Inter-Symbol Interference (ISI)

- If the channel impulse response is not $\delta(t)$ but we have some memory

$$h(t) = \sum_{l=1}^{L} a_l \delta(t - \tau_l)$$

  at reception we have, apart from noise, several delayed copies of the transmitted signal superimposed, each one multiplied by h_l: we have *inter-symbol interference (ISI)*
- In the low pass equivalent model h_l are complex, meaning that the signal reaching the Rx with delay τ_l is faded by $|h_l|$ and rotated (in the real-imaginary plane) by $\angle h_l$
- To avoid this effect we have several possibilities
  - Use some *modulations schemes* robust to ISI: OFDM, DSSS (CDMA) are good examples
  - Use *equalization* (ML, Viterbi, BCJR, LMMSE, CMA,...)

Equalization

In this course we are not reviewing equalization approaches. We just focus on OFDM as a modulation to avoid ISI.

Dispersive-Selective

- If the channel impulse response is not $\delta(t)$ but we have some memory

$$h(t) = \sum_{l=1}^{L} a_l \delta(t - \tau_l)$$

- In the *time domain interpretation*: the ISI is more severe as the delays grow compared to the symbol time (T)
  - It is said that the channel is *dispersive*
- In the *frequency domain interpretation*: the ISI is more severe as the variation of the frequency response changes compared to the bandwidth (W).
  - It is said that the channel is *selective*
- Both interpretations are related, as growing delays involves narrower changes in frequency (recall that if $h(t) = h_0\delta(t - \tau_0)$ we have no ISI:
  - The other delayed copies can be interpreted as coming with an infinite delay.
  - The frequency response is flat, not changing along W.

Dispersive and Selective Channels

- Selective and Dispersive Channels refer to the same problem but from the frequency and the time perspective, respectively.
- In M-QAM the bandwith is, rougly speaking, the inverse of the symbol time, $W = 1/T$

ISI in Matlab

Exercise 3.1 (ISI (Matlab/Octave))

- Observe selectiveness and dispersion. What is the worst channel? Try changing *b*.

```
close all; clear all; rng(30),
Nfft=2048;U=1/sqrt(Nfft)*exp(-1j*2*pi*([0:Nfft-1]')*([0:Nfft-1])/Nfft);%DFT matrix
Np=10;p=ones(1,Np);Ns=10;%p:pulse of Np samples; Ns: # symbols
d = randi([0 3],1,Ns)*2-3; %Data (-3,-1,1,3) values (4-PAM)
x = kron(d,p); %symbols multiplied by pulse
xZP=[x';zeros(Nfft-Ns*Np,1)];xt=U*xZP; %Zero pad. (circ. conv.); xt: DFT
%% Channel
a=1;b=-.5; scale=2; %Norm is to better observe results
h(:,1)=scale*[1,0,0,0,0,0,0,0,0,0]; h(:,2)=scale*[a,b,0,0,0,0,0,0,0,0];
h(:,3)=scale*[a,0,0,0,0,b,0,0,0,0]; h(:,4)=scale*[a,0,0,0,0,0,0,0,b,0];
numberChannels=4; L=10; %L:channel length;
%% Filtering, using DFT
for k1=1:numberChannels
   ht(:,k1)=U*[h(:,k1); zeros(Nfft-L,1)]; %DFT of channel
   yt=xt.*ht(:,k1); y(:,k1)=real(sqrt(Nfft)*U'*yt); % = conv(x,h)
end
%% Representation
T=1e-3; Tm = T/Np; %Suppose T = 1 ms,
figure(1), xaxis=(1:Np*Ns)/Np; plot(xaxis,x,'linewidth',3),hold on,
plot(xaxis,y(1:Np*Ns,2:numberChannels)), xlabel('ms'),legend('x','y1','y2','y3'),
     grid on,
figure(2),plot([1:Nfft]/Nfft*1/Tm,abs(ht)), grid on, hold on,
plot(ones(1,2)*(1/(Np*Tm)),[min(abs(ht(:))),max(abs(ht(:)))],'--k'),
title('DFT of channels'), xlabel('Hz'), legend('\delta(t)','h1','h2','h3','W/2')
```

Introduction

- Usually we have to estimate (know) the channel (*CSI, channel state information*)
  - *Training symbols* or *pilots* are sent for this task, e.g. GSM (training sequence) or LTE (pilots)
- Classical systems try to estimate the transmitted symbols after estimating the channel impulse response
  - This is known as *equalization*
  - Optimal equalization is based on either maximize the probability of
    - the whole word to be transmitted (ML solved with *Viterbi* algorithm)
    - every symbol to be transmitted (MAP solved with *BCJR* algorithm)
  - As the memory (number of taps) of the channel and the constellation order increase, these optimal equalizers become complex. Approximate (linear) solutions:
    - *Zero Forcing*
    - *LMMSE*
- If we use the already estimated bits to improve the equalization we have a *DFE*, decision feed-back equalization
- If we use the output of the channel decoder we have iterative detection and decoding (IDD) approaches, such as *turbo-equalization* schemes.
- *OFDM* (downlink) is another alternative
  - We change the transmission to have no ISI at the receiver
- Modulations using more bandwith than needed are robust to selectiveness
  - E.g. DS-SS (*direct-sequence spread spectrum*) and the Rake receiver
- If the channel response is known to the transmitter, we can easily design ISI-free systems (TDD systems favours this)

OFDM: Outline

- OFDM is a form of multi-carrier system, where information is transmitted parallelized over several *narrowband* subcarriers
- The key idea of OFDM is that each subcarrier is orthogonal to each other while separation between subcarriers is kept to a minimum
- The digital implementation is easy if the Fourier transform is used
  - the *circular* convolution of signal and channel response, converted to frequency domain yields the *product* of the Fourier transforms of the signal and the channel
- For each *block of N symbols*, we proceed as follows,
  1. *transmit data in the frequency domain*: each frequency component, value of the discrete Fourier transform (DFT), is a symbol in the block, we get: $\tilde{\mathbf{d}}$
  2. *transform it to time domain*: get the discrete time response, applying inverse DFT (IDFT) to $\tilde{\mathbf{d}}$, we have: $\mathbf{d}$
  3. *add cyclic prefix*, to ensure cyclic convolution, we get: $\mathbf{x}$
  4. *transmit* through the channel, *with L taps*, we have: $\mathbf{y}$
  5. *remove cyclic prefix and apply DFT*, the product of the DFT of the channel and the transmitted symbols to get: $\tilde{\mathbf{y}}$
- The *cyclic prefix* is added to ensure that the transmission over the channel yields a circular convolution: it also avoids *IBI* (inter-block interference)

OFDM Principle: Notes on the convolution

- Suppose you have a signal in *continuous time* and it goes through a channel. Its transform yields

$$\mathcal{F}[x(t) * h(t)] = X(\omega) \cdot H(\omega) \quad (1)$$

  - Hence, if we transmit the incoming symbols in time as if they were frecuency transform coefficients, $X(t)$, and compute the transform of the received signal, after the channwel,

$$\mathcal{F}[\mathcal{F}^{-1}[X(t)] * h(t)] = X(t) \cdot H(\omega) \quad (2)$$

The (entry-wise) product of the signal by the frequency response of the channel.

- In *discrete time* we focus on circular convolution. The N points transformation yields

$$\mathcal{F}[x[n] \otimes h[n]] = \sqrt{N} \cdot \tilde{x}[k] \cdot \tilde{h}[k] \quad (3)$$

  - Hence, if we think of the incoming symbols in time as the transform coefficients in frequency, $\tilde{x}[n]$,

$$\mathcal{F}[\mathcal{F}^{-1}[\tilde{x}[n]] \otimes h[n]] = \sqrt{N} \cdot \tilde{x}[n] \cdot \tilde{h}[k] \quad (4)$$

We design the system to get a circular convolution after the (standard) convolution with the channel: introducing the cyclic prefix (CP)

OFDM: Steps and Formulation

- *Step 1* is immediate, the DFT of the transmission are the symbols

$$\tilde{\mathbf{d}} = [\tilde{d}_1, \dots, \tilde{d}_{N_c}]^\top$$

- To get the signal in the *time domain*, we
  a) compute the IDFT in *Step 2* to get the samples to transmit $\mathbf{d} = [d[1], \dots, d[N_c]]^\top$
  b) then, in *Step 3*, add the cyclic prefix: copy the last $L-1$ samples to the beginning, and transmit

$$\mathbf{x} = [x[1], \dots, x[N_c + L - 1]]^\top = \Big[\underbrace{d[N_c - L + 2], \dots, d[N_c]}_{\text{cyclic prefix}}, d[1], \dots, d[N_c]\Big]^\top$$

- In *Step 4* we observe, in time, the output of the channel, *that has at most L taps*,

$$\dots, [\text{previous block}], \Big[\underbrace{y[1], \dots, y[L-1]}_{\text{IBI}}, \underbrace{y[L], \dots, y[N_c + L - 1]}_{h \otimes d + \mathbf{w}}\Big], [\text{following block}], \dots$$

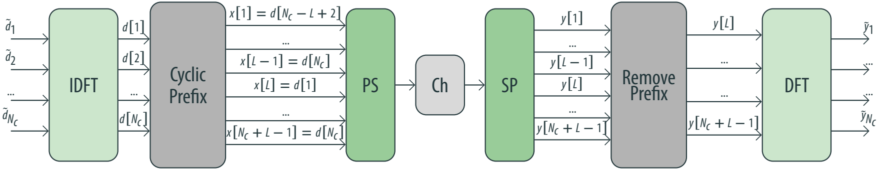
- In *Step 5* remove the cyclic prefix and <u>compute the DFT</u> of

$$\mathbf{y} = [y[L], \dots, y[N_c + L - 1]]$$

hence the output is $\tilde{\mathbf{y}} = [\tilde{y}_1, \dots, \tilde{y}_{N_c}] \Rightarrow \tilde{y}_i = \sqrt{N_c}\tilde{d}_i\tilde{h}_i + \tilde{w}_i$

Note: *IBI*, inter block interference, from the previous block: null if the number of taps of the channel is $\leq L$

OFDM as a Sequence of Blocks



OFDM

The *most important result* is that

$$\tilde{y}_i = \sqrt{N_c}\tilde{h}_i\tilde{d}_i + \tilde{w}_i,\ i = 1,\ldots,N_c \tag{5}$$

where $\tilde{\mathbf{h}}$ is the DFT of the channel response, and $\tilde{\mathbf{w}}$ is the DFT of the noise, having the same distribution than $\mathbf{w}$

There is no ISI: no equalization is needed

- The IDFT+CP+DFT scheme in OFDM was designed to avoid ISI exploiting the circular convolution property of the DFT
- We next review this process resorting to the matrix formulation

OFDM: Matrix Formulation

- The DFT and IDFT operations can be implemented as a product of a vector by a *unitary* matrix, $\mathbf{U}$ and $\mathbf{U}^{-1} = \mathbf{U}^{\mathrm{H}}$

$$\mathbf{U}_{k,n} = \frac{1}{\sqrt{N_c}}\, \mathrm{e}^{-\mathrm{j}2\pi\frac{(k-1)(n-1)}{N_c}}, \qquad k,n = 1,\dots,N_c \tag{6}$$

- The circular convolution, $\otimes$, with the channel can be also rewritten as

$$\mathbf{h} \otimes \mathbf{d} = \mathbf{C}\mathbf{d} \Rightarrow \mathbf{y} = \mathbf{C}\mathbf{d} + \mathbf{w} \tag{7}$$

where

$$\mathbf{C} = \begin{bmatrix} h_1 & 0 & \dots & 0 & h_L & h_{L-1} & \dots & h_2 \\ h_2 & h_1 & 0 & \dots & 0 & h_L & \dots & h_3 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & h_L & h_{L-1} & \dots & h_2 & h_1 \end{bmatrix} \tag{8}$$

is a circulant matrix, and for this type of matrix

$$\mathbf{C} = \mathbf{U}^{-1}\Lambda\mathbf{U} \quad \text{and} \quad \Lambda = \mathrm{diag}\left(\sqrt{N_c}\tilde{\mathbf{h}}\right) = \mathrm{diag}\left(\sqrt{N_c}\mathbf{U}\mathbf{h}\right) \tag{9}$$

OFDM: system design

- Hence $\mathbf{y} = \mathbf{C}\mathbf{d} + \mathbf{w}$ and

$$\tilde{\mathbf{y}} = \mathbf{U}\mathbf{y} = \mathbf{U}\mathbf{U}^{-1}\Lambda\mathbf{U}\mathbf{d} + \mathbf{U}\mathbf{w} \tag{10}$$

- It yields

$$\tilde{y}_i = \sqrt{N_c}\tilde{h}_i\tilde{d}_i + \tilde{w}_i \tag{11}$$

$$\tilde{\mathbf{h}}_i = \mathbf{U}\mathbf{h} \tag{12}$$

- *With no ISI* and can be easily recovered if the channel is known
  - The cyclic prefix, of duration L, allows removing the ISI from the previous block
- The DFT and IDFT can be efficiently computed if N_c is a power of 2
- In frequency we have subcarriers separated by W/N_c

OFDM as a set of AWGN channels

With OFDM the transmission over a selective channel boils down to N_c transmissions over flat ones

OFDM: Frequency

- N_c symbols transmitted: $\tilde{d}_i$, one per sub-carrier
  - Every sub-carrier is amplitude modulated by, $\tilde{d}_i$, one out of N_c symbols
- Bandwith of a sub-carrier $W_{sc} = \frac{W}{N_c} = \frac{1}{N_c T}$ is the
  - Overall bandwidth is $W = \frac{1}{T} = N_c W_{sc}$
- The output, after DFT, are the symbols $\tilde{d}_i$ by $\sqrt{N_c}$ times the i-th DFT value of the channel plus noise
  - NO ISI $\Rightarrow$ no equalization needed
  - Equivalent to N_c flat channels
- The correlation between $|\tilde{h}_n|$ and $|\tilde{h}_m|$ depends on W_c (coh. bandw.)
  - W_c is inversely proportional to the multipath spread delay $T_d = LT$

$$W_c = \frac{1}{2T_d} = \frac{W}{2L} \tag{13}$$

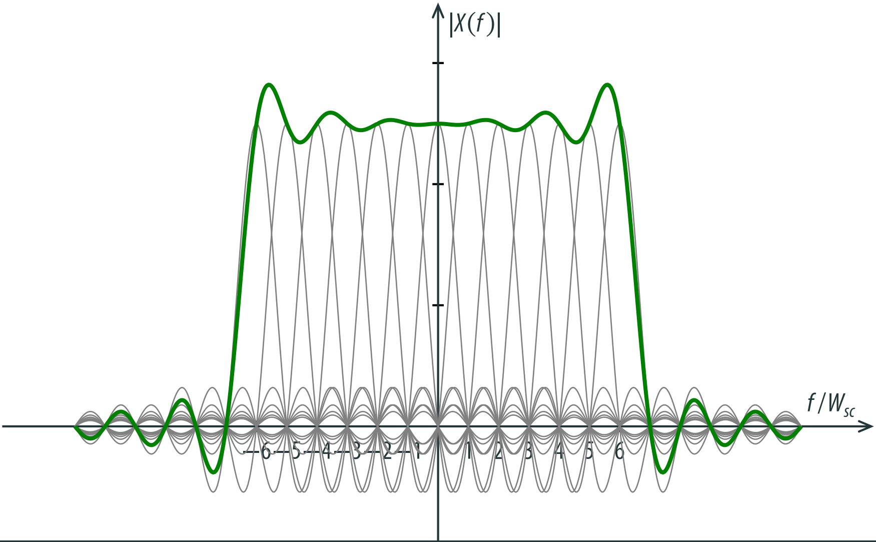
the number of contiguous *correlated* subcarriers in this bandwith is

$$\frac{W_c}{W_{sc}} = \frac{W_c}{W/N_c} = \frac{N_c}{2L} \tag{14}$$

  - Bearing this in mind, channel coding and interleaving between them: *COFDM, coded OFDM*
- If sub-carriers central frequencies shift (Doppler, oscillators errors,...)
  - We have *inter carrier interference (ICI)*

OFDM: Frecuency, Spectrum

- Low-pass equivalent spectrum for $N_c = 13$ subcarriers with bandwidth W_{sc}
- The values at f/W_{sc} are $|\tilde{d}_i|$



OFDM: Time

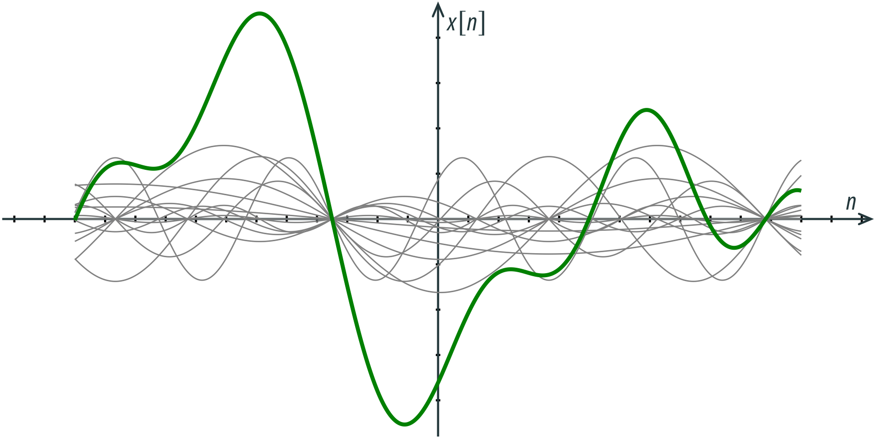
- N_c symbols transmitted:
  - $T = 1/W$ is the duration of every incoming symbol $\tilde{d}_i$
  - $N_c T$ is the duration of every symbol $\tilde{d}_i$, the N_c are sent at the same time
- Prior to every transmission the CP is sent, repeating the last symbols in time.
  - We need also $(L-1)/W$ seconds to transmit the CP
  - $((L-1)T)/(N_c T + (L-1)T)$ of the time we are not transmitting useful information
- If channel spreads further than L time symbols,
  - we have inter block interference (IBI)
- Every symbol d_i is a combination (given by the IDFT matrix) of the original symbols $\tilde{d}_i$
- This combination, for some symbols d_i, may be high → *PAPR, peak to average power ratio*
- The transmission, at every symbol time, is using all the bandwith W
- The brute throughput

$$R_b = \log_2(M)/T = \log_2(M)W = \log_2(M)N_c W_{sc} \tag{15}$$

- Must be included the CP, the pilots used or the channel coding rate to get the *net rate*.

OFDM: Time, Signal

- The inverse transform of every subcarrier is plotted in gray
- Each subcarrier in time has a frequency given by f_i ($f_{i+1} - f_i = W_{sc}$) and an amplitude proportional to $\tilde{d}_i$
- The combination of all of them, in green, is the transmitted signal in time
- For some sequences of $\tilde{d}_i$ the combination may have large peaks: PAPR



OFDM block length setting

- OFDM is inefficient due to the cyclic prefix, along $L' = L - 1$ of the total time we do not transmit useful information, with a loss of efficiency:

$$L'/(L' + N_c) \tag{16}$$

  - We lose $L'/(L' + N_c)$ of the *time*
  - We waste $L'/(L' + N_c)$ of the *average power*

N_c should be as large as possible $\Rightarrow L \ll N_c$: can N_c grow unbounded?

$\Rightarrow$ The block should no go longer than the coherence time, the time the channel is considered to be invariant

Problems in OFDM

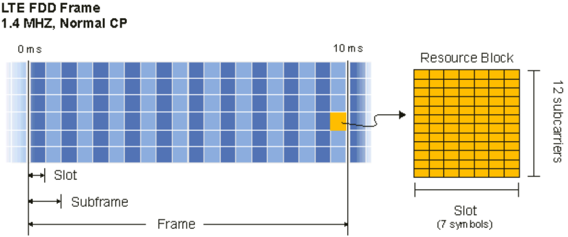
1. Faded subcarriers
   - We may use coding (*coded OFDM, COFDM*)
   - For a given fixed transmitted power we will mainly use the good ones: *waterfilling techniques*
   - We may use different modulations for every subcarrier: *AMC*, adaptive coding and modulation (a.k.a. *VCM*, variable cod. and mod.)
     - *MCS*, indicador of the modulation and coding scheme
2. In time, the signal may have large values along short times: *PAPR (or PAR), peak-to-average power ratio*
   - *Uplink in LTE is SC-OFDM !!*: equalization is needed
3. Frequency offset (e.g. due to Doppler) in subcarriers: the effect is a mixture between subcarriers, *ICI, inter-carrier interference*
4. Accurate time synchronization: otherwise the FFT at receiver fails to recover the data

Practical Values

- Frames of 10 ms, with 10 subframes, 2 slots each of 0.5 ms
- The basic unit is the *resource block (RB)*, two types
  - 12 subcarriers (sc) of 15 kHz (180kHz overall), along 0.5 ms, transmitting 7 (or 6) symbols
  - 24 sc of 7.5 kHz, along 0.5 ms, with 3 symbols
- and the *resource element (RE) is 1 sc* M-QAM (M = 4, 16, 64) (ACM used), *1 symbol time*
- The number of resources elements used by an user in one time slot (TS),

$$N_{RE} = N_{RB} \cdot N_{SC}^{RB} \cdot N_{symb}^{TS} \Rightarrow R_b = \log_2(M) N_{RE}/0.5\text{ ms} \tag{17}$$

  is the # of RB, by the # of sc per RB (12,24), by the # of symbols per frame (3,6,7)
- $W = 20$ MHz and FDD: the normal (7 symb, 15 kHz) cyclic prefix (CP) is $144 \cdot T_s$ where $T_s = 1/(15000 \cdot 2048)$ is the *time unit*, first symbol has $160 T_s$ of CP
- Part of the RE are devoted to pilots and signaling



OFDM in Matlab

Exercise 3.2 (OFDM parameters (Matlab/Octave))

- Change the length of the channel: L=1, L=10, L=110
- Change the noise: EbN0=200, 20 and 10.
- Indicate, for Nsc=1200, the bandwith used.

```
Nsc=2048; Wsc=15e3; CP=144; L=110; %L: Length of channel, < CP
EbNo=20; Ts = 1/(Nsc*Wsc); %Eb/No in dB and sampling time
Nscu=1200; %Number of SC used, maximum number is 1200.
%Data, %QPSK
Bn=randi([0 1],2,Nscu)*2-1; %Note that Eb=1 since Es=2
dt=Bn(1,:)+1j*Bn(2,:); dtZP=[dt, zeros(1,Nsc-Nscu)];
%IDFT and %CP, Just to observe time sequence
U=1/sqrt(Nsc)*exp(-1j*2*pi*([0:Nsc-1]')*([0:Nsc-1])/Nsc);
d=U'*dtZP.'; x=[d(end-CP+2:end); d].';
%Channel
h=randn(1,L); h=h/norm(h); ht=U*[h.'; zeros(Nsc-L,1)]; %Normalized
no2=10^(-EbNo/10)/2;noise=sqrt(no2)*(randn(1,Nsc)+1j*randn(1,Nsc)); %eb=1
%DFFT
y=sqrt(Nsc)*ht.*dtZP.'+U*noise.';
stem([1:Nsc+CP-1]*Ts,real(x)); xlabel('t'),title('Real(Tx signal) vs t')
figure
stem([1:Nsc]*Wsc,abs(y)); xlabel('frequency'),title('Frequency Response')
hold on, plot([1:Nsc]*Wsc,abs((sqrt(Nsc)*ht)),'r');
plot([1:Nsc]*Wsc,abs(dtZP),'g'); xlabel('f'), legend('Rx','Channel','Tx')
```

Exercise, LTE

Exercise 3.3

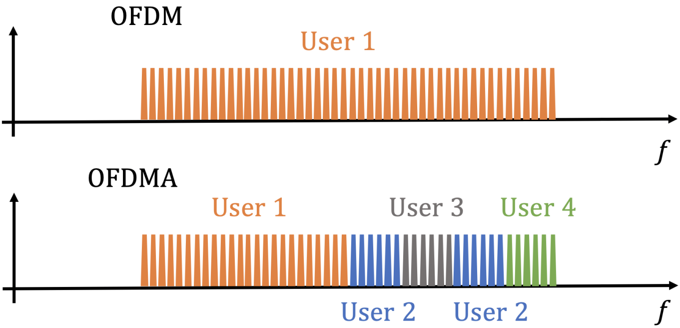
OFDM in LTE

- Given the DL in LTE, using OFDM, with the following features :
  - 4.5 MHz bandwith
  - $W_{sc} = 15$ kHz
  - RB of 12 sc along 0.5 ms, 7 symbols
  - Modulations allowed are QPSK, 16-QAM and 64-QAM
  - Available coding rates are 1/3, 1/2, 3/4.
  - Roughly, 10% of uncoded transmission is signaling
  - Assume the user is transmitting in all slots
- It is asked to
  - Determine the time of the normal cyclic prefix
  - Compute the minimum and maximum data rate available for a user transmitting in a RB
  - Compute the value above if all the bandwith is used

- Notes: `https://home.zhaw.ch/~kunr/NTM1/literatur/LTE%20in%20a%20Nutshell%20-%20Physical%20Layer.pdf`

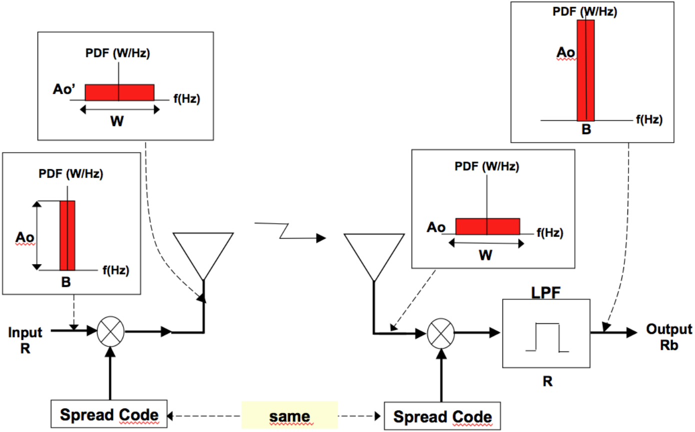
OFDMA

- If different RB are assigned to different users at the same time we have *orthogonal frequency division multiple access, OFDMA.*
- The assignment may change with time: we need a *scheduling*.

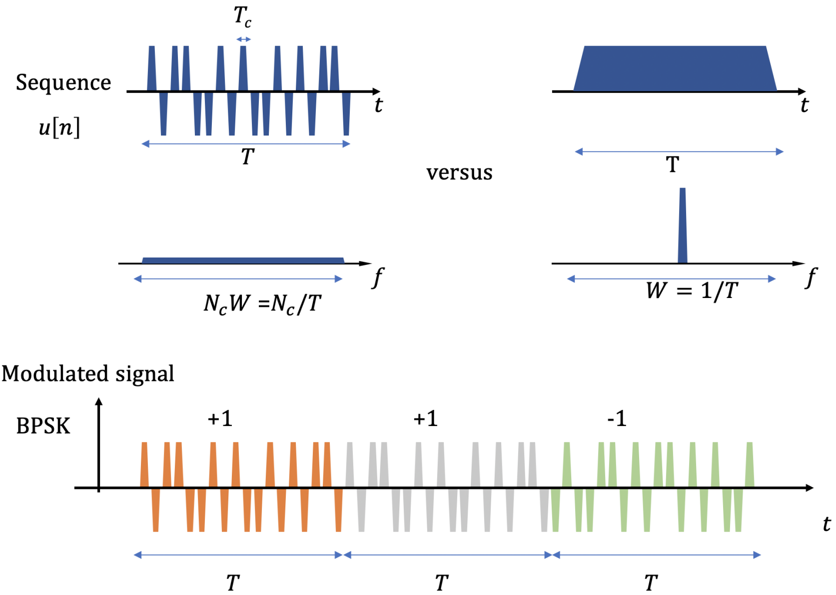


DS-SS: Introduction

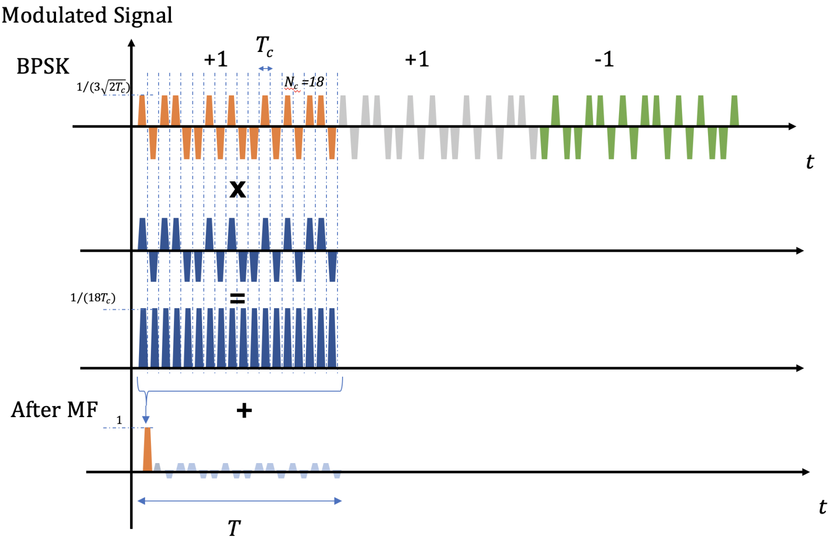
- Assume a binary signal is transmitted with rate B bits/s
- A *pseudosequence (PN)* is used as transmission pulse over a bandwith W
  - Usually a pseudo ramdom sequence of N_{ch} pulses, called *chips*
- W/B is denoted as *processing gain*



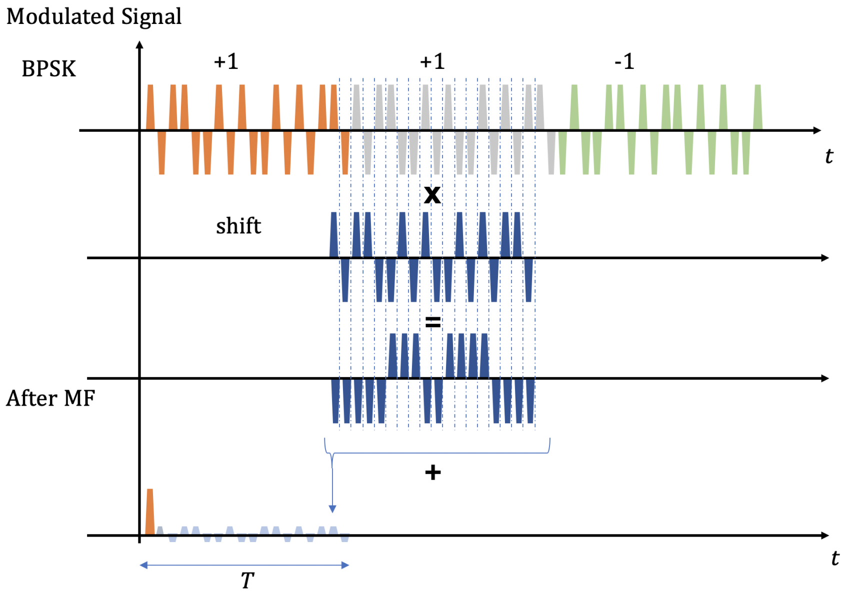
DSSS


T_c
Sequence
$u[n]$
t
T
f
$N_c W = N_c/T$
versus
t
T
f
$W = 1/T$
Modulated signal
BPSK
+1
+1
-1
t
T
T
T

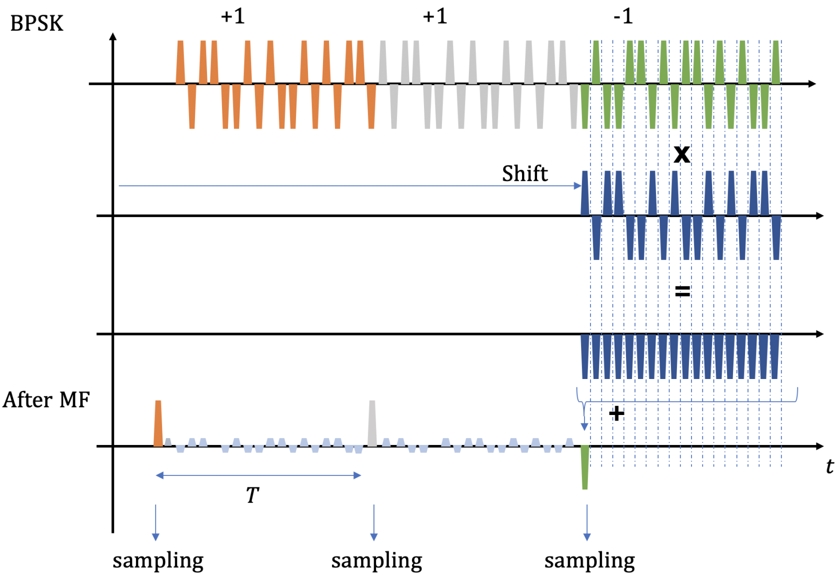
DSSS Demodulation



DSSS Demodulation



DSSS Demodulation



Rake Receiver

- We assume an underspread slow varying channel:
  - $L << N_{ch}$, hence $L \cdot 1/W = T_d << T_c$
  - $N_{ch} \cdot 1/W = T << T_c$
- We assume that the channel, $h_l, l = 1,\ldots,L$, is known (CSIR)
- We transmit one out of two codewords (sequences) $\mathbf{x}_A$ and $\mathbf{x}_B$
- The output yields,

$$y[m] = (h * x)[m] + w[m], \quad m = 1,\ldots,N_{ch} + L - 1 \tag{18}$$

- Assume that we transmit an antipodal modulation $\mathbf{u} = \{\mathbf{x}_A, \mathbf{x}_B = -\mathbf{x}_A\}$
- Define an extended (with L zeros) and shifted (by l) version of $\mathbf{u}$

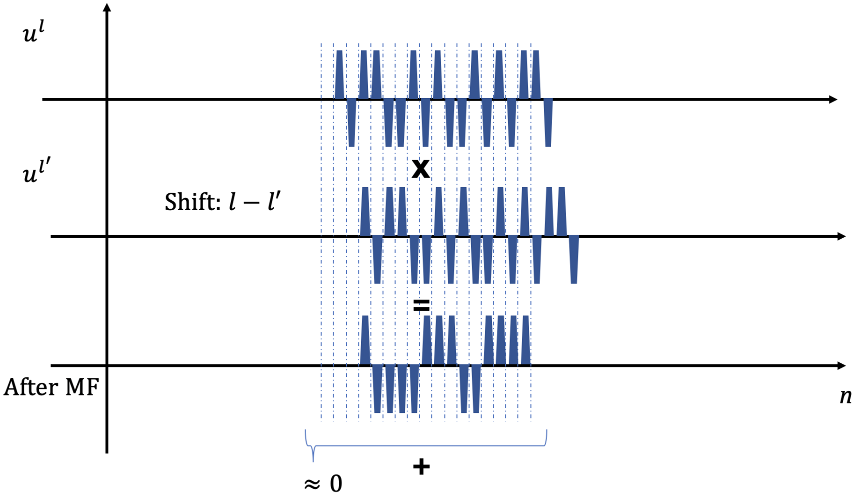
$$\mathbf{u}^{(l)} \stackrel{\text{def}}{=} [\underbrace{0,\ldots,0}_{l \text{ zeros}}, u[1],\ldots,u[n], \underbrace{0,\ldots,0}_{L-l \text{ zeros}}] \tag{19}$$

- The property of $\mathbf{u}$ being PN involves $(\mathbf{u}^{(l)})^{\mathsf{H}}(\mathbf{u}^{(l')}) \approx 0$ if $l \neq l'$

Rake property

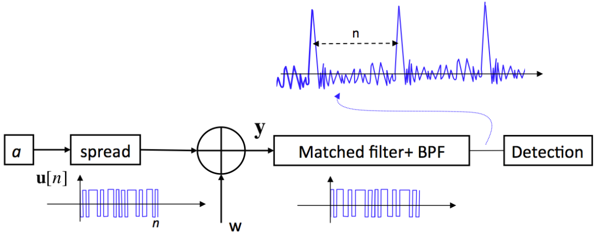
The rake receiver is based on $(\mathbf{u}^{(l)})^{\mathsf{H}}(\mathbf{u}^{(l')}) << (\mathbf{u}^{(l)})^{\mathsf{H}}(\mathbf{u}^{(l)}) = \sum_{i=1}^{n} |u_i[i]|^2,\ l \neq l'$

DSSS RAKE

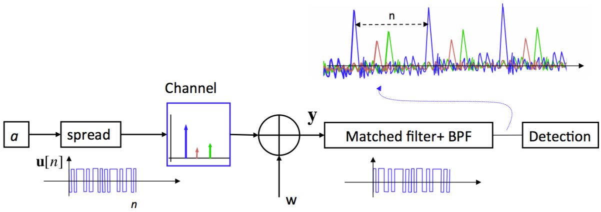


Matched filter for DS-SS

- In a flat channel, the optimal receiver is the *matched filter (MF)*



- In multipath channel we still have a good estimation, the MF is robust, but we can improve it with a rake receiver



Rake Receiver: Design

- We rewrite the channel model, where $a = \pm 1$

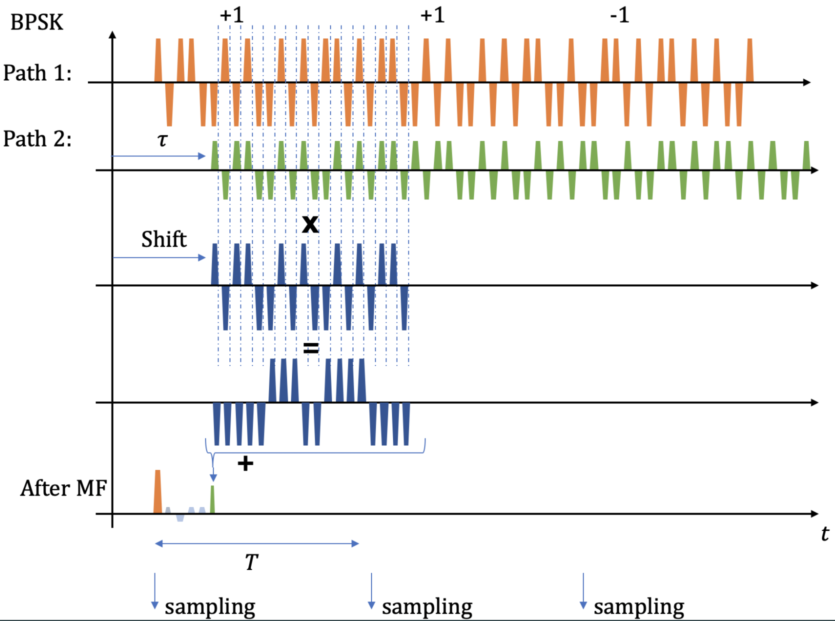
$$\mathbf{y} = a \sum_{l=0}^{L-1} h_l \mathbf{u}^{(l)} + \mathbf{w} \tag{20}$$

- Since we know the channel, the model can be cast as a transmission of $a = \pm 1$ multiplied by a pulse

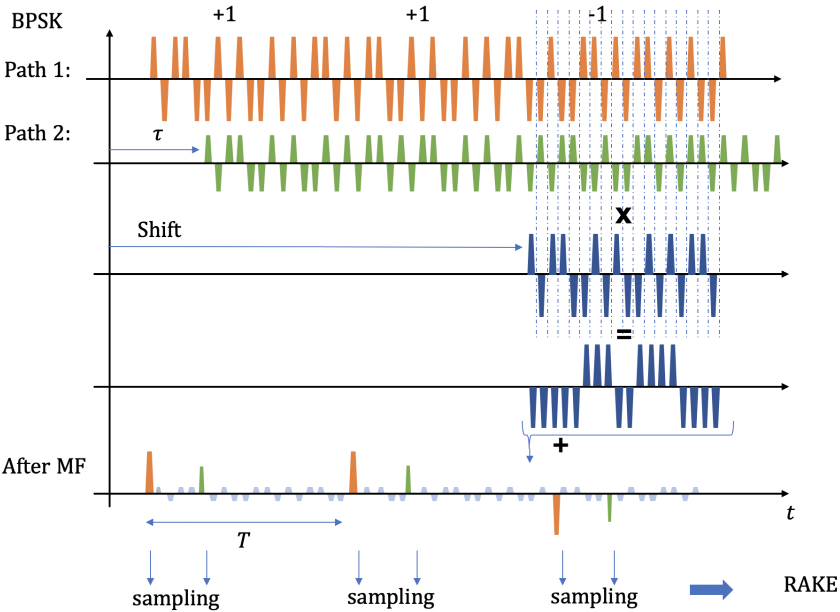
$$\mathbf{u}' = \sum_{l=0}^{L-1} h_l \mathbf{u}^{(l)} \tag{21}$$

- The optimal receiver is the *matched filter*: we project the receive signal $\mathbf{y}$ onto $\mathbf{u}'$ and decide on a
- By the *rake property* this is equivalent to project the receive signal onto L matched filters $\mathbf{u}^{(l)}$, $l = 1,\dots L$, and *maximally ratio combining* them using coefficients h_l
- Note that we need to estimate the channel
  - We could select a subset $\mathcal{S} \subset \{1,\dots,L\}$ of indexes, e.g. with largest $|h_l|$
  - Every $l \in \mathcal{S}$ is called a *finger*
- In the receiver we usually work at chip time, and the vectorial receiver is matched to the chip pulse.

DSSS RAKE


BPSK
+1
+1
-1
Path 1:
Path 2:
τ
Shift
x
=
+
After MF
t
T
sampling
sampling
sampling

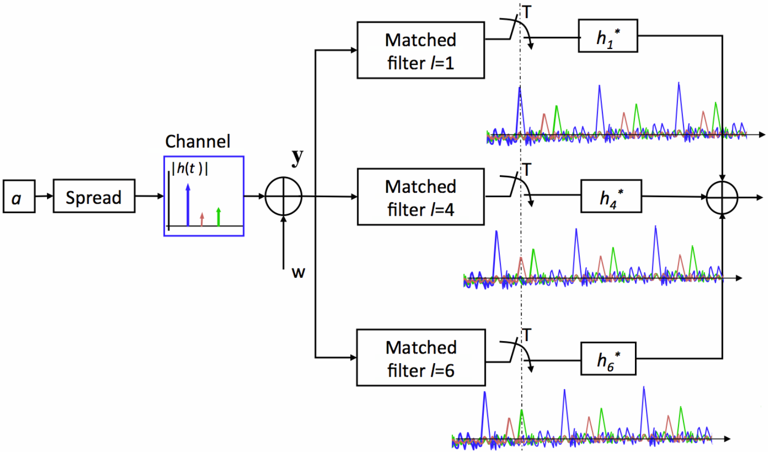
DSSS RAKE



BPSK
+1
+1
-1
Path 1:
Path 2:
τ
Shift
x
=
+
After MF
t
T
sampling
sampling
sampling
RAKE

Rake Receiver for DS-SS

- In this example we select 3 fingers at 1, 4 and 6.



DSSS in Matlab

Exercise 3.4 (DSSS and RAKE)

- Change the channel and the spreading factor and observe the results
- How it is robust to selective fading? Try changing the pulse to a square one.

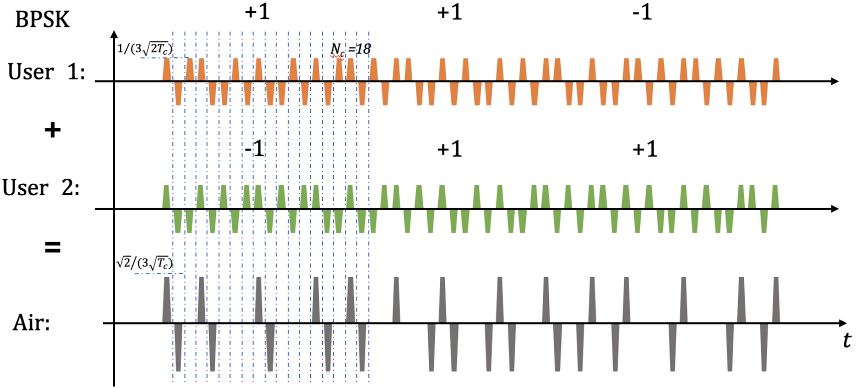
```
close all; clear all; rng(18)
Nc=50; Ns=10; %Nc: # of chips; Ns:Number of 4-PAM symbols
p = randi([0 1],1,Nc)*2-1; p=p/sqrt(Nc); %Samples per symbol and pulse
d = randi([0 3],1,Ns)*2-3; %Data (-3,-1,1,3) values
x = kron(d,p); %symbols by pulse
%% Channel (no noise)
a=1*exp(1j*pi/8); b=-0.8; h=[a,0,0,0,0,0,0,0,0,0,b];normh=norm(h);h=h/normh;
L=length(h); numberChannels=size(h,2);
y=conv(x,h); y=y(1:length(x)); %% Filtering, using conv
%% Representation
T=1e-3; Tm = T/Nc; xaxis=(1:length(x))/Nc;%Suppose T = 1 ms,
figure(1),plot(xaxis,x,'-','linewidth',1),hold on
%plot(xaxis,y), xlabel('ms'),legend('x','y'), grid on %if a,b real valued
%% Reception with Matched Filter
hr=fliplr(p); %Matched filter to the pulse, conj(hr) if complex
yn=conv(y,hr); %Output vectorial receiver (MF)
zn=conv(x,hr); %Output vectorial receiver (MF) No ISI
rn=real(yn(Nc:Nc:Nc*Ns)*conj(a)/abs(a)); %Output vectorial receiver
sn=real(yn(Nc:Nc:Nc*Ns)*conj(a)/normh+yn(Nc+L-1:Nc:Nc*Ns+L-1)*conj(b)/normh); %RAKE
yn=real(yn(1:length(y)));zn=real(zn(1:length(x)));
figure(2),plot(xaxis,zn,'--','linewidth',1),hold on
plot(xaxis,yn), xlabel('ms'),legend('x','y'), grid on
figure(3), stem(d),hold on, stem(sn),stem(rn),xlabel('ms'),legend('d','RAKE','No-
     RAKE'),grid on
```

Facts of DS-SS

- In practice we have a few taps, where some of them are higher than the others and used in the rake as *fingers*
- DS-SS is quite robust to interference: that was way it was adopted by IEEE 802.11b
- DS-SS can be used as multiple access technique
  - Its use in cellular systems has been a major advance to reuse spectrum
- Its time resolution has been used in localization systems such as GPS
- UWB uses a similar idea, can be casted as a DS-SS with no modulation and much more large bandwidth, and where most of the sequence is set to zero.

CDMA

If several users are DSSS modulated with different codes we have *code division multiple access*



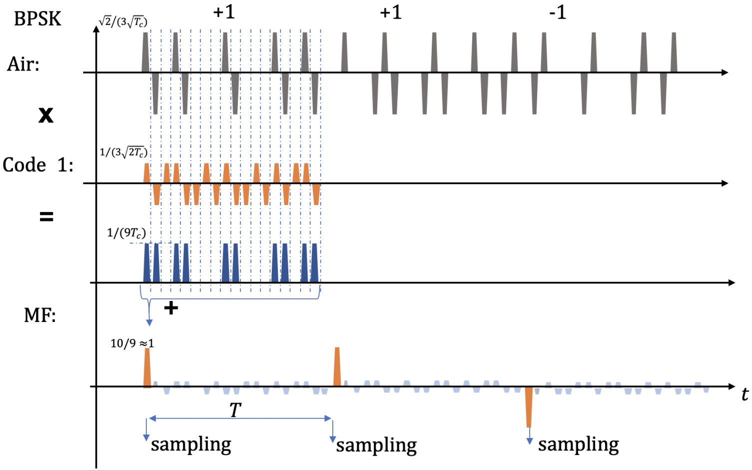
- The mathematical model, discretized at chip time, T_c, yields

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w} \tag{22}$$

where $\mathbf{H}$ is a matrix with as many columns as users and as many rows as chips (18×2 in the example): reception with MF yields $\hat{x} = \mathbf{h}_i^{\mathrm{H}} y$, where $\mathbf{h}_i$ is the ith column of $\mathbf{H}$.

CDMA Detection

To detect, the easiest way is to use a matched filter to the *user of interest (UoI)*.



- Here we depicted the case of two users with equal power and synchronized, recovering User 1.
- A similar scheme, with similar results apply to recover User 2.

Orthogonal codes

- Since users may not be synchronized (e.g. UL) and to be robust to ISI codes must be as orthogonal as possible.
  - one to each other, and
  - to any delayed version of itself and the others.
- The design of good codes is of importance.
- At reception we have interference of others users.
  - If all codes are quasi-orthogonal and users reach the Rx with similar power levels this is not a problem, if we have not too many users.
  - If the UoI is received with less power than others we have a severe interference problem: *multiuser inteference*.
  - This is a typical case when the interferer is near the Rx while the UoI is far: *near-far problem*
  - To solve this a fast power control (e.g. to make the nearby user to lower the power) is needed.
    - In UMTS (3G) we have a 1500Hz power control update rate!!
    - In UMTS we have 3.84 Mcps (*mega chips per second*), with this high bandwith, *wideband CDMA (WCDMA)* is used to denote this multiple access.

Appendix I: DFT

- Given

$$\mathbf{U}_{k,n} = \frac{1}{\sqrt{N}} \exp\left(-\mathrm{j}\,\frac{2\pi(k-1)(n-1)}{N}\right) = u_N^{-(k-1)(n-1)},\ k,n = 1,\ldots,N \tag{23}$$

where $u_N = \mathrm{e}^{\frac{\mathrm{j}2\pi}{N}}$ the DFT can be written as

$$\tilde{x}[k] = \mathcal{F}[x[n]] = \frac{1}{\sqrt{N}} \sum_{k=1}^{N} x(n) u_N^{-(k-1)(n-1)},\ k = 1,\ldots,N \tag{24}$$

and the IDFT

$$x[n] = \mathcal{F}^{-1}[\tilde{x}[k]] = \frac{1}{\sqrt{N}} \sum_{k=1}^{N} \tilde{x}[k] u_N^{(k-1)(n-1)},\ n = 1,\ldots,N \tag{25}$$

We say $x[n] \iff \tilde{x}[k]$

- In $N \times 1$ vector form

$$\tilde{\mathbf{x}} = \mathbf{U}\mathbf{x} \tag{26}$$

and

$$\mathbf{x} = \mathbf{U}^{-1}\tilde{\mathbf{x}} \tag{27}$$

Appendix II: DFT and Circular convolution

- The circular shift property is the key property to explain the circular convolution property

Property: circular shift

$$g[n] = x[\langle n-m\rangle_N] \Longleftrightarrow \tilde{g}[k] = u_N^{-(m-1)(k-1)}\tilde{x}[k]$$

where $x[\langle n-m\rangle_N]$ is signal $x[n]$ with a circular right wise shift by m

Property: circular convolution

$$y[n] = x[n] \otimes h[n] \Longleftrightarrow \tilde{y}[k] = \sqrt{N}\cdot\tilde{x}[k]\cdot\tilde{h}[k]$$

Proof:

$$\sqrt{N}\tilde{y}[k] = \sum_{n=1}^{N} y[n]u_N^{-(n-1)(k-1)} = \sum_{n=1}^{N}\sum_{m=1}^{N} x[m]h[\langle n-m\rangle_N]u_N^{-(n-1)(k-1)}$$

$$= \sum_{m=1}^{N} x[m]\sum_{n=1}^{N} h[\langle n-m\rangle_N]u_N^{-(n-1)(k-1)} = \{\text{circ. shift prop.}\}$$

$$= \sum_{m=1}^{N} x[m]\cdot\sqrt{N}u_N^{-(m-1)(k-1)}\tilde{h}[k]$$

$$= \sqrt{N}\tilde{h}[k]\sum_{m=1}^{N} x[m]u_N^{-(m-1)(k-1)} = N\tilde{h}[k]\tilde{x}[k]$$