EITN15, PWC part 1
Lecture 4: OFDM

Fredrik Rusek, Lund University

April 2012

Outline

- ISI channels
- Some math properties
- Equalization strategies of ISI channels (will be computationally complex)
- OFDM
- Digital – analog conversion in OFDM

Next lecture: advanced synchronization in OFDM

Matched filtering

Transmitted complex baseband signal equals

$$y^c(t) = \sum_k a_k p(t - kT_s)$$

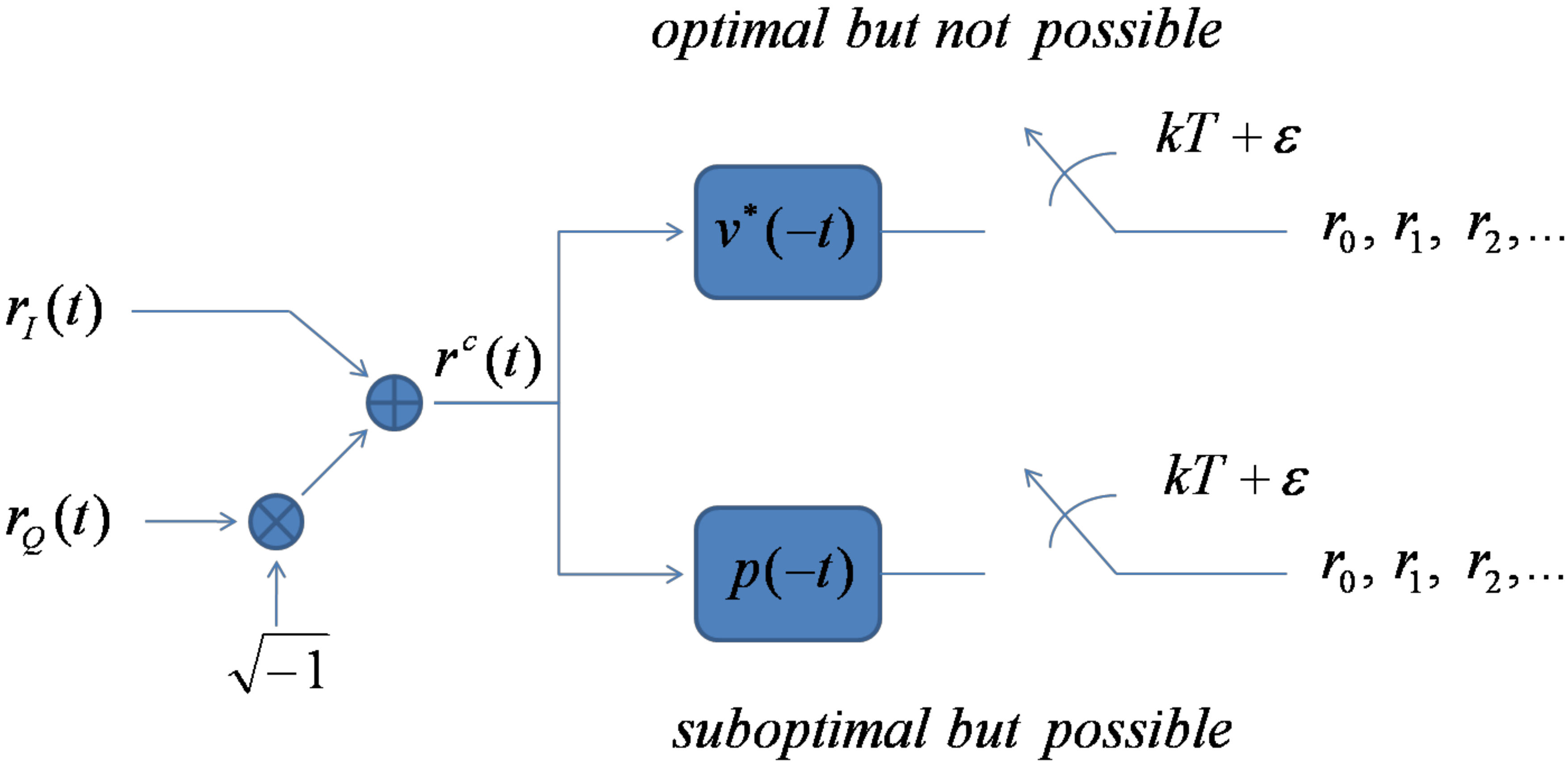
Where {a} are complex data symbols and p(t) any pulse shape.

Received complex baseband signal equals

$$r^c(t) = \sum_k a_k v(t - kT_s) + n^c(t) \text{ where } v(t) = p(t) \star h^c(t)$$

What is the optimal receiver processing?
We know that it is ”matched filtering”, but to what whould we match the filter?

Matched filtering



New problems emerge…..we don't know where to sample, this we model through the offset ε

ISI

Let $z(t)$ denote the matched filter. The signal part of each sample r_k equals

$$\begin{aligned}
r_k &= \left.\int_{-\infty}^{\infty} r^c(\tau) z(t-\tau)\mathrm{d}\tau\right|_{t=kT+\epsilon} \\
&= \left.\int_{-\infty}^{\infty} r^c(\tau) p(\tau-t)\mathrm{d}\tau\right|_{t=kT+\epsilon} \\
&= \int_{-\infty}^{\infty} \left(\sum_{\ell} a_\ell v(\tau-\ell T)\right) p(\tau-kT-\epsilon)\mathrm{d}\tau \\
&= \sum_{\ell} a_\ell \int_{-\infty}^{\infty} v(\tau-\ell T) p(\tau-kT-\epsilon)\mathrm{d}\tau \\
&= \sum_{\ell} a_\ell \int_{-\infty}^{\infty} v(\tau) p(\tau+(\ell-k)T-\epsilon)\mathrm{d}\tau \\
&= \sum_{\ell} a_\ell h_{k-\ell},
\end{aligned}$$

where

$$h_n \triangleq \int_{-\infty}^{\infty} v(\tau) p(\tau-nT-\epsilon)\mathrm{d}\tau.$$

Preliminaries from Math

Definition 1

A matrix U is said to be unitary if $UU^* = U^*U = I$

Property 1

A unitary matrix does not change the magnitude of a vector:

$$\|\mathrm{x}\|^2 = \sum_k |x_k|^2 = \|U\,\mathrm{x}\|^2$$

Property 2

Let X and Y be two random variables with variance V(X) and V(Y). Then, the random variable Z=aX+bY has variance

$$V(Z) = a^2 V(X) + b^2 V(Y)$$

Preliminaries from Math

Property 3

Let $\mathrm{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_N \end{bmatrix}^T$ be a vector of Gaussian random variables, i.e.

$$x_k \in N(0, \sigma^2)$$

$$\mathrm{x} \in N(0, \sigma^2 I)$$

Then, if we multiply x with a matrix H, $\mathrm{y} = H\,\mathrm{x}$

we get that y is Gaussian distributed with correlation matrix H*H, i.e.

$$\mathrm{y} \in N(0, \sigma^2 H^* H)$$

If H is unitary, then H*H=I, and it follows that y and x have the same distributions

Preliminaries from Math

Singular value decompostion (SVD)

Any matrix H (dimension KxL) can be decomposed as

$$H = U\Sigma V^*$$

Where U is KxK unitary, V is LxL unitary, and Σ is a KxL matrix with the singular values along the main diagonal and zeros elsewhere, i.e.

$$\Sigma_{kk} = \lambda_k \quad and \quad \Sigma_{kl} = 0, k \neq l$$

The SVD is similar to matrix diagonalization, but works for non-square matrices.

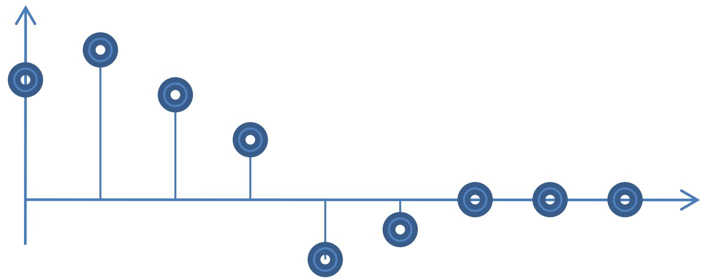
ISI channels

After pulse shaping and carrier modulation at the TX and carrier demodulation, matched filtering and sampling at the Rx, the channel model can be described through

$$y = h * a + n$$

The data symbols are denoted a, h denotes the channel impulse response, and n denotes WGN. All variables are here complex valued.

For example, h could take the form (a real valued h is assumed)



ISI channels

How should we estimate a from y?

Since there is memory is the signal y, it can be represented with a trellis and the Viterbi algorithm can be applied.

How many states are there in the Viterbi algorithm?

This depends on the length of h but also on the modulation format used.

Definition 2

The memory L_{ISI} of the ISI channel is said to be the length of the channel minus 1.

Property 4

Assume that the data symbols are belongs to a M-QAM constellation.

Then the number of states in the Viterbi is $M^{L_{ISI}}$

ISI channels

This number quickly grows large and the Viterbi cannot be used. For the channel given on slide 5 and with QPSK we get

$$L_{ISI} = 5 \qquad M = 4 \qquad \#states = 4^5 = 2^{10} = 1024$$

In high speed real-time applications this workload is prohibitive.

What can be done?

ISI Channels - Zero-forcing equalization

Basic idea: A convolution is a linear operation and linear operations have inverses. Seek to find an impulse response c so that

$$\mathrm{c} * \mathrm{h} = \delta \qquad (1)$$

Then we would get

$$\mathrm{r} = \mathrm{c} * \mathrm{y} = (\mathrm{c} * \mathrm{h}) * \mathrm{a} + \mathrm{c} * \mathrm{n} = \mathrm{a} + \mathrm{w}$$

Note that there is no ISI in r, which allow trivial estimation of a. How should we find a filter c so that (1) is fulfilled? We have

$$(\mathrm{c} * \mathrm{h})[n] = \sum_{l} c_l h_{n-l} = \delta_n$$

ISI Channels - Zero-forcing equalization

For n=0 this yields

$$n = 0: \quad h_0 c_0 = 1 \quad \Rightarrow \quad c_0 = 1/h_0$$

For n=1 we get

$$n = 1: \quad h_0 c_1 + h_1 c_0 = 0 \quad \Rightarrow \quad c_1 = -h_1 / h_0^2$$

This can be repeated for n=2,3,4,………

ZFE – noise enhancement

Consider a sample of the filtered noise w. It equals

$$w_k = \sum_l c_l n_{n-l}$$

From Property 2 we get that each sample has variance

$$V(w_k) = \sum_l c_l^2 V(n_{n-l}) = \sum_l c_l^2 N_0$$

Where it has been assumed that the variance of each noise sample n_k has variance N_0 (each real dimension of the noise has half the variance).

Hence, the noise gets amplified due to the filtering. In general, the sum $\sum_l c_l^2$ is much larger than 1. This situation is named *noise enhancement*.

ISI Channels

Summary

ISI channels appear frequently in practice.

The optimal decoder is the Viterbi algorithm, but it fails due to complexity Issues

The simple ZF equlizer fails due to noise enhancement

There are intermediate schemes as well (MMSE, DFE, etc)

OFDM

Main benefit: Converts the ISI channel into a set of independent parallel
Channels.

This implies optimal detection with trivial complexity!

Matrix representation of convolution

The convolution $\mathrm{y} = \mathrm{h} * \mathrm{a} + \mathrm{n}$

can be represented as $Y = HA + N$

where $Y = \left[y_0, y_1, \ldots, y_{N+L_{ISI}-2} \right]^T$

$$N = \left[n_0, n_1, \ldots, n_{N+L_{ISI}-2} \right]^T$$

$$A = \left[a_0, a_1, \ldots, a_{N-1} \right]^T$$

There are N channel inputs and $N + L_{ISI} - 1$ channel outputs.

The size of H is $\left(N + L_{ISI} - 1 \right) \times N$

$$H = \begin{bmatrix}
h_0 & 0 & 0 & & & 0 \\
h_1 & h_0 & 0 & & & \\
\vdots & h_1 & h_0 & & & \\
\vdots & & h_1 & \ddots & & \\
\vdots & & & \ddots & & \\
h_{L_{ISI}-1} & & & & & \\
0 & h_{L_{ISI}-1} & & & & \\
0 & 0 & \ddots & \ddots & 0 & \\
 & 0 & \ddots & \ddots & h_0 & 0 \\
 & & & & h_1 & h_0 \\
 & & & & & h_1 \\
 & & & & & \vdots \\
 & & & \ddots & & \vdots \\
 & & & 0 & h_{L_{ISI}-1} & \vdots \\
0 & & & 0 & 0 & h_{L_{ISI}-1}
\end{bmatrix}$$

Using the SVD

This can be expressed as $Y = U\Sigma V^* A + N$

The receiver can remove U as

$$R = U^* U\Sigma V^* A + U^* N = \Sigma V^* A + W$$

where W is some new noise vector. From Property 3 it follows that W is statistically distributed exactly as N.

If the transmitter constructs A as $A = VX$
where X are QAM symbols carrying the actual data, then R becomes

$$R = \Sigma V^* VX + W = \Sigma X + W$$

From Property 2 it follows that the energy in X equals that in A.

SVD generates parallel channels

Now recall that Σ is not square.

The *k*th element of R equals $r_k = \Sigma_{kk} a_k + w_k \quad 1 \le k \le N$

THE ISI CHANNEL HAS VANISHED AND OPTIMAL DETECTION CAN BE MADE WITHOUT MEANS OF THE VITERBI ALGORITHM

Major Problems

The approach just described can not work since the Tx does not know the matrix V.

Further, the SVD has cubic complexity in the matrix size. With 100 inputs, around 1000000 operations need to be carried out.

The solution will be to add a cyclic prefix so that U and V are constant for all channel matrices H.

Circulant Matrices

Definition 3

A circulant matrix has the form

i.e., every row is a cyclic shift of the first.

$$T = \begin{bmatrix} t_0 & t_1 & \cdots & \cdots & t_{N-2} & t_{N-1} \\ t_{N-1} & t_0 & t_1 & & & t_{N-2} \\ \vdots & t_{N-1} & t_0 & t_1 & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & \ddots & t_1 \\ t_{N-1} & t_{N-2} & \cdots & \cdots & t_1 & t_0 \end{bmatrix}$$

(Amazing) Property 5

The Eigenvalue decomposition of a circulant matrix is

$$T = F^* \Sigma F$$

where F is the DFT matrix, i.e.

$$f_{kl} = \exp(i2\pi kl / N) / \sqrt{N} \quad 0 \le k, l \le N-1$$

Converting the channel convolution into a cyclic convolution

Property 5 is precisly what is needed in order to make OFDM work.

How can we make use of it?

Consider the toy N=4 case with a 3-tap channel. The channel matrix is

$$H = \begin{bmatrix} h_0 & 0 & 0 & 0 \\ h_1 & h_0 & 0 & 0 \\ h_2 & h_1 & h_0 & 0 \\ 0 & h_2 & h_1 & h_0 \\ 0 & 0 & h_2 & h_1 \\ 0 & 0 & 0 & h_2 \end{bmatrix}$$

which resembles the circulant matrix

$$H_{Circ} = \begin{bmatrix} h_0 & 0 & h_2 & h_1 \\ h_1 & h_0 & 0 & h_2 \\ h_2 & h_1 & h_0 & 0 \\ 0 & h_2 & h_1 & h_0 \end{bmatrix}$$

Converting the channel convolution into a cyclic convolution

Property 5 is precisly what is needed in order to make OFDM work.

How can we make use of it?

Consider the toy N=4 case with a 3-tap channel. The channel matrix is

$$H = \begin{bmatrix} h_0 & 0 & 0 & 0 \\ h_1 & h_0 & 0 & 0 \\ h_2 & h_1 & h_0 & 0 \\ 0 & h_2 & h_1 & h_0 \\ 0 & 0 & h_2 & h_1 \\ 0 & 0 & 0 & h_2 \end{bmatrix}$$

which resembles the circulant matrix

$$H_{Circ} = \begin{bmatrix} h_0 & 0 & h_2 & h_1 \\ h_1 & h_0 & 0 & h_2 \\ h_2 & h_1 & h_0 & 0 \\ 0 & h_2 & h_1 & h_0 \end{bmatrix}$$

It can be seen that the tail of the channel convolution has been cut off and been put in the begining of the output.

Cyclic Prefix

For the previous toy example, if we transmit

$$A = [a_2\ a_3\ a_0\ a_1\ a_2\ a_3]^T$$

This is the cyclic prefix

We will receive 8 output symbols

$$Y = [y_0 \quad y_1 \quad y_2 \quad y_3 \quad y_4 \quad y_5 \quad y_6 \quad y_7]^T$$

Consider the middle 4 outputs, they can be described as

$$\begin{bmatrix} y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} h_0 & 0 & h_2 & h_1 \\ h_1 & h_0 & 0 & h_2 \\ h_2 & h_1 & h_0 & 0 \\ 0 & h_2 & h_1 & h_0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} + \begin{bmatrix} n_2 \\ n_3 \\ n_4 \\ n_5 \end{bmatrix}$$

$$\Leftrightarrow$$

$$Y = H_{Circ} A + N$$

OFDM

We then get

$$Y = F^{*}\Sigma F a + N$$

If a=F*x and r=FY, we get

$$\begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} \lambda_0 & 0 & 0 & 0 \\ 0 & \lambda_1 & 0 & 0 \\ 0 & 0 & \lambda_2 & 0 \\ 0 & 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

In general, for an ISI channel of length L_{ISI} the cyclic prefix length needs to be $L_{ISI}-1$. Since this number cannot be known beforehand, some fairly large number is chosen so that it covers most channels.

OFDM

Property 6

The vector of eigenvalues λ equals the FFT of the first row of the circulant matrix .

Hence, for the circulant representation of convolution, it follows that the eigenvalues are given by the FFT of the channel impulse response.

Consequently, the data symbols X are transmitted on the Frequency response of the channel. This explains the name *Orthogonal Frequency Division Multiplexing*

Orthogonal because there is no ISI and *Frequency division* since the different symbols are transmitted on the frequency response.

OFDM

The structure of the signaling is

Data symbols:

$$X = [\underbrace{x_0 \cdots x_{N-1}}_{\text{symbol 1}} \; \underbrace{x_N \cdots x_{2N-1}}_{\text{symbol 2}} \; \underset{\text{etc etc}}{\cdots\cdots}]^T$$

After IFFT:

$$A = [\underbrace{a_0 \quad \cdots \quad a_{N-1}}_{\text{symbol 1}} \quad \underbrace{a_N \quad \cdots \quad a_{2N-1}}_{\text{symbol 2}} \quad \underset{\text{etc etc}}{\cdots \quad \cdots}]^T$$

With CP:

$$\tilde{A} = [\underbrace{a_{N-Ncp} \cdots a_{N-1}}_{\text{CP1}} \; \underbrace{a_0 \cdots a_{N-1}}_{\text{Symbol 1}} \; \underbrace{a_{2N-Ncp} \cdots a_{2N-1}}_{\text{CP2}} \; \underbrace{a_N \cdots a_{2N-1}}_{\text{Symbol 2}} \cdots\cdots]^T$$

OFDM receiver

Received signal

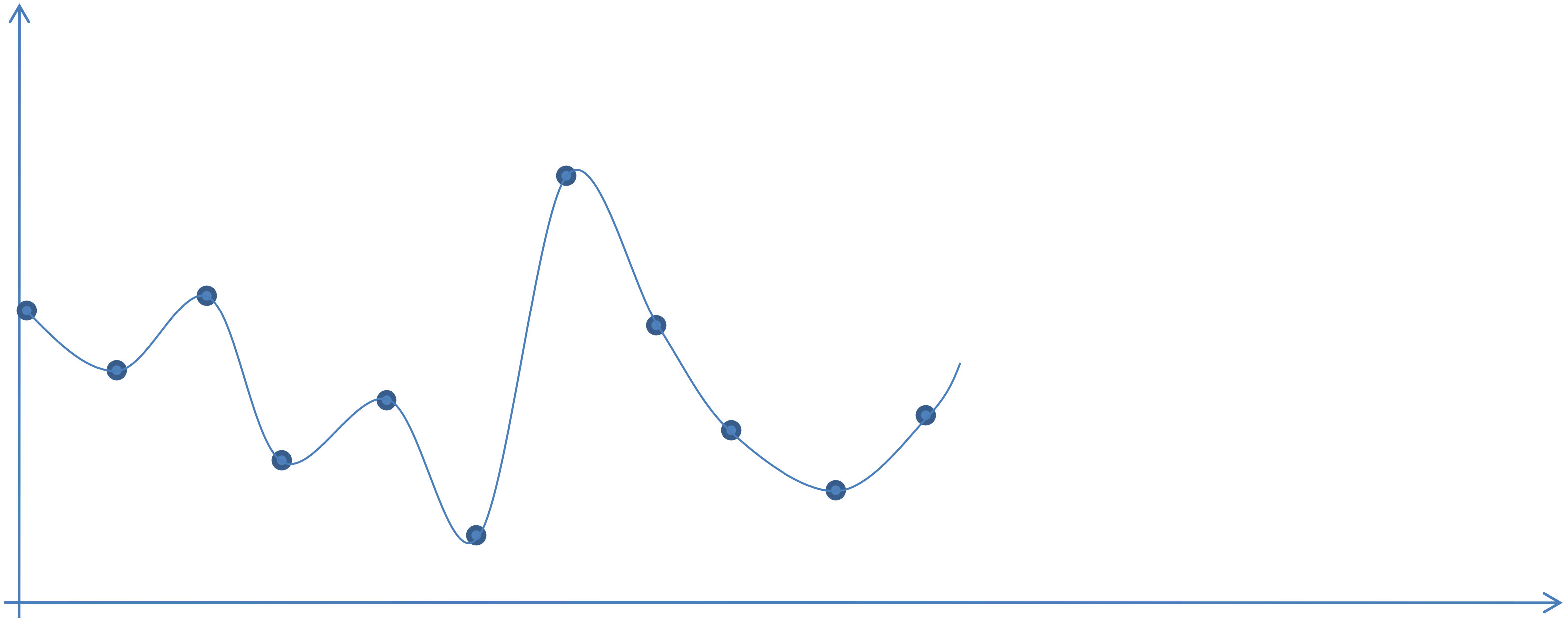
$$R = \Big[\underbrace{r_0 \quad r_1 \quad \cdots \quad r_{Ncp-1}}_{\text{Cyclic prefix: Discard}} \quad \underbrace{r_{Ncp} \quad \cdots \quad r_{Ncp+N-1}}_{\text{Data Send to FFT}} \quad \underbrace{\cdots \quad \cdots \quad \cdots \quad \cdots}_{\text{Cyclic prefix: Discard}}\Big]^T$$

Etc etc

D/A conversion

So far only the discrete-time models have been discussed.

After the IFFT, adding CP, we have a vector a that should
be transmitted. We need to construct a time continuous signal s(t).
(carrier modulation will follow after that)



D/A conversion

So far in the course, we did this as

$$s(t) = \sum a_k p(t - kT)$$

Within the OFDM community, this is not the standard approach.

They do the job by direct D/A conversion

Interlude – the sampling theorem

Recall the sampling theorem. Let a signal s(t) be sampled at the rate $1/T_{samp}$

$$s_d[n] \equiv s(nT_{samp})$$

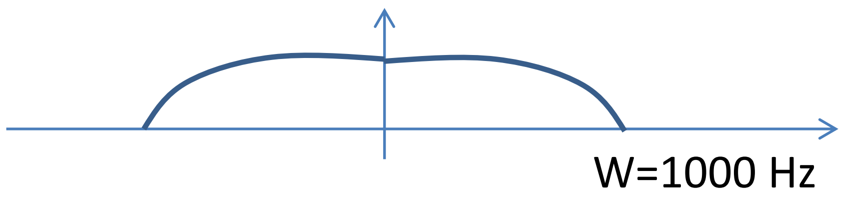
The spectrum $S_d(f)$ equals

$$S_d(f) \propto \sum_{k=-\infty}^{\infty} S\left(\frac{f}{T_{samp}} + \frac{k}{T_{samp}}\right), \qquad -1/2 \leq f \leq 1/2$$

f is *normalized* frequency

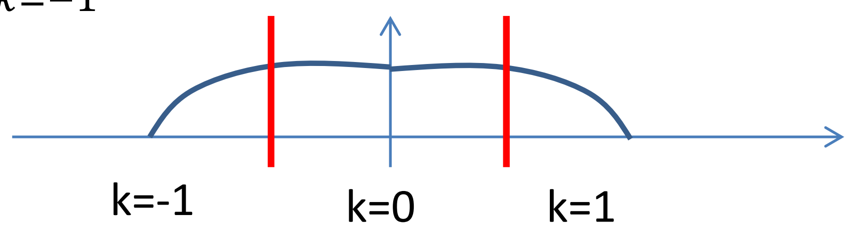
D/A – the sampling theorem

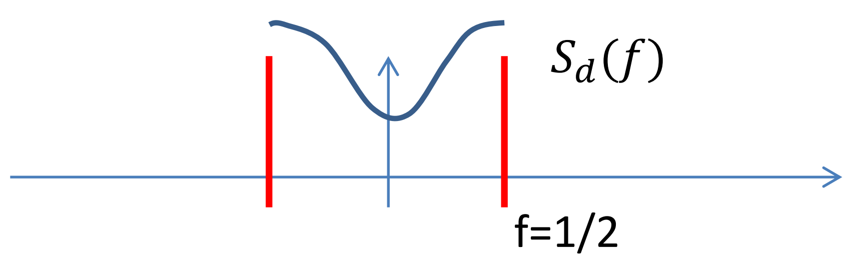
Example:



Sample with $1/T_{samp}$=1000 Hz:

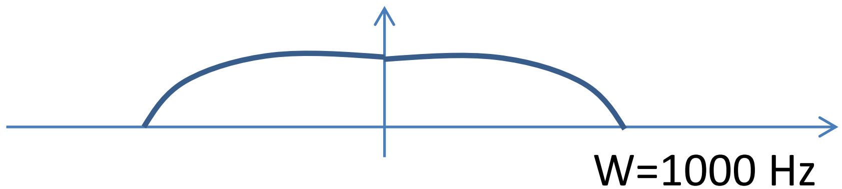
$$S_d(f) \propto \sum_{k=-1}^{1} S(1000f + 1000k) \ , \qquad -1/2 \leq f \leq 1/2$$





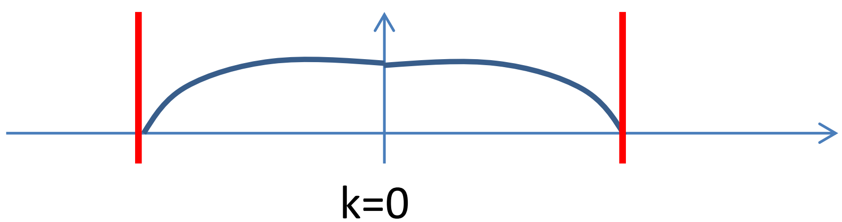
D/A – the sampling theorem

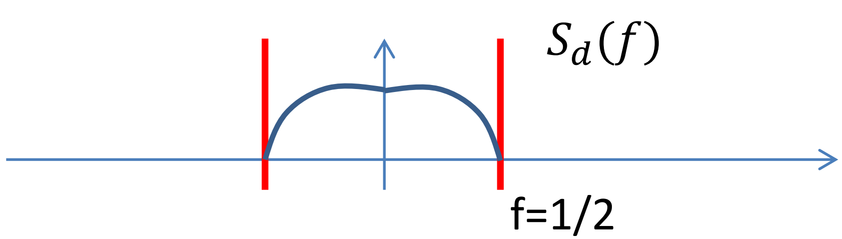
Example:



Sample with $1/T_{samp}$=2000 Hz:

$$S_d(f) \propto S(2000f), \qquad -1/2 \leq f \leq 1/2$$





D/A – the sampling theorem

If the sampling rate satisfies

$$1/T_{samp} \geq 2W$$

there will be no aliasing, and the sampling is loss-less, i.e., the signal s(t) can be perfectly reconstructed from $s_d[n]$. This holds since the two spectrums are Identical.

Note, the sampling theorem was proved by C.E. Shannon in 1948, and not by H. Nyquist (which is a wide spread misunderstanding – especially in Sweden!)

D/A in OFDM

At the transmitter, do the following:

1. Choose the symbol time between two adjacent outputs from the IFFT, denote this by T_{sym}
2. Construct the signal s(t) so that its bandwidth is well localized inside the band $W=[-1/2T_{sym}, 1/2T_{sym}]$

At the receiver, do the following:

1. Set the sampling rate $1/T_{samp}=1/T_{sym}$
2. Filter the received signal with a low-pass filter to cut away anything outside the band $W=[-1/2T_{samp}, 1/2T_{samp}]$
3. Sample with the rate $1/T_{samp}$. The samples are sufficient statistics for optimum detection of the data symbols (since there is no aliasing)

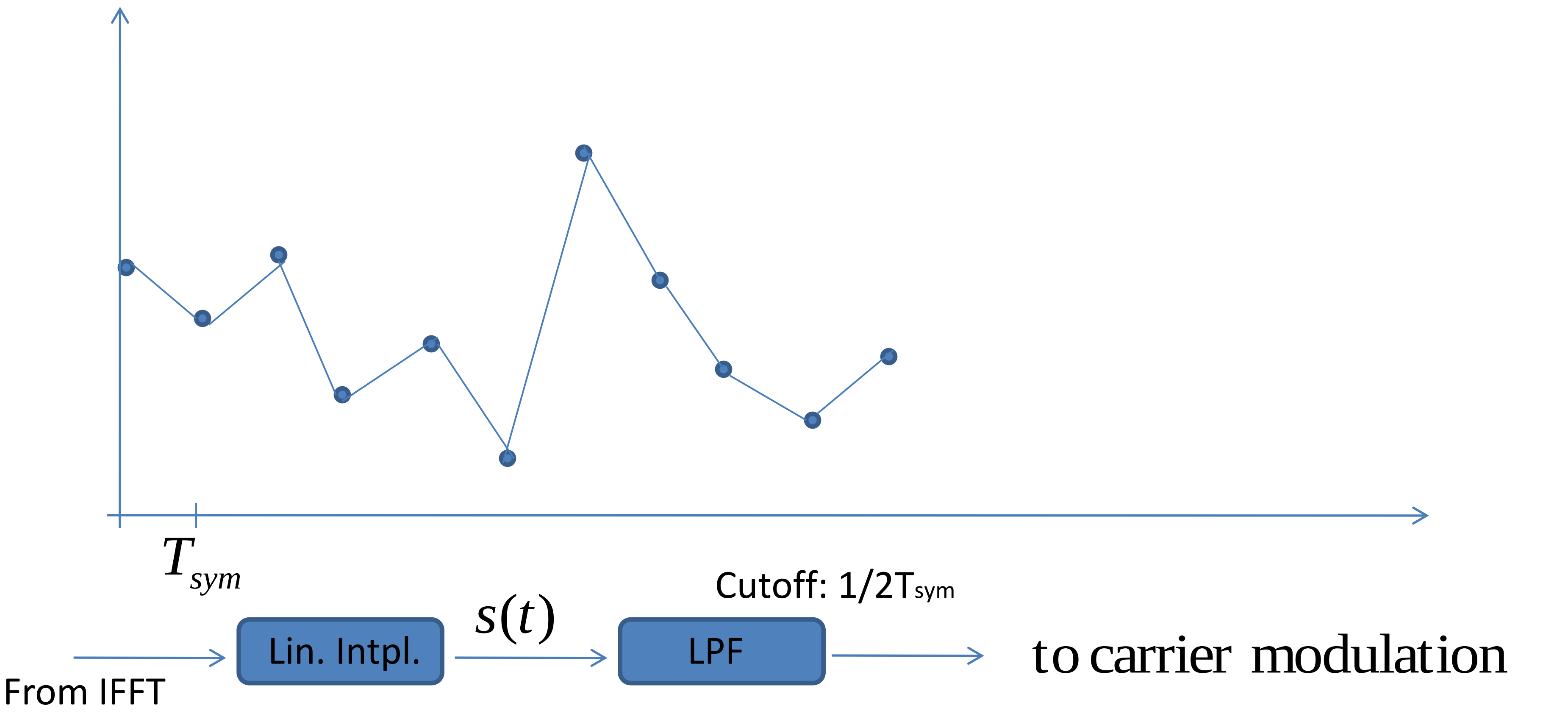
What if the signal s(t) is not well localized inside $W=[-1/2T_{sym}, 1/2T_{sym}]$?

1. Low pass filter the signal at the larger bandwidth
2. Sample with the rate $1/T_{samp}$.

D/A in OFDM

How do we construct s(t) so that it is well localized in the band?

One way is as follows

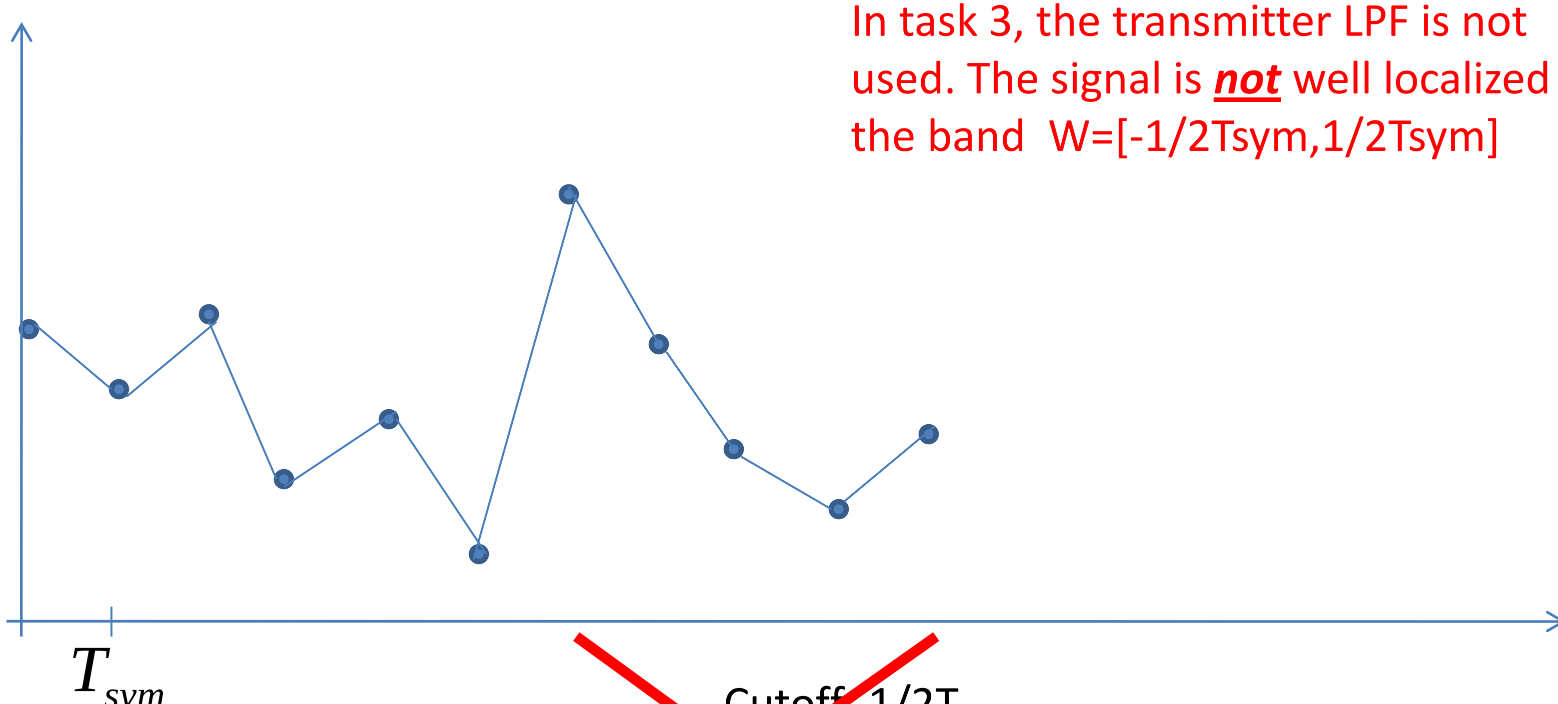


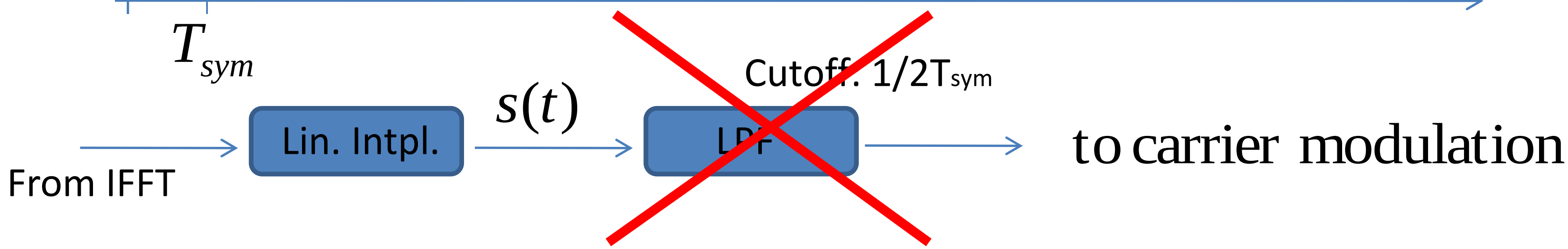
D/A in OFDM

How do we construct s(t) so that it is well localized in the band?

One way is as follows

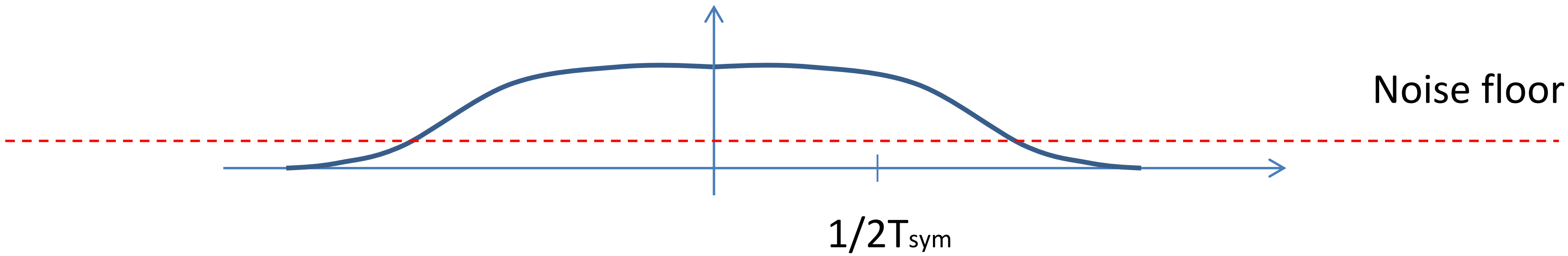
In task 3, the transmitter LPF is not used. The signal is ***<u>not</u>*** well localized to the band W=[-1/2Tsym,1/2Tsym]





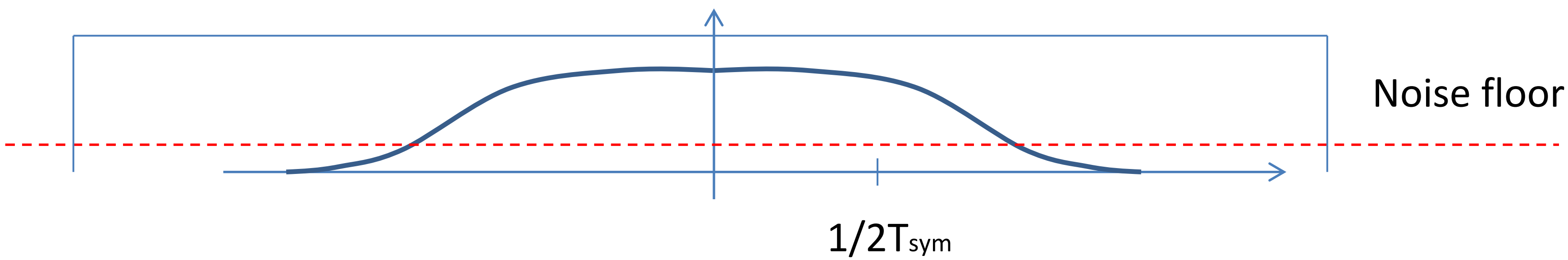
LPF in the task 3

Where to filter the received signal?



LPF in the task 3

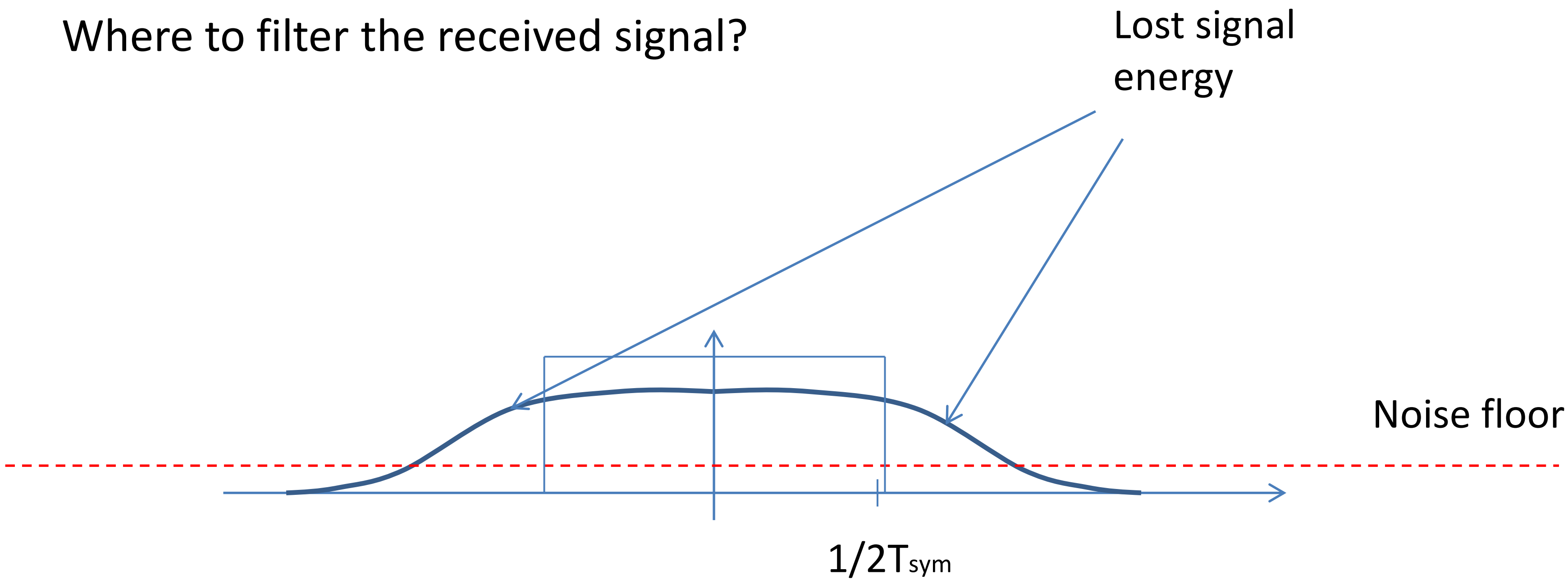
Where to filter the received signal?



Here? No, too much noise will pass the filter.......

LPF in the task 3

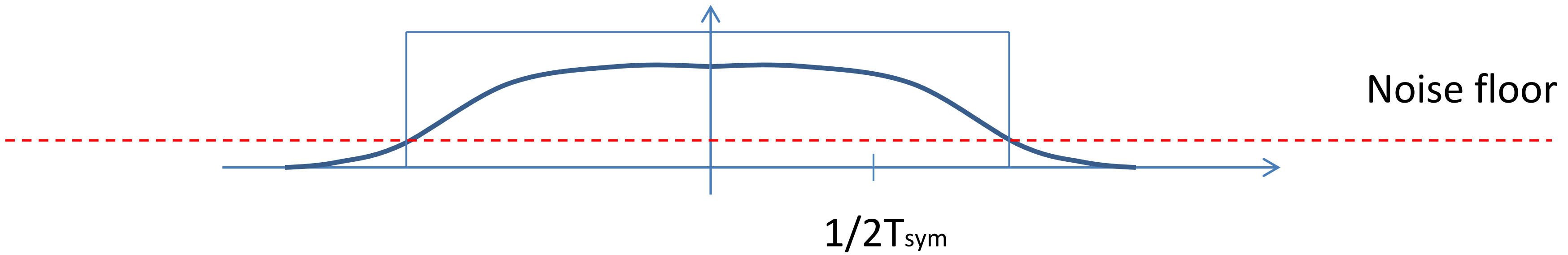
Where to filter the received signal?



Here? No, lots of signal energy is lost, and the noise that hits the discarded signal is not too bad........

LPF in the task 3

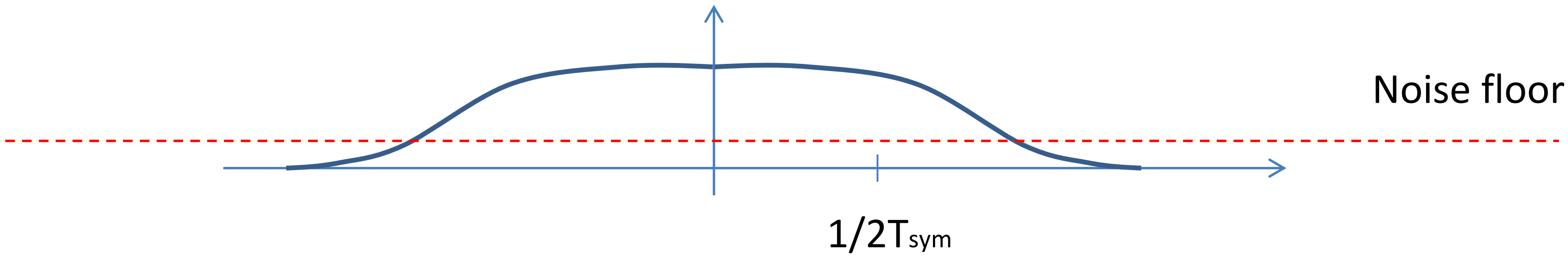
Where to filter the received signal?



Here? Appears to be a good choice!
(It is of course possible to do a rigorous mathematical treatment of the optimal filter point)

LPF in the task 3

Where to filter the received signal?



In task3, the noise floor is unknown, so the filter bandwidth must be found through trial-and-error