ECE 463 Lab 6: Channel Estimation & Correction

1. Introduction

In this lab, we will consider a narrowband channel (approximated by a single tap coefficient) and estimate the channel using the training sequence. There are several different approaches to implement an estimator such as maximum likelihood, minimum mean squared error, and linear least squares. We will implement linear least square estimator that is the simplest and also maximum likelihood estimator in AWGN. Once the channel is estimated, we will inverse the channel coefficient to correct the channel effects.

1.1. Contents

1.2. Report

Submit the answers, figures and the discussions on all the questions. The report is due as a hard copy at the beginning of the next lab.

2. Channel Estimation

In this section, you will implement the following subVIs.

- LLS.gvi: Performs linear least square estimation
- Channel_Estimation.gvi: Estimates the channel response
- sim_channel.gvi: Simulates ISI channel

2.1. Linear Least Square Estimation

Consider a linear system

$$\mathbf{Ax} = \mathbf{y}$$

where A is the known matrix, x is an unknown vector and y is the observation vector. Assuming A is the full-rank matrix, the least-square error solution is

$$\hat{\mathbf{x}} = (\mathbf{A^H A})^{-1}\mathbf{A^H y}$$

where A^H is the Hermitian or conjugate transpose. The squared error (or the square of the L2 norm of the error) is defined as

$$SE = ||\mathbf{A}\hat{\mathbf{x}} - \mathbf{y}||_2^2$$

Design "LLS.gvi" that computes the least square error solution and returns the error.

	Terminal name	Type	Description
Input	*y*	Complex Double array (1D)	Observation vector
	A	Complex Double array (2D)	System matrix
Output	*x_hat*	Complex Double array (1D)	Least-square error solution
	Squared error	Double	Squared error of the solution

- Use “Pseudoinverse Matrix” block. This block returns

$$\mathbf{A_{pinv}} = (\mathbf{A^H A})^{-1}\mathbf{A^H}$$

- Use “A x B” block to perform the multiplication of two matrices or a matrix and a vector.
- Use “Vector Norm” block to compute the norm of a vector. The default norm type is L2-norm.
- There are more blocks related to matrix and vector under “Math-Linear Algebra”.

2.2. Channel Estimation

The channel can be modeled as a system of linear equations as follows

$$y[n] = \sum_{l=0}^{L-1} h[l]t[n-l] + w[n]$$

where y[n] is the received symbols, h[n] is the channel coefficients, t[n] is the N_t-length training sequence for n= 0,1,…,N_t-1, L is the channel length, and w[n] is AWGN. The summation starts from n=L-1 to ensure the training sequence starts with t[0]. Rewrite the above equation in matrix form as

$$\begin{bmatrix} y[L-1] \\ y[L] \\ \vdots \\ y[N_t-1] \end{bmatrix} = \begin{bmatrix} t[L-1] & t[L-2] & \cdots & t[0] \\ t[L] & t[L-1] & \cdots & t[1] \\ t[L+1] & \ddots & & \vdots \\ \vdots & & & \vdots \\ t[N_t-1] & \cdots & & t[N_t-L] \end{bmatrix} \begin{bmatrix} h[0] \\ h[1] \\ \vdots \\ h[L-1] \end{bmatrix}$$

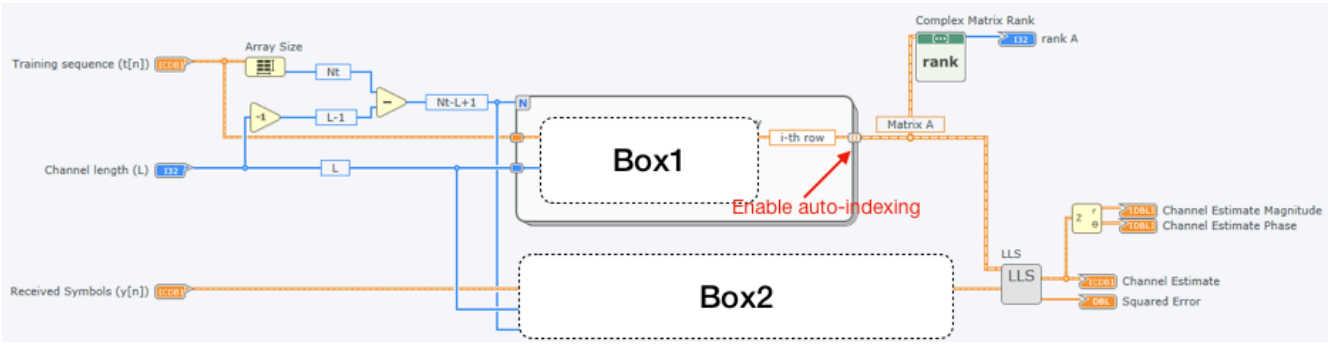
$$\mathbf{y} = \mathbf{A}\mathbf{h}$$

Design “Channel_Estimation.gvi” that estimates the channel coefficients using “LLS.gvi”.

	Terminal name	Type	Description
Input	*Training sequence (t[n])*	Complex Double array (1D)	Training sequence
	Channel length (L)	Integer	
	Received Symbols (y[n])	Complex Double array (1D)	Received symbols at the receiver
Output	*Channel Estimate (h[n])*	Complex Double array (1D)	Estimated channel coefficients
	Channel Estimate Magnitude	Double array (1D)	
	Channel Estimate Phase	Double array (1D)	
	Rank A	Double	Rank of A
	Squared error	Double	Squared error of the channel estimation

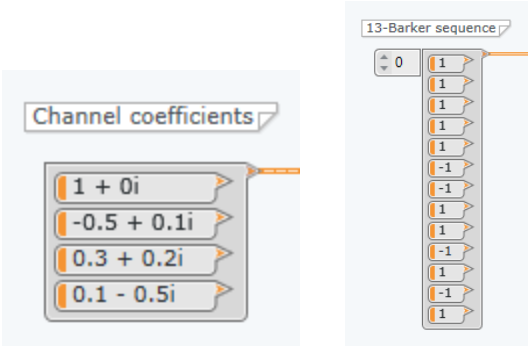
- Complete **Box1** in the following figure to construct the matrix **A** defined above. Note that the size of the matrix **A** is (Nt-L+1)-by-(L). For each iteration of the for loop, an i-th row will be constructed. Make sure “auto-indexing” is enabled to create 2D matrix from the i-th row. (Hint: Use “Array subset” and “Reverse 1D array” blocks.)

- Complete **Box2** in the following figure to construct the vector **y** defined above. Note that the length of the vector **y** is (Nt-L+1) and starts with index L-1. (Hint: Use “Array subset” block.)



2.3. Simulation (Multi-tap channel)

Build "sim_channel.gvi" to simulate the channel estimation. Use the following channel coefficients and convolve with 13-Barker sequence. The result of the convolution will simulate the received symbols. Verify your "Channel_Estimation.gvi" can reconstruct the original channel coefficients.



2.4.Questions

2.4.1. Run the simulation with the right channel length (4 in this case) and report the estimated channel coefficients. Does the estimation match with the original channel?

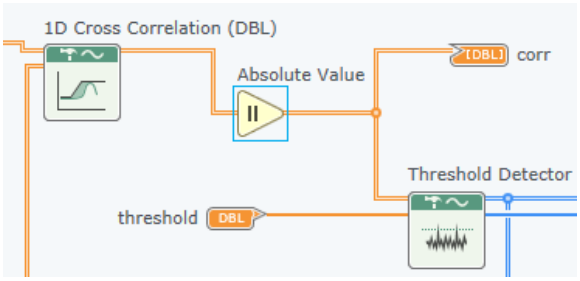
2.4.2. What happen if the channel length over-estimated or under-estimated? Use the same channel coefficients but change the channel length parameter to be greater than 4 and less than 4.

2.4.3. Try a different training sequence such as +1-1+1-1+1-1+1-1+1-1+1-1+1-1+1. What result do you have? Explain why. (Hint: check the rank of the matrix A)

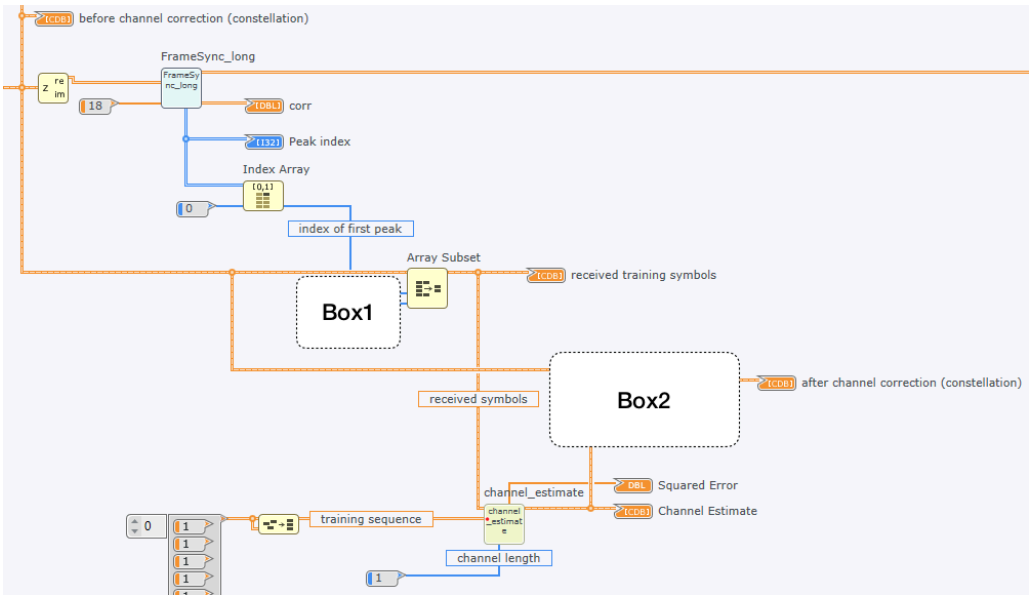
3. Channel Correction (Single-tap channel)

Load BPSK receiver created in the previous lab and plot the constellation. We know the constellation of BPSK is rotated by a random amount of phase due to the channel. We will run narrow band channel estimation (channel length L = 1) and correct the channel to compensate the phase rotation.

- Modify your frame sync block by adding the absolute value block in the following figure. By taking the absolute value of the correlation, the inverted training sequence can be also detected. This will prevent failing the frame sync in BPSK due to the channel phase.



- Complete **Box1** to extract the received training symbols. Use the index of the first correlation peak from your frame sync block. Once you have the received symbols, perform the channel estimation. Make sure **the channel length is 1**. Use the same training sequence used for frame sync. Complete **Box2** to correct the channel.



3.1. Questions

3.1.1. Compare the BPSK constellation before/after channel correction. What change do you see?

4. Signal-to-Noise Ratio (SNR)

The average power of signal x[n] can be computed as

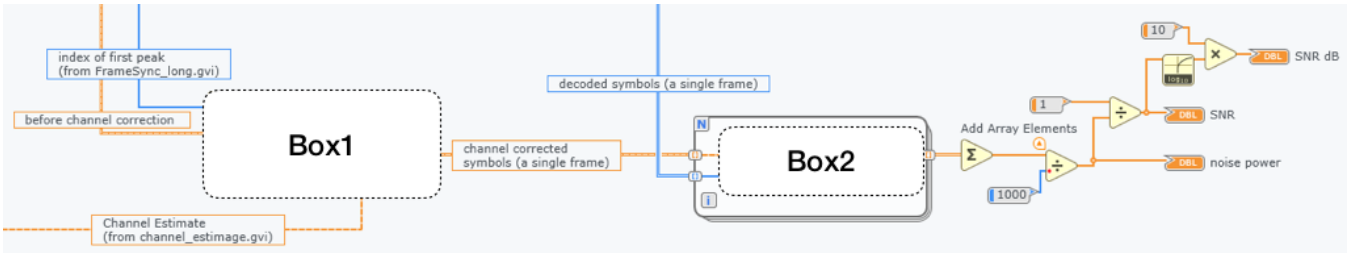
$$\sigma_x^2 = \frac{1}{N}\sum_{n=0}^{N-1} |x[n]|^2$$

In most cases, Signal-to-noise ratio is described in a logarithmic scale with dB units;

$$SNR_{dB} = 10\log_{10}(SNR) = 10\log_{10}\frac{\sigma_x^2}{\sigma_w^2}$$

where $\boldsymbol{\sigma_x^2}$ and $\boldsymbol{\sigma_w^2}$ are the average power of the signal and noise, respectively.

- Complete **Box1** to obtain the symbols of **a single frame** (e.g. 1000 symbols) and to correct the channel. Reuse the index of the first correlation peak from your frame sync block and the channel estimates.
- Complete **Box2** to extract the noise from the received symbols. To do this, use the decoded symbols (Reuse the decoded bits from BER computation and convert them to symbols). In BPSK, the decoded symbols will be either +1 and -1.



4.1. Questions

4.1.1. Try a different value of attenuator to reduce the signal power and compare SNR.