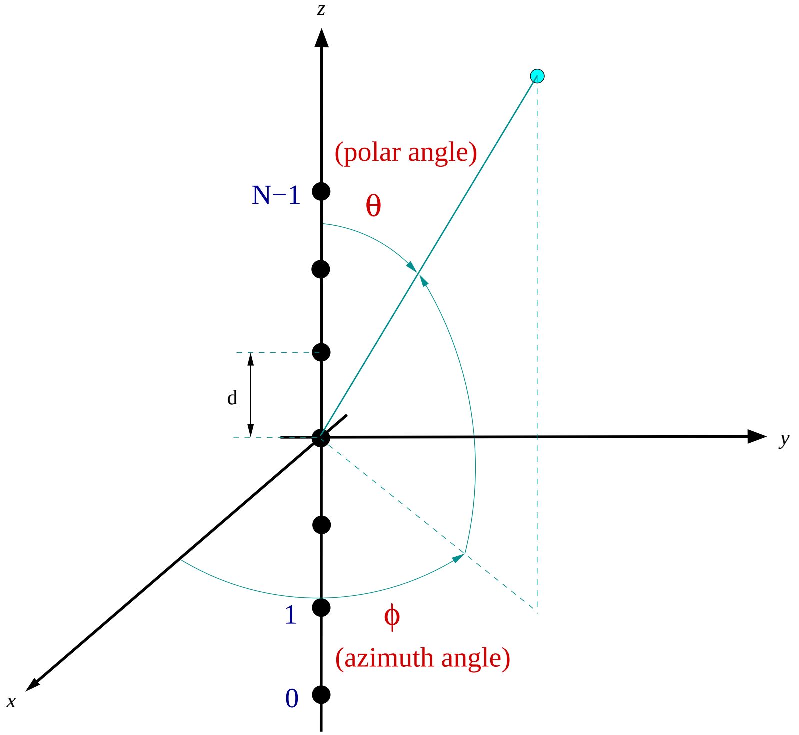
ULA

- An ULA along axis z:



ULA

- An ULA along axis z:

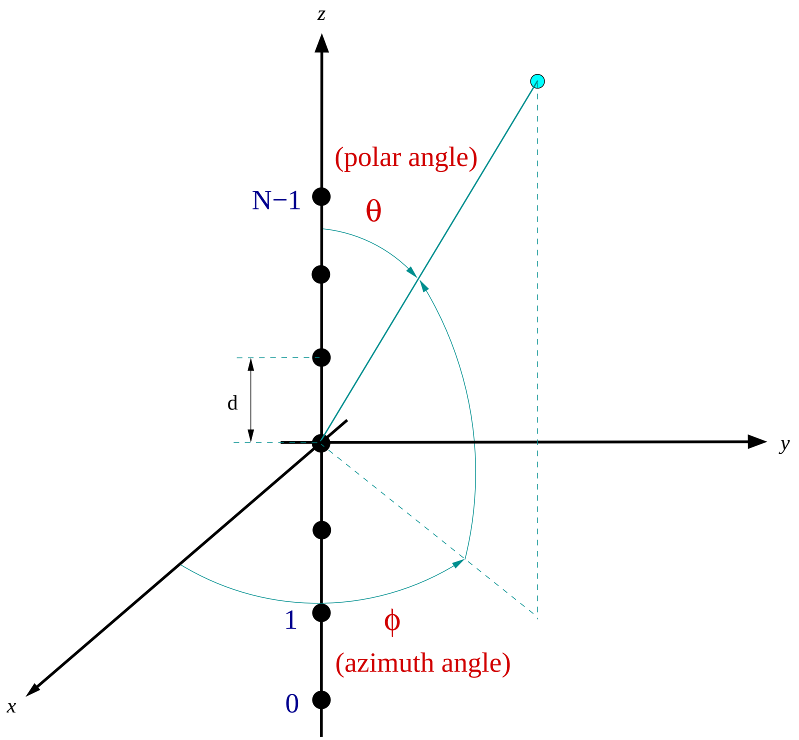


- Location of the elements:

$$\begin{cases} p_{zn} = (n - \frac{N-1}{2})d, \text{ for } n = 0, 1, \cdots, N-1 \\ p_{xn} = p_{yn} = 0 \end{cases}$$

ULA

- An ULA along axis z:



- Location of the elements:

$$\begin{cases} p_{zn} = (n - \frac{N-1}{2})d, & \text{for } n = 0, 1, \cdots, N-1 \\ p_{xn} = p_{yn} = 0 \end{cases}$$

- Therefore, $\boldsymbol{p}_n = \begin{bmatrix} 0 \\ 0 \\ (n - \frac{N-1}{2})d \end{bmatrix}$

ULA

- Array manifold vector:

$$\boldsymbol{v}_{\boldsymbol{k}}(\boldsymbol{k}) = \begin{bmatrix} e^{-j\boldsymbol{k}^T\boldsymbol{p}_0} \\ \vdots \\ e^{-j\boldsymbol{k}^T\boldsymbol{p}_n} \\ \vdots \\ e^{-j\boldsymbol{k}^T\boldsymbol{p}_{N-1}} \end{bmatrix}$$

$$\begin{bmatrix} k_x & k_y & k_z \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \left[n - \frac{N-1}{2}\right] d \end{bmatrix}$$

ULA

- Array manifold vector:

$$\boldsymbol{v}_{\boldsymbol{k}}(\boldsymbol{k}) = \left[e^{-j\boldsymbol{k}^T\boldsymbol{p}_0} \quad e^{-j\boldsymbol{k}^T\boldsymbol{p}_1} \quad \ldots \quad e^{-j\boldsymbol{k}^T\boldsymbol{p}_{N-1}} \right]^T$$

$$\therefore \boldsymbol{v}_{\boldsymbol{k}}(\boldsymbol{k}) = \boldsymbol{v}_{\boldsymbol{k}}(k_z) = \begin{bmatrix} e^{+j\frac{(N-1)}{2}k_z d} \\ e^{+j(\frac{N-1}{2}-1)k_z d} \\ \vdots \\ e^{-j(\frac{N-1}{2})k_z d} \end{bmatrix}$$

Outline

1. Introduction and Fundamentals
2. Sensor Arrays and Spatial Filtering
3. Optimal Beamforming
4. Adaptive Beamforming
5. DoA Estimation with Microphone Arrays

3. Optimal Beamforming

3.1 Introduction

Introduction

- Scope: use the statistical representation of signal and noise to design array processors that are optimal in a statistical sense.

Introduction

- Scope: use the statistical representation of signal and noise to design array processors that are optimal in a statistical sense.
- We assume that the appropriate statistics are known.

Introduction

- Scope: use the statistical representation of signal and noise to design array processors that are optimal in a statistical sense.
- We assume that the appropriate statistics are known.
- Our objective of interest is to estimate the waveform of a plane-wave impinging on the array in the presence of noise and interfering signals.

Introduction

- Scope: use the statistical representation of signal and noise to design array processors that are optimal in a statistical sense.
- We assume that the appropriate statistics are known.
- Our objective of interest is to estimate the waveform of a plane-wave impinging on the array in the presence of noise and interfering signals.
- Even if a particular beamformer developed in this chapter has good performance, it does not guarantee that its adaptive version (next chapter) will. However, if the performance is poor, it is unlikely that the adaptive version will be useful.

3.2 Optimal Beamformers

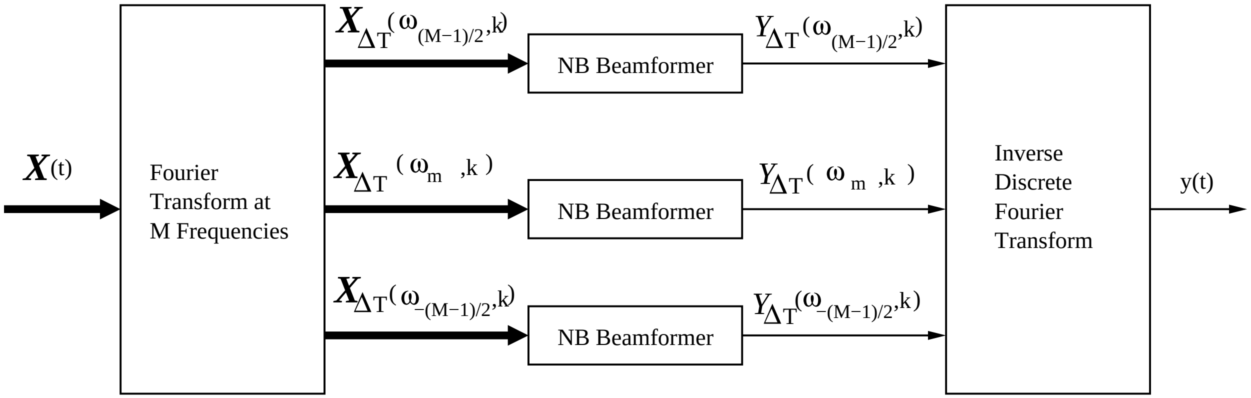
MVDR Beamformer

Snapshot model in the frequency domain:

MVDR Beamformer

Snapshot model in the frequency domain:

- In many applications, we implement a beamforming in the frequency domain ($\omega_m = \omega_c + m\frac{2\pi}{\Delta T}$ and M varies from $-\frac{M-1}{2}$ to $\frac{M-1}{2}$ if odd and from $-\frac{M}{2}$ to $\frac{M}{2} - 1$ if even).



$\boldsymbol{X}(t)$

Fourier Transform at M Frequencies

$\boldsymbol{X}_{\Delta T}(\omega_{(M-1)/2},k)$

$\boldsymbol{X}_{\Delta T}(\omega_m,k)$

$\boldsymbol{X}_{\Delta T}(\omega_{-(M-1)/2},k)$

NB Beamformer

NB Beamformer

NB Beamformer

$Y_{\Delta T}(\omega_{(M-1)/2},k)$

$Y_{\Delta T}(\omega_m,k)$

$Y_{\Delta T}(\omega_{-(M-1)/2},k)$

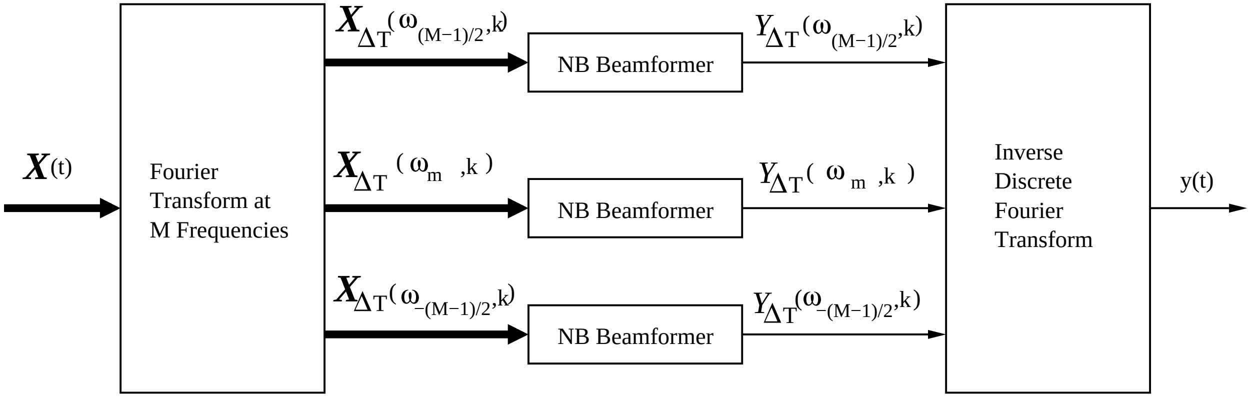
Inverse Discrete Fourier Transform

y(t)

MVDR Beamformer

Snapshot model in the frequency domain:

- In many applications, we implement a beamforming in the frequency domain ($\omega_m = \omega_c + m\frac{2\pi}{\Delta T}$ and M varies from $-\frac{M-1}{2}$ to $\frac{M-1}{2}$ if odd and from $-\frac{M}{2}$ to $\frac{M}{2} - 1$ if even).



$\boldsymbol{X}(t)$

Fourier Transform at M Frequencies

$\boldsymbol{X}_{\Delta T}(\omega_{(M-1)/2}, k)$

$\boldsymbol{X}_{\Delta T}(\omega_m, k)$

$\boldsymbol{X}_{\Delta T}(\omega_{-(M-1)/2}, k)$

NB Beamformer

NB Beamformer

NB Beamformer

$Y_{\Delta T}(\omega_{(M-1)/2}, k)$

$Y_{\Delta T}(\omega_m, k)$

$Y_{\Delta T}(\omega_{-(M-1)/2}, k)$

Inverse Discrete Fourier Transform

y(t)

- In order to generate these vectors, divide the observation interval T in K disjoint intervals of duration ΔT: $(k-1)\Delta T \leq t < k\Delta T, k = 1, \cdots, K$.

MVDR Beamformer

- ΔT must be significantly greater than the propagation time across the array.

MVDR Beamformer

- ΔT must be significantly greater than the propagation time across the array.
- ΔT also depends on the bandwidth of the input signal.

MVDR Beamformer

- ΔT must be significantly greater than the propagation time across the array.
- ΔT also depends on the bandwidth of the input signal.
- Assume an input signal with BW B_s centered in f_c

MVDR Beamformer

- ΔT must be significantly greater than the propagation time across the array.
- ΔT also depends on the bandwidth of the input signal.

- Assume an input signal with BW B_s centered in f_c
- In order to develop the frequency-domain snapshot model for the case in which the desired signals and the interfering signals can de modeled as plane waves, we have two cases: desired signals are deterministic or samples of a random process.

MVDR Beamformer

- Let’s assume the case where the signal is nonrandom but unknown; we initially consider the case of single plane-wave signal.

MVDR Beamformer

- Let’s assume the case where the signal is nonrandom but unknown; we initially consider the case of single plane-wave signal.
- Frequency-domain snapshot consists of signal plus noise: $\boldsymbol{X}(\omega) = \boldsymbol{X}_s(\omega) + \boldsymbol{N}(\omega)$

MVDR Beamformer

- Let's assume the case where the signal is nonrandom but unknown; we initially consider the case of single plane-wave signal.
- Frequency-domain snapshot consists of signal plus noise: $\boldsymbol{X}(\omega) = \boldsymbol{X}_s(\omega) + \boldsymbol{N}(\omega)$
- The signal vector can be written as $\boldsymbol{X}_s(\omega) = F(\omega)\boldsymbol{v}(\omega : \boldsymbol{k}_s)$ where $F(\omega)$ is the frequency-domain snapshot of the source signal and $\boldsymbol{v}(\omega : \boldsymbol{k}_s)$ is the array manifold vector for a plane-wave with wavenumber $\boldsymbol{k}_s$.

MVDR Beamformer

- Let's assume the case where the signal is nonrandom but unknown; we initially consider the case of single plane-wave signal.

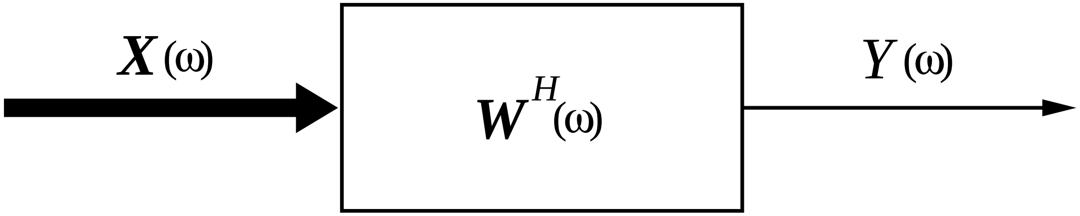
- Frequency-domain snapshot consists of signal plus noise: $\boldsymbol{X}(\omega) = \boldsymbol{X}_s(\omega) + \boldsymbol{N}(\omega)$

- The signal vector can be written as $\boldsymbol{X}_s(\omega) = F(\omega)\boldsymbol{v}(\omega : \boldsymbol{k}_s)$ where $F(\omega)$ is the frequency-domain snapshot of the source signal and $\boldsymbol{v}(\omega : \boldsymbol{k}_s)$ is the array manifold vector for a plane-wave with wavenumber $\boldsymbol{k}_s$.

- The noise snapshot is a zero-mean random vector $\boldsymbol{N}(\omega)$ with spectral matrix given by $\boldsymbol{S}_{\boldsymbol{n}}(\omega) = \boldsymbol{S}_{\boldsymbol{c}}(\omega) + \sigma_\omega^2\boldsymbol{I}$

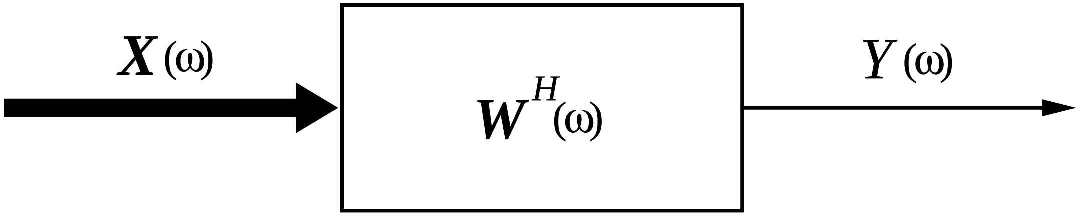
MVDR Beamformer

- We process $\boldsymbol{X}(\omega)$ with the $1 \times N$ operator $\boldsymbol{W}^H(\omega)$:



MVDR Beamformer

- We process $\boldsymbol{X}(\omega)$ with the $1 \times N$ operator $\boldsymbol{W}^H(\omega)$:



- Distortionless criterion (in the absence of noise):

$$
\begin{aligned}
Y(\omega) &= F(\omega) \\
&= \boldsymbol{W}^H(\omega)\boldsymbol{X}_s(\omega) = F(\omega)\boldsymbol{W}^H(\omega)\boldsymbol{v}(\omega : \boldsymbol{k}_s) \\
\Longrightarrow & \boldsymbol{W}^H(\omega)\boldsymbol{v}(\omega : \boldsymbol{k}_s) = 1
\end{aligned}
$$

MVDR Beamformer

- In the presence of noise, we have:

$$Y(\omega) = F(\omega) + Y_n(\omega)$$

MVDR Beamformer

- In the presence of noise, we have:

$$Y(\omega) = F(\omega) + Y_n(\omega)$$

- The mean square of the output noise is:

$$E[|Y_n(\omega)|^2] = \boldsymbol{W}^H(\omega)\boldsymbol{S}_n(\omega)\boldsymbol{W}(\omega)$$

MVDR Beamformer

- In the MVDR beamformer, we want to minimize

$$E[|Y_n(\omega)|^2] \text{ subject to } \boldsymbol{W}^H(\omega)\boldsymbol{v}(\omega : \boldsymbol{k}_s) = 1$$

MVDR Beamformer

- In the MVDR beamformer, we want to minimize

$$E[|Y_n(\omega)|^2] \text{ subject to } \boldsymbol{W}^H(\omega)\boldsymbol{v}(\omega:\boldsymbol{k}_s) = 1$$

- Using the method of Lagrange multipliers, we define the following cost function to be minimized

$$\begin{aligned} F &= \boldsymbol{W}^H(\omega)\boldsymbol{S}_n(\omega)\boldsymbol{W}\omega \\ &\quad + \lambda\left[\boldsymbol{W}^H(\omega)\boldsymbol{v}(\omega:\boldsymbol{k}_s) - 1\right] + \lambda^*\left[\boldsymbol{v}^H(\omega:\boldsymbol{k}_s)\boldsymbol{W}(\omega) - 1\right] \end{aligned}$$

MVDR Beamformer

- In the MVDR beamformer, we want to minimize

$$E[|Y_n(\omega)|^2] \text{ subject to } \boldsymbol{W}^H(\omega)\boldsymbol{v}(\omega:\boldsymbol{k}_s) = 1$$

- Using the method of Lagrange multipliers, we define the following cost function to be minimized

$$\begin{aligned} F &= \boldsymbol{W}^H(\omega)\boldsymbol{S}_n(\omega)\boldsymbol{W}\omega \\ &\quad + \lambda\left[\boldsymbol{W}^H(\omega)\boldsymbol{v}(\omega:\boldsymbol{k}_s) - 1\right] + \lambda^*\left[\boldsymbol{v}^H(\omega:\boldsymbol{k}_s)\boldsymbol{W}(\omega) - 1\right] \end{aligned}$$

- ...and the result (suppressing ω and $\boldsymbol{k}_s$) is

$$\boldsymbol{W}^H_{mvdr} = \Lambda_s \boldsymbol{v}^H \boldsymbol{S}_n^{-1} \text{ where } \Lambda_s = \left[\boldsymbol{v}^H \boldsymbol{S}_n^{-1} \boldsymbol{v}\right]^{-1}$$

MVDR Beamformer

- In the MVDR beamformer, we want to minimize

$$E[|Y_n(\omega)|^2] \text{ subject to } \boldsymbol{W}^H(\omega)\boldsymbol{v}(\omega:\boldsymbol{k}_s)=1$$

- Using the method of Lagrange multipliers, we define the following cost function to be minimized

$$\begin{aligned} F &= \boldsymbol{W}^H(\omega)\boldsymbol{S}_n(\omega)\boldsymbol{W}\omega \\ &\quad + \lambda\left[\boldsymbol{W}^H(\omega)\boldsymbol{v}(\omega:\boldsymbol{k}_s)-1\right] + \lambda^*\left[\boldsymbol{v}^H(\omega:\boldsymbol{k}_s)\boldsymbol{W}(\omega)-1\right] \end{aligned}$$

- ...and the result (suppressing ω and $\boldsymbol{k}_s$) is

$$\boldsymbol{W}^H_{mvdr} = \Lambda_s\boldsymbol{v}^H\boldsymbol{S}_n^{-1} \text{ where } \Lambda_s = \left[\boldsymbol{v}^H\boldsymbol{S}_n^{-1}\boldsymbol{v}\right]^{-1}$$

- This result is referred to as MVDR or *Capon* Beamformer.

Constrained Optimal Filtering

- The *gradient* of ξ with respect to $\boldsymbol{w}$ (real case):

$$\boldsymbol{\nabla}_{\boldsymbol{w}}\xi = \begin{bmatrix} \frac{\partial \xi}{\partial w_0} \\ \frac{\partial \xi}{\partial w_1} \\ \vdots \\ \frac{\partial \xi}{\partial w_{N-1}} \end{bmatrix}$$

Constrained Optimal Filtering

- The *gradient* of ξ with respect to $\boldsymbol{w}$ (real case):

$$\boldsymbol{\nabla}_{\boldsymbol{w}}\xi = \begin{bmatrix} \frac{\partial \xi}{\partial w_0} \\ \frac{\partial \xi}{\partial w_1} \\ \vdots \\ \frac{\partial \xi}{\partial w_{N-1}} \end{bmatrix}$$

- From the definition above, it is easy to show that: $\boldsymbol{\nabla}_{\boldsymbol{w}}(\boldsymbol{b}^T\boldsymbol{w}) = \boldsymbol{\nabla}_{\boldsymbol{w}}(\boldsymbol{w}^T\boldsymbol{b}) = \boldsymbol{b}$

Constrained Optimal Filtering

- The *gradient* of ξ with respect to $\boldsymbol{w}$ (real case):

$$\boldsymbol{\nabla}_{\boldsymbol{w}}\xi = \begin{bmatrix} \frac{\partial\xi}{\partial w_0} \\ \frac{\partial\xi}{\partial w_1} \\ \vdots \\ \frac{\partial\xi}{\partial w_{N-1}} \end{bmatrix}$$

- From the definition above, it is easy to show that:
  $\boldsymbol{\nabla}_{\boldsymbol{w}}(\boldsymbol{b}^T\boldsymbol{w}) = \boldsymbol{\nabla}_{\boldsymbol{w}}(\boldsymbol{w}^T\boldsymbol{b}) = \boldsymbol{b}$

- Also $\boldsymbol{\nabla}_{\boldsymbol{w}}(\boldsymbol{w}^T\boldsymbol{R}\boldsymbol{w}) = \boldsymbol{R}^T\boldsymbol{w} + \boldsymbol{R}\boldsymbol{w}$

Constrained Optimal Filtering

- The *gradient* of ξ with respect to $\boldsymbol{w}$ (real case):

$$\boldsymbol{\nabla}_{\boldsymbol{w}}\xi = \begin{bmatrix} \frac{\partial \xi}{\partial w_0} \\ \frac{\partial \xi}{\partial w_1} \\ \vdots \\ \frac{\partial \xi}{\partial w_{N-1}} \end{bmatrix}$$

- From the definition above, it is easy to show that: $\boldsymbol{\nabla}_{\boldsymbol{w}}(\boldsymbol{b}^T\boldsymbol{w}) = \boldsymbol{\nabla}_{\boldsymbol{w}}(\boldsymbol{w}^T\boldsymbol{b}) = \boldsymbol{b}$
- Also $\boldsymbol{\nabla}_{\boldsymbol{w}}(\boldsymbol{w}^T\boldsymbol{R}\boldsymbol{w}) = \boldsymbol{R}^T\boldsymbol{w} + \boldsymbol{R}\boldsymbol{w}$
- which, when $\mathbf{R}$ is symmetric, leads to $\boldsymbol{\nabla}_{\boldsymbol{w}}(\boldsymbol{w}^T\boldsymbol{R}\boldsymbol{w}) = 2\boldsymbol{R}\boldsymbol{w}$

Constrained Optimal Filtering

- We now assume the complex case $\boldsymbol{w} = \boldsymbol{a} + j\boldsymbol{b}$.

Constrained Optimal Filtering

- We now assume the complex case $\boldsymbol{w} = \boldsymbol{a} + j\boldsymbol{b}$.

- The gradient becomes $\nabla_{\mathbf{w}}\xi = \begin{bmatrix} \frac{\partial \xi}{\partial a_0} + j\frac{\partial \xi}{\partial b_0} \\ \frac{\partial \xi}{\partial a_1} + j\frac{\partial \xi}{\partial b_1} \\ \vdots \\ \frac{\partial \xi}{\partial a_{N-1}} + j\frac{\partial \xi}{\partial b_{N-1}} \end{bmatrix}$

Constrained Optimal Filtering

- We now assume the complex case $\boldsymbol{w} = \boldsymbol{a} + j\boldsymbol{b}$.
- The gradient becomes $\boldsymbol{\nabla}_{\mathbf{w}}\xi = \begin{bmatrix} \frac{\partial\xi}{\partial a_0} + j\frac{\partial\xi}{\partial b_0} \\ \frac{\partial\xi}{\partial a_1} + j\frac{\partial\xi}{\partial b_1} \\ \vdots \\ \frac{\partial\xi}{\partial a_{N-1}} + j\frac{\partial\xi}{\partial b_{N-1}} \end{bmatrix}$
- $\cdots$ which corresponds to $\boldsymbol{\nabla}_{\boldsymbol{w}}\xi = \boldsymbol{\nabla}_{\boldsymbol{a}}\xi + j\boldsymbol{\nabla}_{\boldsymbol{b}}\xi$

Constrained Optimal Filtering

- We now assume the complex case $\boldsymbol{w} = \boldsymbol{a} + j\boldsymbol{b}$.
- The gradient becomes $\nabla_{\mathbf{w}}\xi = \begin{bmatrix} \frac{\partial\xi}{\partial a_0} + j\frac{\partial\xi}{\partial b_0} \\ \frac{\partial\xi}{\partial a_1} + j\frac{\partial\xi}{\partial b_1} \\ \vdots \\ \frac{\partial\xi}{\partial a_{N-1}} + j\frac{\partial\xi}{\partial b_{N-1}} \end{bmatrix}$
- $\cdots$ which corresponds to $\nabla_{\boldsymbol{w}}\xi = \nabla_{\boldsymbol{a}}\xi + j\nabla_{\boldsymbol{b}}\xi$
- Let us define the *derivative* $\frac{\partial}{\partial\mathbf{w}}$ (with respect to $\mathbf{w}$):

$$\frac{\partial}{\partial\mathbf{w}} = \frac{1}{2}\begin{bmatrix} \frac{\partial}{\partial a_0} - j\frac{\partial}{\partial b_0} \\ \frac{\partial}{\partial a_1} - j\frac{\partial}{\partial b_1} \\ \vdots \\ \frac{\partial}{\partial a_{N-1}} - j\frac{\partial}{\partial b_{N-1}} \end{bmatrix}$$

Constrained Optimal Filtering

- The *conjugate derivative* with respect to $\mathbf{w}$ is

$$\frac{\partial}{\partial \mathbf{w}^*} = \frac{1}{2}\begin{bmatrix} \frac{\partial}{\partial a_0} + j\frac{\partial}{\partial b_0} \\ \frac{\partial}{\partial a_1} + j\frac{\partial}{\partial b_1} \\ \vdots \\ \frac{\partial}{\partial a_{N-1}} + j\frac{\partial}{\partial b_{N-1}} \end{bmatrix}$$

Constrained Optimal Filtering

- The *conjugate derivative* with respect to $\mathbf{w}$ is

$$\frac{\partial}{\partial \mathbf{w}^*} = \frac{1}{2}\begin{bmatrix} \frac{\partial}{\partial a_0} + j\frac{\partial}{\partial b_0} \\ \frac{\partial}{\partial a_1} + j\frac{\partial}{\partial b_1} \\ \vdots \\ \frac{\partial}{\partial a_{N-1}} + j\frac{\partial}{\partial b_{N-1}} \end{bmatrix}$$

- Therefore, $\boldsymbol{\nabla}_{\mathbf{w}}\xi = \boldsymbol{\nabla}_{\mathbf{a}}\xi + j\boldsymbol{\nabla}_{\mathbf{b}}\xi$ is equivalent to $2\frac{\partial\xi}{\partial\mathbf{w}^*}$.

Constrained Optimal Filtering

- The *conjugate derivative* with respect to $\mathbf{w}$ is

$$\frac{\partial}{\partial \mathbf{w}^*} = \frac{1}{2}\begin{bmatrix} \frac{\partial}{\partial a_0} + j\frac{\partial}{\partial b_0} \\ \frac{\partial}{\partial a_1} + j\frac{\partial}{\partial b_1} \\ \vdots \\ \frac{\partial}{\partial a_{N-1}} + j\frac{\partial}{\partial b_{N-1}} \end{bmatrix}$$

- Therefore, $\boldsymbol{\nabla}_{\mathbf{w}}\xi = \boldsymbol{\nabla}_{\mathbf{a}}\xi + j\boldsymbol{\nabla}_{\mathbf{b}}\xi$ is equivalent to $2\frac{\partial \xi}{\partial \mathbf{w}^*}$.

- The complex gradient may be slightly tricky if compared to the simple real gradient. For this reason, we exemplify the use of the complex gradient by calculating $\boldsymbol{\nabla}_{\mathbf{w}} E[|e(k)|^2]$.

Constrained Optimal Filtering

- The *conjugate derivative* with respect to $\mathbf{w}$ is

$$\frac{\partial}{\partial \mathbf{w}^*} = \frac{1}{2}\begin{bmatrix} \frac{\partial}{\partial a_0} + j\frac{\partial}{\partial b_0} \\ \frac{\partial}{\partial a_1} + j\frac{\partial}{\partial b_1} \\ \vdots \\ \frac{\partial}{\partial a_{N-1}} + j\frac{\partial}{\partial b_{N-1}} \end{bmatrix}$$

- Therefore, $\nabla_{\mathbf{w}}\xi = \nabla_{\mathbf{a}}\xi + j\nabla_{\mathbf{b}}\xi$ is equivalent to $2\frac{\partial \xi}{\partial \mathbf{w}^*}$.
- The complex gradient may be slightly tricky if compared to the simple real gradient. For this reason, we exemplify the use of the complex gradient by calculating $\nabla_{\mathbf{w}} E[|e(k)|^2]$.
- $\nabla_{\mathbf{w}} E[e(k)e^*(k)] = E\{e^*(k)[\nabla_{\mathbf{w}} e(k)] + e(k)[\nabla_{\mathbf{w}} e^*(k)]\}$

Constrained Optimal Filtering

- We compute each gradient ...

$$\begin{aligned}\boldsymbol{\nabla}_{\mathbf{w}} e(k) =& \boldsymbol{\nabla}_{\mathbf{a}}[d(k) - \mathbf{w}^H \mathbf{x}(k)] + j\boldsymbol{\nabla}_{\mathbf{b}}[d(k) - \mathbf{w}^H \mathbf{x}(k)] \\ =& - \mathbf{x}(k) - \mathbf{x}(k) = -2\mathbf{x}(k)\end{aligned}$$

Constrained Optimal Filtering

- We compute each gradient ...

$$\begin{aligned}\boldsymbol{\nabla}_{\mathbf{w}} e(k) =& \boldsymbol{\nabla}_{\mathbf{a}}[d(k) - \mathbf{w}^H \mathbf{x}(k)] + j \boldsymbol{\nabla}_{\mathbf{b}}[d(k) - \mathbf{w}^H \mathbf{x}(k)] \\ =& - \mathbf{x}(k) - \mathbf{x}(k) = -2\mathbf{x}(k)\end{aligned}$$

- and

$$\begin{aligned}\boldsymbol{\nabla}_{\mathbf{w}} e^*(k) =& \boldsymbol{\nabla}_{\mathbf{a}}[d^*(k) - \mathbf{w}^T \mathbf{x}^*(k)] + j \boldsymbol{\nabla}_{\mathbf{b}}[d^*(k) - \mathbf{w}^T \mathbf{x}^*(k)] \\ =& - \mathbf{x}^*(k) + \mathbf{x}^*(k) = \mathbf{0}\end{aligned}$$

Constrained Optimal Filtering

- We compute each gradient ...

$$\begin{aligned}\nabla_{\mathbf{w}} e(k) =& \nabla_{\mathbf{a}}[d(k) - \mathbf{w}^H \mathbf{x}(k)] + j\nabla_{\mathbf{b}}[d(k) - \mathbf{w}^H \mathbf{x}(k)] \\ =& - \mathbf{x}(k) - \mathbf{x}(k) = -2\mathbf{x}(k)\end{aligned}$$

- and

$$\begin{aligned}\nabla_{\mathbf{w}} e^*(k) =& \nabla_{\mathbf{a}}[d^*(k) - \mathbf{w}^T \mathbf{x}^*(k)] + j\nabla_{\mathbf{b}}[d^*(k) - \mathbf{w}^T \mathbf{x}^*(k)] \\ =& - \mathbf{x}^*(k) + \mathbf{x}^*(k) = \mathbf{0}\end{aligned}$$

- such that the final result is

$$\begin{aligned}\nabla_{\mathbf{w}} E[e(k)e^*(k)] =& -2E[e^*(k)\mathbf{x}(k)] \\ =& -2E[\mathbf{x}(k)[d(k) - \mathbf{w}^H \mathbf{x}(k)]^*\} \\ =& -2\underbrace{E[\mathbf{x}(k)d^*(k)]}_{\mathbf{p}} + 2\underbrace{E[\mathbf{x}(k)\mathbf{x}^H(k)]}_{\mathbf{R}}\mathbf{w}\end{aligned}$$

Constrained Optimal Filtering

- Which results in the Wiener solution $\boldsymbol{w} = \boldsymbol{R}^{-1}\boldsymbol{p}$.

Constrained Optimal Filtering

- Which results in the Wiener solution $\boldsymbol{w} = \boldsymbol{R}^{-1}\boldsymbol{p}$.
- When a set of linear constraints involving the coefficient vector of an adaptive filter is imposed, the resulting problem (LCAF)—admitting the MSE as the objective function—can be stated as minimizing $E[|e(k)|^2]$ subject to $\mathbf{C}^H\mathbf{w} = \mathbf{f}$.

Constrained Optimal Filtering

- Which results in the Wiener solution $\boldsymbol{w} = \boldsymbol{R}^{-1}\boldsymbol{p}$.
- When a set of linear constraints involving the coefficient vector of an adaptive filter is imposed, the resulting problem (LCAF)—admitting the MSE as the objective function—can be stated as minimizing $E[|e(k)|^2]$ subject to $\mathbf{C}^H\mathbf{w} = \mathbf{f}$.
- The output of the processor is $y(k) = \mathbf{w}^H\mathbf{x}(k)$.

Constrained Optimal Filtering

- Which results in the Wiener solution $\boldsymbol{w} = \boldsymbol{R}^{-1}\boldsymbol{p}$.
- When a set of linear constraints involving the coefficient vector of an adaptive filter is imposed, the resulting problem (LCAF)—admitting the MSE as the objective function—can be stated as minimizing $E[|e(k)|^2]$ subject to $\mathbf{C}^H\mathbf{w} = \mathbf{f}$.
- The output of the processor is $y(k) = \mathbf{w}^H\mathbf{x}(k)$.
- It is worth mentioning that the most general case corresponds to having a reference signal, $d(k)$. It is, however, usual to have no reference signal as in Linearly-Constrained Minimum-Variance (LCMV) applications. In LCMV, if $\mathbf{f} = 1$, the system is often referred to as Minimum-Variance Distortionless Response (MVDR).

Constrained Optimal Filtering

- Using Lagrange multipliers, we form

$$\xi(k) = E[e(k)e^*(k)] + \boldsymbol{\mathcal{L}}_R^T Re[\mathbf{C}^H\mathbf{w} - \mathbf{f}] + \boldsymbol{\mathcal{L}}_I^T Im[\mathbf{C}^H\mathbf{w} - \mathbf{f}]$$

Constrained Optimal Filtering

- Using Lagrange multipliers, we form
  $\xi(k) = E[e(k)e^*(k)] + \boldsymbol{\mathcal{L}}_R^T Re[\mathbf{C}^H\mathbf{w} - \mathbf{f}] + \boldsymbol{\mathcal{L}}_I^T Im[\mathbf{C}^H\mathbf{w} - \mathbf{f}]$
- We can also represent the above expression with a complex $\boldsymbol{\mathcal{L}}$ given by $\boldsymbol{\mathcal{L}}_R + j\boldsymbol{\mathcal{L}}_I$ such that

$$\begin{aligned} \xi(k) &= E[e(k)e^*(k)] + Re[\boldsymbol{\mathcal{L}}^H(\mathbf{C}^H\mathbf{w} - \mathbf{f})] \\ &= E[e(k)e^*(k)] + \frac{1}{2}\boldsymbol{\mathcal{L}}^H(\mathbf{C}^H\mathbf{w} - \mathbf{f}) + \frac{1}{2}\boldsymbol{\mathcal{L}}^T(\mathbf{C}^T\mathbf{w}^* - \mathbf{f}^*) \end{aligned}$$

Constrained Optimal Filtering

- Using Lagrange multipliers, we form
  $\xi(k) = E[e(k)e^*(k)] + \boldsymbol{\mathcal{L}}_R^T Re[\mathbf{C}^H\mathbf{w} - \mathbf{f}] + \boldsymbol{\mathcal{L}}_I^T Im[\mathbf{C}^H\mathbf{w} - \mathbf{f}]$
- We can also represent the above expression with a complex $\boldsymbol{\mathcal{L}}$ given by $\boldsymbol{\mathcal{L}}_R + j\boldsymbol{\mathcal{L}}_I$ such that

$$\begin{aligned}\xi(k) &= E[e(k)e^*(k)] + Re[\boldsymbol{\mathcal{L}}^H(\mathbf{C}^H\mathbf{w} - \mathbf{f})] \\ &= E[e(k)e^*(k)] + \frac{1}{2}\boldsymbol{\mathcal{L}}^H(\mathbf{C}^H\mathbf{w} - \mathbf{f}) + \frac{1}{2}\boldsymbol{\mathcal{L}}^T(\mathbf{C}^T\mathbf{w}^* - \mathbf{f}^*)\end{aligned}$$

- Noting that $e(k) = d(k) - \mathbf{w}^H\mathbf{x}(k)$, we compute:

$$\begin{aligned}\boldsymbol{\nabla}_{\mathbf{w}}\xi(k) &= \boldsymbol{\nabla}_{\mathbf{w}}\left\{E[e(k)e^*(k)] + \frac{1}{2}\boldsymbol{\mathcal{L}}^H(\mathbf{C}^H\mathbf{w} - \mathbf{f}) + \frac{1}{2}\boldsymbol{\mathcal{L}}^T(\mathbf{C}^T\mathbf{w}^* - \mathbf{f}^*)\right\} \\ &= E[-2\mathbf{x}(k)e^*(k)] + \mathbf{0} + \mathbf{C}\boldsymbol{\mathcal{L}} \\ &= -2E[\mathbf{x}(k)d^*(k)] + 2E[\mathbf{x}(k)\mathbf{x}^H(k)]\mathbf{w} + \mathbf{C}\boldsymbol{\mathcal{L}}\end{aligned}$$

Constrained Optimal Filtering

- By using $\mathbf{R} = E[\mathbf{x}(k)\mathbf{x}^H(k)]$ and $\mathbf{p} = E[d^*(k)\mathbf{x}(k)]$, the gradient is equated to zero and the results can be written as (note that stationarity was assumed for the input and reference signals): $-2\mathbf{p} + 2\mathbf{R}\mathbf{w} + \mathbf{C}\boldsymbol{\mathcal{L}} = \mathbf{0}$

Constrained Optimal Filtering

- By using $\mathbf{R} = E[\mathbf{x}(k)\mathbf{x}^H(k)]$ and $\mathbf{p} = E[d^*(k)\mathbf{x}(k)]$, the gradient is equated to zero and the results can be written as (note that stationarity was assumed for the input and reference signals): $-2\mathbf{p} + 2\mathbf{R}\mathbf{w} + \mathbf{C}\boldsymbol{\mathcal{L}} = \mathbf{0}$
- Which leads to $\mathbf{w} = \frac{1}{2}\mathbf{R}^{-1}(2\mathbf{p} - \mathbf{C}\boldsymbol{\mathcal{L}})$

Constrained Optimal Filtering

- By using $\mathbf{R} = E[\mathbf{x}(k)\mathbf{x}^H(k)]$ and $\mathbf{p} = E[d^*(k)\mathbf{x}(k)]$, the gradient is equated to zero and the results can be written as (note that stationarity was assumed for the input and reference signals): $-2\mathbf{p} + 2\mathbf{R}\mathbf{w} + \mathbf{C}\boldsymbol{\mathcal{L}} = \mathbf{0}$
- Which leads to $\mathbf{w} = \frac{1}{2}\mathbf{R}^{-1}(2\mathbf{p} - \mathbf{C}\boldsymbol{\mathcal{L}})$
- If we pre-multiply the previous expression by $\mathbf{C}^H$ and use $\mathbf{C}^H\mathbf{w} = \mathbf{f}$, we find $\boldsymbol{\mathcal{L}}$:
  $\boldsymbol{\mathcal{L}} = 2(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{p} - \mathbf{f})$

Constrained Optimal Filtering

- By using $\mathbf{R} = E[\mathbf{x}(k)\mathbf{x}^H(k)]$ and $\mathbf{p} = E[d^*(k)\mathbf{x}(k)]$, the gradient is equated to zero and the results can be written as (note that stationarity was assumed for the input and reference signals): $-2\mathbf{p} + 2\mathbf{R}\mathbf{w} + \mathbf{C}\boldsymbol{\mathcal{L}} = \mathbf{0}$
- Which leads to $\mathbf{w} = \frac{1}{2}\mathbf{R}^{-1}(2\mathbf{p} - \mathbf{C}\boldsymbol{\mathcal{L}})$
- If we pre-multiply the previous expression by $\mathbf{C}^H$ and use $\mathbf{C}^H\mathbf{w} = \mathbf{f}$, we find $\boldsymbol{\mathcal{L}}$:
$\boldsymbol{\mathcal{L}} = 2(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{p} - \mathbf{f})$
- By replacing $\boldsymbol{\mathcal{L}}$, we obtain the *Wiener solution* for the linearly constrained adaptive filter:
$\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p} + \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}(\mathbf{f} - \mathbf{C}^H\mathbf{R}^{-1}\mathbf{p})$

Constrained Optimal Filtering

- The optimal solution for LCAF:
  $\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p} + \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}(\mathbf{f} - \mathbf{C}^H\mathbf{R}^{-1}\mathbf{p})$

Constrained Optimal Filtering

- The optimal solution for LCAF:
  $\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p} + \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}(\mathbf{f} - \mathbf{C}^H\mathbf{R}^{-1}\mathbf{p})$
- Note that if $d(k) = 0$, then $\mathbf{p} = \mathbf{0}$, and we have (LCMV):
  $\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}\mathbf{f}$

Constrained Optimal Filtering

- The optimal solution for LCAF:
  $\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p} + \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}(\mathbf{f} - \mathbf{C}^H\mathbf{R}^{-1}\mathbf{p})$
- Note that if $d(k) = 0$, then $\mathbf{p} = \mathbf{0}$, and we have (LCMV):
  $\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}\mathbf{f}$
- Yet with $d(k) = 0$ but $f = 1$ (MVDR)
  $\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}$

Constrained Optimal Filtering

- The optimal solution for LCAF:
  $\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p} + \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}(\mathbf{f} - \mathbf{C}^H\mathbf{R}^{-1}\mathbf{p})$
- Note that if $d(k) = 0$, then $\mathbf{p} = \mathbf{0}$, and we have (LCMV):
  $\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}\mathbf{f}$
- Yet with $d(k) = 0$ but $f = 1$ (MVDR)
  $\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}$
- For this case, $d(k) = 0$, the cost function is termed minimum output energy (MOE) and is given by
  $E[|e(k)|^2] = \mathbf{w}^H\mathbf{R}\mathbf{w}$

Constrained Optimal Filtering

- The optimal solution for LCAF:
  $\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p} + \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}(\mathbf{f} - \mathbf{C}^H\mathbf{R}^{-1}\mathbf{p})$
- Note that if $d(k) = 0$, then $\mathbf{p} = \mathbf{0}$, and we have (LCMV):
  $\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}\mathbf{f}$
- Yet with $d(k) = 0$ but $f = 1$ (MVDR)
  $\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}$
- For this case, $d(k) = 0$, the cost function is termed minimum output energy (MOE) and is given by $E[|e(k)|^2] = \mathbf{w}^H\mathbf{R}\mathbf{w}$
- Also note that in case we do not have constraints ($\mathbf{C}$ and $\mathbf{f}$ are nulls), the optimal solution above becomes the *unconstrained* Wiener solution $\mathbf{R}^{-1}\mathbf{p}$.

The GSC

We start by doing a transformation in the coefficient vector.

- Let $\mathbf{T} = [\mathbf{C}\ \mathbf{B}]$ such that

$$\mathbf{w} = \mathbf{T}\bar{\mathbf{w}} = [\mathbf{C}\ \mathbf{B}] \begin{bmatrix} \bar{\mathbf{w}}_U \\ -\bar{\mathbf{w}}_L \end{bmatrix} = \mathbf{C}\bar{\mathbf{w}}_U - \mathbf{B}\bar{\mathbf{w}}_L$$

The GSC

We start by doing a transformation in the coefficient vector.

- Let $\mathbf{T} = [\mathbf{C}\ \mathbf{B}]$ such that
  $$\mathbf{w} = \mathbf{T}\bar{\mathbf{w}} = [\mathbf{C}\ \mathbf{B}] \begin{bmatrix} \bar{\mathbf{w}}_U \\ -\bar{\mathbf{w}}_L \end{bmatrix} = \mathbf{C}\bar{\mathbf{w}}_U - \mathbf{B}\bar{\mathbf{w}}_L$$
- Matrix $\mathbf{B}$ is usually called the *Blocking Matrix* and we recall that $\mathbf{C}^H\mathbf{w} = \mathbf{g}$ such that
  $$\mathbf{C}^H\mathbf{w} = \mathbf{C}^H\mathbf{C}\bar{\mathbf{w}}_U - \mathbf{C}^H\mathbf{B}\bar{\mathbf{w}}_L = \mathbf{g}.$$

The GSC

We start by doing a transformation in the coefficient vector.

- Let $\mathbf{T} = [\mathbf{C}\ \mathbf{B}]$ such that

  $$\mathbf{w} = \mathbf{T}\bar{\mathbf{w}} = [\mathbf{C}\ \mathbf{B}] \begin{bmatrix} \bar{\mathbf{w}}_U \\ -\bar{\mathbf{w}}_L \end{bmatrix} = \mathbf{C}\bar{\mathbf{w}}_U - \mathbf{B}\bar{\mathbf{w}}_L$$

- Matrix $\mathbf{B}$ is usually called the *Blocking Matrix* and we recall that $\mathbf{C}^H\mathbf{w} = \mathbf{g}$ such that $\mathbf{C}^H\mathbf{w} = \mathbf{C}^H\mathbf{C}\bar{\mathbf{w}}_U - \mathbf{C}^H\mathbf{B}\bar{\mathbf{w}}_L = \mathbf{g}$.
- If we impose the condition $\mathbf{B}^H\mathbf{C} = \mathbf{0}$ or, equivalently, $\mathbf{C}^H\mathbf{B} = \mathbf{0}$, we will have $\bar{\mathbf{w}}_U = (\mathbf{C}^H\mathbf{C})^{-1}\mathbf{g}$.

The GSC

We start by doing a transformation in the coefficient vector.

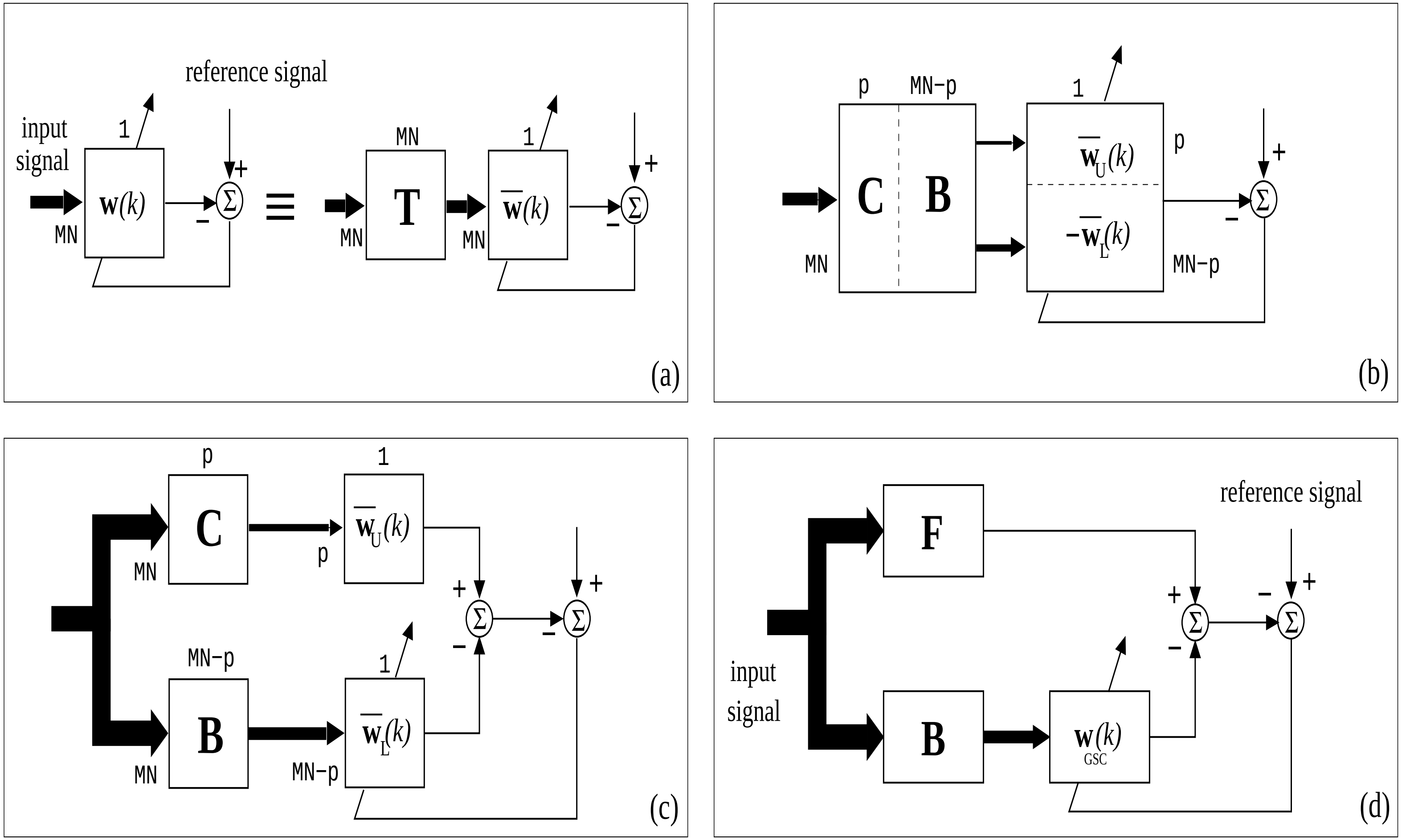
- Let $\mathbf{T} = [\mathbf{C}\ \mathbf{B}]$ such that

  $$\mathbf{w} = \mathbf{T}\bar{\mathbf{w}} = [\mathbf{C}\ \mathbf{B}] \begin{bmatrix} \bar{\mathbf{w}}_U \\ -\bar{\mathbf{w}}_L \end{bmatrix} = \mathbf{C}\bar{\mathbf{w}}_U - \mathbf{B}\bar{\mathbf{w}}_L$$

- Matrix $\mathbf{B}$ is usually called the *Blocking Matrix* and we recall that $\mathbf{C}^H\mathbf{w} = \mathbf{g}$ such that $\mathbf{C}^H\mathbf{w} = \mathbf{C}^H\mathbf{C}\bar{\mathbf{w}}_U - \mathbf{C}^H\mathbf{B}\bar{\mathbf{w}}_L = \mathbf{g}$.
- If we impose the condition $\mathbf{B}^H\mathbf{C} = \mathbf{0}$ or, equivalently, $\mathbf{C}^H\mathbf{B} = \mathbf{0}$, we will have $\bar{\mathbf{w}}_U = (\mathbf{C}^H\mathbf{C})^{-1}\mathbf{g}$.
- $\bar{\mathbf{w}}_U$ is fixed and termed the quiescent weight vector; the minimization process will be carried out only in the lower part, also designated $\mathbf{w}_{GSC} = \bar{\mathbf{w}}_L$.

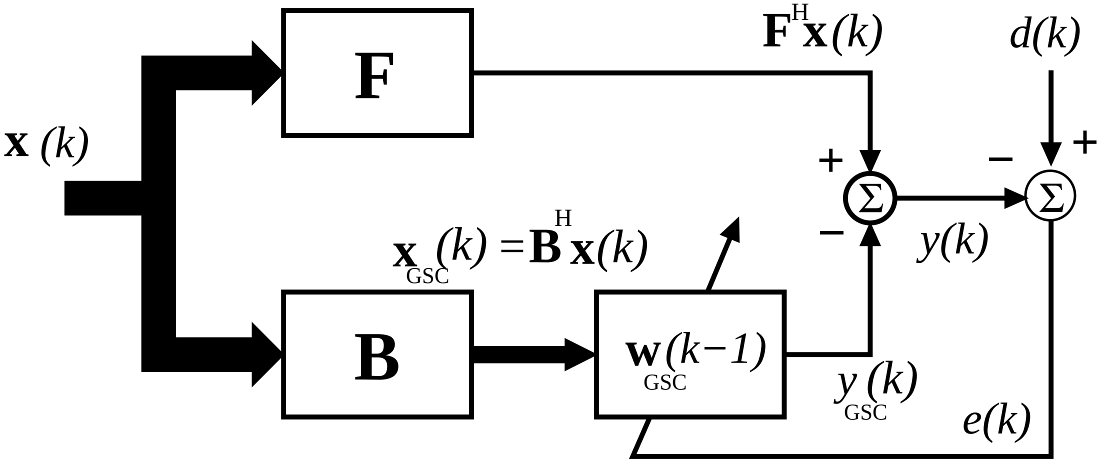
The GSC

- It is shown below how to split the transformation matrix into two parts: a fixed path and an *adaptive* path.



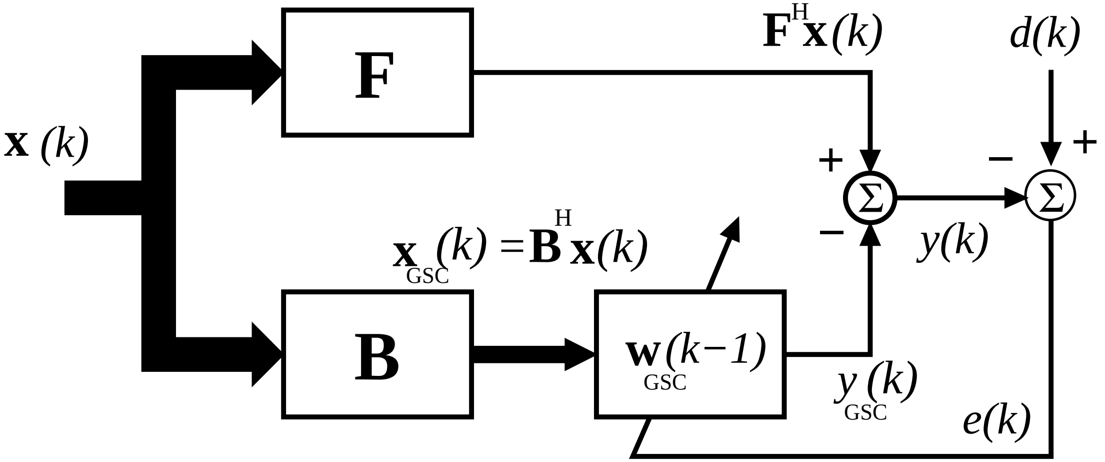
The GSC

- This structure (detailed below) was named the Generalized Sidelobe Canceller (GSC).



The GSC

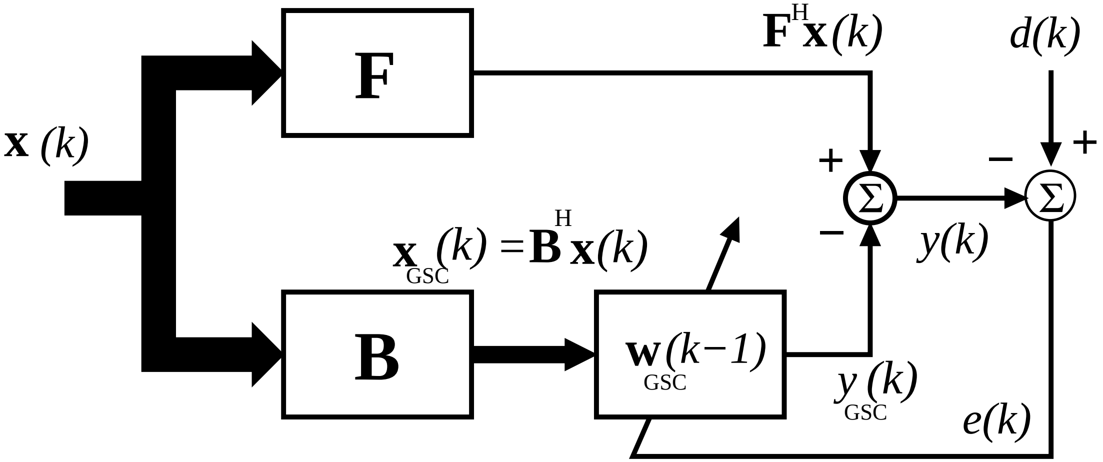
- This structure (detailed below) was named the Generalized Sidelobe Canceller (GSC).



- It is always possible to have the overall equivalent coefficient vector which is given by $\mathbf{w} = \mathbf{F} - \mathbf{B}\mathbf{w}_{GSC}$.

The GSC

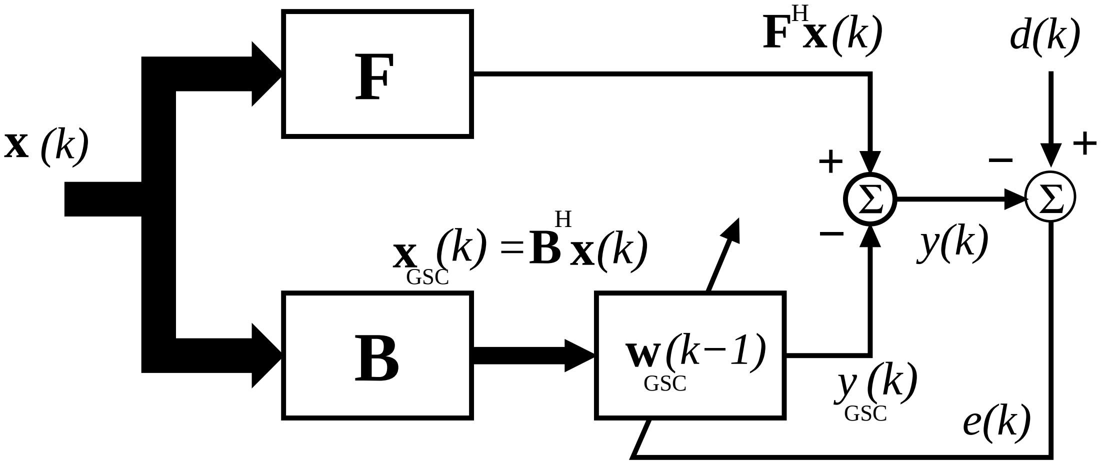
- This structure (detailed below) was named the Generalized Sidelobe Canceller (GSC).



- It is always possible to have the overall equivalent coefficient vector which is given by $\mathbf{w} = \mathbf{F} - \mathbf{B}\mathbf{w}_{GSC}$.
- If we pre-multiply last equation by $\mathbf{B}^H$ and isolate $\mathbf{w}_{GSC}$, we find $\mathbf{w}_{GSC} = -(\mathbf{B}^H\mathbf{B})^{-1}\mathbf{B}^H\mathbf{w}$.

The GSC

- This structure (detailed below) was named the Generalized Sidelobe Canceller (GSC).



- It is always possible to have the overall equivalent coefficient vector which is given by $\mathbf{w} = \mathbf{F} - \mathbf{B}\mathbf{w}_{GSC}$.
- If we pre-multiply last equation by B^H and isolate $\mathbf{w}_{GSC}$, we find $\mathbf{w}_{GSC} = -(\mathbf{B}^H\mathbf{B})^{-1}\mathbf{B}^H\mathbf{w}$.
- Knowing that $\mathbf{T} = [\mathrm{C}\ \mathrm{B}]$ and that $\mathbf{T}^H\mathbf{T} = \boldsymbol{I}$, it follows that $\boldsymbol{P} = \mathbf{I} - \mathrm{C}(\mathrm{C}^H\mathrm{C})^{-1}\mathrm{C}^H = \mathrm{B}(\mathrm{B}^H\mathrm{B})\mathrm{B}^H$.

The GSC

- A simple procedure to find the optimal GSC solution comes from the unconstrained Wiener solution applied to the unconstrained filter: $\mathbf{w}_{GSC-OPT} = \mathbf{R}_{GSC}^{-1}\mathbf{p}_{GSC}$

The GSC

- A simple procedure to find the optimal GSC solution comes from the unconstrained Wiener solution applied to the unconstrained filter: $\mathbf{w}_{GSC-OPT} = \mathbf{R}_{GSC}^{-1}\mathbf{p}_{GSC}$
- From the figure, it is clear that:
$\mathbf{R}_{GSC} = E[\mathbf{x}_{GSC}\mathbf{x}_{GSC}^{H}] = E[\mathbf{B}^{H}\mathbf{x}\mathbf{x}^{H}\mathbf{B}] = \mathbf{B}^{H}\mathbf{R}\mathbf{B}$

The GSC

- A simple procedure to find the optimal GSC solution comes from the unconstrained Wiener solution applied to the unconstrained filter: $\mathbf{w}_{GSC-OPT} = \mathbf{R}_{GSC}^{-1}\mathbf{p}_{GSC}$
- From the figure, it is clear that: $\mathbf{R}_{GSC} = E[\mathbf{x}_{GSC}\mathbf{x}_{GSC}^{H}] = E[\mathbf{B}^{H}\mathbf{x}\mathbf{x}^{H}\mathbf{B}] = \mathbf{B}^{H}\mathbf{R}\mathbf{B}$
- The cross-correlation vector is given as:

$$\begin{aligned}\mathbf{p}_{GSC} &= E[d_{GSC}^{*}\mathbf{x}_{GSC}] \\ &= E\{[\mathbf{F}^{H}\mathbf{x} - d]^{*}[\mathbf{B}^{H}\mathbf{x}]\} \\ &= E[-\mathbf{B}^{H}d^{*}\mathbf{x} + \mathbf{B}^{H}\mathbf{x}\mathbf{x}^{H}\mathbf{F}] \\ &= -\mathbf{B}^{H}\mathbf{p} + \mathbf{B}^{H}\mathbf{R}\mathbf{F}\end{aligned}$$

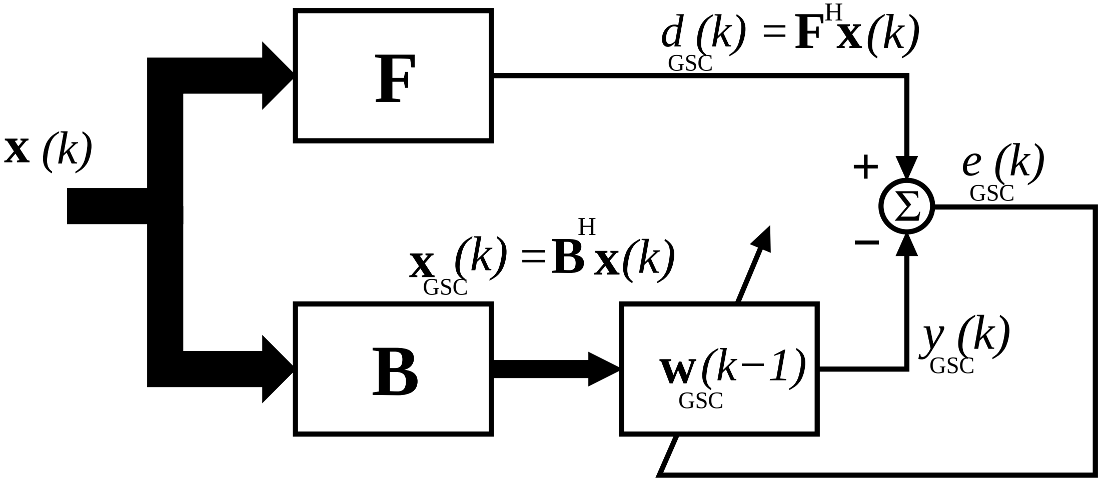
The GSC

- A simple procedure to find the optimal GSC solution comes from the unconstrained Wiener solution applied to the unconstrained filter: $\mathbf{w}_{GSC-OPT} = \mathbf{R}_{GSC}^{-1}\mathbf{p}_{GSC}$
- From the figure, it is clear that: $\mathbf{R}_{GSC} = E[\mathbf{x}_{GSC}\mathbf{x}_{GSC}^H] = E[\mathbf{B}^H\mathbf{x}\mathbf{x}^H\mathbf{B}] = \mathbf{B}^H\mathbf{R}\mathbf{B}$
- The cross-correlation vector is given as:

$$\begin{aligned} \mathbf{p}_{GSC} &= E[d_{GSC}^*\mathbf{x}_{GSC}] \\ &= E\{[\mathbf{F}^H\mathbf{x} - d]^*[\mathbf{B}^H\mathbf{x}]\} \\ &= E[-\mathbf{B}^H d^*\mathbf{x} + \mathbf{B}^H\mathbf{x}\mathbf{x}^H\mathbf{F}] \\ &= -\mathbf{B}^H\mathbf{p} + \mathbf{B}^H\mathbf{R}\mathbf{F} \end{aligned}$$

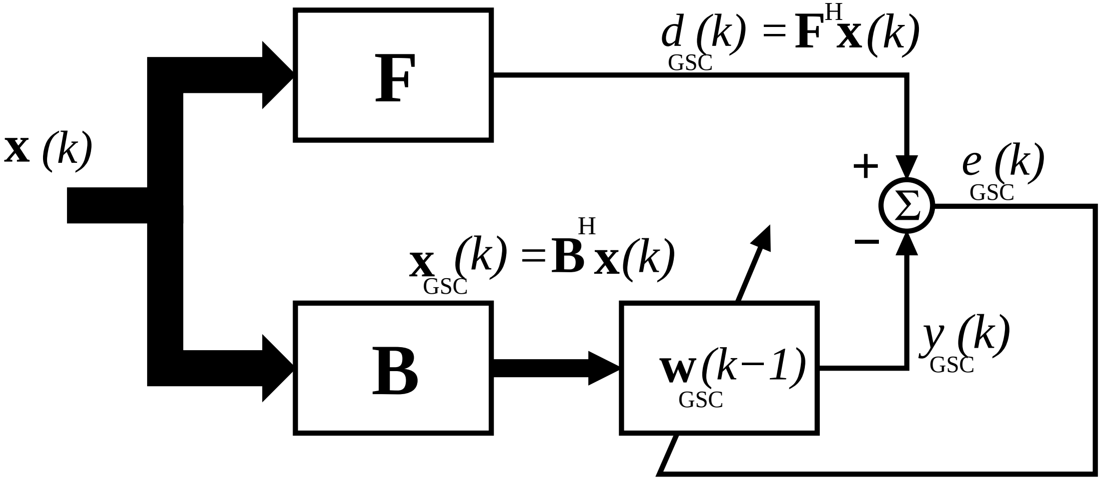
- $\cdots$ and $\mathbf{w}_{GSC-OPT} = (\mathbf{B}^H\mathbf{R}\mathbf{B})^{-1}(-\mathbf{B}^H\mathbf{p} + \mathbf{B}^H\mathbf{R}\mathbf{F})$

- A common case is when $d(k) = 0$:



The GSC

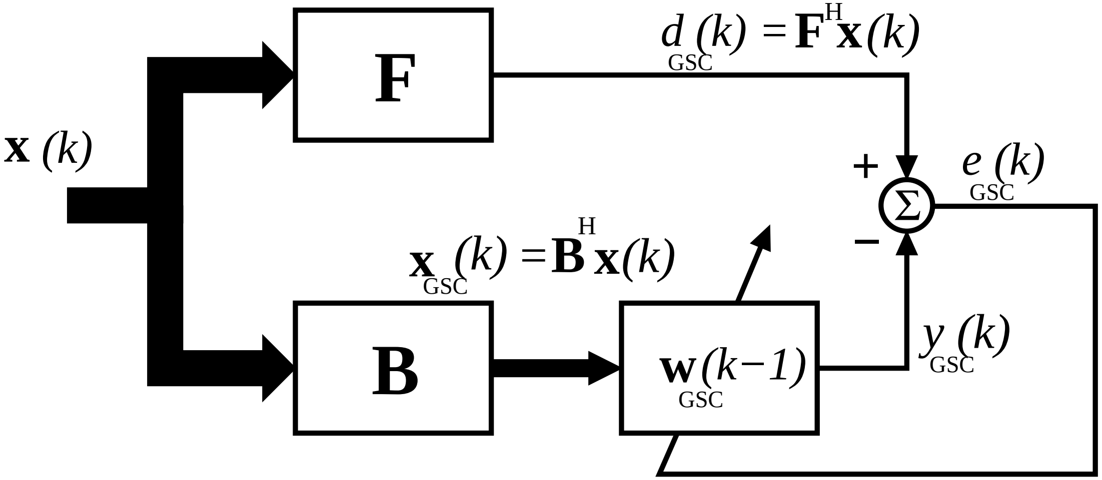
- A common case is when $d(k) = 0$:



- We have dropped the negative sign that should exist according to the notation used. Although we define $e_{GSC}(k) = -e(k)$, the inversion of the sign in the error signal actually results in the same results because the error function is always based on the absolute value.

The GSC

- A common case is when $d(k) = 0$:



- We have dropped the negative sign that should exist according to the notation used. Although we define $e_{GSC}(k) = -e(k)$, the inversion of the sign in the error signal actually results in the same results because the error function is always based on the absolute value.
- In this case, the optimum filter $\mathbf{w}_{OPT}$ is: $\mathbf{F} - \mathbf{B}\mathbf{w}_{GSC-OPT} = \mathbf{F} - \mathbf{B}(\mathbf{B}^H\mathbf{R}\mathbf{B})^{-1}\mathbf{B}^H\mathbf{R}\mathbf{F} = \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}\mathbf{f}$ (LCMV solution)

The GSC

- Choosing the blocking matrix: B plays an important role since its choice determines computational complexity and even robustness against numerical instability.

The GSC

- Choosing the blocking matrix: B plays an important role since its choice determines computational complexity and even robustness against numerical instability.
- Since the only need for B is having its columns forming a basis orthogonal to the constraints, $\mathbf{B}^H\mathbf{C} = \mathbf{0}$, a myriad of options are possible.

The GSC

- Choosing the blocking matrix: $\boldsymbol{B}$ plays an important role since its choice determines computational complexity and even robustness against numerical instability.
- Since the only need for $\mathbf{B}$ is having its columns forming a basis orthogonal to the constraints, $\mathbf{B}^H\mathbf{C} = \mathbf{0}$, a myriad of options are possible.
- Let us recall the paper by Griffiths and Jim where the term GSC was coined; let

$$\mathbf{C}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

The GSC

- Choosing the blocking matrix: B plays an important role since its choice determines computational complexity and even robustness against numerical instability.
- Since the only need for B is having its columns forming a basis orthogonal to the constraints, $\mathrm{B}^H\mathbf{C} = \mathbf{0}$, a myriad of options are possible.
- Let us recall the paper by Griffiths and Jim where the term GSC was coined; let

$$\mathbf{C}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

- With simple constraint matrices, simple blocking matrices satisfying $\mathrm{B}^T\mathbf{C} = \mathbf{0}$ are possible.

The GSC

- For this particular example, the paper presents two possibilities. The first one (orthogonal) is:

The GSC

- For this particular example, the paper presents two possibilities. The first one (orthogonal) is:

$$\mathbf{B}_1^T =$$

$$\begin{bmatrix} 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \end{bmatrix}$$

The GSC

- And the second possibility (non-orthogonal) is:

The GSC

- And the second possibility (non-orthogonal) is:

$$\mathbf{B}_2^T =$$

$$\begin{bmatrix}
1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1
\end{bmatrix}$$

The GSC

- **SVD**: the blocking matrix can be produced with the following Matlab command lines,

```
[U,S,V]=svd(C);
B3=U(:,p+1:M*N); % p=N in this case
```

The GSC

- **SVD**: the blocking matrix can be produced with the following Matlab command lines,

```
[U,S,V]=svd(C);
B3=U(:,p+1:M*N); % p=N in this case
```

B_3^T is given by:

$$\begin{bmatrix}
-0.50 & -0.17 & -0.17 & 0.83 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.25 & -0.42 & 0.08 & 0.08 & 0.75 & -0.25 & -0.25 & -0.25 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.25 & -0.42 & 0.08 & 0.08 & -0.25 & 0.75 & -0.25 & -0.25 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.25 & -0.42 & 0.08 & 0.08 & -0.25 & -0.25 & 0.75 & -0.20 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.25 & -0.42 & 0.08 & 0.08 & -0.25 & -0.25 & -0.25 & 0.75 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.25 & 0.08 & -0.42 & 0.08 & 0.00 & 0.00 & 0.00 & 0.00 & 0.75 & -0.25 & -0.25 & -0.25 \\
0.25 & 0.08 & -0.42 & 0.08 & 0.00 & 0.00 & 0.00 & 0.00 & -0.25 & 0.75 & -0.25 & -0.25 \\
0.25 & 0.08 & -0.42 & 0.08 & 0.00 & 0.00 & 0.00 & 0.00 & -0.25 & -0.25 & 0.75 & -0.25 \\
0.25 & 0.08 & -0.42 & 0.08 & 0.00 & 0.00 & 0.00 & 0.00 & -0.25 & -0.25 & -0.25 & 0.75
\end{bmatrix}$$

The GSC

- **QRD**: the blocking matrix can be produced with the following Matlab command lines,

```
[Q,R]=qr(C);
B4=Q(:,p+1:M*N);
```

- **QRD**: the blocking matrix can be produced with the following Matlab command lines,

```
[Q,R]=qr(C);
B4=Q(:,p+1:M*N);
```

- B_4 was identical to B_3 (SVD).

The GSC

- **QRD**: the blocking matrix can be produced with the following Matlab command lines,

```
[Q,R]=qr(C);
B4=Q(:,p+1:M*N);
```

- B_4 was identical to B_3 (SVD).
- Two other possibilities are: the one presented in [Tseng Griffiths 88] where a decomposition procedure is introduced in order to offer an effective implementation structure and the other one concerned to a narrowband BF implemented with GSC where B is combined with a wavelet transform [Chu Fang 99].

The GSC

- **QRD**: the blocking matrix can be produced with the following Matlab command lines,
```
    [Q,R]=qr(C);
    B4=Q(:,p+1:M*N);
```
- B_4 was identical to B_3 (SVD).
- Two other possibilities are: the one presented in [Tseng Griffiths 88] where a decomposition procedure is introduced in order to offer an effective implementation structure and the other one concerned to a narrowband BF implemented with GSC where B is combined with a wavelet transform [Chu Fang 99].
- Finally, a new efficient linearly constrained adaptive scheme which can also be visualized as a GSC structure can be found in [Campos&Werner&Apolinário IEEE-TSP Sept. 2002].

Outline

1. Introduction and Fundamentals
2. Sensor Arrays and Spatial Filtering
3. Optimal Beamforming
4. Adaptive Beamforming
5. DoA Estimation with Microphone Arrays

4. Adaptive Beamforming

4.1 Introduction

Introduction

- Scope: instead of assuming knowledge about the statistical properties of the signals, beamformers are designed based on statistics gathered online.

Introduction

- Scope: instead of assuming knowledge about the statistical properties of the signals, beamformers are designed based on statistics gathered online.
- Different algorithms may be employed for iteratively approximating the desired solution.

Introduction

- Scope: instead of assuming knowledge about the statistical properties of the signals, beamformers are designed based on statistics gathered online.
- Different algorithms may be employed for iteratively approximating the desired solution.
- We will briefly cover a small subset of algorithms for constrained adaptive filters.

Introduction

- Linearly constrained adaptive filters (LCAF) have found application in numerous areas, such as spectrum analysis, spatial-temporal processing, antenna arrays, interference suppression, among others.

Introduction

- Linearly constrained adaptive filters (LCAF) have found application in numerous areas, such as spectrum analysis, spatial-temporal processing, antenna arrays, interference suppression, among others.
- LCAF algorithms incorporate into the solution application-specific requirements translated into a set of linear equations to be satisfied by the coefficients.

Introduction

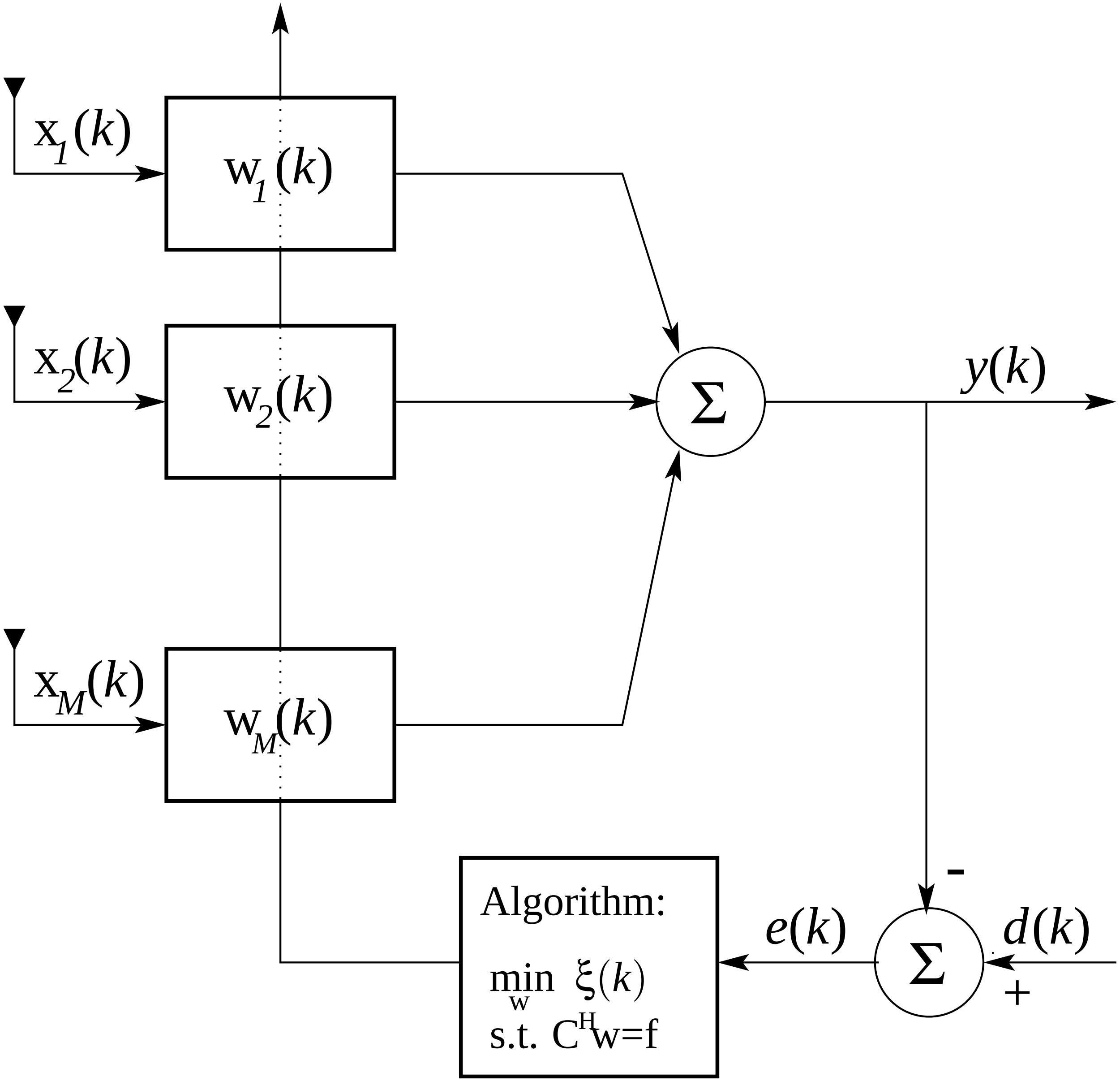
- Linearly constrained adaptive filters (LCAF) have found application in numerous areas, such as spectrum analysis, spatial-temporal processing, antenna arrays, interference suppression, among others.
- LCAF algorithms incorporate into the solution application-specific requirements translated into a set of linear equations to be satisfied by the coefficients.
    - For example, if direction of arrival of the signal of interest is known, jammer suppression can take place through spatial filtering without the need of training signal, or

Introduction

- Linearly constrained adaptive filters (LCAF) have found application in numerous areas, such as spectrum analysis, spatial-temporal processing, antenna arrays, interference suppression, among others.
- LCAF algorithms incorporate into the solution application-specific requirements translated into a set of linear equations to be satisfied by the coefficients.
  - For example, if direction of arrival of the signal of interest is known, jammer suppression can take place through spatial filtering without the need of training signal, or in systems with constant-envelope modulation (e.g., M-PSK), a constant-modulus constraint can mitigate multipath propagation effects.

4.2 Constrained FIR Filters

Broadband Array Beamformer



Optimal Constrained MSE Filter

We look for

$$\min_{\mathbf{w}} \xi(k) \quad \text{s.t. } \mathbf{C}^H\mathbf{w} = \mathbf{f},$$

where

- $\xi(k) = E\left[|e(k)|^2\right]$
- $\mathbf{C}$ is the $MN \times p$ constraint matrix
- $\mathbf{f}$ is the $p \times 1$ gain vector

Optimal Constrained MSE Filter

The optimal beamformer is

$$\mathbf{w}(k) = \mathbf{R}^{-1}\mathbf{p} + \mathbf{R}^{-1}\mathbf{C}\left(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C}\right)^{-1}\left(\mathbf{f} - \mathbf{C}^H\mathbf{R}^{-1}\mathbf{p}\right)$$

where:

- $\mathbf{R} = E\left[\mathbf{x}(k)\mathbf{x}^H(k)\right]$ and $\mathbf{p} = E\left[d^*(k)\mathbf{x}(k)\right]$
- $\mathbf{w}(k) = \left[\mathbf{w}_1^T(k)\ \mathbf{w}_2^T(k)\ \cdots\ \mathbf{w}_M^T(k)\right]^T$
- $\mathbf{x}(k) = \left[\mathbf{x}_1^T(k)\ \mathbf{x}_2^T(k)\ \cdots\ \mathbf{x}_M^T(k)\right]^T$
- $\mathbf{x}_i^T(k) = [x_i(k)\ x_i(k-1)\ \cdots\ x_i(k-N+1)]$

The Constrained LS Beamformer

In the absence of statistical information, we may choose

$$\min_{\mathbf{w}} \left[\xi(k) = \sum_{i=0}^{k} \lambda^{k-i} |d(i) - \mathbf{w}^H \mathbf{x}(i)|^2 \right] \text{ s.t. } \mathbf{C}^H \mathbf{w} = \mathbf{f}$$

with $\lambda \in (0, 1]$,

The Constrained LS Beamformer

In the absence of statistical information, we may choose

$$\min_{\mathbf{w}} \left[\xi(k) = \sum_{i=0}^{k} \lambda^{k-i} |d(i) - \mathbf{w}^H \mathbf{x}(i)|^2 \right] \text{ s.t. } \mathbf{C}^H \mathbf{w} = \mathbf{f}$$

with $\lambda \in (0, 1]$, which gives, as solution,

The Constrained LS Beamformer

In the absence of statistical information, we may choose

$$\min_{\mathbf{w}} \left[\xi(k) = \sum_{i=0}^{k} \lambda^{k-i} |d(i) - \mathbf{w}^H \mathbf{x}(i)|^2 \right] \text{ s.t. } \mathbf{C}^H \mathbf{w} = \mathbf{f}$$

with $\lambda \in (0, 1]$, which gives, as solution,

$$\begin{aligned} \mathbf{w}(k) &= \mathbf{R}^{-1}(k)\mathbf{p}(k) \\ &\quad + \mathbf{R}^{-1}(k)\mathbf{C}\left(\mathbf{C}^H\mathbf{R}^{-1}(k)\mathbf{C}\right)^{-1}\left[\mathbf{f} - \mathbf{C}^H\mathbf{R}^{-1}(k)\mathbf{p}(k)\right], \end{aligned}$$

where

$\mathbf{R}(k) = \sum_{i=0}^{k} \lambda^{k-i}\mathbf{x}(i)\mathbf{x}^H(i)$, and $\mathbf{p}(k) = \sum_{i=0}^{k} \lambda^{k-i} d^*(i)\mathbf{x}(i)$.

The Constrained LMS Algorithm

A (cheaper) alternative cost function is

$$\min_{\mathbf{w}} \left[\xi(k) = \|\mathbf{w}(k) - \mathbf{w}(k-1)\|^2 + \mu|e(k)|^2\right] \text{ s.t. } \mathbf{C}^H\mathbf{w}(k) = \mathbf{f},$$

The Constrained LMS Algorithm

A (cheaper) alternative cost function is

$$\min_{\mathbf{w}} \left[\xi(k) = \|\mathbf{w}(k) - \mathbf{w}(k-1)\|^2 + \mu|e(k)|^2\right] \text{ s.t. } \mathbf{C}^H\mathbf{w}(k) = \mathbf{f},$$

which gives, as solution,

The Constrained LMS Algorithm

A (cheaper) alternative cost function is

$$\min_{\mathbf{w}} \left[\xi(k) = \|\mathbf{w}(k) - \mathbf{w}(k-1)\|^2 + \mu |e(k)|^2 \right] \text{ s.t. } \mathbf{C}^H \mathbf{w}(k) = \mathbf{f},$$

which gives, as solution,

$$\mathbf{w}(k) = \mathbf{w}(k-1) + \mu e^*(k) \left[\mathbf{I} - \mathbf{C} \left(\mathbf{C}^H \mathbf{C} \right)^{-1} \mathbf{C}^H \right] \mathbf{x}(k),$$

where $e(k) = d(k) - \mathbf{w}^H(k-1)\mathbf{x}(k)$, μ is a positive small constant called step size.

The Constrained LMS Algorithm

A (cheaper) alternative cost function is

$$\min_{\mathbf{w}} \left[\xi(k) = \|\mathbf{w}(k) - \mathbf{w}(k-1)\|^2 + \mu|e(k)|^2\right] \text{ s.t. } \mathbf{C}^H\mathbf{w}(k) = \mathbf{f},$$

which gives, as solution,

$$\mathbf{w}(k) = \mathbf{P}\left[\mathbf{w}(k-1) + \mu e^*(k)\mathbf{x}(k)\right] + \mathbf{F},$$

where $e(k) = d(k) - \mathbf{w}^H(k-1)\mathbf{x}(k)$, μ is a positive small constant called step size, $\mathbf{P} = \mathbf{C}\left(\mathbf{C}^H\mathbf{C}\right)^{-1}\mathbf{C}^H$, and $\mathbf{F} = \mathbf{C}\left(\mathbf{C}^H\mathbf{C}\right)^{-1}\mathbf{f}$.

The Constrained AP Algorithm

We may wish to trade complexity for speed of convergence:

$$\min_{\mathbf{w}} \left[\xi(k) = \|\mathbf{w}(k) - \mathbf{w}(k-1)\|^2\right] \text{ s.t. } \begin{cases} \mathbf{X}^T(k)\mathbf{w}^*(k) = \mathbf{d}(k) \\ \mathbf{C}^H\mathbf{w}(k) = \mathbf{f}, \end{cases}$$

where

- $\mathbf{d}(k) = [d(k)\ d(k-1)\ \cdots\ d(k-L+1)]^T$
- $\mathbf{X}(k) = [\mathbf{x}(k)\ \mathbf{x}(k-1)\ \cdots\ \mathbf{x}(k-L+1)]^T$

The Constrained AP Algorithm

We may wish to trade complexity for speed of convergence:

$$\min_{\mathbf{w}} \left[\xi(k) = \|\mathbf{w}(k) - \mathbf{w}(k-1)\|^2\right] \text{ s.t. } \begin{cases} \mathbf{X}^T(k)\mathbf{w}^*(k) = \mathbf{d}(k) \\ \mathbf{C}^H\mathbf{w}(k) = \mathbf{f}, \end{cases}$$

which gives, as solution,

The Constrained AP Algorithm

We may wish to trade complexity for speed of convergence:

$$\min_{\mathbf{w}} \left[\xi(k) = \|\mathbf{w}(k) - \mathbf{w}(k-1)\|^2\right] \text{ s.t. } \begin{cases} \mathbf{X}^T(k)\mathbf{w}^*(k) = \mathbf{d}(k) \\ \mathbf{C}^H\mathbf{w}(k) = \mathbf{f}, \end{cases}$$

which gives, as solution,

$$\mathbf{w}(k) = \mathbf{P}\left[\mathbf{w}(k-1) + \mu\mathbf{X}(k)\mathbf{t}(k)\right] + \mathbf{F}$$

where

- $\mathbf{e}(k) = \mathbf{d}(k) - \mathbf{X}^T(k)\mathbf{w}^*(k-1)$
- $\mathbf{t}(k) = \left[\mathbf{X}^H(k)\mathbf{P}\mathbf{X}(k)\right]^{-1}\mathbf{e}^*(k)$

Outline

1. Introduction and Fundamentals
2. Sensor Arrays and Spatial Filtering
3. Optimal Beamforming
4. Adaptive Beamforming
5. DoA Estimation with Microphone Arrays

5. DOA Estimation with Microphone Arrays

5.0 Signal Preparation

- It is usual to find a delayed signal represented by a multiplication of the signal with exponential $e^{j\omega_o\tau}$

- It is usual to find a delayed signal represented by a multiplication of the signal with exponential $e^{j\omega_o\tau}$
- First thing to note: when this is the case, the signal is narrow band with a center frequency in ω_0 (in the continuous-time domain, it corresponds to a carrier frequency $\Omega_0 = f_s\omega_0$)

- It is usual to find a delayed signal represented by a multiplication of the signal with exponential $e^{j\omega_o\tau}$

- First thing to note: when this is the case, the signal is narrow band with a center frequency in ω_0 (in the continuous-time domain, it corresponds to a carrier frequency $\Omega_0 = f_s\omega_0$)

- But, most importantly, the delay is well represented only if the signal is also analytic, i. e., having only non-negative frequency components.

- It is usual to find a delayed signal represented by a multiplication of the signal with exponential $e^{j\omega_o\tau}$

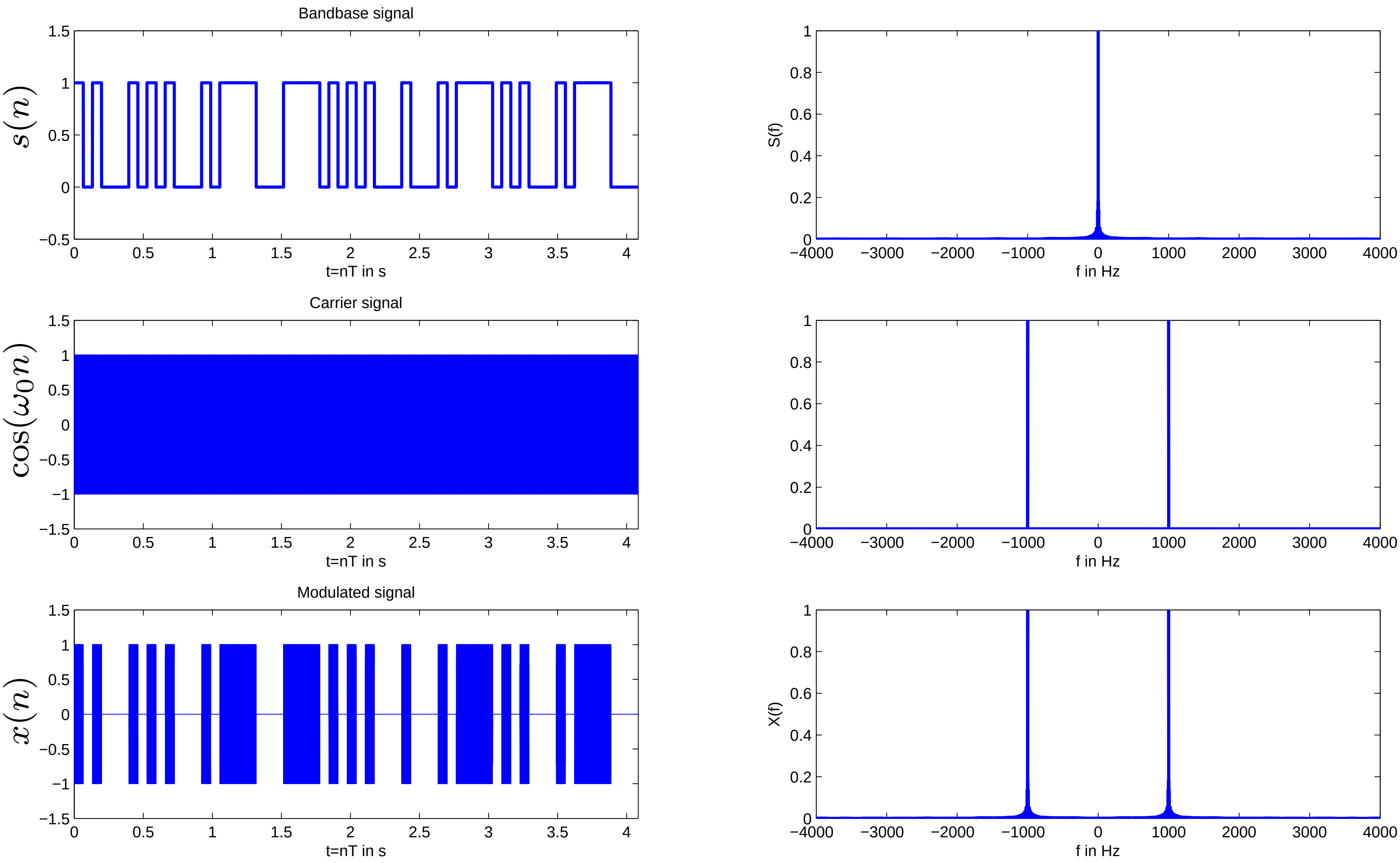
- First thing to note: when this is the case, the signal is narrow band with a center frequency in ω_0 (in the continuous-time domain, it corresponds to a carrier frequency $\Omega_0 = f_s\omega_0$)

- But, most importantly, the delay is well represented only if the signal is also analytic, i. e., having only non-negative frequency components.

- An analytic signal, mathematically, can be obtained by multiplying its Fourier transform by the continuous Heaviside step function:

$$X_a(e^{j\omega}) = 2X(e^{j\omega})u(\omega), u(\omega) = \begin{cases} 0, \omega < 0 \\ 1, \omega = 0 \\ 1, \omega > 0 \end{cases}$$

- Let $x(n) = s(n)\cos(\omega_0 n)$, $s(n)$ having a maximum frequency component (ω_m) much lower than ω_0:



- If $x(n) = s(n)e^{j\omega_0 n}$, then
  $x(n)e^{-j\omega_0 \tau} = s(n)e^{j\omega_0 (n-\tau)} \approx x(n-\tau)$ if $\tau \ll 1/\omega_m$

- If $x(n) = s(n)e^{j\omega_0 n}$, then
  $x(n)e^{-j\omega_0\tau} = s(n)e^{j\omega_0(n-\tau)} \approx x(n-\tau)$ if $\tau \ll 1/\omega_m$
- But if $x(n) = s(n)cos(\omega_0 n)$, then $x(n)e^{-j\omega_0\tau} \neq x(n-\tau)$

- If $x(n) = s(n)e^{j\omega_0 n}$, then $x(n)e^{-j\omega_0\tau} = s(n)e^{j\omega_0(n-\tau)} \approx x(n-\tau)$ if $\tau \ll 1/\omega_m$
- But if $x(n) = s(n)cos(\omega_0 n)$, then $x(n)e^{-j\omega_0\tau} \neq x(n-\tau)$
- We can make $x(n) = s(n)cos(\omega_0 n) = \underbrace{\frac{s(n)}{2}e^{j\omega_0 n}}_{x_+(n)} + \underbrace{\frac{s(n)}{2}e^{-j\omega_0 n}}_{x_-(n)}$ such that $x(n-\tau) \approx x_+ e^{-j\omega_0\tau} + x_-(n)e^{+j\omega_0\tau} = s(n)cos(\omega_0(n-\tau))$

- If $x(n) = s(n)e^{j\omega_0 n}$, then $x(n)e^{-j\omega_0\tau} = s(n)e^{j\omega_0(n-\tau)} \approx x(n-\tau)$ if $\tau \ll 1/\omega_m$
- But if $x(n) = s(n)cos(\omega_0 n)$, then $x(n)e^{-j\omega_0\tau} \neq x(n-\tau)$
- We can make $x(n) = s(n)cos(\omega_0 n) = \underbrace{\frac{s(n)}{2}e^{j\omega_0 n}}_{x_+(n)} + \underbrace{\frac{s(n)}{2}e^{-j\omega_0 n}}_{x_-(n)}$ such that
$$x(n-\tau) \approx x_+e^{-j\omega_0\tau} + x_-(n)e^{+j\omega_0\tau} = s(n)cos(\omega_0(n-\tau))$$
- $\cdots$ but, how to obtain $x_+(n)$ or a scaled copy? Using the Hilbert Transform $x_H(n) = \mathcal{HT}\{x(n)\}$ where
$$X_H(e^{j\omega}) = \begin{cases} jX(e^{j\omega}), -\pi < \omega < 0 \\ X(e^{j\omega}), \omega = 0 \\ -jX(e^{j\omega}), 0 < \omega < \pi \end{cases}$$

- Knowing that $x(n) = x_-(n) + x_+(n) = \mathcal{F}^{-1}\left\{X_-(e^{j\omega}) + X_+(e^{j\omega})\right\}$, we compute $y(n) = x(n) + jx_H(n)$

- Knowing that $x(n) = x_-(n) + x_+(n) = \mathcal{F}^{-1}\left\{X_-(e^{j\omega}) + X_+(e^{j\omega})\right\}$, we compute $y(n) = x(n) + jx_H(n)$

- $y(n) =$ $\mathcal{F}^{-1}\{X_-(e^{j\omega}) + X_+(e^{j\omega}) + j[\underbrace{jX_-(e^{j\omega}) - jX_+(e^{j\omega})}_{X_H(e^{j\omega})}]\}$

- Knowing that $x(n) = x_-(n) + x_+(n) = \mathcal{F}^{-1}\left\{X_-(e^{j\omega}) + X_+(e^{j\omega})\right\}$, we compute $y(n) = x(n) + jx_H(n)$

- $y(n) =$ $\mathcal{F}^{-1}\{X_-(e^{j\omega}) + X_+(e^{j\omega}) + j[\underbrace{jX_-(e^{j\omega}) - jX_+(e^{j\omega})}_{X_H(e^{j\omega})}]\}$

$$= \mathcal{F}^{-1}\{X_-(e^{j\omega}) + X_+(e^{j\omega}) - X_-(e^{j\omega}) + X_+(e^{j\omega})\}$$

- Knowing that $x(n) = x_-(n) + x_+(n) = \mathcal{F}^{-1}\left\{X_-(e^{j\omega}) + X_+(e^{j\omega})\right\}$, we compute $y(n) = x(n) + jx_H(n)$

- $y(n) =$
$\mathcal{F}^{-1}\{X_-(e^{j\omega}) + X_+(e^{j\omega}) + j[\underbrace{jX_-(e^{j\omega}) - jX_+(e^{j\omega})}_{X_H(e^{j\omega})}]\}$

$$= \mathcal{F}^{-1}\{X_-(e^{j\omega}) + X_+(e^{j\omega}) - X_-(e^{j\omega}) + X_+(e^{j\omega})\}$$

- Therefore $y(n) = 2\mathcal{F}^{-1}\{X_+(e^{j\omega})\} = s(n)e^{j\omega_0 n}$ which is analytic!

Signal Model

- Consider $x_m(t)$ the signal from the m-th microphone (prior to the A/D converter) corresponding to audio from D sources (directions θ_1 to θ_D) plus noise:
$x_m(t) = s_1(t - \bar{\tau}_m(\theta_1)) + \cdots + s_D(t - \bar{\tau}_m(\theta_D)) + n_m(t)$

Signal Model

- Consider $x_m(t)$ the signal from the m-th microphone (prior to the A/D converter) corresponding to audio from D sources (directions θ_1 to θ_D) plus noise:
  $x_m(t) = s_1(t - \bar{\tau}_m(\theta_1)) + \cdots + s_D(t - \bar{\tau}_m(\theta_D)) + n_m(t)$

- Assuming $\bar{\tau}_m(\theta_d) = T\tau_m(\theta_d)$ in s ($\tau_m(\theta_d)$ in number of samples), after the A/D converter and $\{.\} + j\mathcal{HT}\{.\}$ to make it an analytic signal, we could write
  $x_m(n) = s_1(n)e^{-j\omega_0\tau_m(\theta_1)} + \cdots + s_D(n)e^{-j\omega_0\tau_m(\theta_D)} + n_m(n)$

Signal Model

- Consider $x_m(t)$ the signal from the m-th microphone (prior to the A/D converter) corresponding to audio from D sources (directions θ_1 to θ_D) plus noise:
$$x_m(t) = s_1(t - \bar{\tau}_m(\theta_1)) + \cdots + s_D(t - \bar{\tau}_m(\theta_D)) + n_m(t)$$

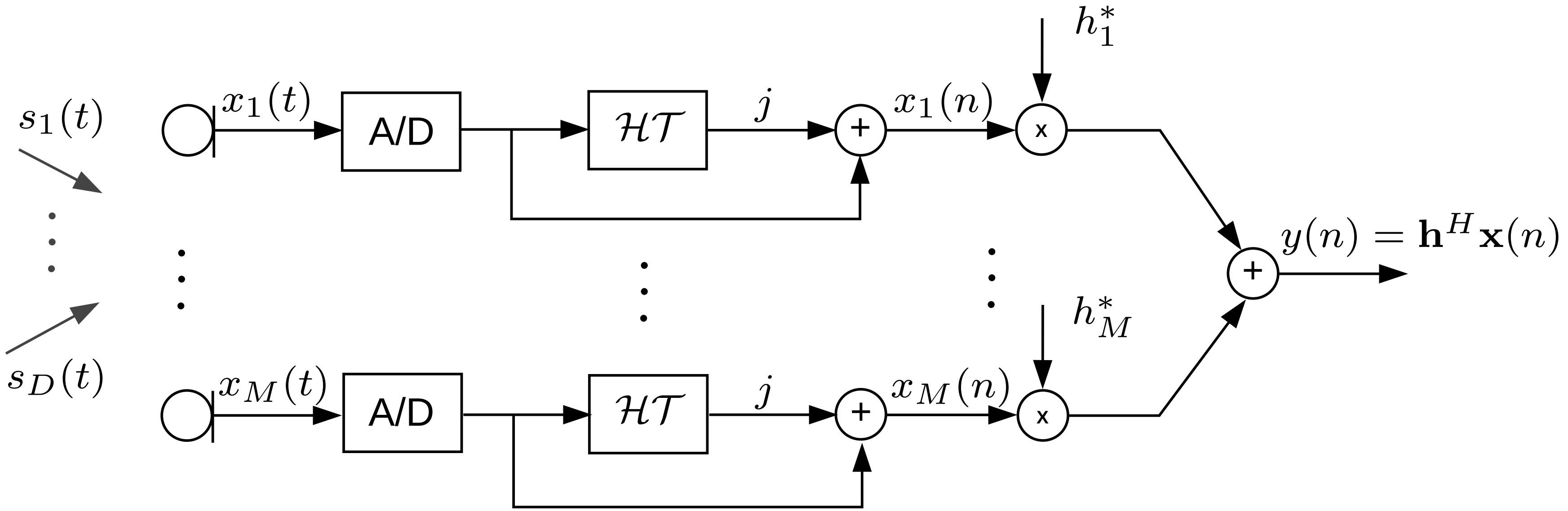
- Assuming $\bar{\tau}_m(\theta_d) = T\tau_m(\theta_d)$ in s ($\tau_m(\theta_d)$ in number of samples), after the A/D converter and $\{\boldsymbol{.}\} + j\mathcal{HT}\{\boldsymbol{.}\}$ to make it an analytic signal, we could write
$$x_m(n) = s_1(n)e^{-j\omega_0\tau_m(\theta_1)} + \cdots + s_D(n)e^{-j\omega_0\tau_m(\theta_D)} + n_m(n)$$

- For an array with M microphones, we would have:

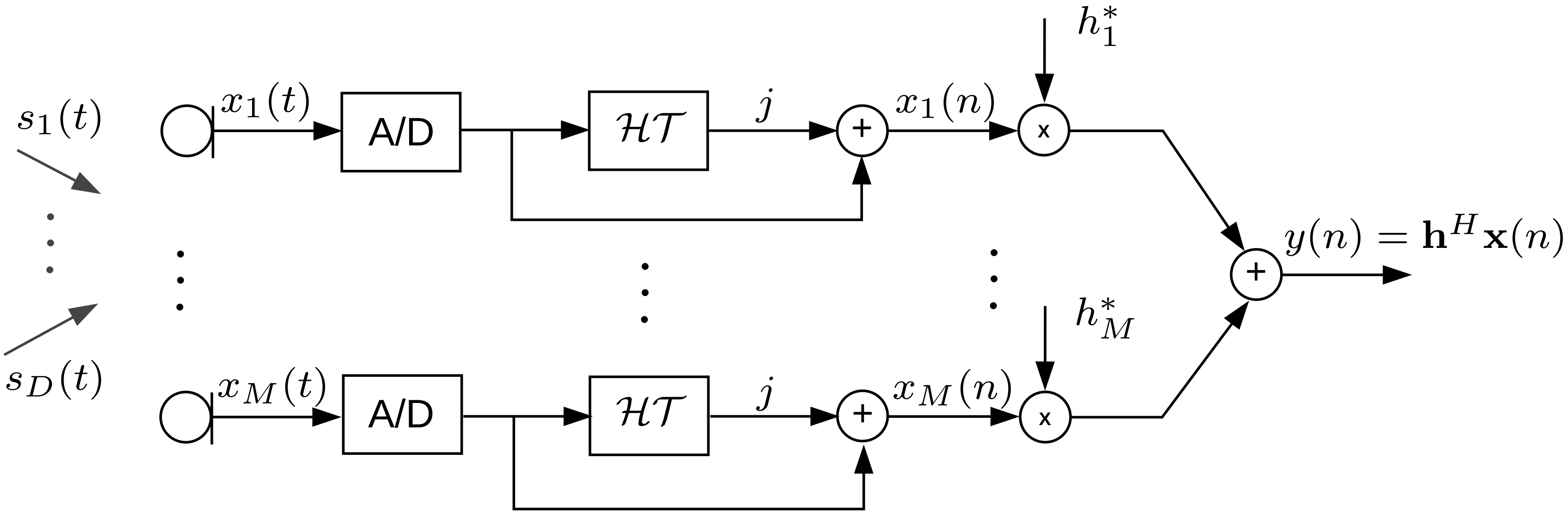
$$\underbrace{\mathbf{x}(n)}_{M\times 1} = \underbrace{\mathbf{A}}_{M\times D}\underbrace{\mathbf{s}(n)}_{D\times 1} + \underbrace{\mathbf{n}(n)}_{M\times 1}$$

5.1 Signal model

- Assume, initially, we have D narrowband signals coming from unknown directions:

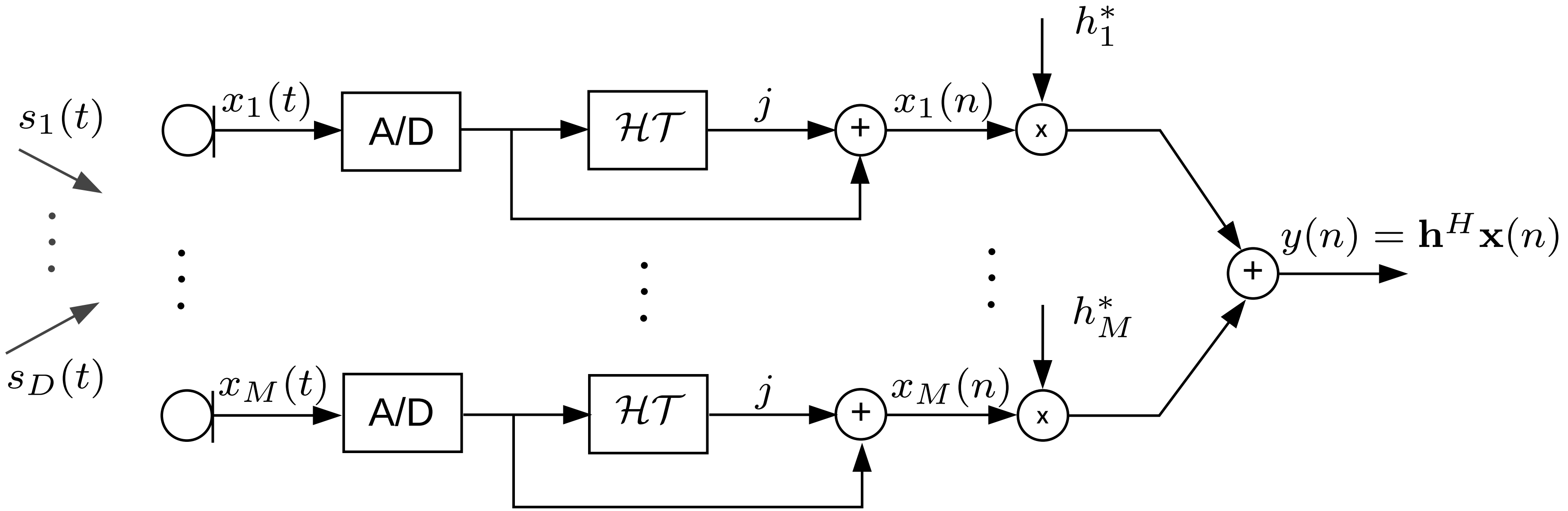


- Assume, initially, we have D narrowband signals coming from unknown directions:



- $$\mathbf{x}(n) = \begin{bmatrix} e^{-j\omega_0\tau_1(\theta_1)}s_1(n) + \cdots + e^{-j\omega_0\tau_1(\theta_D)}s_D(n) + n_1(n) \\ \vdots \\ e^{-j\omega_0\tau_M(\theta_1)}s_1(n) + \cdots + e^{-j\omega_0\tau_M(\theta_D)}s_D(n) + n_M(n) \end{bmatrix}$$

- Assume, initially, we have D narrowband signals coming from unknown directions:



- $$\mathbf{x}(n) = \begin{bmatrix} e^{-j\omega_0\tau_1(\theta_1)}s_1(n) + \cdots + e^{-j\omega_0\tau_1(\theta_D)}s_D(n) + n_1(n) \\ \vdots \\ e^{-j\omega_0\tau_M(\theta_1)}s_1(n) + \cdots + e^{-j\omega_0\tau_M(\theta_D)}s_D(n) + n_M(n) \end{bmatrix}$$

- Such that the output signal can be written as
$y(n) = \mathbf{h}^H\mathbf{x}(n) = \mathbf{h}^H\left[\mathbf{A}\mathbf{s}(n) + \mathbf{n}(n)\right]$

- If we now assume one single signal, $s(n)$, coming from direction θ, then
  $\mathbf{x}(n) = s(n)\mathbf{a}(\theta) + \mathbf{n}(n)$

- If we now assume one single signal, $s(n)$, coming from direction θ, then
  $\mathbf{x}(n) = s(n)\mathbf{a}(\theta) + \mathbf{n}(n)$
- And the output signal becomes
  $y(n) = \mathbf{h}^H\mathbf{a}(\theta)s(n) + \mathbf{h}^H\mathbf{n}(n)$

- If we now assume one single signal, $s(n)$, coming from direction θ, then
  $\mathbf{x}(n) = s(n)\mathbf{a}(\theta) + \mathbf{n}(n)$
- And the output signal becomes
  $y(n) = \mathbf{h}^H\mathbf{a}(\theta)s(n) + \mathbf{h}^H\mathbf{n}(n)$
- If we make $\mathbf{h}^H\mathbf{a}(\theta) = 1$, the output signal would correspond to $y(n) = s(n) + \underbrace{\mathbf{h}^H\mathbf{n}(n)}_{\text{noise}}$

- If we now assume one single signal, $s(n)$, coming from direction θ, then
  $\mathbf{x}(n) = s(n)\mathbf{a}(\theta) + \mathbf{n}(n)$

- And the output signal becomes
  $y(n) = \mathbf{h}^H\mathbf{a}(\theta)s(n) + \mathbf{h}^H\mathbf{n}(n)$

- If we make $\mathbf{h}^H\mathbf{a}(\theta) = 1$, the output signal would correspond to $y(n) = s(n) + \underbrace{\mathbf{h}^H\mathbf{n}(n)}_{\text{noise}}$

- Also note that $E[|y(n)|^2] = \mathbf{h}^H\mathbf{R}_x\mathbf{h}$, $\mathbf{R}_x = E[\mathbf{x}(n)\mathbf{x}^H(n)]$

5.2 Non-parametric methods: BF (beamforming a.k.a. Delay & Sum) and Capon

DS DoA

- If $\mathbf{x}(n)$ were spatially white, i.e. $\mathbf{R}_x = \mathbf{I}$, we would obtain $E[|y(n)|^2] = \mathbf{h}^H\mathbf{h}$

DS DoA

- If $\mathbf{x}(n)$ were spatially white, i.e. $\mathbf{R}_x = \mathbf{I}$, we would obtain $E[|y(n)|^2] = \mathbf{h}^H\mathbf{h}$
- Minimizing $E[|y(n)|^2] = \mathbf{h}^H\mathbf{h}$ s.t. $\mathbf{h}^H\mathbf{a}(\theta) = 1$, the result,

DS DoA

- If $\mathbf{x}(n)$ were spatially white, i.e. $\mathbf{R}_x = \mathbf{I}$, we would obtain $E[|y(n)|^2] = \mathbf{h}^H\mathbf{h}$
- Minimizing $E[|y(n)|^2] = \mathbf{h}^H\mathbf{h}$ s.t. $\mathbf{h}^H\mathbf{a}(\theta) = 1$, the result, after using Lagrange multiplier, taking the gradient,

  and equating to zero, is $\mathbf{h} = \mathbf{a}(\theta)/M$ which leads to $E[|y(n)|^2] = \frac{\mathbf{a}^H(\theta)\mathbf{R}_x\mathbf{a}(\theta)}{M^2}$

DS DoA

- If $\mathbf{x}(n)$ were spatially white, i.e. $\mathbf{R}_x = \mathbf{I}$, we would obtain $E[|y(n)|^2] = \mathbf{h}^H\mathbf{h}$
- Minimizing $E[|y(n)|^2] = \mathbf{h}^H\mathbf{h}$ s.t. $\mathbf{h}^H\mathbf{a}(\theta) = 1$, the result, after using Lagrange multiplier, taking the gradient, and equating to zero, is $\mathbf{h} = \mathbf{a}(\theta)/M$ which leads to $E[|y(n)|^2] = \frac{\mathbf{a}^H(\theta)\mathbf{R}_x\mathbf{a}(\theta)}{M^2}$
- Omitting factor $\frac{1}{M^2}$, we estimate the autocorrelation matrix as $\hat{\mathbf{R}}_x = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}(n)\mathbf{x}^H(n)$ and find the direction of interest by varying θ and obtaining the peak in $\boxed{P_{DS}(\theta) = \mathbf{a}^H(\theta)\hat{\mathbf{R}}_x\mathbf{a}(\theta)}$

Capon

- In the method known as Capon, we minimize $E[|y(n)|^2] = \mathbf{h}^H \mathbf{R}_x \mathbf{h}$ subject to $\mathbf{h}^H \mathbf{a}(\theta) = 1$

Capon

- In the method known as Capon, we minimize $E[|y(n)|^2] = \mathbf{h}^H \mathbf{R}_x \mathbf{h}$ subject to $\mathbf{h}^H \mathbf{a}(\theta) = 1$
- Using Lagrange multiplier, we write $\xi = \mathbf{h}^H \mathbf{R}_x \mathbf{h} + \lambda(\mathbf{h}^H \mathbf{a}(\theta) - 1)$, and make $\nabla_{\mathbf{h}} \xi = \mathbf{0}$ such that $\mathbf{h} = \frac{\mathbf{R}_x^{-1} \mathbf{a}(\theta)}{\mathbf{a}^H(\theta) \mathbf{R}_x^{-1} \mathbf{a}(\theta)}$

Capon

- In the method known as Capon, we minimize $E[|y(n)|^2] = \mathbf{h}^H \mathbf{R}_x \mathbf{h}$ subject to $\mathbf{h}^H \mathbf{a}(\theta) = 1$

- Using Lagrange multiplier, we write $\xi = \mathbf{h}^H \mathbf{R}_x \mathbf{h} + \lambda(\mathbf{h}^H \mathbf{a}(\theta) - 1)$, and make $\nabla_{\mathbf{h}} \xi = \mathbf{0}$ such that $\mathbf{h} = \frac{\mathbf{R}_x^{-1} \mathbf{a}(\theta)}{\mathbf{a}^H(\theta) \mathbf{R}_x^{-1} \mathbf{a}(\theta)}$

- Replacing the above coefficient vector in $E[|y(n)|^2]$, we obtain $E[|y(n)|^2] = \frac{1}{\mathbf{a}^H(\theta) \mathbf{R}_x^{-1} \mathbf{a}(\theta)}$

Capon

- In the method known as Capon, we minimize $E[|y(n)|^2] = \mathbf{h}^H \mathbf{R}_x \mathbf{h}$ subject to $\mathbf{h}^H \mathbf{a}(\theta) = 1$

- Using Lagrange multiplier, we write $\xi = \mathbf{h}^H \mathbf{R}_x \mathbf{h} + \lambda(\mathbf{h}^H \mathbf{a}(\theta) - 1)$, and make $\nabla_{\mathbf{h}} \xi = \mathbf{0}$ such that $\mathbf{h} = \frac{\mathbf{R}_x^{-1} \mathbf{a}(\theta)}{\mathbf{a}^H(\theta) \mathbf{R}_x^{-1} \mathbf{a}(\theta)}$

- Replacing the above coefficient vector in $E[|y(n)|^2]$, we obtain $E[|y(n)|^2] = \frac{1}{\mathbf{a}^H(\theta) \mathbf{R}_x^{-1} \mathbf{a}(\theta)}$

- Therefore, in the Capon DoA, we estimate $\hat{\mathbf{R}}_x = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}(n) \mathbf{x}^H(n)$ and find the direction of interest by varying θ and obtaining the peak in

$$\boxed{P_{CAPON}(\theta) = \frac{1}{\mathbf{a}^H(\theta) \hat{\mathbf{R}}_x^{-1} \mathbf{a}(\theta)}}$$

5.3 Eigenvalue-Based DoA

MUSIC

- Coming back to the previous model of D sources, we write $\mathbf{x}(n) = \mathbf{A}\mathbf{s}(n) + \mathbf{n}(n)$

MUSIC

- Coming back to the previous model of D sources, we write $\mathbf{x}(n) = \mathbf{A}\mathbf{s}(n) + \mathbf{n}(n)$
- We assume $D < M$ (number of signals lower than the number of sensors); this method is known as *parametric* for we make this assumption

MUSIC

- Coming back to the previous model of D sources, we write $\mathbf{x}(n) = \mathbf{A}\mathbf{s}(n) + \mathbf{n}(n)$
- We assume $D < M$ (number of signals lower than the number of sensors); this method is known as *parametric* for we make this assumption
- Also note that $\mathbf{A}$ is $M \times D$, $\mathbf{s}$ is $D \times 1$, and $\mathbf{n}(n)$ is $M \times 1$

MUSIC

- Coming back to the previous model of D sources, we write $\mathbf{x}(n) = \mathbf{A}\mathbf{s}(n) + \mathbf{n}(n)$
- We assume $D < M$ (number of signals lower than the number of sensors); this method is known as *parametric* for we make this assumption
- Also note that $\mathbf{A}$ is $M \times D$, $\mathbf{s}$ is $D \times 1$, and $\mathbf{n}(n)$ is $M \times 1$
- We then write $\mathbf{R}_x = E\left[\mathbf{x}(n)\mathbf{x}^H(n)\right] = \mathbf{A}\mathbf{R}_s\mathbf{A}^H + \mathbf{R}_n$, this last matrix becoming $\mathbf{R}_n = \sigma_n^2\mathbf{I}$ when assuming spatially white noise; $\mathbf{R}_s$ is the $D \times D$ autocorrelation matrix of the signal vector, i.e., $E\left[\mathbf{s}(n)\mathbf{s}^H(n)\right]$

MUSIC

- $\mathbf{R}_x = \mathbf{A}\mathbf{R}_s\mathbf{A}^H + \mathbf{R}_n$ with $D < M$ implies that $\mathbf{A}\mathbf{R}_s\mathbf{A}^H$ is singular (rank D), its determinant is equal to zero and, therefore, $\det\left[\mathbf{R}_x - \sigma_n^2\mathbf{I}\right] = 0$ and σ_n^2 is a (minimum) eigenvalue with multiplicity $M - D$

MUSIC

- $\mathbf{R}_x = \mathbf{A}\mathbf{R}_s\mathbf{A}^H + \mathbf{R}_n$ with $D < M$ implies that $\mathbf{A}\mathbf{R}_s\mathbf{A}^H$ is singular (rank D), its determinant is equal to zero and, therefore, $\det\left[\mathbf{R}_x - \sigma_n^2\mathbf{I}\right] = 0$ and σ_n^2 is a (minimum) eigenvalue with multiplicity $M - D$

- Spectral decomposition of matrix $\mathbf{R}_x$: vector $\mathbf{e}_m$ being an eigenvector of $\mathbf{R}_x$ means that $\mathbf{R}_x\mathbf{e}_m = \lambda_m\mathbf{e}_m$. Collecting all eigenvectors in matrix $\mathbf{E}$, we may write $\mathbf{R}_x\mathbf{E} = \mathbf{E}\boldsymbol{\Lambda} = \left[\mathbf{e}_1 \cdots \mathbf{e}_M\right]$ diag $\left\{\left[\lambda_1 \cdots \lambda_M\right]\right\}$
$\Rightarrow \mathbf{R}_x = \mathbf{E}\boldsymbol{\Lambda}\mathbf{E}^H$

MUSIC

- $\mathbf{R}_x = \mathbf{A}\mathbf{R}_s\mathbf{A}^H + \mathbf{R}_n$ with $D < M$ implies that $\mathbf{A}\mathbf{R}_s\mathbf{A}^H$ is singular (rank D), its determinant is equal to zero and, therefore, $\det\left[\mathbf{R}_x - \sigma_n^2\mathbf{I}\right] = 0$ and σ_n^2 is a (minimum) eigenvalue with multiplicity $M - D$

- Spectral decomposition of matrix $\mathbf{R}_x$: vector $\mathbf{e}_m$ being an eigenvector of $\mathbf{R}_x$ means that $\mathbf{R}_x\mathbf{e}_m = \lambda_m\mathbf{e}_m$. Collecting all eigenvectors in matrix $\mathbf{E}$, we may write $\mathbf{R}_x\mathbf{E} = \mathbf{E}\mathbf{\Lambda} = \left[\mathbf{e}_1 \cdots \mathbf{e}_M\right]$ diag $\{[\lambda_1 \cdots \lambda_M]\}$
$\Rightarrow \mathbf{R}_x = \mathbf{E}\mathbf{\Lambda}\mathbf{E}^H$

- Dividing matrix $\mathbf{E}$ in two parts, the first D columns and the last $N = M - D$ columns, we have:
$\mathbf{E} = [\underbrace{\mathbf{e}_1 \cdots \mathbf{e}_D}_{\mathbf{E}_S}\,\underbrace{\mathbf{e}_{D+1} \cdots \mathbf{e}_M}_{\mathbf{E}_N}] = [\mathbf{E}_S \;\; \mathbf{E}_N]$

MUSIC

- Noting that $\mathbf{E}\mathbf{E}^H = \mathbf{I}$, we can write $\mathbf{E}_S\mathbf{E}_S^H + \mathbf{E}_N\mathbf{E}_N^H = \mathbf{I}$

MUSIC

- Noting that $\mathbf{E}\mathbf{E}^H = \mathbf{I}$, we can write $\mathbf{E}_S\mathbf{E}_S^H + \mathbf{E}_N\mathbf{E}_N^H = \mathbf{I}$
- The columns of $\mathbf{E}_S$ span the D-dimensional signal subspace while the columns of $\mathbf{E}_N$ span the N-dimensional noise subspace

MUSIC

- Noting that $\mathbf{E}\mathbf{E}^H = \mathbf{I}$, we can write $\mathbf{E}_S\mathbf{E}_S^H + \mathbf{E}_N\mathbf{E}_N^H = \mathbf{I}$
- The columns of $\mathbf{E}_S$ span the D-dimensional signal subspace while the columns of $\mathbf{E}_N$ span the N-dimensional noise subspace
- A vector in the signal subspace is a linear combination of the columns of $\mathbf{E}_S$. An example: $\sum_{d=1}^{D} x_d \mathbf{e}_d = \mathbf{E}_S \mathbf{x}, \mathbf{x} = [x_1 \cdots x_D]^T$

MUSIC

- Noting that $\mathbf{E}\mathbf{E}^H = \mathbf{I}$, we can write $\mathbf{E}_S\mathbf{E}_S^H + \mathbf{E}_N\mathbf{E}_N^H = \mathbf{I}$
- The columns of $\mathbf{E}_S$ span the D-dimensional signal subspace while the columns of $\mathbf{E}_N$ span the N-dimensional noise subspace
- A vector in the signal subspace is a linear combination of the columns of $\mathbf{E}_S$. An example: $\sum_{d=1}^{D} x_d \mathbf{e}_d = \mathbf{E}_S\mathbf{x}, \mathbf{x} = [x_1 \cdots x_D]^T$
- We can find the distance d from a vector $\mathbf{v}$ to the signal subspace $\mathbf{E}_S$ by obtaining $\mathbf{x}$ that minimizes $d = |\mathbf{v} - \mathbf{E}_S\mathbf{x}|$; the result is $d^2 = \mathbf{v}^H\mathbf{E}_N\mathbf{E}_N^H\mathbf{v}$

MUSIC

- The squared distance from vector $\mathbf{a}(\theta)$ to the signal subspace (spanned by $\mathbf{E}_S$) is $d^2 = \mathbf{a}^H(\theta)\mathbf{E}_N\mathbf{E}_N^H\mathbf{a}(\theta)$

MUSIC

- The squared distance from vector $\mathbf{a}(\theta)$ to the signal subspace (spanned by $\mathbf{E}_S$) is $d^2 = \mathbf{a}^H(\theta)\mathbf{E}_N\mathbf{E}_N^H\mathbf{a}(\theta)$
- When θ belongs to $\{\theta_1 \cdots \theta_D\}$, this distance should be close to zero

MUSIC

- The squared distance from vector $\mathbf{a}(\theta)$ to the signal subspace (spanned by $\mathbf{E}_S$) is $d^2 = \mathbf{a}^H(\theta)\mathbf{E}_N\mathbf{E}_N^H\mathbf{a}(\theta)$
- When θ belongs to $\{\theta_1 \cdots \theta_D\}$, this distance should be close to zero
- Its inverse will present peaks. In algorithm MUSIC, we estimate D from the eigenvalues of $\hat{\mathbf{R}}_x$; from its eigenvectors, we form $\mathbf{E}_S$ and $\mathbf{E}_N$, and by varying θ, we shall find peaks in the directions of θ_1 to θ_D in

$$\boxed{P_{MUSIC}(\theta) = \frac{1}{d^2_{\mathbf{a}(\theta)}} = \frac{1}{\mathbf{a}^H(\theta)\mathbf{E}_N\mathbf{E}_N^H\mathbf{a}(\theta)}}$$

MUSIC

- The squared distance from vector $\mathbf{a}(\theta)$ to the signal subspace (spanned by $\mathbf{E}_S$) is $d^2 = \mathbf{a}^H(\theta)\mathbf{E}_N\mathbf{E}_N^H\mathbf{a}(\theta)$
- When θ belongs to $\{\theta_1 \cdots \theta_D\}$, this distance should be close to zero
- Its inverse will present peaks. In algorithm MUSIC, we estimate D from the eigenvalues of $\hat{\mathbf{R}}_x$; from its eigenvectors, we form $\mathbf{E}_S$ and $\mathbf{E}_N$, and by varying θ, we shall find peaks in the directions of θ_1 to θ_D in

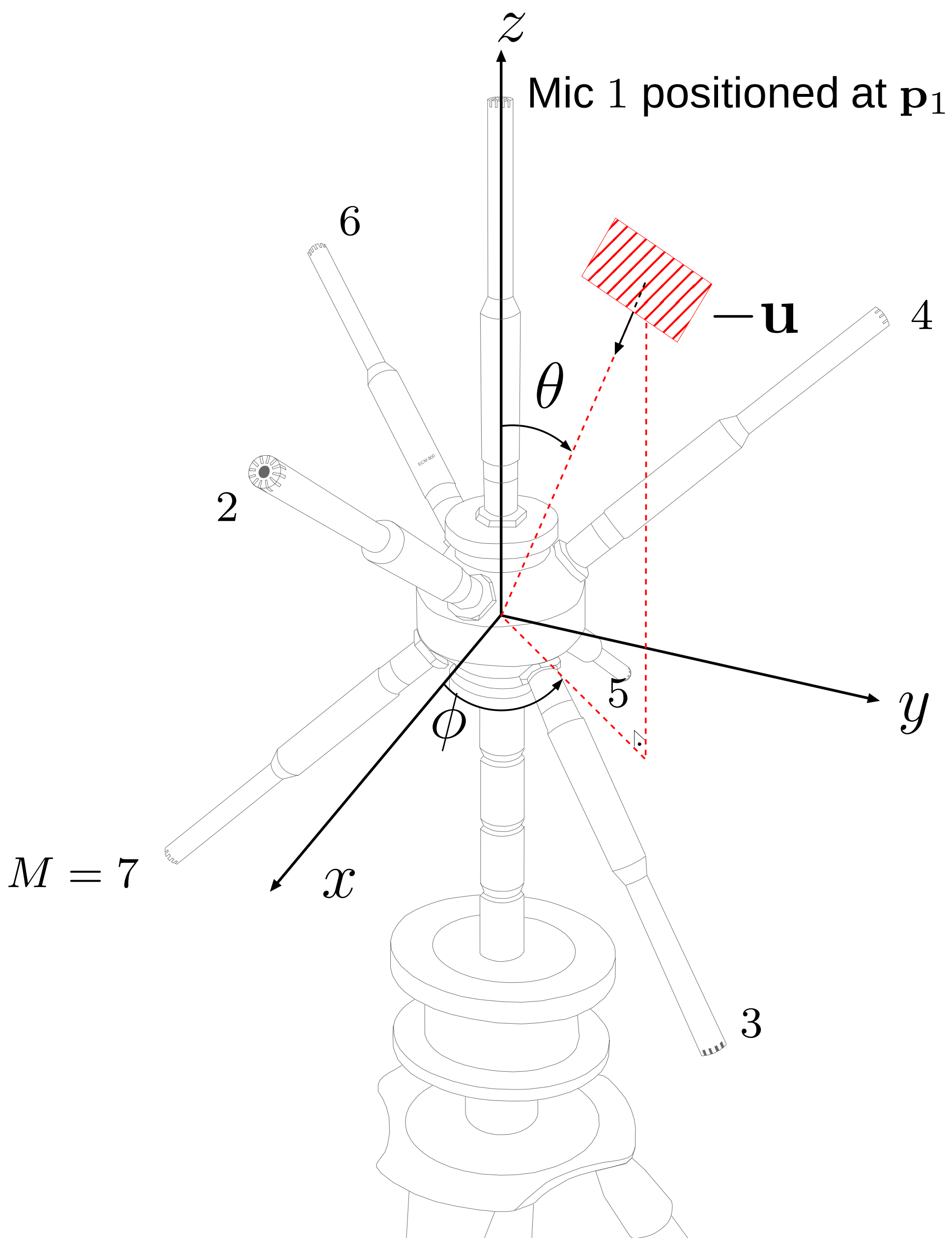
$$\boxed{P_{MUSIC}(\theta) = \frac{1}{d^2_{\mathbf{a}(\theta)}} = \frac{1}{\mathbf{a}^H(\theta)\mathbf{E}_N\mathbf{E}_N^H\mathbf{a}(\theta)}}$$

- If $\mathbf{R}_S$ is required, we compute

$$\mathbf{R}_S = \left(\mathbf{A}^H\mathbf{A}\right)^{-1}\mathbf{A}^H\left(\mathbf{R}_x - \sigma_n^2\mathbf{I}\right)\mathbf{A}\left(\mathbf{A}^H\mathbf{A}\right)^{-1}$$

5.4 GCC-Based DoA

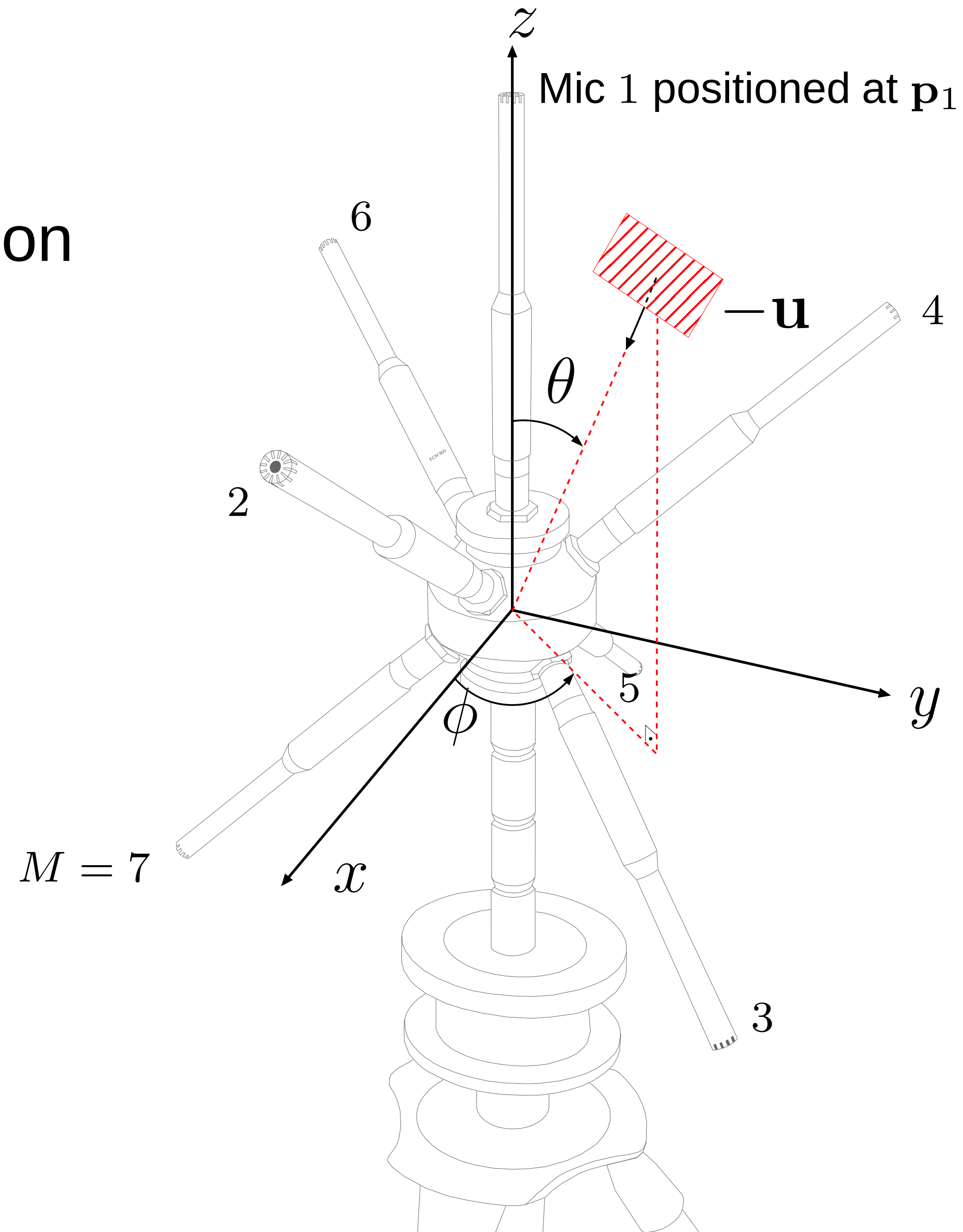
- M microphones of an array are in positions $\mathbf{p}_1$ to $\mathbf{p}_M$:



GCC

- M microphones of an array are in positions $\mathbf{p}_1$ to $\mathbf{p}_M$:

- $-\mathbf{u}$: unit vector in the direction of propagation

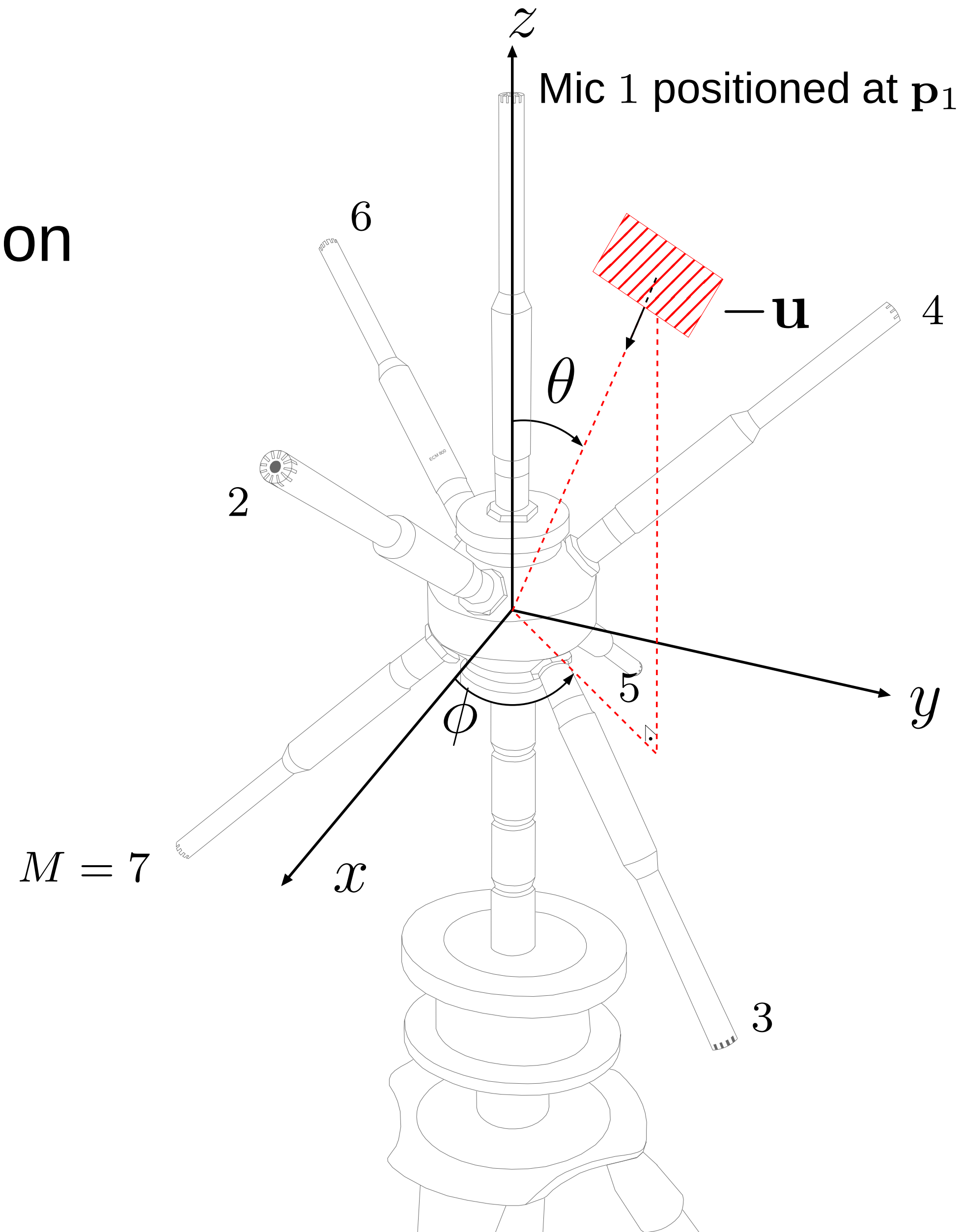


GCC

- M microphones of an array are in positions $\mathbf{p}_1$ to $\mathbf{p}_M$:

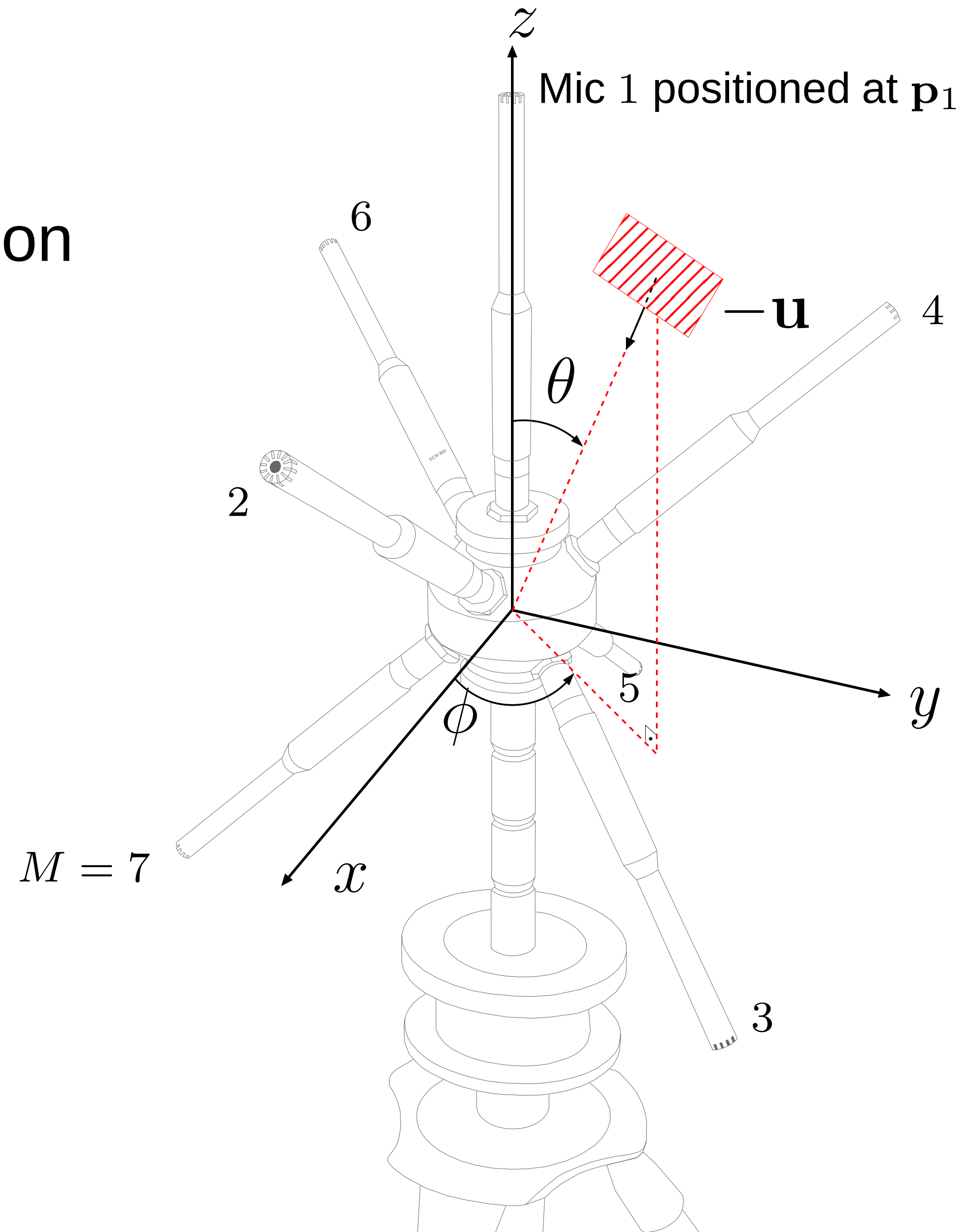
- $-\mathbf{u}$: unit vector in the direction of propagation

- θ: *grazing angle* ($\frac{\pi}{2}$ - elevation angle)



GCC

- M microphones of an array are in positions $\mathbf{p}_1$ to $\mathbf{p}_M$:
- $-\mathbf{u}$: unit vector in the direction of propagation
- θ: *grazing angle* ($\frac{\pi}{2}$ - elevation angle)
- ϕ: horizontal angle (*azimuth*)



GCC

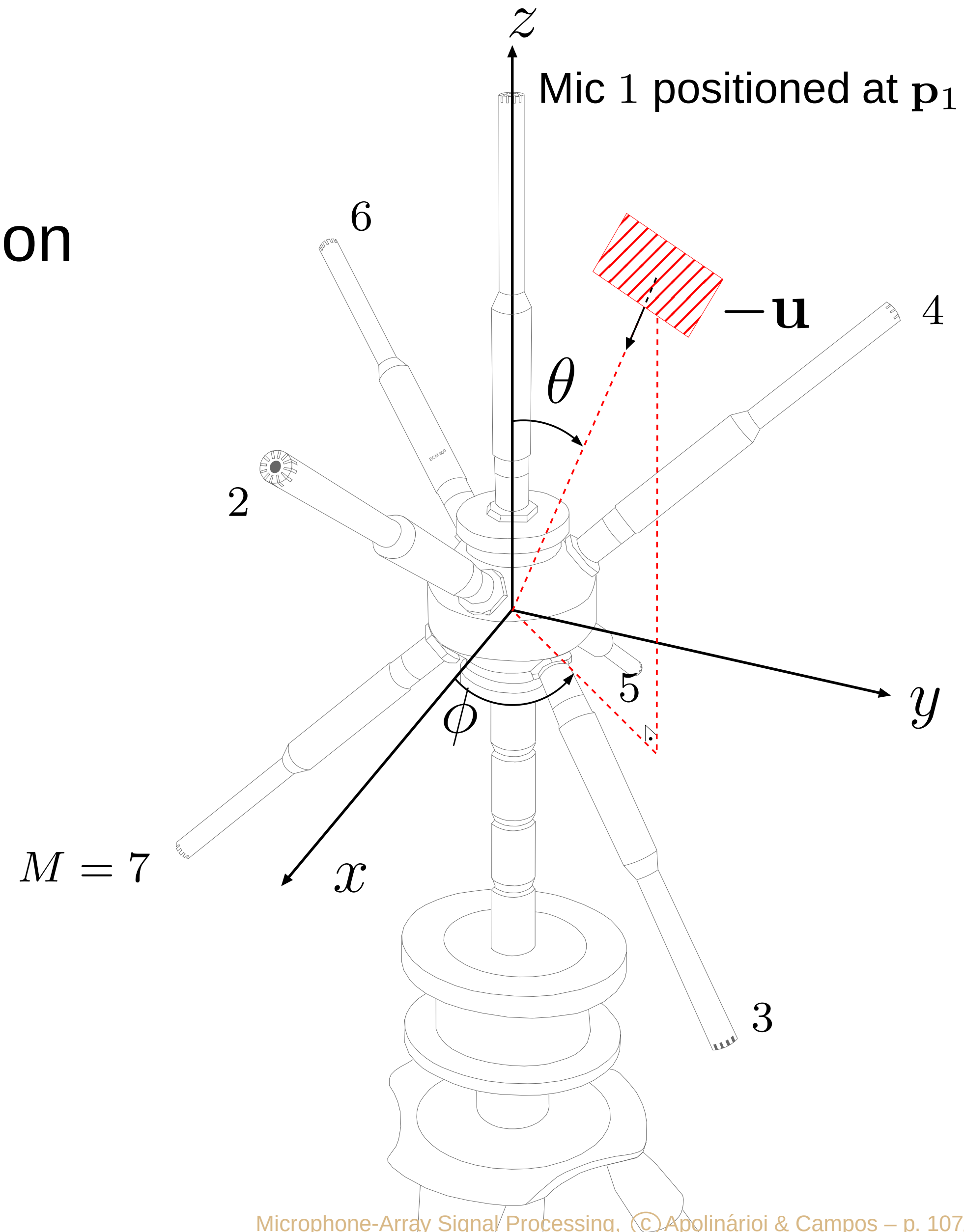
- M microphones of an array are in positions $\mathbf{p}_1$ to $\mathbf{p}_M$:

- $-\mathbf{u}$: unit vector in the direction of propagation

- θ: *grazing angle* ($\frac{\pi}{2}$ - elevation angle)

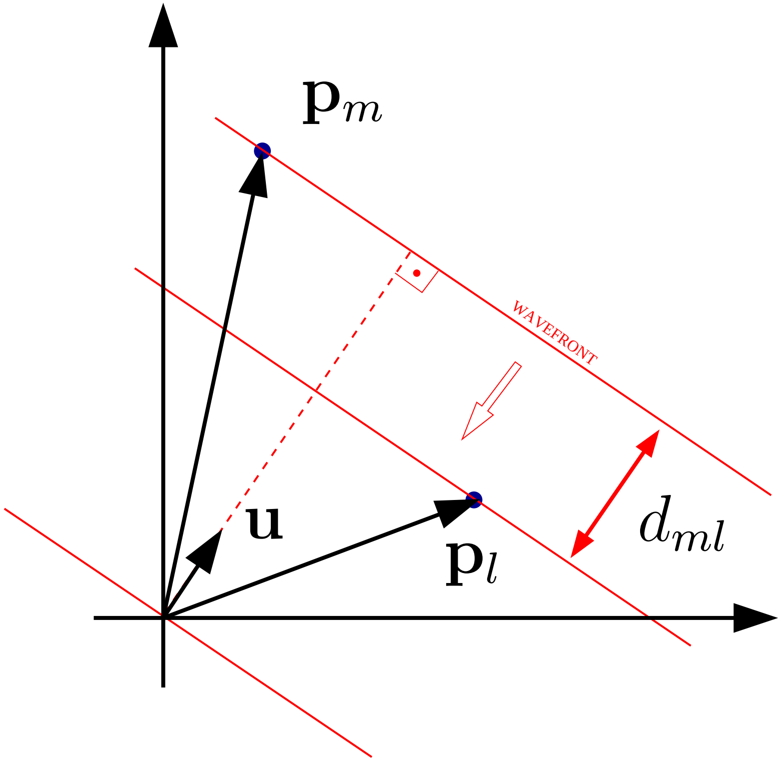
- ϕ: horizontal angle (*azimuth*)

- $$\mathbf{u} = \begin{bmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ \cos\theta \end{bmatrix}$$



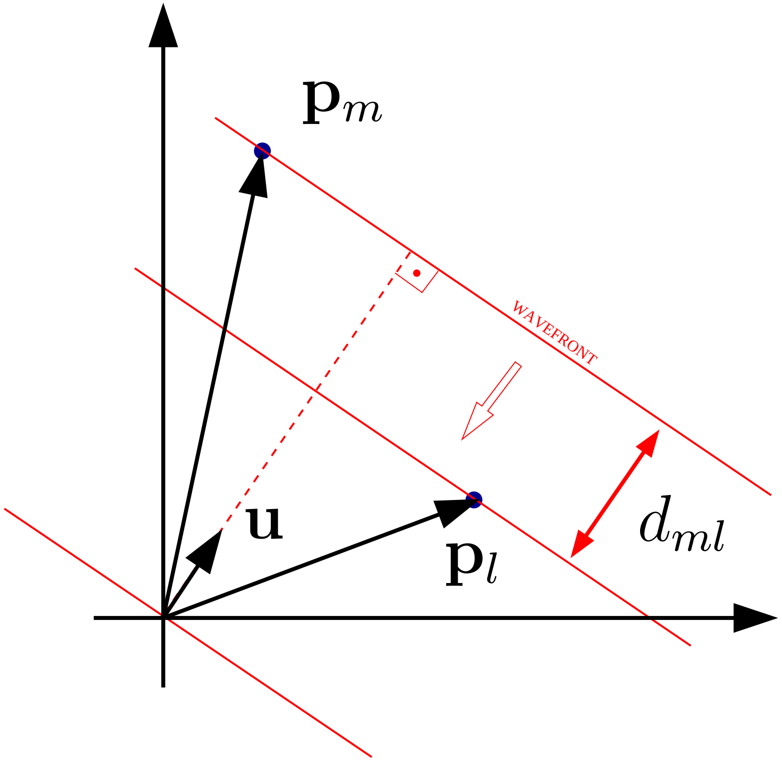
GCC

- We are interested in the TDoA between mics m and l



GCC

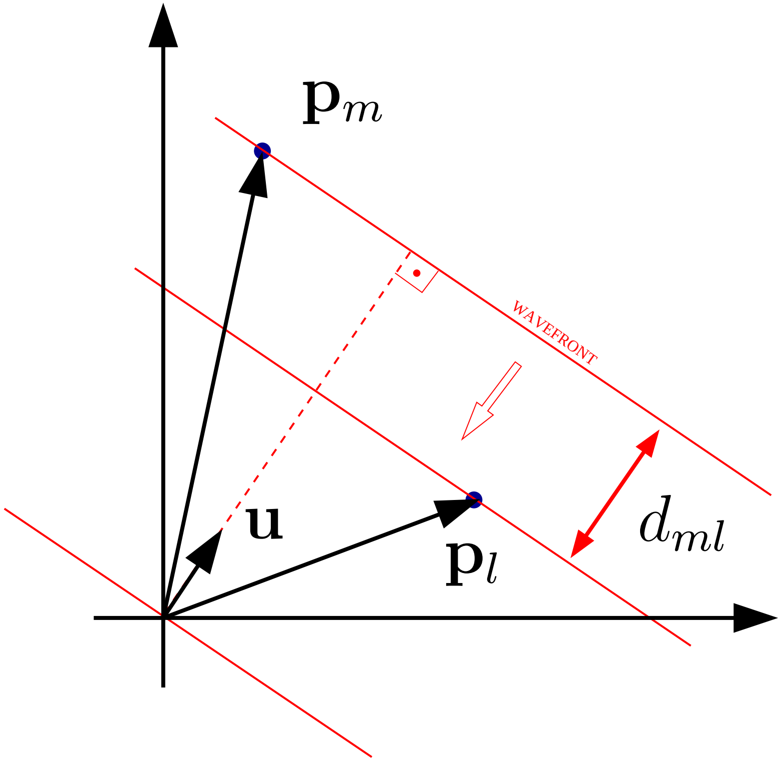
- We are interested in the TDoA between mics m and l
- Note that $d_{ml} = \mathbf{u}^T \underbrace{(\mathbf{p}_m - \mathbf{p}_l)}_{\Delta \mathbf{p}_{ml}}$



GCC

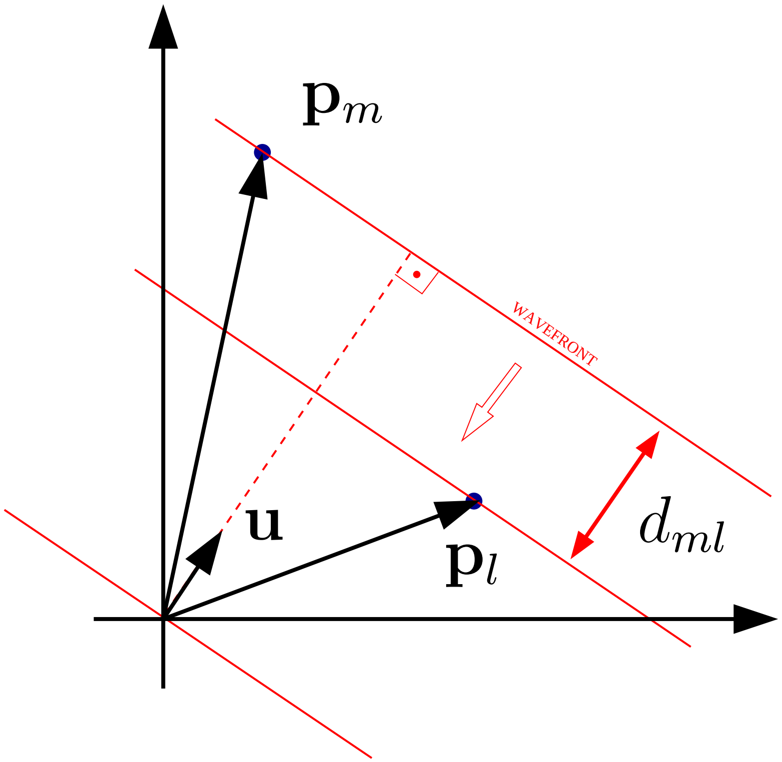
- We are interested in the TDoA between mics m and l
- Note that $d_{ml} = \mathbf{u}^T \underbrace{(\mathbf{p}_m - \mathbf{p}_l)}_{\Delta \mathbf{p}_{ml}}$
- TDoA:

$$\bar{\tau}_{ml} = \frac{d_{ml}}{v_{sound}} = \tau_{ml} T = \frac{\tau_{ml}}{f_s}$$



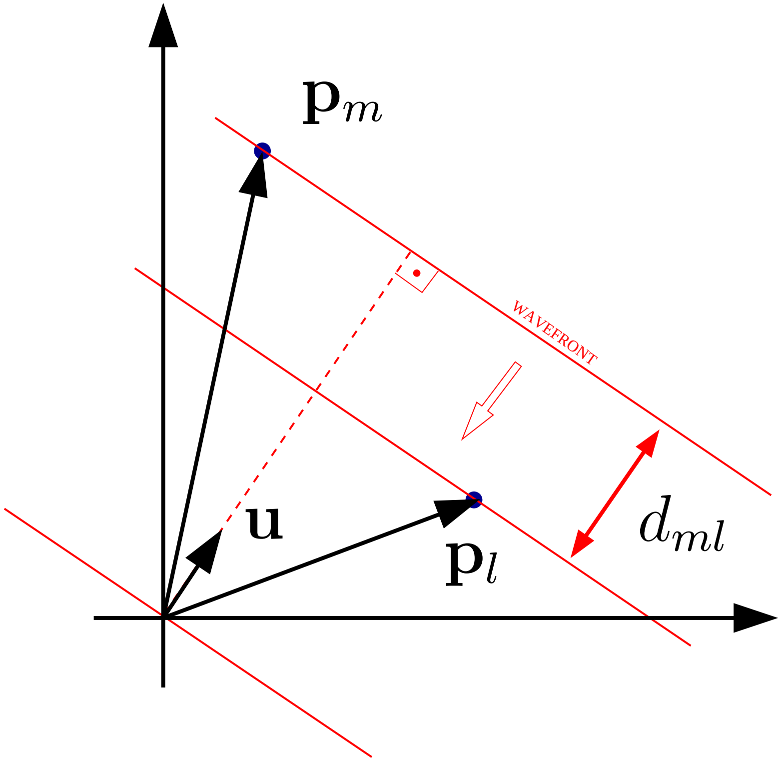
GCC

- We are interested in the TDoA between mics m and l
- Note that $d_{ml} = \mathbf{u}^T (\underbrace{\mathbf{p}_m - \mathbf{p}_l}_{\Delta \mathbf{p}_{ml}})$
- TDoA:
  $$\bar{\tau}_{ml} = \frac{d_{ml}}{v_{sound}} = \tau_{ml} T = \frac{\tau_{ml}}{f_s}$$
- τ_{ml} (in number of samples) is to be obtained from the peak of $\hat{r}_{x_m x_l}(\tau)$



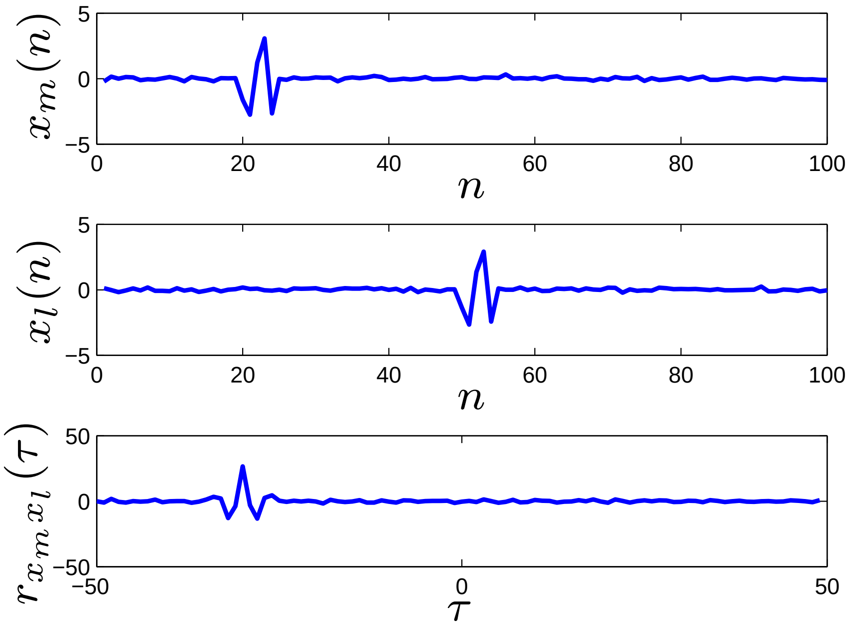
GCC

- We are interested in the TDoA between mics m and l
- Note that $d_{ml} = \mathbf{u}^T(\underbrace{\mathbf{p}_m - \mathbf{p}_l}_{\Delta\mathbf{p}_{ml}})$
- TDoA:
  $\bar{\tau}_{ml} = \frac{d_{ml}}{v_{sound}} = \tau_{ml}T = \frac{\tau_{ml}}{f_s}$
- τ_{ml} (in number of samples) is to be obtained from the peak of $\hat{r}_{x_m x_l}(\tau)$
- $r_{x_m x_l}(\tau) = E[x_m(n)x_l(n-\tau)]$



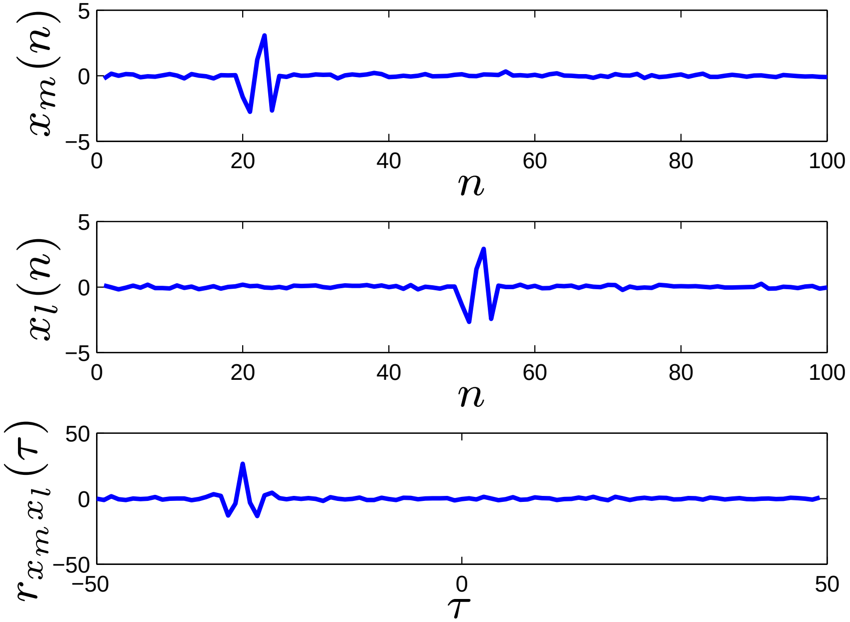
GCC

- When the sound frontwave first hits microphone m ($\tau_{ml} < 0$):

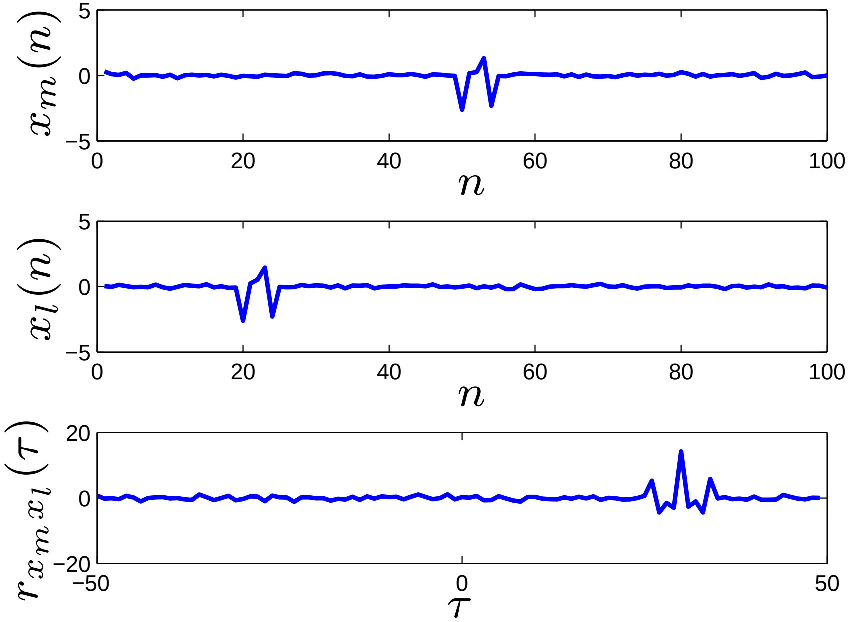


GCC

- When the sound frontwave first hits microphone m ($\tau_{ml} < 0$):



- When it first hits mic l ($\tau_{ml} > 0$):



GCC

- An estimate for the correlation can be given as:

$$\hat{r}_{x_m x_l}(\tau) = \sum_{-\infty}^{\infty} x_m(n) x_l(n-\tau) = x_m(\tau) * x_l(-\tau)$$

GCC

- An estimate for the correlation can be given as:
$$\hat{r}_{x_m x_l}(\tau) = \sum_{-\infty}^{\infty} x_m(n) x_l(n-\tau) = x_m(\tau) * x_l(-\tau)$$
- The cross-power spectrum density (CPSD):
$$\hat{R}_{x_m x_l}(e^{j\omega}) = \mathcal{F}\{x_m(\tau) * x_l(-\tau)\} = X_m(e^{j\omega}) X_l(e^{-j\omega})$$

GCC

- An estimate for the correlation can be given as:
  $\hat{r}_{x_m x_l}(\tau) = \sum_{-\infty}^{\infty} x_m(n) x_l(n-\tau) = x_m(\tau) * x_l(-\tau)$
- The cross-power spectrum density (CPSD):
  $\hat{R}_{x_m x_l}(e^{j\omega}) = \mathcal{F}\{x_m(\tau) * x_l(-\tau)\} = X_m(e^{j\omega}) X_l(e^{-j\omega})$
- We may assume the model
  $x_m(n) = s(n) * h_m(n) + n_m(n)$ and similarly for $x_l(n)$

GCC

- An estimate for the correlation can be given as:
  $\hat{r}_{x_m x_l}(\tau) = \sum_{-\infty}^{\infty} x_m(n) x_l(n-\tau) = x_m(\tau) * x_l(-\tau)$
- The cross-power spectrum density (CPSD):
  $\hat{R}_{x_m x_l}(e^{j\omega}) = \mathcal{F}\{x_m(\tau) * x_l(-\tau)\} = X_m(e^{j\omega}) X_l(e^{-j\omega})$
- We may assume the model
  $x_m(n) = s(n) * h_m(n) + n_m(n)$ and similarly for $x_l(n)$
- Hence, considering very small additive error and real sequences, we find
  $\hat{R}_{x_m x_l}(e^{j\omega}) \approx |S(e^{j\omega})|^2 H_m(e^{j\omega}) H_l^*(e^{j\omega})$ and
  $\hat{r}_{x_m x_l}(\tau) \approx \frac{1}{2\pi} \int_{-\pi}^{\pi} H_m(e^{j\omega}) H_l^*(e^{j\omega}) \hat{R}_s(e^{j\omega}) e^{j\omega\tau} d\omega$

GCC

- An estimate for the correlation can be given as:
$\hat{r}_{x_m x_l}(\tau) = \sum_{-\infty}^{\infty} x_m(n) x_l(n-\tau) = x_m(\tau) * x_l(-\tau)$
- The cross-power spectrum density (CPSD):
$\hat{R}_{x_m x_l}(e^{j\omega}) = \mathcal{F}\{x_m(\tau) * x_l(-\tau)\} = X_m(e^{j\omega}) X_l(e^{-j\omega})$
- We may assume the model
$x_m(n) = s(n) * h_m(n) + n_m(n)$ and similarly for $x_l(n)$
- Hence, considering very small additive error and real sequences, we find
$\hat{R}_{x_m x_l}(e^{j\omega}) \approx |S(e^{j\omega})|^2 H_m(e^{j\omega}) H_l^*(e^{j\omega})$ and
$\hat{r}_{x_m x_l}(\tau) \approx \frac{1}{2\pi} \int_{-\pi}^{\pi} H_m(e^{j\omega}) H_l^*(e^{j\omega}) \hat{R}_s(e^{j\omega}) e^{j\omega\tau} d\omega$
- Which motivates the GCC:

$$\boxed{r_{x_m x_l}^{G}(\tau) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \psi(\omega) \hat{R}_{x_m x_l}(e^{j\omega}) e^{j\omega\tau} d\omega}$$

Types of $\psi(\omega)$

- Classical cross-correlation:

  $\psi(\omega) = 1$

Types of $\psi(\omega)$

- Classical cross-correlation:

  $$\psi(\omega) = 1$$

- Maximum Likelihood (ML):

  $$\psi(\omega) = \frac{|X_m(e^{j\omega})||X_l(e^{j\omega})|}{\hat{R}_{n_n}(e^{j\omega})\hat{R}_{x_m}(e^{j\omega})+\hat{R}_{n_l}(e^{j\omega})\hat{R}_{x_l}(e^{j\omega})}$$

Types of $\psi(\omega)$

GCC

- Classical cross-correlation:

  $\psi(\omega) = 1$

- Maximum Likelihood (ML):

  $$\psi(\omega) = \frac{|X_m(e^{j\omega})||X_l(e^{j\omega})|}{\hat{R}_{nn}(e^{j\omega})\hat{R}_{xm}(e^{j\omega})+\hat{R}_{n_l}(e^{j\omega})\hat{R}_{x_l}(e^{j\omega})}$$

  - $\hat{R}_{x_m}(e^{j\omega}) = |X_m(e^{j\omega})|^2$
  - $\hat{R}_{x_l}(e^{j\omega}) = |X_l(e^{j\omega})|^2$
  - $\hat{R}_{n_m}(e^{j\omega}) = |N_m(e^{j\omega})|^2$ (estimated during silence interval)
  - $\hat{R}_{n_l}(e^{j\omega}) = |N_l(e^{j\omega})|^2$ (estimated during silence interval)

GCC

- PHAT (Phase Transform):

$$\psi(\omega) = \frac{1}{|\hat{R}_{x_m x_l}(e^{j\omega})}|$$

GCC

- PHAT (Phase Transform):
  $\psi(\omega) = \frac{1}{|\hat{R}_{x_mx_l}(e^{j\omega})}|$

- Replacing this function in the expression of $r^{G}_{x_mx_l}(\tau)$:
  $r^{PHAT}_{x_mx_l}(\tau) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{\hat{R}_{x_mx_l}(e^{j\omega})}{|\hat{R}_{x_mx_l}(e^{j\omega})|} e^{j\omega\tau} d\omega$ in which,
  after making $\hat{R}_{x_mx_l}(e^{j\omega}) = |S(e^{j\omega})|^2 H_m(e^{j\omega}) H_l^*(e^{j\omega})$,
  we have $r^{PHAT}_{x_mx_l}(\tau) = \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{j(\sphericalangle H_m - \sphericalangle H_l + \omega\pi)} d\omega$

GCC

- PHAT (Phase Transform):
  $\psi(\omega) = \frac{1}{|\hat{R}_{x_m x_l}(e^{j\omega})}|$

- Replacing this function in the expression of $r^{G}_{x_m x_l}(\tau)$:
  $r^{PHAT}_{x_m x_l}(\tau) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{\hat{R}_{x_m x_l}(e^{j\omega})}{|\hat{R}_{x_m x_l}(e^{j\omega})|} e^{j\omega\tau} d\omega$ in which,
  after making $\hat{R}_{x_m x_l}(e^{j\omega}) = |S(e^{j\omega})|^2 H_m(e^{j\omega}) H_l^*(e^{j\omega})$,
  we have $r^{PHAT}_{x_m x_l}(\tau) = \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{j(\sphericalangle H_m - \sphericalangle H_l + \omega\pi)} d\omega$

- For the PHAT, in case of having
  $h_m(n) = \alpha_m \delta(n)$ and $h_l(n) = \alpha_l \delta(n - \Delta\tau)$,
  the cross-correlation would be
  $r^{PHAT}_{x_m x_l}(\tau) = \delta(\tau + \Delta\tau) \Rightarrow$ peak in $\tau_{ml} = -\Delta\tau$
  (a perfect indication of a temporal delay!)

LS solution

- Assuming we have all possible ($M(M-1)/2$) delays τ_{ml}, we want angles ϕ and θ

LS solution

- Assuming we have all possible ($M(M-1)/2$) delays τ_{ml}, we want angles ϕ and θ
- We define a cost function:

$$\xi = \left(\bar{\tau}_{12} - \Delta\bar{\mathbf{p}}_{12}^{T}\mathbf{u}\right)^{2} + \cdots + \left(\bar{\tau}_{(M-1)M} - \Delta\bar{\mathbf{p}}_{(M-1)M}^{T}\mathbf{u}\right)^{2}$$

with $\bar{\tau}_{ml} = \tau_{ml}/f_s$ and $\Delta\bar{\mathbf{p}}_{ml} = (\mathbf{p}_m - \mathbf{p}_l)/v_{sound}$

LS solution

- Assuming we have all possible $(M(M-1)/2)$ delays τ_{ml}, we want angles ϕ and θ

- We define a cost function:

  $$\xi = \left(\bar{\tau}_{12} - \Delta\bar{\mathbf{p}}_{12}^T\mathbf{u}\right)^2 + \cdots + \left(\bar{\tau}_{(M-1)M} - \Delta\bar{\mathbf{p}}_{(M-1)M}^T\mathbf{u}\right)^2$$

  with $\bar{\tau}_{ml} = \tau_{ml}/f_s$ and $\Delta\bar{\mathbf{p}}_{ml} = (\mathbf{p}_m - \mathbf{p}_l)/v_{sound}$

- We then find $\mathbf{u}$ that minimizes ξ by making $\nabla_{\mathbf{u}}\xi = \mathbf{0}$:

  $$\mathbf{A}\mathbf{u} = \mathbf{b}$$

  where $\mathbf{A} = \Delta\bar{\mathbf{p}}_{12}\Delta\bar{\mathbf{p}}_{12}^T + \cdots + \Delta\bar{\mathbf{p}}_{(M-1)M}\Delta\bar{\mathbf{p}}_{(M-1)M}^T$

  and $\mathbf{b} = \bar{\tau}_{12}\Delta\bar{\mathbf{p}}_{12} + \cdots + \bar{\tau}_{(M-1)M}\Delta\bar{\mathbf{p}}_{(M-1)M}$

LS solution

- Assuming we have all possible $(M(M-1)/2)$ delays τ_{ml}, we want angles ϕ and θ
- We define a cost function:
  $$\xi = (\bar{\tau}_{12} - \Delta\bar{\mathbf{p}}_{12}^T\mathbf{u})^2 + \cdots + \left(\bar{\tau}_{(M-1)M} - \Delta\bar{\mathbf{p}}_{(M-1)M}^T\mathbf{u}\right)^2$$
  with $\bar{\tau}_{ml} = \tau_{ml}/f_s$ and $\Delta\bar{\mathbf{p}}_{ml} = (\mathbf{p}_m - \mathbf{p}_l)/v_{sound}$
- We then find $\mathbf{u}$ that minimizes ξ by making $\nabla_{\mathbf{u}}\xi = \mathbf{0}$:
  $$\mathbf{A}\mathbf{u} = \mathbf{b}$$
  where $\mathbf{A} = \Delta\bar{\mathbf{p}}_{12}\Delta\bar{\mathbf{p}}_{12}^T + \cdots + \Delta\bar{\mathbf{p}}_{(M-1)M}\Delta\bar{\mathbf{p}}_{(M-1)M}^T$
  and $\mathbf{b} = \bar{\tau}_{12}\Delta\bar{\mathbf{p}}_{12} + \cdots + \bar{\tau}_{(M-1)M}\Delta\bar{\mathbf{p}}_{(M-1)M}$
- And this unit vector is given as $\mathbf{u} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \mathbf{A}^{-1}\mathbf{b}$

Azimuth and elevation

- Knowing $\mathbf{u}$ and also the fact that it corresponds to

$$\begin{bmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ \cos\theta \end{bmatrix}, \cdots$$

Azimuth and elevation

- Knowing $\mathbf{u}$ and also the fact that it corresponds to

$$\begin{bmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ \cos\theta \end{bmatrix}, \cdots$$

- $\cdots$ we compute the azimuth:

$$\boxed{\phi = \arctan\frac{u_y}{u_x}}$$

Azimuth and elevation

- Knowing $\mathbf{u}$ and also the fact that it corresponds to

$$\begin{bmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ \cos\theta \end{bmatrix}, \cdots$$

- $\cdots$ we compute the azimuth:

$$\boxed{\phi = \arctan\frac{u_y}{u_x}}$$

- And the elevation:

$$\boxed{\text{elevation} = 90^\circ - \theta = 90^\circ - \arccos u_z}$$

Last slide ☺

Thank you!