S7001E	Lab 3 - MATLAB Assignments

- For this lab, you may work in groups of maximum two students. Each group should hand in a concise report documenting and explaining the results for each of the exercises. Handwritten computations can be included.
- **Please find the lab deadline in Canvas.**
- The MATLAB code should be included as appendices to the report.
- Some useful MATLAB commands: `filter`, `freqz`, `fft`, `inv`, `stem`, `toeplitz`.

Prediction

A moving average (MA) process is described by

$$X[n] = \sum_{k=0}^{L} d_k W[n-k].$$

Such a process is denoted MA(L) and is said to be of order L, (see section 8.5 in Stark and Woods 4th Ed., or section 6.5 in 3rd Ed., under "ARMA-Models"). The input $W[n]$ is an independent random sequence.

1. Find theoretically the best linear MMSE (LMMSE) predictor for an MA(3) process

$$x_n = d_0 w_n + d_1 w_{n-1} + d_2 w_{n-2} + d_3 w_{n-3}$$

   where the coefficients are

$$[d_0, d_1, d_2, d_3] = [2, 1, 2, 1]$$

   and the input $W[n]$ to the MA(3) filter is white Gaussian noise with mean zero and variance 3, that is, $W_n \sim \mathcal{N}(0, 3)$. The predictor should use the three latest values of the process to predict the next value, that is,

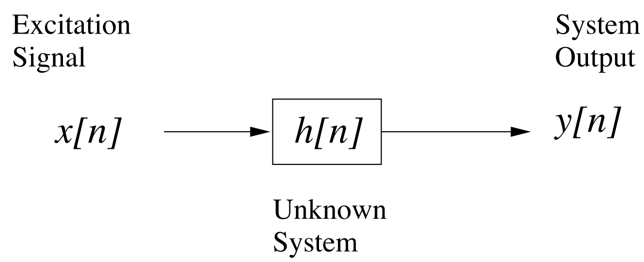
$$\hat{X}[n+1] = a_0 X[n] + a_1 X[n-1] + a_2 X[n-2].$$

2. Generate 1000 normally distributed random numbers and use them to generate the MA(3) process from the first sub-assignment. Use the theoretically determined predictor and plot the sequence $\{x_n\}$ together with the output $\{\hat{x}_n\}$ from the predictor. Plot also the difference between the sequences, that is, the error sequence. Comment on the result. Is there any delay or start up phenomenon?

3. *This assignment is not compulsory for Lab 3 (do it if you like), but still interesting:* Make a predictor of higher order, for example using the latest 4 or 5 samples, for the same process. Does it seem to be better? Calculate the variance of the error for the different predictors both theoretically and for the specific realizations. Compare and comment on the results.

System Identification

Consider the block diagram below, in which an unknown system with impulse response $h[n]$ is excited by a known input sequence $x[n]$, and whose measured output is denoted as $y[n]$.



We assume we have (noise free) measurements of $x[n]$ and $y[n]$ and now explore a simple method to identify the unknown system ("system identification") from its (known) input/output signal pairs.

Note: The unknown system, $h[n]$, may in principle represent any type of (physical) system such as a radio or wireline communication channel, airplanes or other objects in the air identified with Radar, underwater objects (submarines, fish, sea-floor, etc.) identified with Sonar, underground identification of oil and gas, identification of hidden cracks in an object, etc.

1. Download the file *input_output.mat* from Canvas, which represent the unknown system's input and output sequences, $x[n]$ and $y[n]$, respectively.

2. From the data file estimate and plot the auto-correlation functions $\widehat{R}_{XX}[k]$, $\widehat{R}_{YY}[k]$, and cross-correlation function $\widehat{R}_{XY}[k]$, for a reasonable number of lags (about 50).

3. Estimate and plot the power spectral densities of the input and output from your estimated correlations. You can use the MATLAB command `fft` to calculate the discrete Fourier transform of a sequence. What do the power spectral densities suggest to you about the system?

4. Assume the unknown system is a linear time (or shift) invariant system of the form

$$y[n] = \sum_{k=0}^{M} h[k]x[n-k].$$

   Next you'll use your estimated correlations, to compute an estimated system parametrization $\widehat{h}[n]$. Our estimated system's output is $\widehat{y}[n]$ computed as

$$\widehat{y}[n] = \sum_{k=0}^{M} \widehat{h}[k]x[n-k].$$

Our method follows the LMMSE criterion. We'll make use of the orthogonality principle and our assumed LTI system and obtain, (see notes in Canvas for the Wiener filter)

$$E\Big\{ \left(y[n] - \sum_{k=0}^{M} \hat{h}[k]x[n-k] \right) x[m] \Big\} = 0 \qquad \text{for } m = n, n-1, \ldots, n-M$$

Finally, converting this to vector-matrix format yields,

$$\begin{bmatrix} E\{y[n]x[n]\} \\ E\{y[n]x[n-1]\} \\ \vdots \\ E\{y[n]x[n-M]\} \end{bmatrix} = \begin{bmatrix} E\{x[n]x[n]\} & E\{x[n-1]x[n]\} & \ldots & E\{x[n-M]x[n]\} \\ E\{x[n]x[n-1]\} & E\{x[n-1]x[n-1]\} & \ldots & E\{x[n-M]x[n-1]\} \\ \vdots & \vdots & \ddots & \vdots \\ E\{x[n]x[n-M]\} & E\{x[n-1]x[n-M]\} & \ldots & E\{x[n-M]x[n-M]\} \end{bmatrix} \begin{bmatrix} h[0] \\ h[1] \\ h[2] \\ \vdots \\ h[M] \end{bmatrix}$$

Or, in more compact notation,

$$\begin{bmatrix} R_{YX}[0] \\ R_{YX}[1] \\ R_{YX}[2] \\ \vdots \\ R_{YX}[M] \end{bmatrix} = \begin{bmatrix} R_{XX}[0] & R_{XX}[1] & R_{XX}[2] & \ldots & R_{XX}[M] \\ R_{XX}[1] & R_{XX}[0] & R_{XX}[1] & \ldots & R_{XX}[M-1] \\ R_{XX}[2] & R_{XX}[1] & R_{XX}[0] & \ldots & R_{XX}[M-2] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R_{XX}[M] & R_{XX}[M-1] & R_{XX}[M-2] & \ldots & R_{XX}[0] \end{bmatrix} \begin{bmatrix} h[0] \\ h[1] \\ h[2] \\ \vdots \\ h[M] \end{bmatrix}$$

Substituting our previous estimates of the cross-correlation and auto-correlation, we can solve for our estimated system impulse response as

$$\widehat{H} = \begin{bmatrix} \widehat{h}[0] \\ \widehat{h}[1] \\ \vdots \\ \widehat{h}[M] \end{bmatrix} = \widehat{R}^{-1}\widehat{P}$$

where $\widehat{R}$ represents the $(M+1)\times(M+1)$ auto-correlation matrix estimate, and $\widehat{P}$ represents the $(M+1)\times 1$ cross-correlation vector estimate.

Calculate $\widehat{H}$ for values of $M = 2, \ldots, 20$. For each estimate, calculate the mean-square estimate error (that is $E\{y - \widehat{y}\}$) and plot the MSE vs. M. Explain what's happening.

5. The above gives us a method to estimate our unknown system's impulse response for a given assumed system order (that is, M). There is an issue remaining, namely, "What order model should we adopt for $\widehat{H}$?" This is known as the "model-order" estimation problem, and has a variety of approaches. One simple method is to select the model order as where the MSE vs. M curve starts to flatten out. Based on this criterion, what is your final model parametrization?