Przetwarzanie sygnałów

Ćwiczenie 3

Dyskretna transformacja Fouriera

dr hab. inż. Grzegorz Jóźwiak, dr hab. inż. Tomasz Piasecki (tomasz.piasecki@pwr.edu.pl)

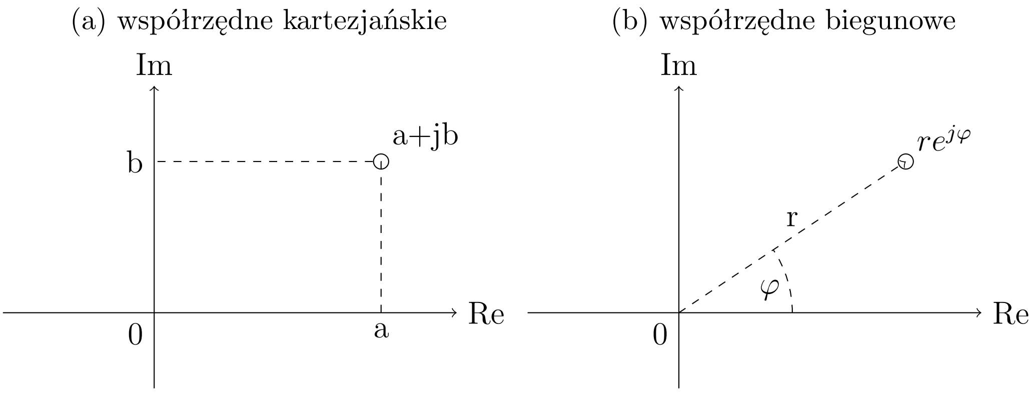
Spis treści

Katedra Nanometrologii

Wydział Elektroniki, Fotoniki i Mikrosystemów
Politechnika Wrocławska

1 Liczby zespolone

Liczby zespolone są uogólnieniem liczb rzeczywistych. Liczba zespolona jest sumą liczby rzeczywistej i liczby rzeczywistej przemnożonej przez jednostkę urojoną j $(a + jb)$. Jednostka urojona j spełnia równość $j^2 = -1$. Zatem liczbę zespoloną można podzielić na dwie części: część rzeczywistą a i część urojoną b. Liczba zespolona, której część urojona jest równa zero, jest po prostu liczbą rzeczywistą. Zbiór liczb zespolonych oznacza się zwykle dużą literą $\mathbb{C}$, podobnie jak zbiór liczb rzeczywistych oznacza się dużą literą $\mathbb{R}$. Liczby zespolone można przedstawić graficznie na płaszczyźnie zespolonej jako punkty. Oś odciętych (pozioma) informuje o wartościach części rzeczywistej liczb zespolonych, natomiast oś rzędnych (pionowa) – o wartościach części urojonej (rysunek 1a).



Rysunek 1: Reprezentacje liczby zespolonej na płaszczyźnie zespolonej.

Zatem oś odciętych można traktować jako oś liczb rzeczywistych, które stanowią podzbiór liczb zespolonych. Dla liczb zespolonych zdefiniowane są operacje algebraiczne, takie jak:

- dodawanie: $(a + jb) + (c + jd) = (a + c) + j(b + d)$,
- odejmowanie: $(a + jb) - (c + jd) = (a - c) + j(b - d)$,
- mnożenie $(a + jb)(c + jd) = (ac - bd) + j(bc + ad)$,
- dzielenie: $\frac{a+jb}{c+jd} = \frac{(a+jb)(c-jd)}{c^2+d^2} = \frac{ac+bd}{c^2+d^2} + j\frac{bc-ad}{c^2+d^2}$.

Płaszczyznę liczb zespolonych można przedstawić również w układzie biegunowym (rysunek 1b). W układzie tym położenie punktu definiuje się za pomocą długości wektora i kąta jaki tworzy on z dodatnią półosią odciętych (poziomą).

Położenie punktu w układzie biegunowym można łatwo wyznaczyć, przechodząc z algebraicznej postaci liczby zespolonej do postaci trygonometrycznej $a + jb = r(\cos\varphi + j\sin\varphi)$ lub wykładniczej $a + jb = re^{j\varphi}$. Aby takiego przejścia dokonać, należy obliczyć r i φ , korzystając z zależności $r = \sqrt{a^2 + b^2}$, $\varphi = \arctan\left(\frac{b}{a}\right)$. Aby przejść z postaci wykładniczej na algebraiczną wykorzystuje się postać trygonometryczną oraz oblicza wartości $a = r\cos\varphi$ i $b = r\cos\varphi$, co daje nam wartości odpowiednio części urojonej i rzeczywistej.

Reprezentacja liczb zespolonych w układzie biegunowym wiąże się z pewnymi niedogodnościami. Pierwsza z nich wynika z okresowości funkcji trygonometrycznych $\sin(\varphi) = \sin(2\pi n+\varphi)$ i $\cos(\varphi) = \cos(2\pi n+\varphi)$ dla $n \in \mathbb{Z}$. Dlatego $re^{j\varphi} = re^{2\pi n+\varphi}$ dla $n \in \mathbb{Z}$ (zbiór liczb całkowitych).

Drugą niedogodnością jest konieczność rozpatrzenia przypadku, w którym część rzeczywista liczby zespolonej jest równa 0. Należy też pamiętać, że funkcja arcus tangens (arctan) przyjmuje wartości z przedziału $\left\langle -\frac{\pi}{2}, \frac{\pi}{2} \right\rangle$, podczas gdy kąt φ powinien przyjmować wartości od $\langle -\pi, \pi)$. Aby zatem obliczyć poprawny kąt fazowy φ należy odpowiednio skorygować wynik funkcji arctan dla przypadków, kiedy część rzeczywista liczby zespolonej jest ujemna. Wówczas – w zależności od tego, czy część urojona jest dodatnia czy ujemna – należy do wyniku funkcji arctan dodać lub od tego wyniku odjąć liczbę π.

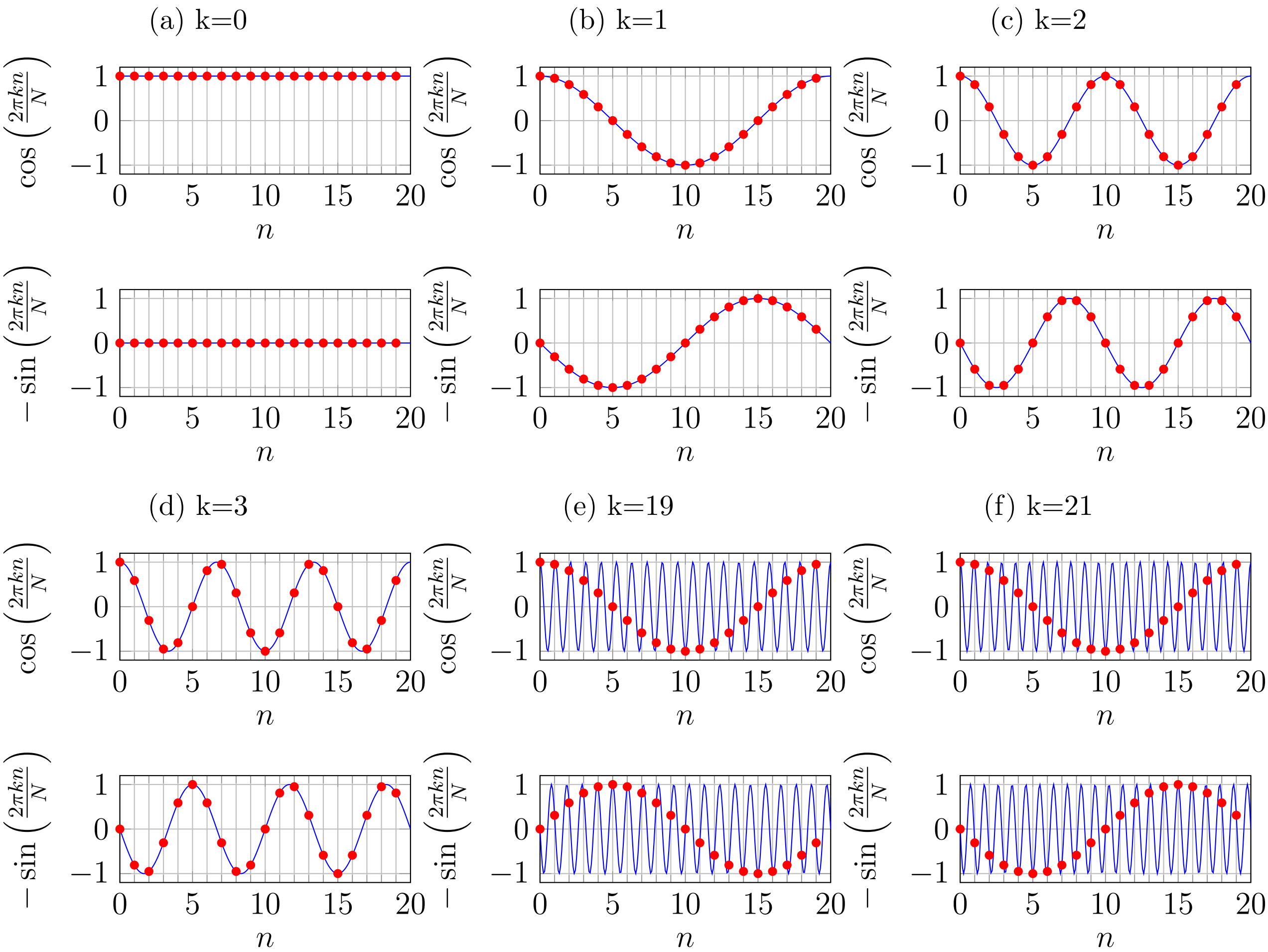
2 Dyskretna Transformacja Fouriera (ang. Discrete Fourier Transform –DFT)

2.1 Funkcje bazowe DFT

W dziedzinie przetwarzania sygnałów operacja rozkładu na zbiór funkcji bazowych (składowych) polega na obliczeniu współczynników skalujących amplitudę tych funkcji. Obliczony zbiór współczynników nazywa się widmem sygnału. Zbiór funkcji bazowych zwykle powstaje przez parametryzację pewnej funkcji. W przypadku DFT jest to funkcja zespolona w postaci:

$$e^{-j\frac{2\pi kn}{N}} = \cos\left(\frac{2\pi kn}{N}\right) - j\sin\left(\frac{2\pi kn}{N}\right) \tag{1}$$

w której N jest liczbą próbek rozkładanego dyskretnego sygnału, n numerem próbki sygnału rozkładanego, a k parametrem, będącym liczbą całkowitą, generującym zbiór funkcji dla operacji rozkładu. Przykładowe wykresy ilustrujące przebiegi rzeczywistej i urojonej części funkcji bazowych DFT przedstawiono na rysunku 2.



Rysunek 2: Wartości wybranych funkcji bazowe DFT (punkty) dla $N = 20$. Linią przedstawiono przebieg wyrażony funkcją (1).

Jeśli $k = 0$ składowa rzeczywista i urojona wartości funkcji bazowej jest taka sama dla każdego n (rysunek 2a). Dla $k = 1$ składowe rzeczywista punktów funkcji bazowej odpowiadają punktom funkcji cos, której 1 okres rozciąga się na N próbek. Wartości urojone - analogicznie, ale funkcji $-\sin$ (rysunek 2b). Dla $k = 2$ funkcje cos i $-\sin$ mają 2 okresy na przestrzeni N próbek (rysunek 2c), itd.

Ciekawe rzeczy dzieją się gdy dalej zwiększyć k. Jeśli $k = N - 1$, czyli 19 z przykładu z rysunku 2e, to wartości składowej rzeczywistej funkcji bazowej są identyczne, jak dla $k = 1$ a składowej urojonej mają przeciwny znak, jak te dla $k = 1$. Gdy z kolei $k = N + 1$ to uzyskuje się już zupełnie identycznie wyglądającą funkcję bazową, jak dla $k = 1$ (rysunek 2f).

Dla skończonego ciągu N próbek ma sens zatem tworzenie zbioru maksymalnie N różnych funkcji bazowych. Obliczanie kolejnych mija się z celem, gdyż funkcje tak utworzone będą powtórzeniem istniejących. Ogólnie, identyczne będą funkcje bazowe dla $k = k_1$ oraz $k = k_1 + m \cdot N$, gdzie k_1 to dowolna wartość całkowita z przedziału $[0, N - 1]$, a m to dowolna wartość całkowita.

Funkcje ciągłe w dziedzinie czasu, będące rekonstrukcjami funkcji bazowych DFT, opisanych wyżej, miałyby równanie:

$$e^{\frac{j2\pi kt}{NT_s}} = \cos\left(\frac{2\pi kt}{NT_s}\right) + j\sin\left(\frac{2\pi kt}{NT_s}\right) \tag{2}$$

, w którym wyrażenie

$$\frac{k}{NT_s} = \frac{kf_s}{N} = f_k \tag{3}$$

ma sens częstotliwości f_k, wyrażonej w hercach, odpowiadającej k-tej funkcji bazowej.

2.2 DFT sygnału zespolonego

Najważniejszą zaletą dyskretnej transformaty Fouriera (DFT) jest to, że można ją obliczyć numerycznie. Fakt ten stał się szczególnie istotny w momencie pojawienia się układów mikroprocesorowych.

Jeśli dany jest sygnał $x[n]$ składający się z N próbek, to najprostszy ze sposobów na obliczenie jego widma dyskretnego $X[k]$ to realizacja poniższego równania:

$$X[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi kn}{N}}. \tag{4}$$

Czym jest to równanie? Zapiszmy je inaczej. Niech $x[n]$ pozostanie jak jest, natomiast przed obliczeniem k-tej składowej widma $X[k]$ przygotujmy sobie wektor $f[n]$ o długości N, którego kolejne próbki $f[n] = e^{-j\frac{2\pi kn}{N}}$. Jest to wektor liczb zespolonych, zawierający wartości kolejnych punktów funkcji bazowych[1]. Teraz (4) można zapisać:

$$X[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n] \cdot f[n] = \frac{1}{N}(x \cdot f) \tag{5}$$

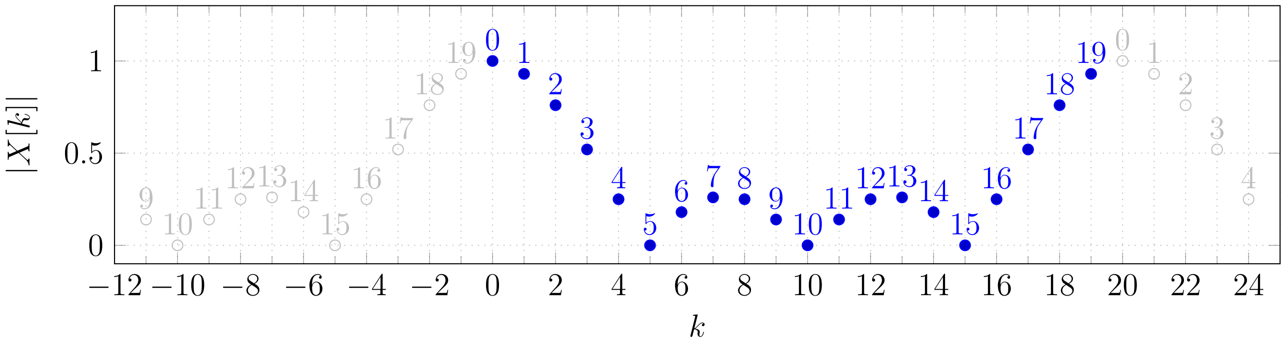
Oprócz czynnika normalizującego amplitudę widma $\frac{1}{N}$ wyliczenie k-tej składowej widma polega po prostu na policzeniu iloczynu skalarnego wektora próbek transformowanego sygnału x z wektorem wartości funkcji bazowych f. W pakiecie Octave do obliczenia takiego iloczynu skalarnego służy funkcja `dot(x, f)`.

Do obliczenia całego widma $X[k]$ czynność tę trzeba wykonać dla wszystkich potrzebnych wartości k. Jakie to mogą być wartości omówiono dalej. Pamiętaj jednak, że dla każdego k trzeba by na nowo obliczyć wektor $f[n]$.

Składowe widma $X[k]$ są liczbami zespolonymi, dla których można wyznaczyć moduł i argument. Zbiór zespolonych składowych $X[k]$ nazywamy widmem zespolonym. Można je rozbić na widmo rzeczywiste i urojone lub obliczyć moduły i argumenty liczb, które tworzą widma amplitudowe i fazowe.

[1] Wartości elementów tego wektora przedstawiono czerwonymi kropkami na rysunku 2.

Jak wykazano w rozdziale 2.1 ma sens obliczanie maksymalnie N różnych funkcji bazowych, bo potem zaczną się one powtarzać. Tak samo składowe widma oblicza się dla N różnych k, po potem zaczną się one powtarzać. Często wybiera się zakres wartości całkowitych $k \in [0, N-1]$. Tak na przykład działają wbudowane w Octave funkcje obliczające DFT sygnału, jak na przykład funkcja `fft(x)`. Widmo w ten sposób powstałe pokazano w kolorowej części wykresu 3.



Rysunek 3: Przykładowe dyskretne widmo amplitudowe impulsu prostokątnego o czasie trwania 4 próbki i amplitudzie 5 dla $N = 20$. Kolorem przedstawiono wartości widma dla $k \in [0, N-1]$. Kolorem szarym zaznaczono wartości współczynników widma, które są powtórzeniem tych obliczonych dla $k \in [0, N-1]$. Opis przy punkcie oznacza numer współczynnika powtarzanego.

Takie podejście ma jednak swoje wady. Na przykład, jeżeli obliczyć korzystając z (3) częstotliwość ostatniej funkcji bazowej dla $k = N-1$ otrzyma się zakładając $N = 20$ wartość $f_k = 0,95 \cdot f_s$. A przecież próbkując z częstotliwością f_s nie powinno się widzieć w sygnale częstotliwości większych niż $0,5 \cdot f_s$, co wynika z twierdzenia Nyquista-Shannona![2]

Można postąpić inaczej. Zamiast obliczać współczynnik dla $k = N-1$ obliczmy go dla k mniejszego o N, czyli $k = N-1-N = -1$. Funkcja bazowa będzie taka sama dla obu przypadków. Podobnie postąpić można dla połowy wszystkich k. W ten sposób zmienimy przedział wartości k do liczb całkowitych z następującego przedziału[3]:

$$k \in \left[\lfloor -\frac{N}{2} \rfloor, \lfloor \frac{N}{2} - 1 \rfloor\right] \tag{6}$$

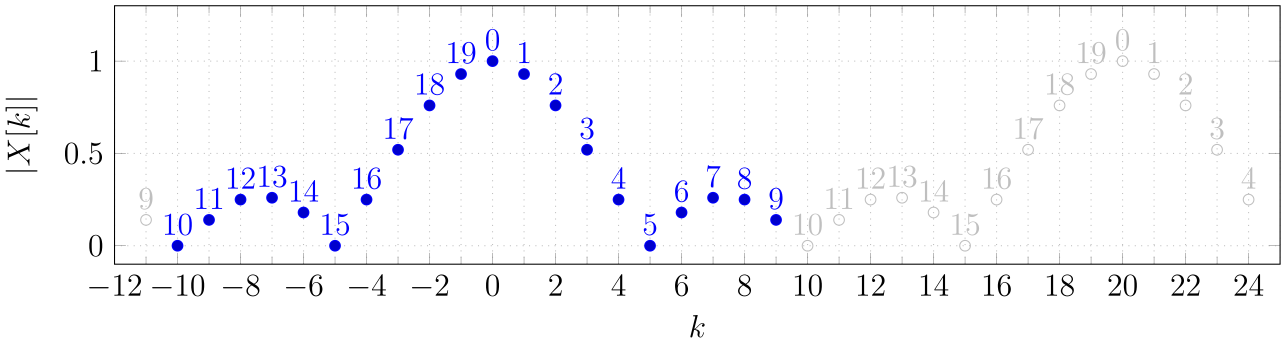
Przykładowo, jeśli $N = 20$ to zamiast k od 0 do 19 użyjemy k z przedziału od -10 do 9. Częstotliwość f_k składowej dla $k = -10$ to $-0.5 \cdot f_s$, dla $k = 0$ będzie to 0 a dla $k = 9$ będzie to $0,45 \cdot f_s$. Wartość bezwzględna częstotliwości składowych widma nie przekroczy nigdzie połowy f_s. Dodatkowo, częstotliwość równą 0 Hz otrzymamy (niemal) pośrodku widma, co zbliża je pod tym względem do tego, otrzymywanego na podstawie ciągłej transformacji Fouriera. Widmo takie rozciąga się na obie strony składowej $k = 0$ i nazywamy je dwustronnym. Jego przykład pokazano na kolorowej części wykresu z rysunku 4.

Ze względu na użyteczność dwustronnego widma dyskretnego pakiet Octave posiada wbudowaną funkcję `fftshift (x)` przesuwającą próbki w wektorze ze składowymi widma, uzyskanym wcześniej za pomocą `fft (x)`. Mając zatem w skrypcie w Octave wektor `x` z próbkami sygnału i skalar `N` z liczbą próbek, to zormalizowane widmo dwustronne obliczamy:

```
y = fftshift(fft (x)) / N;
```

[2]Nie jest to błąd, Octave działa dobrze, szczegóły dlaczego tak jest wyjaśniane są na wykładzie kursu.

[3](symbol $\lfloor\ \rfloor$ oznacza zaokrąglenie wartości w dół - N może być bowiem również nieparzyste przez co podzielone na dwa da ułamek a tymczasem k musi być całkowite).



Rysunek 4: Przykładowe dwustronne dyskretne widmo amplitudowe prezentujące te same informacje, co na rysunku 3. Kolorem zaznaczono zakres k od $\lfloor -\frac{N}{2} \rfloor$ do $\lfloor \frac{N}{2} - 1 \rfloor$. Numery oznaczają indeksy składowych widma z rysunku 3 użytych do utworzenia widma dwustronnego.

Znając zespolone współczynniki $X[k]$ DFT, możemy z powrotem zsyntezować (wyznaczyć, obliczyć) sygnał $x[n]$ korzystając z równania syntezy:

$$x[n] = \sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi kn}{N}} \tag{7}$$

2.3 Jednostronne DFT sygnału rzeczywistego

W przypadku sygnałów rzeczywistych możemy obliczyć ich DFT korzystając z ogólnej definicji ale istnieje też sposób bardziej efektywny. Przyjmijmy dla uproszczenia, że liczba próbek N jest parzysta. Spójrzmy na funkcje bazowe z rysunku 2b i 2e. Są to funkcje bazowe dla $k = 1$ i $k = 19$ przy $N = 20$. Zwróć uwagę, że część rzeczywista funkcji bazowej jest taka sama w obu przypadkach a część urojona ma przeciwny znak, są zatem liczbami zepolonymi sprzężonymi. Jeśli wartości próbek sygnału są liczbami rzeczywistymi oznacza to, że składowe widma obliczone dla $k = 1$ i $k = 19$ też mają takie same wartości rzeczywiste i przeciwne znaki części urojonej, czyli są liczbami zespolonymi sprzężonymi. Ogólnie taka zależność występuje dla par składowych widma k i $N - k$ od $k = 1$ do $k = \frac{N}{2} - 1$. W takim razie te składowe, które mają parę, mają takie same wartości modułów, czyli tę samą wartość mają składowe widma amplitudowego, co widać np. na rysunku 3.

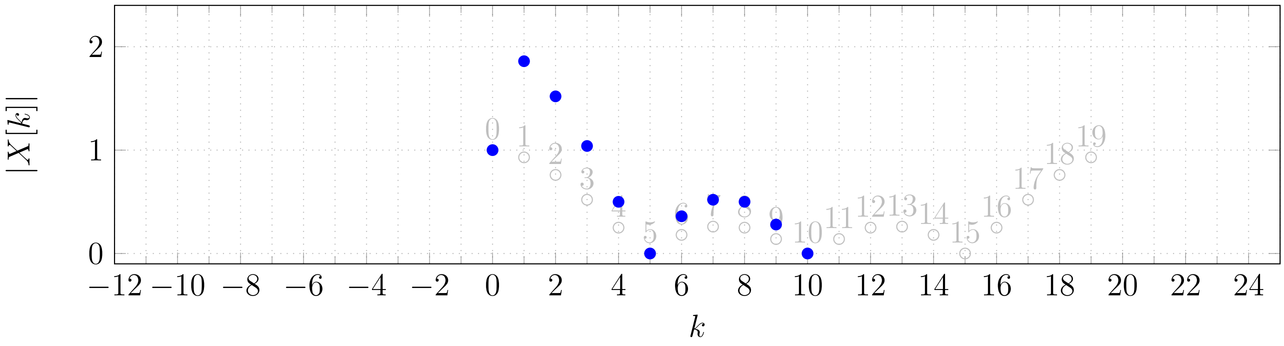
Dla $k = 0$ i $k = \frac{N}{2}$ już takiej zależności nie ma - nie ma „pary" zależnych od siebie składowych.

Dzięki temu pełnię informacji o widmie sygnału rzeczywistego można podać obliczając mniejszą liczbę składowych i ograniczyć się wyłącznie do przedziału od $k = 0$ do $k = \frac{N}{2}$, czyli prowadzi się je zatem dla niemal dwukrotnie mniejszej liczby iteracji, niż liczba próbek N. Widmo takie nie zawiera informacji o składowych o ujemnych wartościach k czy f_k, nazywane jest zatem widmem jednostronnym. Jeśli ktoś będzie potrzebował odtworzyć na tej podstawie widmo dwustronne, to składowe dla ujemnych k będzie mógł odtworzyć wiedząc, że są to wartości sprzężone do istniejących w widmie jednostronnym.

Składowe jednostronnego widma $X[k]$ mające „parę", czyli o indeksach $k \in \left[1, \frac{N}{2} - 1\right]$ oblicza się podwajając wartość (4). Ich podwojenie powoduje, że na przykład sumowanie wszystkich składowych da taki sam efekt, jak przy widmie dwustronnym. Składowe bez pary, czyli dla $k = 0$ i $k = \frac{N}{2}$ oblicza bez zmian, bezpośrednio z (4). Sposób obliczenia składowych widma jednostronnego prezentuje się zatem następująco:

$$X[k] = \begin{cases} \dfrac{2}{N} \displaystyle\sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi kn}{N}} & \text{dla } k \in \left[1, \frac{N}{2} - 1\right] \\ \dfrac{1}{N} \displaystyle\sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi kn}{N}} & \text{dla } k = \left\{0, \frac{N}{2}\right\} \end{cases} \tag{8}$$

a przykładowe widmo takie wykreślono na rysunku 5.



Rysunek 5: Przykładowe jednostronne dyskretne widmo amplitudowe prezentujące te same informacje, co na rysunku 3. Kolorem zaznaczono składowe widma jednostronnego, na szaro wskazano wartości uzyskane w trakcie obliczania widma zwykłego, dla $k \in [0, N-1]$. Zwróć uwagę na to, które wartości tego widma zostały podwojone.

Współczynniki $Re[k]$ i $Im[k]$ obliczone w sposób uproszczony dla sygnałów rzeczywistych pozwalają syntetyzować z powrotem sygnał $x[n]$ i operację tę można opisać następującym równaniem:

$$x[n] = \sum_{k=0}^{N/2} X[k] e^{j\frac{2\pi kn}{N}} \tag{9}$$

Możemy zauważyć, że do zsyntetyzowania sygnału x[n] na podstawie widma jednostronnego, wykorzystujemy teraz mniejszą liczbę składowych (N/2+1) niż w przypadku sygnałów zespolonych.

2.4 Transformacje znormalizowane i unitarne

W poprzednich punktach podawano sposoby na obliczenie tzw. znormalizowanej transformaty Fouriera. Amplitudy (moduły) poszczególnych współczynników składających się na widmo zespolone powiązane są z amplitudami funkcji bazowych, na które rozłożono transformowany sygnał. Zgodność tę uzyskuje się normalizując wyniki sum z równań (4), (8) i innych, dzieląc je przez N.

Co daje to normowanie? Jeśli np. sygnałem dyskretnym są próbki sygnału okresowego, to całkowita wartość samych tych sum będzie tym większa, im większe N czyli im dłużej sygnał ten próbkowaliśmy. Dzieląc w obliczeniach przez N eliminujemy ten efekt uzyskując wartość zależną wyłącznie od amplitudy próbkowanego sygnału.

Niekiedy jednak potrzebna będzie nie informacja o amplitudach ale o mocy sygnału. Więcej o tym będzie mowy w kolejnych ćwiczeniach. Aby transformacja zachowywała informację o mocy sygnału, musi być ona tzw. transformacją unitarną. Dyskretne transformacje unitarne oblicza się niemal identycznie, jak normalizowane, co opisano równaniami (4) i (8), pomijana jest jedynie operacja dzielenia przez liczbę próbek N.

3 Funkcje do przygotowania przed zajęciami

Przygotuj funkcję do obliczania częstotliwości f_k (patrz p. 2.3) poszczególnych składowych jednostronnego widma Fouriera sygnału rzeczywistego `gen_rfreq`. Funkcja ta przyjmuje jako parametry liczbę próbek sygnału `N`, którego widmo ma być prezentowane, oraz częstotliwość próbkowania `fs`.

```
function freq = gen_rfreq(N, fs)
    k = 0 : (N/2); # zakres od 0 .. N/2
    freq = k * fs/N;
endfunction
```

Sprawdź jej działanie, wywołując ją z linii poleceń. Czy `gen_rfreq(10, 5)` zwróci takie wartości, jakich się spodziewasz? Co w wypadku, gdy pierwszy argument będzie nieparzysty, na przykład 7? Może trzeba tę funkcję ulepszyć?

Napisać należy też ciało podobnej funkcji `gen_cfreq`, która ma obliczać wektor zawierający częstotliwości f_k poszczególnych składowych dwustronnego widma Fouriera sygnału, czyli z przedziału składowych określonych w (6).

```
function freq = gen_cfreq(N, fs)
    ...
endfunction
```

Sprawdź jej działanie, wywołując ją z linii poleceń. Czy `gen_cfreq(20,20)` zwróci takie wartości, jak się spodziewasz? Popatrz, jakie powinny być wyniki w wypadku, gdy liczba próbek `N` równa jest częstotliwości próbkowania `fs`. Przy okazji, testowanie tych funkcji tylko przy takich parametrach potrafi ukryć błędy w nich zawarte. Dlatego, aby mieć pewność, że działają prawidłowo nie ograniczaj się do takich przypadków testując swoje funkcje.

Sprawdź też inne przypadki, np `gen_cfreq(10,2)`. Czy funkcja działa dobrze, gdy pierwszy argument jest nieparzysty?

Poniżej niedokończona funkcja `y = sig_fft(x)` licząca widmo połówkowe sygnału rzeczywistego (x), stanowiąca implementację algorytmu opisanego w punkcie 2.3:

```
function y=sig_rdft(x)
    N=length(x);
    n = 0:(N-1); # wektor numerow probek
    y=zeros(1,N/2+1);
    for k=0:N/2
        f = exp(-1i * 2 * pi * k * n / N); #wektor wartosci funkcji bazowej
        y(k+1) = dot(x, f) / N; # obliczenie wartosci skladowej widma
    endfor
    ...
endfunction
```

Brakuje w niej części, oznaczonej wielokropkiem, która odpowiedzialna jest za podwojenie wartości tych składowych widma, które trzeba podwoić w widmie połówkowym. Dokończ tę funkcję. Sprawdź, czy uzyskasz prawidłowe wyniki dla sygnałów `x = ones(10, 1)'`, czyli wektora 10 jedynek reprezentującego funkcję stałą oraz sygnału zawierającego 20 próbek generowanych funkcją `gen_sin()` z poprzednich zajęć. Niech generowany przez nią sinus zawiera pełny jeden czy pełne dwa okresy funkcji sinus na 20 próbek. Testując funkcje za pomocą poleceń `plot(x)` lub `plot(time, x)` można wykreślić sygnał `x` w dziedzinie czasu. Wektor `time` jest tu wektorem wartości czasów próbek sygnału `x`, jak na zajęciach poprzednich. Za pomocą `plot(abs(y))` lub `plot(freq, abs(y))` można wykreślić widmo amplitudowe na podstawie widma zespolonego y, obliczanego testowaną funkcją. Funkcja `abs` służy do obliczenia modułów liczb zespolonych a wektor `freq` to wektor obliczany wcześniej zrobioną funkcją `gen_rfreq(N, fs)`.

Jako zadanie dodatkowe spróbuj przerobić tę funkcję tak, aby działała dla sygnałów nieparzystej długości. Jeśli Ci się to nie uda pamiętaj, by na zajęciach używać jej tylko do sygnałach o długości parzystej.

4 Origin - poznawane funkcje

`real(x)` - obliczenie części rzeczywistej liczby zespolonej x
`imag(x)` - obliczenie części urojonej liczby zespolonej x
`abs(x)` - obliczenie modułu liczby zespolonej x lub wartości bezwzględnej gdy x jest rzeczywiste
`real(x)` - obliczenie części rzeczywistej liczby zespolonej x
`arg(x)` - obliczenie argumentu (w radianach) liczby zespolonej x
`fft(x)` - obliczenie szybkiej dyskretnej transformaty Fouriera sygnału zapisanego w wektorze x; wynikiem jest wektor składowych zespolonego widma, obliczonych dla $k \in [0, N-1]$
`ifft(x)` - obliczenie szybkiej odwrotnej transformaty Fouriera dla widma zespolonego zapisanego w wektorze x; funkcja oczekuje widma w takiej formie, jak wynik `fft(x)`
`fftshift(x)` - przekształcenie wektora ze składowymi widma będącego wynikiem `fft(x)` na widmo dwustronne; ponowne zastosowanie tej funkcji na wektorze z widmem dwustronny przywraca jego pierwotną formę
`dot(x, y)` - obliczenie iloczynu skalarnego wektorów `x` i `y`

5 Pytania i zadania na kartkówkę

1. Co trzeba zrobić z wynikiem funkcji $\varphi = \arctan\left(\frac{Im(x)}{Re(x)}\right)$ aby obliczyć argument liczby zespolonej $x = -2 + j$?
2. Co trzeba zrobić z wynikiem funkcji $\varphi = \arctan\left(\frac{Im(x)}{Re(x)}\right)$ aby obliczyć argument liczby zespolonej $x = -2 - j$?
3. Podaj przykład liczb zespolonej której moduł można obliczyć w pamięci.
4. Jaką częstotliwość będzie miała 5. (licząc od zera) składowa DFT sygnału o liczbie próbek równiej 100 i okresie próbkowania równym 0,1 s?
5. Oblicz prążek $k = 1$ DFT sygnału rzeczywistego `x[n] = [1 3 0 2 3 1]`.
6. Oblicz prążek $k = 2$ DFT sygnału `x[n] = [1 3 j 0 1 2]`.
7. Czy obliczając DFT dla sygnałów rzeczywistych możemy używać równania (4)?
8. Jak będzie się różniła amplituda współczynników DFT obliczanych równaniem (4) i (8) dla sygnałów rzeczywistych?
9. Jak będzie się różniła amplituda współczynników DFT o indeksie 0 i $N/2$ obliczanych równaniem (4) i (8) dla sygnałów rzeczywistych o liczbie próbek sygnału N.
10. Podaj przykład równań sygnału rzeczywistego i jego widma DFT.
11. Podaj przykład równań sygnału zespolonego i jego widma DFT.

Uwaga. Wartości liczbowe podane w pytaniach są przykładowe. Na kartkówce podobne zadania będą zawierały inne dane.

6 Funkcje stworzone w trakcie realizacji poprzednich ćwiczeń

`gen_delta(time)` - generacja delty Kroeneckera
`gen_gauss(time, u, s)` - generacja impulsu Gaussa

`gen_sin(time, fsin, A, fi)` - generacja sygnału harmonicznego
`gen_time(N, fs)` - generacja czasów próbek
`gen_triangle(time, A, tr, tf)` - generacja impulsu trójkątnego
`sig_conv(x,y)` - obliczenie splotu sygnałów
`sig_delay_N(x, Nd)` - opóźnienie sygnału

7 Zadania do realizacji na zajęciach

Realizując zadania pisz skrypty wywołujące funkcje Octave bądź Twoje własne prowadzące do realizacji postawionego celu. Pierwsze kroki można robić w linii poleceń, sprawdzając co i jak. Dąż jednak do tego, aby całość zadania ująć w jednym skrypcie.

Wszystkie funkcje i skryptu powinny być napisane czytelnie, z zachowaniem zasad formatowania kodów źródłowych języków wysokiego poziomu.

Zadania wymagające przedstawienia wyników w postaci tabel czy wykresów realizuj z pomocą funkcji zapisujących wektory danych do pliku. Chodzi tu o to aby wyniki na wykresie były zarchiwizowane w postaci pliku.

Na koniec zbierz sprawozdanie, pliki ze skryptami, pliki z danymi i inne niezbędne pliki, skompresuj je w jedno archiwum i zamieść jako rozwiązanie zadania w ePortalu.

7.1 Funkcje bazowe w DFT

6. linijka funkcji `sig_rdft` opisanej w punkcie 3 zawiera przepis na obliczenie wektora wartości funkcji bazowych `f` jeśli wcześniej zdefiniuje się skalar `k` oznaczający numer składowej, oraz wektor `n` z numerami próbek. Jak wygenerować ten drugi za pomocą zakresu napisane jest w 3. linijce tej funkcji.

Sporządź niezależny skrypt zaczynający się od przypisania zmiennej `N` liczby próbek z zakresu np. 10 - 50 i przypisania zmiennek `k` numeru składowej widma. Następnie w skrypcie oblicz wektor `n` oraz `k` przepisując odpowiednie fragmenty kodu wzmiankowanej wyżej funkcji. Obserwuj na podglądzie wartości zmiennych bądź w oknie "Workspace" programu Origin, jakie wartości mają tworzone przez Ciebie wektory. Czy wszystko działa jak należy?

Wykreśl następnie na dwóch wykresach część rzeczywistą wektora `f` i część urojoną wektora `f`. Działanie to wykonaj dla wskazanych przez prowadzącego następujących wartości `k`, z których przykładowe to: 0, 1, 2, -1, 10, `N-1`, `N`, `N+1`. Wykresy te umieść w sprawozdaniu, skomentuj to, jak wyglądają, czy zgadza się to z Twoimi przewidywaniami. Za wykonanie tego zadania otrzymasz 1 pkt.

7.2 Jednostronne DFT sygnału rzeczywistego

Napisz skrypt zaczynający się od przypisania zmiennym `N` liczby próbek zakresu 20-200 i częstotliwości próbkowania `fs` z zakresu 1-1000. Liczba próbek musi być parzysta, chyba, że przygotowując się do zajęć zadbałeś/zadbałaś o to, by wszystkie funkcje przygotowane w domu działały dobrze dla nieparzystej liczby próbek.

Następnie wygeneruj wektor z wartościami czasu `time` i sygnałów (jak na poprzednich zajęciach). Sygnały nazywaj `x1`, `x2`, itd. Sygnałami tymi mogą być:

- sygnał sinusoidalny o częstotliwości spełniającej kryterium Shannona i dodatkowo takiej, aby całkowita liczba jej okresów przypadała na czas trwania sygnału. Niektóre z tych częstotliwości można obliczyć dzieląc `fs` przez czynnik pierwszy `fs` lub przez dowolny iloczyn czynników pierwszych `fs`.
- sumę dwóch lub trzech sygnałów sinusoidalnych, jak wyżej, o różnych częstotliwościach.
- deltę Kroneckera zwykłą bądź opóźnioną

- impuls Gaussa
- impuls trójkątny
- sumę dwóch lub trzech sygnałów spośród wszystkich tu wymienionych
- lub inne, pomysłów można mieć tu mnóstwo

Za pomocą przygotowanych w domu funkcji oblicz wektor `freq` z częstotliwościami składowych widma jednostronnego sygnału rzeczywistego oraz same widma:

```
freq = gen_rfreq(N, fs);
y1 = sig_rdft(x1);
y2 = sig_rdft(x2);
...
```

dla obliczonych sygnałów. Zacznij od jednego, rozbudowuj skrypt w miarę postępów.

Dalej w skrypcie instrukcje:

```
figure(1)
plot(time, x1)

figure(2)
stem(freq, abs(y1))
```

spowodują wyświetlenie na rysunkach sygnału `x1` w dziedzinie czasu i widma amplitudowego `abs(y1)`[4]. Zależnie od liczby składowej widma może się okazać, że rozkaz `stem` lepiej zastąpić `plot`. Samodzielnie o tym zadecyduj, co ładniejsze.

Zależnie od własnej pomysłowości i biegłości w Octave[5] wykresy te można przedstawiać w bardziej złożony sposób - jako podwykresy na jednym rysunku, kolorami, bawiąc się grubościami linii, symboli, czy nadając tytuły i opisy osi.

W sprawozdaniu umieść pary rysunków pokazujących sygnał i jego widmo amplitudowe. Opisz osie, jeśli tego nie zapewnił skrypt Octave. Podaj parametry sygnału i opisz wygląd widma częstotliwościowego tego sygnału.

Za pełne wykonanie tego zadania otrzymasz 2 punkty.

7.3 Szybka transformata Fouriera

Zapisz skrypt, który używałeś/używałaś w poprzednim punkcie pod nową nazwą i przerób go tak, by obliczał dwustronne widma sygnałów i odpowiadający im wektor `freq`. Funkcje realizujące te zadania powinny były zostać opracowane przez Ciebie przed zajęciami. Powtórz czynności z poprzedniego punktu łącznie z raportem w sprawozdaniu.

Za pełne wykonanie tego zadania otrzymasz 2 punkty.

Jako zadanie dodatkowe możesz wykonać następujący eksperyment. Mając obliczone znormalizowane, dwustronne widmo zespolone zrekonstruuj sygnał w dziedzinie czasu za pomocą wbudowanej w Octave funkcji `ifft(y)`. Wykreśl zrekonstruowany sygnał na osobnym wykresie. Czy wyszedł z powrotem sygnał wyjściowy? Nie? A powinien. Zastanów się, co trzeba zrobić aby to nastąpiło. Wskazówki są już zawarte w instrukcji.

[4] Czy wiesz, dlaczego do wykreślenia widma amplitudowego, którego składowe zapisano w wektorze `y1` trzeba użyć funkcji `abs`?

[5] która, mam nadzieję, będzie się poprawiała wraz z kolejnymi ćwiczeniami