Lab 9: Multipath Propagation and Equalization

Wadhwani Electronics Lab

Department of Electrical Engineering
Indian Institute of Technology, Bombay.

Aim of the experiment

1. To understand 8-PSK Modulation and Demodulation scheme
2. To study the effect of multipath propagation using a suitable multipath model.
3. To understand the working of an equalizer when the signal propagates through a multipath environment.

Pre-lab Work

- Study how to determine a transfer function of a discrete-time signal using z-transform and how to find its inverse.
- Make sure that you have read the supporting material uploaded along with this document.
- For additional information about equalizers, refer to Adaptive Filter Theory by Simon Haykin.

Part 1a: Implementing 8-PSK Transmitter

- Implement an 8-PSK transmitter using the
  - "Random Source" block (generating bytes with min=0, max=8)
  - "Chunks to Symbols" block with eight constellation points (figure out the eight points using a similar method as you did for QPSK).
  - Use Root-Raised Cosine (RRC) filter for pulse shaping with sps=8, symbol_rate=10k (same values used in lab 8)

  Note: The flow remains the same as the QPSK transmitter that you have done in lab 8.
- Upconvert this baseband 8-PSK-modulated signal using a carrier signal with 100kHz frequency.
- Add Gaussian noise to this modulated 8-PSK signal and use a slider to vary the noise amplitude in the range of 0 to 1.

Part 1b: Multipath model(channel)

- Make a multipath model (H(z)) with tap coefficients 1 and 0.5 (such that coefficient of 1 corresponds to direct line-of-sight reception and coefficient of 0.5 corresponds to a signal reflected from an object and received after **one symbol period delay**).
- Pass this noisy 8-PSK modulated signal through this multipath model.

Note: The delay parameter with value one in the "Delay" block and z^{-1} in the "IIR Filter" block correspond to one sample delay. So, set the values accordingly to get one symbol delay in either of these blocks.

Part 1c: Equalizer design(receiver)

- Implement a feed-forward 4-tap equalizer with adjustable coefficients
  - Limit the number of tap coefficients to 4 (in addition to the one direct signal to the output with gain coefficient = 1) and neglect tap coefficients that come after the fourth tap. Use sliders to change their values (each slider should have a value between -1 and +1). The tap delays should be equal to the symbol period delay.
- Connect the received signal obtained after performing IQ demodulation as the input of the equalizer and observe the equalizer output constellation as you adjust the coefficients.
- Generate error magnitude from the equalizer output (y) as the absolute value of $e = |y|^2 - R^2$ where R is the radius of the circle where your constellation points lie.
- To observe this error output, send it through a low pass filter implemented by the "IIR Filter" block (with feedforward taps = [0.001], feedback taps = [1, 0.98], "Old Style of Taps"="True"), followed by a Scope Sink.
- Adjust the taps manually to reduce the error. Observe that reducing error improves the constellation. Try to minimize the error by adjusting coefficients.

Equalizer for known multipath model

- Now invert H(z) analytically to find the first four tap coefficients of E(z)=1/H(z)
- Set the slider values corresponding to the obtained coefficients.
- Observed constellation should be very good and the error signal should have a minimum value.

Part 1d: Implementing the receiver

- Implement match filter after down-converting the output of the equalizer to the appropriate sampling rate.
- To extract the symbols from the waveform
  - Use the "skip head" block to remove the initial garbage values added by matched filtering.
  - Then add the "Decimating FIR filter" block to decimate by sps.
  - Observe the final received 8-PSK constellation.

Note: No need to use "Threshold block" for mapping to 8-PSK chunks.

Part 1e: For a real message signal

Repeat the above process for the given text file and try to retrieve the text after equalization and match filtering.

- For this use the text file as input to the "File Source" block and then use "Unpack k bits" followed by "Pack k bits" blocks to get the eight symbols "000", ..., "111". Add this flow as an input to the "Chunks to Symbols" block to map to the 8-PSK symbols.
  Q: What is the value of k while unpacking and packing the bits?
- Perform 8-PSK modulation and then pass it through the multipath model after adding the noise.
- Perform demodulation, equalization, and match filtering with required down-conversion to appropriate sample rate and also decimate by sps.
- Observe the final received 8-PSK constellation.
- Now use the "Constellation Decoder" block for symbol decoding followed by "Unpack k bits" and "Pack k bits" blocks and store the result using the "File Sink" block to a text file.(see whether you are able to retrieve the text)

Part 2: CMA Equalizer for multipath model

For GNURadio version: 3.9

- Use the in–built "CMA Equalizer" block in GNU Radio to observe the desired constellation.
- The CMA equalizer block should be used after "Decimating FIR Filter".
  - The gain parameter of the "CMA Equalizer" block is the step size by which the algorithm keeps updating. Take its starting value as $1e-4$. The maximum value of step size can go to $1e-9$.
  - The modulus parameter of the "CMA Equalizer" block is the radius of the circle on which your constellation points lie.
  - Set the samples per symbol to 1.

..contd

- Observe the output constellation. Since the CMA equalizer is an adaptive algorithm, it takes some time to update, so wait till the eight constellation points appear.
- Observe the received text file (the starting values in the text file will be garbage since the algorithm didn't settle, so the actual received text will be found later at the end page of the text file).

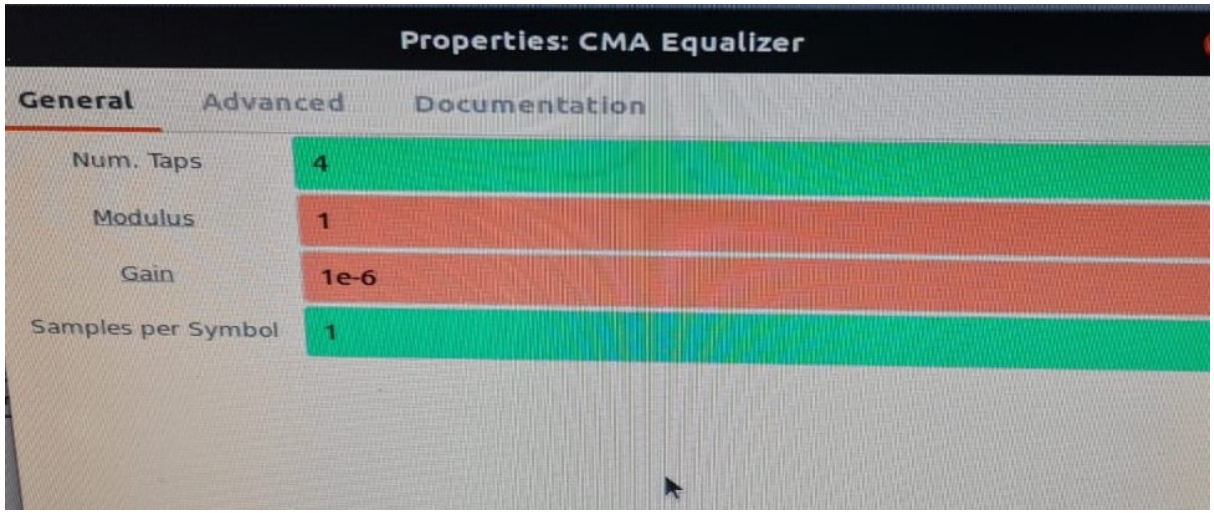

Figure: CMA Equalizer block

...contd

For GNURadio version: 3.10

- Use the in–built "Linear Equalizer" block in GNU Radio to observe the desired constellation.
- The "Linear Equalizer" block should be used after "Decimating FIR Filter".
- Use the blocks as shown in the figure.

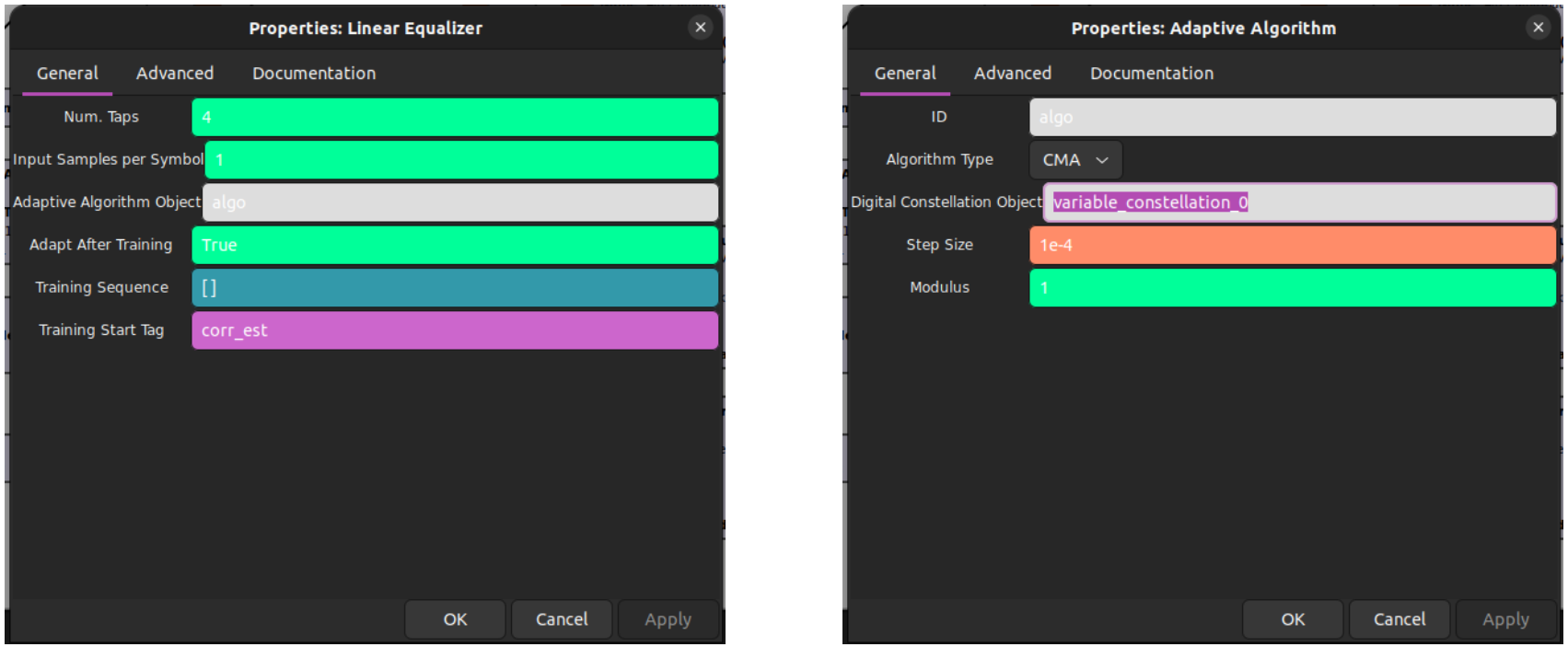

Figure: Linear Equalizer and Adaptive Algorithm Blocks

...contd

- The values in the "Adaptive Algorithm" block are as follows:
  - The value of the Digital Constellation Object is the id name of the Constellation Decoder block used.
  - Set the algorithm type to CMA.
  - Take the step size with a starting value of $1e-4$. The maximum value of step size can go to $1e-9$.
  - The modulus parameter is the radius of the circle on which your constellation points lie.
- The values in the "Linear Equalizer" block are as follows:
  - Set the number of taps to 4 and input samples per symbol as 1.
  - The value of the adaptive algorithm object is the id name of the "Adaptive Algorithm" block.

..contd

- Observe the output constellation. Since the CMA equalizer is an adaptive algorithm, it takes some time to update, so wait till the eight constellation points appear.
- Observe the received text file (the starting values in the text file will be garbage since the algorithm didn't settle, so the actual received text will be found later at the end page of the text file).