

Ćwiczenie: Własności dyskretnej transformaty Fouriera (DFT)

Dyskretna transformata Fouriera (DFT - ang. discrete Fourier Transform) to jedno z podstawowych narzędzi w numerycznym przetwarzaniu sygnałów. Celem ćwiczenia będzie zapoznanie się z jej własnościami.

DTFT

Dany jest sygnał dyskretny $x(n)=x(nT)$, gdzie T jest okresem próbkowania a n oznacza indeks próbki. Dla takiego sygnału, o ile ma skończoną energię, można zdefiniować transformatę Fouriera z dyskretnym czasem (DTFT - discrete time Fourier transform)

$$DTFT[x(n)] = X(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} x(n) \exp(-j\Omega n), \quad (1)$$

gdzie $\Omega=\omega T=2\pi f/f_s$ jest pulsacją znormalizowaną do częstotliwości próbkowania $f_s=1/T$. Tak więc DTFT sygnału można rozumieć jako funkcję zależną od kąta (Ω), albo też jako liczbę zespoloną o module równym 1 i argumentem Ω . DTFT ma następujące własności:

-jest funkcją ciągłego argumentu Ω ,

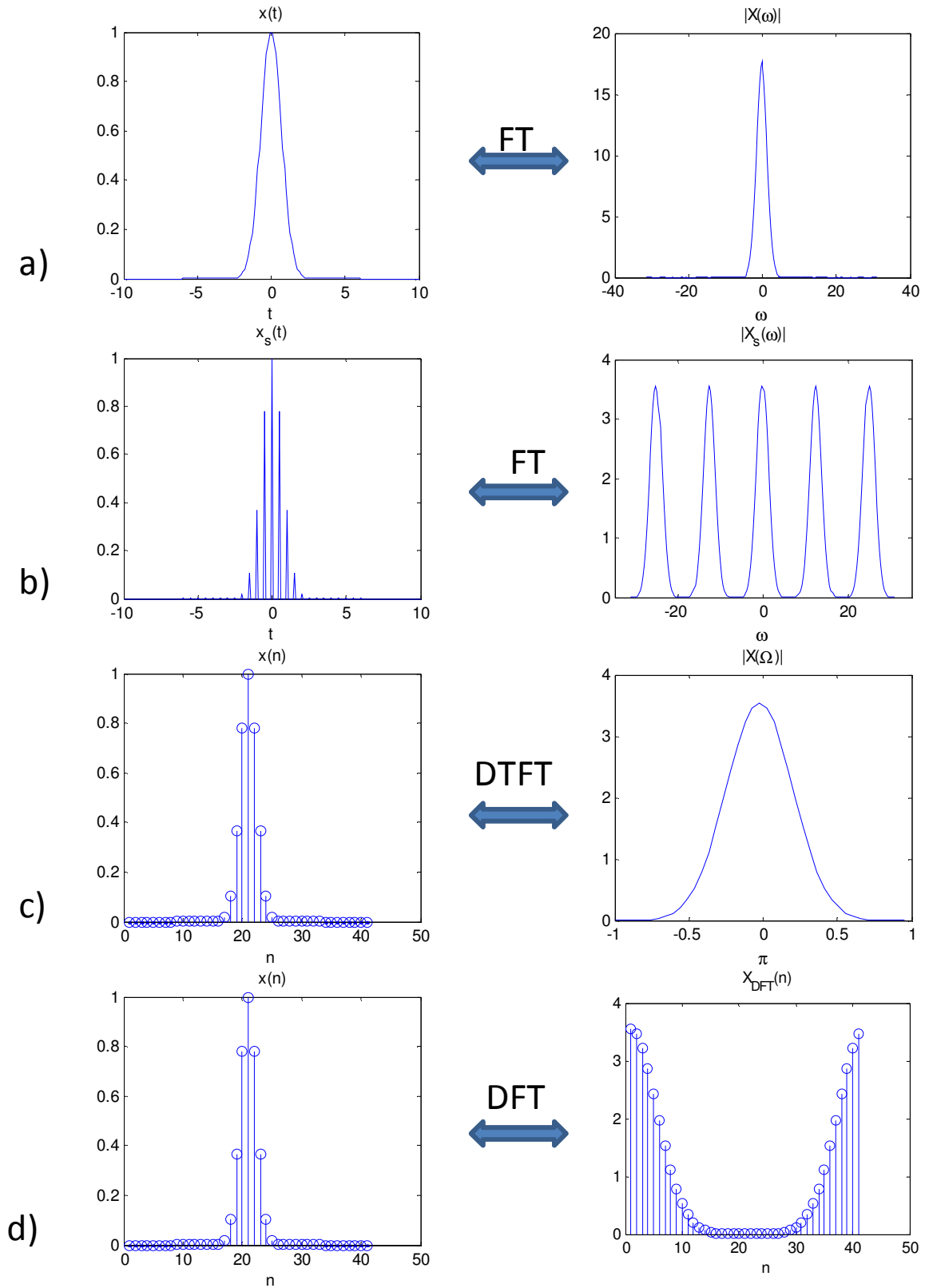
-jest funkcją okresową, o okresie 2π : $\exp(j(\Omega + 2\pi)n) = \exp(j\Omega n) \exp(jn2\pi) = \exp(j\Omega n)$. Ta własność ma inną interpretację, wynikającą z dualności transformaty Fouriera: wiadomo, że sygnał okresowy ma widmo dyskretne, a zatem sygnał dyskretny ma widmo okresowe.

-jeżeli sygnał jest rzeczywisty, to $X(\Omega) = X^*(2\pi - \Omega)$, a zatem pełną informację o widmie sygnału daje przedział $\Omega \in [0, \pi)$, gdzie π jest nazywana częstotliwością Nyquista.

Zależności pomiędzy transformatą Fouriera a DTFT pokazano na rys. 1. Na rys. 1 a) pokazano postać czasową przykładowego sygnału $x(t)$ oraz jego widmo $|X(\omega)|$. Jeżeli sygnał $x(t)$ zostanie spróbkowany z pulsacją ω_s , jego widmo $|X(\omega)|$ powtarza się periodycznie z okresem ω_s . Pokazano to na rys. 1 b). Ze względu na periodyczność cała informacja o sygnale zawarta jest w widmie z przedziału $[0, \omega_s)$. Wygodnie jest zatem stosować przekształcenie Fouriera w dziedzinę pulsacji znormalizowanej Ω , której wartość 2π odpowiada po prostu pulsacji próbkowania. To właśnie realizuje przekształcenie DTFT, które dla próbek $x(n)$ sygnału $x(t)$ pokazane zostało na rys. 1 c).

Własności DTFT są analogiczne do własności transformaty Fouriera. Szczegółowo zostały omówione np. w [1] (rozdz. 8.3.1). Są to: liniowość, twierdzenie o przesunięciu w czasie, twierdzenie o modulacji, twierdzenie o iloczynie, twierdzenie o splocie, twierdzenia o różniczkowaniu oraz całkowaniu, twierdzenie o skalowaniu.¹ Istotne jest również to, że dla sygnałów rzeczywistych widmo DTFT posiada symetrię Hermitowską, tzn. $X(\Omega) = X^*(2\pi - \Omega)$. Dla sygnałów zespolonych już takiej własności nie ma.

¹ Powtórzyć wymienione własności TF



Rys. 1. Zależności pomiędzy przekształceniami Fouriera tego samego sygnału. a) relacja pomiędzy sygnałem analogowym i jego analogowym widmem (transformata Fouriera FT), b) relacja pomiędzy sygnałem analogowym spróbkowanym i jego FT, c) pomiędzy próbkami sygnału i widmem ciągłym DTFT, d) pomiędzy próbkami sygnału i widmem dyskretnym DFT dla okna 41 próbek

DFT

W praktycznych zastosowaniach numerycznych nie mamy możliwości obliczenia sumy po nieskończonej liczbie próbek i konieczne jest wybranie skończonej liczby N próbek do obliczeń. Trzeba również określić pożądaną liczbę obliczanych próbek widma. Najczęściej przyjmuje się, że wynosi ona N , tyle samo co liczba próbek sygnału poddanych analizie. Wtedy widmo obliczane jest dla dyskretnej pulsacji o wartościach

$$\Omega_m = \frac{2\pi m}{N}.$$

Podstawiając powyższy wzór do równania (1) otrzymujemy wzór na tzw. dyskretne przekształcenie Fouriera (DFT).

$$X(n) = \sum_{k=0}^{N-1} x(k) \exp\left(-j2\pi \frac{nk}{N}\right)$$

Przekształcenie odwrotne do tego przekształcenia (IDFT) ma postać

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \exp\left(j2\pi \frac{nk}{N}\right)$$

Należy podkreślić, że oba przekształcenia działają na wektorach próbek i nie ma praktycznego znaczenia, który z wektorów odpowiada dziedzinie czasu, a który częstotliwości - to jest tylko interpretacja, która zależy od nas.

DFT, oprócz wymienionych wcześniej własności DTFT, ma następujące cechy:

- Moduł DFT sygnału $x(n)$ oraz $[x(n+k)]_N$ są takie same ($[\]_N$ oznacza przesunięcie cykliczne w ramach wektora o długości N). Zmienia się tylko faza (o czynnik $2\pi k/N$)
- Współczynniki DFT sygnału rzeczywistego posiadają symetrię Hermitowską, tzn. (dla parzystej liczby N)

$$[X(0), X(1), \dots, X(N-1)] = [X(0), X(1), \dots, X(N/2), X^*(N/2-1), \dots, X^*(1)]$$

Tak więc unikalne są wartości dla indeksu 0 oraz dla indeksu $N/2$, które odpowiadają odpowiednio składowej stałej oraz częstotliwości Nyquista². Tę zależność ilustruje rys. 2, na którym pokazano na płaszczyźnie zespolonej jak próbkowana jest pulsacja Ω w transformacie DFT. Widoczna jest unikalność próbek położonych na osi OX , oraz zespolone sprzężenie próbek n i $N-n$.

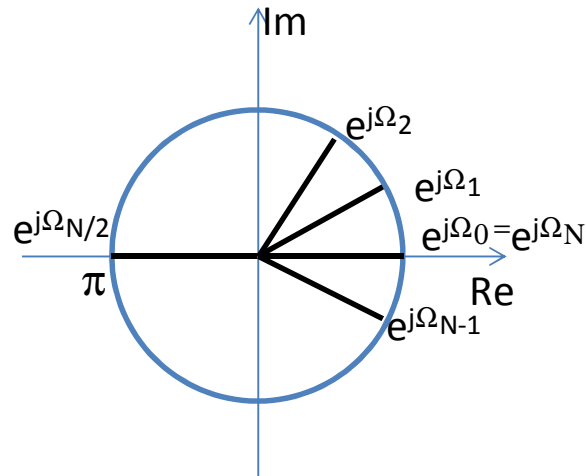
DFT dla sygnału $x(n)$ rozważanego w poprzednim punkcie zilustrowano na rys. 1 d). Zgodnie z wcześniejszymi rozważaniami, próbki $0:N/2-1$ stanowią składową stałą i "widmo dodatnie", zaś próbki od $N/2:N$ częstotliwość Nyquista i "widmo ujemne" (porównaj z rys. 1 a)).

Rozdzielczość DFT. Teoretyczna rozdzielczość DFT, czyli odległość pomiędzy sąsiednimi prążkami wynosi

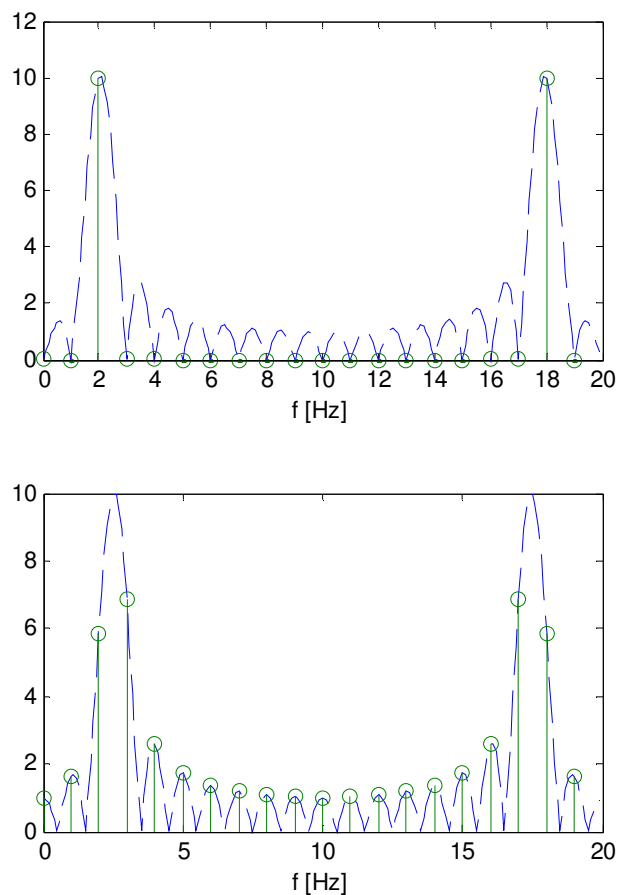
$$\Delta f = F_s/N$$

² ćwiczenie: Jakiej częstotliwości odpowiada k -ty indeks DFT?

i można ją poprawić (pomniejszyć) na wydłużając czas obserwacji (zwiększając N). Drugą metodą, która nie wymaga zwiększenia czasu obserwacji, jest uzupełnienie N próbek sygnału N_0 próbkami o wartości 0. Ta druga metoda daje jednak widmo będące splotem faktycznego DFT sygnału z widmem prostokąta wycinającego N niezerowych próbek z $N + N_0$ próbek.



Rys. 2. Odwzorowanie DFT o parzystej liczbie próbek względem argumentu DTFT Ω

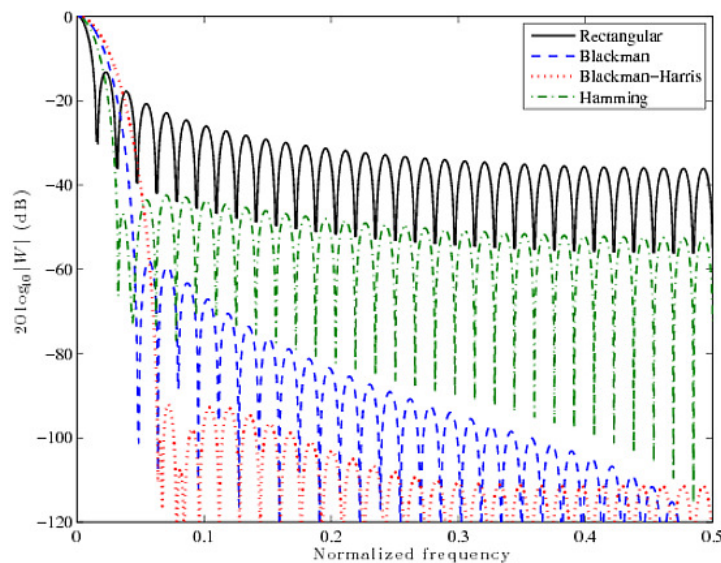


Rys. 3. Moduł DFT dla funkcji harmoniczej o częstotliwości 2 Hz oraz 2,5 Hz spróbkowanej z częstotliwością $f_s=20$ Hz na $N=20$ próbkach. Zielone kółka odpowiadają wartościom $DFT\{x(n)\}$, niebieska obwiednia to transformata Fouriera z $x(n)w(n)$

Wyciek. Zauważmy, że operacja wybrania N próbek z sygnału o nieskończonej liczbie próbek odpowiada przemnożeniu sygnału $x(n)$ przez okno prostokątne $w(n)=\text{rect}(n/(N))$. DFT oblicza zatem widmo splotu sygnału $x(n)$ z widmem okna. Najlepiej widać to dla sygnałów harmoniczných, które w dziedzinie częstotliwości reprezentowane są przez dwa prążki o przeciwnych częstotliwościach. Wypadkowe widmo jest zatem cyklicznym splotem widma prostokąta ($Sa(x)$) z 2 prążkami, co przekłada się w przybliżeniu na funkcję Sa przesuniętą w dziedzinie częstotliwości. Jeżeli liczba próbek DFT N jest wielokrotnością liczby próbek przypadającą na okres przebiegu harmonicznego, maksimum funkcji Sa przypada na częstotliwości odpowiadającej badanej funkcji harmonicznej, zera funkcji Sa przypadają zaś na pozostałych częstotliwościach DFT (rys. 3a). Zauważamy, że szerokość prążka głównego (między zerami) wynosi $2\pi/N$, zaś odległość pomiędzy kolejnymi zerami π/N . Jeżeli natomiast liczba próbek DFT nie jest wielokrotnością liczby okresów, wtedy maksimum i zera $Sa(x)$ wypadają pomiędzy częstotliwościami prążków DFT, co powoduje niezerowe wartości DFT na wszystkich częstotliwościach (rys. 3b). Zjawisko to nazywa się wyciekaniem. W praktyce rzadko mamy możliwość badania sygnałów o zgodnej liczbie okresów i próbek, stąd istnieje potrzeba zabezpieczania się przed wyciekaniem.

Okna DFT. Wyciek można marginalizować dwoma metodami - zwiększając liczbę próbek N badanego sygnału (co zawęży widmo Sa) ale wymaga dodatkowego czasu obserwacji sygnału. Druga metoda polega na stosowaniu okien innych niż prostokątne. W tym drugim przypadku próbki sygnału $x(n)$ przed operacją DFT poddaje się jeszcze wymnożeniu przez okno $w(n)$. Uzasadnienie takiej operacji jest następujące. Obliczone DFT jest splotem DFT sygnału z funkcją okna. Jeżeli próbki $x(n)$ bezpośrednio poddawane są DFT, odpowiada to tak naprawdę wymnożeniu ich przez okno prostokątne. Wiadomo, że okno prostokątne ma widmo typu $Sa(x)$. Ma zatem największy możliwy poziom listków bocznych, stąd też wynikają duże wartości DFT rejestrowane dla częstotliwości innych niż właściwa funkcja harmoniczna. Okna o mniejszych wartościach amplitudy dla próbek skrajnych i większych dla próbek centralnych, charakteryzują się niższym poziomem prążków bocznych, choć mają szerszy listek główny. Wyciek zmniejszany jest zatem kosztem rozdzielczości częstotliwościowej.

Rodzaje okien i ich własności omówiono np. w [1] w rozdz. 8.5.1. Wymianę rozdzielczości częstotliwościowej na poziom listków bocznych ilustruje rys. 4.



Rys. 4. Widmo okien DFT [www.dspstackexchange.com]

Tak więc jeżeli priorytetowa jest rozdzielczość częstotliwościowa, powinno się stosować okno prostokątne. Jednak aby wykryć składowe w widmie o niskich amplitudach, należy wybrać inny rodzaj okna.

Algorytm FFT. Złożoność algorytmu DFT określamy jako N^2 mnożeń zespolonych (mnożenie zespolone=4 mnożenia rzeczywiste). Wynika to z faktu, że dla N punktów z osobna musimy obliczyć sumę (2), a w każdej sumie jest N mnożeń:

$$X(m) = \sum_{k=0}^{N-1} x(k)w_N^{mk} \quad (2)$$

gdzie $w_N = \exp\left(-\frac{j2\pi}{N}\right)$ jest pierwiastkiem N -tego stopnia z 1 (który wystarczy obliczyć tylko raz i przechować w tablicach). Istnieje szereg algorytmów, które pozwalają obliczyć DFT szybciej. Klasycznym i najpopularniejszym przykładem takiego algorytmu, jest algorytm szybkiej transformaty Fouriera (FFT) radix2, który pozwala obliczać DFT dla $N=2^m$. Zmniejszenie nakładu obliczeniowego FFT bierze się z podziału ciągu $x(n)$ na dwa podciągi próbek nieparzystych i parzystych $x(n)$ o połowicznej liczbie próbek. Wzór (2) można wtedy zapisać

$$X(m) = \sum_{k=0}^{\frac{N}{2}-1} x(2k)w_N^{2mk} + \sum_{k=0}^{\frac{N}{2}-1} x(2k+1)w_N^{m(2k+1)}$$

Dalej, ponieważ $w_N^2 = w_{N/2}$ oraz wyciągając w_N^m przed drugą sumę możemy napisać

$$X(m) = \sum_{k=0}^{\frac{N}{2}-1} x(2k)w_{N/2}^{mk} + w_N^m \sum_{k=0}^{\frac{N}{2}-1} x(2k+1)w_{N/2}^{mk}$$

Zauważamy, że DFT dla próbki m kształtowane jest przez sumę DFT ciągu próbek parzystych oraz nieparzystych, przy czym to drugie DFT dodawane jest do pierwszego po wymnożeniu przez czynnik $\exp\left(-\frac{j2\pi m}{N}\right)$. Zauważamy ponadto, że w dziedzinie czasu czynnik ten jest odpowiedzialny za przesunięcie o jedną próbkę. Oznaczając $X_{2k}(m)$ i $X_{2k+1}(m)$ DFT dla próbek parzystych i nieparzystych, oraz korzystając z własności w_N^{mk} można otrzymać wzór na obliczenie FFT na każdym z etapów:

$$\begin{aligned} X(m) &= X_{2k}(m) + w_N^m X_{2k+1}(m) \\ X\left(\frac{N}{2} + m\right) &= X_{2k}(m) - w_N^m X_{2k+1}(m) \end{aligned} \quad (3)$$

dla $m=0, 1, \dots, N/2-1$

Otrzymujemy stąd przepis na obliczenie FFT:

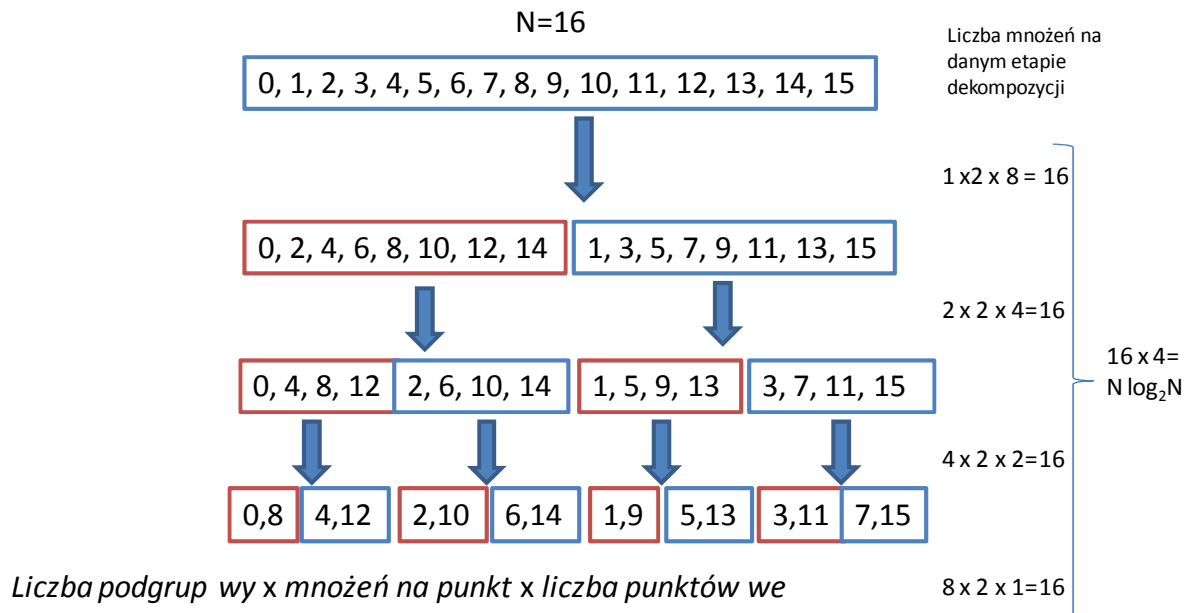
1. Podziel ciąg próbek $x(n)$ o N próbek na dwa podciągi o $N/2$ próbek, wybierając do nich co drugą próbkę

2a. Jeżeli $N>2$. Oblicz DFT każdego z 2 podciągów (idź do 1)

2b. Jeżeli $N=2$, to DFT wynosi $X(0)=x(0)+x(1)$, $X(1)=x(0)-x(1)$

3. Dodaj do siebie DFT parzystego i nieparzystego podciągu, ten ostatni wymnożony przez czynnik przesunięcia w czasie (z (+) dla m od 0 do N/2-1 i z (-) dla mod N/2 do N).

Złożoność obliczeniową FFT przedstawiono poglądowo na rys. 5 dla N=16. Wynika stąd, że liczba mnożeń zespolonych na każdym etapie dekompozycji wynosi N a etapów jest $\log_2 N$. Jednak przyglądając się (3) można zauważyć, że połowa mnożeń przez w_N^m się powtarza i stąd książkowy wzór $O(N/2 \log_2 N)$.



Rys. 5. Złożoność obliczeniowa algorytmu FFT. wy - blok złożony z 2 bloków we.

Literatura

[1] Tomasz P. Zieliński, Cyfrowe przetwarzanie sygnałów od teorii do zastosowań, WKiŁ, Warszawa 2005, e-book dostępny dla studentów PW w bibliotece cyfrowej

Zadania przygotowujące do wykonania w Matlab/Octave przed ćwiczeniem

Do wykonania ćwiczenia niezbędna będzie podstawowa wiedza z zakresu języka Matlab. Język Matlab jest prostym językiem skryptowym wysokiego poziomu, przeznaczonym do obliczeń numerycznych. Środowisko Matlab posiada wiele funkcji użytecznych do (cyfrowego) przetwarzania sygnałów. Część to funkcje wbudowane (np. fft), inne należą do pakietu Signal Processing Toolbox.

W Matlabie można zarówno pisać skrypty (nazwapliku.m), definiować funkcje (zapisując w osobnym pliku funkcja.m),

```
% zawartość pliku funkcja.m
function y=funkcja(x)
y=x
```

jak też, co niezwykle użyteczne i pouczające, pisać komendy bezpośrednio w command line. Można też najpierw uruchomić skrypt deklarujący zmienne

```
% skrypt skrl.m
a=1;
b=2;
```

LCPS - Własności dyskretnej transformaty Fouriera

a następnie wykonać odpowiednią operację w command line

```
>> c=a+b
```

```
c =
```

```
3
```

(brak średnika po komendzie wypływa wynik w command line).

W Matlabie/Octave, gdy tylko możliwe, posługujemy się wektorami, czyli tablicami jednowymiarowymi wartości lub macierzami, czyli tablicami $N \times M$. Silnik obliczeniowy Matlab'a zoptymalizowany jest pod obliczenia macierzowe. Przykładowo, pisząc skrypt:

```
a=1:2:20000;  
b=ones(1,10000);  
D=rand(10000,10000); %macierz o elementach losowych z przedziału 0-1  
%tic/toc zegar start/end
```

```
%operacja 1 - mnozenie element po elemencie  
tic  
c=a.*b;  
toc
```

```
%operacja 2 - mnozenie macierzowe  
tic  
d=D*b';  
toc
```

```
%operacja 1 w petli  
tic  
for k=1:10000  
    c(k)=a(k)*b(k);  
end
```

```
toc
```

```
%operacja 2 w petli  
tic  
for k=1:10000  
    for kk=1:10000  
        d(k)=D(k, kk)*b(kk);  
    end  
end  
toc
```

otrzymujemy wynik czasu poszczególnych operacji

Elapsed time is 0.000059 seconds.

Elapsed time is 0.170478 seconds.

Elapsed time is 0.000076 seconds.

Elapsed time is 1.529724 seconds.

Ze względu na popularność operacji macierzowo-wektorowych, w Matlabie zdefiniowano szereg operatorów algebraicznych dla macierzy (wektorów)

+, -, *, /, ^ które służą do dodawania, odejmowania, mnożenia, dzielenia, potęgowania macierzy, a także odpowiadające im operacje z kropką, np. .*, ./, .^, które wykonują dane działanie na każdym z elementów wektora/macierzy osobno.

W Matlabie używa się często notacji a:b:c do wyboru indeksów wektora, czy deklaracji wektora liczb całkowitych. Oznacza ona wektor złożony z liczb od a do b do c, czyli a, a+b, a+2b, ..., <c

Operator ' oznacza transpozycję (lub sprzężenie Hermitowskie dla macierzy zespolonych).

W celu sprawnego zaliczenia ćwiczenia LCPS 2 wykonaj wcześniej poniższe zadania w programie Matlab.

Zadanie 1: wektory i macierze

Zadeklaruj 10 elementowy wektor złożony z samych jedynek oraz 10 elementowy wektor liczb całkowitych rosnących od 0 do 9.

- a) Oblicz ich iloczyn elementowy
- b) Oblicz sumę i różnicę wektorów
- c) Oblicz ich iloczyn skalarny (posługuj się tylko * i ')
- d) Oblicz iloczyn Kroneckera AB tych wektorów (j.w.)
- e) Oblicz potęgę 2 wektora nr 2
- f) Oblicz pierwiastek wektora 2
- g) Utwórz nowy wektor z parzystych próbek wektora 2 oraz z nieparzystych próbek (notacja a:b:c)
- h) Oblicz produkt Kroneckera BA tych wektorów, następnie oblicz iloczyn macierzowy i elementowy produktów Kroneckera
- i) Zadeklaruj wektor zer $N \times 1$; zadeklaruj macierz jednostkową $M \times M$;

Zadanie 2: wykresy

Zadeklaruj dziedzinę 1000 próbek czasu t z częstotliwością próbkowania 100 Hz. Zadeklaruj wektor x_1 jako $\sin(2\pi \cdot 10 \cdot t)$, x_2 jako $\cos(2\pi \cdot 20 \cdot t)$. Narysuj wykres obu wektorów we właściwej skali czasu i na jednym wykresie (`plot(t,x1,t,x2)`). Po wyświetleniu wykresu wejdź w opcje: axes properties i pobaw się rodzajami krzywych, podpisami na osiach, itp.

Zadanie 3: podstawowe sygnały

Zdefiniuj dziedzinę czasu (jak w ćw. 2) Następnie wygeneruj (i wyplotuj) następujące sygnały:

- a) $x_1 = 2\cos(2\pi \cdot 5 \cdot t)$
- b) $x_2 = \text{rect}((t-1)/2) + 2\text{rect}((t+3)/4)$ % użyj `rectpuls(t)` lub `rpulse`
- c) $x_3 = \text{Sa}(2\pi \cdot (t-3))$ % $\text{Sa}(x) = \sin(x)/x$; Matlab ma funkcje `sinc(x) = sin(pi*x)/(pi*x)`

d) $x_4 = \exp(-t.^2/a)$ % impuls gaussowski o wybranej szerokości

e) $x_5 = x_1 * x_2$

f) $\text{suma}_{(n=-10)^{10}} \text{rect}(t-nT)$ % okresowo powtarzające się prostokąty (z okresem T)

Zadanie 4: podstawowe DFT

Oblicz DFT sygnałów z ćwiczenia 3. Wyplotuj je tak, aby na osi X była a) częstotliwość b) pulsacja c) pulsacja znormalizowana. Czy otrzymane widma są zgodne z intuicją?

Inne uwagi:

- Matlab co do zasady operuje na liczbach zmiennoprzecinkowych
 - Każda funkcja posiada commandlinowy help (\gg help funkcja) i obszerną dokumentację
 - Zdefiniowane stałe: pi, j, i, lj, li
 - Wektor można zdefiniować podając współczynniki w nawiasie [1 2 3] -wektor 1xN, [1 2 3]' - wektor Nx1, [1;2;3] wektor Nx1, [1:5;5:-1:1] 2x5
 - zeros(M,N); ones(M,N) -macierz 0/1 o wymiarach MxN
 - eye(N) - macierz jednostkowa o wymiarze N
 - real(x), imag(x) - część rzeczywista/urojona x
 - circshift(1:10,[0 2]) - cykliczne przesunięcie wektora o 2
- ans = 9 10 1 2 3 4 5 6 7 8