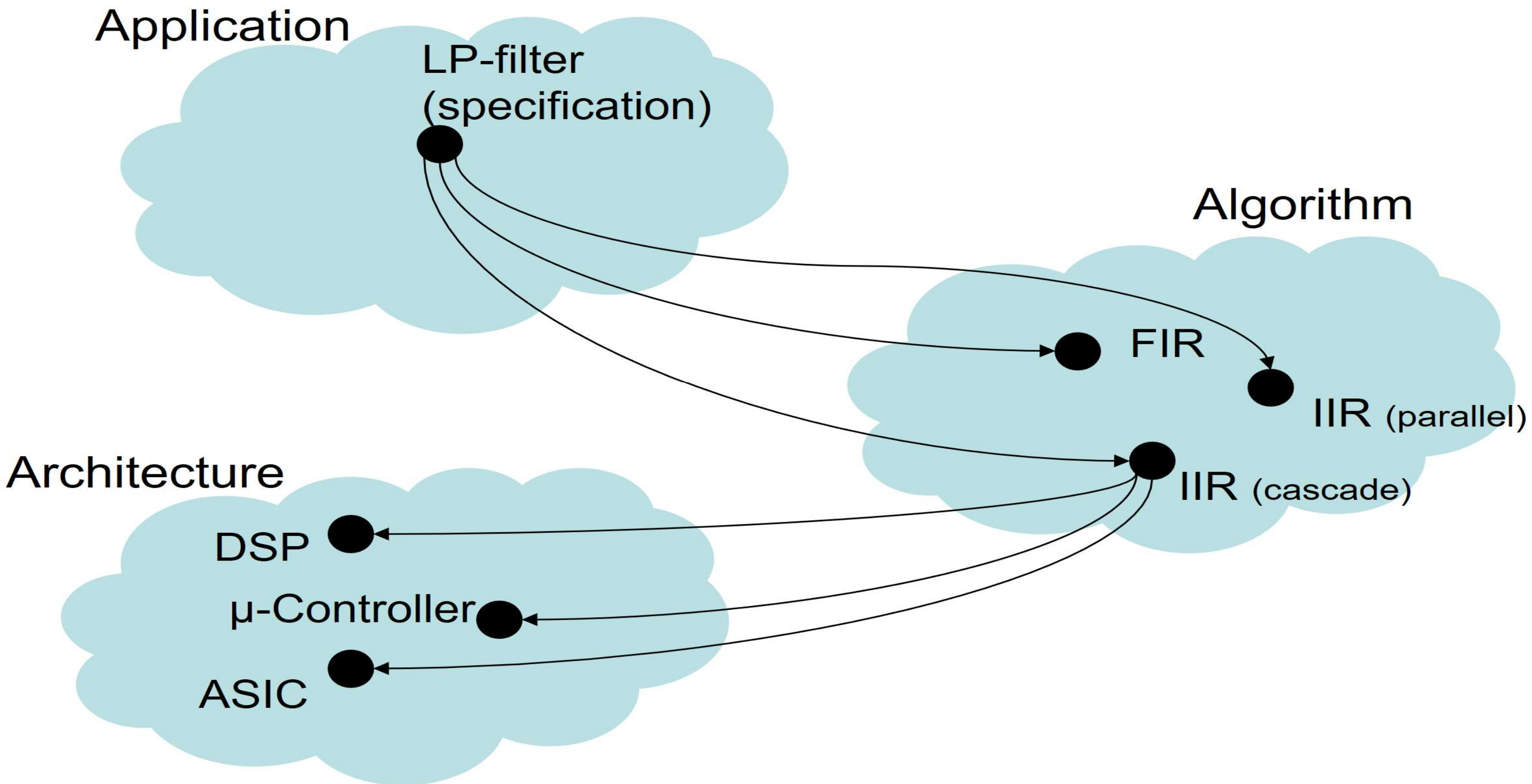
IEE 1711: Applied Signal Processing

Professor Muhammad Mahtab Alam (muhammad.alam@taltech.ee)
Lab Instructor: Julia Berdnikova



Outline

- Lecture 8: Digital Communication – Optimal Receiver
  - Brief Followup

- Lecture 9: Digital Communication – Optimal Receiver (Cont..)
  - Equalization – Overview
    - Digital Equalizer's types
  - Zero-forcing Equalization

    Linear filters

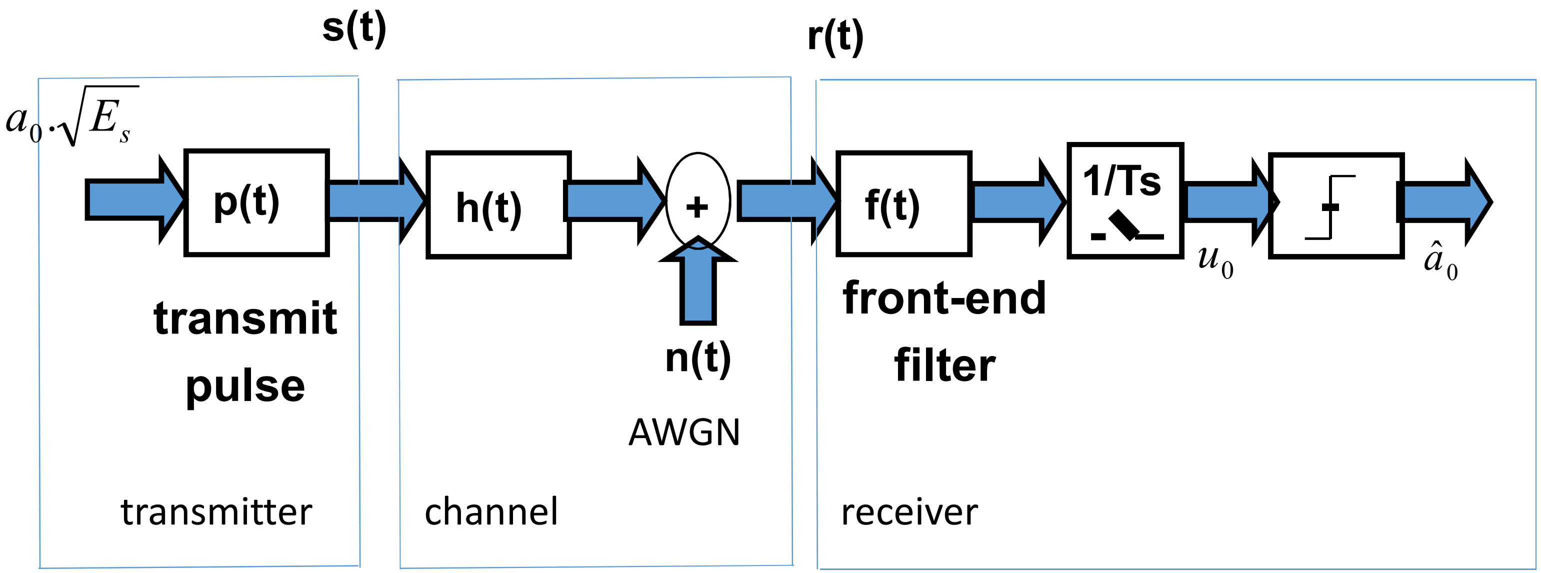
    Decision feedback equalizers
  - MMSE Equalization
  - Adaptive Equalization

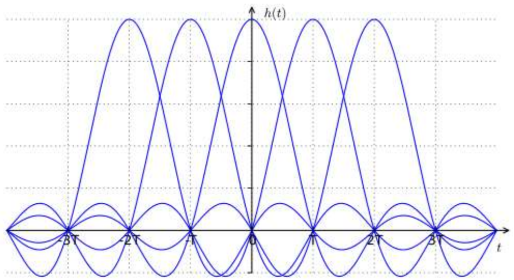
  - Summary

Optimal receiver design

Receiver:

A receiver structure is postulated (front-end filter + symbol-rate sampler + memory-less decision device). For transmission of 1 symbol, it was found that the front-end filter should be `matched' to the received pulse.





Consecutive raised-cosine impulses, demonstrating zero-ISI property

Digital equalizer types

- Viterbi equalizer: Finds the maximum likelihood (ML) optimal solution to the equalization problem. Its goal is to minimize the probability of making an error over the entire sequence.
- **Linear equalizer:** processes the incoming signal with a linear filter
  - MMSE equalizer: designs the filter to minimize $E[|e|^2]$, where e is the error signal, which is the filter output minus the transmitted signal.
  - Zero forcing equalizer: approximates the inverse of the channel with a linear filter.
- Decision feedback equalizer: augments a linear equalizer by adding a filtered version of previous symbol estimates to the original filter output.
- Blind equalizer: estimates the transmitted signal without knowledge of the channel statistics, using only knowledge of the transmitted signal's statistics.
- Adaptive equalizer: is typically a linear equalizer or a DFE. It updates the equalizer parameters (such as the filter coefficients) as it processes the data. Typically, it uses the MSE cost function; it assumes that it makes the correct symbol decisions, and uses its estimate of the symbols to compute e, which is defined above.
- BCJR equalizer: uses the BCJR algorithm (also called the Forward-backward algorithm) to find the maximum *a posteriori* (MAP) solution. Its goal is to minimize the probability that a given bit was incorrectly estimated.
- Turbo equalizer: applies turbo decoding while treating the channel as a convolutional code.

Optimal receiver

Problem Statement :

- Optimal receiver structure consists of

  * Whitened Matched Filter (WMF) front-end

    (= matched filter + symbol-rate sampler + equalizer filter)

  * Maximum Likelihood Sequence Estimator (MLSE),

    (instead of simple memory-less decision device)

- Equalization – Overview
- Maximum Likelihood Sequence Estimator
- Zero-forcing Equalization

  Linear filters

  Decision feedback equalizers

Equalization

Linear equalization (LE):

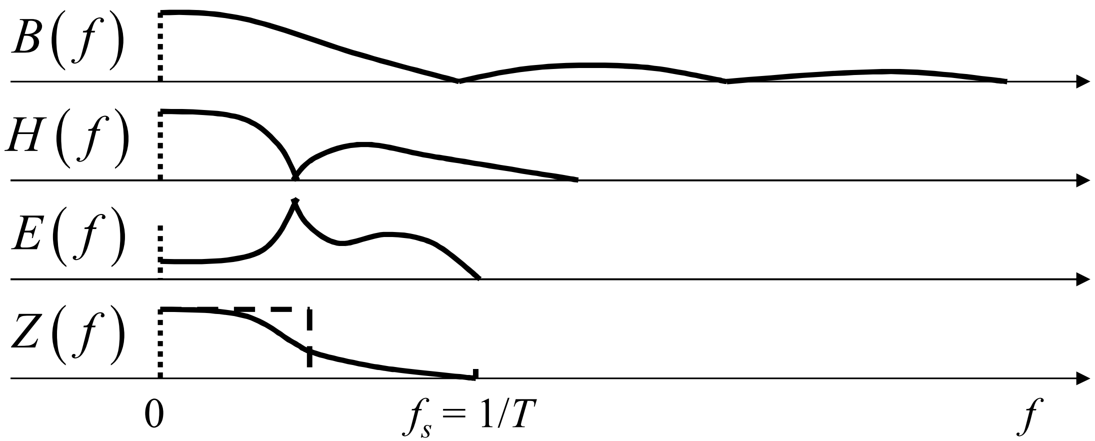
Performance is not very good when the frequency response of the frequency selective channel contains deep fades.

Zero-forcing algorithm aims to eliminate the intersymbol interference (ISI) at decision time instants (i.e. at the center of the bit/symbol interval).

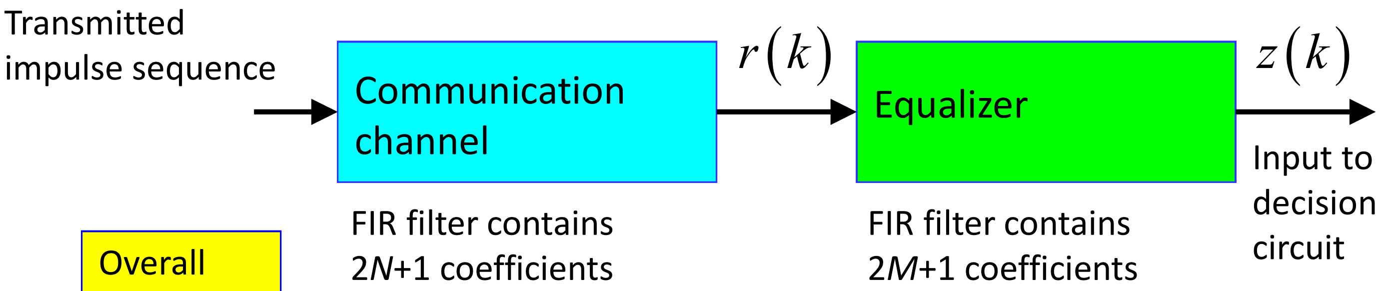
Linear equalization, zero-forcing algorithm

Basic idea: $$Z(f) = B(f)H(f)E(f)$$

Raised cosine spectrum	=	Transmitted symbol spectrum	Channel frequency response (incl. T & R filters)	Equalizer frequency response



Zero-forcing equalizer



Overall channel

Channel impulse response

$$h(k) = \sum_{n=-N}^{N} h_n \delta(k-n)$$

Equalizer impulse response

$$c(k) = \sum_{m=-M}^{M} c_m \delta(k-m)$$

Coefficients of equivalent FIR filter

$$f_k = \sum_{m=-M}^{M} c_m h_{k-m} \quad (-M \le k \le M)$$

Zero-forcing Equalizers

`Figure of merit'

$$\boxed{\gamma_{\mathrm{LE}} \le \gamma_{\mathrm{DFE}} \le \gamma_{\mathrm{MLSE}} \le \gamma_{\mathrm{MF}}}$$

- receiver with higher `figure of merit' has lower error probability
- γ_{MF} is `matched filter bound' (transmission of 1 symbol)
- DFE-performance lower than MLSE-performance, as DFE relies on only the first channel impulse response sample h_0 (eliminating all other h_i 's), while MLSE uses energy of all taps h_i .

Problem statement (revisited)

- Cheap alternative for MLSE/Viterbi ?
- Solution: equalization filter + memory-less decision device (`slicer’)

  Linear filters

  Non-linear filters (decision feedback)
- Complexity : linear in number filter taps
- Performance : with channel coding, approaches MLSE performance

Minimum Mean Square Error Equalizers

- Zero-forcing equalizers: minimize noise at slicer input under zero-ISI constraint
- A **minimum mean square error** (**MMSE**) estimator is an estimation method which minimizes the mean square error (MSE), which is a common measure of estimator quality, of the fitted values of a dependent variable.
    - In estimation/ decision theory, a **Bayes estimator** or a **Bayes action** is an estimator or decision rule that minimizes the posterior expected value of a loss function (i.e., the **posterior expected loss**). Equivalently, it maximizes the posterior expectation of a utility function. An alternative way of formulating an estimator within Bayesian statistics is maximum a posteriori estimation.
- Generalize the criterion of optimality to allow for residual ISI at the slicer & reduce noise variance at the slicer

=Minimum mean-square error equalizers

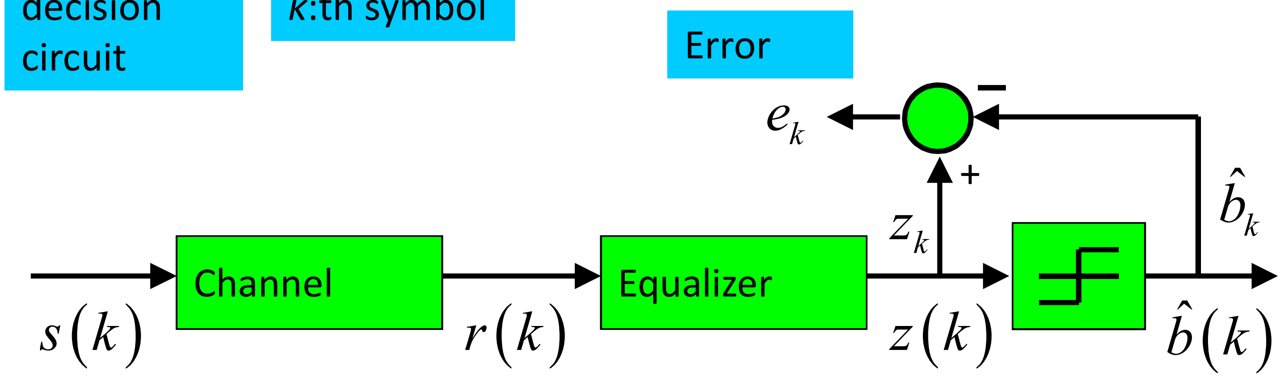
Minimum Mean Square Error (MMSE)

The aim is to minimize: $J = E\left| e_k \right|^2$

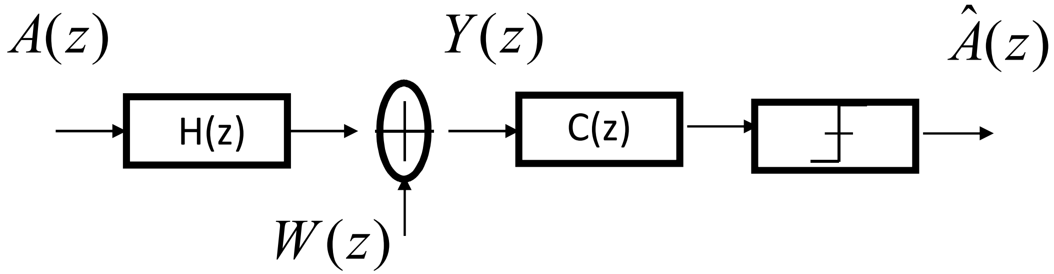
$e_k = z_k - \hat{b}_k$ (or $\hat{b}_k - z_k$ depending on the source)

Input to decision circuit

Estimate of *k*:th symbol



MMSE Equalizers



$$C(z) = \frac{S_A(z).H^*(\frac{1}{z^*})}{S_A(z).H(z).H^*(\frac{1}{z^*}) + S_W(z)} = \frac{H^*(\frac{1}{z^*})}{H(z).H^*(\frac{1}{z^*}) + \sigma_W^2}$$

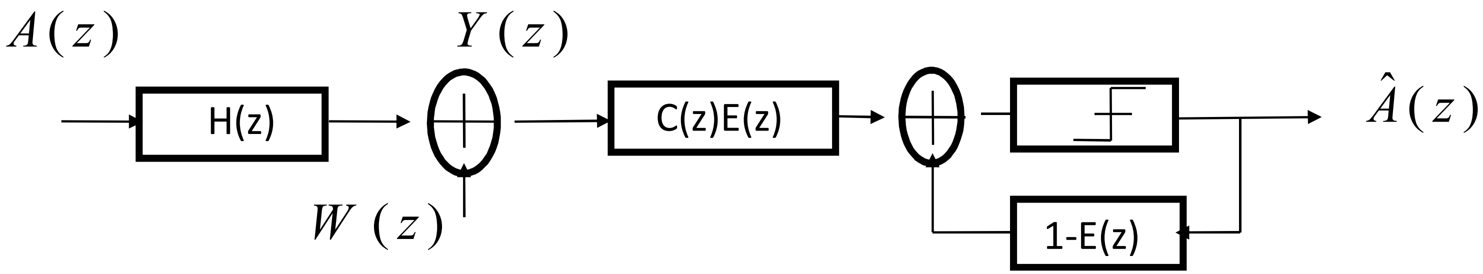
- $S_A(z) = 1 =$ signal power spectrum (normalized)
- $S_W(z) = \sigma_W^2 =$ noise power spectrum (white)
- for zero noise power -> zero-forcing $C(z) = H^{-1}(z)$
- $H^*(\frac{1}{z^*})$ (in the nominator) is a discrete-time matched filter, often `difficult' to realize in practice

(stable poles in H(z) introduce anticausal MF)

MMSE Equalizers

MMSE Decision Feedback Equalizer :

- MMSE-LE has correlated `slicer errors' (=difference between slicer in- and output)
- MSE may be further reduced by incorporating a `whitening' filter (prediction filter) E(z) for the slicer errors



- E(z)=1 -> linear equalizer

MSE vs. equalizer coefficients

$$J = E|e_k|^2 =$$ quadratic multi-dimensional function of equalizer coefficient values

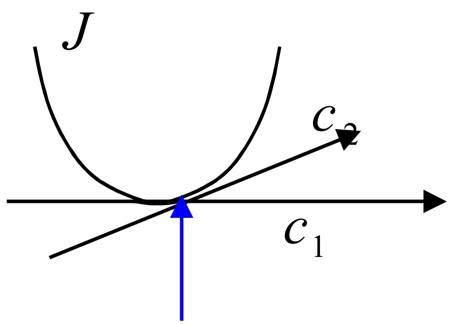


Illustration of case for two real-valued equalizer coefficients (or one complex-valued coefficient)

MMSE aim: find minimum value directly (Wiener solution), or use an algorithm that recursively changes the equalizer coefficients in the correct direction (towards the minimum value of J)!

Wiener solution

We start with the Wiener-Hopf equations in matrix form:

$$\mathbf{R}\mathbf{c}_{\text{opt}} = \mathbf{p}$$

$\mathbf{R}$ = correlation matrix (M x M) of received (sampled) signal values r_k

$\mathbf{p}$ = vector (of length M) indicating cross-correlation between received signal values r_k and estimate of received symbol $\hat{b}_k$

$\mathbf{c}_{\text{opt}}$ = vector (of length M) consisting of the optimal equalizer coefficient values

(We assume here that the equalizer contains M taps)

Correlation matrix R & vector p

$$\mathbf{R} = E\left[\mathbf{r}(k)\mathbf{r}^{*T}(k)\right]$$

where $\mathbf{r}(k) = \left[r_k, r_{k-1}, \ldots, r_{k-M+1}\right]^T$

M samples

$$\mathbf{p} = E\left[\mathbf{r}(k)\hat{b}_k^{*}\right]$$

- Before we can perform the stochastical expectation operation, we must know the stochastical properties of the transmitted signal (and of the channel if it is changing).
- Usually we do not have this information => some non-stochastical algorithm like Least-mean-square (LMS) must be used.

Algorithms

Stochastical information (**R** and **p**) is available:

1. Direct solution of the Wiener-Hopf equations:

$$\mathbf{R}\mathbf{c}_{\text{opt}} = \mathbf{p} \quad \Rightarrow \quad \mathbf{c}_{\text{opt}} = \mathbf{R}^{-1}\mathbf{p}$$

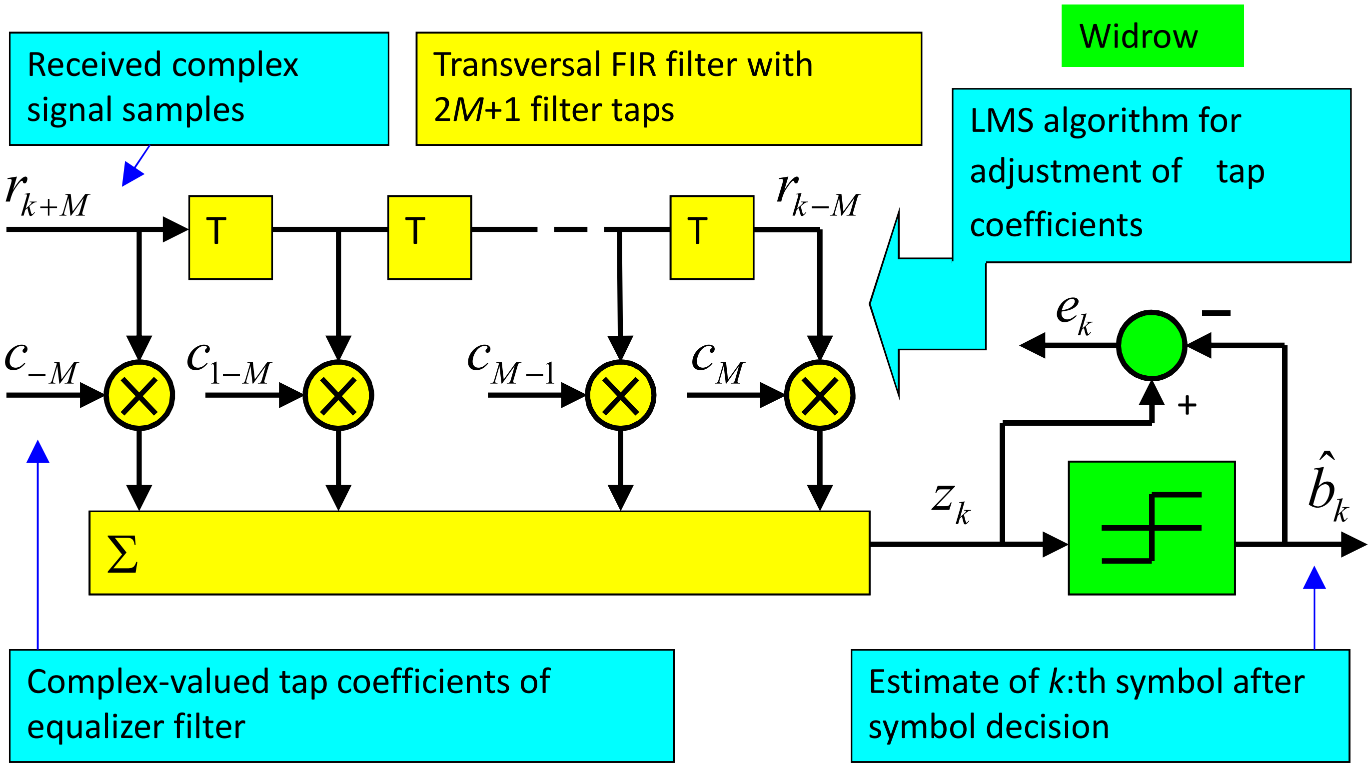
Inverting a large matrix is difficult!

2. Newton's algorithm (fast iterative algorithm)
3. Method of steepest descent (this iterative algorithm is slow but easier to implement)

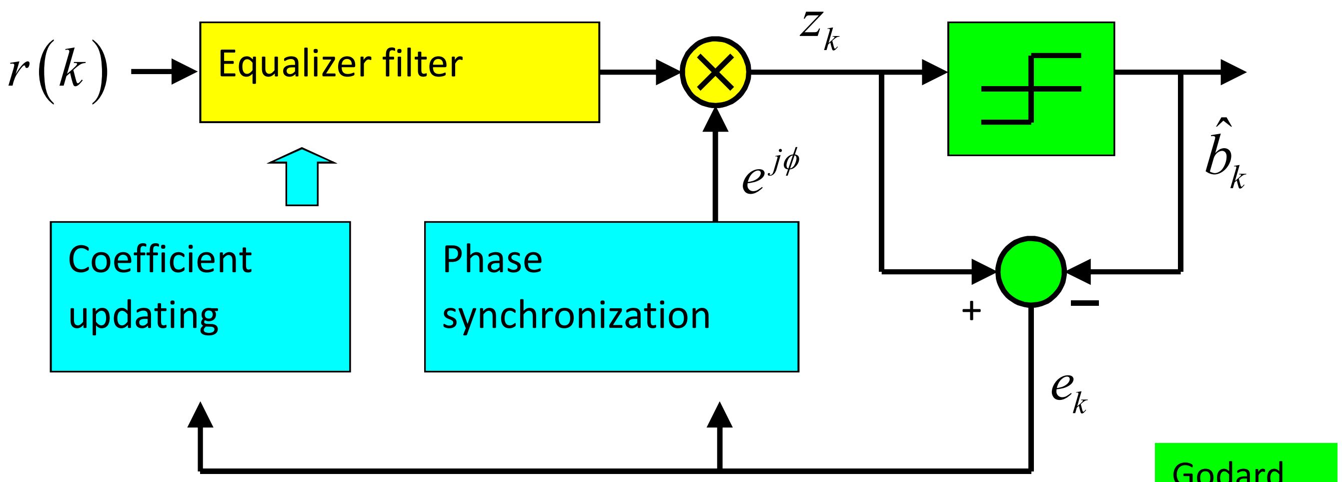
R and p are not available:

Use an algorithm that is based on the received signal sequence directly. One such algorithm is Least-Mean-Square (LMS).

Conventional linear equalizer of LMS type



Joint optimization of coefficients and phase



Godard

Proakis, Ed.3, Section 11-5-2

Minimize:

$$J = E\left| e_k \right|^2 \qquad e_k = z_k - \hat{b}_k = \left(\sum_{m=-M}^{M} c_m \, r_{k-m} \right) \exp\left(j\phi \right) - \hat{b}_k$$

Least-mean-square (LMS) algorithm

(derived from "method of steepest descent")

for convergence towards minimum mean square error (MMSE)

Real part of *n*:th coefficient:
$$\mathrm{Re}\{c_n(i+1)\} = \mathrm{Re}\{c_n(i)\} - \Delta \frac{\partial |e_k|^2}{\partial \left[\mathrm{Re}\{c_n\}\right]}$$

Imaginary part of *n*:th coefficient:
$$\mathrm{Im}\{c_n(i+1)\} = \mathrm{Im}\{c_n(i)\} - \Delta \frac{\partial |e_k|^2}{\partial \left[\mathrm{Im}\{c_n\}\right]}$$

$$|e_k|^2 = e_k e_k^{\ *}$$

Phase:
$$\phi(i+1) = \phi(i) - \Delta_\phi \frac{\partial |e_k|^2}{\partial \phi}$$

$2(2M+1)+1$ equations

Iteration index

Step size of iteration

Effect of iteration step size

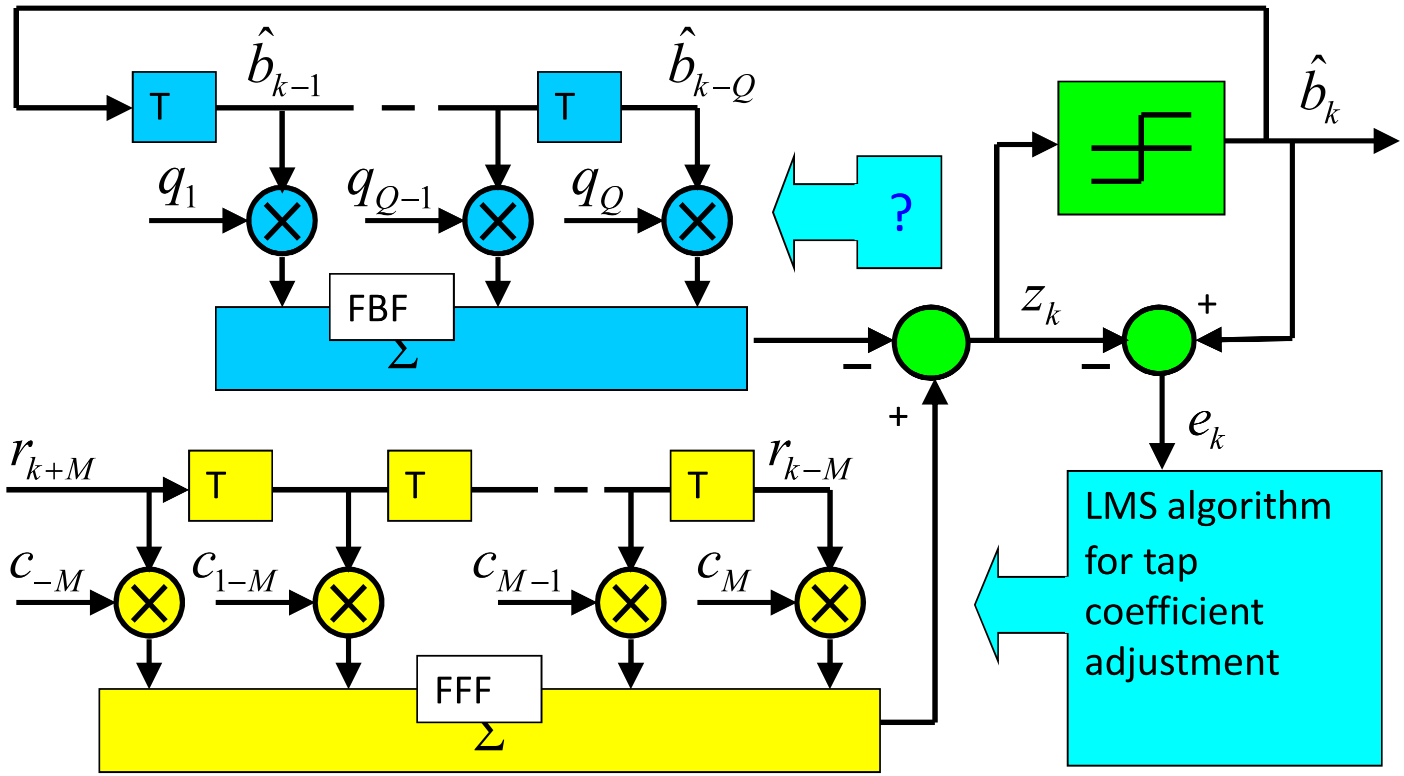
Δ

smaller ← → larger

Δ_ϕ

Slow acquisition	Poor stability
Poor tracking performance	Large variation around optimum value

Decision feedback equalizer



Decision feedback equalizer (cont.)

The purpose is again to minimize $J = E\left|e_k\right|^2 \approx \left|e_k\right|^2$

where $\left|e_k\right| = \left|z_k - \hat{b}_k\right| = \left|\sum_{m=-M}^{M} c_m r_{k-m} - \sum_{n=1}^{Q} q_n \hat{b}_{k-n} - \hat{b}_k\right|$

Feedforward filter (FFF) is similar to filter in linear equalizer

tap spacing smaller than symbol interval is allowed => fractionally spaced equalizer
=> oversampling by a factor of 2 or 4 is common

Feedback filter (FBF) is used for either reducing or canceling samples of previous symbols at decision time instants

tap spacing must be equal to symbol interval

Decision feedback equalizer (cont.)

The coefficients of the feedback filter (FBF) can be obtained in either of two ways:

- Recursively (using the LMS algorithm) in a similar fashion as FFF coefficients

Proakis, Ed.3, Section 11-2

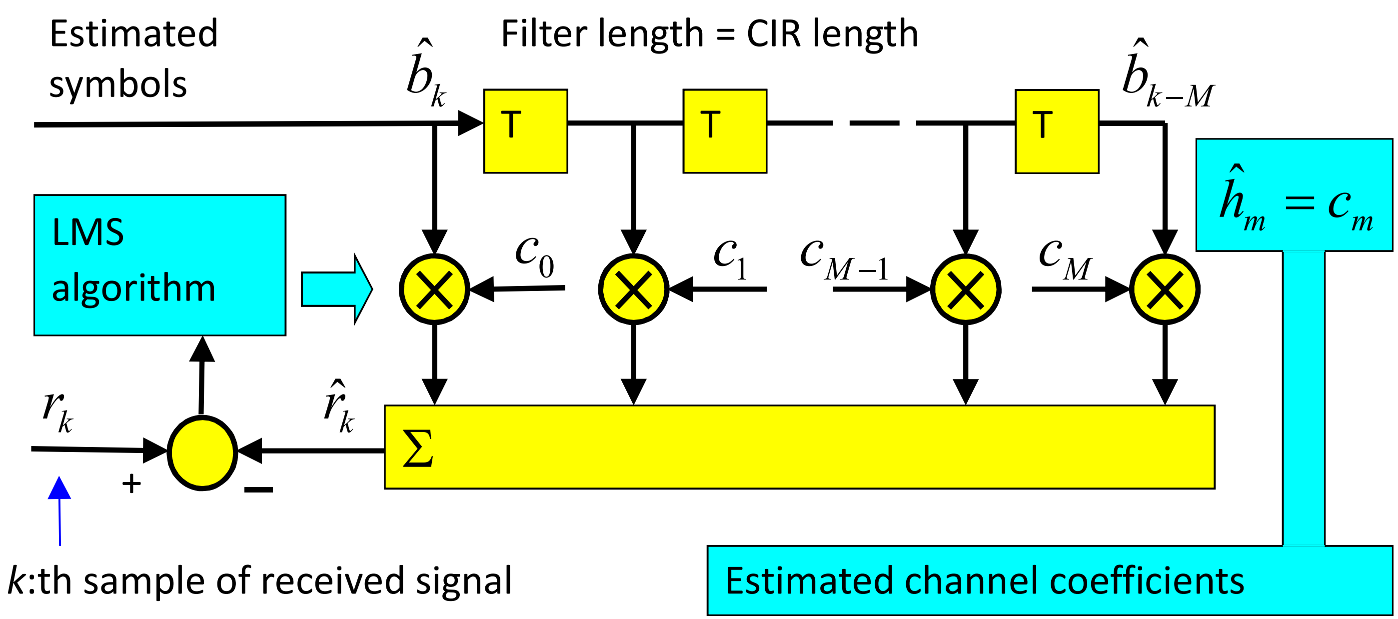
- By calculation from FFF coefficients and channel coefficients (we achieve exact ISI cancellation in this way, but channel estimation is necessary):

$$q_n = -\sum_{m=-M}^{M} c_m\, h_{n-m} \qquad n = 1, 2, \ldots, Q$$

Proakis, Ed.3, Section 10-3-1

Channel estimation circuit

Proakis, Ed.3, Section 11-3



Channel estimation circuit (cont.)

1. Acquisition phase

- Uses “training sequence”
- Symbols are known at receiver, $\hat{b}_k = b_k$.

2. Tracking phase

- Uses estimated symbols (decision directed mode)
- Symbol estimates are obtained from the decision circuit (note the delay in the feedback loop!)
- Since the estimation circuit is adaptive, time-varying channel coefficients can be tracked to some extent.

Alternatively: blind estimation (no training sequence)

Channel estimation circuit in receiver

Mandatory for MLSE-VA, optional for DFE