**Obraz zawierający logo, tekst, Czcionka, Grafika

Opis wygenerowany automatycznie**

**Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie**

**WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI,**

**INFORMATYKI I INŻYNIERII BIOMEDYCZNEJ**

Raport

**Podstawy telekomunikacji**

Autor: Grzegorz Lis, Karolina Sawosz

Kierunek studiów: Mikroelektronika w Technice i Medycynie

# Kraków, 2023

**Laboratorium 1**

***Opis problemu***

Celem laboratoriów było zbudowanie sygnału prostokątnego za pomocą harmonicznych sinusa, dobranie odpowiedniej częstotliwości próbkowania sygnałów, obliczenie transformat Fouriera (dyskretnej i odwrotnej) oraz wyliczenie splotów liniowego oraz kołowego, zarówno analitycznie, jak i za pomocą programu.

***Podstawy matematyczne rozwiązania***

Discrete Fourier Transform:

Splot liniowy:

Splot kołowy:

***Symulacje i obserwacje***

Zadanie 1

Listing 1. Wyświetlenie sygnału prostokątnego za pomocą harmonicznych sinusa.

for i = 1:2:harmonics

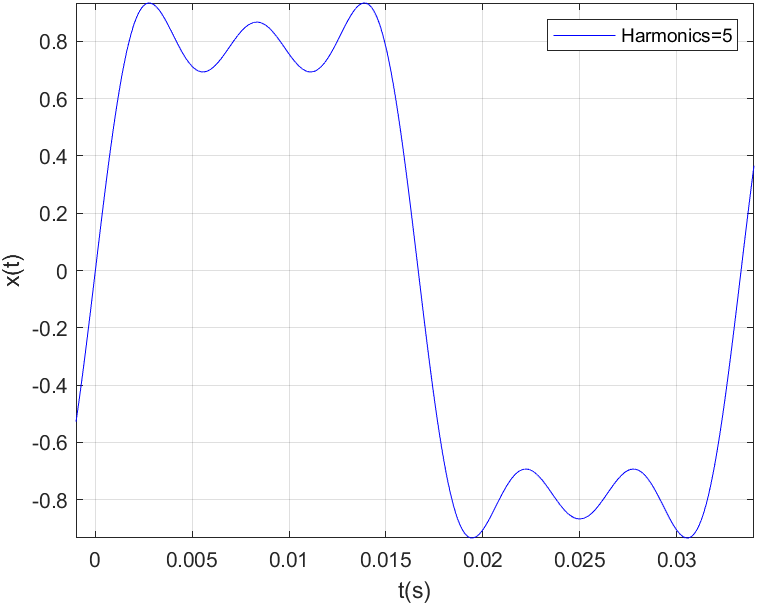
x = A\*sin(2\*pi\*f\*t\*i)/i;

pulse = pulse + x;

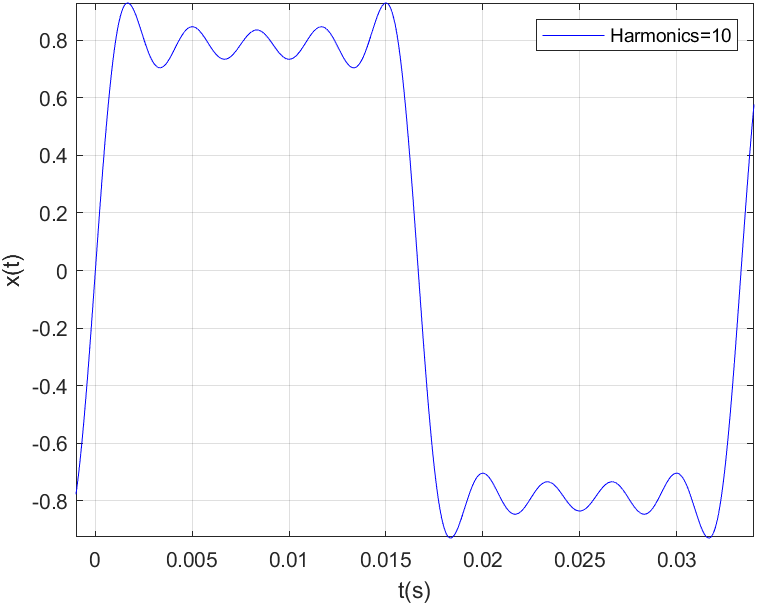
end

Obraz zawierający tekst, Wykres, diagram, linia

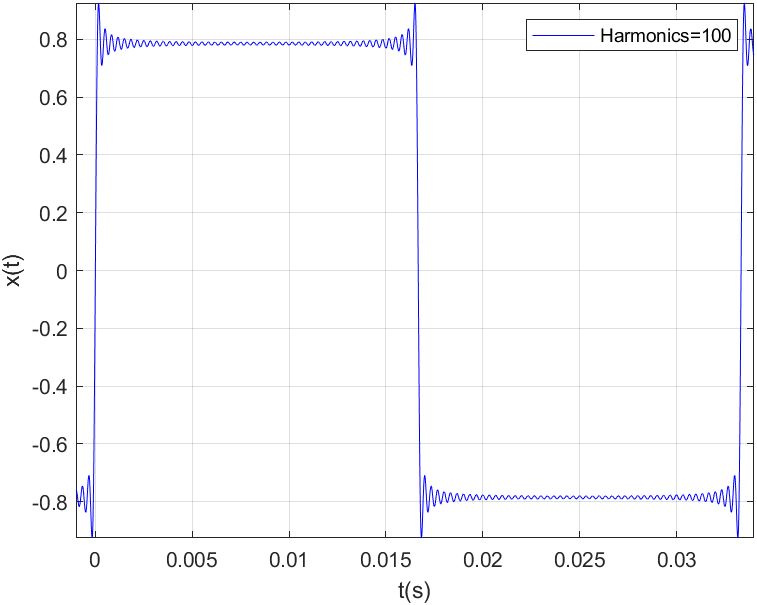
Opis wygenerowany automatycznie

Rysunek . Sygnał prostokątny przedstawiony za pomocą 3 harmonicznych sinusa. 

Rysunek . Sygnał prostokątny przedstawiony za pomocą 5 harmonicznych sinusa.



Rysunek . Sygnał prostokątny przedstawiony za pomocą 10 harmonicznych sinusa.



Rysunek . Sygnał prostokątny przedstawiony za pomocą 100 harmonicznych sinusa.

Obserwując wykresy, można zauważyć, że im więcej harmonicznych sygnału sinusoidalnego, tym bardziej sygnał staje się prostokątny i tym ostrzejsze staje się zbocze.

Zadanie 2

a) Listing 2. Dobrana częstotliwość próbkowania i DFT podanego sygnału.

Tmin = 0;

Tmax = 2;

dt = 0.05;

t = Tmin:dt:Tmax;

x = 5 + 2\*cos(2\*pi\*t - (pi/2)) + 3\*cos(4\*pi\*t);

y = dft(x);

function y = dft(x)

y = zeros(size(x));

for i=0:(length(x)-1)

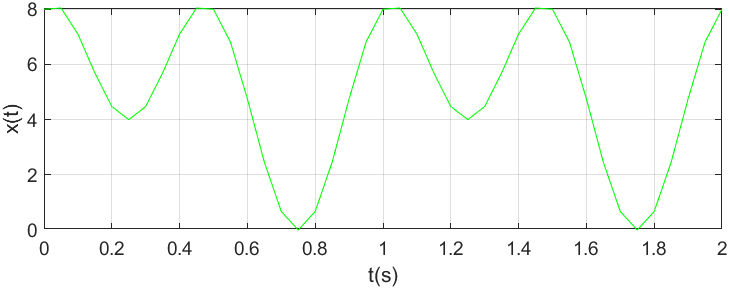
for j=0:(length(x)-1)

y(i+1) = y(i+1) + x(j+1)\*exp(-1i\*2\*pi/length(x)\*i\*j);

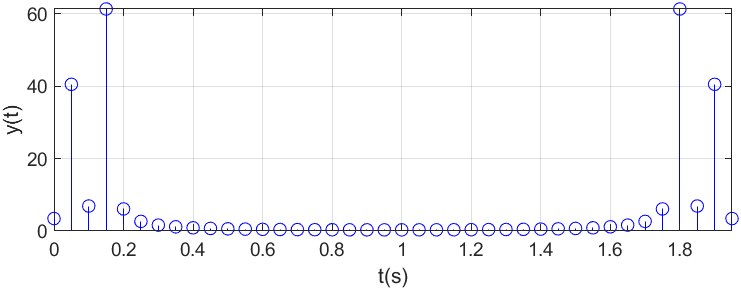
end

end

end



Rysunek . Spróbkowany sygnał cosinusa.



Rysunek . Widmo DFT sygnału cosinusa.

Czas obserwacji wynosi dwa pełne okresy sygnału. Po zaobserwowaniu okresowości można założyć, że sygnał będzie powtarzalny, skąd nie ma potrzeby obserwacji przez dłuższy czas.  
W przypadku krótszego czasu można niepoprawnie określić okres sygnału, co w następstwie skutkuje sfałszowaniem wyników. Dobrana częstotliwość wynosi 20Hz, co jest częstotliwością optymalną.

b) Listing 3. Implementacja przetwornika SAR.

for j = 1:length(t)%SAR ADC

adc = min\_range;

for i = 1:1:n\_bits

adc = adc + range/(2^i);

adc\_output(j,i)=1;

if adc>x(j)

adc = adc - range/(2^i);

adc\_output(j,i)=0;

end

end

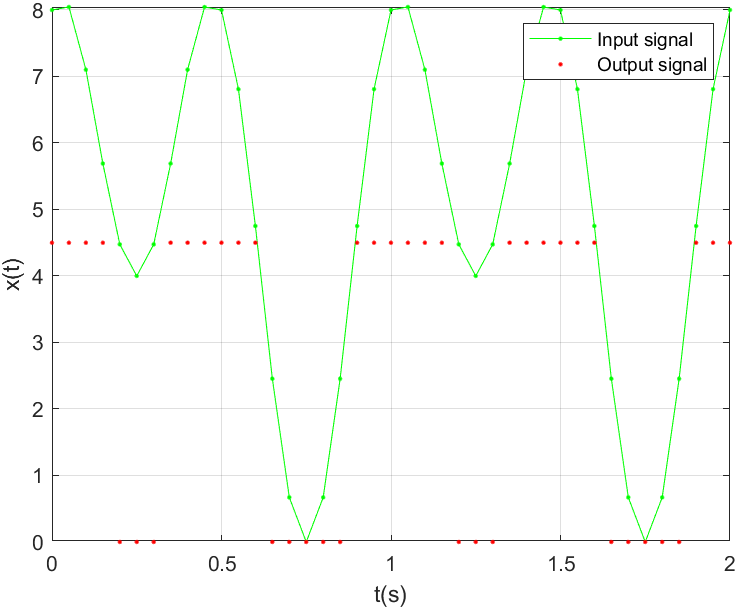
if max\_quantization<(x(j)-adc)

max\_quantization = x(j)-adc;

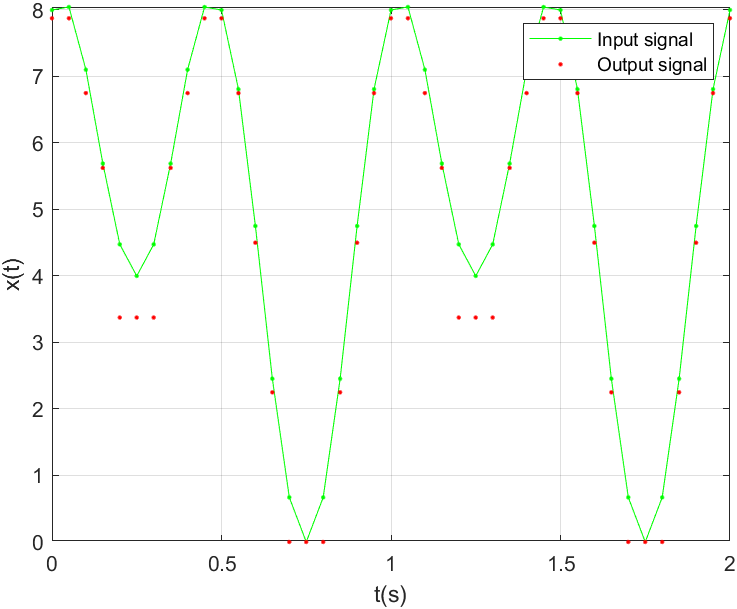
end

adc\_plot(j) = adc;

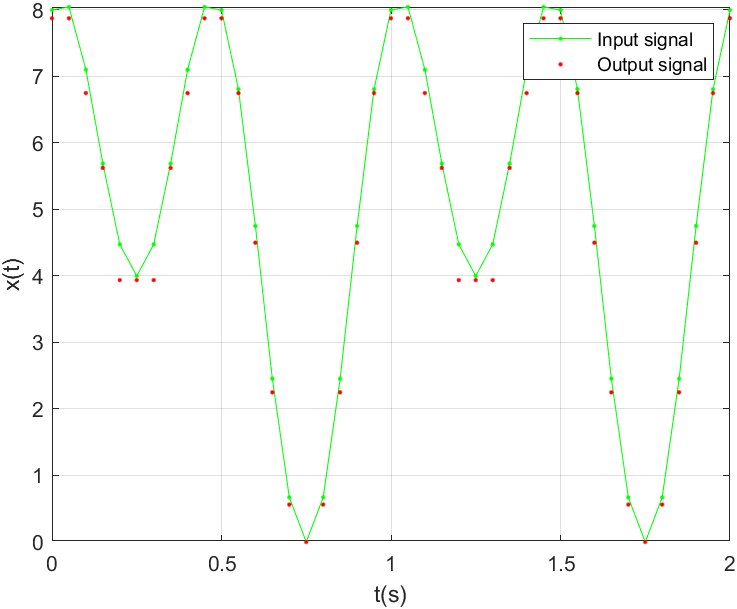
end



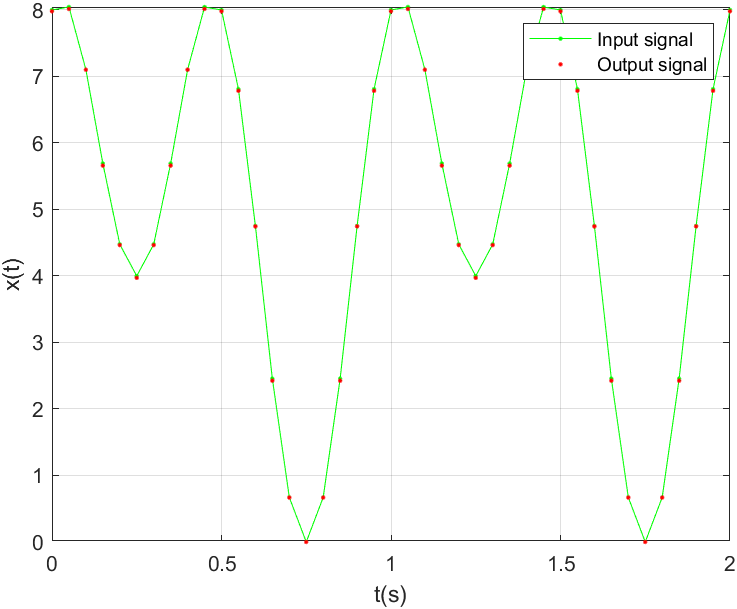
Rysunek . Wynik przetwornika 1-bitowego.



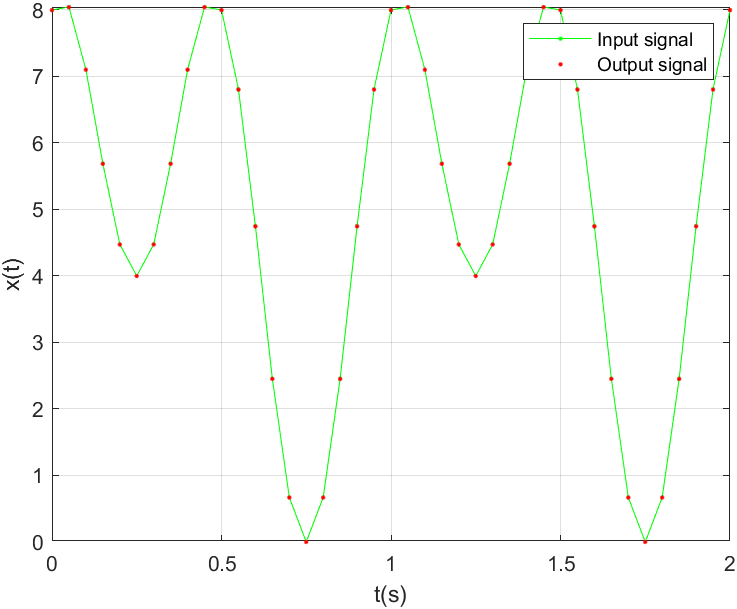
Rysunek . Wynik przetwornika 3-bitowego.



Rysunek . Wynik przetwornika 4-bitowego.



Rysunek . Wynik przetwornika 8-bitowego.



Rysunek . Wynik przetwornika 10-bitowego.

Tabela . Porównanie przedziału kwantyzacji z maksymalnym błędem kwantyzacji w zależności od bitów przetwornika.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **1-bit** | **3-bit** | **4-bit** | **8-bit** | **10-bit** |
| Połowa przedziału kwantyzacji [V] | 2.2500 | 0.5625 | 0.2812 | 0.0176 | 0.0044 |
| Maksymalny błąd kwantyzacji [V] | 4.4751 | 1.1001 | 0.5376 | 0.0308 | 0.0063 |

Obraz próbkowanego sygnału wraz ze wzrostem liczby bitów jest coraz dokładniejszy, co widać po maksymalnym błędzie kwantyzacji (wraz ze wzrostem ilości bitów, błąd drastycznie maleje). Błędy kwantyzacji maleją o 2-3 rzędy wielkości w zależności od rozdzielczości przetwornika.

Tabela . Współczynnik zawartości harmonicznych.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **1-bit** | **2-bit** | **4-bit** | **8-bit** |
| THD [dB] | -9.67 | -9.71 | -9.24 | -9.67 |

Zadanie 3

Listing 4. Widmo DFT sygnału z różnymi częstotliwościami próbkowania w przetworniku SAR.

for j = 1:length(t)%SAR ADC

adc = min\_range;

for i = 1:1:n\_bits

adc = adc + range/(2^i);

adc\_output(j,i)=1;

if adc>x(j)

adc = adc - range/(2^i);

adc\_output(j,i)=0;

end

end

adc\_plot(j) = adc;

end

x\_fft = dft(adc\_plot);

function y = dft(x)

y = zeros(size(x));

for i=0:(length(x)-1)

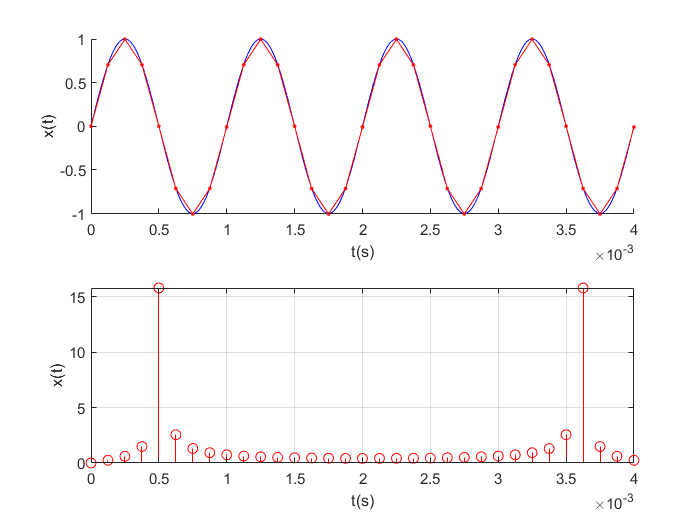
for j=0:(length(x)-1)

y(i+1) = y(i+1) + x(j+1)\*exp(-1i\*2\*pi/length(x)\*i\*j);

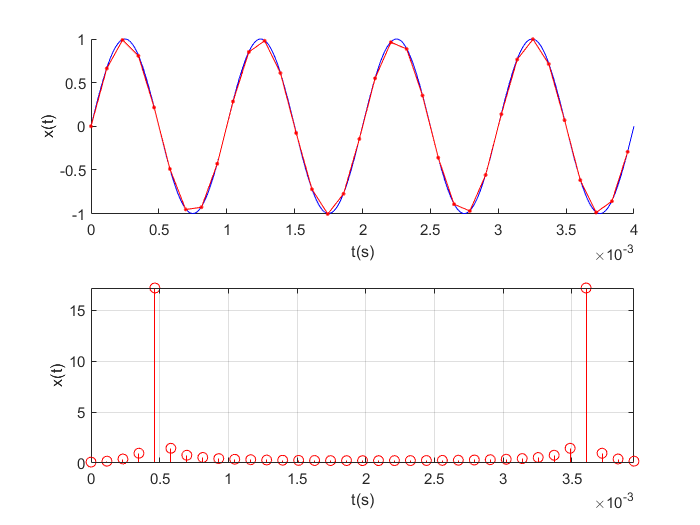
end

end

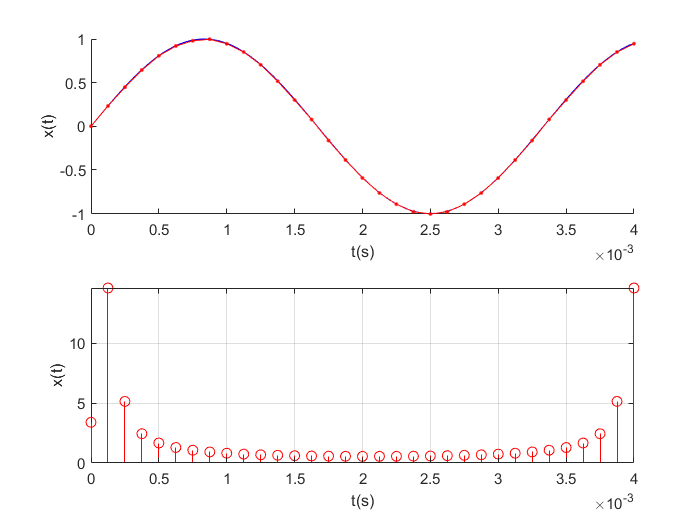
end



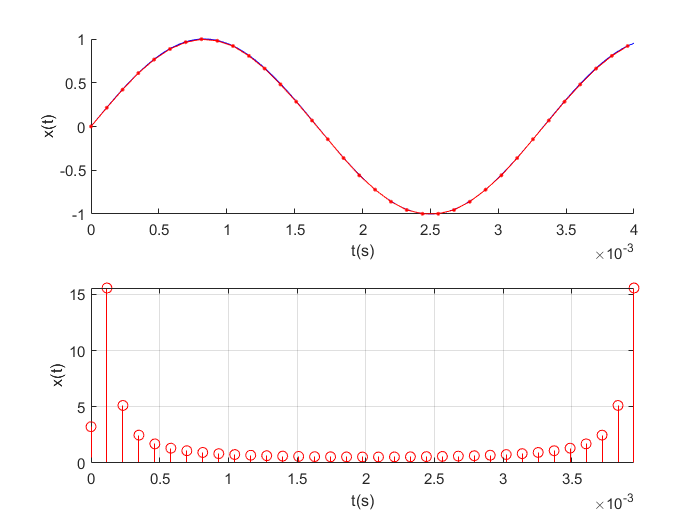
Rysunek . Sygnał sinusoidalny o f=1kHz próbkowany częstotliwością 8kHz i widmo DFT.



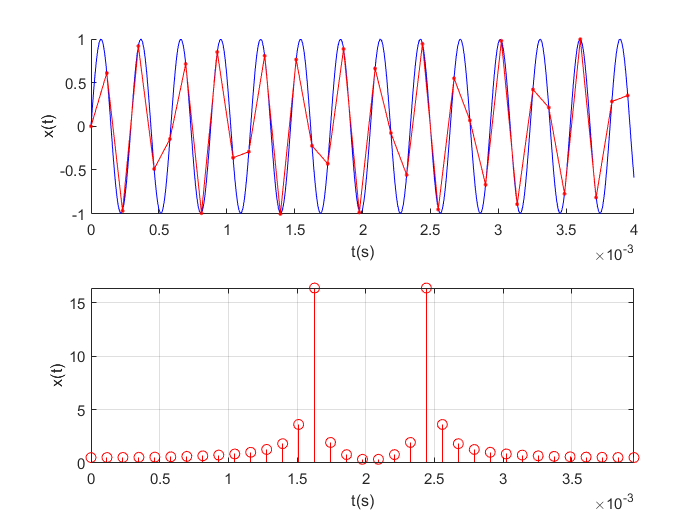
Rysunek . Sygnał sinusoidalny o f=1kHz próbkowany częstotliwością 8,6kHz i widmo DFT.



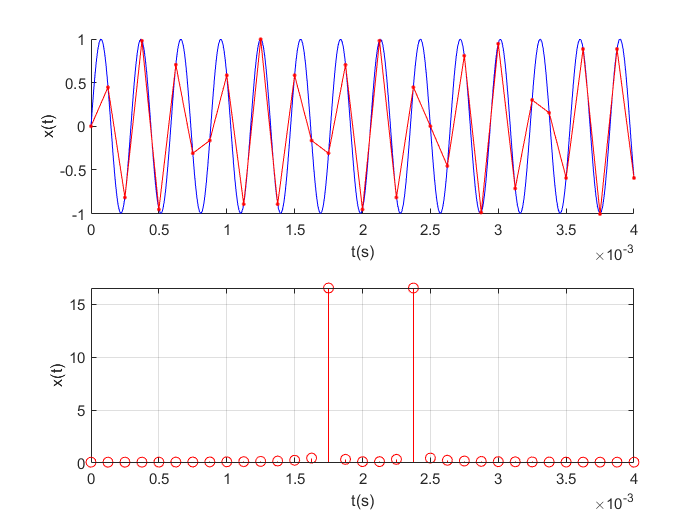
Rysunek . Sygnał sinusoidalny o f=300Hz próbkowany częstotliwością 8kHz i widmo DFT.



Rysunek . Sygnał sinusoidalny o f=300Hz próbkowany częstotliwością 8,6kHz i widmo DFT.



Rysunek . Sygnał sinusoidalny o f=3,4kHz próbkowany częstotliwością 8kHz i widmo DFT



Rysunek . Sygnał sinusoidalny o f=3,4kHz próbkowany częstotliwością 8,6kHz i widmo DFT.

Analiza widma dla każdego z przypadków pokazuje, że każdy sygnał sinusoidalny ma dwie składowe. Sygnał na wyjściu jest nieregularnie zniekształcony.

Zadanie 4

a) Listing 5. Sygnał sinusoidalny, widmo DFT (część realna i urojona), faza oraz usunięcie błędu o zerach.

sinus = sin(2\*pi\*W/N\*t);

plot(t,sinus,'.-b');

stem(t,real(dft(sinus)),'-r');

stem(t,imag(dft(sinus)),'-r');

stem(t,angle(dft(sinus)),'-r');

stem(t,angle(eliminate\_zero\_error(dft(sinus),1)),'-r');

function y = dft(x)

y = zeros(size(x));

for i=0:(length(x)-1)

for j=0:(length(x)-1)

y(i+1) = y(i+1) + x(j+1)\*exp(-1i\*2\*pi/length(x)\*i\*j);

end

end

end

function y = eliminate\_zero\_error(x, limit)

real\_x = abs(real(x));

imag\_x = abs(imag(x));

for i=1:length(x)

if real\_x(i)<limit

real\_x(i) = 0;

end

if imag\_x(i)<limit

imag\_x(i) = 0;

end

end

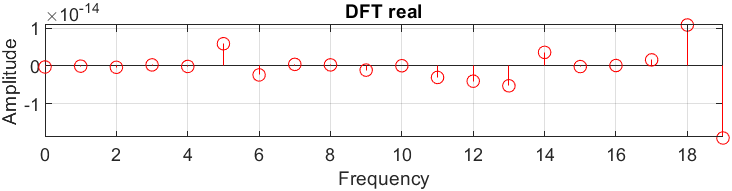
y = real\_x + 1i \* imag\_x;

end

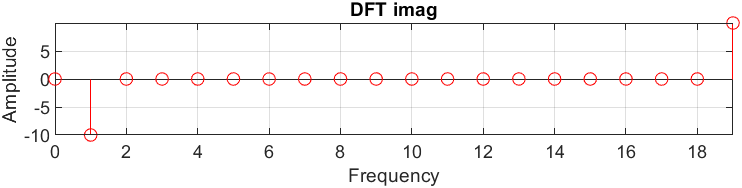
Obraz zawierający linia, Wykres, diagram, tekst

Opis wygenerowany automatycznie

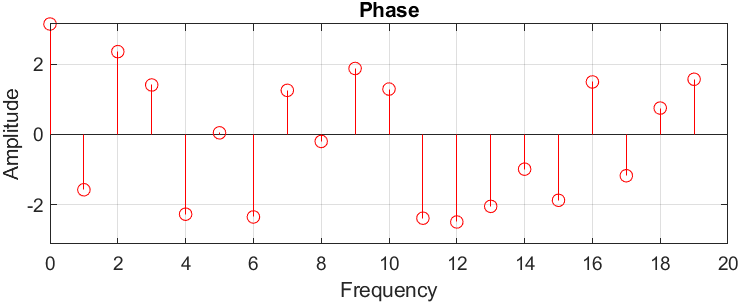
Rysunek . Sygnał sinusoidalny o 1 okresie.



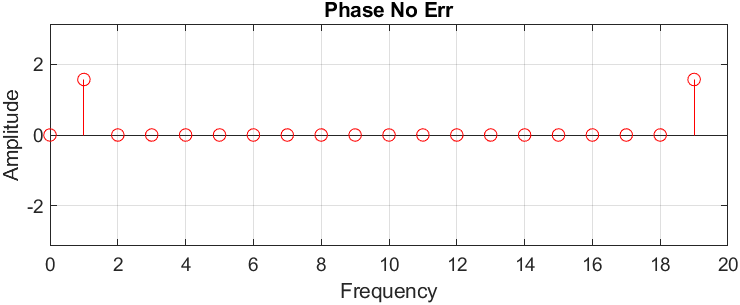
Rysunek . Widmo DFT sygnału o 1 okresie - część realna.



Rysunek . Widmo DFT sygnału o 1 okresie - część urojona.



Rysunek . Widmo DFT fazy sygnału o 1 okresie.

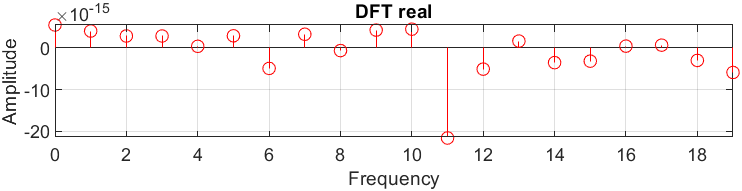


Rysunek . Wyeliminowane błędy w fazie dla sygnału o 1 okresie.

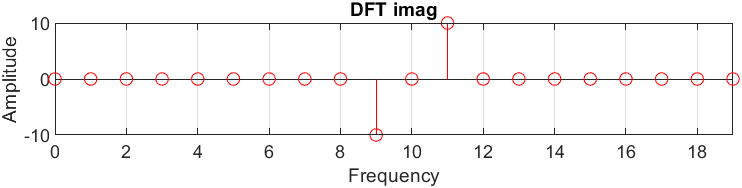
Obraz zawierający linia, Wykres, Czcionka, tekst

Opis wygenerowany automatycznie

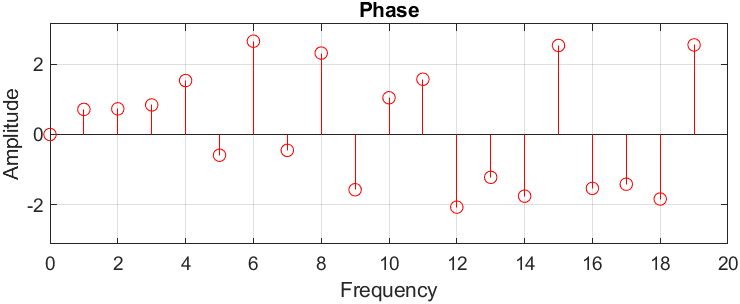
Rysunek 23. Sygnał sinusoidalny o 9 okresach.



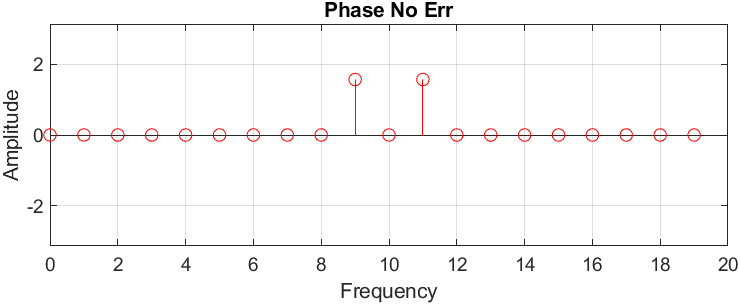
Rysunek 24. Widmo DFT sygnału o 9 okresach - część realna.



Rysunek 25. Widmo DFT sygnału o 9 okresach - część urojona.



Rysunek 26. Widmo DFT fazy sygnału o 9 okresach.



Rysunek . Wyeliminowane błędy w fazie dla sygnału o 9 okresach.

Widmo DFT idealnego sygnału sinusoidalnego powinno mieć tylko część urojoną (natomiast cosinus tylko część rzeczywistą). Wyznaczone widmo potwierdza przypuszczenia, ponieważ wartości rzeczywiste są bardzo małego rzędu w porównaniu do wartości urojonych. Faza bez eliminowania błędów jest niedokładna ze względu na niedokładności sygnału i istnienie części rzeczywistej.

b) Listing 6. Wygenerowanie N próbek sygnału sinusoidalnego dla różnych częstotliwości.

f = 0.57\*dt;

sinus = sin(2\*pi\*f\*t);

plot(t,sinus,'.-b');

stem(t,real(dft(sinus)),'-r');

stem(t,imag(dft(sinus)),'-r');

stem(t,angle(dft(sinus)),'-r');

stem(t,angle(eliminate\_zero\_error(dft(sinus),1)),'-r');

function y = dft(x)

y = zeros(size(x));

for i=0:(length(x)-1)

for j=0:(length(x)-1)

y(i+1) = y(i+1) + x(j+1)\*exp(-1i\*2\*pi/length(x)\*i\*j);

end

end

end

function y = eliminate\_zero\_error(x, limit)

real\_x = abs(real(x));

imag\_x = abs(imag(x));

for i=1:length(x)

if real\_x(i)<limit

real\_x(i) = 0;

end

if imag\_x(i)<limit

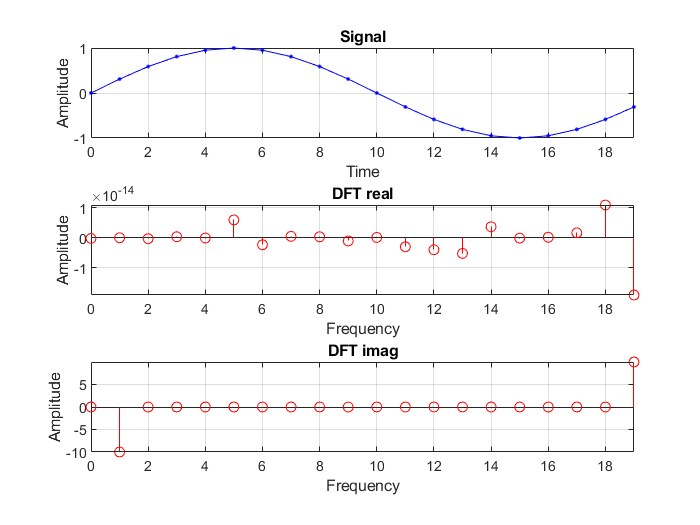
imag\_x(i) = 0;

end

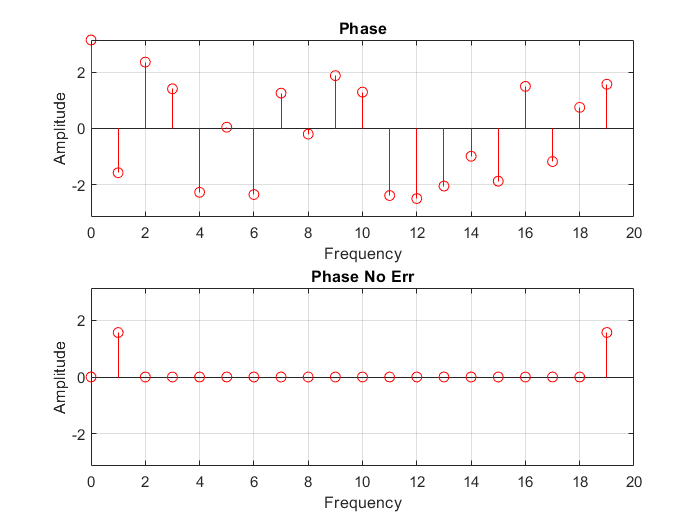
end

y = real\_x + 1i \* imag\_x;

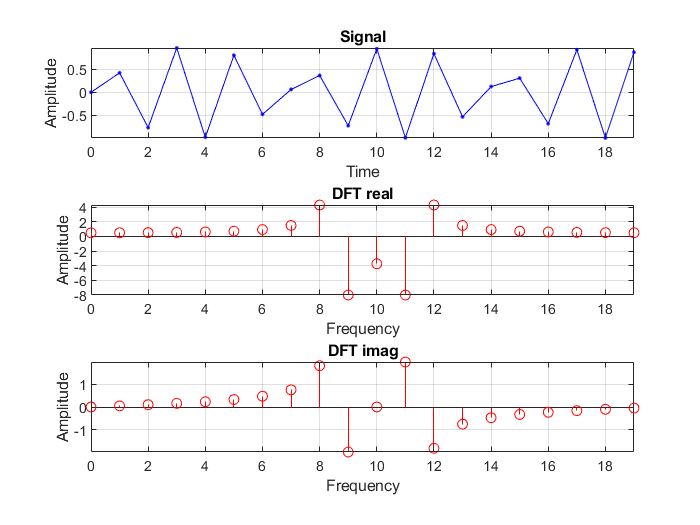
end



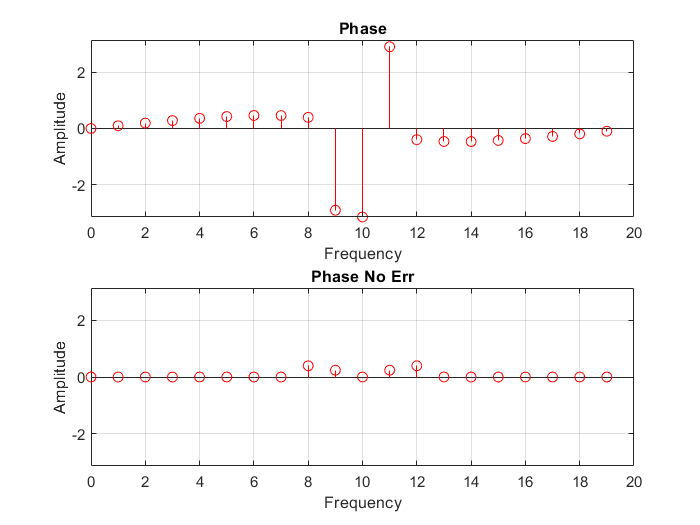
Rysunek . Sygnał sinusoidalny o częstotliwości f=0.05fs i widmo DFT.



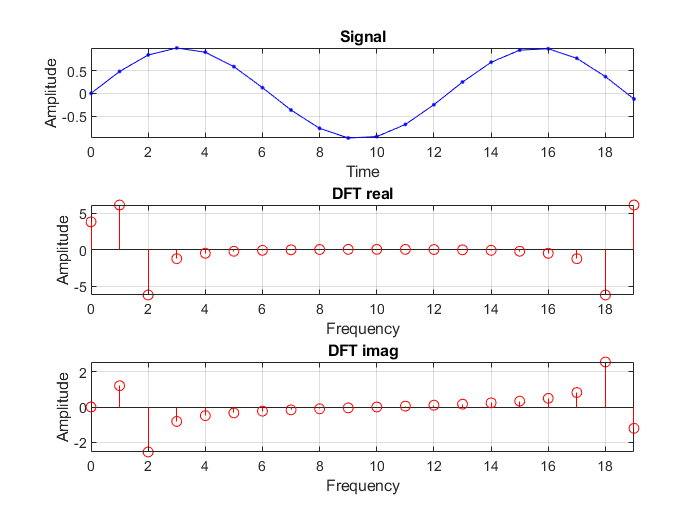
Rysunek . Widmo DFT fazy i wyeliminowanie błędów zer sygnału o f=0.05fs.



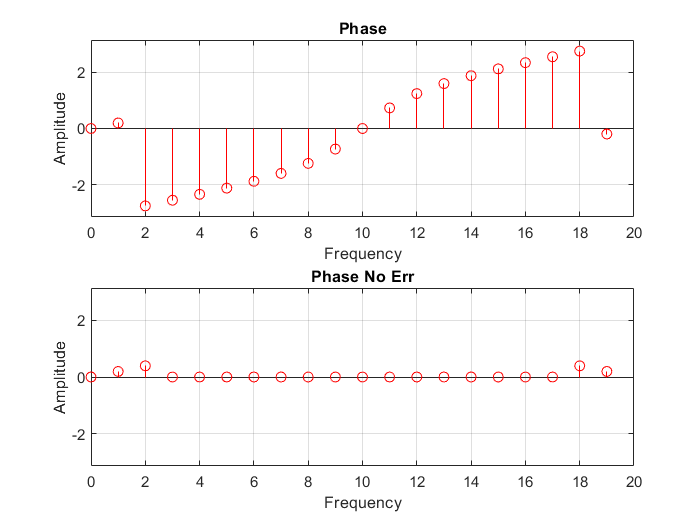
Rysunek . Sygnał sinusoidalny o częstotliwości f=0.43fs i widmo DFT.



Rysunek . Widmo DFT fazy i wyeliminowanie błędów zer sygnału o f=0.43fs.



Rysunek . Sygnał sinusoidalny o częstotliwości f=1.08fs i widmo DFT.



Rysunek . Widmo DFT fazy i wyeliminowanie błędów zer sygnału o f=1.08fs.

Częstotliwość sygnału powinna być równa iloczynowi częstotliwości próbkowania i numeru prążka. Dla częstotliwości równej 0.05fs niezerowe prążki mają numery 1 i 19, dla 0.43fs prążki od 1 do 19, natomiast dla 1.08fs prążki od 1 do 8 i od 11 do 19. Przeciek widma widać dla sygnałów z coraz większymi częstotliwościami. Razem ze wzrostem częstotliwości następuje wzrost przecieku widma (rzeczywistej części DFT). Minimalna częstotliwość próbkowania, aby uniknąć przecieku widma, powinna być podwojoną wartością częstotliwości sygnału. Zjawisko aliasingu zostało zaobserwowane na przykładzie próbek 0.05fs i 1.08fs. Jest to efekt spowodowany poprzez zły dobór częstotliwości próbkowania. Konsekwencją tego jest złe odtworzenie oryginalnego sygnału (np. sygnał o większej częstotliwości zostanie odtworzony jako sygnał o mniejszej częstotliwości).

c) Listing 7. Wygenerowanie N próbek sygnału prostokątnego.

for n=0:(N-1)

if (n >= 0) && (n <= (L-1))

x(n+1) = 1;

else

x(n+1) = 0;

end

end

plot(t,x,'.-b');

stem(t,real(dft(x)),'-r');

stem(t,imag(dft(x)),'-r');

stem(t,abs(eliminate\_zero\_error(dft(x),0.01)),'-r');

stem(t,angle(eliminate\_zero\_error(dft(x),1)),'-r');

function y = dft(x)

y = zeros(size(x));

for i=0:(length(x)-1)

for j=0:(length(x)-1)

y(i+1) = y(i+1) + x(j+1)\*exp(-1i\*2\*pi/length(x)\*i\*j);

end

end

end

function y = eliminate\_zero\_error(x, limit)

real\_x = abs(real(x));

imag\_x = abs(imag(x));

for i=1:length(x)

if real\_x(i)<limit

real\_x(i) = 0;

end

if imag\_x(i)<limit

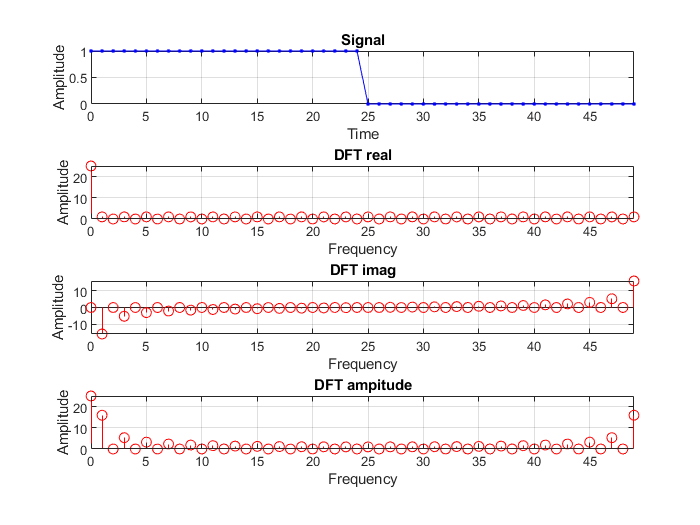
imag\_x(i) = 0;

end

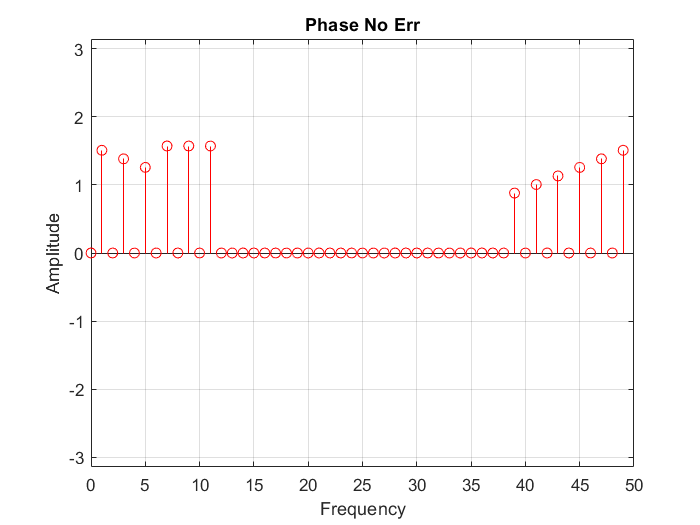
end

y = real\_x + 1i \* imag\_x;

end



Rysunek . Widmo DFT, widmo amplitudowe sygnału prostokątnego dla L=25.



Rysunek . Faza widma sygnału prostokątnego dla L=25.

Prążki widma zerują się dla indeksów nieparzystych. Amplitudy prążków maleją. Zarówno część rzeczywista, jak i urojona widma są symetryczne. Część rzeczywista jest symetryczna względem prostej x = 25, natomiast część urojona względem punktu (25;0). Nie licząc przypadku L=25, dla innych wartości faza widma sygnału oraz widmo amplitudowe także są symetryczne.

Zadanie 5

a) Obliczenie wyjścia y za pomocą splotu liniowego opisanego odpowiedzią impulsową ℎ(𝑛)=[1; 2; 3; 1], gdy na wejście podany jest sygnał 𝑥(𝑛)=[1;1;0.5;−1;2;1;4].

Listing 8. Splot liniowy podanych sygnałów.

y = splot\_liniowy(x,h)

function y = splot\_liniowy(x,h)

y = zeros(1,length(x)+length(h));

for n = 1:(length(x)+length(h))

for k = 1:(length(x)+length(h)-1)

if ((n-k) >= 1) && ((n-k) <= length(h))

hnk = h(n-k);

else

hnk = 0;

end

if (k >= 1) && (k <= length(x))

xk = x(k);

else

xk = 0;

end

y(n) = y(n) + (xk \* hnk);

end

end

end

Obraz zawierający tekst, Czcionka, algebra, zrzut ekranu

Opis wygenerowany automatycznie

Rysunek 32. Wynik splotu liniowego.

Długość sygnału wynosi *N = NX+NH-1.*

b) Obliczenie splotu kołowego.

4.5

Listing 9. Splot kołowy podanych sygnałów.

h= [1, 2, 3, 1] ;

x = [1, 1, 0.5, -1];

y = splot\_kolowy(x,h)

function y = splot\_kolowy(x, h)

L = length(x);

y = zeros(1, L);

for n = 1:L

for k = 1:L

nk\_wrap = mod((n - k), L)+1;

if (k <= L) && (k >=1) && (nk\_wrap >= 1) && (nk\_wrap <= L)

y(n) = y(n) + (x(k) \* h(nk\_wrap));

end

end

end

end

Obraz zawierający tekst, zrzut ekranu, Czcionka, biały

Opis wygenerowany automatycznie

Rysunek 33. Wynik splotu kołowego.

Pierwsze trzy próbki sygnału splotu kołowego różnią się od wyniku splotu liniowego.

c) Obliczenie splotu liniowego za pomocą splotu kołowego.

- określenie długości sygnałów,

NX = 7

NH = 4

*N = NX+NH-1 = 10*

ℎ[𝑛] = [1; 2; 3; 1; 0; 0; 0; 0; 0; 0]

𝑥[𝑛] = [1; 1; 0.5; −1; 2; 1; 4; 0; 0; 0]

- obliczenie DFT,

H[0] = 1+2+3+1+0+0+0+0+0+0 = 7

H[1] =

H[2] =

H[3] =

H[4] =

H[5] =

H[6] =

H[7] =

H[8] =

H[9] =

]

Analogicznie dla X[k]:

]

- obliczenie odwrotnej DFT,

Podobnie zostały wyliczone kolejne elementy tej transformaty.

Listing 10. Wyliczenie splotu liniowego za pomocą splotu kołowego.

y = splot\_liniowy\_kolowym(x,h)

function y = dft(x)

y = zeros(size(x));

for i=0:(length(x)-1)

for j=0:(length(x)-1)

y(i+1) = y(i+1) + x(j+1)\*exp(-1i\*2\*pi/length(x)\*i\*j);

end

end

end

function y = splot\_liniowy\_kolowym(x,h)

dlugosc = length(x)+length(h)-1;

hn = zeros(1,dlugosc);

xn = zeros(1,dlugosc);

for i = 1:length(x)

xn(i) = x(i);

end

for i = 1:length(h)

hn(i) = h(i);

end

dft\_xh = dft(xn) .\* dft(hn);

y = real(ifft(dft\_xh));

end

Obraz zawierający tekst, Czcionka, zrzut ekranu, linia

Opis wygenerowany automatycznie

Rysunek 34. Wynik splotu liniowego obliczonego za pomocą splotu kołowego.

Wynik splotu liniowego, obliczonego poprzez splot kołowy jest zgodny z wynikiem splotu liniowego w pierwszym podpunkcie. Długość splotu kołowego powinna być równa sumie długości sygnałów pomniejszonej o 1.

d) Widmo DFT:

]

]

Odwrotna DFT:

Obie wartości zostały wyliczone w podpunkcie powyżej.

Użyte zostały funkcje z poprzedniego podpunktu.

Obraz zawierający tekst, zrzut ekranu, Czcionka, dokument

Opis wygenerowany automatycznie

Rysunek 35. Wynik widma DFT oraz IDFT sygnałów.

Wartości IDFT są równe wynikom splotu liniowego z podanych sygnałów.

***Wnioski***

Zadanie 1

Aby przebieg prostokątny spełniał swoje założenia, należy brać pod uwagę sygnał sinusa  
z liczbą harmoniczną większą lub równą 10.

Zadanie 2

Sygnał należy próbkować częstotliwością optymalną (wynoszącą dziesięciokrotność częstotliwości minimalnej) lub częstotliwością minimalną (sprawdza się głównie w przypadku sygnałów okresowych).

Im większa rozdzielczość przetwornika, tym mniejszy błąd kwantyzacji, zatem lepsze odwzorowanie sygnału.

Zadanie 3

Sygnał jest nieregularnie zniekształcony, ponieważ częstotliwość próbkowania nie jest wielokrotnością częstotliwości sygnału.

Zadanie 4

Sygnał sinusoidalny powinien mieć znaczącą część urojoną widma DFT, ponieważ jest funkcją nieparzystą (transformata Fourier’a). Cosinus natomiast powinien mieć składowe w części rzeczywistej, gdyż jest funkcją parzystą.

Przy szybkich, okresowych sygnałach należy zwracać szczególną uwagę na częstotliwość próbkowania, bo jeśli okaże się za mała, może wystąpić niepożądane zjawisko aliasingu.

Zadanie 5

Splot kołowy polega na wymnożeniu przez siebie transformat Fourier’a, natomiast splot liniowy jest operacją wykonywaną na dwóch ciągach próbek, jego wynikiem jest nowa sekwencja próbek.