

# Cyfrowe przetwarzanie sygnału

## Zadanie 1

Mateusz Woskowicz 208345

Szymon Marciniak 203937

## 1. Wstęp

Celem sprawozdania jest podsumowanie prac nad programem, którego zadaniem jest generowanie sygnałów cyfrowych, ich wizualizacja oraz wykonywanie na nich operacji.

## 2. Sygnały możliwe do wygenerowania

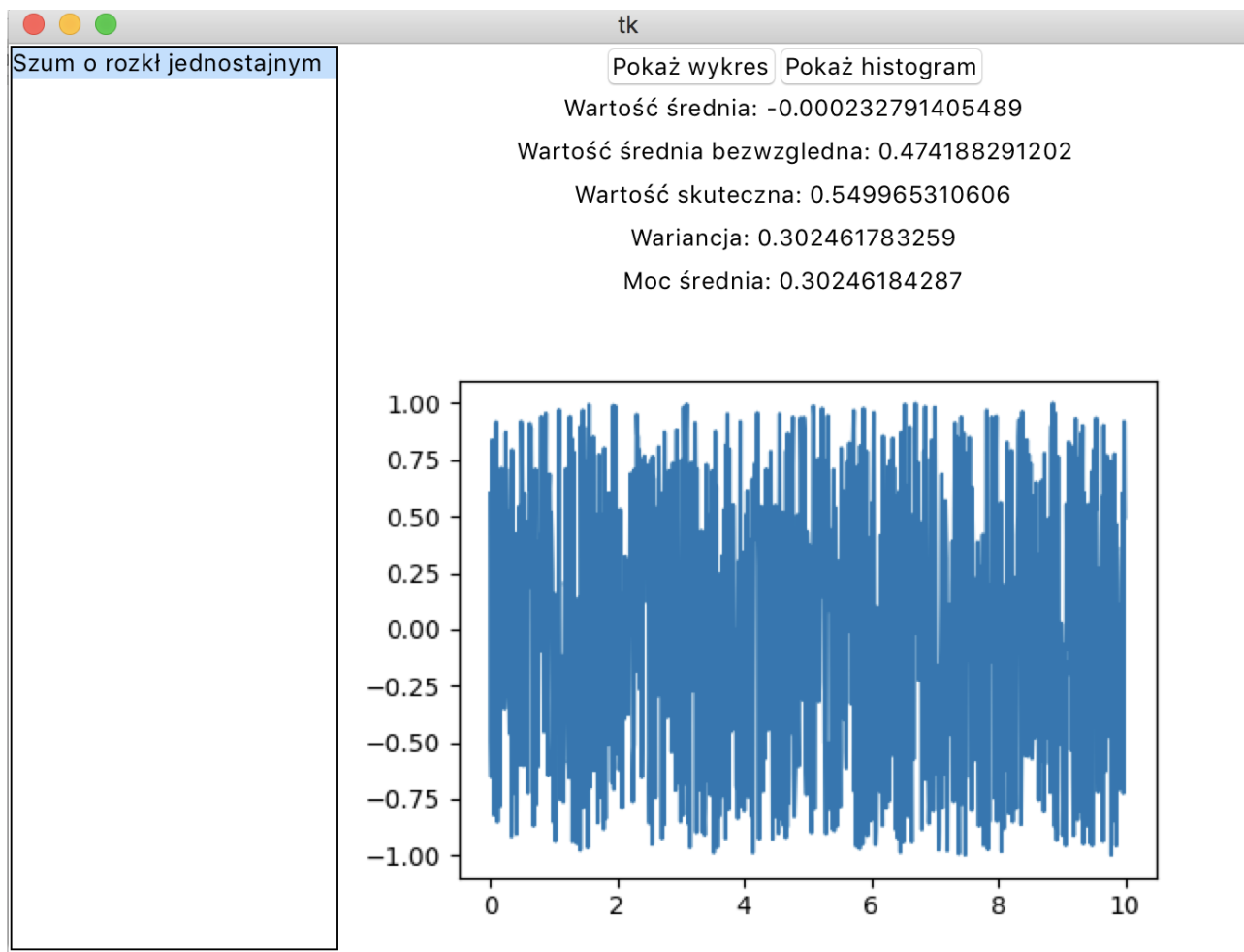
W projekcie zaimplementowane zostało generowanie następujących sygnałów i szumów:

- szum o rozkładzie jednostajnym
- szum gaussowski
- sygnał sinusoidalny
- sygnał sinusoidalny wyprostowany jednopółwkowo
- sygnał sinusoidalny wyprostowany dwupółwkowo
- sygnał prostokątny
- sygnał prostokątny symetryczny
- sygnał trójkątny
- skok jednostkowy
- impuls jednostkowy
- szum impulsowy

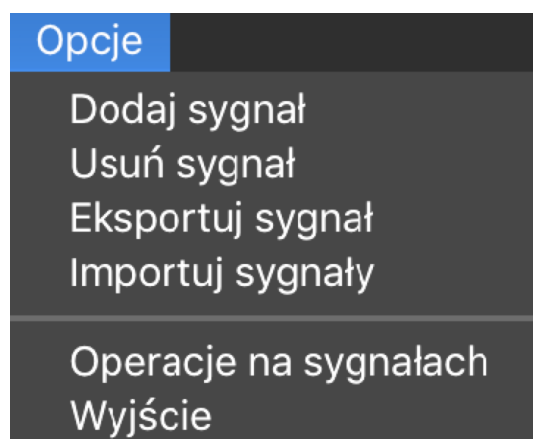
## 3. Obsługa programu

Z menu kontekstowego programu należy wybrać opcję „Dodaj sygnał”. Następnie należy z listy sygnałów wybrać pożądaną przez nas sygnał, wpisując wymagane parametry. Po wciśnięciu przycisku „Dodaj”, zostanie on dodany do listy wszystkich sygnałów.

Typ funkcji	szum o rozkładzie jednostajnym
Nazwa	<input type="text"/>
Amplituda	<input type="text"/>
Czas początkowy	<input type="text"/>
Czas trwania	<input type="text"/>
Częstotliwość próbkowania	<input type="text"/>
<input type="button" value="Dodaj"/>	



Wszystkie dostępne akcje menu kontekstowego pokazane są na poniższym zrzucie ekranu.



Aby wykonać operacje matematyczne na dodanych wcześniej sygnałach, należy wybrać interesujące nas sygnały, a następnie wybrać z listy pożądane działanie. Dostępne operacje matematyczne to dodawanie, odejmowanie, mnożenie oraz dzielenie.



W celu eksportu sygnału do pliku, należy kliknąć na interesujący nas sygnał, następnie wybierając z menu kontekstowego opcję „Eksportuj sygnał”. W celu importowania wcześniej wyeksportowanych sygnałów wystarczy wybrać opcję „Importuj sygnały”. Wszystkie zarchiwizowane wcześniej sygnały zostaną przywrócone.

## 4. Implementacja

Do generowania danych wykorzystano wzory z instrukcji do zadania 1, przepisując je na język programowania *Python*. Do utworzenia interfejsu użytkownika wykorzystano bibliotekę *Tkinter*. Wykresy generowane są za pomocą biblioteki *matplotlib*.

Podstawową klasą widoku jest *BasicView*, która zarządza interakcją z użytkownikiem. Funkcje zdefiniowane w plikach *noise\_generator* oraz *signal\_generator* odpowiadają za tworzenie szumów bądź sygnałów z określonymi wcześniej parametrami. Zwracają one obiekty klasy *CPSSignal*.