

Przetwarzanie sygnałów

Ćwiczenie 3

Dyskretna transformacja Fouriera

dr hab. inż. Grzegorz Józwiak, 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 sygnału zespolonego	3
2.3 Jednostronne DFT sygnału rzeczywistego	4
2.4 Transformacje znormalizowane i unitarne	6
3 Pytania i zadania na kartkówkę	6
4 Funkcje stworzone w trakcie realizacji poprzednich ćwiczeń	7
5 Zadania	7
5.1 Funkcje pomocnicze	7
5.2 Jednostronne DFT sygnału rzeczywistego	8
5.3 Szybka transformata Fouriera	8

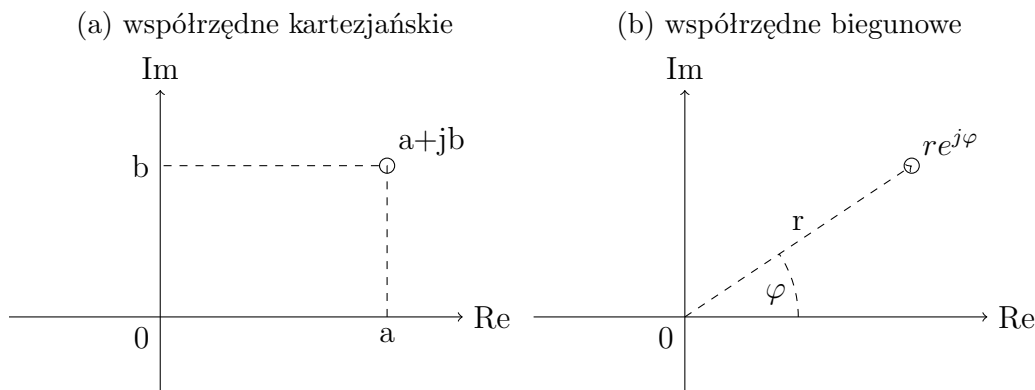


Katedra
Nanometrologii

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 $b = r \sin \varphi$, 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 $re^{j\varphi} = re^{2\pi n + j\varphi}$ 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 .

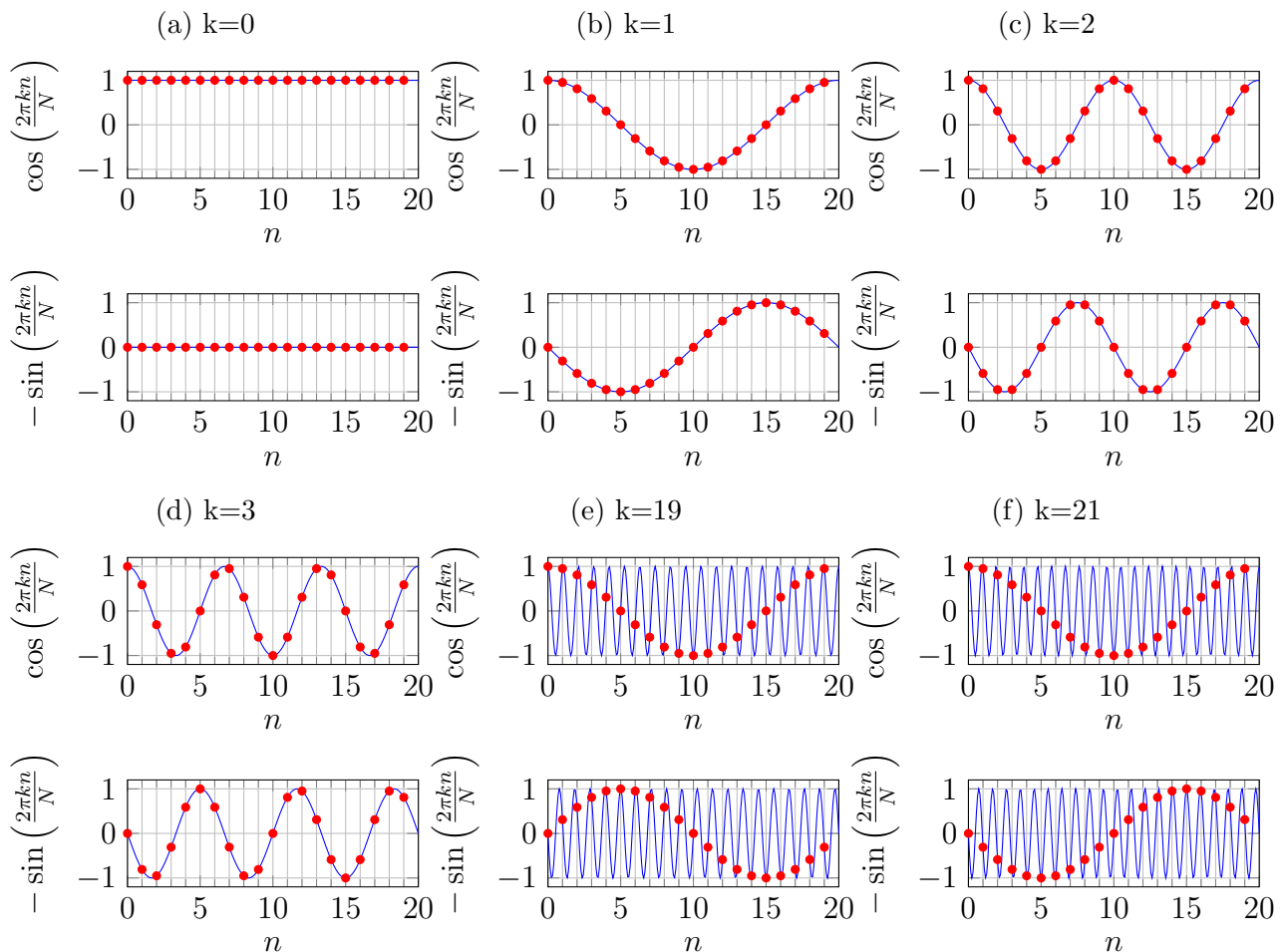
2 Dyskretna Transformacja Fouriera (ang. Discrete Fourier Transform –DFT)

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 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:

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

, 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 sygnału zespolonego

Najważniejszą zaletą DFT jest to, że można ją obliczyć numerycznie. Fakt ten stał się szczególnie istotny w momencie pojawienia się układów mikroprocesorowych. Najprostszym algorytmem umożliwiającym obliczenie współczynników (parametrów skalujących amplitudę, nazywanych również składowymi widma - nie mylić z parametrem k i częstotliwością) poszczególnych funkcji bazowych dla sygnałów zespolonych jest bezpośrednia implementacja równania:

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

Ponieważ funkcje bazowe są ortogonalne, to równanie może mieć tak prostą postać, w którym zwykły, skalarny iloczyn wykorzystuje się do obliczenia wartości współczynników $X[k]$ widma.

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

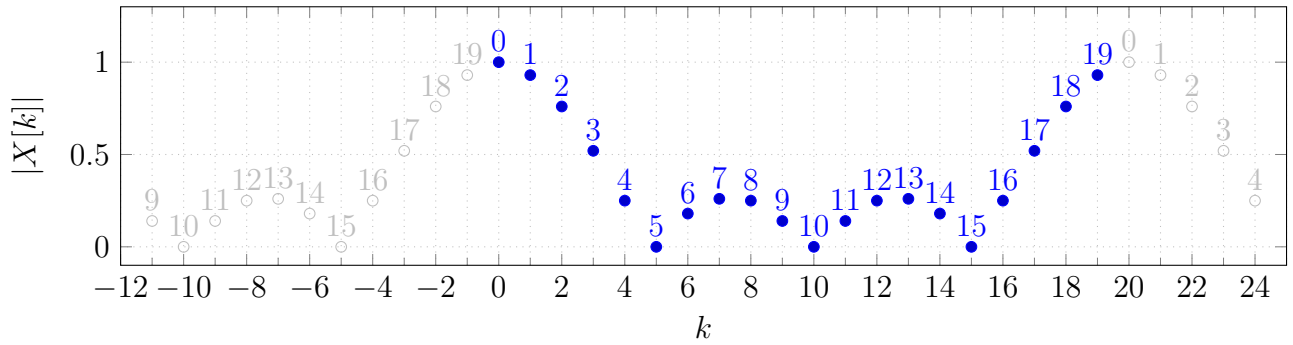
Jak wykazano w rozdziale 2.1 ma sens obliczanie maksymalnie N różnych funkcji bazowych, bo potem zaczną się one powtarzać. Tak samo współczynniki widma oblicza się dla N różnych k , po potem zaczną się one powtarzać. Często wybiera się zakres wartości całkowitych $k \in [0, N - 1]$. Tak na przykład działają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

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 $0,5 \cdot f_s$!¹

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 przedziału² od $[-\frac{N}{2}] + 1$ do $[\frac{N}{2}]$.

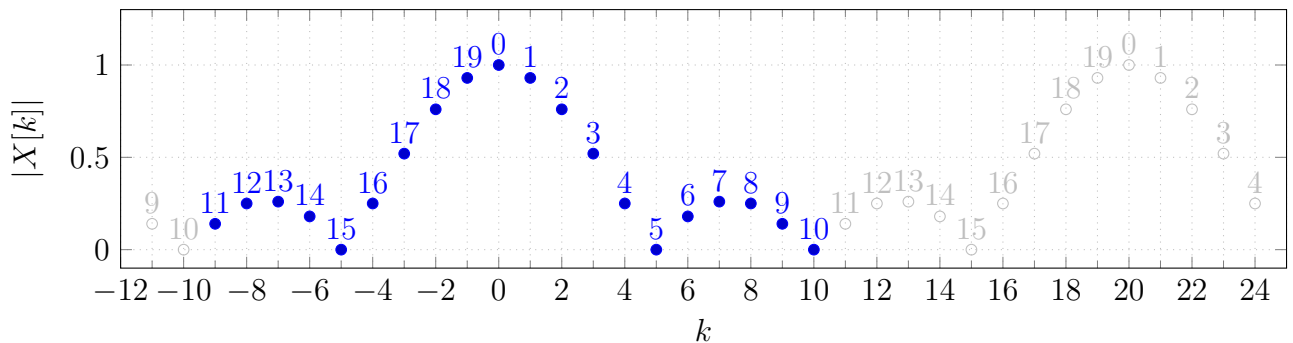
¹Nie jest to błąd, Octave działa dobrze, szczegóły dlaczego tak jest wyjaśniane są na wykładzie kursu.

²(symbol $\lfloor \cdot \rfloor$ oznacza zaokrąglenie wartości w dół - N może być bowiem również nieparzyste a k musi być całkowite).



Rysunek 3: Przykładowe dyskretne widmo amplitudowe impulsu prostokątnego o czasie trwania 4 próbek 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.

Przykładowo, jeśli $N = 20$ to zamiast k od 0 do 19 użyjemy k z przedziału od -9 do 10. Częstotliwość f_k składowej dla $k = -9$ to $-0.45 \cdot f_s$, dla $k = 0$ będzie to 0 a dla $k = 9$ będzie to $0,5 \cdot f_s$. 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 rozciąga się na obie strony składowej $k = 0$ i nazywamy je dwustronnym. Jego przykład pokazano na kolorowej części wykresu z rysunku 4.



Rysunek 4: Przykładowe dwustronne dyskretne widmo amplitudowe prezentujące te same informacje, co na rysunku 3. Kolorem zaznaczono zakres k od $[-\frac{N}{2}] + 1$ do $[\frac{N}{2}]$. 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:

$$x[n] = \sum_{k=0}^{N-1} X[k] e^{j \frac{2\pi kn}{N}} \quad (5)$$

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 amplitudowego, co widać np. na rysunku 3.

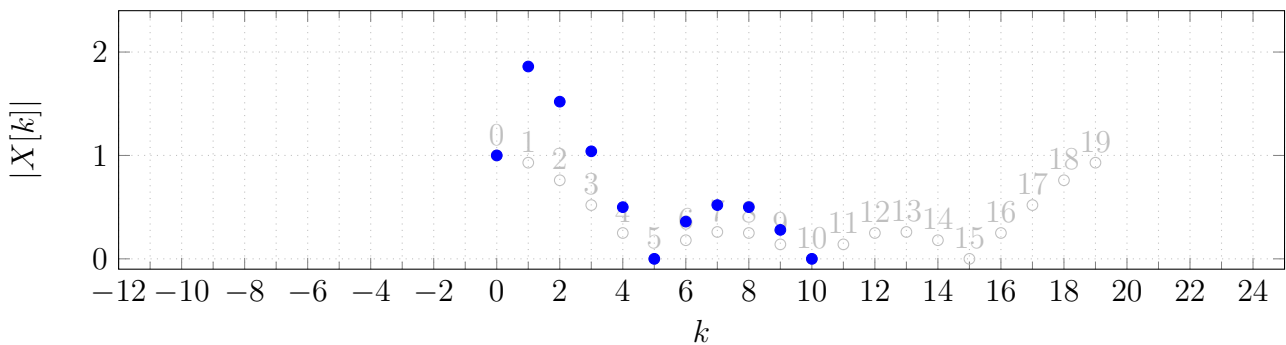
Dla $k = 0$ i $k = \frac{N}{2}$ już takiej zależności nie ma - nie ma „pary” zależnych od siebie składowych.

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.

Współczynniki 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 współczynników da taki sam efekt, jak przy widmie dwustronnym. Współczynniki bez pary, czyli dla $k = 0$ i $k = \frac{N}{2}$ oblicza bez zmian, bezpośrednio z (4). Sposób obliczenia współczynników widma jednostronnego prezentuje się zatem następująco:

$$X[k] = \begin{cases} \frac{2}{N} \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi kn}{N}} & \text{dla } k \in [1, \frac{N}{2} - 1] \\ \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi kn}{N}} & \text{dla } k = \{0, \frac{N}{2}\} \end{cases} \quad (6)$$

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 punkty 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.

Obliczenia z równań (6) można dodatkowo uprościć matematycznie rozbijając powyższy wzór na dwa osobne - jeden służący do obliczenia części rzeczywistej składowej widma $Re[k] = \Re(X[k])$ i drugi do obliczenia części urojonej składowej widma $Im[k] = \Im(X[k])$. Zapis zespolony funkcji bazowych zmieniamy na rzeczywisty, zgodnie z (1). Biorąc powyższe pod uwagę, wzór na obliczenie składowych widma połówkowego sygnału rzeczywistego prezentuje się następująco:

$$Re[k] = \begin{cases} \frac{2}{N} \sum_{n=0}^{N-1} x[n] \cos\left(\frac{2\pi kn}{N}\right) & \text{dla } k \in \left[1, \frac{N}{2} - 1\right] \\ \frac{1}{N} \sum_{n=0}^{N-1} x[n] \cos\left(\frac{2\pi kn}{N}\right) & \text{dla } k = \left\{0, \frac{N}{2}\right\} \end{cases} \quad (7)$$

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

Zapis jest bardziej rozbudowany, niż w (6) ale wszystkie operacje są wykonywane w dziedzinie liczb rzeczywistych.³

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:

$$x[n] = \sum_{k=0}^{N/2} \left[Re[k] \cos\left(\frac{2\pi kn}{N}\right) + Im[k] \sin\left(\frac{2\pi kn}{N}\right) \right] \quad (9)$$

Możemy zauważyć, że do zsyntetyzowania sygnału $x[n]$, wykorzystujemy teraz mniejszą liczbę par ($Re[k]$ i $Im[k]$) współczynników ($N/2+1$) 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), (6) 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, pomijana jest jedynie operacja dzielenia przez N .

3 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.

³To zasada dość ogólna. Jeżeli mamy równanie, w którym występują wartości zespolone, wektory czy macierze, to często jest ono zwięzłym przedstawieniem skomplikowanych operacji matematycznych, które i tak de facto trzeba wykonać.

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 (6) dla sygnałów rzeczywistych?
9. Jak będzie się różniła amplituda współczynników DFT o indeksie 0 i $N/2$ obliczanych równaniem (4) i (6) 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.

4 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

5 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.

5.1 Funkcje pomocnicze

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

Napisać należy też ciało podobnej funkcji `gen_cfreq`, która ma obliczać tablicę zawierającą częstotliwości f_k poszczególnych składowych widma Fouriera sygnału w jego ogólnej, zespolonej formie. Przyjmujemy, że wartość bezwzględna żadnej wartości f_k nie może przekroczyć połowy

`fs` a widmo będzie być wyśrodkowane wokół składowej $k = 0$ (patrz p. 2.1). Sprawdź, czy funkcja zadziała właściwie dla przypadku, w którym N jest nieparzyste.

Działanie funkcji powinno być zaprezentowane w taki sposób, aby jednoznacznie udowodnić, że funkcje działają poprawnie. Na przykład wykaż najpierw ręcznie, jakie powinny być wartości i wskaż, że funkcje zwracają pożądane wyniki. Najprościej robić to na przykładach z niewielkimi wartościami N .

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
```

5.2 Jednostronne DFT sygnału rzeczywistego

Zadanie drugie polega na usunięciu błędów z funkcji obliczającej jednostronne DFT sygnału rzeczywistego `sig_rdft`.

Funkcja jako parametr przyjmuje tablicę liczb rzeczywistych, a w wyniku zwraca tablicę liczb zespolonych, będących zespolonym widmem częstotliwościowym.

Działanie funkcji należy przetestować dla sygnału, dla którego znamy prawidłowy wynik DFT. Sygnał ten należy wygenerować używając funkcji, które powstały przy realizacji poprzedniego ćwiczenia. Wykorzystaj takie funkcje, dla której postać ich widm jest Tobie znana. Prezentując działanie funkcji należy wyjaśnić, dlaczego uważasz, że funkcja działa prawidłowo. Za prawidłowe wykonanie zadania otrzymasz 2 pkt.

```
function y=sig_rdft(x)
    N=length(x);
    y=zeros(1,N/2+1);
    for k=0:N/2
        Tr=0;
        Ti=0;
        for n=0:N-1
            Tr+=2*x(n)*cos(2*pi*k*n/N)/N;
            Ti+=-2*x(n)*sin(2*pi*k*n/N)/N;
        endfor
        y(k+1) = Tr + 1i * Ti;
    endfor
endfunction
```

5.3 Szybka transformata Fouriera

Program Origin posiada wbudowaną funkcję `fft`. Jest ona implementacją unitarnej, szybkiej transformaty Fouriera sygnału zespolonego. Argumentem tej funkcji jest tablica zawierająca próbki sygnału, analogicznie jak `sif_rdft` z poprzedniego punktu. Rezultatem jest tablica zawierająca zespolone współczynniki widma, obliczone dla $k \in [0, N - 1]$.

Stwórz bliźniacze funkcje `sig_fft` i `sig_fft_N`:

```
function y = sig_fft(x)
```

```
...
```

```
endfunction
```

```
function y = sig_fft_N(x)
```

```
...
```

```
endfunction
```

obliczające dwustronne widmo zespolone sygnału x używając funkcji `fft`. Jedyną różnicą pomiędzy nimi jest taka, że `sig_fft` zwraca widmo znormalizowane, `sig_fft_N` widmo uzyskane transformacją unitarną, nienormalizowaną.

Obie funkcje mają wywoływać wbudowaną w Origin funkcję `fft` i modyfikować zwracaną przez nią tablicę ze współczynnikami widma unitarnego. Wartości należy uporządkować tak, jakby były obliczane dla k z przedziału od $\lfloor -\frac{N}{2} \rfloor + 1$ do $\lfloor \frac{N}{2} \rfloor$ (patrz punkt 2.2). Funkcje te mają stanowić zatem parę dla funkcji `gen_cfreq`, która zwraca tablicę wartości f_k składowych widma. Jak to osiągnąć?

Przykładowo, jeśli tablica x zawiera 20 próbek, to `y=fft(x)` spowoduje wpisanie do tablicy y dwudziestu wartości zespolonych, jak zaznaczone kolorem na rysunku 3. Pamiętaj jednak, że składowa $X[0]$ obliczona dla $k = 0$ wyląduje w $y(1)$, itd.

Znormalizowanie wartości jest proste - należy tablicę podzielić przez liczbę próbek. Drugie wymaganie co do rezultatu osiągnąć jest trudniej. Przyjrzyj się uważnie rysunkowi 4. Oznaczono na nim to, które elementy widma obliczonego za pomocą funkcji `fft` powinny znaleźć się przy których wartościach k .

Widać tu, że rezultat ten możemy osiągnąć częściowo przestawiając kolejność elementów w wynikowej tablicy, a częściowo posługując się wiedzą o powtarzających się funkcjach bazowych. W przykładzie naszym bowiem $X[-1] = X[-1 + N] = X[19]$, $X[-2] = X[-2 + N] = X[18]$, itd. Realizacja tego programowo, dla dowolnego N stanowi istotę zadania.⁴

Działanie funkcji `sig_fft` zademonstruj generując czasy i wartości próbek sygnału funkcjami `gen_time` i `gen_sin` dla wybranej przez siebie liczby próbek i częstotliwości próbkowania, po czym wygeneruj częstotliwości składowych widma i wartości widma funkcjami `gen_cfreq` i `sig_fft`. Teraz widmo amplitudowe można przedstawić na wykresie za pomocą polecenia

```
stem(f,abs(y))
```

jeżeli f to tablica z częstotliwościami składowych widma a y to tablica z wartościami składowych widma. Funkcja `stem` działa podobnie jak `plot`, ale wykres ma charakterystyczną formę, która dobrze sprawdza się do prezentacji widm o liczbie składowych nie przekraczających kilkadziesiąt. Jeśli N jest większe, widma rysuj funkcją `plot`.

Sprawdź, czy funkcja `sig_fft` działa dobrze. Zastanów się, jakie sygnały użyć by to sprawdzić. Muszą to być takie sygnały, których kształt widma amplitudowego dasz radę przewidzieć.

Funkcja `sig_fft_N` ma działać w sposób niemal identyczny do `sig_fft` pomijając jedynie operację normalizowania widma.

Za prawidłowe wykonanie zadania otrzymasz 2 pkt.

⁴Można ten fragment kodu przetestować bardzo prosto. Tymczasowo zamień w ciele tworzonej funkcji przypisanie `y=fft(x)`; na przypisanie `y=0:(length(x)-1);`. W ten sposób tablica zawierać będzie zamiast składowych widma numery tych składowych. Teraz, po zmianie kolejności elementów w tej tablicy, powinna ona zawierać liczby w takiej kolejności ułożone, jak zaznaczone kolorem na rysunku 4, czyli dla $N = 20$ powinno to być: 11, 12, ... 19, 0, 1, ..., 9, 10. Sprawdź, czy otrzymujesz dobre wyniki dla innych N , w tym nieparzystych.