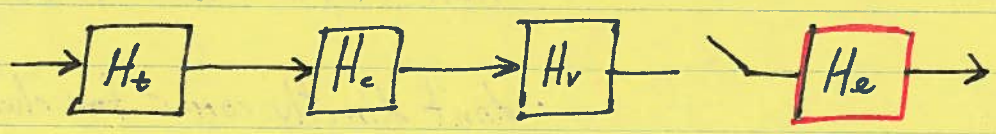
12 Equalization

12.1 Overview

- We have seen optimal tx + rx filter settings

$$|H_r(f)| = \frac{\alpha |P_r|^{\frac{1}{2}}}{G_n^{1/4} |H_c|^{\frac{1}{2}}} \qquad |H_t| = \frac{(A/\alpha) |P_r|^{\frac{1}{2}} G_n^{1/4}}{|H_c|^{1/2}}$$

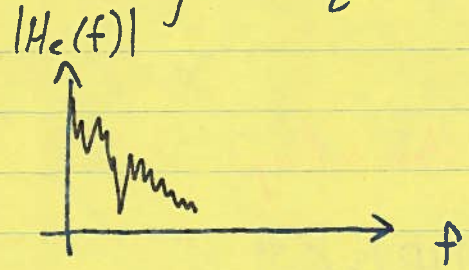
- note the inverse relationship to the CHANNEL FILTER
- In general these settings not easy to obtain (especially simultaneously)
- So we let H_r and H_t achieve more limited pulse shaping and noise filtering jobs...
- ... and leave job of correcting for tx/rx nonidealities + channel to another filter... the EQUALIZER



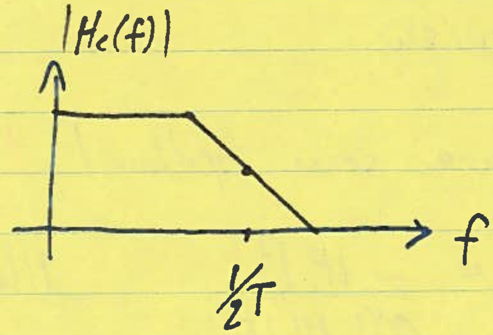
want... $H_t H_c H_r H_e = H_{RC}$: Nyquist filter

$$\sum_{k=-\infty}^{\infty} H_{RC}\left(f + \frac{k}{T}\right) = \text{const.}$$

- a good equalizer turns

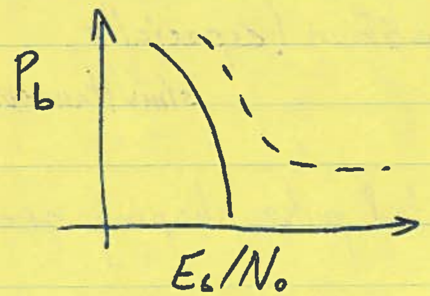


to

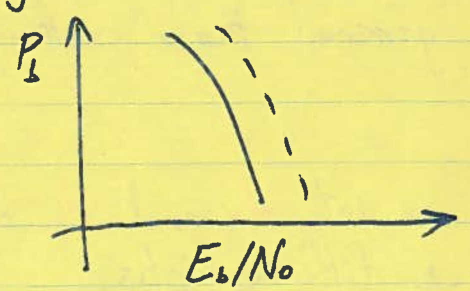


and in the process corrects AMPLITUDE & PHASE DISTORTION

which otherwise can cause significant degradation to BER performance



unlike simple noise errors



12.2 Equalizer Types

Filter-Based

Linear
Transversal (FIR) filters
- ZFE
- MMSE

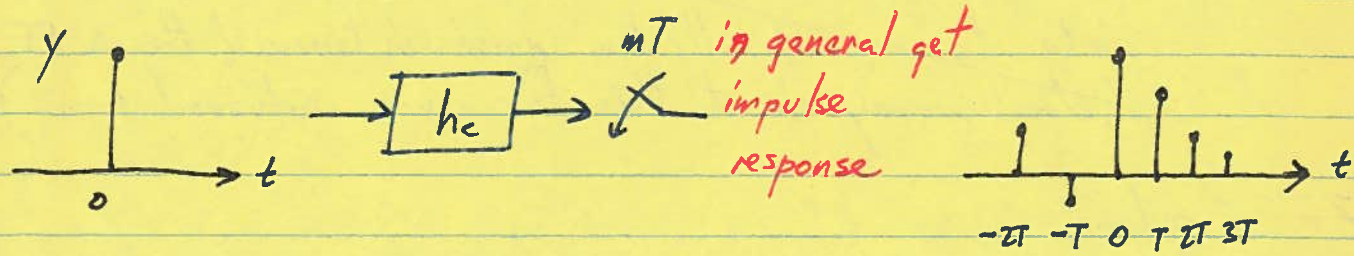
Nonlinear
Feedback post/after decision
- DFE

Non-Filter Based (MLSE)

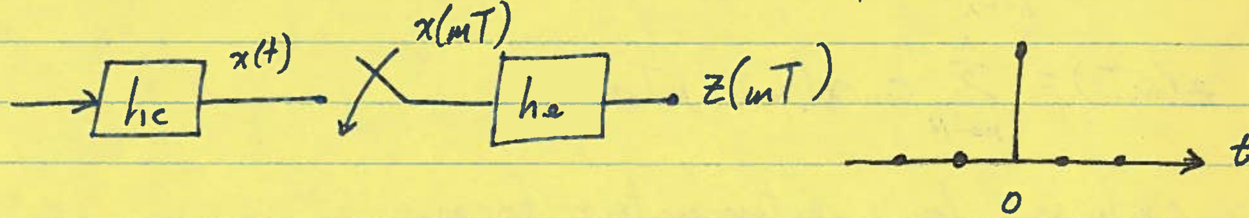
- don't directly correct for channel
- observe sequence for some time & make guess
- dynamic programming (Viterbi)

12.3 ZFE

- the channel...



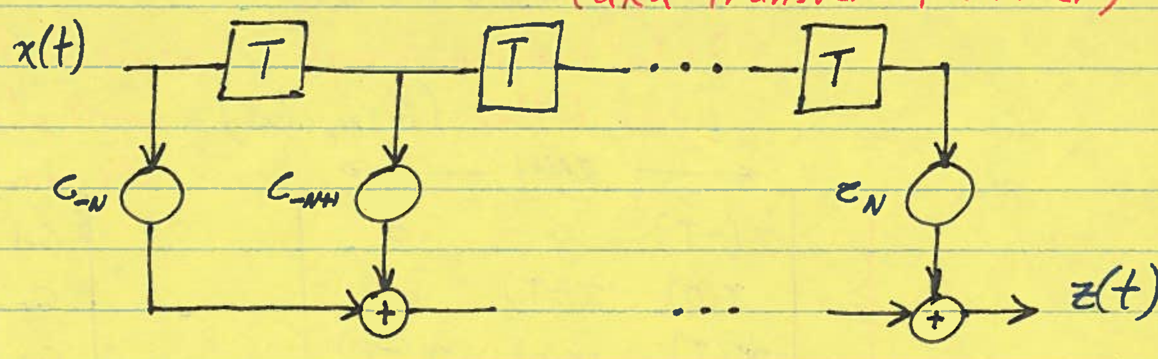
- to this we attach the ZFE... which attempts to achieve



- it does this by essentially trying to obtain/realize

$$H_e(f) = \frac{1}{H_c(f)} = \frac{1}{|H_c(f)|} \cdot e^{-j\theta_c(f)}$$

- we attempt this in the form of a FIR filter implementation (aka Transversal Filter)



- of course the finite structure can only serve as an approximation to the ultimate goal

-

• Note: Because ZFE seeks to invert the channel there's the option of lumping some M.F. functions (opportunity) into it. specifically the noise whitener of the WMF (don't worry about this for now... advanced topic)

ZFE's output

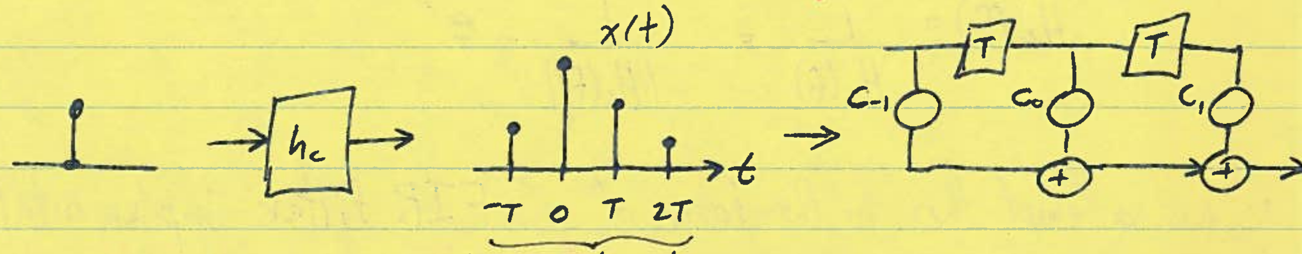
$$z(t) = \sum_{n=-N}^{N} c_n\, x(t-nT)$$

$$z(mT) = \sum_{n=-N}^{N} c_n\, x(mT-nT)$$

• express in vector/matrix form

$$\bar{z} = \bar{X}\,\bar{c}$$

← convolution matrix : satate of equalizer as a fn. of time



channel impulse response

① what's moving through your filter (K impulses)

② what its being multiplied by as its moving through

2N+1

time = K+2N

$$\begin{bmatrix} \\ \\ \\ \\ \\ \\ \end{bmatrix} = \begin{bmatrix} x(-T) & 0 & 0 \\ x(0) & x(-T) & 0 \\ x(T) & x(0) & x(-T) \\ x(2T) & x(T) & x(0) \\ 0 & x(2T) & x(T) \\ 0 & 0 & x(2T) \end{bmatrix} \cdot \begin{bmatrix} c_{-1} \\ c_0 \\ c_1 \end{bmatrix}$$

$$\bar{z} = \bar{X} \cdot \bar{c}$$

$$\bar{z} = \bar{X} \cdot \bar{c}$$

your desired output vector: a 1 and the rest zeros

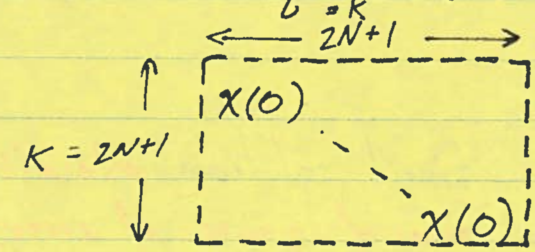
- how to find your equalizer settings ???

$$\bar{c} = \bar{X}^{-1} \bar{z}$$

either underdetermined or overdetermined

- can't invert $\bar{X}$ if it's not square, so...

... choose a square version of it (book's approach)



$x(0)$ at the corner's
i.e. take square sample of convolution matrix that tracks $x(0)$ of channel impulse response as it propagates through ZFE

- e.g. 

$$\begin{array}{cc} & x(-1) \\ & x(0) \\ K & x(1) \\ & x(2) \\ & x(3) \end{array} \begin{bmatrix} 36 \\ 230 \\ 97 \\ 37 \\ 18 \end{bmatrix} \text{mV} = \bar{x} \qquad \text{*say you want } \bar{z} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\bar{X} = \begin{bmatrix} 230 & 36 & 0 & 0 & 0 \\ 97 & 230 & 36 & 0 & 0 \\ 37 & 97 & 230 & 36 & 0 \\ 18 & 37 & 97 & 230 & 36 \\ 0 & 18 & 37 & 97 & 230 \end{bmatrix}, \quad \bar{c} = \bar{X}^{-1} \bar{z} = \begin{bmatrix} -0.77 \\ 4.93 \\ -1.98 \\ 0.14 \\ -0.13 \end{bmatrix} \text{ normalize using } \frac{c_i}{\Sigma c_i} \begin{bmatrix} -.097 \\ .624 \\ -.25 \\ .017 \\ .016 \end{bmatrix}$$

- another approach relies on a least squares strategy

12.4 ZFE-LS

- least-squares: a standard approach to the solution of overdetermined systems

$$\bar{c}_{LS} = \arg\min_{\bar{c}} \left\| \bar{z} - \bar{X}\bar{c} \right\|^2$$

← entry (component) by component take difference, square it add it to the results obtaine from other entries

$$\sum_i \left(z_i - \sum_j x_{ij} c_j\right)^2$$

result:

$$\bar{c}_{LS} = \left(\bar{X}^T \cdot \bar{X}\right)^{-1} \bar{X}^T \cdot \bar{z}$$

→ final solution minimizes sum of squares of errors between desired $\bar{z}$ entries & $\bar{c}$ entries chosen to attain them

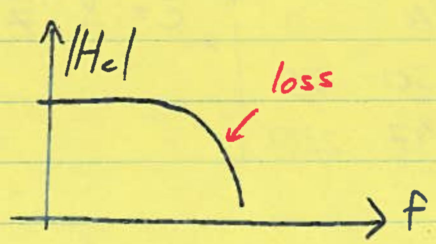
this is a MMSE... but not exactly a MMSE equalizer

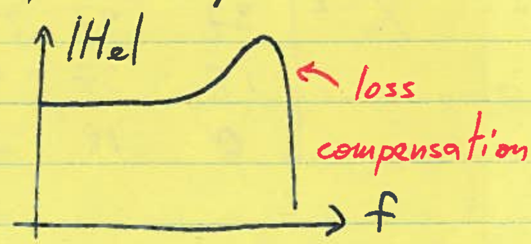
12.5 MMS Equalizer

- A key problem with ZFE...

noise enhancement ... how?

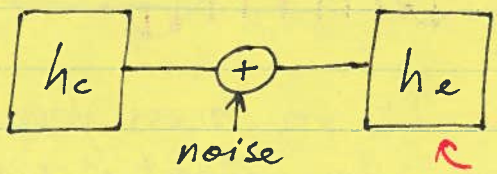
- EQ boosts response such that originally sent signal level gets through





- thus get original signal level back and no distortion

- but in the process you amplified noise



loss compensation (i.e. just getting back my original signal) amplifies noise

- so build an equalizer that tries to find the best tradeoff between noise and ISI

- end up with something similar to ZFE-LS

- instead of $\bar{c}_{LS} = \bar{c}_{ZFE} = (\bar{X}^T X)^{-1} \bar{X}^T \bar{z}$

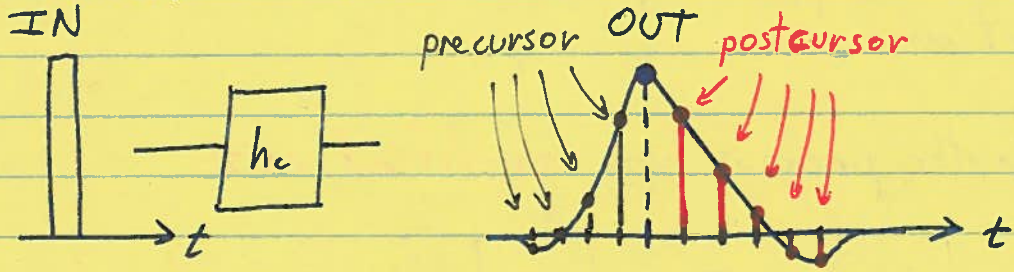
  use $\bar{c}_{MMSE} = \left(\frac{\overset{\text{identity matrix}}{I}}{SNR} + \bar{X}^T X \right)^{-1} \bar{X}^T \bar{z}$ $\qquad SNR = \frac{\sigma_x^2}{\sigma_n^2}$

- as SNR drops... clearly that part of $\bar{c}_{MMSE}$ becomes more important as a filter tap determinator

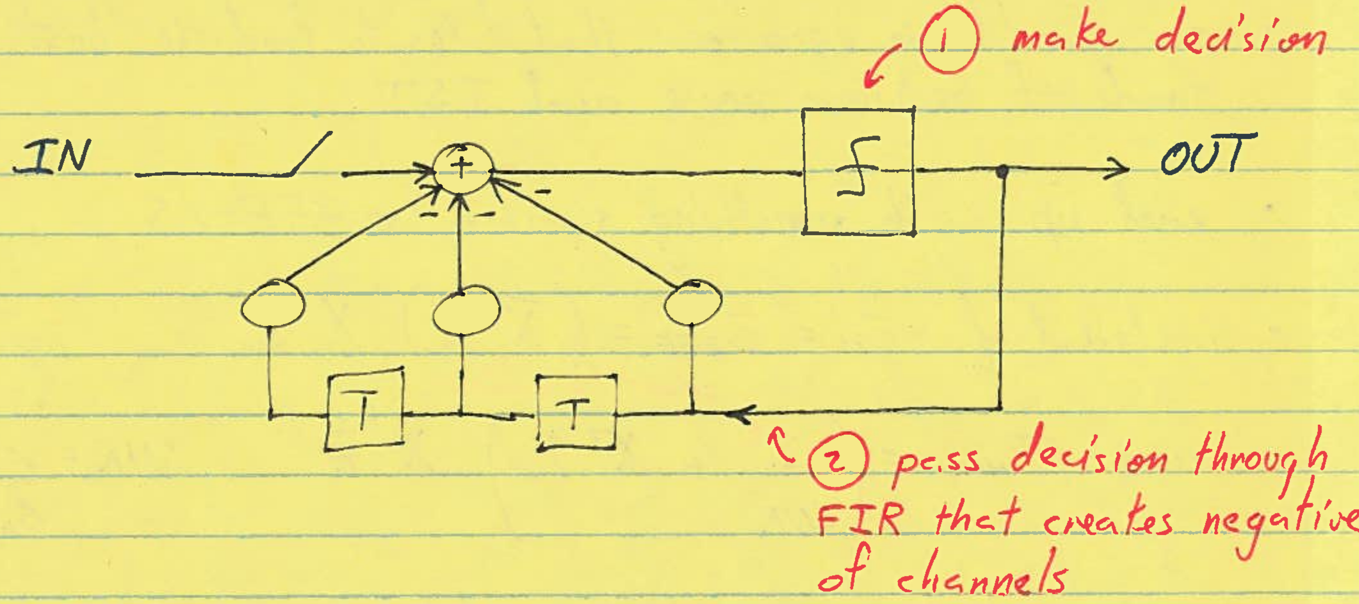
12.6 DFE

- A nonlinear approach

- Consider...



- with regards to postcursor IF you guess your OUT bit correctly and you know the channel response you can exactly figure out what the postcursor is and thus perfectly cancel that (i.e. the postcursor) contribution to ISI

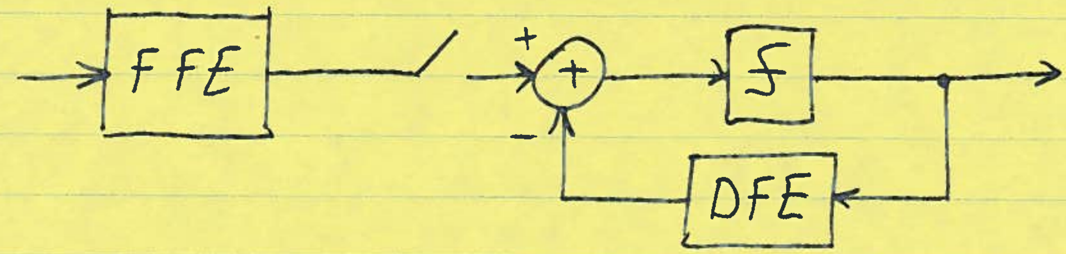


- key advantage?

- NO noise enhancement: feedback signal based on perfect decision, noise has no way of making it back through the filter

- DFE only handles postcursors

- may still need feedforward filter for precursors



- what happens when detector makes mistake?
- you make ISI worse!!!
  - double ISI & create error propagation
- not a big issue in wires at 10^{-8} or 10^{-9} BER (as you shoot for 10^{-15} to 10^{-20})
- less common in wireless because of its higher BER