Excercise 1 (4 hours)

Discrete Fourier Transform

1. Observe the module, real and imaginary part of DFT performed on following signals sampled with frequency 8kHz:

   a) x=1, signal length 8 samples
   b) x=sin(2π*1000*t), signal length 8 samples
   c) x=sin(2π*2000*t), signal length 8 samples
   d) x=sin(2π*3000*t), signal length 8 samples
   e) x=sin(2π*4000*t), signal length 8 samples
   f) x=sin(2π*5000*t), signal length 8 samples
   g) x=cos(2π*2000*t), signal length 8 samples
   h) x=cos(2π*4000*t), signal length 8 samples
   i) x=-1, signal length 8 samples
   j) x=1, signal length 16 samples
   k) x=sin(2π*1000*t), signal length 16 samples
   l) x=sin(2π*1000*t+0,5π), signal length 8 samples
   m) x=sin(2π*1000*t), signal length 18 samples
   n) x=sin(2π*1000*t), signal length 20 samples
   o) x=j*sin(2π*2000*t), signal length 8 samples
   p) x=j*cos(2π*2000*t), signal length 8 samples
   q) x=sin(2π*2000*t)+ j*sin(2π*2000*t), signal length 8 samples
   r) x=sin(2π*2000*t)+ j*cos(2π*2000*t), signal length 8 samples

   Present the results in the report as well as conclusions about spectrum parameters as the function of the phase, frequency and amplitude of the signal and the number of periods of the transformed signal.

2. Find the real periodical signal for which the 4-point DFT gives all coefficients equal to 1. Describe the way of evaluating the signal parameters.

3. Using 128-DFT and four window functions (rectangular, Hamming, Blackman and Bartlett) observe and compare modules of DFT coefficients obtained for following signals (sanpling frequency is 8kHz):

   a) x=sin(2π*2000*t)+ sin(2π*2062,5*t)
   b) x=sin(2π*2000*t)+ sin(2π*2125*t)
   c) x=sin(2π*2000*t)+ sin(2π*2250*t)
   d) x=sin(2π*2031*t)

   Compare transform parameters (resolution, sinking).

Excercise 2 (1 hour)

Let us consider a bandpass signal of the center frequency 124 Hz and the bandwidth 12 Hz. Let us assume this signal is formed as the sum of two cosines:

$$x(t)= \cos(2\pi*120*t)+ 2\cos(2\pi*128*t)$$

Generate 1024 samples of this signal according to low-pass as well as band-pass signals sampling rules and transform with DFT. Compare results and discuss them. Are sampling frequencies 29Hz and 35 Hz chosen correctly for this signal? Validate the answer.

Exercise 3 (3 hours)

Amplitude modulation

Useful functions

`y = ammod(x,Fc,Fs,ini_phase,carramp)`

Modulates the carrier frequency Fc (Hz) with message signal X using double sideband amplitude modulation. Sample frequency is Fs (Hz). The modulated signal has `ini_phase` initial phase and carrier amplitude `carramp`.

`z = amdemod(y,Fc,Fs,ini_phase,carramp,num,den)`

AM demodulator demodulates the amplitude modulated signal y from the carrier frequency Fc (Hz). y and Fc have sample frequency Fs (Hz). The modulated signal y has zero initial phase, and zero carrier amplitude, for suppressed carrier modulation. A lowpass filter is used in the demodulation. The default filter is: [NUM,DEN] = butter(5,Fc*2/Fs).

`y = ssbmod(x,Fc,Fs,ini_phase,'upper')`

Modulates the carrier frequency Fc (Hz) with message signal X using single sideband amplitude modulation. Sample frequency is Fs (Hz). The modulated signal has `ini_phase` initial phase and carrier amplitude `carramp`. With the last parameter '*upper*' upper side band is used.

`z = ssbdemod(y,Fc,Fs,ini_phase,num,den)`

SSB demodulator demodulates the amplitude modulated signal y from the carrier frequency Fc (Hz). y and Fc have sample frequency Fs (Hz). The modulated signal y has zero initial phase, and zero carrier amplitude, for suppressed carrier modulation. A lowpass filter is used in the demodulation. The default filter is: [NUM,DEN] = butter(5,Fc*2/Fs).

`Y = AWGN(X,SNR)`

Adds white Gaussian noise to X. The SNR is in dB. The power of X is assumed to be 0 dBW. If X is complex, then AWGN adds complex noise.

Sample

```
Fs = 8000;
Fc = 1000;
t = [0:2047]'/Fs;
x = sin(2*pi*256*t)+2*sin(2*pi*128*t);
y = ammod(x,Fc,Fs,0,1);
y=y+awgn(y,20);
z = amdemod(y,Fc,Fs,0,1);
```

Exercises

1. Generate AM and AM-SC modulated signal. Signal and modulations parameters will be given by the supervisor. Observe spectrum of modulated signals. Compare original and demodulated signals. Perform demodulation for following cases:
    a. There is no noise in the channel, carriers in the modulator and demodulator are identical.
    b. There is Gaussian noise in the channel, carriers in the modulator and demodulator are identical.
    c. There is no noise in the channel, carrier frequencies in the modulator and demodulator differ by 5%.
    d. There is no noise in the channel, carrier phases in the modulator and demodulator differ by 45°.
2. Repeat previous exercise for AM-SSB modulation
3. Compare analyzed modulations with respect to bandwidth utilization, energy utilization efficiency, resistance for distortions.

Exercise 4 (2 hours)

Frequency and phase modulation

Useful functions

```
y = fmmod(x,Fc,Fs,freqdev,ini_phase)
```
FMMOD uses the message signal x to modulate the carrier frequency Fc (Hz) and sample frequency Fs (Hz), where Fs > 2*Fc. FREQDEV (Hz) is the frequency deviation of the modulated signal. INI_PHASE specifies the initial phase of the modulation.

```
z = fmdemod(y,Fc,Fs,freqdev,ini_phase)
```
FMDEMOD demodulates the FM modulated signal Y at the carrier frequency Fc (Hz). Y and Fc have sample frequency Fs (Hz). FREQDEV is the frequency deviation (Hz) of the modulated signal. INI_PHASE specifies the initial phase of the modulation.

```
y = pmmod(x,Fc,Fs,phasedev,ini_phase)
```
PMMOD uses the message signal x to modulate the carrier frequency Fc (Hz) and sample frequency Fs (Hz), where Fs > 2*Fc. FREQDEV (Hz) is the frequency deviation of the modulated signal. INI_PHASE specifies the initial phase of the modulation.

```
z = pmdemod(y,Fc,Fs,freqdev,ini_phase)
```
PMDEMOD demodulates the PM modulated signal Y at the carrier frequency Fc (Hz). Y and Fc have sample frequency Fs (Hz). FREQDEV is the frequency deviation (Hz) of the modulated signal. INI_PHASE specifies the initial phase of the modulation.

Sample
```
Fs = 8000;
Fc = 3000;
t = [0:Fs]'/Fs;
x = sin(2*pi*300*t)+2*sin(2*pi*600*t);
dev = 50;
y = fmmod(x,Fc,Fs,dev);
z = fmdemod(y,Fc,Fs,dev);
```

Exercises

1. Generate FM modulated signal. Signal and modulations parameters will be given by the supervisor. Observe spectrum of modulated signals. Compare original and demodulated signals. Perform demodulation for following cases:
    a. There is no noise in the channel, carriers in the modulator and demodulator are identical.
    b. There is Gaussian noise in the channel, carriers in the modulator and demodulator are identical.
    c. There is no noise in the channel, carrier frequencies in the modulator and demodulator differ by 5%.
    d. There is no noise in the channel, carrier phases in the modulator and demodulator differ by 45°.

2. Repeat previous exercise for PM modulation
3. Compare analyzed modulations with respect to bandwidth utilization, energy utilization efficiency, resistance for distortions.

Exercise 5 (3 hours)

Digital modulations

Useful functions:

```
h = modem.qammod(M)
```
M-value QAM baseband modulator definition.

```
h = modem.pammod(M)
```
M-value PAM (ASK) baseband modulator definition.

```
h = modem.pskmod(M)
```
M-value PSK baseband modulator definition.

```
h = modem.qamdemod(M)
```
M-value QAM baseband demodulator definition.

```
h = modem.pamdemod(M)
```
M-value PAM (ASK) baseband demodulator definition.

```
h = modem.pskdemod(M)
```
M-value PSK baseband demodulator definition.

```
y=modulate(h, x)
```
Baseband *x* signal modulation with *h* modulator.

```
y=modulate(x1, Fc, Fs, 'mod_type',x2)
```
MODULATE modulates the message signal x1 with a carrier frequency Fc and sampling frequency Fs, using the modulation scheme in MOD_TYPE (use 'qam'). For QAM modulation x2 describes imaginary part of modulated signal and x1 describes real part of modulated signal. Fs must satisfy Fs > 2*Fc + BW, where BW is the bandwidth of the modulated signal.

```
y=demodulate(h, x)
```
Baseband *x* signal demodulation with *h* modulator.

```
[y1 y2]=demod(x, Fc, Fs, 'mod_type')
```
DEMOD demodulates the message signal x with a carrier frequency Fc and sampling frequency Fs, using the modulation scheme in MOD_TYPE (use 'qam'). For QAM modulation y2 describes imaginary part of demodulated signal and y1 describes real part of demodulated signal. Fs must satisfy Fs > 2*Fc + BW, where BW is the bandwidth of the modulated signal.

```
[a,b] = biterr(x, z)
```
BITERR compares the unsigned binary representation of the elements in the two vectors *x* and *y*. The number of differences in the binary representation is output in *a*. The ratio of *a* to the total number of bits used in the binary representation is output in *b*.

```
[a,b]= symerr(x, z)
```

SYMERR compares the elements in the two vectors x and y. The number of differences is output in *a*. The ratio of *a* to the total number of elements is output in *b*.

```
out=randint(m, n, r)
```
Generates *m*x*n* matrix with random integers within the range [0;*r*-1].

```
scatterplot(x)
```
Generates the scatter plot for signal x.

Sample 1
```
M = 16;
x=randint(20,1,16);
Fs=1000;
Fc=100;
Fd=10;
FsDFd=Fs/Fd;
tmp = x(:, ones(1, FsDFd))';
yy = tmp(:);
h=modem.qammod('M',16);
y=modulate(h,yy);
z=modulate(real(y),Fc,Fs,'qam',imag(y));
```

Sample 2
```
x = randint(5000,1,M);
Fs=1000;
Fc=100;
Fd=10;
FsDFd=Fs/Fd;
tmp = x(:, ones(1, FsDFd))';
yy = tmp(:);
h=modem.qammod('M',16);
y=modulate(h,yy);
ynoisy = awgn(y,20);
phasenoise = randn(5000*FsDFd,1)*.015;
ynoisy = ynoisy.*exp(j*2*pi*phasenoise);
scatterplot(ynoisy);
hd=modem.qamdemod('M',16);
z=demodulate(hd,ynoisy);
[num,rt]= symerr(x,z)
```

Exercises

1. Observe modulated signal *x* in the baseband. Compare symbols parameters. Modulation parameters will be given by the supervisor.
    a. 4-ASK for *x*={0,2,1,3}
    b. 4-PSK for *x*={0,2,1,3}
    c. 8-QASK for *x*={0,2,1,3,7,5,4,6}

2. Transmit signals from point 1 over 10kHz channel with carrier frequency 5kHz. Assume symbol rate 200 Hz and 1000 Hz. Observe parameters of generated symbols. Compare frequency spectrums of generated signals.
3. Observe constellations for 16-QASK, 8-PSK i 8-ASK modulations. Observe phase jitter and the noise influence on these diagrams. Evaluate maximum theoretical values of noise and phase jitter. Comment results.

Exercise 6 (3 hours)

Noise in the channel

1. Design the demodulator for given by the supervisor modulation type.
2. Plot relationship between SNR in the channel and bit error rate for given modulation for few sets of Fc and Fd values.
3. Find maximum error-free transmission speed for this modulation and next find SNR resulting in bit error rate equal to 20%. Next decrease transmission speed till bit error rate < 0.01%. Comment results.