

Komputerowe przetwarzanie sygnałów

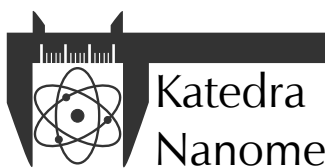
Ćwiczenie 3

Dyskretna transformacja Fouriera

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

Spis treści

1	Liczby zespolone	1
2	Dyskretna Transformacja Fouriera (ang. Discrete Fourier Transform –DFT)	2
2.1	Funkcje bazowe DFT	2
2.2	DFT	3
2.3	Jednostronne DFT sygnału rzeczywistego	5
2.4	Transformacje znormalizowane i unitarne	6
2.5	Właściwości przekształcenia Fouriera	7
2.6	Wyciek widma	7
3	Funkcje do przygotowania przed zajęciami	11
4	Origin - poznawane funkcje	13
5	Pytania i zadania na kartkówkę	13
6	Funkcje stworzone w trakcie realizacji poprzednich ćwiczeń	13
7	Zadania	14
7.1	Funkcje pomocnicze	14
7.2	Liniowość transformaty Fouriera	14
7.3	Mnożenie sygnałów i splot widm	15
7.4	Wyciek widma	15

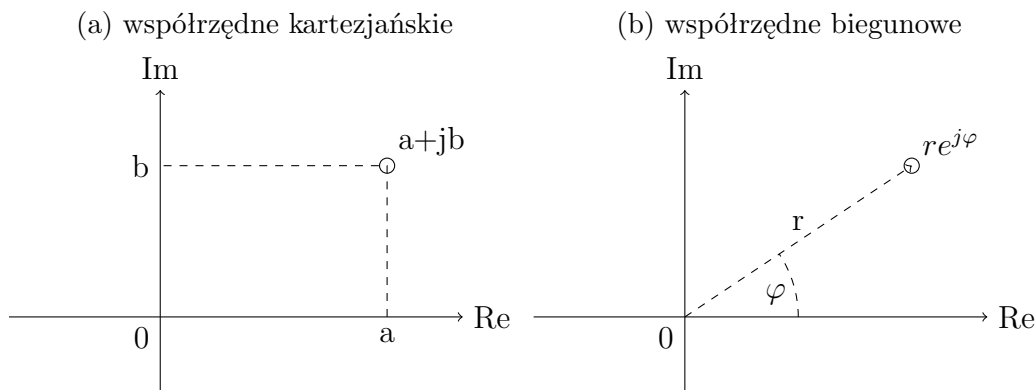


Katedra
Nanometrologii

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 pomnożonej 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 $b = r \sin \varphi$, 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 $re^{j\varphi} = re^{j(2\pi n + \varphi)}$ 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 $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$, podczas gdy kąt φ powinien przyjmować wartości od $(-\pi, \pi)$. 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 .

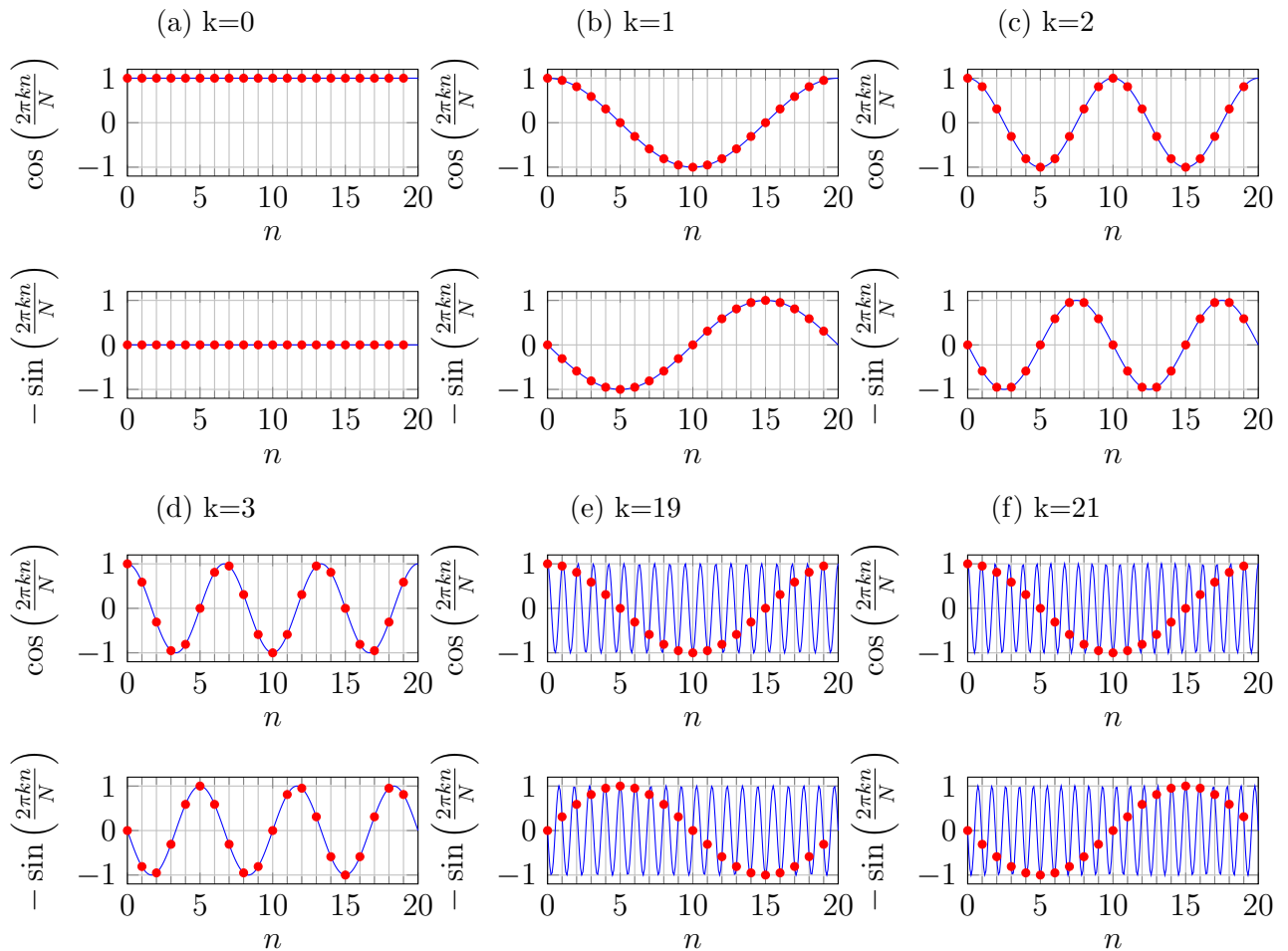
2 Dyskretna Transformacja Fouriera (ang. Discrete Fourier Transform –DFT)

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:

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

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 $N = 20$. Linia 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 rzeczywiste 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:

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

, w którym wyrażenie

$$\frac{k}{NT_s} = \frac{k f_s}{N} = f_k \quad (3)$$

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

2.2 DFT

Najważniejszą zaletą dyskretniej 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 próbek¹, to najprostszy ze sposobów na obliczenie jego widma dyskretnego $X[k]$ to realizacja poniższego równania:

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

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 bazowych². Teraz (4) można zapisać:

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

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 $f[n]$.

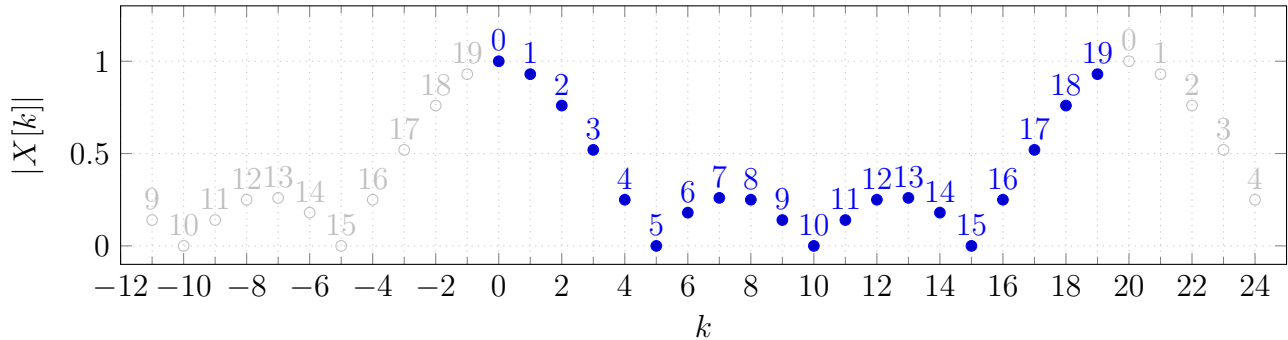
Składowe widma $X[k]$ są liczbami zespolonymi, dla których można wyznaczyć moduł i argument. Zbiór zespolonych składowych $X[k]$ nazywamy widmem zespolonym. Można je rozbić

¹Próbki te mogą być liczbami rzeczywistymi lub urojonymi. Na zajęciach będziesz mieć do czynienia tylko z sygnałami rzeczywistymi.

²Wartości elementów tego wektora przedstawiono czerwonymi kropkami na rysunku 2.

na widmo rzeczywiste i urojone lub obliczyć moduły i argumenty liczb, które tworzą widma amplitudowe i fazowe.

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 próbując z częstotliwością f_s nie powinno się widzieć w sygnale częstotliwości większych niż $0,5 \cdot f_s$, co wynika z twierdzenia Nyquista-Shannona!³

Można postąpić inaczej. Skoro składowe widma powtarzają się co N to 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 przedziału⁴:

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

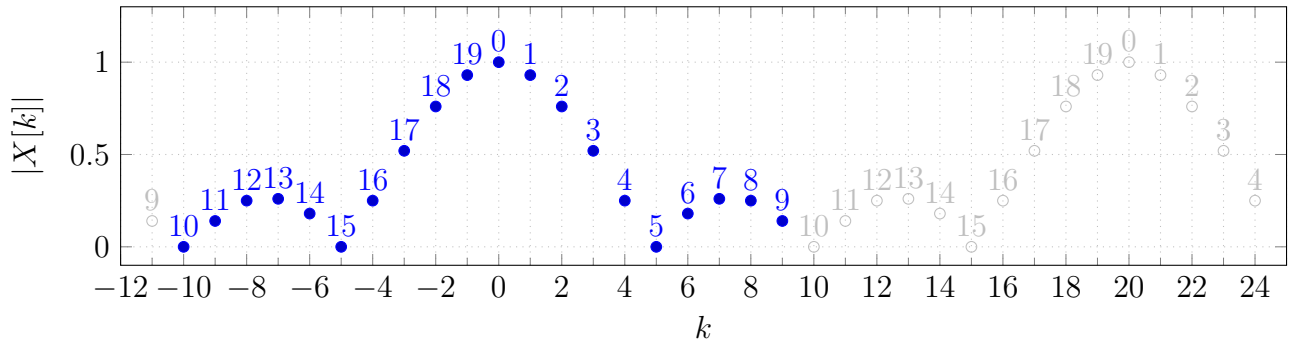
Przykładowo, jeśli $N = 20$ to zamiast k od 0 do 19 użyjemy k z przedziału od -10 do 9. Częstotliwość f_k składowej dla $k = -10$ to $-0,5 \cdot f_s$, dla $k = 0$ będzie to 0 a dla $k = 9$ będzie to $0,45 \cdot f_s$. Wartość bezwzględna częstotliwości składowych widma nie przekroczy nigdzie połowy f_s . 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)` przesuującą 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 znormalizowane widmo dwustronne obliczamy:

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

³Nie jest to błąd, Octave działa dobrze, szczegóły dlaczego tak jest wyjaśniane są na wykładzie kursu.

⁴(symbol $\lfloor \cdot \rfloor$ 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 $[-\frac{N}{2}]$ do $[\frac{N}{2} - 1]$. 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:

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

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 zespolonymi 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 amplitudowe, 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.

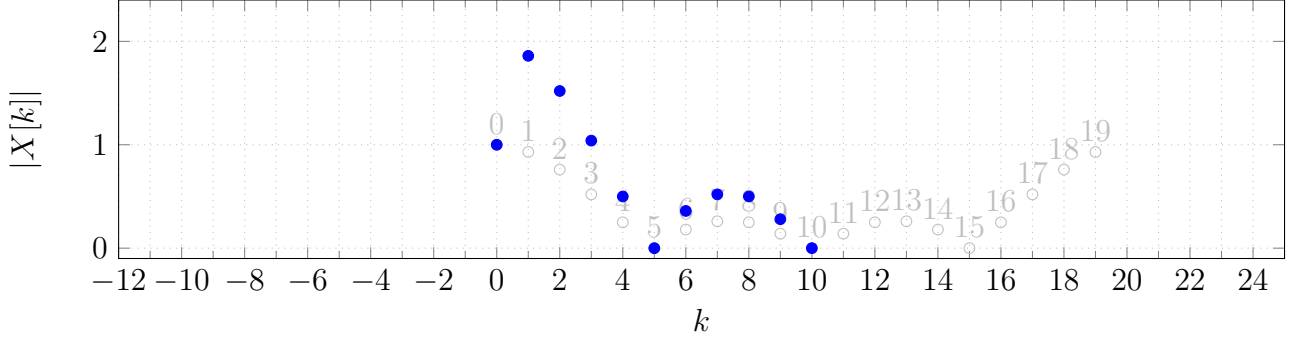
Dzięki temu pełnię informacji o widmie sygnału rzeczywistego można podać obliczając mniejszą liczbę składowych i ograniczyć się wyłącznie do przedziału od $k = 0$ do $k = \frac{N}{2}$, czyli prowadzi się je zatem dla niemal dwukrotnie mniejszej liczby iteracji, niż liczba próbek N . Widmo takie nie zawiera 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 $k \in [1, \frac{N}{2} - 1]$ 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:

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

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:

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

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 .

2.5 Właściwości przekształcenia Fouriera

Niech $FT\{x\}$ oznacza transformację Fouriera sygnału x , a $FT^{-1}\{X\}$ oznacza odwrotne przekształcenie Fouriera widma X .

Homogeniczność

Wszystkie przekształcenia Fouriera są homogeniczne, tzn. k -krotna zmiana amplitudy sygnału spowoduje k -krotną zmianę amplitudy transformaty:

$$a \cdot X = FT\{a \cdot x\} \quad (10)$$

$$a \cdot x = FT^{-1}\{a \cdot X\} \quad (11)$$

Addytywność

Każde przekształcenie Fouriera jest addytywne, tzn. transformata sumy sygnałów jest równa sumie transformat tych sygnałów:

$$FT\{x_1 + x_2\} = FT\{x_1\} + FT\{x_2\} \quad (12)$$

$$FT^{-1}\{X_1 + X_2\} = FT^{-1}\{X_1\} + FT^{-1}\{X_2\} \quad (13)$$

Homogeniczność i addytywność świadczą o liniowości przekształceń Fouriera, a można je udowodnić, korzystając bezpośrednio z definicji transformacji Fouriera.

Mnożenie sygnałów a przekształcenie Fouriera

Transformata iloczynu sygnałów jest splotem ich transformat:

$$FT\{x_1 x_2\} = FT\{x_1\} * FT\{x_2\} \quad (14)$$

W odpowiednich warunkach, transformata splotu sygnałów jest iloczynem transformat tych sygnałów, z dokładnością do czynnika skalującego:

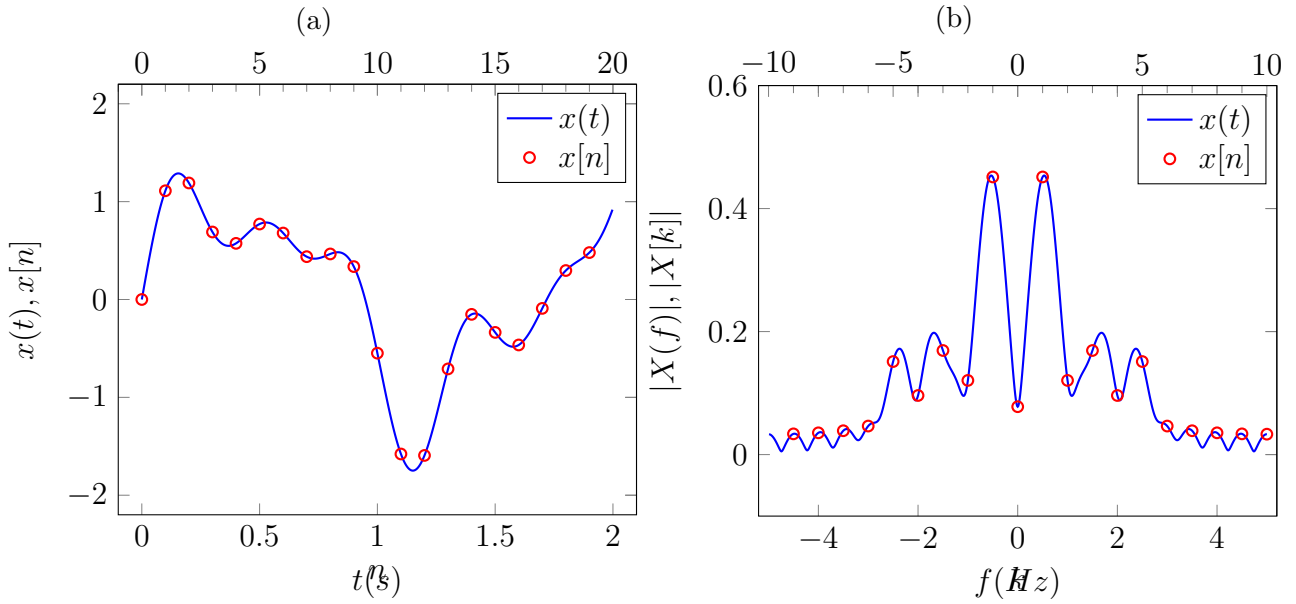
$$FT\{x_1 * x_2\} = 2\pi \cdot FT\{x_1\} \cdot FT\{x_2\} \quad (15)$$

2.6 Wyciek widma

To właściwość dotycząca tylko DFT, którą wyjaśnić można następująco. Próbkę sygnału dyskretnego reprezentują wartości sygnału ciągłego w chwilach próbkowania, to oczywiste. Co mniej oczywiste to fakt, że współczynniki widma dyskretnego też reprezentują wybrane wartości widma ciągłego dla szeregu dyskretnych częstotliwości (rysunek 6).

Innym faktem jest to, że analizując za pomocą DFT próbki sygnału, którego rzeczywisty czas trwania jest dłuższy, niż czas zbierania próbek, nie zawsze reprezentujemy w próbkach wszystkie niezbędne informacje o tym sygnale.

Weźmy jako przykład sygnał o nieskończonym czasie trwania, okresowy o modelu ciągłym $x(t) = \cos(2\pi f_0 t)$ o częstotliwości $f_0 = 10$ Hz. Zbierzmy $N = 20$ próbek z częstotliwością próbkowania $f_s = 100$ Hz. Próbkowanie z tymi parametrami trwa zatem tyle, co dwa pełne okresy sygnału, co zilustrowano na rysunku 7a. Próbkując wycinamy jedynie 200 ms z całej historii sygnału $x(t)$. Przycinanie to można zapisać matematycznie jako mnożenie sygnału z funkcją okna $w(t)$, która w tym przypadku ma postać impulsu prostokątnego trwającego od 0 przez 200 ms.



Rysunek 6: Przykład zależności sygnału (a) i widma (b) ciągłego i dyskretnego. Linia reprezentuje ciągły sygnał $x(t)$ i jego widmo amplitudowe $|X(f)|$. Punkty reprezentują próbki sygnału $x[n]$ i jego widmo amplitudowe $|X[k]|$. Oś pozioma została wyskalowana i w jednostkach fizycznych (dolna) i w „numerach” próbek czy składowych (górna).

Oznacza to, że zebrane próbki sygnału dyskretnego $x[n]$ reprezentują nie pełny sygnał $x(t)$ a jego iloczyn z funkcją okna $x(t) \cdot w(t)$. Jakie uzyskamy widmo $X_w(\omega)$ sygnału okienkowanego? Z (14) wynika, że będzie to:

$$X_w(\omega) = X(\omega) * W(\omega) \quad (16)$$

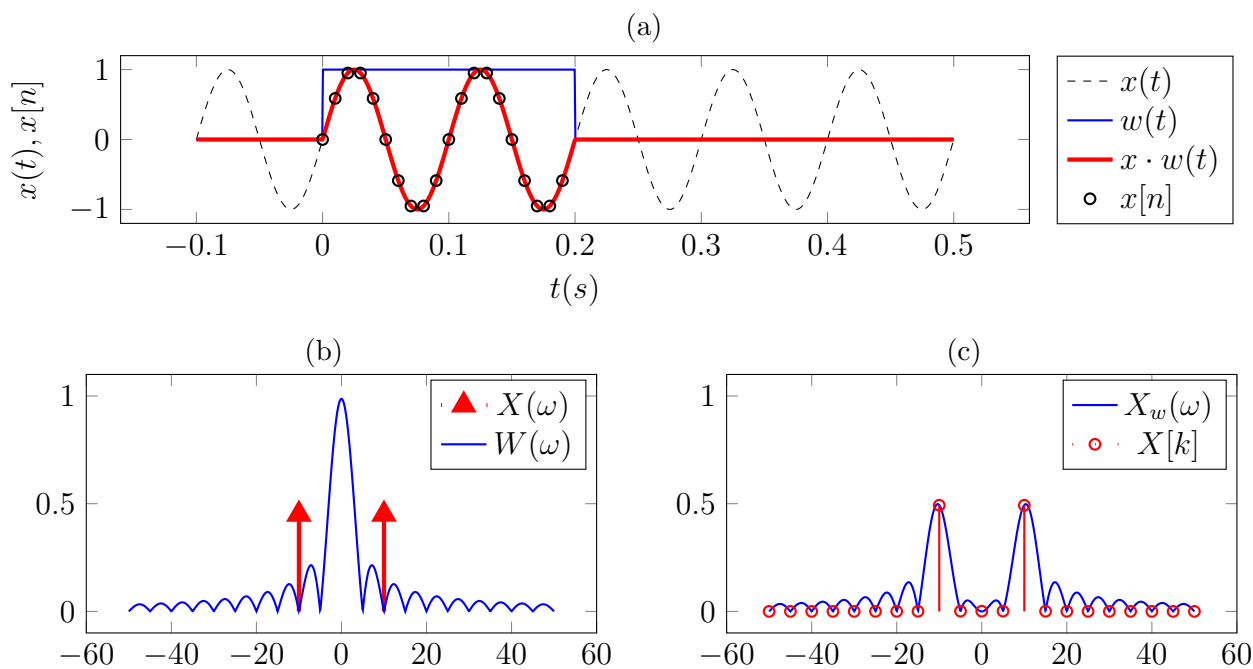
czyli widmo sygnału okienkowanego jest splotem widma sygnału przed okienkowaniem z widmem funkcji okna $W(\omega)$ (rysunek 8b).

Przykład dobrano tak, by można było zgadnąć, jaki będzie rezultat tego splotu. Dla czytelności wykres przedstawiono w funkcji częstotliwości w hercach. Widmo obustronne sygnału przed okienkowaniem to delta Diraca stojące przy częstotliwościach $\pm f_0$. Widmo funkcji okna to funkcja *sinc* o częstotliwości przekroczeń zera zależnej od szerokości impulsu prostokątnego będącego funkcją okna. Splecenie tych dwóch przebiegów da dwie kopie impulsu *sinc*, których środki występują przy częstotliwościach $\pm f_0$ (rysunek 7c).

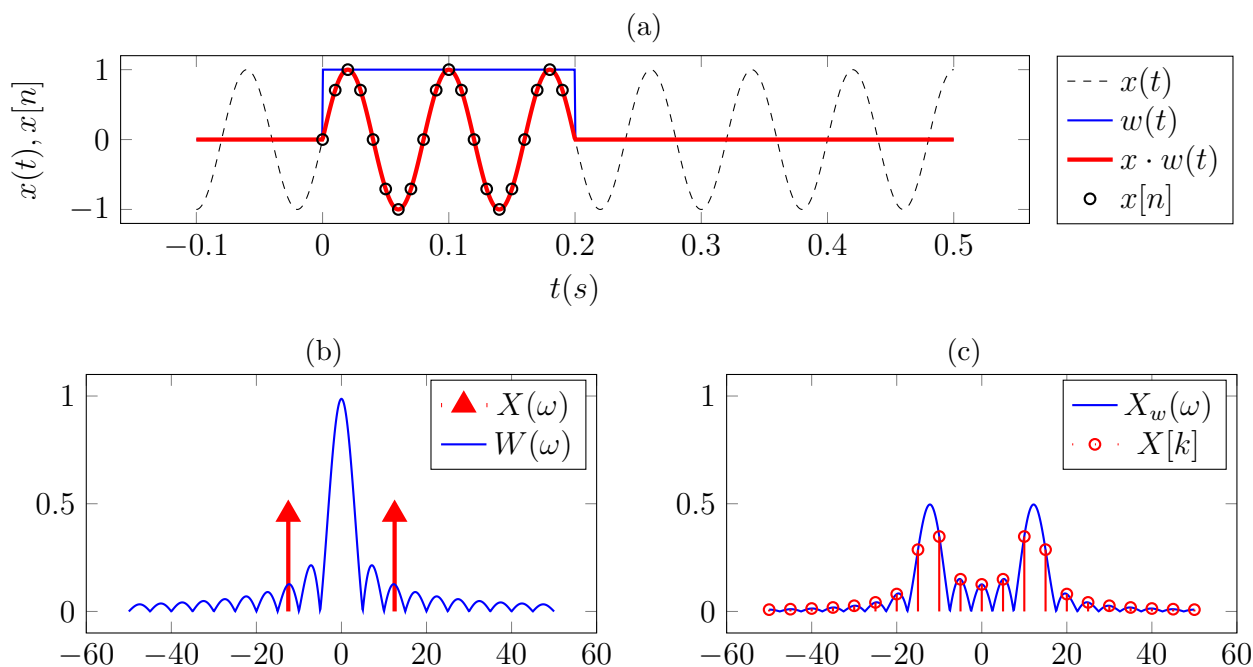
Widmo dyskretne $X[k]$ w tym przypadku ma mimo wszystko kształt zgodny z oczekiwaniami. Jedyne niezerowe współczynniki tego widma przypadają na wartości k odpowiadające środkowym listkom przebiegów *sinc*. Wszystkie pozostałe trafiły na miejsca, w których funkcja *sinc* przecina 0!

Stało się to jednak tylko przez zbieg okoliczności - w oknie próbkowania ujęto całkowitą liczbę okresów sygnału okresowego $x(t)$. Jeśli nie będzie to miało miejsca dojdzie do **wycieku widma**. Przykładowo, na rysunku 8 zilustrowano sytuację zbliżoną do poprzedniej, ale zwiększono częstotliwość sygnału okresowego do 12,5 Hz.

Postępując analogicznie jak poprzednio uzyskuje się widmo dyskretne, którego składowe nie wskazują jednoznacznie na to, jaką postać miał sygnał próbkowany. Nie ma wśród składowych widma dyskretnego takiej, której częstotliwość dokładnie pasowałby do częstotliwości sygnału okresowego. Składowe dyskretne przypadają na 0, 5, 10, 15, 20, itd. herców a sygnał miał częstotliwość 12,5 Hz, leżącą dokładnie pośrodku pomiędzy dwiema z nich. Uzyskuje się widmo dyskretne w którym składowe 10 i 15 herców są największe, ale oprócz tego informacja o sygnale rozlała się, wyciekła, do wielu sąsiednich składowych, stąd nazwa zjawiska - wyciek widma. W wypadku, gdy częstotliwość sygnału okresowego przypada dokładnie pośrodku pomiędzy dwie-



Rysunek 7: Ilustracja widma sygnału okresowego $x(t)$ próbkowanego skończony czas: a) okienkowanie sygnału ciągłego funkcją $w(t)$ o kształcie impulsu prostokątnego trwającego 2 okresy sygnału okienkowanego ze wskazaniem próbek w ten sposób pobranych, b) widma sygnału okresowego oraz funkcji okna, c) splot widm sygnału i funkcji okna ze wskazaniem częstotliwości, jakie znajdują się w widmie dyskretnym.

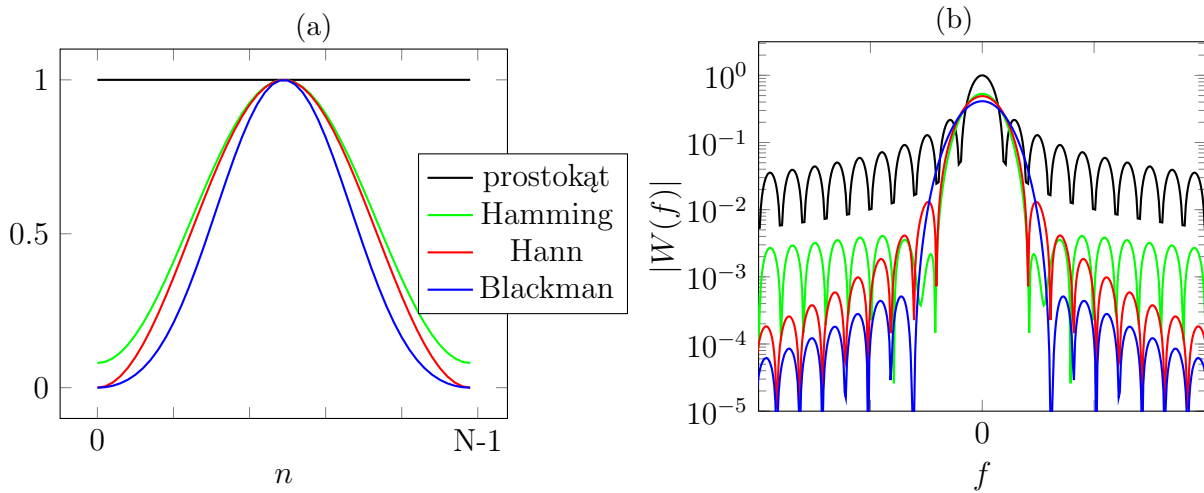


Rysunek 8: Ilustracja widma sygnału okresowego $x(t)$ próbkowanego skończony czas w sposób powodujący wyciek widma: a) okienkowanie sygnału ciągłego funkcją $w(t)$ o kształcie impulsu prostokątnego trwającego 2,5 okresów sygnału okienkowanego, b) widma sygnału okresowego oraz funkcji okna, c) splot widm sygnału i funkcji okna ze wskazaniem częstotliwości, jakie znajdują się w widmie dyskretnym.

ma składowymi dyskretnymi uzyskuje się maksymalny wyciek widma. Przypominając sobie wiedzę z poprzedniego ćwiczenia o zależności częstotliwości składowych od parametrów próbkowania naucz się przewidywać, czy dla sygnału okresowego o częstotliwości f_0 wyciek wystąpi czy nie.

Podsumowując, stosując DFT do analizy sygnałów należy zdawać sobie sprawę z możliwości wystąpienia wycieku widma. Nie zawsze wiele niezerowych składowych w takim widmie oznacza, że sygnał poddawany analizie widmowej był sygnałem je zawierającym.

Wyciek widma jest tym intensywniejszy, im „szersze” jest widmo funkcji okna. Można stosować inne funkcje okienkujące, niż impuls prostokątny. Spowoduje to, że wyciek będzie wpływał na mniejszą liczbę składowych widma. Wadą tego rozwiązania jest to, że nawet jeśli analizowany DFT sygnał nie powodował wycieku, to w widmie nie zobaczymy jednej składowej go reprezentując a też kilka.



Rysunek 9: Przykłady funkcji okienkujących: a) w dziedzinie czasu (lub nr. próbki), b) w dziedzinie częstotliwości.

Stosuje się różne funkcje okienkujące mające minimalizować wyciek widma. Ich kształty pokazano na rysunku 9. Mają one przebieg impulsu łagodniej zaczynającego się i kończącego się, niż impuls prostokątny. Wyraża się je wzorami podanymi w tabeli 1.

Tabela 1: Wybrane funkcje okna dla N próbek wraz z funkcją Octave generującą tablicę z wartościami funkcji okna.

Nazwa	wyrażenie	funkcja Octave
prostokąt	$w(n) = 1$	<code>ones(1,N)</code>
Hamminga	$w(n) = 0,5383 - 0,4616 \cos\left(\frac{2\pi n}{N-1}\right)$	<code>hamming(N)</code>
Hanna	$w(n) = 0,5 - 0,5 \cos\left(\frac{2\pi n}{N-1}\right)$	<code>hann(N)</code>
Blackmana	$w(n) = 0,42 - 0,5 \cos\left(\frac{2\pi n}{N-1}\right) + 0,08 \cos\left(\frac{4\pi n}{N-1}\right)$	<code>blackman(N)</code>

Widma rozmaitych funkcji okna ilustrują co się za ich pomocą da uzyskać. Okno prostokątne ma wiele listków bocznych o dużych amplitudach ale wąski listek środkowy. Okno Hamminga ma listki boczne o amplitudach mniejszych o ponad rząd wielkości ale szerszy listek środkowy. Kolejne okna, Hanna, Blackmana coraz lepiej tłumią listki boczne kosztem poszerzenia głównego.

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 składowej 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.

Przygotuj funkcję:

```
function f = sig_sinfreq(N, fs, o)
f = ...;
endfunction
```

obliczającą częstotliwość w hercach f funkcji sinusoidalnej, której o okresów rozciągać się będzie na N próbek próbkowanych z częstotliwością f_s . Przetestuj jej działanie w prostym skrypcie, którego fragment przedstawiono poniżej:

```
N = ...;
fs = ...;
time = gen_time(N, fs);
freq = sig_sinfreq(N, fs, ...); # tu podaj liczbę okresów
x1 = gen_sin(time, freq, 1, 0);
plot(...)
```

Po podaniu odpowiednich liczby próbek i częstotliwości próbkowania obliczana jest `freq`. Po wygenerowaniu przebiegu sinusoidalnego i jego wyplotowaniu sinus powinien mieć tyle okresów, ile podano jako ostatni argument funkcji `sig_sinfreq`. Upewnij się, że działa ona dobrze dla różnych N i fs .

Następnie sprawdź, jak działają funkcje służące do generacji okien, wymienione w tabeli 1. Ich użycie może wymagać dodania biblioteki `signal`, co można zrobić następującym poleceniem:

```
pkg load signal
```

Przyjmij niewielkie N na przykład równe 10. Fragment skryptu bądź polecenia wpisywane w linię poleceń mogą wyglądać następująco:

```
N = 10;
w1 = ones(1,N)
w2 = hamming(N)
```

Umyślnie nie użyto myślników na końcach linii. Zaobserwuj dzięki wynikom przypisania wyświetlonym w linii poleceń i/lub w oknie *Workspace* czym różnią się `w1` i `w2`. Są one tablicami o innych proporcjach. Czasem to nie przeszkadza ale dla uniknięcia konsekwencji tych różnic należy niekiedy posługiwać się znakiem apostrofu, który dokonuje transpozycji tablicy:

```
w2 = hamming(N)'
```

Sprawdź w ten sposób wszystkie wymienione w tabeli funkcje obliczające funkcje okien. Wyniki działania wszystkich z nich w razie potrzeby zmodyfikuj tak, aby ostatecznie uzyskać poziomy wektor wartości funkcji okna. Jeśli chcesz, wykreśl sobie przebiegi tych funkcji. Te ładniej wyglądają dla większej liczby N .

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 dwustronnym przywraca jego pierwotną formę
`dot(x, y)` - obliczenie iloczynu skalarnego wektorów x i y
`hann(N)` - zwracają współczynniki filtra okna Hann'a o długości N
`hamming(N)` - zwracają współczynniki filtra okna Hamminga o długości N
`blackman(m)` - zwracają współczynniki filtra okna Blackmana o długości N
`trapez(time, x)` - numerycznie wyznacza całkę z punktów x przy użyciu metody trapezowej skalując jej wartość używając wektora `time` do skalowania
`operator '` - transpozycja tablicy

5 Pytania i zadania na kartkówkę

1. Jaką częstotliwość będzie miała 5. (licząc od zera) składowa DFT sygnału o liczbie próbek równiej 100 i okresie próbkowania równym 0,1 s?
 2. Oblicz składową $k = 1$ DFT sygnału rzeczywistego $x[n] = [1 \ 3 \ 0 \ 2 \ 3 \ 1]$.
 3. Oblicz składową $k = 2$ DFT sygnału $x[n] = [1 \ 3 \ j \ 0 \ 1 \ 2]$.
 4. Czy obliczając DFT dla sygnałów rzeczywistych możemy używać równania (4)?
 5. Jak będzie się różniła amplituda współczynników DFT obliczanych równaniem (4) i (8) dla sygnałów rzeczywistych?
 6. 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 .
 7. Dane są dwa sygnały $x_1[n]$ i $x_2[n]$ próbkowane w identyczny sposób. Wiesz, że $X_1[2] = 3 + 2j$, oraz $X_2[2] = 4 - 3j$. Ile wynosi $X[2]$, jeżeli $x[n] = 3x_1[n] - 2x_2[n]$?
 8. Jak wygląda widmo sygnału będącego iloczynem impulsu prostokątnego i funkcji $\cos(\omega_0 t)$?
 9. Próbkę sygnału $x[n]$ o pewnym numerze n_1 ma wartość 5. Funkcja okna dla tej próbki ma wartość 0,34. Ile wynosi wartość próbki n_1 sygnału poddanego okienkowaniu?
 10. Dobierz częstotliwość sinusoidy spróbkowanej z częstotliwością $f_s = 10 \text{ kHz}$ tak, aby dla 100 punktowej DFT uniknąć zjawiska wycieku.
 11. Dobierz częstotliwość sinusoidy spróbkowanej z częstotliwością $f_s = 10 \text{ kHz}$ tak, aby dla 100 punktowej DFT zaobserwować maksymalny wyciek.
- 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

W trakcie zajęć sprawdzana będzie znajomość implementacji, prawidłowość działania i umiejętność korzystania z funkcji opisanych w poniższych zadaniach. Powinny one zostać przygotowane przed przyjściem na zajęcia.

Część zadań nich polega na usunięciu błędów w podanych definicjach funkcji. Każda z definicji wykonuje operacje, które zostały teoretycznie omówione w poprzednich rozdziałach niniejszej instrukcji. Za każde zadanie można otrzymać podaną przy nim liczbę punktów pod warunkiem, że zostanie ono w całości poprawnie zrealizowane.

Wszystkie funkcje, których argumentem bądź wynikiem jest widmo sygnału w postaci tablicy, powinny w pierwszym jej elemencie (czyli elemencie o indeksie 1) zawierać pierwszy element widma (czyli obliczony dla początkowej wartości k).

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

7.1 Funkcje pomocnicze

Należy usunąć błędy w funkcji do obliczania częstotliwości (patrz p. 2.3) poszczególnych składowych widma Fouriera sygnału rzeczywistego `gen_rfreq`. Funkcja ta przyjmuje jako parametry liczbę próbek sygnału N , którego widmo ma być prezentowane, oraz częstotliwość próbkowania fs .

Oprócz tego należy napisać ciało podobnej funkcji `gen_cfreq`, która ma generować częstotliwości poszczególnych składowych widma Fouriera sygnału w jego ogólnej, zespolonej formie, przy czym widmo ma być wyśrodkowane wokół składowej stałej za pomocą funkcji `fftshift` (patrz p. 2.1).

Działanie funkcji powinno być zaprezentowane w taki sposób, aby jednoznacznie udowodnić, że funkcje działają poprawnie. Za prawidłowe wykonanie zadania otrzymasz 1 pkt.

```
function freq = gen_rfreq(N, fs)
    freq = zeros(1, N/2);
    for k = 0:N/2
        freq(k) = k*fs/N;
    endfor
endfunction
```

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

7.2 Liniowość transformaty Fouriera

Ustal liczbę próbek $N = 30..200$ i częstotliwość próbkowania fs , zgodnie ze wskazówkami prowadzącego. Przygotuj skrypt obliczający dwa przebiegi sinusoidalne x_1 i x_2 . Każdy z sumo-

wanych sinusów powinien mieć kontrolowaną przez Ciebie za pomocą zmiennych `o1` i `o2` liczbę okresów mieszczących się w oknie próbkowania, z zakresu 3..10.

Oblicz obustronne widma tych sygnałów, sumę widm tych sygnałów oraz widmo sumy tych sygnałów. Wykaż liniowość transformacji Fouriera porównując wykresy widm amplitudowych tak uzyskanych.

Za prawidłowe wykonanie zadania otrzymasz 1 pkt.

7.3 Mnożenie sygnałów i splot widm

Używając stworzonych przez siebie funkcji wygeneruj sygnał dwa sygnały: harmoniczny i impulsowy (impuls trójkątny, impuls Gaussa lub inny). Na osobnych wykresach zaprezentuj widma amplitudowe tych sygnałów oraz widma iloczynu tych sygnałów. Pomocna będzie funkcja `figure` pozwalająca na wykreślenie kilku niezależnych wykresów. Jak wytłumaczysz kształt widma amplitudowego iloczynu sygnałów? Za prawidłowe wykonanie zadania otrzymasz 1 pkt.

7.4 Wyciek widma

Ustal liczbę próbek $N = 200..500$ i częstotliwość próbkowania fs , zgodnie ze wskazówkami prowadzącego. Przygotuj skrypt generujący przebieg podobny, jak w zadaniu 7.2, czyli sumę dwóch przebiegów sinusoidalnych, ale teraz niech `o1` i `o2` będą w przedziale 15..25 oraz niech **jeden** z sumowanych przebiegów sinusoidalnych ma liczbę okresów będących liczbą ułamkową, aby spowodować silny wyciek widma.

Zadanie polega na porównaniu ze sobą efektów używania różnych funkcji okna. Zakładając, że sygnał będący sumą sinusów jest w wektorze w zmiennej `x` oraz zdefiniowane są standardowe zmienne określające liczbę próbek, częstotliwość próbkowania, itp, fragment skryptu realizującego takie zadanie może wyglądać następująco:

```
w = ...; # tu wstaw funkcje okna
x_w = x .* w; # nałożenie okna na wektor x
y = sig_fft(x);
y_w = sig_fft(x_w);
...
```

Porównaj ze sobą trzy widma amplitudowe: jedno uzyskane z oknem prostokątnym i dwa inne z oknami wskazanymi przez prowadzącego. Opisz cechy tak uzyskanych widm. Jak zmieniają się w widmie składowe związane z sinusem o częstotliwości niepowodującej wycieku i tego, przy którym wyciek wystąpił?

Dla co najmniej jednego z okien (ale nie prostokątnego) wykreśl wykres sygnału zokienkowanego `x_w` w zależności od czasu. Skomentuj jego wygląd, porównaj z przebiegiem sygnału przed okienkowaniem.

Za prawidłowo wykonane zadanie otrzymasz 2 pkt.