Lecture 7

Equalizers

LUND UNIVERSITY

1

Contents

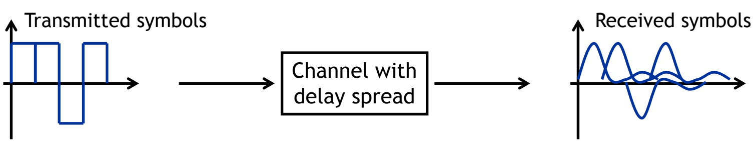
LUND UNIVERSITY

INTER-SYMBOL INTERFERENCE

LUND UNIVERSITY

Inter-symbol interference Background

Even if we have designed the basis pulses of our modulation to be interference free in time, i.e. no leakage of energy between consecutive symbols, multi-path propagation in our channel will cause a delay-spread and **inter-symbol interference (ISI)**.

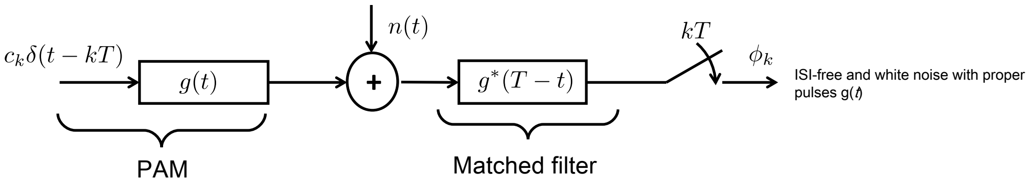


ISI will degrade performance of our receiver, unless mitigated by some mechanism. This mechanism is called an **equalizer**.

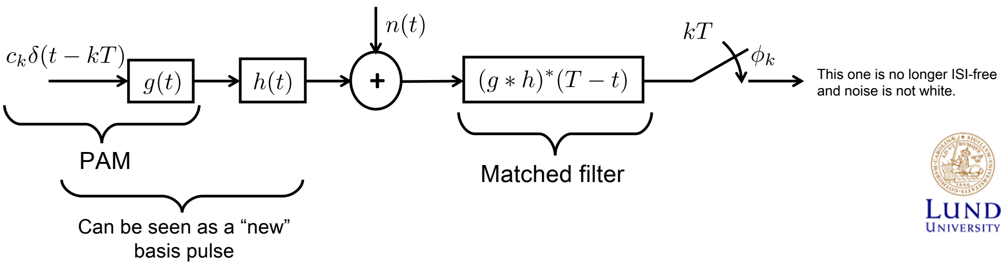
LUND UNIVERSITY

Including a channel impulse response

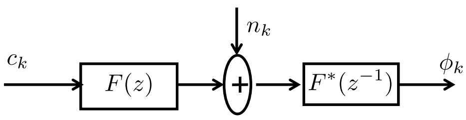
What we have used so far (PAM and optimal receiver):



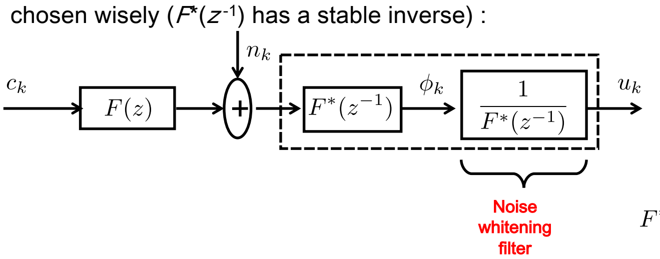
Including a channel impulse response $h(t)$:



We can create a discrete time equivalent of the "new" system:

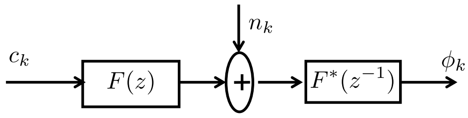


where we can say that $F(z)$ represent the basis pulse and channel, while $F^*(z^{-1})$ represent the matched filter.
We can now achieve white noise quite easily, if (the not unique) $F(z)$ is chosen wisely ($F^*(z^{-1})$ has a stable inverse) :



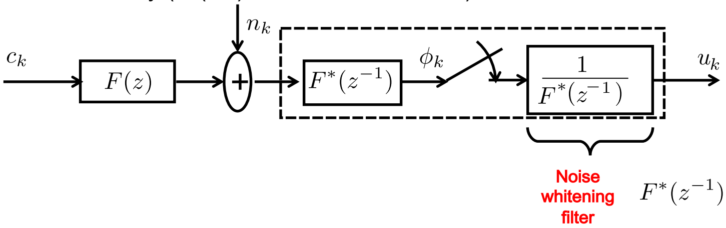
$$F^*(z^{-1})\frac{1}{F^*(z^{-1})} = 1$$

We can create a discrete time equivalent of the "new" system:



where we can say that $F(z)$ represent the basis pulse and channel, while $F^*(z^{-1})$ represent the matched filter.

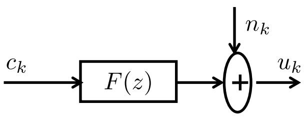
We can now achieve white noise quite easily, if (the not unique) $F(z)$ is chosen wisely ($F^*(z^{-1})$ has a stable inverse) :



$$F^*(z^{-1})\frac{1}{F^*(z^{-1})} = 1$$

Discrete-time channel model

With the application of a noise-whitening filter, we arrive at a discrete-time model



where we have ISI and white additive noise, in the form

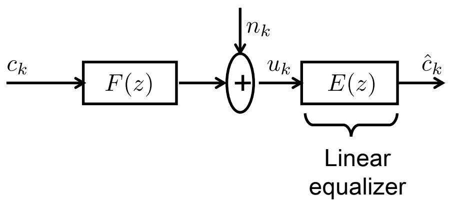
$$u_k = \sum_{j=0}^{L} f_j c_{k-j} + n_k$$

The coefficients f_j represent the causal impulse response of the discrete-time equivalent of the channel $F(z)$, with an ISI that extends over L symbols.

LINEAR EQUALIZER

Linear equalizer principle

The principle of a linear equalizer is very simple: Apply a filter *E*(*z*) at the receiver, mitigating the effect of ISI:



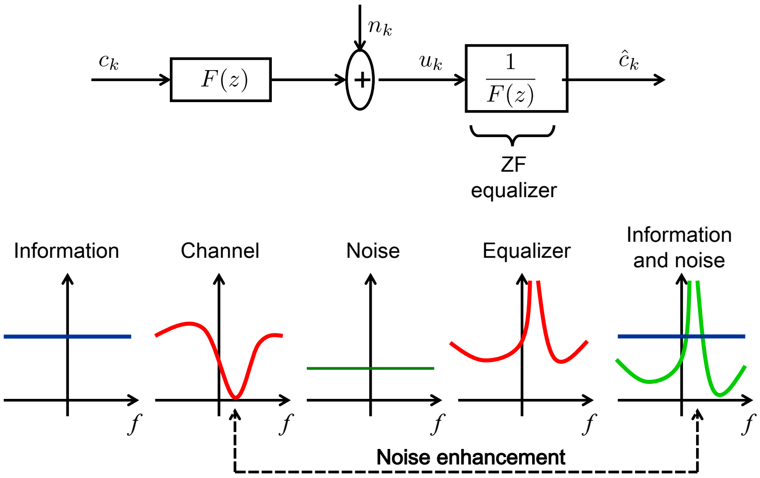
Now we have two different strategies:

1) Design *E*(*z*) so that the ISI is totally removed — Zero-forcing

2) Design *E*(*z*) so that we minimize the mean squared error $E\{|\varepsilon_k|^2\} = E\{|c_k - \hat{c}_k|^2\}$ — MSE

Zero-forcing equalizer

The **zero-forcing equalizer** is designed to remove the ISI completely



Zero-forcing equalizer

A serious problem with the zero-forcing equalizer is the **noise enhancement**, which can result in infinite noise power spectral densities after the equalizer.

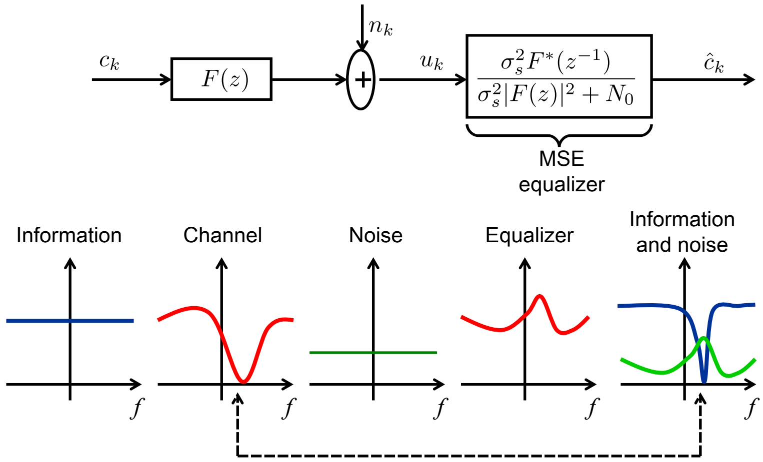
The noise is enhanced (amplified) at frequencies where the channel has a high attenuation.

Another, related, problem is that the resulting noise is colored, which makes an optimal detector quite complicated.

By applying the minimum mean squared-error criterion instead, we can at least remove some of these unwanted effects.

Mean Square Error equalizer

The **MSE equalizer** is designed to minimize the error variance



MSE equalizer

The **MSE equalizer** removes the most problematic noise enhancements as compared to the ZF equalizer. The noise power spectral density cannot go to infinity any more.

This improvement from a noise perspective comes at the cost of not totally removing the ISI.

The noise is still colored after the MSE equalizer which, in combination with the residual ISI, makes an optimal detector quite complicated.

DECISION-FEEDBACK
EQUALIZER

LUND UNIVERSITY

Decision-feedback equalizer principle

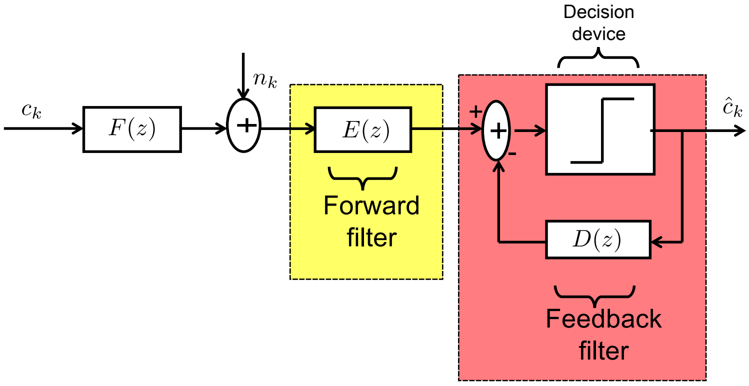
We have seen that taking care of the ISI using only a linear filter will cause (sometimes severe) noise coloring.

A slightly more sophisticated approach is to subtract the interference caused by already detected data (symbols).

This principle of detecting symbols and using feedback to remove the ISI they cause (before detecting the next symbol), is called **decision-feedback equalization** (DFE).

LUND UNIVERSITY

Decision-feedback equalizer

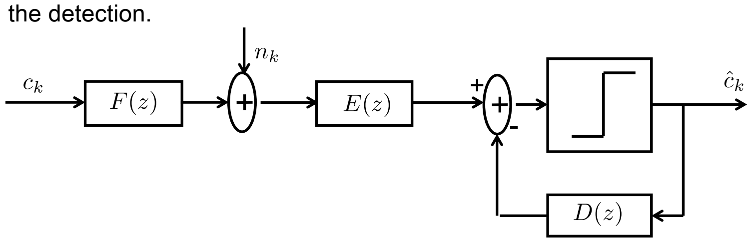


This part shapes the signal to work well with the decision feedback.

This part removes ISI on **"future"** symbols from the currently detected symbol.

Decision-feedback equalizer
Zero-forcing DFE

In the design of a ZF-DFE, we want to completely remove all ISI before the detection.



This enforces a relation between the $E(z)$ and $D(z)$, which is (we assume that we make *correct* decisions!)

$$F(z)E(z) - D(z) = 1$$

As soon as we have chosen $E(z)$, we can determine $D(z)$. (See textbook for details!)

Decision-feedback equalizer
Zero-forcing DFE

Like in the linear ZF equalizer, forcing the ISI to zero before the decision device of the DFE will cause noise enhancement.

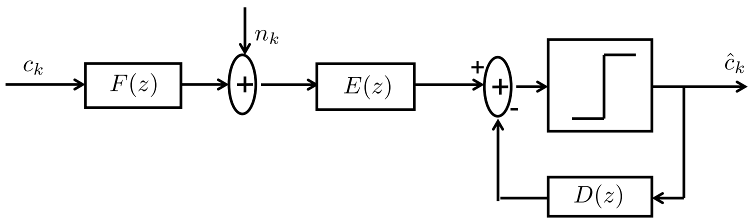
Noise enhancement can lead to high probabilities for making the wrong decisions ... which in turn can cause error propagation, since we may add ISI instead of removing it in the decision-feedback loop.

Due to the noise color, an optimal decision device is quite complex and causes a delay that we cannot afford, since we need them immediately in the feedback loop.

LUND UNIVERSITY

MSE-DFE
Mean Squared Error – Decision Feedback Equalizer

To limit noise enhancement problems, we can concentrate on minimizing mean squared-error (MSE) before the decision device instead of totally removing the ISI.



The overall strategy for minimizing the MSE is the same as for the linear MSE equalizer (again assuming that we make correct decisions). (See textbook for details!)

LUND UNIVERSITY

Decision-feedback equalizer MSE-DFE

By concentrating on minimal MSE before the detector, we can reduce the noise enhancements in the MSE-DFE, as compared to the ZF-DFE.

The performance of the MSE-DFE equalizer is (in most cases) better than the previous equalizers ... but we still have the error propagation problem that can occur if we make an incorrect decision.

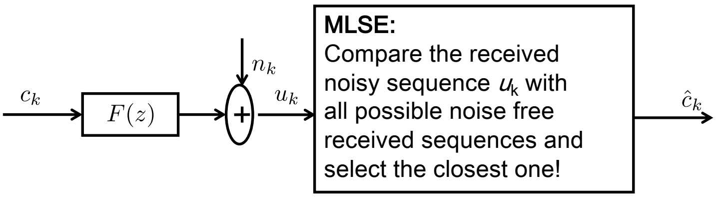
LUND UNIVERSITY

MAXIMUM-LIKELIHOOD SEQUENCE ESTIMATION

LUND UNIVERSITY

Maximum-likelihood sequence estimation

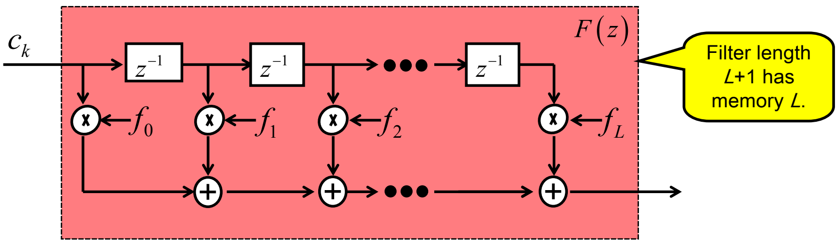
The optimal equalizer, in the sense that it with the highest probability correctly detects the transmitted sequence is the **maximum-likelihood sequence estimator (MLSE)**.



For sequences of length N bits, this requires comparison with 2^N different noise free sequences.

Maximum-likelihood sequence estimation

The discrete-time channel $F(z)$ is very similar to the convolution encoder discussed during Lecture 7 (but with here complex input/output and rate 1):



We can build a trellis and use the **Viterbi algorithm** to efficiently calculate the best path!

Since we know the L+1 tap impulse response f_j, j = 0, 1, ... , L, of the channel, the receiver can, given a sequence of symbols $\{c_m\}$, create the corresponding "noise free (NF) signal alternative" as

$$u_m^{\mathrm{NF}} = \sum_{j=0}^{L} f_j c_{m-j}$$

where NF denotes Noise Free.

The squared Euclidean distance (optimal for white Gaussian noise) to the received sequence $\{u_m\}$ is

$$d^2\left(\{u_m\},\{u_m^{\mathrm{NF}}\}\right) = \sum_m \left|u_m - u_m^{\mathrm{NF}}\right|^2 = \sum_m \left|u_m - \sum_{j=0}^{L} f_j c_{m-j}\right|^2$$

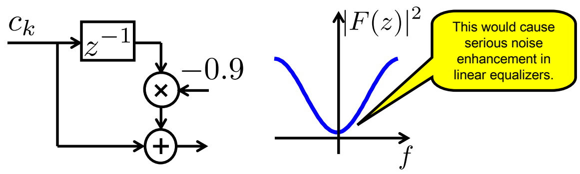
The MLSE decision is then the sequence of symbols $\{c_m\}$ minimizing this distance

$$\{\hat{c}_m\} = \underset{\{c_m\}}{\operatorname{argmin}} \sum_m \left|u_m - \sum_{j=0}^{L} f_j c_{m-j}\right|^2$$

The Viterbi-equalizer

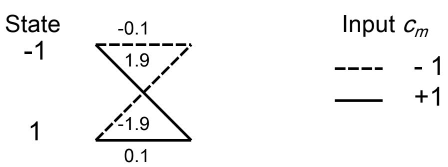
Let's use an example to describe the **Viterbi-equalizer**.

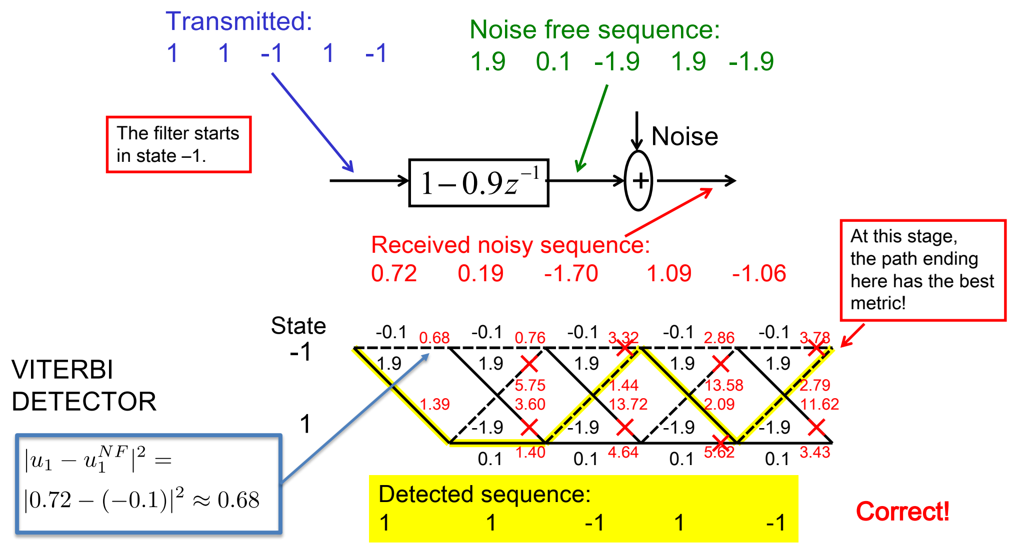
Discrete-time channel:



Further, assume that our symbol alphabet is –1 and +1 (representing the bits 0 and 1, respectively).

The fundamental trellis stage:





The Viterbi-equalizer

The Viterbi-equalizer (detector) is optimal in terms of minimizing the probability of detecting the wrong sequence of symbols.

For transmitted sequences of length N over a length L+1 channel, it reduces the brute-force maximum-likelihood detection complexity of 2^N comparisons to N stages of 2^L comparisons through elimination of trellis paths. L is typically MUCH SMALLER than N.

Even if it reduces the complexity considerably (compared to brute-force ML) it can have a too high complexity for practical implementations if the length of the channel (ISI) is large.

Some final thoughts

We have not covered the topic of channel estimation, which is required since the equalizers need to know the channel. (See textbook for details!)

In practice, a channel estimate will never be exact. This means that equalizers in reality are never optimal in that sense.

The channel estimation problem becomes more problematic in a fading environment, where the channel constantly changes. This requires good channel estimators that can follow the changes of the channel so that the equalizer can be updated continuously. This can be a very demanding task, requireing high processing power and special training sequences transmitted that allow the channel to be estimated.

In GSM there is a known training sequence transmitted in every burst, which is used to estimate the channel so that a Viterbi-equalizer can be used to remove ISI.

LUND UNIVERSITY

LUND UNIVERSITY