

POLITECHNIKA ŚWIĘTOKRZYSKA
Wydział Elektrotechniki, Automatyki i Informatyki
Katedra Elektrotechniki Przemysłowej i Automatyki
Zakład Urządzeń i Systemów Automatyki

Cyfrowe Przetwarzanie Sygnałów

Generowanie i wizualizacja sygnałów cyfrowych

Instrukcja laboratoryjna

Paweł Strączyński
Robert Kazała

2019

1 Generowanie wykresów funkcji jednowymiarowych

Na listingu 1.1 przedstawiono przykład generowania wykresów funkcji jednowymiarowych.

Listing 1.1: Generowanie wykresu funkcji jednowymiarowej

```
1 from pylab import *
2 tp=0.1
3 fs=1000
4 t=arange(0,tp,1/float(fs))
5 y1=sin(2*pi*10*t)
6 y2=sin(2*pi*120*t)
7 y=y1+y2
8 clf
9 subplot(2,1,1)
10 plot(t,y1,t,y2)
11 subplot(2,1,2)
12 plot(t,y)
13 show()
```

Wygenerowany sygnał składa się z dwóch składowych sinusoidalnych o amplitudach równych 1, częstotliwościach 10 i 120 oraz zerowym przesunięciu fazowym. Zgodnie z ogólną zależnością:

$$y = A \sin(\omega t + \varphi) \quad (1)$$

gdzie:

A - amplituda sygnału,

ω - pulsacja sygnału,

φ - przesunięcie fazowe.

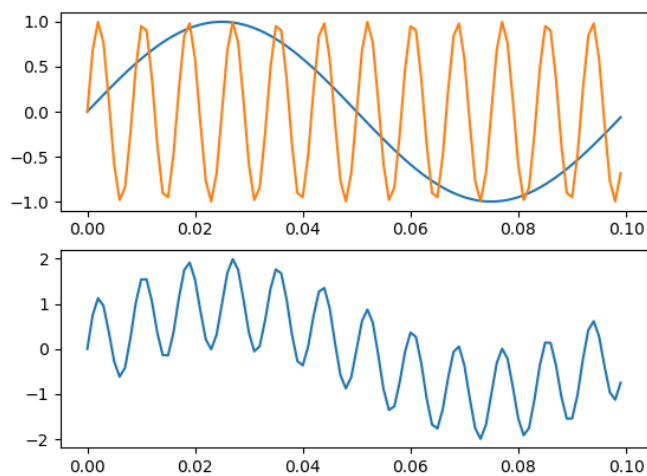
Rysunek 1.1 przedstawia wykres powstały w wyniku działania skryptu z listingu 1.1. Wykres przedstawia funkcje składowe sygnału - $\sin(2\pi \cdot 10 \cdot t)$, $\sin(2\pi \cdot 120 \cdot t)$ oraz sygnał powstały w wyniku sumowania funkcji składowych.

Listing 1.2: Generowanie wykresu przebiegu prostokątnego o zadanym wypełnieniu

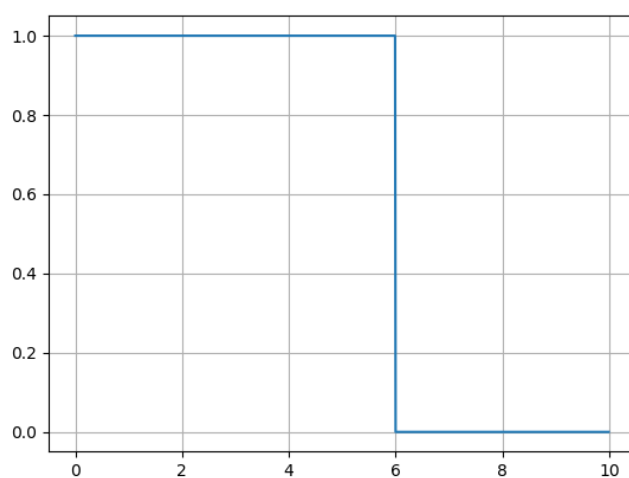
```
1 from pylab import *
2
3 def PWM(duty_cycle,freq,time):
4     t=arange(0,time,1/float(freq))
5     return t,t%time*1/time<duty_cycle
6
7 t_,y_ =PWM(0.6,100,10)
```

```
8 plot(t_, y_)
9 grid(True)
10 show()
```

Na listingu 1.2 przedstawiono przykład funkcji generującej sygnał prostokątny o zadanym wypełnieniu.



Rysunek 1.1: Wykres funkcji jednowymiarowej



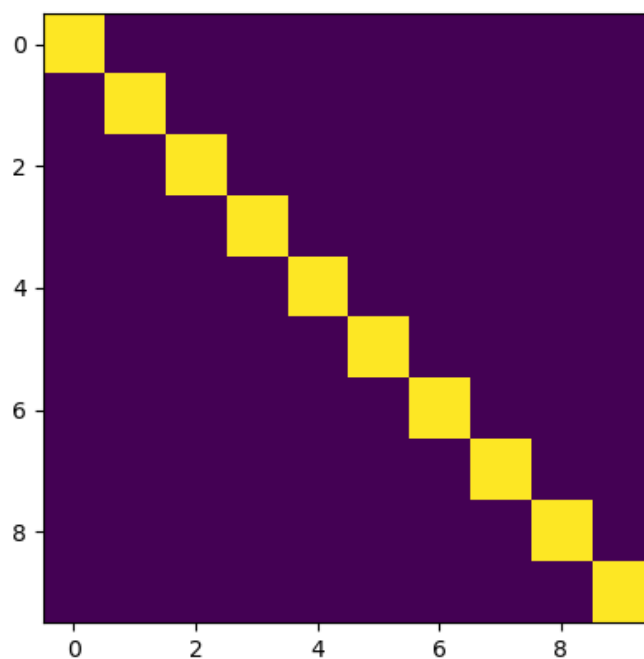
Rysunek 1.2: Wykres funkcji prostokątnej o zadanym wypełnieniu

2 Indeksowanie tablic

W języku Python istnieje kilka rodzajów indeksowania tablic. Na listingach 2.3 oraz 2.4 przedstawiono sposób z wykorzystaniem funkcji `range()` oraz `xrange()`.

Listing 2.3: Indeksowanie tablic z wykorzystaniem funkcji `range`

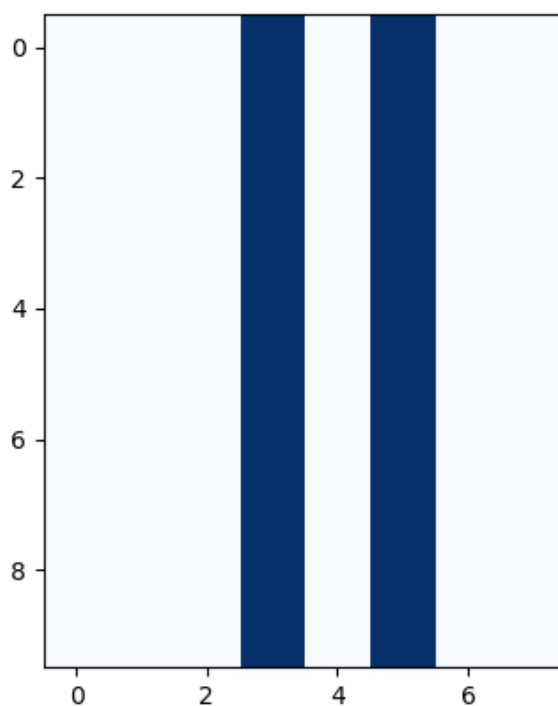
```
1 from pylab import *
2
3 mat1 = eye(10)
4 mat2 = zeros(mat1.shape)
5
6 for r in range(mat1.shape[0]):
7     for c in range(mat1.shape[1]):
8         print r,c
9         mat2[r,c]=mat1[r,c]
10
11 imshow(mat2)
12 show()
```



Rysunek 2.1: Indeksowanie tablic z wykorzystaniem funkcji `range`

Listing 2.4: Indeksowanie tablic z wykorzystaniem funkcji xrange

```
1 from pylab import *
2 A=zeros((10,8))
3 A[:,(3,5)]=1
4 B=zeros(A.shape)
5 for r in xrange(A.shape[0]):
6     for c in xrange(A.shape[1]):
7         print r,c
8         B[r,c]=A[r,c]
9
10 imshow(B,interpolation='none', cmap='Blues')
11 show()
```



Rysunek 2.2: Indeksowanie tablic z wykorzystaniem funkcji xrange

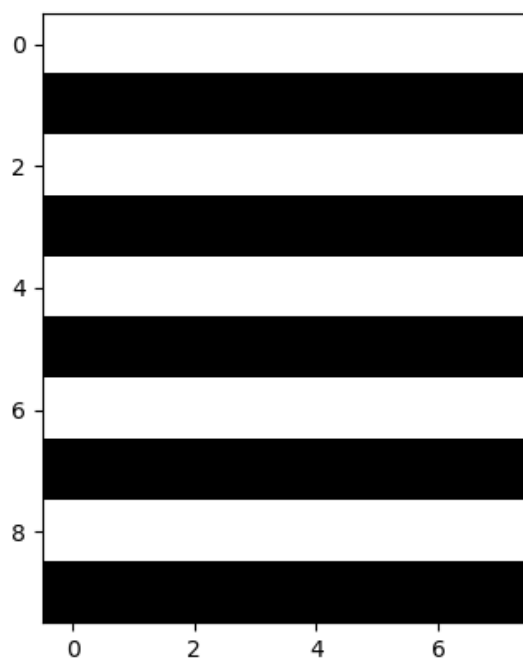
Wygodnym narzędziem do szybkiego indeksowania tablic w oparciu o listy o złożonej zawartości, są wytworniki list (ang. list comprehensions). Przykład wykorzystania wytworników list do indeksowania dwuwymiarowej tablicy przedstawiono na listingu poniżej.

Listing 2.5: Indeksowanie tablic z wykorzystaniem wytworników list

```

1 from pylab import *
2 A=zeros((10,8))
3 A[0:-1:2,:]=1
4 B=zeros(A.shape)
5 for (y,x) in [(y,x) for y in range(A.shape[0]) for x in range(A.shape[1]) if A[y,x]> 0.5]:
6     print y,x
7     B[y,x]=A[y,x]
8
9 imshow(B, interpolation='none', cmap=cm.gray)
10 show()

```



Rysunek 2.3: Indeksowanie tablic z wykorzystaniem wytworników list

Kolejne dwa listingi przedstawiają sposób indeksowania tablic z wykorzystaniem funkcji `ndenumerate()` oraz `ndindex()`.

Listing 2.6: Indeksowanie tablic z wykorzystaniem `ndenumerate`

```

1 from pylab import *
2 A = zeros((11,9))
3 A[:,::2,::2]=1

```

```
4 B=zeros(A.shape)
5 for (y,x),i in ndenumerate(A):
6     print y,x,i
7     B[y,x]=A[y,x]
8
9 imshow(B,interpolation='none', cmap=cm.autumn)
10 show()
```

Listing 2.7: Indekoswanie tablic z wykorzystaniem ndindex

```
1 from pylab import *
2 A = zeros((11,9))
3 A[5::3,0:8:3]=1
4 B=zeros(A.shape)
5 for y,x in ndindex(A.shape):
6     print y,x
7     B[y,x]=A[y,x]
8
9 imshow(B,interpolation='none', cmap=cm.autumn)
10 show()
```

3 Zadania do wykonania

1. Uruchomić i przeanalizować przykłady z instrukcji
2. Utworzyć funkcje generujące przebiegi: skokowy, sinusoidalny, prostokątny, piłokształtny o zadawanej częstotliwości i długości trwania.
3. Utworzyć przykładowe wykresy dla własnych funkcji jednowymiarowych wykorzystując różne sposoby indeksowania.
4. Utworzyć funkcje do tworzenia obrazów testowych z możliwością zadawania parametrów: linie poziome, linie pionowe, krata, szachownica.