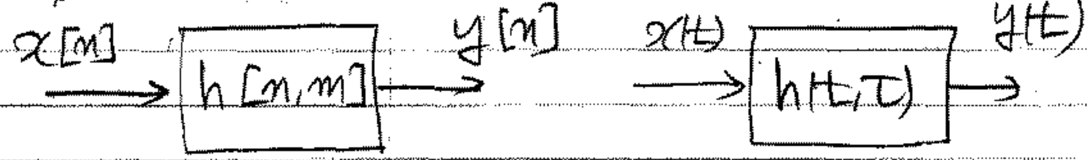
MIMO systems

○ Review of Linear SISO systems

- linear system

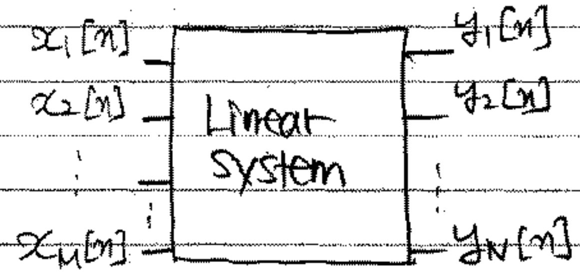


$$y[n] = \sum_{m=-\infty}^{\infty} h[n,m]\, x[m], \quad y(t) = \int_{-\infty}^{\infty} h(t,\tau)\, x(\tau)\, d\tau$$

- linear time-invariant system

$$y[n] = \sum_{m=-\infty}^{\infty} h[n-m]\, x[m], \quad y(t) = \int_{-\infty}^{\infty} h(t-\tau)\, x(\tau)\, d\tau$$

○ Discrete-time linear MIMO systems

- 

- $$\underline{x}[n] = \begin{bmatrix} \sum_{m=-\infty}^{\infty} x_1[n]\, \delta_1[n-m] \\ \sum_{m=-\infty}^{\infty} x_2[n]\, \delta_2[n-m] \\ \vdots \\ \sum_{m=-\infty}^{\infty} x_M[n]\, \delta_M[n-m] \end{bmatrix}$$

$$\underline{y}[n] = L\left(\underline{x}[n]\right)$$

$$= L\left(\begin{bmatrix} \sum_{m=-\infty}^{\infty} x_1[m]\,\delta_1[n-m] \\ \vdots \\ \sum_{m=-\infty}^{\infty} x_M[m]\,\delta_M[n-m] \end{bmatrix}\right)$$

$$= L\left(\begin{bmatrix} \sum_{m=-\infty}^{\infty} x_1[m]\,\delta_1[n-m] \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \sum_{m=-\infty}^{\infty} x_2[m]\,\delta_2[n-m] \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \cdots + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \sum_{m=-\infty}^{\infty} x_M[m]\,\delta_M[n-m] \end{bmatrix}\right)$$

$$= \sum_{m=-\infty}^{\infty} x_1[m]\, L\left(\begin{bmatrix} \delta_1[n-m] \\ 0 \\ \vdots \\ 0 \end{bmatrix}\right)$$

$$+ \sum_{m=-\infty}^{\infty} x_2[m]\, L\left(\begin{bmatrix} 0 \\ \delta_2[n-m] \\ 0 \\ \vdots \\ 0 \end{bmatrix}\right)$$

$$+ \cdots$$

$$+ \sum_{m=-\infty}^{\infty} x_M[m]\, L\left(\begin{bmatrix} 0 \\ \vdots \\ 0 \\ \delta_M[n-m] \end{bmatrix}\right)$$

$$\triangleq \sum_{m=-\infty}^{\infty} x_1[m]\, \underline{h}_1[n,m]$$

$$+ \sum_{m=-\infty}^{\infty} x_2[m]\, h_2[n,m]$$

$$+ \sum_{m=-\infty}^{\infty} x_M[m]\, \underline{h}_M[n,m]$$

$$= \sum_{m=-\infty}^{\infty} \begin{bmatrix} \underline{h}_1[n,m] & \underline{h}_2[n,m] & \cdots & \underline{h}_M[n,m] \end{bmatrix} \begin{bmatrix} x_1[m] \\ x_2[m] \\ \vdots \\ x_M[m] \end{bmatrix}$$

$$\triangleq \sum_{m=-\infty}^{\infty} H[n,m]\, \underline{x}[m]$$

- $\left[H[n,m]\right]_{ij}$ is the response at the i-th output and time n to the impulse applied at the j-th input and time m. $i=1,2,\cdots,N$. $j=1,2,\cdots,M$

- If the linear system is time-invariant then

$$\underline{y}[n] = \sum_{m=-\infty}^{\infty} H[n-m]\, \underline{x}[m] = \sum_{m=-\infty}^{\infty} H[m]\, \underline{x}[n-m]$$

○ Continuous-time linear MIMO systems

$$\underline{y}(t) = \int_{-\infty}^{\infty} H(t,\tau)\, \underline{x}(\tau)\, d\tau$$

where $\left[H(t,\tau)\right]_{ij}$ is the response at the i-th output and time t to the impulse at the j-th input and time τ.

- If LTI, then

$$\underline{y}(t) = \int_{-\infty}^{\infty} H(t-\tau)\,\underline{x}(\tau)\,d\tau = \int_{-\infty}^{\infty} H(\tau)\,\underline{x}(t-\tau)\,d\tau$$

○ Fourier transform of CT MIMO LTI system

- $$y_i(t) = \sum_{j=1}^{M} \left[H(t-\tau)\right]_{ij} x_j(\tau)\,d\tau$$

$\mathcal{F} \updownarrow$

$$Y_i(f) = \sum_{j=1}^{M} \left[H(f)\right]_{ij} X_j(f)$$

$$\boxed{\underline{Y}(f) = H(f)\,\underline{X}(f)}$$

where $\underline{Y}(f) = \begin{bmatrix} Y_1(f) \\ Y_2(f) \\ \vdots \\ Y_N(f) \end{bmatrix}$, $\underline{X}(f) = \begin{bmatrix} X_1(f) \\ X_2(f) \\ \vdots \\ X_M(f) \end{bmatrix}$,

$$H(f) \triangleq \begin{bmatrix} H_{1,1}(f) & H_{1,2}(f) & \cdots & H_{1,M}(f) \\ & & \vdots & \\ & & & H_{N,M}(f) \end{bmatrix}$$

○ Z-transform of DT MIMO LTI system

- $$\underline{y}[n] = \sum_{m=-\infty}^{\infty} H[m]\,\underline{x}[n-m]$$

$Z \updownarrow$

$$\boxed{\underline{Y}(z) = H(z)\,\underline{X}(z)}$$

where $\underline{Y}(z) \triangleq \sum_{m=-\infty}^{\infty} \underline{y}[m] z^{-m}$, $\underline{X}(z) \triangleq \sum_{m=-\infty}^{\infty} \underline{x}[m] z^{-m}$

$$[H(z)]_{ij} \triangleq \sum_{m=-\infty}^{\infty} [H[m]]_{ij} z^{-m}$$

$$\Updownarrow$$

$$\underline{Y}(e^{j2\pi f}) = H(e^{j2\pi f}) \underline{X}(e^{j2\pi f})$$

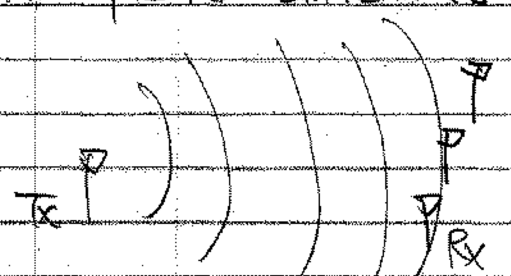
○ Memoryless MIMO channels

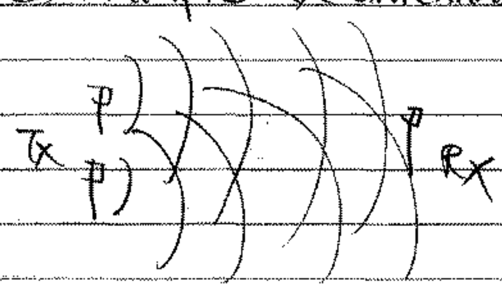
$$\underline{y}[n] = H \underline{x}[n]$$

$$\left(H[m] = H\delta[m] \Rightarrow \underline{y}[n] = \sum_{m=-\infty}^{\infty} H[m] \underline{x}[n-m] = H \underline{x}[n] \right)$$

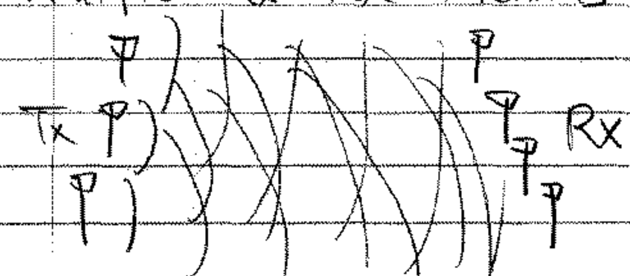
○ Examples

(i) Multiple Rx antennas (SIMO) (ii) Multiple Tx antennas (MISO)

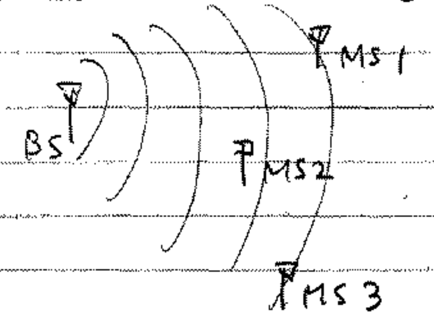




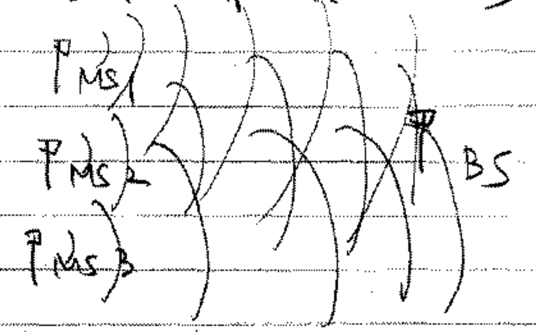
(iii) Multiple Tx-Rx antennas (MIMO)



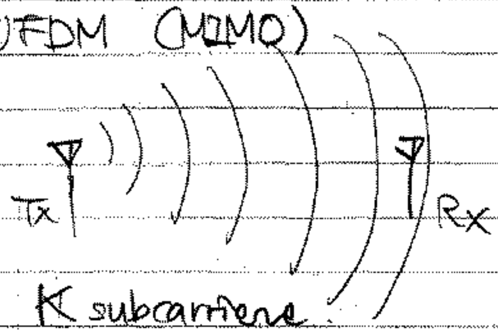
(iv) Cellular downlink (SIMO)



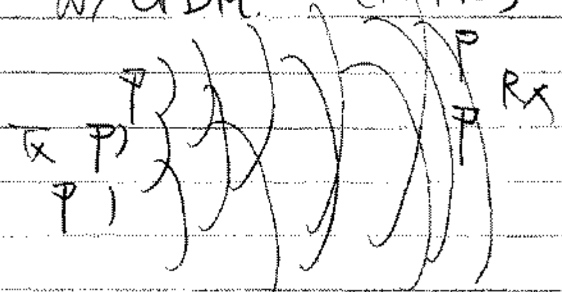
(v) Cellular uplink (MISO)



(vi) OFDM (MIMO)



(vii) Multiple Tx-Rx antennas w/ OFDM (MIMO)



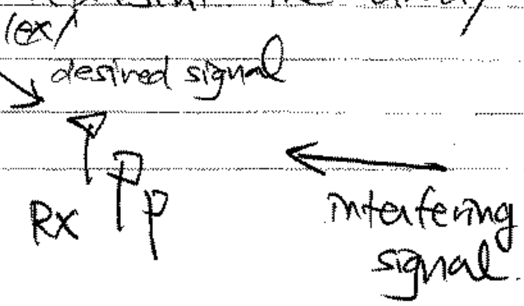
○ Interference rejection & adaptive algorithm

∘ Consider a multiple Rx antenna system that has the received signal modeled as

$$\underline{y}[n] = d[n]\,\underline{p} + e[n]\,\underline{g} + \underline{N}[n]$$

where $\{d[n]\}_{n=-\infty}^{\infty}$ is a sequence of desired data symbols, $\{e[n]\}_{n=-\infty}^{\infty}$ is a sequence of jamming symbols, and $\underline{N}[n]$ is the AWGN.

$\underline{p}$ and $\underline{g}$ represent the array response of each signal. (ex)



- We want to design an LMMSE receiver that minimizes the MSE $E\{|\underline{w}^T \underline{y}[n] - d[n]|^2\}$.
- The optimal solution $\underline{w}_{opt}$ weights the antenna outputs differently so that MSE is minimized.

This type of a problem is a typical problem in multi-Rx antenna system.

Due to the limited time, we briefly discuss how to find $\underline{w}$ adaptively.

Derivation of the LMS algorithm

Overview

As an example that shows how to design an adaptive algorithm, we derive the least mean square (LMS) algorithm that adaptively solves the quadratic minimization problem.

Problem to be solved

- Suppose that the observation is modeled as

$$\underline{y}[n] = d[n]\,\underline{p} + \underline{N}[n] \quad , n \in \mathbb{Z}$$

where $\underline{N}[n]$ is a sequence of uncorrelated random vectors with $E\{\underline{N}[n]\} = \underline{0}$ and $E\{\underline{N}[n]\,\underline{N}[n]^T\} = R\,(>0)$ where R unknown

& $d[n]$ is an uncorrelated zero-mean, unit variance data sequence, &

$\underline{p}$ is an unknown non-random vector

- The problem to be solved is to find the vector $\underline{w}$ that minimizes $E\{(\underbrace{\underline{w}^T\underline{y}[n]}_{\hat{d}[n]} - d[n])^2\}$

$\hat{d}[n]$: linear estimate of $d[n]$

When $\underline{p}$, R are known.

- If $\underline{p}$ and R are known, $\underline{w}$ can be easily solved by using the orthogonality principle

$$E\{(\underline{w}^T\underline{y}[n] - d[n])^T\,\underline{y}[n]\} = 0$$

$\Leftrightarrow$

[redacted]

$$(\underline{p}\underline{p}^T + R)\underline{w} = \underline{p}$$

$$\Rightarrow \underline{w}_{opt} = (\underline{p}\underline{p}^T + R)^{-1}\underline{p}$$

- $\underline{w}_{opt}$ can also be found not by inverting the matrix but by iterating a simple step:

$$\underline{w}[m+1] = \underline{w}[m] - \underbrace{\mu}_{\text{a step size}} \nabla f \qquad m = 1, 2, \cdots, \infty.$$

where f is the objective function given by

$$f(\underline{w}) = \underline{w}^T(\underline{p}\underline{p}^T + R)\underline{w} - 2\underline{w}^T\underline{p} + 1$$

$$\nabla f(\underline{w}) = 2(\underline{p}\underline{p}^T + R)\underline{w} - 2\underline{p}$$

- This can be rewritten as

$$\underline{w}[m+1] = \underline{w}[m] - \mu\left(E\{\underline{y}[n]\underline{y}[n]^T\}\underline{w}[m] - E\{d[n]\underline{y}[n]\}\right) \quad \text{---}(*)$$

○ When $\underline{p}$, R are unknown, but $d[n]$ is known

- By using the method of moment estimation, we estimate

$$E\{\underline{y}[n]\underline{y}[n]^T\} \simeq \underline{y}[m]\underline{y}[m]^T$$

and

$$E\{d[n]\underline{y}[n]\} \simeq d[m]\underline{y}[m]$$

- Thus, (*) reduces to

$$\underline{w}[m+1] = \underline{w}[m] - \mu\left(\underline{y}[m]\underline{y}[m]^T\underline{w}[m] - d[m]\underline{y}[m]\right)$$

$$= \underbrace{\underline{w}[m]}_{\text{current estimate}} \underbrace{- \mu\,(\overbrace{\underline{y}[m]^T\underline{w}[m] - d[m]}^{\text{estimation error}})\,\underline{y}[m]}_{\text{correction term}}$$

or

$$\underline{w}[m+1] = \left(I - \mu\,\underline{y}[m]\underline{y}[m]^T\right)\underline{w}[m] + \mu\, d[m]\,\underline{y}[m]$$

- "$d[m]$ is known" assumption can be solved by transmitting a training sequence.

○ Comments

As seen above,
- first, we find a deterministic algorithm that iteratively find the optimal solution.
- second, we replace $E\{\cdot\}$ by method-of-moments estimates, especially by very simple estimates.
- finally, we analyze the performance (somehow) to see whether the designed adaptive algorithm performs acceptably.