EITN21, PWC
OFDM synchronization and Nomenclatura for Convolutional Codes

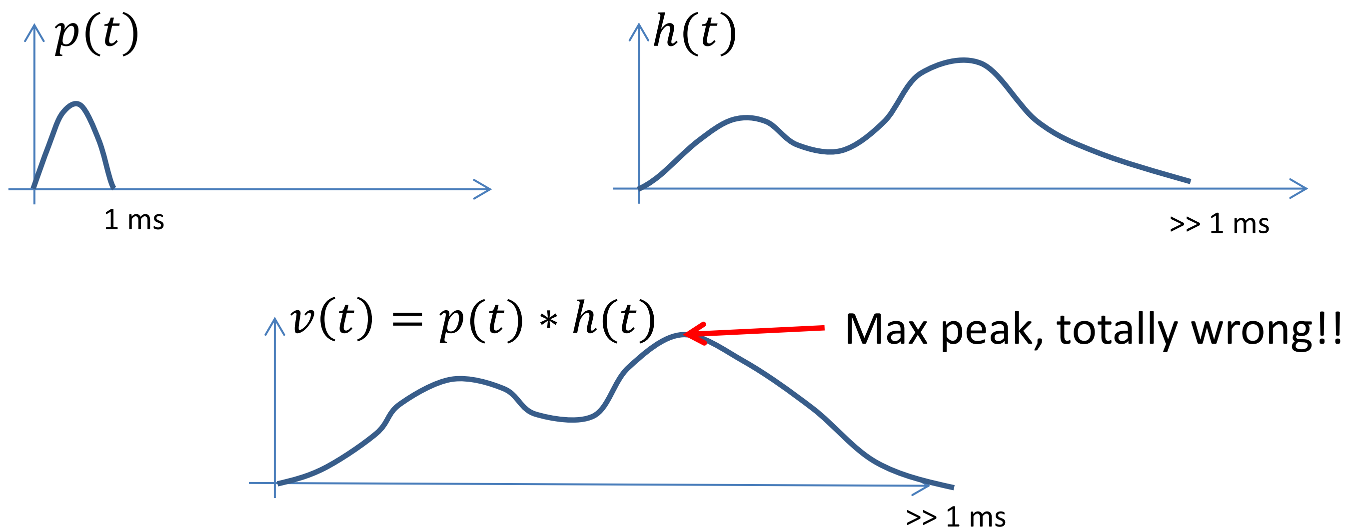
Fredrik Rusek, Lund University

Periodic Pilots

To find the start of the signal, we previously used the ”peak” of the filtered received baseband signal.

If the symbol time is much less than the duration of the channel, this may cause incorrect synchronization

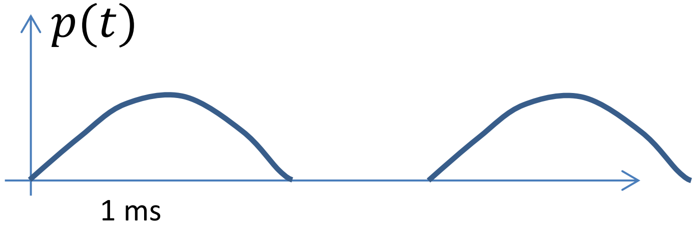
Example:



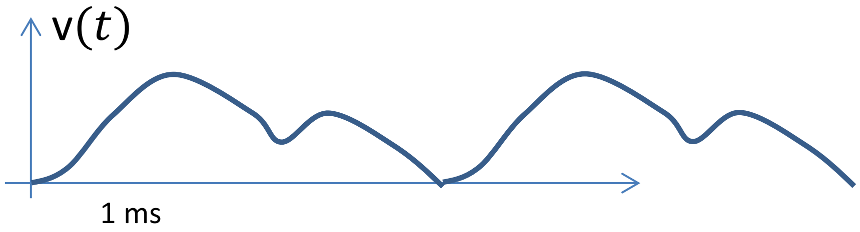
Periodic Pilots

In OFDM, the time interval between two outputs of the IFFT is usually much shorter than the duration of the channel, so ”looking for peaks” is not a clever idea.

Now forget about OFDM for the moment....
Assume that the pilot structure is periodic:



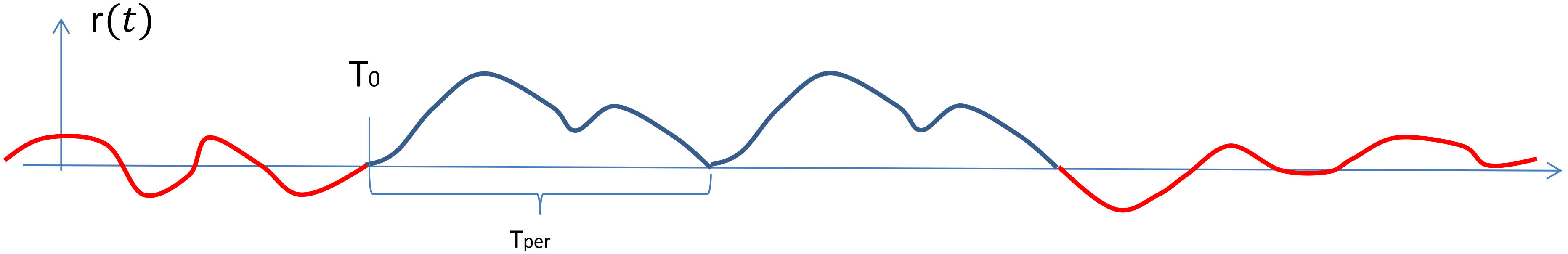
We don’t know the channel, so we don’t know v(t). But we do know that v(t) will also be periodic!



Periodic Pilots

We can look for this correlation in the received signal (shown without noise)
It is assumed for the moment that the data-carrying signal does not contain any correlation.

Task: find T_0

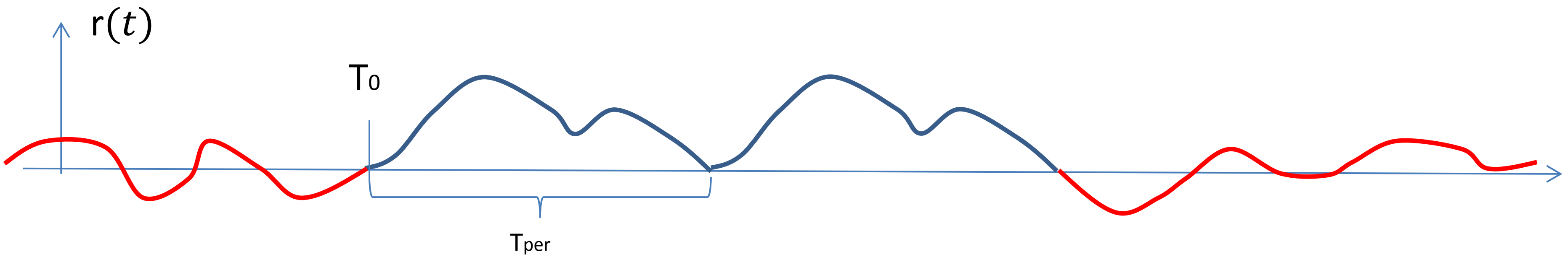


Periodic Pilots

We can look for this correlation in the received signal (shown without noise)

It is assumed for the moment that the data-carrying signal does not contain any correlation.

<u>**Task: find T_0**</u>

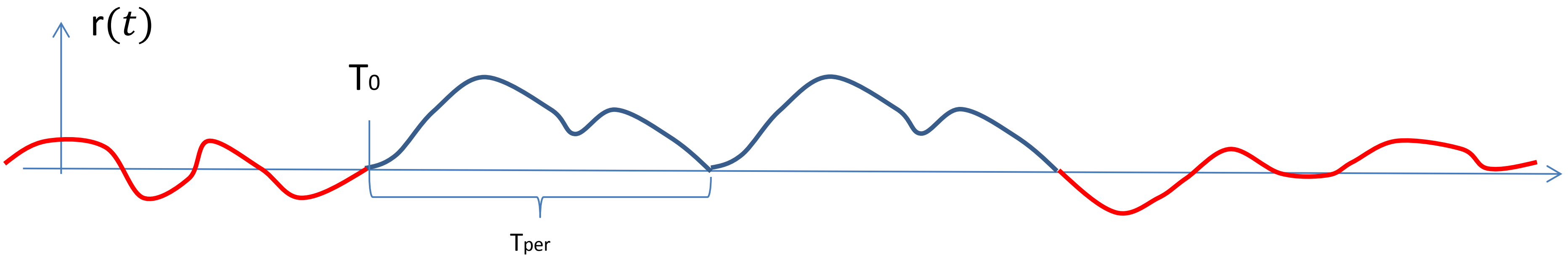


Define $$\mu(t) \equiv \int_0^{T_{per}} r(t+\tau)r^*(t+T_{per}+\tau)d\tau$$

Periodic Pilots

We can look for this correlation in the received signal (shown without noise)
It is assumed for the moment that the data-carrying signal does not contain any correlation.

Task: find T_0

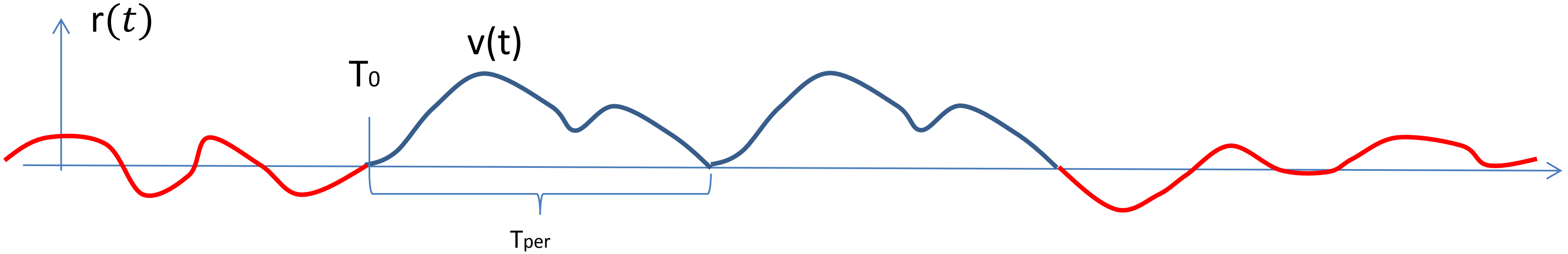


$$\mu(T_0) = \int_0^{T_{per}} r(T_0 + \tau) r^*\left(T_0 + T_{per} + \tau\right) d\tau$$

Periodic Pilots

We can look for this correlation in the received signal (shown without noise)
It is assumed for the moment that the data-carrying signal does not contain any correlation.

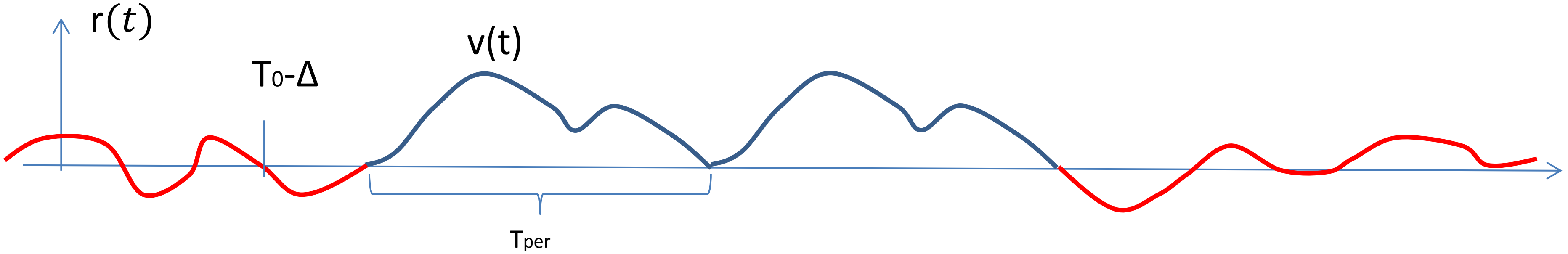
Task: find T_0



$$\mu(T_0) = \int_0^{T_{per}} r(T_0+\tau) r^*\left(T_0+T_{per}+\tau\right) d\tau = \int_0^{T_{per}} |r(T_0+\tau)|^2 d\tau = E_v$$

Periodic Pilots

We can look for this correlation in the received signal (shown without noise)
It is assumed for the moment that the data-carrying signal does not contain any correlation.

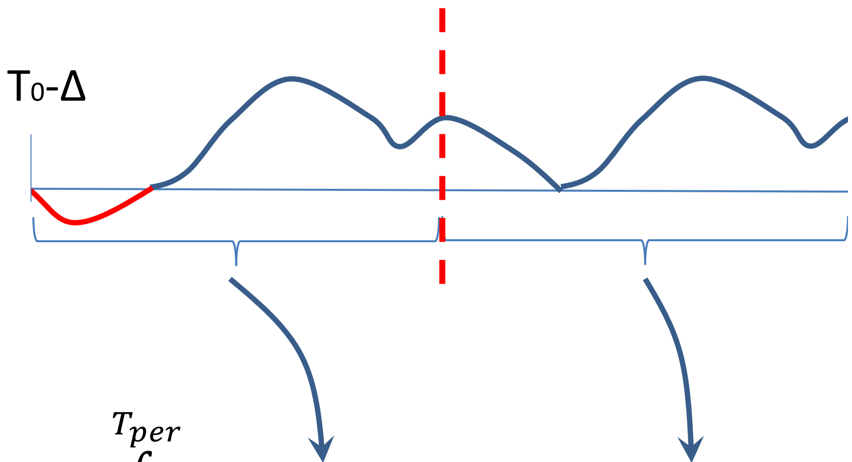


$$\mu(T_0 - \Delta) = \int_0^{T_{per}} r(T_0 - \Delta + \tau)r^*\left(T_0 - \Delta + T_{per} + \tau\right)d\tau$$

Periodic Pilots

We can look for this correlation in the received signal (shown without noise)

It is assumed for the moment that the data-carrying signal does not contain any correlation.



$$\mu(T_0 - \Delta) = \int_0^{T_{per}} r(T_0 - \Delta + \tau) r^*(T_0 - \Delta + T_{per} + \tau) d\tau$$

Which is larger: $\mu(T_0)$ or $\mu(T_0 - \Delta)$?

Periodic Pilots

Normalize by Cauchy-Schwarz:

$$\int f(x)g(x)dx \leq \sqrt{\int |f(x)|^2} \sqrt{\int |g(x)|^2}$$

$$\frac{\int f(x)g(x)dx}{\sqrt{\int |f(x)|^2} \sqrt{\int |g(x)|^2}} \leq 1$$

Equality if g(x)=k f(x), for some constant k

Periodic Pilots

Define

$$\tilde{\mu}(t) \equiv \frac{\int_0^{T_{per}} r(t+\tau) r^*\left(t+T_{per}+\tau\right) d\tau}{\sqrt{\int |r(t+\tau)|^2} \sqrt{\int \left|r(t+T_{per}+\tau)\right|^2}}$$

Now

$$\tilde{\mu}(T_0) = 1$$

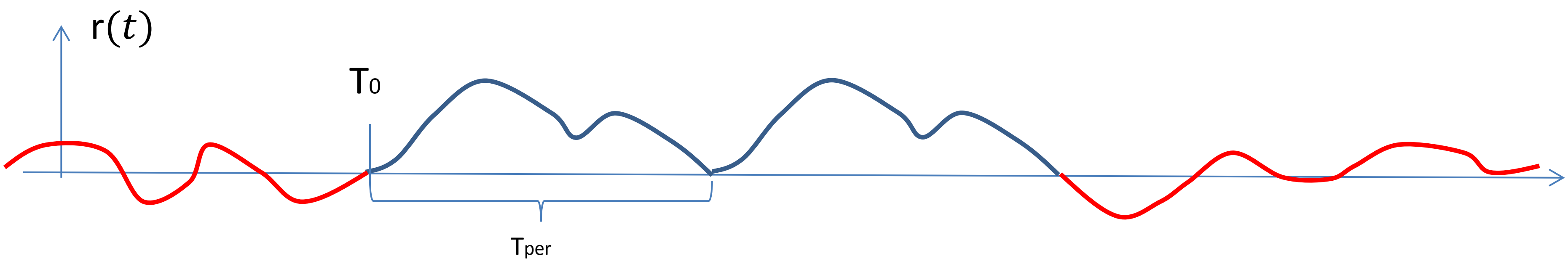
And

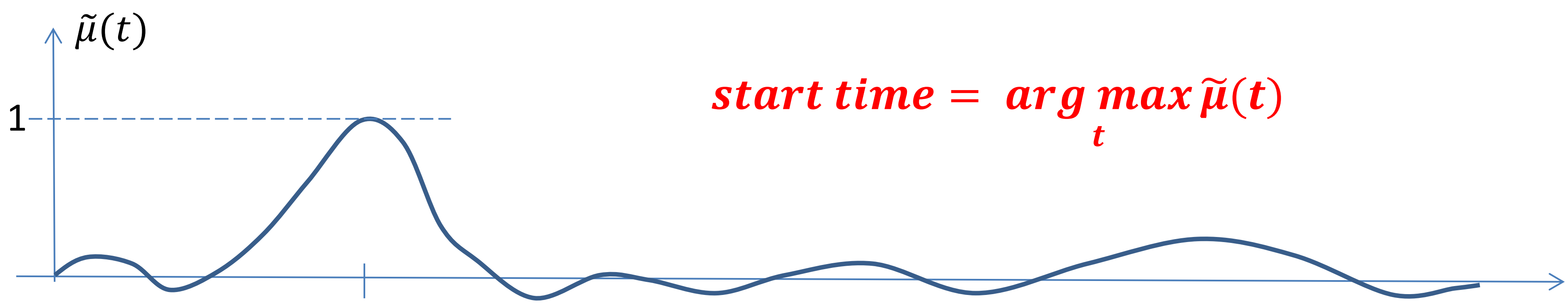
$$\tilde{\mu}(T_0 - \Delta) < 1$$ For any Δ, with (very) high probability

Periodic Pilots

We can look for this correlation in the received signal (shown without noise)
It is assumed for the moment that the data-carrying signal does not contain any correlation.

Task: find T_0



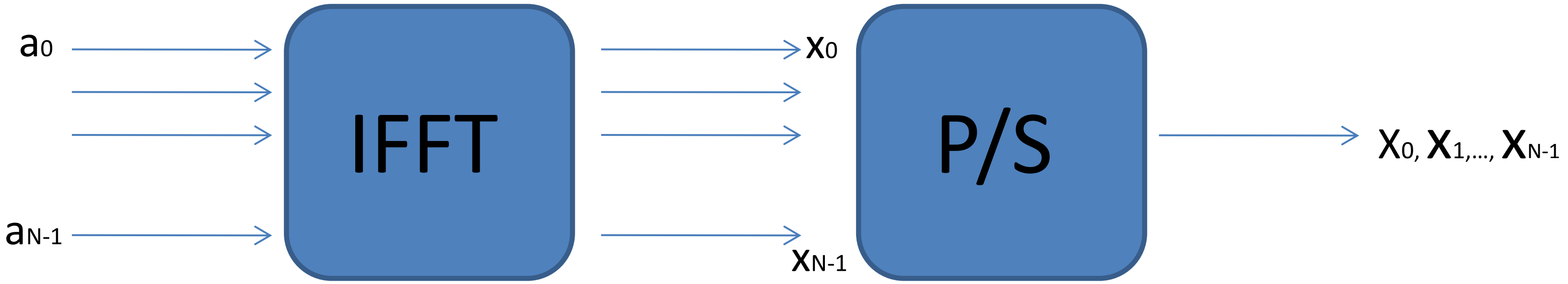


$$start\ time = \arg\max_{t} \tilde{\mu}(t)$$

Periodic pilots in OFDM

How to construct a periodic pilot in OFDM?

Given the input symbols $(a_0, a_1, \ldots, a_{N-1})$, the transmitted symbols $(x_0, x_1, \ldots, x_{N-1})$ are found by applying the IFFT.



Mathematically we have

$$x_k = \sum_{l=0}^{N-1} a_l \exp(-i2\pi lk / N)$$

Periodic pilots in OFDM

Now suppose that every second input symbol equals 0, i.e.,

$$a_{2k+1} = 0, \quad k = 0,1,2,\dots.$$

We get,

$$x_k = \sum_{l=0}^{N-1} a_l \exp(-i2\pi lk / N) = \sum_{even\, l} a_l \exp(-i2\pi lk / N)$$

Now consider $x_{k+N/2}$

$$x_{k+N/2} = \sum_{even\, l} a_l \exp(-i2\pi l(k+N/2)/N) = \sum_{even\, l} a_l \exp(-i2\pi lk / N - il\pi)$$

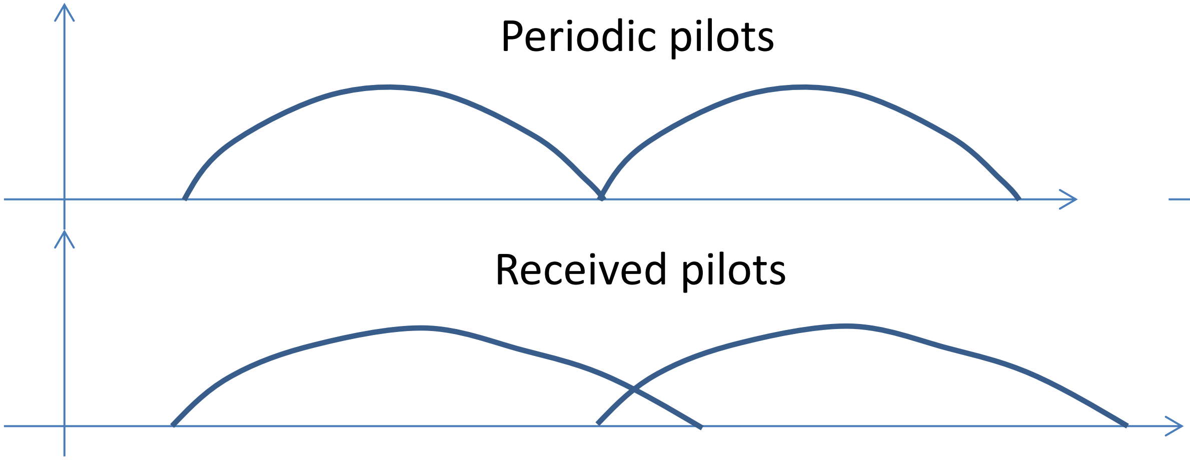
$$= \sum_{even\, l} a_l \exp(-i2\pi lk / N) = x_k$$

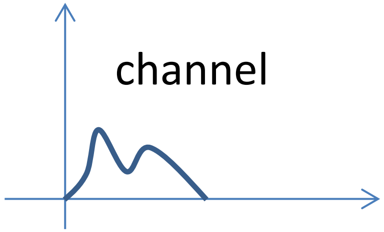
Periodic pilots in OFDM

Hence, the transmitted signal is periodic with period N/2.

In OFDM, the time between two outputs of the IFFT is short in comparison with the Channel.

But, the period of the pilots is N/2 such intervals, which is much longer than the duration of the channel (which is the key feature of OFDM)



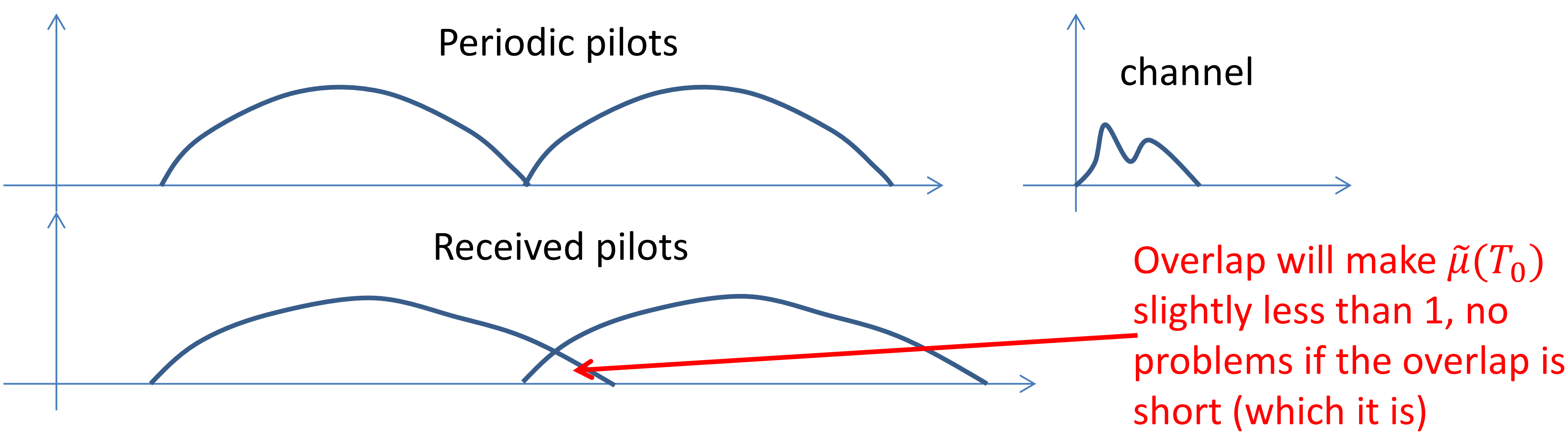


Periodic pilots in OFDM

Hence, the transmitted signal is periodic with period N/2.

In OFDM, the time between two outputs of the IFFT is short in comparison with the Channel.

But, the period of the pilots is N/2 such intervals, which is much longer than the duration of the channel (which is the key feature of OFDM)

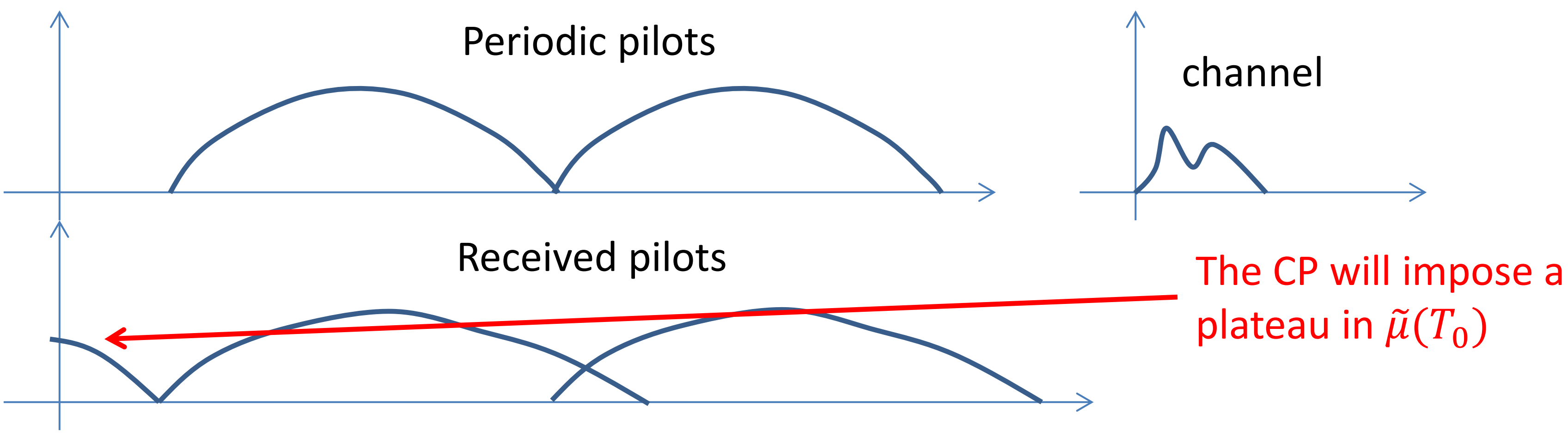


Periodic pilots in OFDM

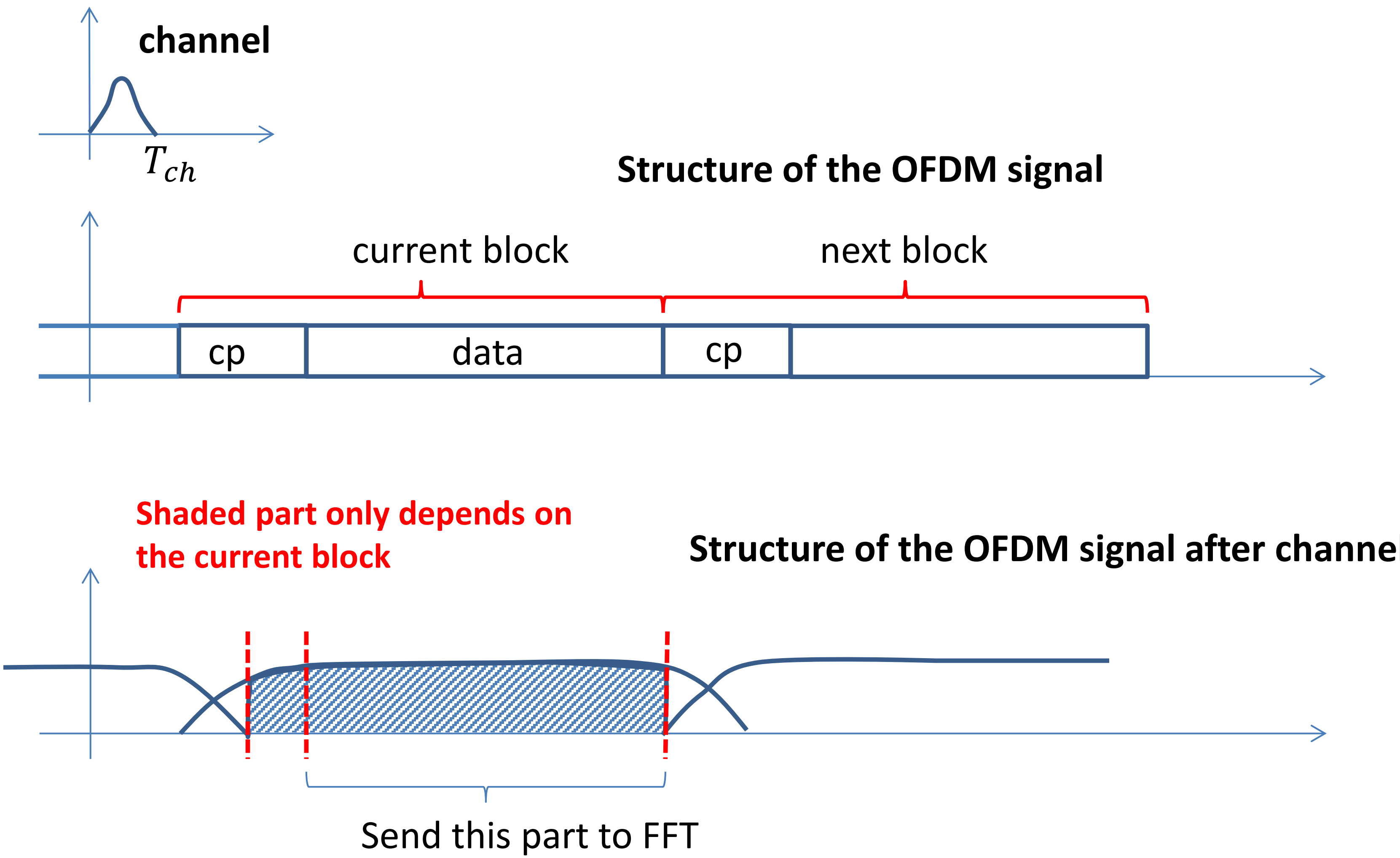
Hence, the transmitted signal is periodic with period N/2.

In OFDM, the time between two outputs of the IFFT is short in comparison with the Channel.

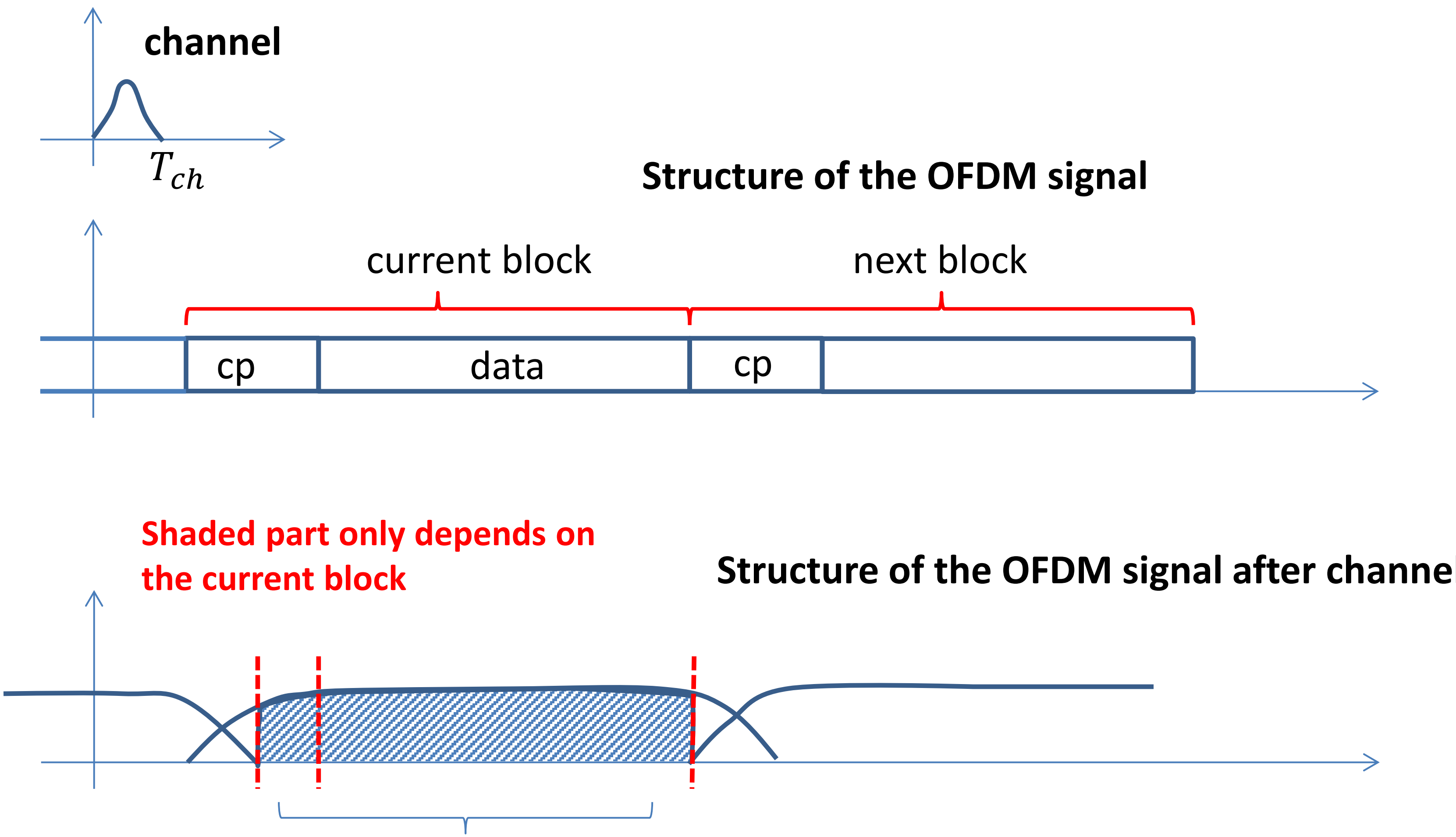
But, the period of the pilots is N/2 such intervals, which is much longer than the duration of the channel (which is the key feature of OFDM)



Mismatch in OFDM synchronization



Mismatch in OFDM synchronization



What happens if we send this part to the IFFT? (error in the synchronization)

Mismatch in OFDM synchronization

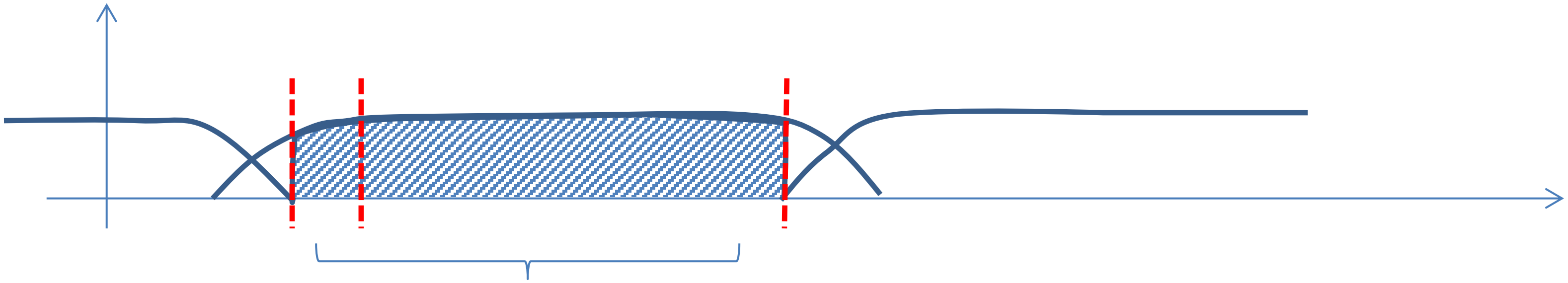
Example: CP length=3. Channel memory=1 symbol

One transmitted block after IFFT and insertion of CP

$$[\ x_{N-3}\ x_{N-2}\ x_{N-1} \mid x_0\ x_1\ x_2\ .....x_{N-3}\ x_{N-2}\ x_{N-1}\ ]$$

Received block after convolution with channel

$$[\ r_{-3}\ r_{-2}\ r_{-1} \mid r_0\ r_1\ r_2\ .....r_{N-3}\ r_{N-2}\ r_{N-1}\ ]$$



What happens if we send this part to the IFFT? (error in the synchronization)

Mismatch in OFDM synchronization

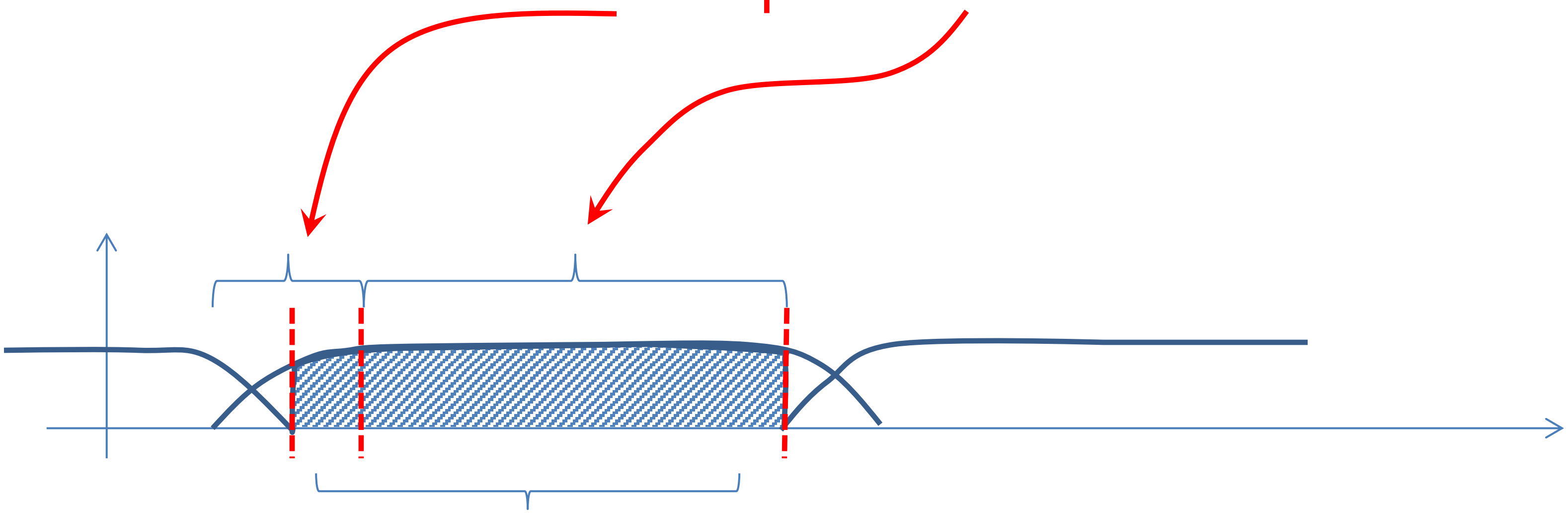
Example: CP length=3. Channel memory=1 symbol

One transmitted block after IFFT and insertion of CP

$$[\ x_{N-3}\ x_{N-2}\ x_{N-1}\ |\ x_0\ x_1\ x_2\ \ldots\ldots x_{N-3}\ x_{N-2}\ x_{N-1}\ ]$$

Received block after convolution with channel

$$[\ r_{-3}\ r_{-2}\ r_{-1}\ |\ r_0\ r_1\ r_2\ \ldots\ldots r_{N-3}\ r_{N-2}\ r_{N-1}\ ]$$



What happens if we send this part to the IFFT? (error in the synchronization)

Mismatch in OFDM synchronization

Example: CP length=3. Channel memory=1 symbol, **h**=[h_0 h_1]

One transmitted block after IFFT and insertion of CP

$$[\; x_{N-3}\; x_{N-2}\; x_{N-1} \,|\, x_0\; x_1\; x_2\; \ldots\ldots x_{N-3}\; x_{N-2}\; x_{N-1} \;]$$

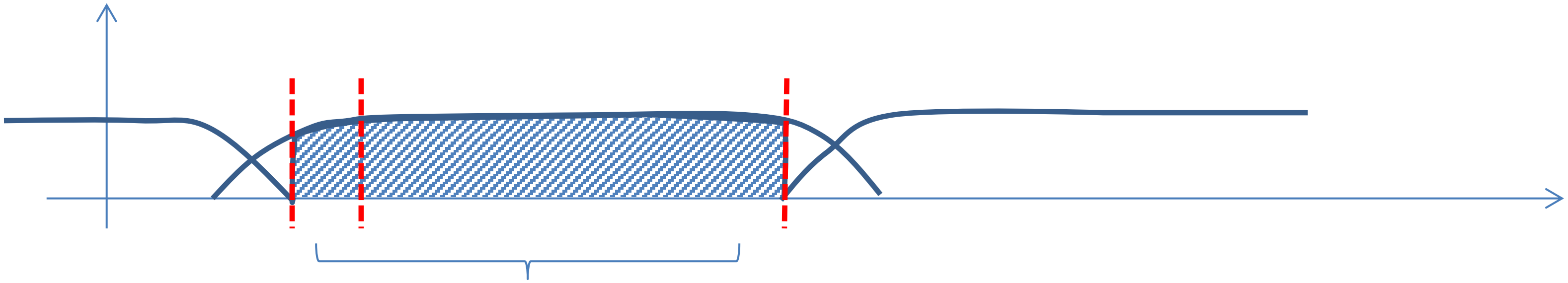
Received block after convolution with channel

$$[\; r_{-3}\; r_{-2}\; r_{-1} \,|\, r_0\; r_1\; r_2\; \ldots\ldots r_{N-3}\; r_{N-2}\; r_{N-1} \;]$$

$$r_{N-1} = h_0 x_{N-1} + h_1 x_{N-2} = r_{-1}$$

$$r_{N-2} = h_0 x_{N-2} + h_1 x_{N-3} = r_{-2}$$

$$\boldsymbol{r_{N-3} = h_0 x_{N-3} + h_1 x_{N-4} \neq r_{-3}}$$



What happens if we send this part to the IFFT? (error in the synchronization)

Mismatch in OFDM synchronization

Example: CP length=3. Channel memory=1 symbol, **h**=[h_0 h_1]

One transmitted block after IFFT and insertion of CP

[x_{N-3} x_{N-2} x_{N-1} | x_0 x_1 x_2 …..x_{N-3} x_{N-2} x_{N-1}]

Received block after convolution with channel

[r_{-3} r_{-2} r_{-1} | r_0 r_1 r_2 …..r_{N-3} r_{N-2} r_{N-1}]

$$r_{N-1} = h_0 x_{N-1} + h_1 x_{N-2} = r_{-1}$$

$$r_{N-2} = h_0 x_{N-2} + h_1 x_{N-3} = r_{-2}$$

$$\boldsymbol{r_{N-3} = h_0 x_{N-3} + h_1 x_{N-4} \neq r_{-3}}$$

So, the last 2 samples of the received block are ***cyclic***

[r_{N-2} r_{N-1} | r_0 r_1 r_2 …..r_{N-3} r_{N-2} r_{N-1}]

Mismatch in OFDM synchronization

Example: CP length=3. Channel memory=1 symbol, **h**=[h_0 h_1]

$$[r_{N-2}\ r_{N-1}\ r_0\ r_1\ r_2 \ldots\ldots r_{N-3}\ r_{N-2}\ r_{N-1}]$$

Let r[n]=[r_0 r_1 r_2 .....r_{N-3} r_{N-2} r_{N-1}] be the ideal symbols to be sent to the FFT

Let r_s[n] =[r_{N-s} ...r_{N-1} r_0 r_1 r_2 .....r_{N-s-1}] be the symbols sent to the FFT if there is ***s*** symbols synchronization error

$$r_s[n] = r[(n-s) mod\ N]$$

Take the FFT of r_s[n]

$$r_s[n] = r[(n-s) mod\ N] \underset{FFT}{\longleftrightarrow} R_s(k) = R(k)\exp\left(-\frac{j2\pi sk}{N}\right), k = 0 \ldots N-1$$

So the output of the FFT can be written as

$$R_s(k) = a_k H_k \exp\left(-\frac{j2\pi sk}{N}\right)$$

Whether one estimates H_k or $H_k\, exp\left(-\frac{j2\pi sk}{N}\right)$ is irrelevant

Summary: Mismatch in OFDM synchronization

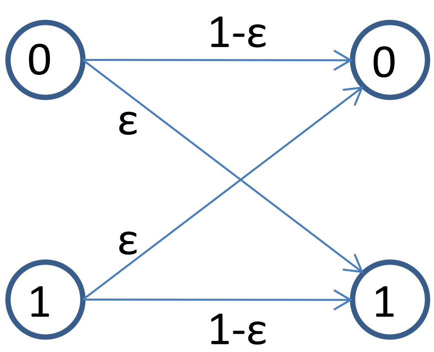
- If you estimate the start of the OFDM block to be later than what it really is, there will be problems as parts of the data is lost
- If you estimate the start earlier than what it irealIy is, there will be no problems if the mismatch **s** is smaller than the CP-duration minus the length of the channel

A little coding: Binary symmetric channels

The Tx-Rx chain

Channel inputs -> pulse shaping -> carrier modulation -> channel -> carrier demodulation -> synchronization -> matched filter -> sampling -> bit decision

Can be modeled with the binary symmetric channel (BSC)model



A bit is erroneous decoded with probability ε

Coded Transmission

If the transmitted bits are not protected, a fraction ε will be incorrectly decoded. The probability ε may be large, which jeopardize the system.

What is the simplest possible countermeasure?

In my opinion, it is to send each data symbol twice.

Data symbol	Transmitted symbols
0	00
1	11

However, this doesn't work. What should the decoder do if he receives 01?

Must be at least of length 3:

Data symbol	Transmitted symbols
0	000
1	111

Repetition codes

This is examples of repetition codes of rate R=1/n, n being the number of transmitted symbols per information symbol, i.e. n=2 and 3 on the last slide. Obviously, repetition codes are not very efficient. But they are simple enough so that exact expressions for the bit error rate can be derived. This shows that coding does indeed improve system performance.

With n=3, what is the probability that the receiver decodes incorrectly?

Say that 0 is transmitted, we get

Received	decoded	correct
000	0	Yes
001	0	Yes
010	0	Yes
011	1	No
100	0	Yes
101	1	No
110	1	No
111	1	No

Repetition codes

Hence, we decode correctly if and only if there are less than two errors on the channel.

$$\Pr((\hat{u} \neq u)) = \Pr((2 \; or \; 3 \; errors \; on \; channel)) = 3\varepsilon^2(1-\varepsilon) + \varepsilon^3$$

$$= 3\varepsilon^2 - 2\varepsilon^3 \approx 3\varepsilon^2$$

Now put in some numbers: Say that ε=.001, then the error probability is reduced down to .000003. This is a significant improvement of the system reliability!

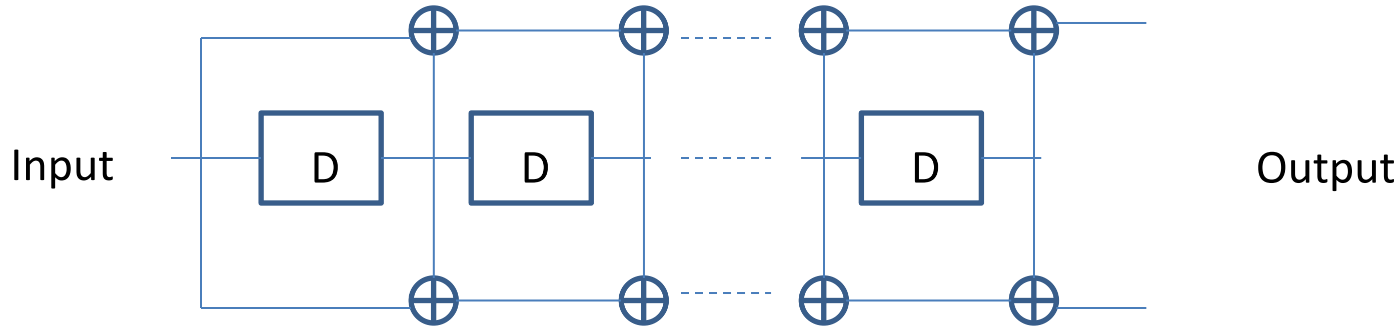
However, the code rate, definied as the number of information bits per code Bit is R=1/3. Can we do better?

Yes, with convolutional codes!

Convolutional Codes

We only look at rate R=½ codes, i.e. two outputs per information bit.

The structure of a conv. code is



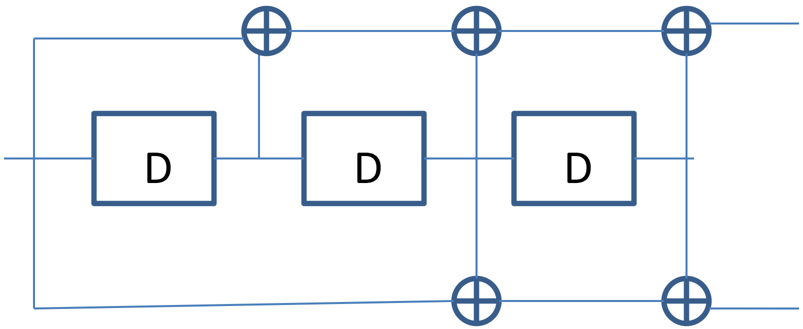
Two parameters (except for the rate R) specify the code completely:

The constraint length ν= # D-elements +1

Code generators: Specify the connections

Code Generators

Not all connections may be activated:



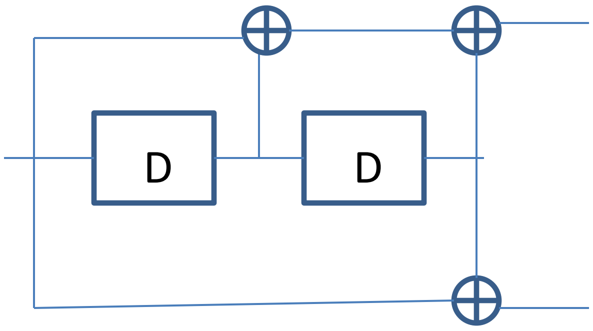
How should we give this machine a name?
Represent an activated/deactivated connections with 1/0
In the example above we get (from left to right): 1111
In the bottom we get: 1011
This is represented in octal notation, and <u>we pad with zeros from the left</u> in order to get a multiple of 3

00 1111	→	001 111	→	17
00 1011	→	001 011	→	13

And the name is (17,13), with constraint length 4

The (7,5) code

This has the name (7,5), constraint length 3 code



This is one of the most researched codes that exist.

Conventions

At the beginning of the encoding, it is assumed that the content in all the D -elements is 0.

It is also assumed that when all bits have been encoded, then the content of all the D-elements should be 0. This is achieved by concatenating ν-1 0 at the end of information block.

If the data bits
010010110 …….100100
Are to be transmitted by a ν=6 code, then one appends 5 zeros at the end so that the transmitted bits becomes
010010110 …….100100 00000

Error probability

To derive the error probability of a convolutional code is much tougher than for a repetition code.

But the performance is very good.

A convolutional code with R=1/2 and with constraint length 3-6, is much better than the R=1/3 repetition code.

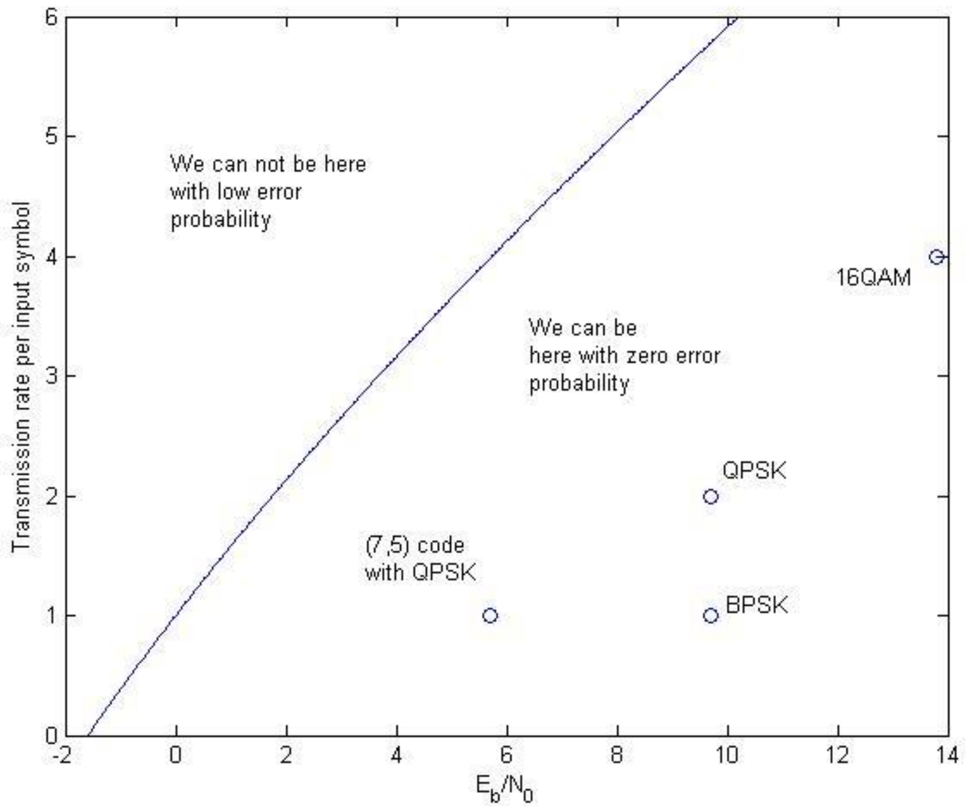
Decoding is done via the Viterbi Algorithm: complexity is exponential in the constraint length

Conv codes in matlab

You do not have to implement any convolutional decoder yourself.

In matlab, we can use vitdec.m

Information Theory dictates what can be achieved with coding



Turbo and LDPC codes operate close the the Shannon limit