

Przetwarzanie sygnałów

Ćwiczenie 2

Podstawowe operacje na sygnałach cyfrowych

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

Spis treści

1	Wstęp	1
1.1	Próbki sygnałów dyskretnych	1
1.2	Wybrane sygnały impulsowe	2
1.3	Splot	2
2	Pytania i zadania na kartkówkę	3
3	Zadania	3
3.1	Funkcje narzędziowe	4
3.2	Operacje arytmetyczne na sygnałach i parametry sygnałów	6
3.3	Splot	7



Katedra
Nanometrologii

Wydział Elektroniki, Fotoniki i Mikrosystemów
Politechnika Wrocławska

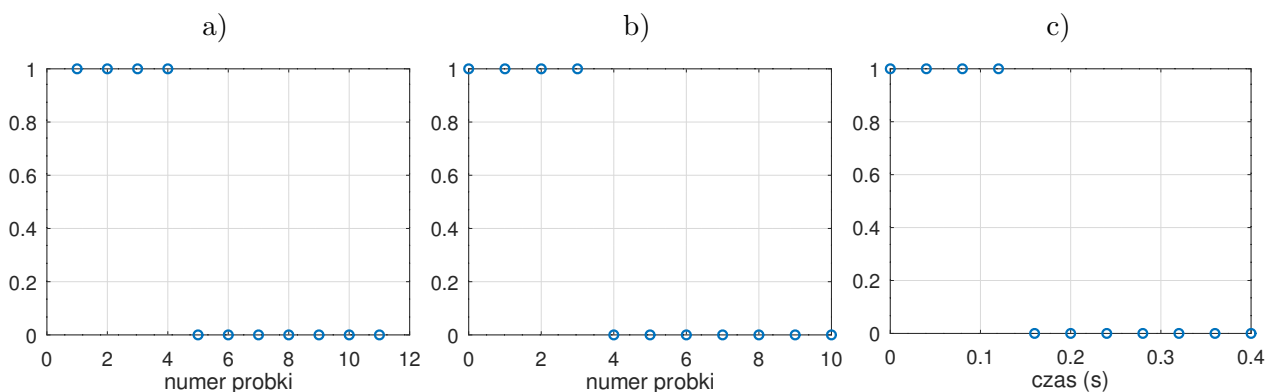
1 Wstęp

1.1 Próbki sygnałów dyskretnych

Sygnał dyskretny zasadniczo przedstawiamy w postaci ciągu wartości, które można zapisać między innymi w jednowymiarowej tablicy. Na przykład 11 kolejnych próbek impulsu prostokątnego, którego czas trwania to 4 próbki, można zapisać w tablicy x używając następującej instrukcji Octave:

```
x = [1 1 1 1 0 0 0 0 0 0 0];
```

Chcąc wykreślić przebieg tego wykresu w najprostszy sposób za pomocą instrukcji¹ `plot(x, "o")`; otrzyma się wykres nie do końca zgodny z oczekiwaniami (rysunek 1a). Próbką początkowa wykreślona zostanie na współrzędnej poziomej 1 zamiast 0. Wynika to ze sposobu indeksowania tablic w Octave.



Rysunek 1: Przykłady wykresów ilustrujących przebieg sygnału dyskretnego: a) uzyskany wprost z tabeli wartości z niewłaściwymi współrzędnymi poziomymi, b) z numerami próbek na osi poziomej, c) z czasem na osi poziomej.

Chcąc skorygować tę niedogodność można użyć zakresów. Aby numery próbek zgadzały się z przyjętym w przetwarzaniu sygnałów zwyczajem, że początkowa próbka ma indeks 0, należy współrzędne poziome punktów określić jako liczby od 0 do 10.² Można to załatwić podając jako pierwsze dwa argumenty funkcji `plot` dwie tablice: jedną z wartościami współrzędnych poziomych (odciętych) a drugą na współrzędne pionowe (rzędne) punktów. Ta pierwsza może być podana w postaci zakresu `0:10` lub w postaci ogólnej:

```
plot(0:length(x)-1, x, "o");
```

gdzie pierwszy argument funkcji `plot` to zakres od 0 do liczby określonej długością tablicy x (w naszym wypadku jest to 11) pomniejszona o 1. Uzyska się w ten sposób wykres z rysunku 1b.

Dodatkowo, często chcemy wykreślić punkty tworzące sygnał dyskretny na wykresie, na którym pozioma oś to nie numery próbek a czas wyrażony w jednostkach czasu (np. sekundach), z którego próbki te pochodziły. Przyjmijmy, że częstotliwość próbkowania $f_s = 25$ Hz. Oznacza to, że próbka o numerze 0 pochodzi z czasu 0, próbka o numerze 1 pochodzi z czasu równego $\frac{1}{25}$ s, próbka 2 z czasu równego $\frac{2}{25}$ s, itd. Nieznacznie zatem modyfikując wyrażenie na wartości odciętych³ można wykreślić te same dane ale w funkcji czasu, jak na rysunku 1c.

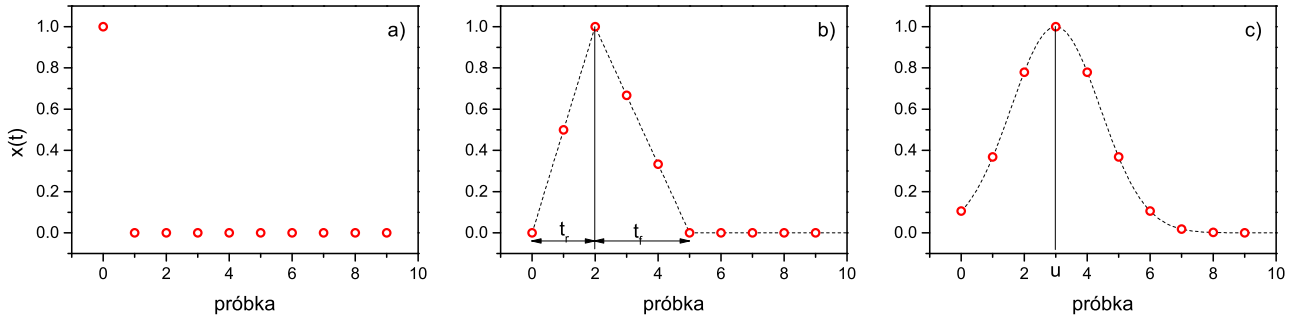
¹Drugi parametr: "o" powoduje, że wykres składa się tylko z okrągłych i pustych w środku punktów. Istnieje wiele sposobów na formatowanie wykresów, na przykład "-o" da wykres połączonych liniami punktów, itd. Więcej szczegółów znajdziesz w pomocy do funkcji `plot`.

²Zwróć uwagę, że 11 próbek sygnału ma indeksy od 0 do 10 a nie do 11, to często popełniany błąd!

³Pozostawiamy to do wymyślenia Tobie.

1.2 Wybrane sygnały impulsowe

W cyfrowym przetwarzaniu sygnałów używa się pewnych typowych sygnałów, jak sygnał harmoniczny, stały itp. Niektóre z typowych sygnałów impulsowych przedstawiono na rysunku 2.



Rysunek 2: Wybrane typowe sygnały impulsowe: a) impuls Kroeneckera, b) impuls trójkątny $t_r = 2$, $t_f = 3$, c) impuls Gaussa $u = 3$, $s = 1$.

W impulsie Kroeneckera początkowa próbka sygnału równa jest 1, pozostałe próbki to 0.

Impuls trójkątny to sygnał, w którym wartości próbek najpierw liniowo narastają od 0 do wartości maksymalnej w czasie narostu t_r , a następnie opadają do 0 w czasie opadania t_f .

Impuls Gaussa wyrażony jest krzywą Gaussa, którą można przedstawić w dziedzinie czasu t lub numeru próbki n jako:

$$x(t) = e^{-\frac{(t-u)^2}{2s^2}} \quad (1)$$

$$x[n] = e^{-\frac{(n-u)^2}{2s^2}}, \quad (2)$$

gdzie u to położenie środka impulsu (jego czas bądź numer próbki) a s jest parametrem wyrażającym szerokość impulsu.

1.3 Splot

Splotem sygnałów nazywamy działanie, którego argumentami są dwa sygnały i które zdefiniowane jest dla sygnałów ciągłych następującym wzorem:

$$s(t) = x_1(t) * x_2(t) = \int_{-\infty}^{\infty} x_1(u)x_2(t-u)du \quad (3)$$

Dla sygnałów dyskretnych x i y operację wyznaczania n -tej próbki splotu zapisać można następująco:

$$z[n] = x[n] * y[n] = \sum_{k=0}^{N_y-1} y[k]x[n-k] \quad (4)$$

Długość splotu dwóch sygnałów x , y wynosi $N_x + N_y - 1$, gdzie N_x jest długością sygnału x a N_y długością sygnału y . Obliczenie splotu wymaga zatem zsumowania iloczynów odpowiednich próbek splatanych sygnałów. To, które mają być mnożone wynika z indeksów sygnałów w wyrażeniu sumy. Przy obliczaniu splotu należy zwrócić uwagę na to, że kiedy indeksy sygnałów x i y wychodzą poza dozwolony zakres, to wartości sygnału zastępuje się zerami. Ilustrację tego procesu przedstawiono w Tabeli 1.

Tabela 1: Ilustracja obliczenia elementu $n = 0$, $n = 2$ i $n = 5$ splotu sygnałów x i y o długościach, odpowiednio, $N_x = 3$, $N_y = 5$. Tabela zawiera wartości indeksów sygnałów k i $n - k$, symbole próbek sygnałów x i y o tych indeksach (lub 0 jeśli brak danej próbki) oraz niezerowe iloczyny $y[k]x[n - k]$, których suma da n -tą wartość splotu $x * y$.

		$n = 0$								
k		-3	-2	-1	0	1	2	3	4	5
$y[k]$		0	0	0	y_0	y_1	y_2	y_3	y_4	0
$n - k$		3	2	1	0	-1	-2	-3	-4	-5
$x[n - k]$		0	x_2	x_1	x_0	0	0	0	0	0
$y[k]x[n - k]$		0	0	0	y_0x_0	0	0	0	0	0
		$n = 2$								
k		-3	-2	-1	0	1	2	3	4	5
$y[k]$		0	0	0	y_0	y_1	y_2	y_3	y_4	0
$n - k$		5	4	3	2	1	0	-1	-2	-3
$x[n - k]$		0	0	0	x_2	x_1	x_0	0	0	0
$y[k]x[n - k]$		0	0	0	y_0x_2	y_1x_1	y_2x_0	0	0	0
		$n = 5$								
k		-3	-2	-1	0	1	2	3	4	5
$y[k]$		0	0	0	y_0	y_1	y_2	y_3	y_4	0
$n - k$		8	7	6	5	4	3	2	1	0
$x[n - k]$		0	0	0	0	0	0	x_2	x_1	x_0
$y[k]x[n - k]$		0	0	0	0	0	0	y_3x_2	y_4x_1	0

2 Pytania i zadania na kartkówkę

1. Ile próbek sygnału dyskretnego uzyska się próbkując z częstotliwością 3 kHz przez 300 ms?
2. Jaką częstotliwość próbkowania trzeba zastosować, aby w ciągu 7 sekund zebrać 56000 próbek?
3. Ile czasu upłynie od rozpoczęcia próbkowania do zebrania 16500 próbek, gdy częstotliwość próbkowania to 44100 Hz?
4. Dane są tablice: $\mathbf{a} = [2 \ 3 \ 8 \ 1 \ 3 \ 6 \ 5 \ 1 \ 2 \ 3 \ 7]$ i $\mathbf{b} = [8 \ 1 \ 4 \ 2 \ 9 \ 7 \ 1 \ 2 \ 3 \ 4 \ 2]$. Oblicz ręcznie wartość $d = \sum_{i=0}^4 a_{i+5} \cdot b_{8-i}$ przyjmując, że indeksy początkowych elementów w tablicach to a_0 i b_0 .
5. Jak będzie wyglądał fragment skryptu Octave, który zrealizuje działanie na elementach jednowymiarowych tablic \mathbf{a} i \mathbf{b} dokładnie według wzoru: $d = \sum_{i=0}^4 a_{i+5} \cdot b_{8-i}$. Zmienne w skrypcie mają mieć takie same nazwy jak we wzorze, a początkowe elementy w tablicach to a_0 i b_0 .
6. Jaką długość będzie miał splot sygnałów o długości 10 i 30 próbek?
7. Jaką wartość będzie miała 4. (licząc od 0) próbka splotu sygnałów $x * y$, jeżeli $\mathbf{x} = [1 \ 5 \ 4 \ 3 \ 2 \ 3 \ 4 \ 5 \ 6]$ i $\mathbf{y} = [1 \ 3 \ 0 \ 1]$?

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

3 Zadania

W trakcie zajęć sprawdzana będzie znajomość implementacji, prawidłowość działania i umiejętność korzystania z funkcji opisanych w poniższych zadaniach. Obowiązkiem jest przygotowanie ich przed przyjsciem na zajęcia.

Każde z zadań 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 liczbę punktów pod warunkiem, że zostanie ono w całości poprawnie zrealizowane.

Wszystkie funkcje, których argumentem bądź wynikiem jest sygnał w postaci tablicy, powinny w pierwszym jej elemencie (czyli elemencie o indeksie 1) zawierać pierwszy element sygnału (dla $t = 0$).

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

3.1 Funkcje narzędziowe

W zadaniu pierwszym należy poprawić treść kilku funkcji narzędziowych, w których kodzie znajdują się błędy. Funkcje te i ich prawidłowe działanie potrzebne będą w kolejnych ćwiczeniach. Funkcje to:

- `gen_time`, stworzona na podstawie informacji z punktu 1.1, która na podstawie podanej liczby próbek `N` i częstotliwości próbkowania `fs` generuje jednowymiarową tablicę zawierającą czasy kolejnych próbek, przy czym czas pierwszej z nich powinien wynosić 0. Wartości tej tablicy mają służyć jako wartości odciętych `N` próbek sygnału na ich wykresach w funkcji czasu.
- `gen_sin`, która na podstawie wygenerowanej tabeli z czasami próbek `time` generuje tabelę z przebiegiem sinusoidalnym o częstotliwości `fsin`, amplitudzie `A` i przesunięciu fazowym `fi`.
- `gen_delta`, która na podstawie wygenerowanej tabeli z czasami próbek `time` generuje deltę Kroeneckera.
- `gen_triangle`, która na podstawie wygenerowanej tabeli z czasami próbek `time` generuje impuls trójkątny o amplitudzie `A`, czasie narostu `tr` i czasie opadania `tf`.
- `gen_gauss`, która na podstawie wygenerowanej tabeli z czasami próbek `time` generuje impuls Gaussa, którego środek przypada na czas `u` a szerokość opisano parametrem `s`.
- `sig_delay_N`, która opóźnia sygnał `x` o `Nd` próbek.

Proszę pamiętać o tym, że w jednym pliku ze skryptem może znajdować się jedna funkcja, oraz że nazwa pliku musi dokładnie odpowiadać nazwie funkcji (n.p. implementacja funkcji `gen_time` musi być zapisana w pliku `gen_time.m`).

```
function time = gen_time(N, fs)
    time = zeros(1, N);
    for i=1:N
        time(i) = i/fs;
    endfor
endfunction
```

```
function signal=gen_sin(time, fsin, A, fi)
    signal = A*sin(time + fsin + fi);
endfunction
```

```
function signal = gen_delta(time)
    N = length(time);
    signal = zeros(N);
    signal(0) = 1;
endfunction
```

```

function signal = gen_triangle (time, A, tr, tf)
    N = length(time);
    signal = zeros(1, N);
    for n = 1:N
        if time(n) < tr # jezeli trwa narost
            if (tr<0)
                signal(n) = A*time(n)/tr;
            endif;
        elseif time(n) <= tr+tf # jezeli trwa spadek
            if (tf>0)
                signal(n) = A - A*(time(n)-tr)/tf;
            else
                signal(n) = A;
            endif
        endif
    endfor
endfunction

```

```

function signal=gen_gauss(time, u, s)
    N = length(time);
    signal = zeros(1,N);
    for n=1:N-1
        signal(n) = exp(-(time(n)-u)^2/2*s^2);
    endfor
endfunction

```

```

function y = sig_delay_N(x, Nd)
    N = length(x);
    y = zeros(1, N);
    for i=0:N-Nd-1
        y(i+Nd) = x(i);
    endfor
endfunction

```

Prawidłowe działanie funkcji należy zilustrować za pomocą wykresów generowanych przebiegów w funkcji czasu. Za ich wykonanie otrzymuje się 2 pkt.

Wskazówki do testowania funkcji narzędziowych

Funkcje najwygodniej jest testować za pomocą osobnego skryptu zapisanego w pliku. Jego nazwa nie ma znaczenia. Na przykład zawartość skryptu testującego funkcje z zadania 1. może wyglądać następująco (skrypt jest niedokończony, w miejsce wielokropków należy wpisać parametry generowanych sygnałów czy operacji na sygnałach):

```

time = gen_time(...); # generacja tablicy z czasami próbek
x1 = gen_sin(time, ...); # generacja przebiegu sinusoidalnego
x2 = gen_triangle(time, ...); # generacja impulsu trójkątnego
x3 = gen_delta(time); # generacja delty Kroeneckera
x3d = sig_delay_N(x3, ...); # opóźniona delta Kroeneckera

```

```

#wykreslenie wygenerowanych sygnalow
figure(1);

```

```

plot(time, x1, time, x2); # sinus i trójkąt na rysunku 1
figure(2);
plot(time, x3, time, x3d); # ilustracja przesunięcia delty Kroeneckera

```

Generując sygnały dostosuj czasy trwania bądź częstotliwości do częstotliwości próbkowania. Aby przebieg sygnału dyskretnego w funkcji czasu był czytelny kryterium Shannona - Nyquista musi być spełnione z dużym nadmiarem.

Odsłuchiwanie sygnałów okresowych

Dodatkowym testem, mającym też charakter zabawy, jest odsłuchanie sygnałów tworzonych w Octave przez głośniki komputera. Aby móc to zrobić należy generować sygnały o częstotliwościach próbkowania idących w dziesiątki kHz, na przykład powszechną w cyfrowym audio częstotliwość $f_s = 44.1$ kHz. Dźwięk taki musi trwać sekundę czy kilka sekund, musi zatem zawierać odpowiednią do tego liczbę próbek.

Zakładając, że w tablicy `x` zapisano wartości z przedziału od -1 do 1 odpowiadające próbkom dźwięku (na przykład utworzonych funkcją `gen_sin(...)`), a zmienna `fs` równa jest wyrażonej w hercach częstotliwości próbkowania, można ten dźwięk odtworzyć za pomocą następujących funkcji Octave:

```

#utworzenie odtwarzacza audio probek sygnału x z czestotliwoscia probkowania fs
player = audioplayer(x, fs);
#uruchomienie odtwarzacza
playblocking (player);

```

Uwaga, dźwięk wykorzystujący cały przedział wartości od -1 do 1 może być bardzo głośny. Ostrożnie zatem z ustawieniami głośności w komputerze, zwłaszcza przy korzystaniu ze słuchawek!

Metodę tę można użyć do np. odsłuchiwania dźwięków o różnych częstotliwościach, na przykład odpowiadających poszczególnym nutom, oraz ich sum składających je w akordy. Więcej informacji w Wikipedii: [https://pl.wikipedia.org/wiki/D%C5%BAwi%C4%99k_\(muzyka\)](https://pl.wikipedia.org/wiki/D%C5%BAwi%C4%99k_(muzyka)), <https://pl.wikipedia.org/wiki/Tr%C3%B3jd%C5%BAwi%C4%99k>.

3.2 Operacje arytmetyczne na sygnałach i parametry sygnałów

Wygeneruj tablicę `time` zawierającą czasy próbek w.g. parametrów podanych przez prowadzącego. Używając funkcji z poprzedniego punktu wygeneruj kilka różnych sygnałów. Zaprezentuj wskazane przez prowadzącego działania na sygnałach, takie jak na przykład:

- dodawanie sygnałów,
- mnożenie sygnałów,
- dodawanie składowej stałej do sygnału,
- zmianę amplitudy sygnału przez jego pomnożenie przez stałą.
- przesuwanie sygnału w czasie o wskazaną liczbę próbek, sumowanie sygnałów przesuniętych o różne opóźnienia,
- obliczenie ich wskazanych parametrów, jak na przykład: wartość średnią, wartość skuteczną czy inne.

Prawidłowe wykonanie operacji arytmetycznych należy zilustrować za pomocą wykresów. Najwygodniej stworzyć skrypt, który będzie te operacje arytmetyczne wykonywał oraz wyświetlał wykresy. Wykonanie zadania otrzymuje się 1 pkt.

3.3 Splot

Usuń błędy z funkcji `sig_conv` obliczającej wartość splotu dwóch sygnałów `x` i `y`. Działanie zilustruj wykonując operację splotu na dwóch sygnałach, których splot jest łatwy do obliczenia bez użycia komputera. Co to za sygnał? Przypomnij sobie informacje o splocie przedstawione na wykładzie.

Wyniki przedstaw na wykresach. Zwróć uwagę na wartości wyniku oraz na długość sygnału wynikowego. Za prawidłowe wykonanie zadania otrzymasz 2 pkt.

```
function z = sig_conv(x,y)
    Nx = length(x);
    Ny = length(y);
    z = zeros(1,Nx+Ny); #utworzenie pustej tablicy na wynik splotu
    for n=0:Nx+Ny-1 # dla każdego elementu tablicy wynikowej
        c=0; # zsumuj odpowiednie iloczyny probek sygnalow
        for k=0:Ny-1
            if ((n-k)<=0) && ((n-k)<Nx) # jesli pozwalaja na to indeksy probek
                c = c + x(n-k)*y(k); # dodaj do c odpowiedni iloczyn
            endif
        endfor;
        z(n+1) = c; # zapis wyniku w tablicy
    endfor;
endfunction
```