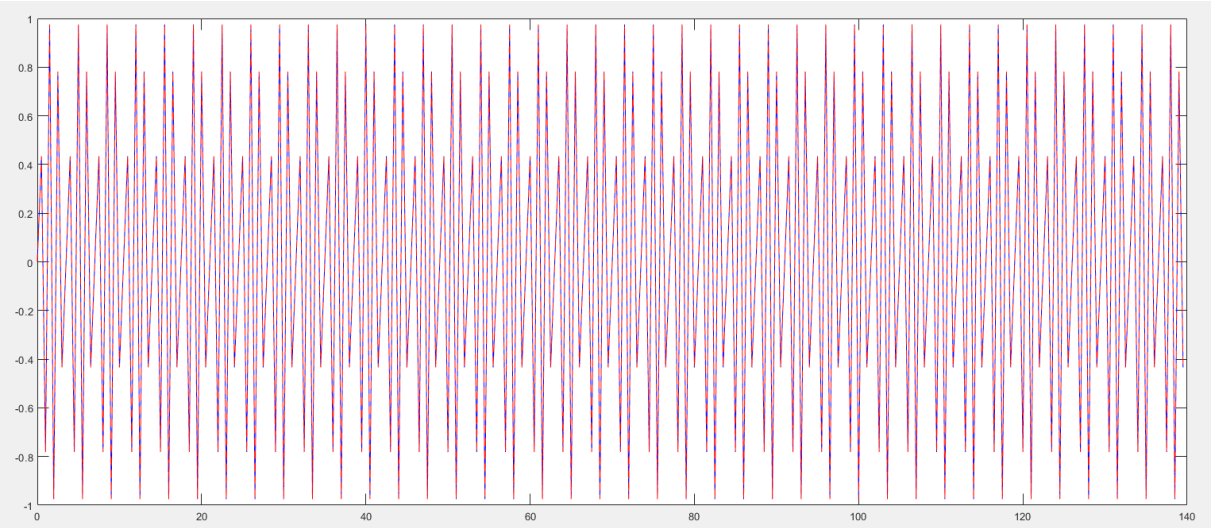
Andrzej Żaba nr indeksu: 401490 Lab 5.

Zad1.

```
clc
clear all
close all
%% declaration of frequency vector
Fs = 140;
dt = 1/Fs;
Tk = 2;
t = 0:dt:(Tk-dt);

L = length(t)

df = Fs / L;

fvec = (0:L-1)*df;
%%
f = 60;
x = sin(2*pi*f*t);

transform = fft(x);
inverseTransform = ifft(transform);

plot(fvec,x,'b')
hold on
plot(fvec,inverseTransform,'r--')

maxBlad = 0;
for k = 1:L
   blad(k) =  abs(x(k) - inverseTransform(k));
  % if(blad(k) > maxBlad)
  %      maxBlad = blad(k);
  % end
end

maxBlad = max(blad)
```

```
maxBlad =

    4.4176e-29
```



Obserwacje i wnioski:

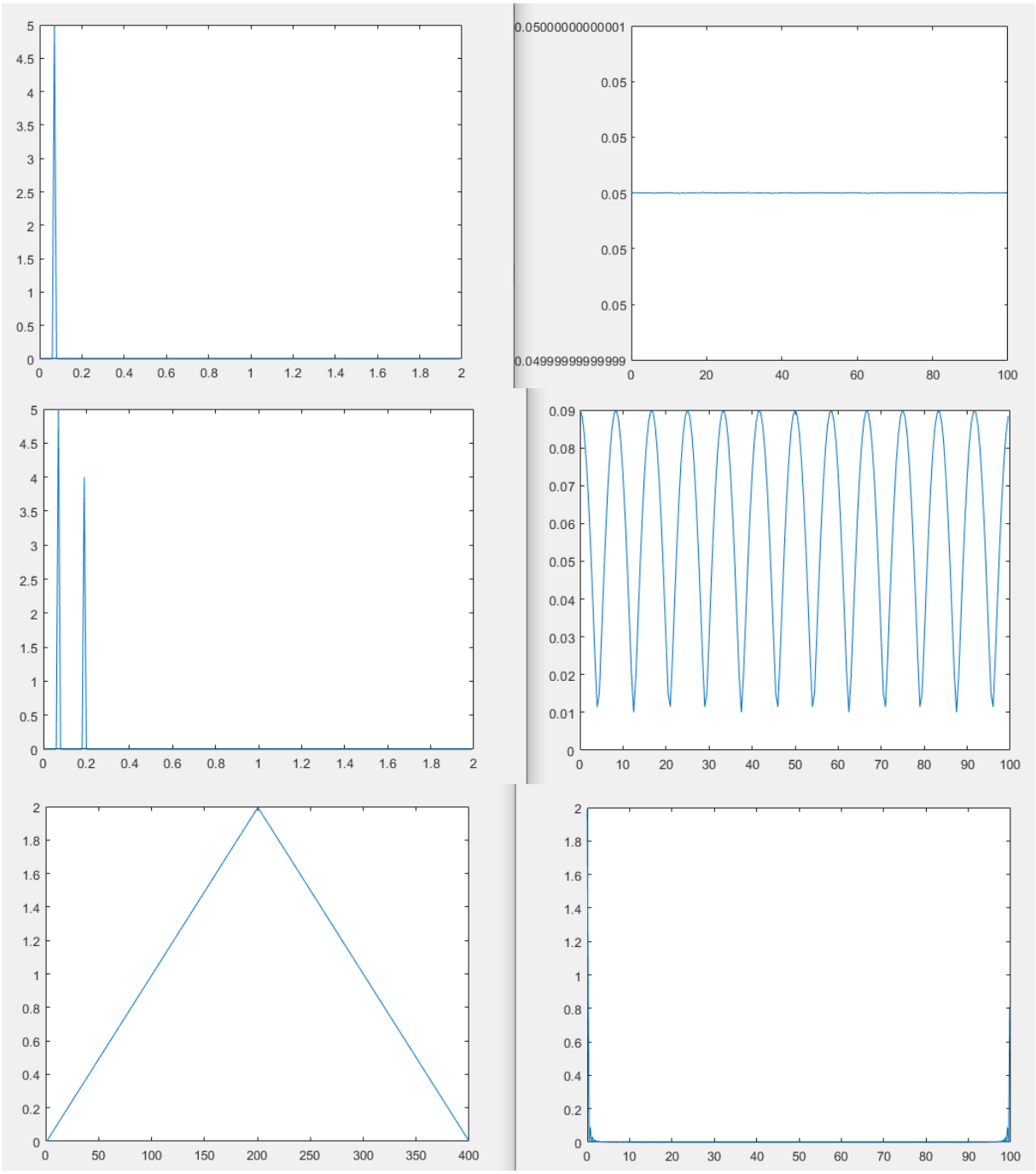
Sygnał zmodyfikowany najpierw transformatą Fouriera a potem odwrotną transformatą Fouriera niemal dokładnie pokrywa się z sygnałem początkowym . Analizując też otrzymany maksymalny błąd możemy zauważyć, iż rozbieżności są naprawdę znikome – rzędu 10^(-29).

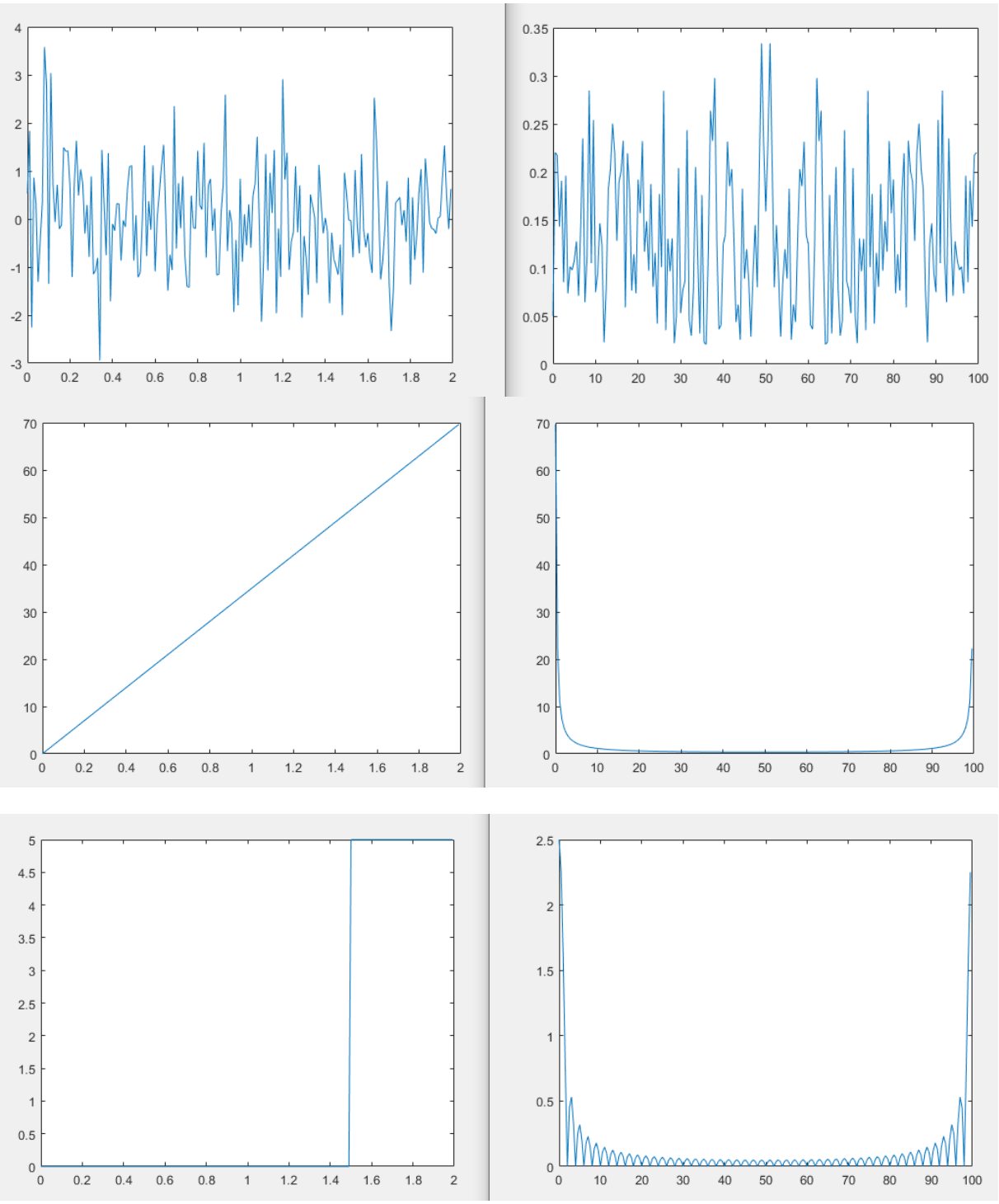
Zad2.

```
clc
clear all
close all
%% Declaration of time vector
Tk = 2;
dt = 0.1;
t=0:dt:Tk;
L = length(t);

%% Declaration of frequancy vector
Fs = 100;
dt = 1/Fs;
Tk = 2;
t = 0:dt:(Tk-dt);
L = length(t)
df = Fs / L;
fvec = (0:L-1)*df;


%% Impulse
X1 = zeros(1,L);
```

```
X1(8) = 5;

figure(1)
plot(t,X1)
figure(11)
plot(fvec,2*abs(fft(X1))/ L )

%% Two Impulses

X2 = zeros(1,L);
X2(8) = 5;
X2(20) = 4;

figure(2)
plot(t,X2)
figure(12)
plot(fvec,2*abs(fft(X2))/ L )

%% Noise
X3 = zeros(1,L);

for k = 1:L
    X3(k) = randn()
end

figure(3)
plot(t,X3)
figure(13)
plot(fvec,2*abs(fft(X3))/ L )

%% triangle signal


X4 = t;
X4 = [X4 fliplr(X4)];
figure(7)
plot(X4)

t1=1:dt:(Tk-dt)+1;
tt = [t t1];

N2 = length(tt)
df2=Fs/N2
fvec2=df2*(0:N2-1)
figure(17)
```

```
plot(fvec2,2*abs(fft(X4))/ N2 )

signal1 = triang(length(X4));
signal2 = repmat(signal1,10,1);
figure(4)
plot(signal2)
%figure(14)
%plot(fvec,2*abs(fft(signal2))/ L )



%% Ramp
X5 = 35 * t;
figure(5)
plot(t,X5)

figure(15)
plot(fvec,2*abs(fft(X5))/ L )
%% Skok jednostkowy

X6 = zeros(1,L);
X6(t>=1.5) = 5;

figure(6)
plot(t, X6)
figure(16)
plot(fvec,2*abs(fft(X6))/ L )
```


0.05000000000001
0.04999999999999

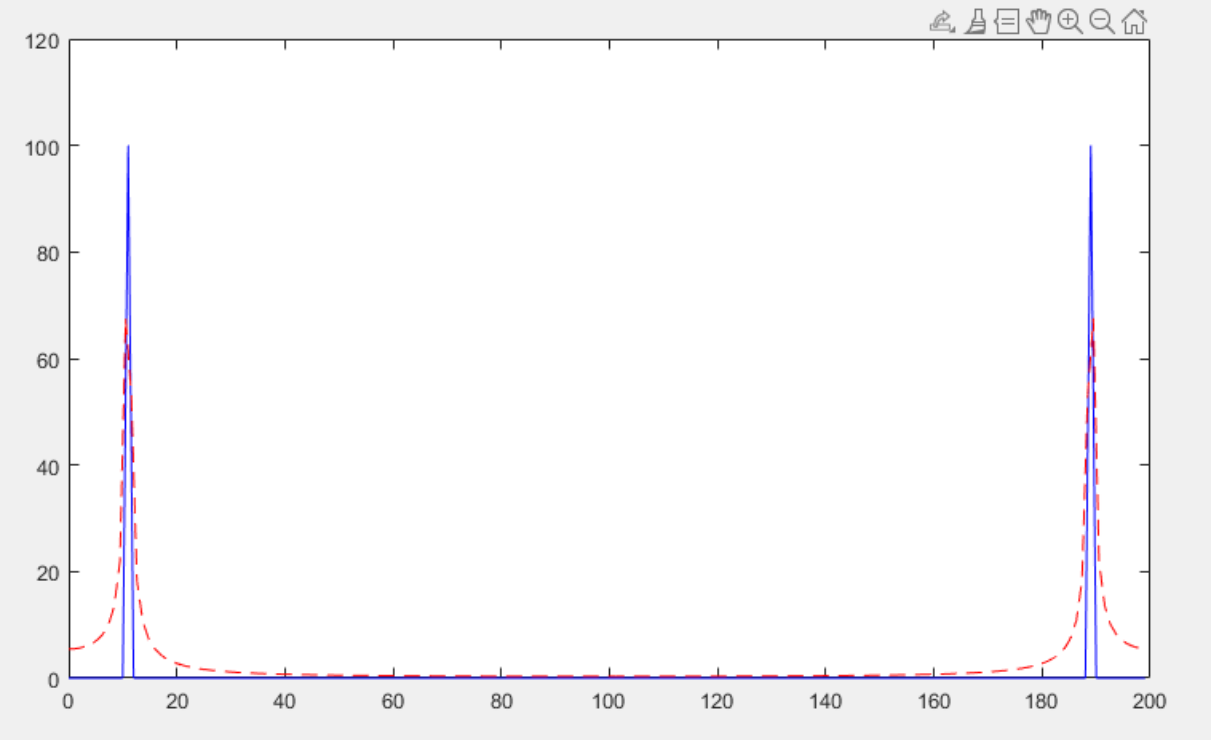


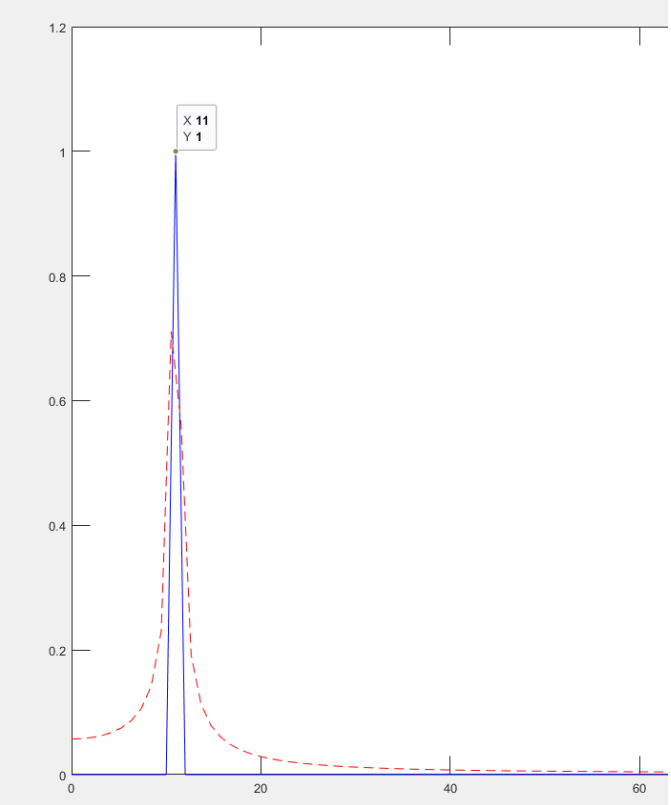
Wszystkie wykresy wydają się być wygenerowane poprawnie.

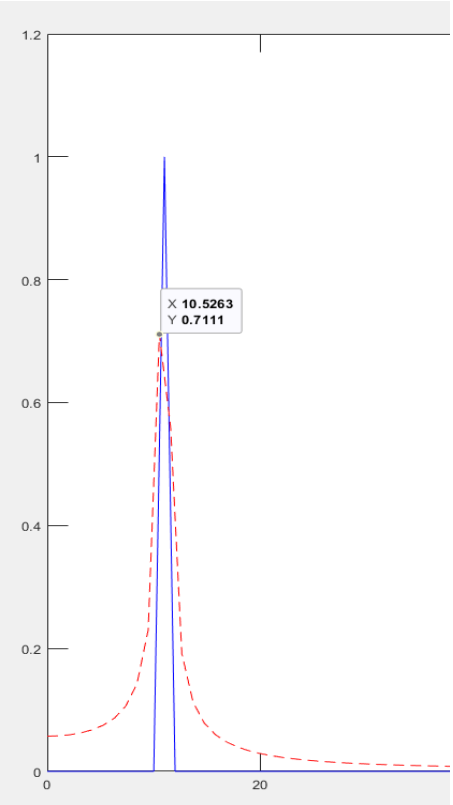
Zad3

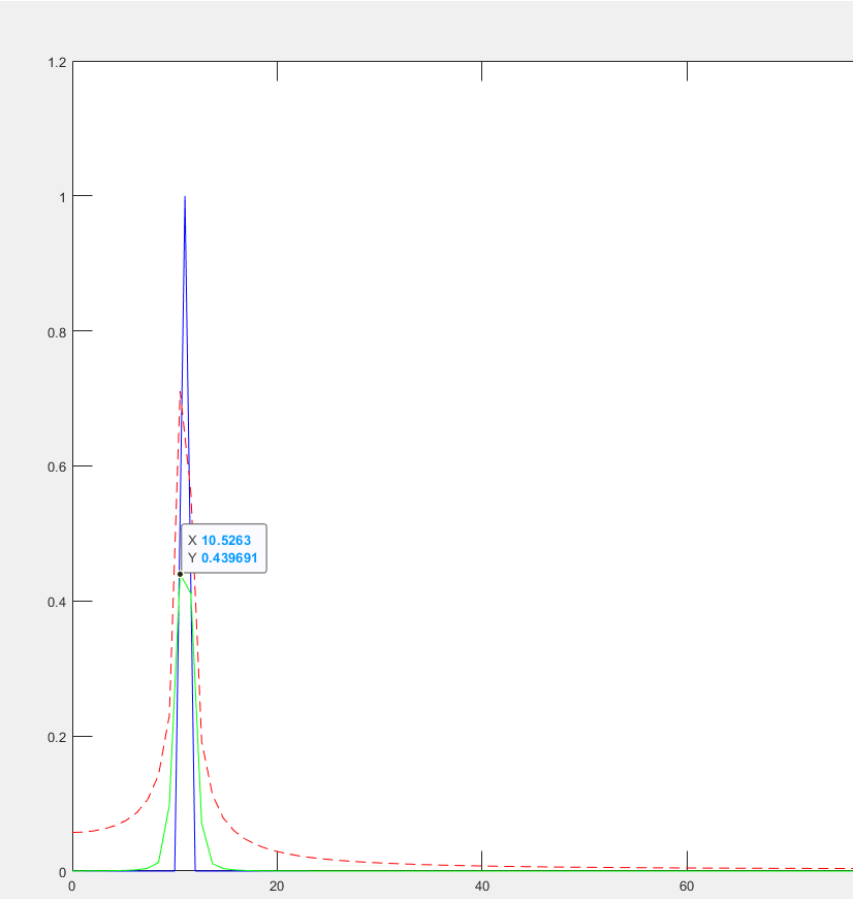
```
clc
clear all
close all
%% Declaration of transformed vector 1
Tk = 1;
Fs = 200
```

```
dt = 1/Fs;
t = 0:dt:(Tk-dt);

L = length(t)

df = Fs / L;

fvec = (0:L-1)*df;

f=11;
S = sin(2*pi*f*t);

figure(1)
transform1 = abs(fft(S))
plot(fvec,2 * transform1 / L,'b')
hold on

%% Declaration of transformed vector 2
Tk = 0.95;
dt = 1/Fs;
t = 0:dt:(Tk-dt);
L = length(t)
df = Fs / L;
fvec = (0:L-1)*df;
S = sin(2*pi*f*t);

transform2 = abs(fft(S))
plot(fvec,2 * transform2 / L,'r--')
%hold off
%% Hanning window implementation
okno = hanning(length(S))
size(okno)
sygnalPoOknie = okno.*S'

figure(2)
plot(t,S,"b--"); hold on
plot(t,sygnalPoOknie,"r")
plot(t,okno,'k:')


figure(1)
plot(fvec, 2*(abs(fft(sygnalPoOknie))) / L ,'g')
```


120
100
80
60
40
20
0
0
20
40
60
80
100
120
140
160
180
200

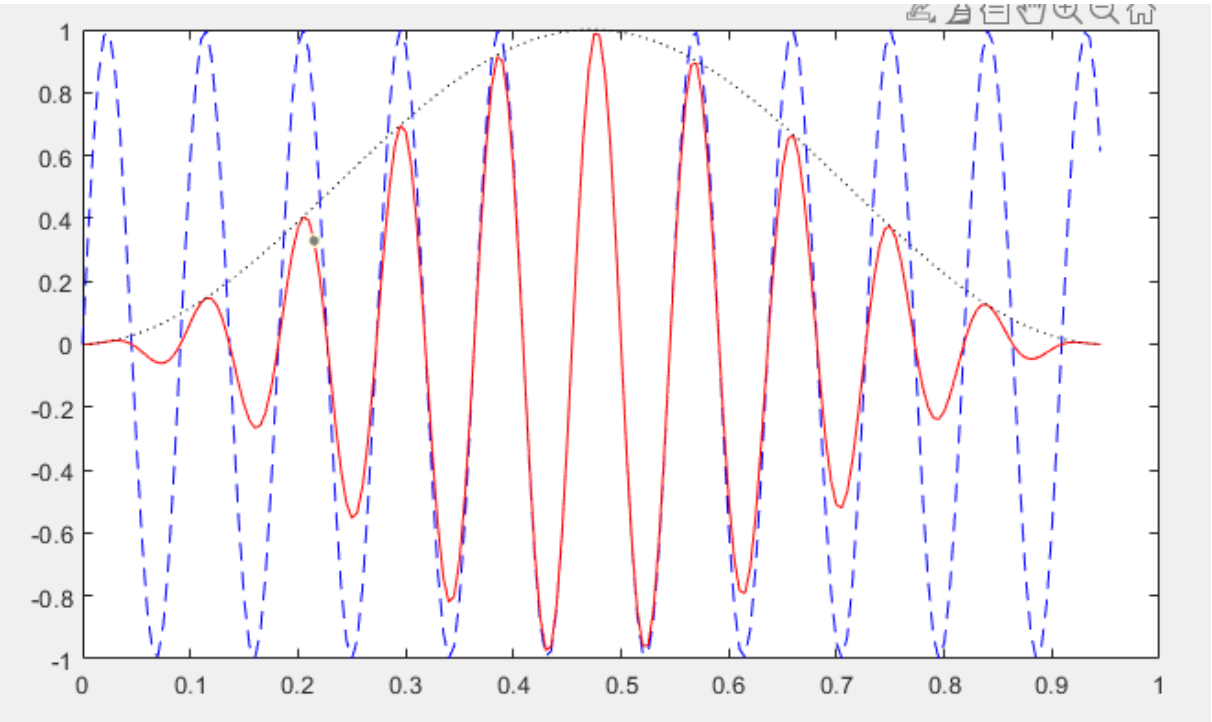

X 11
Y 1
1.2
1
0.8
0.6
0.4
0.2
0
0
20
40
60


X 10.5263
Y 0.7111
1.2
1
0.8
0.6
0.4
0.2
0
0
20



Komentarz:

Jak widać na powyższym wykresie, po przekształceniu Fouriera sygnału trwającego 1s widzimy, że składa się on w całości z jednej częstotliwości wynoszącej 11Hz. Reszta punktów przyjmuje na tyle marginalne wartości, że spokojnie możemy je pominąć. Natomiast jeżeli chodzi o sygnał trwający 0.95s to po przekształceniu widzimy, że jest on szerzej rozdystrybuowany na osi X. Tzn. składa się z całego spektra częstotliwości z których największy udział w sygnale ma częstotliwość ok. 10.5 Hz i stanowi ona ok. 71% sygnału.



Niebieski – sygnał oryginalny – 0.95s

Czerwony – sygnał po zastosowaniu okna Hanninga

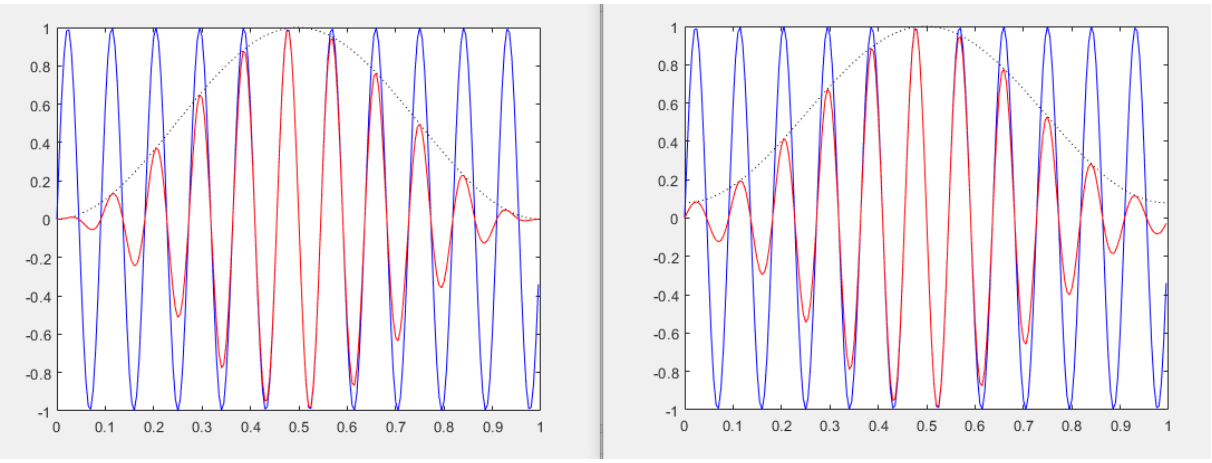
Czarny – zakres okna Hanninga

Jak widzimy na powyższym wykresie po implementacji okna Hanninga nasz sygnał z równej sinusoidy zamienił się w sygnał który wraz z czasem zaczyna nabierać na sile po czym po osiągnięciu maksimum zaczyna w podobny sposób maleć. Jest to efekt zastosowanego okna, które dostosowuje sygnał do swojego kształtu zamieniając przy tym wartości w pikach sygnału na odpowiednio dostosowane do swojej postaci. Może ono pełnić rolę filtra.

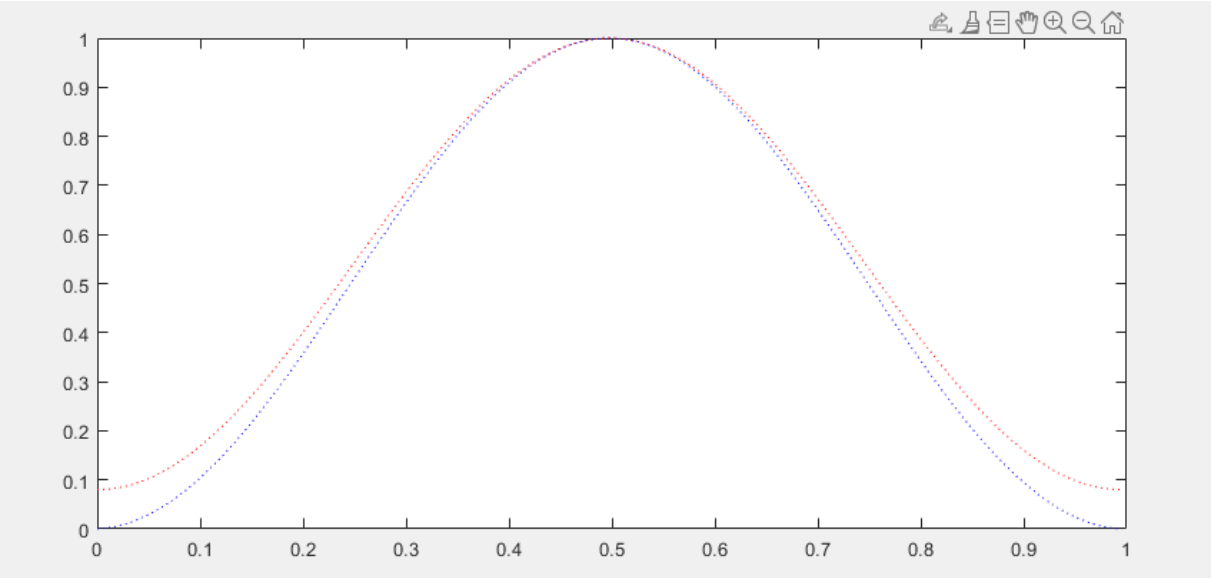
Zad4.

```
clc
clear all
close all
%% Declaration of basic parameters and signal
% taken directly form taks 3
Tk = 1;
Fs = 200
dt = 1/Fs;
t = 0:dt:(Tk-dt);
L = length(t)
df = Fs / L;
fvec = (0:L-1)*df;
```

```
f=11;

S = sin(2*pi*f*t);

figure(1)
plot(t,S,'b')
hold on
%% Hanning's Window
hann = hanning(length(S));
size(hann);
AfterHanning = hann.*S';

plot(t,hann,'k:')
plot(t,AfterHanning,'r')


figure(2)
plot(t,S,'b'); hold on
%% Hamming's Window
hamm = hamming(length(S))
size(hamm)
AfterHamming = hamm.*S'

plot(t,hamm,'k:')
plot(t,AfterHamming,'r')


figure(3)
plot(t,hann,'b:')
hold on
plot(t,hamm,'r:')
```



Jak widzimy na powyższych wykresach sygnał S po zastosowaniu okien Hanninga (pierwszy) oraz Hamminga (drugi) różnią się nieco od siebie.



Natomiast powyżej mamy nałożone na siebie oba okna:

okno Hanninga – niebieskie,
okno Hamminga – czerwone

Oba okna zaczynają się i kończą w tych samych miejscach – na końcach badanego przedziału. Jednak okno Hanninga zaczyna się i kończy od wartości zero , natomiast okno Hamminga zaczyna się i kończy nieco wyżej bo od wartości 0.08. Okno Hammina nieco łagodniej zmierza do swojego maksimum, natomiast okno Hanninga jest bardziej strome. Oba okna maja swoje maksimum w tym samym punkcie – w środku przedziału.

Przy implementacji okna Hanninga na sygnał po transformacie Fouriera powinniśmy otrzymać bardziej konkretne wartości sygnału niż w przypadku okna Hamminga. Tzn. przy oknie Hamminga mogą zostać uwzględnione poboczne wartości które będą miały niewielki, ale jednak udział w całości sygnału. Może to prowadzić do większej ilości szumów i zakłóceń.