UESTC4004
Digital Communications

Instructors: Prof. Muhammad Imran and Sajjad Hussain

This week: Prof. Muhammad Imran muhammad.imran@glasgow.ac.uk

3.4 Equalization

- Nyquist filtering and pulse shaping schemes assumes that the channel is precisely known and its characteristics do not change with time

- However, in practice we encounter channels whose frequency response are either unknown or change with time
  - For example, each time we dial a telephone number, the communication channel will be different because the communication route will be different
  - However, when we make a connection, the channel becomes time-invariant
  - The characteristics of such channels are not known a priori

- Examples of time-varying channels are radio channels
  - These channels are characterized by time-varying frequency response characteristics

Channel distortions

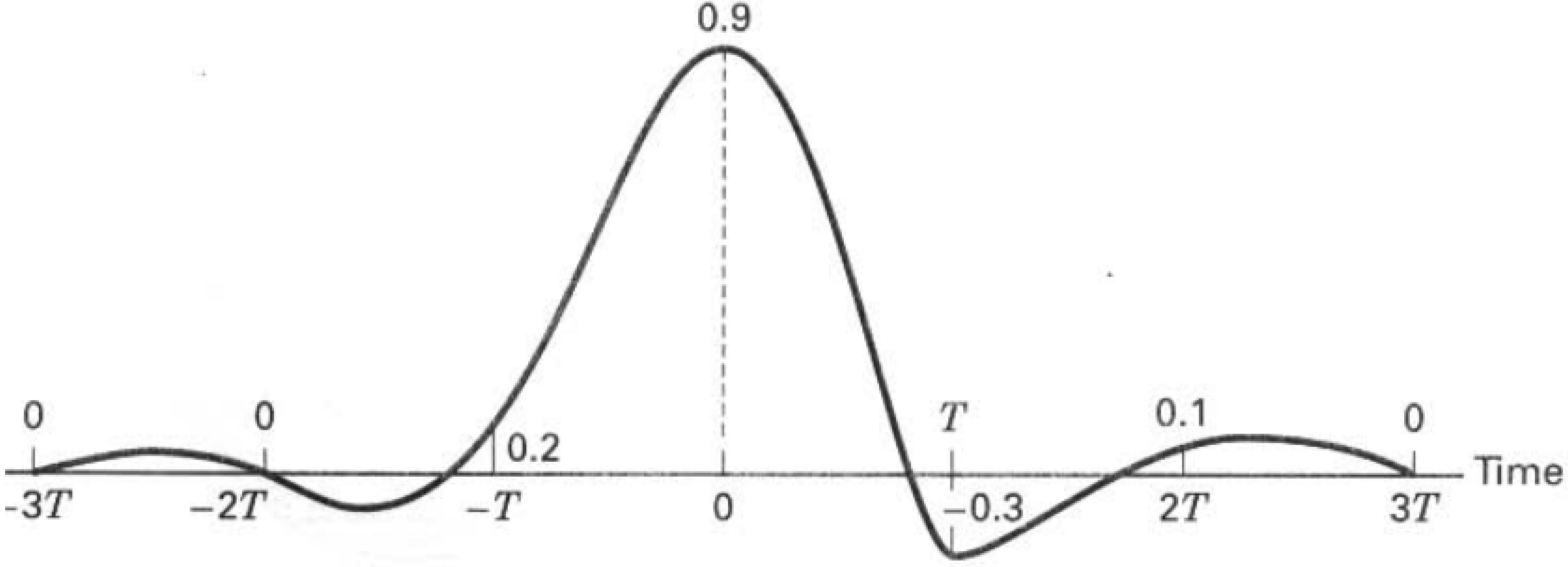
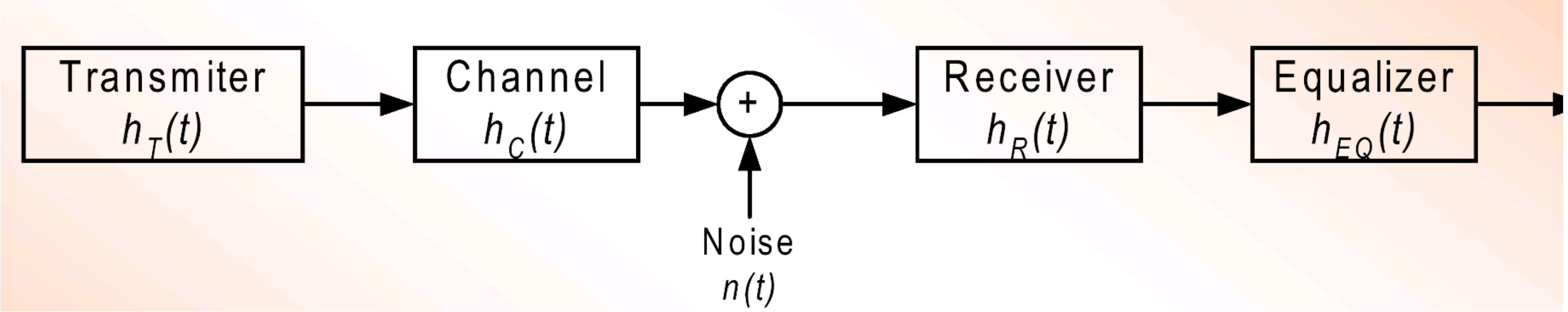


Figure 3.25 Received pulse exhibiting distortion.

- To compensate for channel induced ISI we use a process known as ***Equalization:*** a technique of correcting the frequency response of the channel
- The filter used to perform such a process is called an ***equalizer***



- Since $H_R(f)$ is matched to $H_T(f)$, we usually worry about $H_C(f)$
- The goal is to pick the frequency response $H_{EQ}(f)$ of the ***equalizer*** such that

$$H_c(f)H_{EQ}(f)=1 \quad \Rightarrow \quad H_{EQ}(f)=\frac{1}{H_C(f)}e^{-j\theta_c(f)}$$

- where
- $$|H_{EQ}(f)|=\frac{1}{|H_C(f)|}$$ and the phase characteristics $$\Theta_{EQ}(f)=-\Theta_C(f)$$

Problems with Equalization

- It can be difficult to determine the inverse of the channel response
  - If the channel response is zero at any frequency, then the inverse is not defined at that frequency
  - The receiver generally does not know what the channel response is. Channel changes in real time so equalization must be adaptive

- The equalizer can have an infinite impulse response even if the channel has a finite impulse response
  - The impulse response of the equalizer must usually be truncated

Equalization Techniques or Structures

- **Three Basic Equalization Structures**
  - **Linear Transversal Filter**
    - Simple implementation using Tap Delay Line or FIR filters
    - FIR filter has guaranteed stability (although adaptive algorithm which determines coefficients may still be unstable)
  - **Decision Feedback Equalizer**
    - Extra step in subtracting estimated residual error from signal
  - **Maximal Likelihood Sequence Estimator (Viterbi)**
    - “Optimal” performance
    - High complexity and implementation problem (not heavily used)

Linear Transversal Equalizer

- This is simply a linear filter with adjustable parameters
- The parameters are adjusted on the basis of the measurement of the channel characteristics
- A common choice for implementation is the ***transversal filter*** (Tap Delay Line) or the FIR filter with adjustable tap coefficient

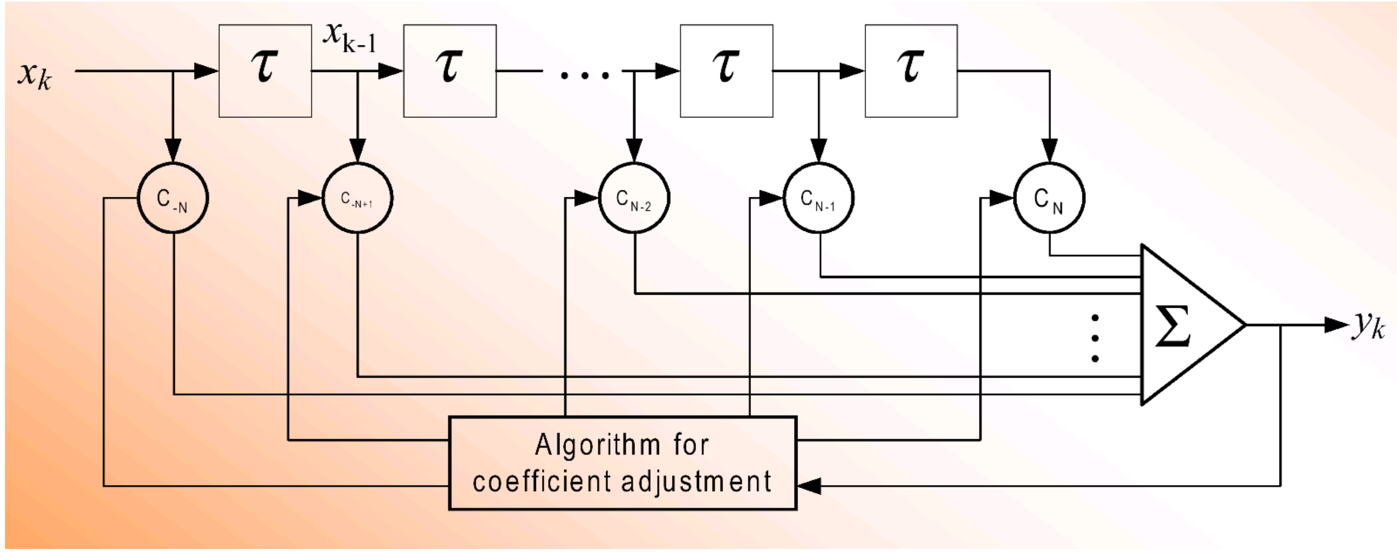


Fig. 3.26

- **Total number of taps = 2N+1**
- **Total delay = 2Nτ**

- N is chosen sufficiently large so that equalizer spans length of the ISI.
- Normally the ISI is assumed to be limited to a finite number of samples
- The output y_k of the Tap Delay Line equalizer in response to the input sequence $\{x_k\}$ is

$$y_k = \sum_{n=-N}^{N} x(k-n)c_n, \qquad k = -2N, \ldots, 2N$$

where c_n is the weight of the n^{th} tap

- Ideally, we would like the equalizer to eliminate ISI resulting in

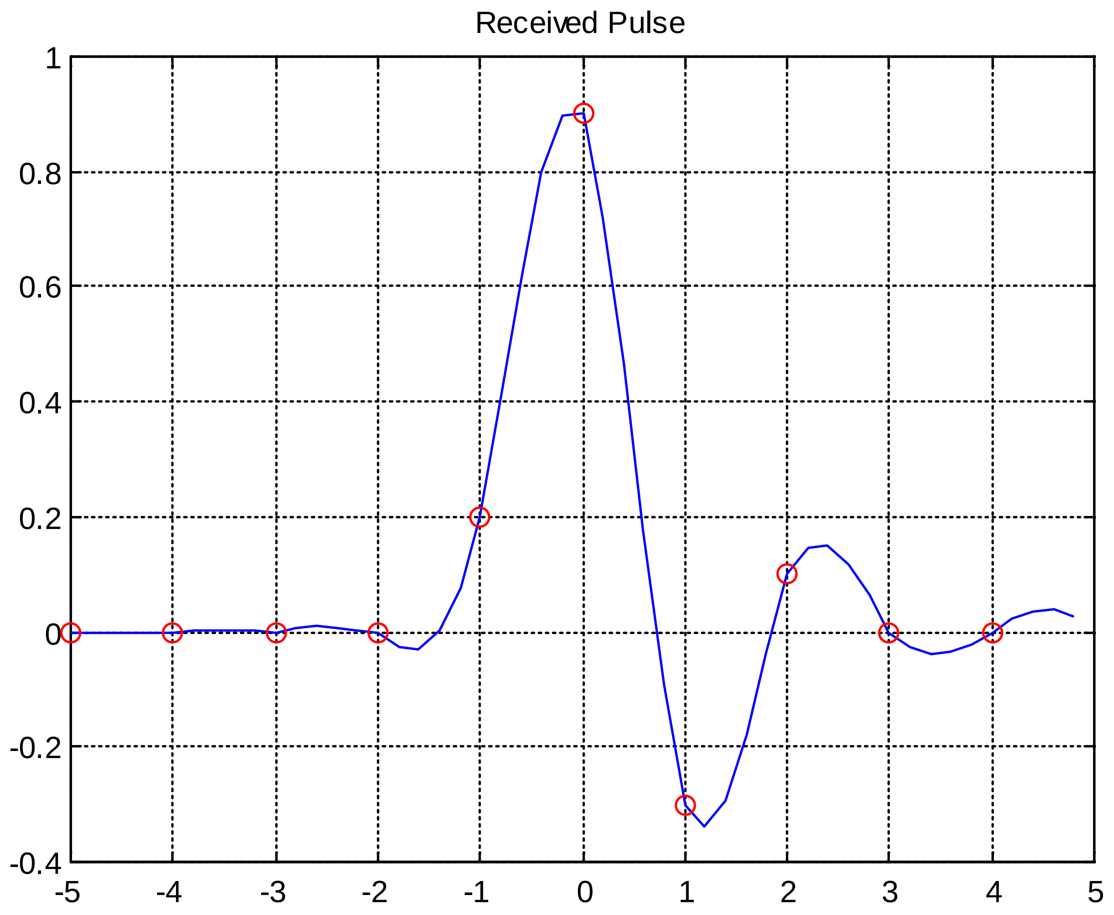
$$y_k = \begin{cases} 1, & k = 0 \\ 0, & k \neq 0 \end{cases}$$

- But this cannot be achieved in practice.

- Zero forcing Equalizer : However, the tap gains can be chosen such that

$$y_k = \begin{cases} 1, & k = 0 \\ 0, & k = \pm 1, \pm 2, \ldots\ldots \pm N \end{cases}$$

Example 3.5: (Page 155)



Zero-Forcing Solution

- For N=1

$$k=-1, \qquad y(-1)=c_{-1}x(-1-(-1))+c_0x(-1-(0))+c_1x(-1-(1))$$

$$k=0, \qquad y(0)=c_{-1}x(0-(-1))+c_0x(0-(0))+c_1x(0-(1))$$

$$k=1, \qquad y(1)=c_{-1}x(1-(-1))+c_0x(1-(0))+c_1(1-(1))$$

$$\Rightarrow \begin{bmatrix} y(-1) \\ y(0) \\ y(1) \end{bmatrix} = \underbrace{\begin{bmatrix} x(0) & x(-1) & x(-2) \\ x(1) & x(0) & x(-1) \\ x(2) & x(1) & x(0) \end{bmatrix}}_{(2N+1)\times(2N+1)} \underbrace{\begin{bmatrix} c_{-1} \\ c_0 \\ c_1 \end{bmatrix}}_{(2N+1)\times 1}$$

- For N=2

$$\begin{bmatrix} y(-2) \\ y(-1) \\ y(0) \\ y(1) \\ y(2) \end{bmatrix} = \begin{bmatrix} x(0) & x(-1) & x(-2) & x(-3) & x(-4) \\ x(1) & x(0) & x(-1) & x(-2) & x(-3) \\ x(2) & x(1) & x(0) & x(-1) & x(-2) \\ x(3) & x(2) & x(1) & x(0) & x(-1) \\ x(4) & x(3) & x(2) & x(1) & x(0) \end{bmatrix} \begin{bmatrix} c_{-2} \\ c_{-1} \\ c_{0} \\ c_{1} \\ c_{2} \end{bmatrix}$$

Generalizing results:

$$\mathbf{c} = \mathbf{X}^{-1} y$$

$$where \quad \mathrm{y} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad for\ N = 1$$

Example

- Design a 3-taps zero forcing equalizer for input $x(n) = \{0, -0.1, 0.15, 0.87, 0.12, -0.2, 0\}$ in which $x(0) = 0.87$.

- Answer: c=[0.236 1.220 -0.226]’.