

Innlevering4

Kap2; Oppgave1:

```
%definerer variabler
x1 = [4, -1, 0, 5, 0];
x2 = [6, -1, 7, -1, 1];
x3 = [-4, 3, 2, 1, 0];

%konvolusjon
resultat1 = conv(x1, x2 + x3)
resultat2 = conv(x1, x2) + conv(x1, x3)

%sammenligner resultatene
if isequal(resultat1, resultat2)
    disp("konvolusjon er distributiv.")
else
    disp("Konovlusjon er ikke distributiv.")
end
```

output:

```
>> innlevering41

resultat1 =

     8     6    34     1    14    44     0     5     0

resultat2 =

     8     6    34     1    14    44     0     5     0

konvolusjon er distributiv.
```

Oppgave2:

```
%digitalt filter
x11 = [10,14,2,6,8,20,1,17,14,7,15,16];

%legger til nuller i begynnelsen av sekvensen
x = [zeros(1,3), x11];

%ikke-rekursivt filter
```

```

y_ikke_rekursiv = zeros(1, length(x));
for n= 3:length(x)
    y_ikke_rekursiv(n) = 1/3*(x(n)+x(n-1)+x(n-2));
end

%Rekursivt filter
x = [zeros(1,3), x11];
y_rekursiv = zeros(1, length(x));
for n = 4:length(x)
    y_rekursiv(n) = y_rekursiv(n-1) + 1/3*(x(n)-x(n-3));
end

disp(y_ikke_rekursiv);
disp(y_rekursiv);

```

output:

```

>> innlevering41
      0      0      0      3.3333      8.0000      8.6667      7.3333      5.3333      11.3333      9.6667      12.6667      10.6667      12.6667      12.0000      12.6667
      0      0      0      3.3333      8.0000      8.6667      7.3333      5.3333      11.3333      9.6667      12.6667      10.6667      12.6667      12.0000      12.6667

```

Oppgave3

```

Ts = 1.25*10^-3;
T = 1/100;
Fs = 1/Ts;
t = 0:1/Fs:T;

%definerer x_t
x_t = 2*cos(2*pi*100*t);

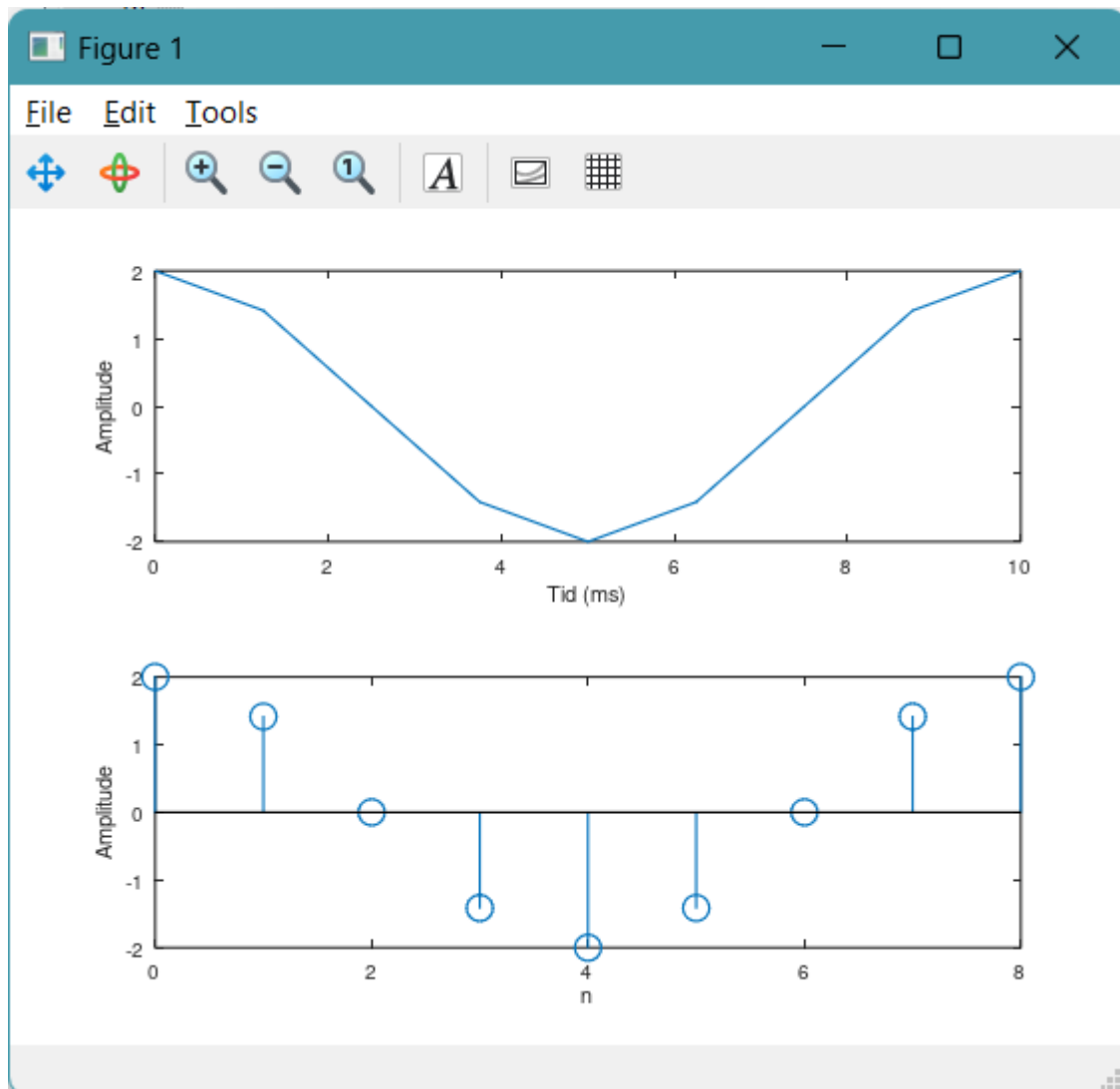
%punktprøve
n = 0:length(t)-1;
x_n = x_t(n+1);

%plotter x_t
subplot(2,1,1);
plot(t*1000, x_t);
xlabel('Tid (ms)');
ylabel('Amplitude')

%plotter x_n
subplot(2,1,2);
stem(n, x_n);
xlabel('n');
ylabel('Amplitude');

```

output:



Oppgave4

kode

```
%Oppgave 4
%Forklar sammenhengen mellom N og M verdier og frekvensspekteret ser ut.
N=20; n=[0:N-1]; M=4;
x=sin(2*pi/M*n);
X=fft(x); % Transform til frekvensdomenet (DFT gir komplekst resultat)
y=ifft(X); % Skriv eventuelt ut for å vise at inverstransformen y blir lik x
% Plot signalet i tidsdomenet (opprinnelig signal og invers-transform)
figure(2);
subplot(2,1,1);
stem(n,x);
title('Opprinnelig signal');
grid;
subplot(2,1,2);
stem(n,real(y));
```

```

title('Invers-transform');
grid;
% Bruk real(y) om numerisk unøyaktighet gir noen små imaginære komponenter
% Kunne også brukt absoluttverdien, siden imaginærdelen evt. er meget liten.
% (Vi vet at vi skal tilbake til den reelle sekvensen vi startet med.)
% Plot frekvensspekteret (magnitudo og fase)
figure(3);
subplot(2,1,1);
stem(n,abs(X));
ylabel('Magnitudo');
grid;
title('Diskret Fourier Transform (fft)');
subplot(2,1,2);
stem(n,angle(X)*180/pi);
ylabel('Fase [deg]');
grid;

% Prøv også med sekvensene nedenfor
x1=[5, 4, -1, -2, 3, 2, 2, -2]; % tilfeldig sekvens
x2=[1, 1, 1, 1, 1, 1]; % konstant signal
x3=ones(1,20); % konstant signal
n1=[0:length(x1)-1];
n2=[0:length(x2)-1];
n3=[0:length(x3)-1];
% Transformer, inverstransformer, og plot resultatene slik som ovenfor.
X1 = fft(x1);
Y1 = ifft(X1);
X2 = fft(x2);
Y2 = ifft(X2);
X3 = fft(x3);
Y3 = ifft(X3);

figure(4);
subplot(3,2,1);
stem(n1,x1);
grid;

subplot(3,2,2);
stem(n1, real(Y1));
grid;

subplot(3,2,3);
stem(n1,abs(X1));
grid;

subplot(3,2,4);
stem(n1, angle(X1)*180/pi);
grid;

%plot for x2
figure(5);
subplot(3,2,1);
stem(n2,x2);
grid;

subplot(3,2,2);
stem(n2, real(Y2));
grid;

```

```

subplot(3,2,3);
stem(n2,abs(X2));
grid;

subplot(3,2,4);
stem(n2, angle(X2)*180/pi);
grid;

%plot for x3
figure(6);
subplot(3,2,1);
stem(n3,x3);
grid;

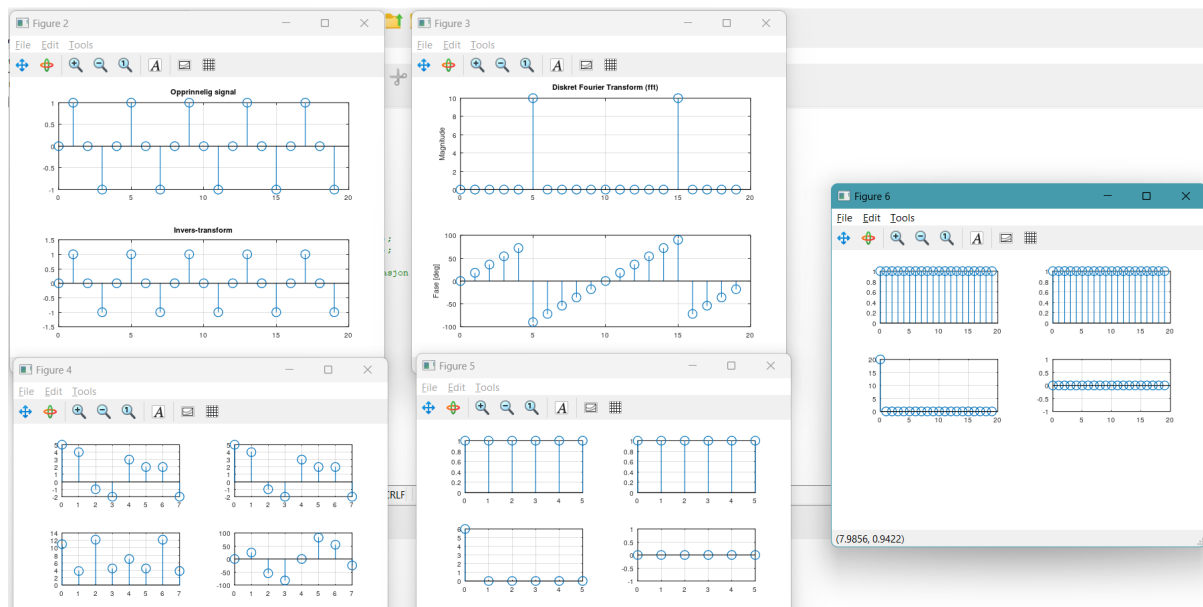
subplot(3,2,2);
stem(n3, real(Y3));
grid;

subplot(3,2,3);
stem(n3,abs(X3));
grid;

subplot(3,2,4);
stem(n3, angle(X3)*180/pi);
grid;

```

output



Oppgave5

kode

```

%oppgave5
fs = 10000; %10kHz
#k = 12;
N = 100;
#fk = fs*k/N;

%tidskonstant
t = (1:N)/fs;

%signaler
signal1 = cos(2*pi*1200*t);
signal2 = cos(2*pi*1250*t);

%Diskre fourier transformasjon
DFT1 = fft(signal1);
DFT2 = fft(signal2);

%frekvenser
f = (0:N-1)*fs/N;

%plot
figure(7);
plot(f, abs(DFT1));
grid;
hold on;
plot(f, abs(DFT2));
hold off;

```

output:

