Homework #7

Written: Not Due
Computer assignment due: Jan. 16, 2020

Optional Problems: #5.1, 5.11[†], 5.17[†], 5.20

[†]: Do not need to sketch.

Matlab Problems:

Scalar LMMSE estimator and its application to serial processing

Suppose you want to evaluate performance of two "equalizers" (this is not an typical equalizer in the typical sense because of the structure of the channel matrix, so I am going to call this an estimator as they attempt to estimate the data that has been transmitted), the zero-forcing (ZF) and Linear MMSE (LMMSE) estimator. We have already discussed the former. The LMMSE is a special case of the MMSE estimator in which we **assume** the form the estimator to be linear. To elaborate, suppose you want to estimate the transmitted signal $x[n]$ by collecting N samples of the receive signal $\mathbf{y}[n] \triangleq \begin{bmatrix} y[n] \\ y[n-1] \\ \vdots \\ y[n-N+1] \end{bmatrix}$. To minimize the MSE between the estimate of $x[n]$, denoted as $\widehat{x}[n]$, we want to solve

$$\min_{\mathbf{w}} \; E_{y[n],x[n]}\left[|x[n]-\widehat{x}[n]|^2\right],$$

where $\mathbf{w} \triangleq \begin{bmatrix} w[0] \\ w[1] \\ \vdots \\ w[N-1] \end{bmatrix}$ contains the coefficients of the LMMSE estimator and the expectation is taken with respect to both $y[n]$ and $x[n]$ (Recall a probabilistic estimation approach assumes **both** the observed data and the data one is trying to estimate to be random). The reason why this is a linear estimator is because we desire to obtain $\widehat{x}[n]$ by a linear equation (with respect to $\mathbf{w}$)

$$\begin{aligned}
\widehat{x}[n] &= \mathbf{w}^H \mathbf{y}[n] \\
&= \sum_{k=0}^{N-1} w^*[k] y[n-k] \\
&= \sum_{k=0}^{N-1} w^*[k] \sum_p h[p-k] x[n-p-k].
\end{aligned}$$

This is depicted in Fig. 1 where the LMMSE estimator can be called an LMMSE equalizer to remove ISI.

To obtain $\mathbf{w}$, we expand the objective, differentiate the result with respect to $\mathbf{w}$ and set it to zero to solve for $\mathbf{w}$. Expanding the objective,

$$\begin{aligned}
E_{y[n],x[n]}\left[|x[n]-\widehat{x}[n]|^2\right] &= E\left[|x[n]-\mathbf{w}^H\mathbf{y}[n]|^2\right] \\
&= E\left[\left(x[n]-\mathbf{w}^H\mathbf{y}[n]\right)\left(x^*[n]-\mathbf{y}[n]^H\mathbf{w}^H\right)\right] \\
&= r_{xx}(0) - \mathbf{r}_{xy}^H\mathbf{w} - \mathbf{w}^H\mathbf{r}_{yx} + \mathbf{w}^H\mathbf{R}_{yy}\mathbf{w},
\end{aligned}$$

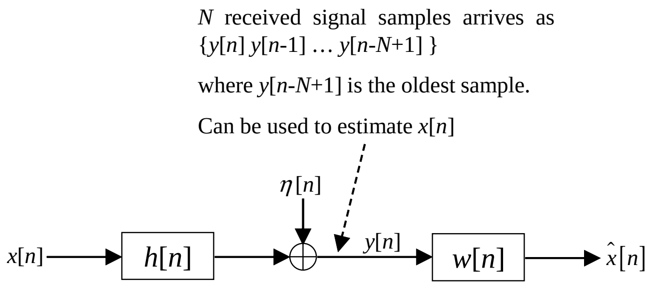

Fig. 1. System block diagram for a single-input single-output system.

where $\mathbf{r}_{xy}^H \triangleq E\left[x[n]\mathbf{y}[n]^H\right]$, $\mathbf{r}_{yx} \triangleq E\left[\mathbf{y}[n]x^*[n]\right] \in \mathbb{C}^N$, and $\mathbf{R}_{yy} = E[\mathbf{y}[n]\mathbf{y}[n]^H] \in \mathbb{C}^{N\times N}$. Differentiating with respect to $\mathbf{w}$

$$\frac{\partial}{\partial \mathbf{w}} E\left[|x[n] - \mathbf{w}^H\mathbf{y}[n]|^2\right] = -2\mathbf{r}_{yx} + 2\mathbf{R}_{yy}\mathbf{w} = \mathbf{0}_{N\times 1}$$
$$\Longrightarrow \mathbf{w} = \mathbf{R}_{yy}^{-1}\mathbf{r}_{yx}.$$

Therefore,

$$\widehat{x}[n] = \mathbf{w}^H\mathbf{y}[n] = \mathbf{r}_{yx}^H\mathbf{R}_{yy}^{-1}\mathbf{y}[n],$$

because $\mathbf{R}_{yy}$ is Hermitian symmetric.

<u>Vector LMMSE estimator and its application to parallel processing</u>

The solution can be extended to estimate a set of transmitted samples $\mathbf{x}[n] \triangleq \begin{bmatrix} x[n] \\ x[n-1] \\ \vdots \\ x[n-N+1] \end{bmatrix}$ so that the estimated signal vector $\widehat{\mathbf{x}}[n] \triangleq \begin{bmatrix} \widehat{x}[n] \\ \widehat{x}[n-1] \\ \vdots \\ \widehat{x}[n-N+1] \end{bmatrix}$ can be obtained as

$$\begin{aligned} \widehat{\mathbf{x}}[n] &= \mathbf{W}^H\mathbf{y}[n] \\ &= \mathbf{R}_{yx}^H\mathbf{R}_{yy}^{-1}\mathbf{y}[n], \end{aligned}$$

where $\mathbf{R}_{yx} \triangleq E\left[\mathbf{y}[n]\mathbf{x}[n]^H\right] \in \mathbb{C}^{N\times N}$.

The above LMMSE solution can also be applied to estimate transmitted signal distorted by an $N \times N$ multi-input multi-output (MIMO) channel $\mathbf{H} \in \mathbb{C}^{N\times N}$. Assume the elements of $\mathbf{H}$ are complex Gaussian distributed, with zero mean and unit variance, i.e. the $(m,n)^{th}$ element of $\mathbf{H}$ is $\mathbf{H}_{mn} \sim \mathcal{CN}(0,1)$. Also, the transmit signal vector, now defined as $\mathbf{x}[n] \triangleq \begin{bmatrix} x_0[n] \\ x_1[n] \\ \vdots \\ x_{N-1}[n] \end{bmatrix}$, which passes through this channel in a linear fashion, is corrupted by AWGN

$\boldsymbol{\eta}[n] \triangleq \begin{bmatrix} \eta_0[n] \\ \eta_1[n] \\ \vdots \\ \eta_{N-1}[n] \end{bmatrix}$, resulting in a receive signal modeled as $\mathbf{y}[n]$ where

$$\mathbf{y}[n] \triangleq \begin{bmatrix} y_0[n] \\ y_1[n] \\ \vdots \\ y_{N-1}[n] \end{bmatrix} = \mathbf{H}\mathbf{x}[n] + \boldsymbol{\eta}[n] \tag{1}$$

and is depicted in Fig. 2.

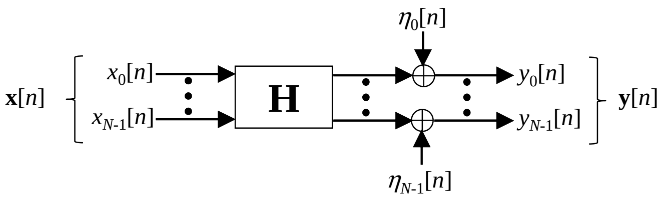

Fig. 2. System block diagram for a MIMO system with N input and N output.

Implementation

In such a MIMO system, at every sampling instant n at the receiver, you can simultaneously obtain N number of received signal samples $y_0[n], y_1[n], \ldots, y_{N-1}[n]$, or $\mathbf{y}[n]$. You can then use either your ZF or LMMSE estimator to obtain $\widehat{\mathbf{x}}[n] \triangleq \begin{bmatrix} \widehat{x}_0[n] \\ \widehat{x}_1[n] \\ \vdots \\ \widehat{x}_{N-1}[n] \end{bmatrix}$. As a nascent communications engineer, you are tasked to implement and evaluate the entire MIMO system as described by (1) using the following specifications.

- Transmit symbol: 4-QAM modulated, statistically independent and uniform distributed, i.e. $1+j$, $-1+j$, $-1-j$, and $1-j$, with power equals to σ_{xx}^2
- Additive noise: statistically uncorrelated, complex Gaussian distributed with zero mean and variance $\sigma_{\eta\eta}^2$, which shall be determined from the **receive** SNR
- The 'S' in SNR is actually referring to symbol, i.e. the QAM symbol. Each symbol represents two bits because of the use of 4-QAM
- Channel elements $\mathbf{H}_{mn}$ are complex Gaussian distributed with zero mean and unit variance
- Assumptions:
  - Transmit symbol and noise samples are statistically independent
  - Channel elements (in the design process) are assumed deterministic and are known at the receiver
  - Assume the received signal $\mathbf{y}[n]$ and transmitted signal $\mathbf{x}[n]$ are zero-mean (otherwise the LMMSE equation will need to be slightly modified to avoid problem with bias)
  - All signals are assumed ergodic
- Demodulation: perfect and coherent (hence, no error from the demodulation process)
- Receiver: ZF and LMMSE estimator
- Receive SNR range: 0 to 18 dB, with 3 dB increment
- Performance metric: BER
- N = 2 and 4

From (1) and the assumptions above, we can see that

$$\begin{aligned}\mathbf{R}_{yy} &= E\left[(\mathbf{H}\mathbf{x}[n] + \boldsymbol{\eta}[n])\left(\mathbf{x}^H[n]\mathbf{H}^H + \boldsymbol{\eta}^H[n]\right)\right] \\ &= \mathbf{H}\mathbf{R}_{xx}\mathbf{H}^H + \mathbf{R}_{\eta\eta} \\ \mathbf{R}_{yx} &= E\left[(\mathbf{H}\mathbf{x}[n] + \boldsymbol{\eta}[n])\,\mathbf{x}^H[n]\right] \\ &= \mathbf{H}\mathbf{R}_{xx},\end{aligned}$$

where $\mathbf{R}_{xx} \triangleq E\left[\mathbf{x}[n]\mathbf{x}^H[n]\right] = \sigma_{xx}^2\mathbf{I}_N$, $\mathbf{R}_{\eta\eta} = E\left[\boldsymbol{\eta}[n]\boldsymbol{\eta}^H[n]\right] = \sigma_{\eta\eta}^2\mathbf{I}_N$. Using these correlation matrices, the transmit symbol can be estimated using the LMMSE estimator as

$$\begin{aligned}\widehat{\mathbf{x}}[n] &= \mathbf{R}_{yx}^H\mathbf{R}_{yy}^{-1}\mathbf{y}[n] \\ &= \mathbf{R}_{xx}^H\mathbf{H}^H\left(\mathbf{H}\mathbf{R}_{xx}\mathbf{H}^H + \mathbf{R}_{\eta\eta}\right)^{-1}\mathbf{y}[n] \\ &= \sigma_{xx}^2\mathbf{H}^H\left(\sigma_{xx}^2\mathbf{H}\mathbf{H}^H + \sigma_{\eta\eta}^2\mathbf{I}_N\right)^{-1}\mathbf{y}[n]\end{aligned}$$

Evaluation

To evaluate the performance, implement a Monte Carlo simulation in MATLAB that will allow you to compare the bit error rate (BER) of using the ZF and LMMSE receiver. You are advised to collect at least 10^6 samples in error for each SNR value. That is, for each SNR value, do not stop your Monte Carlo loop until you have collected at least 10^6 samples that are incorrectly estimated (after demodulation). For evaluation, plot

- BER vs. receive SNR curve using the above SNR range and increment using ZF and LMMSE estimators, for $N = 2$ and 4 (single plot)
- Scatter plots for $N = 2$ and 4 for ZF and LMMSE estimator for SNR = 0, 9, and 18 dB (12 plots)

Please comment on the

- BER performance for different values of SNR
  - What happens to the BER performance of ZF and LMMSE as SNR$\rightarrow \infty$? Show mathematically why this is the case?
- BER performance for different values of N
  - What happens to the BER performance as N increases?
  - Why do you see this phenomenon? What is the cause?
    * What happens to the BER performance when N becomes large?
    * In actual deployment, what would you need to assume to guarantee such a performance?

You need to submit your code and documentation.

Hints on simulation:

- The ensemble averaging required for the implementation of the LMMSE receiver needs to be **approximated** by time-averaging. **If you like to get extra credit** instead of using the assumptions for $\mathbf{R}_{xx}$ and $\mathbf{R}_{\eta\eta}$ as stated above, the MATLAB functions `xcorr` and `toeplitz` should come in handy in computing the autocorrelation matrices.
- It is necessary to compute the **receive**, not the transmit, SNR correctly. Make sure you perform the correct "measurement" before computing the proper SNR.
- Even though the design of the LMMSE assumes the channel to be deterministic, in the simulation, it is required that the elements of $\mathbf{H}$ be randomly generated as specified above. The randomization needs to be done for every Monte Carlo loop

- You can assume the channel is held static for $N \times N_{symbol}$, where N_{symbol} denotes the total number of (vector) symbols when $\mathbf{H}$ is static
- After N_{symbol} symbols are collected, you need to randomly generate another $\mathbf{H}$ again until you have reach the error requirement specified above