Sprawozdanie z Laboratorium nr. 9

* Filip Pasternak
* Grupa lab. 7 - piątek 18:30

# ZAD 1

# Wczytać sygnały, oraz wyznaczyć ich średnią.

x = [0 1 3];

mean(x)

ans = 1.3333

data\_w = data;

mean(data\_w)

ans = -2.4346e-05

mean(data\_p)

ans = -4.2175e-04

# ZAD 2

# Wyznaczyć RMS sygnałów.

rms(x)

ans = 1.8257

rms(data\_w)

ans = 0.1855

rms(data\_p)

ans = 0.1415

# ZAD 3

# Wyznaczyć współczynnik szczytów sygnałów.

peak2rms(x)

ans = 1.6432

peak2rms(data\_w)

ans = 4.2778

peak2rms(data\_p)

ans = 4.7018

# ZAD 4

# Wyznaczyć współczynnik peak2peak sygnałów.

peak2peak(x)

ans = 3

peak2peak(data\_w)

ans = 1.5757

peak2peak(data\_p)

ans = 1.2662

# ZAD 5

# Wyznaczyć wartość maksymalną i minimalną sygnałów.

% Min i max x

min(x)

ans = 0

max(x)

ans = 3

% Min i max waitrak.wav

min(data\_w)

ans = -0.7821

max(data\_w)

ans = 0.7936

% Min i max przekladnia.wav

min(data\_p)

ans = -0.6008

max(data\_p)

ans = 0.6654

# ZAD 6

# Obliczyć wariancję sygnałów.

var(x)

ans = 2.3333

var(data\_w)

ans = 0.0344

var(data\_p)

ans = 0.0200

# ZAD 7

# Obliczyć odchylenie standardowe sygnałów.

std(x)

ans = 1.5275

std(data\_w)

ans = 0.1855

std(data\_p)

ans = 0.1415

# ZAD 8

# Obliczyć energię sygnałów.

% energia

energia\_x = sum(abs(x).^2)

ans = 10

energia\_w = sum(abs(data\_w).^2)

ans = 1.5179e+03

energia\_p = sum(abd(data\_p).^2)

ans = 883.1704

# ZAD 9

# Obliczyć średnią moc sygnałów.

power\_x = bandpower(x)

power\_x = 3.3333

power\_w = bandpower(data\_w)

power\_w = 0.0344

power\_p = bandpower(data\_p)

power\_p = 0.0200

# ZAD 10

# Obliczyć współczynnik RMS sygnałów.

rss\_x = rssq(x)

rss\_x = 3.1623

rss\_w = rssq(data\_w)

rss\_w = 38.9602

rss\_p = rssq(data\_p)

rss\_p = 29.7182

# ZAD 11

# Użyć funkcji *seqperiod()* dla podanych macierzy.

# Funkcja ta służy do znajdowania okresu z jakim w kolumnach tworzą się pewne powtarzające się kombinacje np. 121212.

x\_zad11\_a = [4 0 1 6;

2 0 2 7;

4 0 1 5;

2 0 5 6];

x\_zad11\_b = [4 0 1 6;

2 0 2 7;

4 0 1 5;

2 0 5 6;

1 0 1 7];

seqperiod(x\_zad11\_a)

ans = 1×4

2 1 4 3

seqperiod(x\_zad11\_b)

ans = 1×4

5 1 4 3

# ZAD 12

# Zapoznać się z funkcją *findpeaks()*

fs=5000;

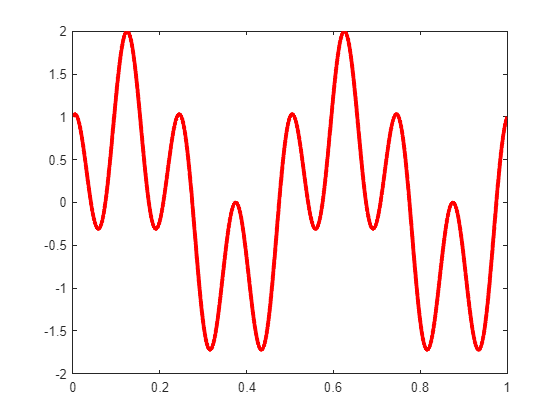
t=0:(1/fs):1;

X=sin(2\*pi\*2\*t);

Y=cos(2\*pi\*8\*t);

s=X+Y;

plot(t, s, 'r', 'LineWidth', 3);



peaks = findpeaks(s)

peaks = 1×8

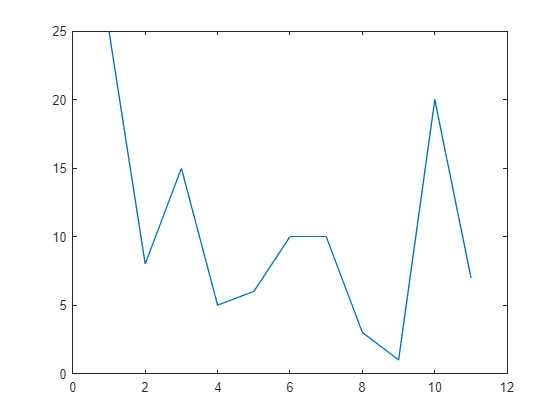
1.0314 2.0000 1.0314 0 1.0314 2.0000 1.0314 ⋯

# ZAD 13

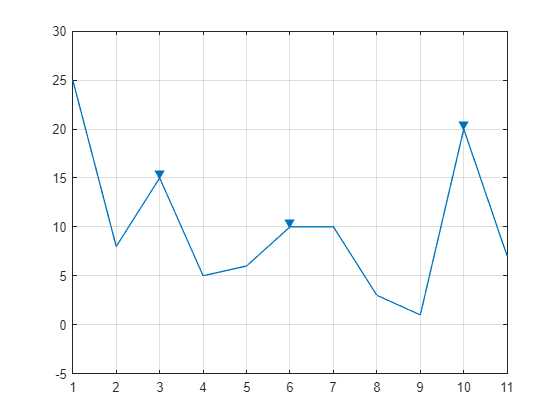
# Użyć *findpeaks()* do danego zestawu danych.

s = [25 8 15 5 6 10 10 3 1 20 7];

plot(s)



findpeaks(s)



peaks = findpeaks(s)

peaks = 1×3

15 10 20

# ZAD 14

# Zapoznać się z działaniem parametru SortStr funkcji *findpeaks().*

peaks = findpeaks(s, "SortStr","descend")

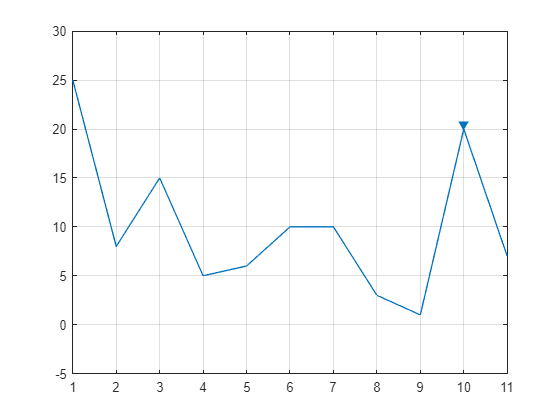
peaks = 1×3

20 15 10

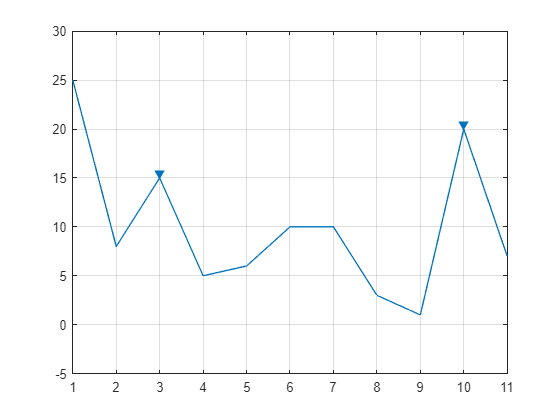
# ZAD 15

# Zapoznać się z parametrem Threshhold funkcji *findpeaks()*

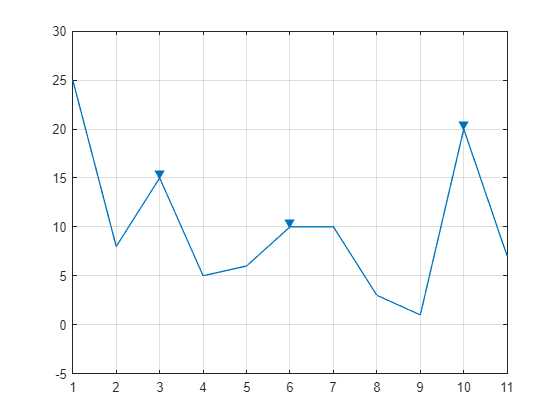
findpeaks(s, "Threshold", 10)



findpeaks(s, "Threshold", 5)



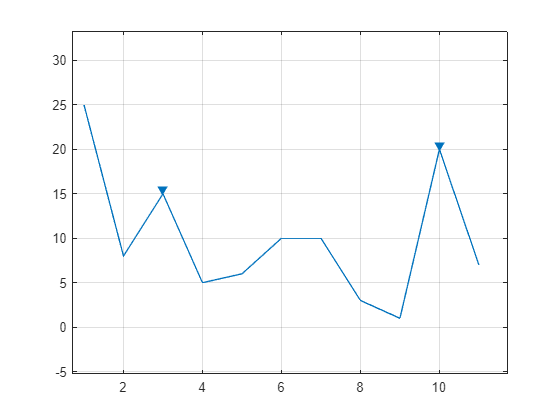
findpeaks(s, "Threshold", 0)



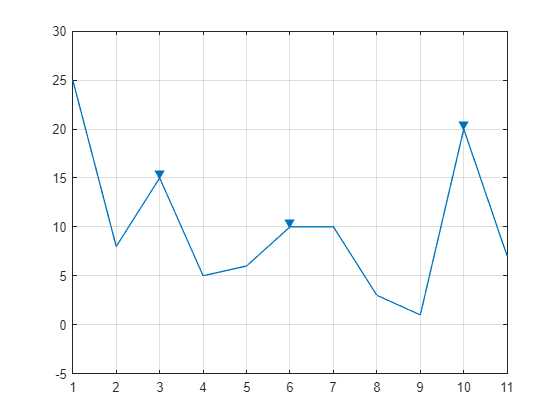
# ZAD 16

# Zapoznać się z parametrem MinPeakHeight funkcji *findpeaks()*

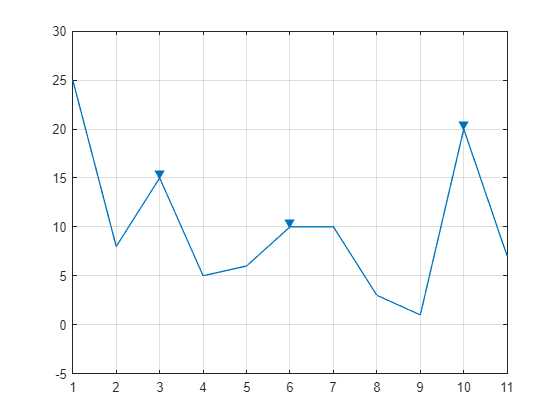
findpeaks(s, "MinPeakHeight", 12)



findpeaks(s, "MinPeakHeight", 8)



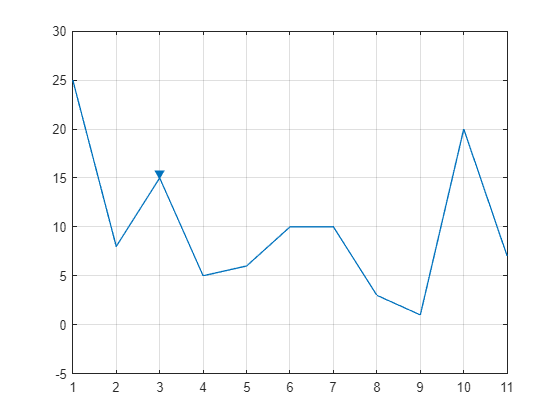
findpeaks(s, "MinPeakHeight", 3)



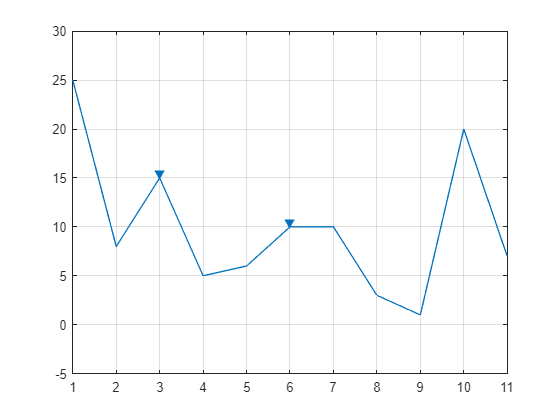
# ZAD 17

# Zapoznać się z parametrem NPeaks funkcji *findpeaks()*

findpeaks(s, "NPeaks", 1)



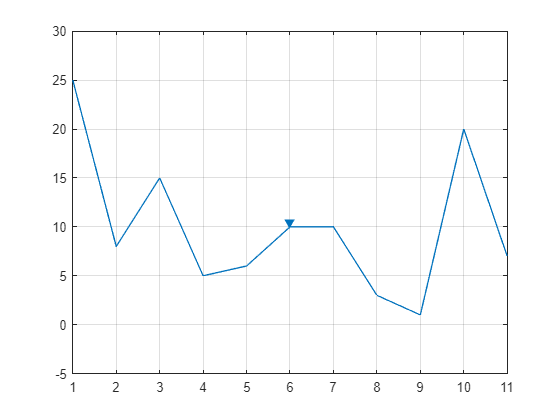
findpeaks(s, "NPeaks", 2)



# ZAD 18

# Zapoznać się z parametrem MinPeakWidth funkcji *findpeaks()*

findpeaks(s, "MinPeakWidth", 1)



# ZAD 19

# Użyć uzyskanej wiedzy do porównania ze sobą sygnałów podanych w ćwiczeniu z zastosowaniem K-NN.

data\_p20 = data\_p;

data\_w20 = data\_w;

% Porownywanie

dif\_rms\_23\_20\_w = sum(abs(rms(data\_w23) - rms(data\_w20)))

dif\_rms\_23\_20\_w = 0.0068

dif\_rms\_23\_21\_w = sum(abs(rms(data\_w23) - rms(data\_w21)))

dif\_rms\_23\_21\_w = 0.0036

dif\_rms\_24\_20\_w = sum(abs(rms(data\_w24) - rms(data\_w20)))

dif\_rms\_24\_20\_w = 8.8342e-04

dif\_rms\_24\_21\_w = sum(abs(rms(data\_w24) - rms(data\_w21)))

dif\_rms\_24\_21\_w = 0.0023

dif\_rms\_23\_20\_p = sum(abs(rms(data\_p23) - rms(data\_p20)))

dif\_rms\_23\_20\_p = 0.0054

dif\_rms\_23\_21\_p = sum(abs(rms(data\_p23) - rms(data\_p21)))

dif\_rms\_23\_21\_p = 0.0053

dif\_rms\_24\_20\_p = sum(abs(rms(data\_p24) - rms(data\_p20)))

dif\_rms\_24\_20\_p = 0.0034

dif\_rms\_24\_21\_p = sum(abs(rms(data\_p24) - rms(data\_p21)))

dif\_rms\_24\_21\_p = 0.0033

# Pytania:

1. Podstawowymi parametrami sygnałów są średnia kwadratowa (RMS), wartość średnia, wariancja, lub RSS (Root Sum Square)
2. Parametry sygnałów mogą być zastosowane do rozpoznawania obiektów wszelkiego typu, jak inteligentny monitoring lub wszelkie programy zamieniające mowę na tekst.
3. Wartość złożona z RMS, RSS i wartości maksymalnej sygnału nie będzie dla nas użyteczna z uwagi na silne skorelowanie tych parametrów.
4. *findpeaks()* pozwala na znalezienie ekstremów podanym sygnale oraz daje dużą elastyczność co do stawiania warunków jakie ekstrema mają zostać wzięte pod uwagę. Została przetestowana w zadaniach 11-18.

# Wnioski:

Matlab posiada szeroki wachlarz możliwości dotyczący interpretowania i opisywania sygnałów za pomocą przydatnych parametrów. Posiada wbudowane funkcje do ich wyznaczania co znacznie ułatwia działania na tych sygnałach i nas odciąża w kwestii uporczywych wzorów matematycznych. Parametry te można wykorzystywać do porównywania ze sobą róznych sygnałów co może zostać, przy odpowiedniej ilości danych, uzyte do porównywania ze sobą różnych sygnałów i znajdowania między nimi podobieństw.