Lecture 20: Mitigation Techniques for Multipath Fading Effects

Multipath Fading Mitigation Techniques

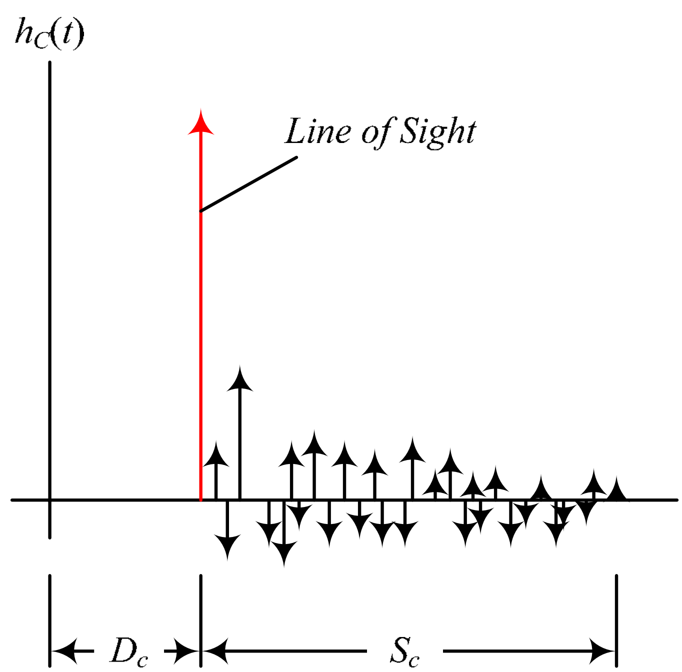
We should consider multipath fading as a fact that we have to live with. There is not much that we can do to reduce it or stop it as long as the received signal is received via multiple paths with significantly varying path lengths, and the mobile phone or the surrounding objects move. However, there are several techniques that we can employ to reduce the effect of multipath fading on the received signal and received bits. These techniques are

1) Equalization
2) Diversity
3) Channel Coding

In this lecture and the next two lectures we discuss each of these techniques in some details

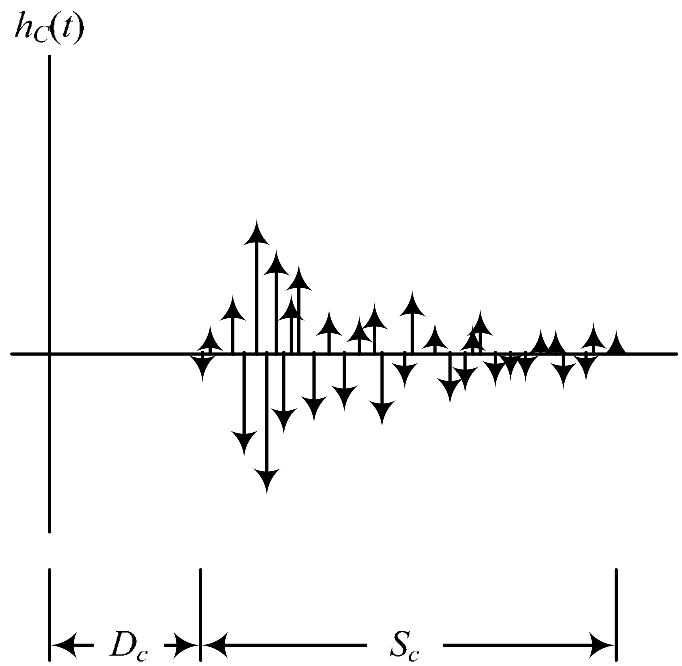
Introduction

One of the effects of multipath reception is that a transmitted impulse (delta function) at a specific time instant is received as multiple (or sometimes an infinite number) of deltas (or impulses) with different delays. In fact, the impulse response of multipath channels is a large number of impulses with positive delays. The following is a typical impulse response of a multipath channel with a line of sight component:



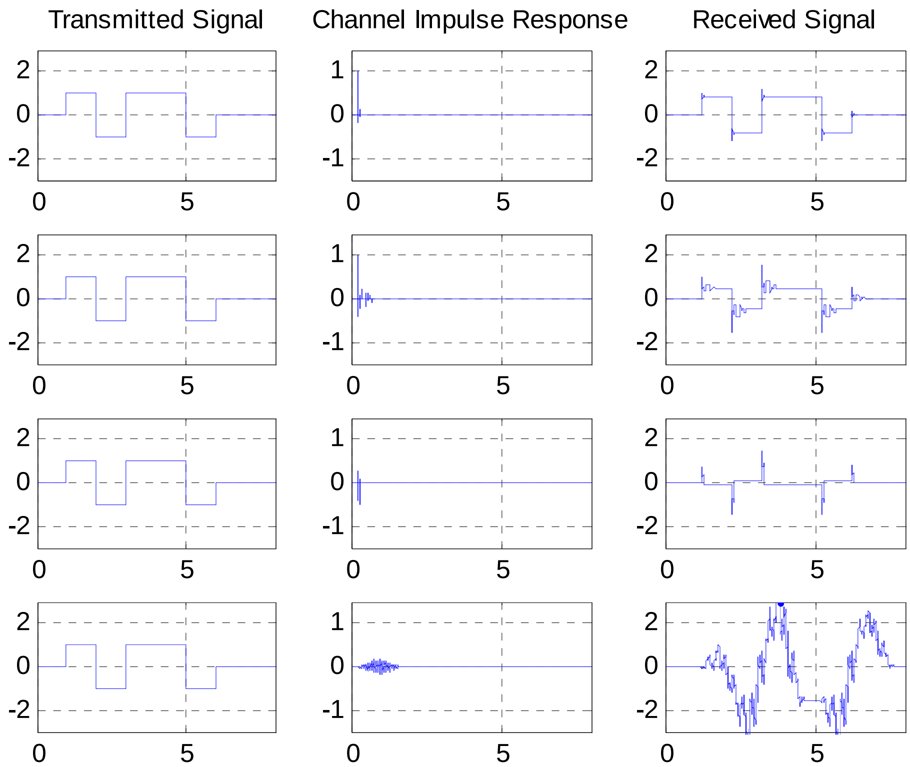
where D_c is the channel delay that represents the time the first electromagnetic component takes to travel from the transmitter to the receiver, and S_c is the spread of the channel representing the time difference between the first and last components to reach the receiver. The above represents the impulse response of a channel with a line of sight component. The line of site component is usually the strongest component of all and because the line of sight path is the shortest, the line of sight component is the first one to arrive. The line of sight channel is experienced when mobile phones are outside and have a direct path to the tower.

A non-line of sight channel has a different impulse response where the strongest component may not be the first one to arrive at the receiver and there may be little difference between the strength of different components. The following is a typical impulse response of non-line of sight channel:



The effect of the spread of the channel is the smearing (spreading) of the transmitted signal at the receiver. This effectively causes the transmitted bits of a digital signal to get mixed together (inter-symbol interference)

For example, consider the following several digital bits that are transmitted over several channels with different impulse responses. The output of each channel is illustrated.



As seen in the first channel, due to the relatively short response of the channel and the high line of sight component, the received signal is close to the transmitted one. The second channel has longer spread with a large line of sight component resulting in slightly more distortion to the bits. The third channel has short spread but no line of sight component and therefore the distortion in the transmitted signal is significant. The last channel does not have a line of sight component and has a relatively long spread (duration of 1.5 bits) resulting in a significant distortion to the transmitted signal.

A problem with mobile channels is that not only they are multipath channels, but because of the movement of the mobile phone or its surroundings, these channels are highly time-varying (the impulse response changes with time). Therefore, even if a receiver was able to identify the impulse response of the channel, this impulse response quickly changes and the receiver needs to continually track the changes in the channel impulse response as it changes with time.

Equalization

What Does an Equalizer Do?

An equalizer is a device that identifies the impulse response of the channel and tries to reverse it by employing a filter that has the inverse impulse response of the channel. A channel with finite impulse response that is used for transmitting digital data with symbol period T_S can simply be thought of as a filter with a particular finite impulse response (the spread of the channel is finite in duration). The impulse response of the channel (or filter) with finite impulse response is usually represented in the Z domain as

$$H_C(z) = a_0 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + \ \ldots \ + a_N z^{-N}$$

where z^{-1} represents a delay unit of duration T_S seconds, and a_i is the ith channel coefficient, and N indicates that the spread of the channel is delay N symbol periods (or NT_S seconds). Such a response is called a finite impulse response because the duration of the response due to an impulse has a finite amount of time. On the other hand, the duration of an infinite impulse response channel is infinite in duration, and therefore an impulse that is sent through an infinite impulse response channel will produce a non-zero output that theoretically continues for ever. Many finite impulse response channel have also finite impulse response components as indicated above.

To reverse the effect of such a channel, the equalizer has to employ a filter with the inverse response of the channel or

$$H_E(z) = \frac{1}{H_C(z)} = \frac{1}{a_0 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + \ \ldots \ + a_N z^{-N}}$$

Now, the inverse of a finite impulse response filter is generally an infinite impulse response filter. However, in many cases a finite impulse response filter can approximate the inverse of the channel. Finite impulse response equalizers are generally more desirable because of the stability problems and difficulty in adjusting infinite impulse response equalizers compared to finite impulse response equalizers. Therefore, usually the impulse response of the equalizer filter is represented as a finite impulse response in the form

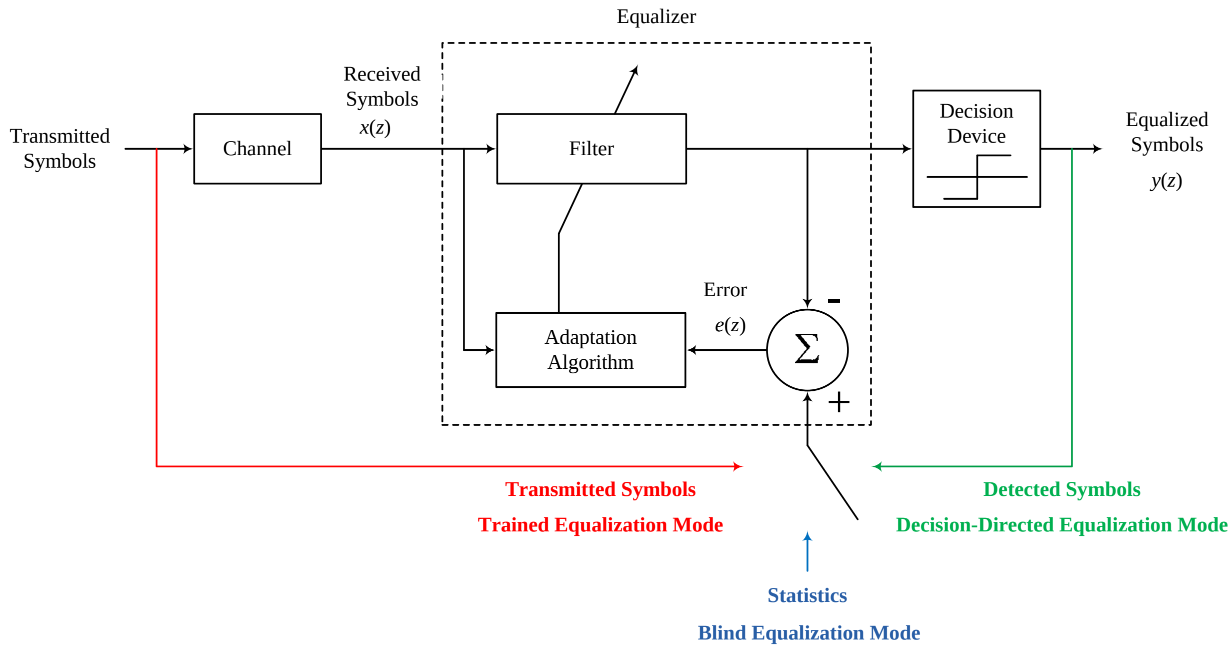
$$H_E(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + \ \ldots \ + b_M z^{-M}$$

where b_i is the ith equalizer filter coefficient, and M indicates that the duration of the equalizer filter is M symbol periods (or MT_S seconds) where M is usually larger than N .

Structure of an Equalizer

An equalizer is simply a filter that tries to determine the impulse response of the channel and attempts to reverse its effect by adapting its filter part to the inverse of the channel impulse response. The structure of a typical equalizer is shown below

Lecture 20: Mitigation Techniques for Multipath Fading Effects



An equalizer contains the following components:

1. **Discrete-Time Filter:** the purpose of this filter is to try to reverse the channel distortion by having an impulse response close to the inverse response of the channel,
2. **Adaptation Algorithm:** this block adapts the equalizer filter (modify its coefficients to make the filter with the closest response to the inverse of the channel,
3. **Subtractor:** that computes an error signal between the actual output of the equalizer and the desired output of the equalizer at a particular time to be used in the process of adapting the equalizer filter,
4. **Decision Device:** that takes the soft results at the output of the equalizer filter (continuous amplitude signals) and converts this data to hard results (discrete amplitude signals).

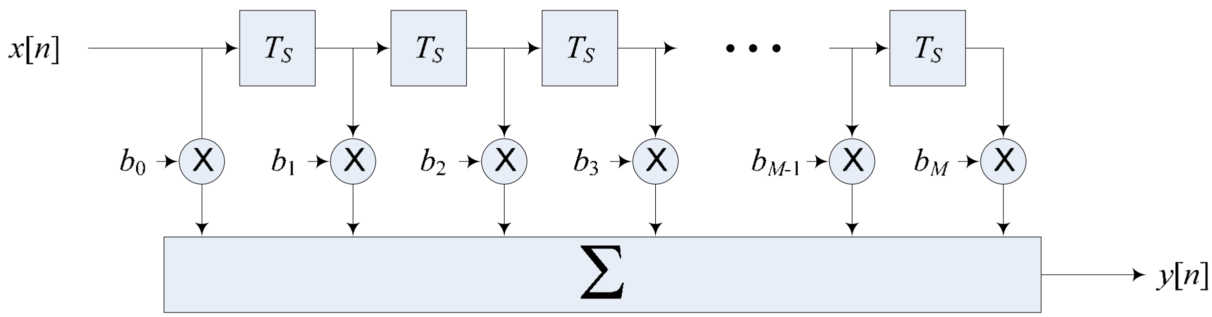
An equalizer has several modes of operation. Namely, these modes are:

1. **Training Mode:** The operation of most equalization processes starts by having the transmitter send to the receiver a known sequence data that is known to both the transmitter and the receiver. The purpose of this training sequence is to give the equalizer at the receiver a way for determining the distortion that the channel brought to this known sequence. By observing the received signal at the output of the channel and comparing this with the sequence of bits that the receiver knows with certainty has been transmitted (by taking the difference between the two sequences), the equalizer can apply an adaptation algorithm that modifies the filter coefficients in small steps until the adaptation of the equalizer filter is achieved such that the difference between the actual output of the equalizer and the training sequence is close to zero. The length of the training sequence must NOT be too short that not enough data is given to the equalizer to adapt its filter and NOT too long that precious time is wasted in transmitting a known training sequence to a fully adapted equalizer instead of transmitting real data. The best training sequence are random in nature.

2. **Blind Mode:** The problem with the training mode of an equalizer is that precious time is wasted in transmitting a training sequence that it self carries no data as this training sequence is fully known to the receiver. In specific applications, an adaptation mode of the equalizer can be performed without depending on a training sequence but only on some statistics of the transmitted sequence. For example, the equalizer may adapt its filter coefficients by knowing the different amplitudes and probabilities of the transmitted symbols. So for example, if the equalizer notices that symbols with lower amplitudes are being transmitted at a higher rate that expected, it may need to amplify specific frequencies of the received signal to return the probabilities of the different symbols to their original values. The blind mode of equalization is sometimes used in applications where transmitting a training sequence is not practical because the distortion of the channel is not sever, for example, and an equalizer will probably be able to detect the distortion of the channel without a training sequence, or the transmission of a training sequence would waste precious time that the communication system cannot afford to waste. In addition to blind adaptation of the equalizer filter being significantly more complex than trained adaptation, it is less guaranteed to reach the desired results, slower in adapting the filter, and is sometimes computationally intensive.
3. **Decision Directed (or Decision Feedback) Mode:** Whither the training mode or the blind mode of equalizer adaptation was used for the initial adaptation of the equalizer coefficients, the equalizer usually switches its mode of operation to this third mode once the initial adaptation is reached. The reason for doing this is to allow the equalizer to track and adapt to slow changes in the response of the channel as it changes over time. This mode of operation takes advantage of the fact that the output of the equalizer after the initial adaptation is close to the transmitted sequence of data with the exception of a small difference that results from imperfection in the initial equalizer adaptation and possibly noise that has been added to the transmitted signal as it was traveling through the channel. If the deviation from the perfect operation is small, the equalizer can pass its output through a decision device that determines the symbols that were most likely transmitted and then use the difference (or error) between the symbols before and after the decision device to further adapt the equalizer to eliminate this error as much as possible. As long as the channel does not change dramatically in a very short period of time and as long as the amount of added noise is reasonably small, the equalizer should be able to keep tracking the channel and adapt to its slow changes. If the channel impulse response changes significantly at a sudden, it may be necessary for the transmitter (via a request from the receiver) to transmit a training sequence again after the sudden change in the channel to allow the equalizer to readapt its filter to the new significantly different channel.

Structure of an Equalizer Filter

The structure of a finite impulse response (FIR) equalizer filter is what is called "tapped delay line" structure which is shown below. In this structure, a series of delays units each delays the signal by an amount equal to the symbol period. A tap, or access point is connected to each delayed version of the signal. Each tap is scaled by a particular value (known as a filter coefficient) and the scaled taps are all added to make the filter output signal. The equalizer adaptation process modifies the coefficients of the different taps.



The output sampled signal $y[n]$ is given by

$$y[n] = b_0 \cdot x[n] + b_1 \cdot x[n-1] + b_2 \cdot x[n-2] + b_3 \cdot x[n-3] + \ \dots \ + b_M \cdot x[n-M]$$

The adaptation process simply adjusts the coefficients b_0 through b_M until the filter response is the inverse response of the channel or close to it.

Adaptation Algorithms

Many adaptation algorithms have been proposed to adjust the equalizer coefficients. Different adaptation algorithms differ in complexity, computational requirements, and adaptation speed. The simplest algorithm is called the Least Mean Squares (LMS) which is discussed briefly below. In this algorithm, let `B` be a vector of length M containing the filter coefficients, `W` is a vector containing the received samples, `R` is a vector of length M containing the received samples that are still inside the equalizer, `Z` is a scalar that contains the output of the equalizer at a particular time instant, `E` is a scalar that contains the error value the error value at a particular time instant, and `D` is a vector that contains the desired values of the samples, `mu` is called the adaptation step size:

1. All filter coefficients are initialized to zero [`B = zeros(M,1)`]
2. The samples currently inside the equalizer filter are initialized to zero [`R = zeros(M,1)`]
3. Enter into a loop
4. Move the different samples of the received signal through the delays of the filter by one unit. [`R(2:N) = R(1:N-1)`]. The last sample will exit the filter and the location of the first sample will be vacated.
5. Feed the latest received sample to the input of the equalizer. This sample will take the place that was just vacated in the previous step [`R(1) = W(k)`]
6. Compute the output of the equalizer at that time instant [`Z(k) = C .* R`]
7. Compute error between desired sample and output of filter [`E(k) = Z(k) - D(k)`]
8. Update the filter coefficients using [`B = B - mu * E(k) * R`]
9. Repeat steps 4 to 8 until the training sequence is finished.