Multipath Propagation and Equalization

EE340: Prelab Reading Material for Experiment 8

AUTUMN 2022

1 Multipath

propagation in wireless systems refers to the phenomenon in which the transmitted signal reaches the receiving antenna by taking two or more different paths as shown in Figure 1. It can be caused by various factors such as ionospheric reflection and refraction, reflection from water bodies, terrestrial objects, etc. Since the various paths are of different lengths, they arrive at the receiver with different delays. This can lead to Inter Symbol Interference (ISI). When ISI occurs, a part or, all of a given symbol which is transmitted is spread into the subsequent symbols, thereby resulting in errors at the receiver output.

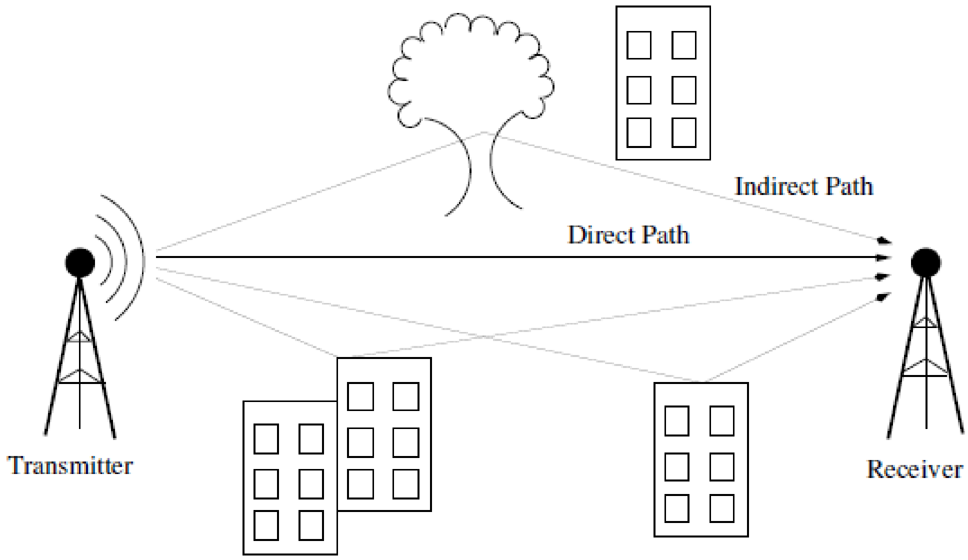

Figure 1: Multipath Propagation in a Wireless System

To model the multipath propagation at the receiver, consider the signal received through the direct path to be an impulse of unit amplitude. All other subsequent signals that arrive at the receiver after suffering reflections and refractions can be modeled as time delayed impulses with amplitude less than unity. Hence the addition of a signal and its time delayed versions obtained at the receiver input results in ISI.

We can use the z-transform to get a system transfer function for this multipath model (as shown in Figure 2, where the coefficients of the transfer function are called as taps.s

2 Equalizer

An equalizer is a signal processing block that is used to reduce the effect of ISI on the transmitted symbols. Hence they are designed in such a way that they have the inverse transfer function of

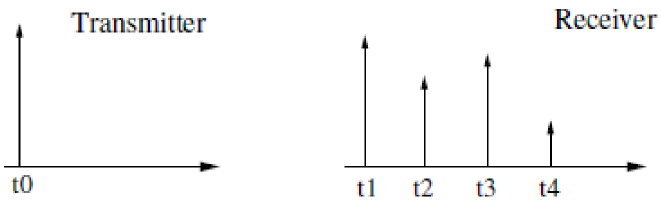

Figure 2: A Multipath Model

the multipath model.

Let the transfer function H(z) of the multipath model be

$$H(z) = 1 + \sum_{i=1}^{n} az^{-i}$$

where ai is the channel tap coefficient for i th tap.
Then, the equalizer transfer function E(z) is,

$$E(z) = \frac{1}{H(z)}$$

$$E(z) = 1 + \sum_{j=1}^{n} b_j z^{-j}$$

where bj represent the equalizer tap coefficient for j th tap and such an equalizer is called feed forward equalizer.

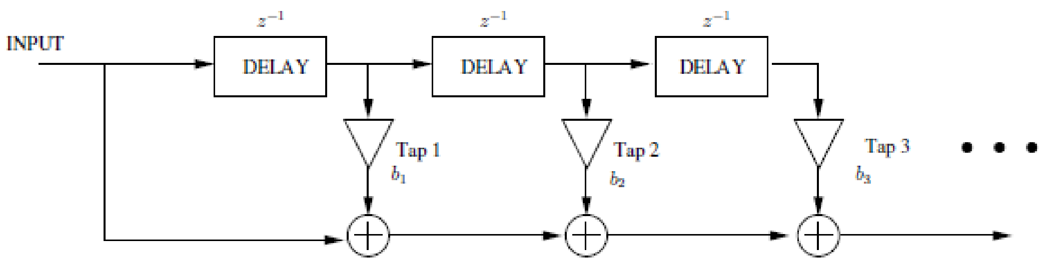

Figure 3: A Feed-Forward Equalizer

The channel tap coefficients (ai s) can vary slowly with time due to changes in atmosphere or due to moving objects. In such cases, the equalizer tap coefficients (bj s) also need to adapt to the time varying channel. In general adaptive equalizer algorithms are used as the channel coefficients are difficult to predict in advance (particularly for wireless channels). Most popular among them are Least Mean Square (LMS) algorithm and Constant Modulus Algorithm (CMA).

2.1 Least Mean Square Algorithm

LMS is a linear adaptive algorithm which minimises the mean square error (i.e. tap coeffiecients bj s are updated/adjusted by minimizing the E[—error— 2]). Let d[n] be the training sequence which is used for adjusting the taps/coefficients before random data/sequence arrives (can you think of the reason why adaptive mechanism is not performed when random data arrive ?), u[n] be the input to the receiver, y[n] be the output of the receiver as shown in Figure 4.
Then, before the random data arrives, the taps/coefficients in LMS algorithm are updated by the equation

$$b_j[n+1] = b_j[n] + u[n]e * [n]$$

where e[n] = d[n] - y[n], µ is the step size of LMS algorithm and determines the convergence rate. Too small a step size will make the algorithm take a lot of iterations while too large step size may diverge the weight taps (can you think of the reason?).

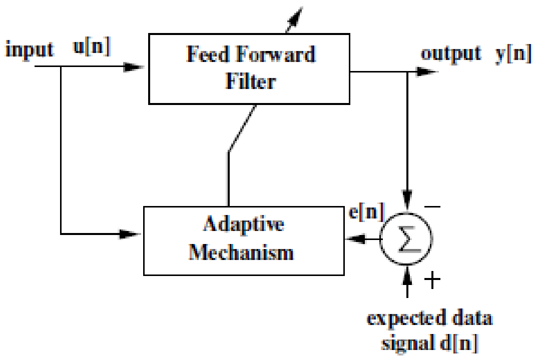

Figure 4: Least Mean Square Algorithm

2.2 Constant Modulus Algorithm (CMA)

The CMA equalizer is an adaptive equalizer that works well when the signal has a constant modulus, i.e., when the signal constellation points lie on a circle (such as QPSK or 8-PSK). Multipath effects distort the received signal and its constellation diagram. The equalizer tries to ensure that the signal samples from the equalizer output lie on a circle. This is done by minimizing the dispersion of the equalizer output y[n] around a circular contour with a predefined radius R (for n *th* symbol), which is termed as cost function and given by equation.

$$J[n] = E[||y[n]|^2 - R^2|^2]$$

where y[n] = yI [n] + jyQ[n] and yI [n], yQ[n] are the in-phase and quadrature phase components of the signal obtained at the equalizer output for n *th* symbol, and R is the radius of the circle. Let u[n] be the signal to be equalized and coefficient of k *th* tap of the adaptive filter taps with L taps be bk[n] , then the equalizer output and error are given by:

$$y[n] = \sum_{j=0}^{L-1} b_j^*[n]u[n-j] = B_n^H U_n$$

$$e[n] = |y[n]|^2 - R^2$$

Then the update equation of the filter coefficients for a CMA equalizer is given by:

$$b_j[n+1] = b_j[n] - \mu e[n]y^*[n]u[n]$$

where µ is the step factor which is to be carefully selected.

Usually, minimizing the difference ensures that the equalizer has compensated for the multipath effects added by the channel, and as a result, output samples lie on the desired constellation. It should be noted that if there are any phase and frequency offsets, the points may appear anywhere on a circle, and hence a carrier frequency and phase synchronization block (such as a costas loop) is required to remove these offsets to obtain the desired constellation plot.

**Some of the project ideas include implementation of an adaptive equalizer (e.g., the equalizer used in lab can be made adaptive) or implementing other equalizers or developing GNU radio blocks to implement an equalizer. More information regarding equalizers can be found in Simon Haykin, Adaptive Filter Theory and GNU Radio wiki page.

Generally, some training sequences (symbol sequences know to the transmitter) are used for initial adjustment of tap coefficients. However, CMA algorithm can be used for "blind-adaptation," which means no training sequences are required (can you think of the reason?).

3 Polyphase Arbitrary Re-Sampler

To create a signal from these symbols that can be visualized as a "waveform", multiple samples are required to represent the symbol. The Polyphase Arbitrary Re-Sampler does this job. The average number of samples representing the waveform for one symbol period is called the "samples-per-symbol" or the sps. For example, if our incoming signal is coming at a symbol rate of 100 kSymbols/s, and we are sampling it at 1 MHz sample rate, the sps = 10. If the sample-rate is changed by up-sampling or downsampling the signal in GNU radio, the sps will change accordingly, but not the symbol rate. The Polyphase Arbitrary Re-Sampler block implements the pulse shaping filter, which is specified using FIR taps in its properties. In this experiment, we have used Root Raised Cosine (RRC) filter defined by tx taps. This block maps the complex amplitudes (assuming ONE sample represents one symbol at the input) and passes each input through the RRC filter with desired samples-per-symbol (sps). Typically you should have an sps of at least two for baseband signals to comfortably satisfy the Nyquist Criterion (especially if the pulse shaping filter is not a brick wall filter). Carrier modulated signals must have much larger sps compared to that of the baseband signal (why?).

4 Costas Loop

This block is used for estimation of residual carrier frequency offset (that could not be removed by the FLL) and carrier phase offset, and their compensation. This block calculates the residual frequency

and phase error in the signal and provides the required compensation so that the transmitted con- stellation is retrieved back correctly at the output. You will implement Costas Loop for frequency

phase offset estimation and correction yourself in Experiment 10. For the Costas Loop block to work in GNU-radio, SPS of the signal at its input should be 1, i.e. the Costas Loop should be applied after the Polyphase Clock Synchronizer (having 'output sps = 1'), which is discussed next. The order of the Costas Loop is the number of symbol levels (for example 4 for QPSK and 8 for 8-PSK).

5 Polyphase Clock Synchronizer for Data Timing Recovery

In addition to the offset in the carrier frequency, an offset in the data (symbol) clock rate ($1/T_s$) and phase also exists. For example, the symbol rate at the transmitter side may be 100 kSymbols/sec (and corresponding data clock be 100 kHz), whereas at the receiver, the sampling data clock may be 100.01 kHz. Also, the sampling clock edges have to be aligned to the centre of symbol periods to sample symbols with best SNR. This operation of clock synchronization and alignment is carried out by the Polyphase Clock Synchronizer block [4]. You will implement clock timing synchronization yourself in hardware in Experiment 8. You can give the complex output samples from this block to the Scope Sink in the X-Y mode (with output sps=1) to observe the received symbol constellation.