

*Laboratorium
Przetwarzania Sygnałów*

Ćwiczenie 3

„Filtracja i korelacja sygnałów dyskretnych”

Opracowali:

- prof. dr hab. inż. Krzysztof Kałużyński
- dr inż. Beata Leśniak-Plewińska
- dr inż. Jakub Żmigrodzki

Zakład Inżynierii Biomedycznej,
Instytut Metrologii i Inżynierii Biomedycznej,
Wydział Mechatroniki Politechniki Warszawskiej.

Warszawa, 2017

1. Cel ćwiczenia.

W ramach ćwiczenia studenci zapoznają się z podstawowymi właściwościami i metodami projektowania filtrów cyfrowych oraz z podstawowymi właściwościami i zastosowaniami funkcji oraz współczynników korelacji i autokorelacji sygnałów deterministycznych i losowych.

2. Wymagane wiadomości.

1. Filtracja:

- a) Podstawowe właściwości filtrów NOI i SOI.
- b) Definicje charakterystyk:
 - amplitudowej,
 - fazowej,
 - opóźnienia grupowego,
 - odpowiedzi impulsowej,
- c) Podstawowe właściwości systemów liniowych niezmiennych w czasie (LTI).

2. Korelacja:

- a) Funkcje korelacji wzajemnej i autokorelacji i ich właściwości.
- b) Współczynniki korelacji wzajemnej i autokorelacji i ich właściwości.
- c) Związek pomiędzy funkcją autokorelacji a widmową gęstością mocy.

3. Literatura.

- T. Zieliński, „Cyfrowe przetwarzania sygnałów”, WKŁ, 2005.
- R.G. Lyons, „Wprowadzenie do cyfrowego przetwarzania sygnałów”, WKŁ, 2006.

4. Spis funkcji i instrukcji przydatnych podczas realizacji ćwiczenia.

Opisane w tym punkcie funkcje nie są standardowymi wbudowanymi funkcjami MATLAB'a i zostały stworzone na potrzeby Laboratorium PTS.

O ile w trakcie realizacji ćwiczenia zajdzie potrzeba skorzystania ze standardowej funkcji (polecenia) lub instrukcji MATLAB'a, należy się z nią zapoznać wykorzystując pomoc MATLAB'a za pomocą polecenia help lub doc, np. help sin lub doc sin.

```
[t, syg] = sygnal (typ, Amp, fsyg, faza, sk_stala, wsp_wyp, fp, T, typ_wykresu)
```

Funkcja służy do generacji sygnałów sinusoidalnych, prostokątnych, trójkątnych bądź losowych.

Parametrami wyjściowymi funkcji są:

1. **t** – wektor, którego wartości reprezentują czas akwizycji kolejnych próbek sygnału,
2. **syg** - wektor wartości generowanego sygnału.

Parametrami wejściowymi funkcji są:

1. **typ** - typ sygnału jaki ma zostać wygenerowany:
 - a) **'sinus'** – generuje sygnał sinusoidalny o zadanej amplitudzie, częstotliwości, fazie oraz składowej stałej,
 - b) **'prostokat'** – generuje sygnał prostokątny o zadanej amplitudzie, częstotliwości, fazie, wypełnieniu oraz składowej stałej,
 - c) **'trojkat'** – generuje sygnał trójkątny o zadanej amplitudzie, częstotliwości, fazie, wypełnieniu oraz składowej stałej,
 - d) **'losowy'** – generuje szum biały o rozkładzie Gaussa o zadanej amplitudzie.
2. **Amp** - amplituda generowanego sygnału,
3. **fsyg** - częstotliwość generowanego sygnału,
4. **faza** - faza generowanego sygnału
5. **sk_stala** - składowa stała generowanego sygnału
6. **wsp_wyp** - współczynnik wypełnienia generowanego sygnału,
7. **fp** - częstotliwość próbkowania generowanego sygnału,
8. **T** - czas trwania (próbkowania) generowanego sygnału,
9. **typ_wykresu** – określa typ prezentacji graficznej:
 - a) **'dyskretny'** – generowany jest wykres dyskretny (obrazujący tylko znane wartości sygnału),
 - b) **'ciagly'** – generowany jest wykres ciągły (próbki pomiędzy znanymi wartościami sygnału są aproksymowane prostymi łączącymi dwa kolejne punkty),
 - c) **[]** - brak wykresu,

Przykład:

```
[t1, s1] = sygnal ('sinus', 5, 20, 0, 0, [], 100, 1, []);
```

Powyższa komenda wygeneruje sygnał dyskretny **s1** (w funkcji **t1**) powstały na skutek spróbkowania sinusoidalnego sygnału ciągłego o amplitudzie równej 5 i częstotliwości 20 Hz. Częstotliwość próbkowania wynosi 100 Hz a czas akwizycji sygnału T=1 s. Brak automatycznej generacji wykresu.

```
[f, wa, faza] = WA(typ, syg, nfft, fp, typ_wykresu)
```

Funkcja wyznacza widmo amplitudowe (jedno lub dwustronne) sygnału **syg**.

Parametrami wyjściowymi funkcji są:

1. **f** – wektor częstotliwości dla których zostały wyznaczone wartości amplitudy,
2. **wa** – wektor wartości widma amplitudowego sygnału **syg**,
3. **faza** – wektor wartości fazy.

Parametrami wejściowymi funkcji są:

1. **typ** – określa czy ma zostać wyznaczone widmo amplitudowe jedno czy dwustronne,
 - a) 'jednostronne',
 - b) 'dwustronne',
2. **syg** – sygnał wejściowy,
3. **nfft** – długość (w próbkach) okna danych poddanych DTF, jeżeli **nfft** jest większe od długości danych wejściowych (**syg**) to dane uzupełniane są zerami tak aby długość okna danych wynosiła **nfft**, w odwrotnej sytuacji tylko próbki od pierwszej do **nfft** są podane analizie. Jeżeli jako wartość zostanie podana pusta tablica (`[]`) to **nfft** automatycznie będzie równe długości wektora danych wejściowych (**syg**),
4. **fp** – częstotliwości próbkowania sygnału wejściowego (**syg**),
5. **typ_wykresu** – określa typ prezentacji graficznej:
 - a) 'dyskretny' – generowany jest wykres dyskretny (obrazujący tylko znane wartości sygnału),
 - b) 'ciagly' – generowany jest wykres ciągły (próbki pomiędzy znanymi wartościami sygnału są aproksymowane prostymi łączącymi dwa kolejne punkty),
 - c) `[]` - brak wykresu,

Przykład:

```
[f2, wa2, faza2] = WA('jednostronne', s2, 1024, 250, 'ciagly');
```

Powyższa komenda wyznaczy jednostronne widmo amplitudowe sygnału **s2** (próbkowanego częstotliwością 250 Hz) o długości DTF 1024 próbek, wynik zostanie zapisany na zmiennych **f2**, **wa2** i **faza2** oraz zostanie utworzony wykres ciągły przedstawiający widmo amplitudowe sygnału **s2**.

```
[f, wgm] = WGM(syg, nfft, fp, okno, beta, typ_wykresu)
```

Funkcja wyznacza widmową gęstość mocy sygnału wejściowego (**syg**).

Parametrami wyjściowymi są:

1. **f** - wektor częstotliwości dla których zostały wyliczone wartości widmowej gęstości mocy.
2. **wgm** - wektor wartości widmowej gęstości mocy sygnału wejściowego (**syg**).

Parametrami wejściowymi są:

1. **syg** – sygnał wejściowy,
2. **nfft** – długość (w próbkach) okna danych poddanych DTF, jeżeli **nfft** jest większe od długości danych wejściowych (**syg**) to dane uzupełniane są zerami tak aby długość okna danych wynosiła **nfft**, w odwrotnej sytuacji tylko próbki od pierwszej do **nfft** są podane analizie. Jeżeli jako wartość zostanie podana pusta tablica (`[]`) to **nfft** automatycznie będzie równe długości wektora danych wejściowych (**syg**),
3. **fp** – częstotliwości próbkowania sygnału wejściowego (**syg**),
4. **okno** – rodzaj okna czasowego (funkcji granic):
 - a) **'prostokatne'** – okno prostokątne,
 - b) **'hann'** – okno Hanny,
 - c) **'hamming'** – okno Hamminga,
 - d) **'blackman'** – okno Blackmana,
 - e) **'kaiser'** – okno Kaisera,
5. **beta** – współczynnik okna Kaisera,
6. **typ_wykresu** – określa typ prezentacji graficznej:
 - a) **'linowy'** – generowany jest liniowy wykres ciągły widmowej gęstości mocy wyrażonej w jednostkach 1/Hz,
 - b) **'logarytmiczny'** – generowany ciągły wykres widmowej gęstości mocy wyrażonej w decybelach (wykres jest znormalizowany do wartości maksymalnej),
 - c) `[]` - brak wykresu,

Przykład:

```
[f3, wgm3] = WGM(s3, 512, 1000, 'hann', [], 'logarytmiczny');
```

Powyższa komenda wyznaczy wartości widmowej gęstości mocy (**wgm3**) sygnału **s3** (próbkowanego częstotliwością 1000 Hz) o długości DTF 512 próbek, wynik zostanie zapisany na zmiennych **f3** i **wgm3** oraz zostanie utworzony wykres logarytmiczny danych wyjściowych.

$p = P_WGM(wgm, fp, nfft, f1, f2)$

Funkcja wyznacza moc (**p**) zawartą w zadanym pasmie sygnału na podstawie wartości widmowej gęstości (**wgm**) mocy sygnału.

Parametrem wyjściowym jest **p** - moc sygnału w zadanym pasmie (od **f1** do **f2**),

Parametrami wejściowymi są:

1. **wgm** - wektor reprezentujący wartości widmowej gęstość mocy sygnału *s*,
2. **fp** - częstotliwość próbkowania sygnału *s*,
3. **nfft** - długość DTF która zastosowano do wyznaczenia WGM sygnału *s*,
4. **f1, f2** - częstotliwości graniczne pasma dla którego zostanie oszacowana moc,

Przykład:

```
[f3, wgm3] = WGM(s3, 512, 1000, 'hann', [], 'logarytmiczny') ;  
p3_20_50= P_WGM(wgm3, 1000, 512, 20, 50);
```

Powyższe komendy wyznaczają:

1. wartości widmowej gęstości mocy (**wgm3**) sygnału **s3** (próbkowanego częstotliwością 1000 Hz) o długości DTF 512 próbek, wynik zostanie zapisany na zmiennych **f3** i **wgm3** oraz zostanie utworzony wykres logarytmiczny danych wyjściowych,
2. wartość mocy (**p3_20_50**) sygnału **s3** w pasmie od 20 do 50 Hz.

spektrogram(syg, okno, dl_okna, nfft, fp)

Funkcja wyznacza spektrogram sygnału wejściowego (**syg**) i prezentuje rezultat w postaci dwuwymiarowej (czas, częstotliwość) funkcji widmowej gęstości mocy. Wartości widmowej gęstości mocy są wyrażone w decybelach (normalizacja do wartości maksymalnej).

Parametrami wejściowymi są:

1. **syg** - sygnał wejściowy
2. **nfft** – długość (w próbkach) okna danych poddanych DTF,
3. **fp** - częstotliwość próbkowania sygnału wejściowego (**syg**),
4. **okno** – rodzaj okna czasowego (funkcji granic):
 - a) **'rect'** – okno prostokątne,
 - b) **'hann'** – okno Hanna,
 - c) **'hamming'** – okno Hamminga,
5. **dl_okna** - długość okna danych wyrażona w próbkach,

Przykład:

```
spektrogram(s4, 'hann', 256, 1024, 1000);
```

```
[czas, kor] = korelacja(A, B, fp, typ_estymatora)
```

Funkcja wyznacza wartości korelacji wzajemnej sygnałów **A** i **B** a następnie wykreśla ich przebieg w funkcji opóźnienia. Uwaga: jeśli **A=B** to wyznaczana jest funkcja **autokorelacji**.

Parametrem wyjściowym są:

1. **czas** – wektor opóźnień dla których wyznaczone są kolejne wartości korelacji,
2. **kor** – wektor wartości korelacji wzajemnej lub autokorelacji.

Parametrami wejściowymi są:

1. **A** - wektor reprezentujący wartości pierwszego sygnału,
2. **B** - wektor reprezentujący wartości drugiego sygnału,
3. **fp** - częstotliwość próbkowania sygnałów wejściowych **A** i **B**,
4. **typ_estymatora** – rodzaj estymatora użytego przy obliczaniu wartości funkcji korelacji:
 - a) **'obciążony'** – estymator obciążony,
 - b) **'nieobciążony'** – estymator nieobciążony,
 - c) **'wsp'** – współczynnik korelacji wzajemnej,

Przykład:

```
[tau67, kor67] = korelacja(s6, s7, 500, 'wsp')
```

Powyższa komenda wyznacza wartości współczynnika korelacji wzajemnej (**kor67**) pomiędzy sygnałami **s6** i **s7** (oba sygnały próbkowane częstotliwością 500 Hz), wynik zostanie zapisany na zmiennych **tau67** i **kor67** oraz zostanie utworzony wykres danych wyjściowych,

5. Przebieg ćwiczenia

Przed przystąpieniem do realizacji ćwiczenia należy poprosić prowadzącego o wskazanie folderu zawierającego dane niezbędne do realizacji ćwiczenia.

Uwaga: Pliki zawierające dane, które należy wykorzystać w ćwiczeniu zawierają następujące zmienne (x oznacza nr sygnału):

- s_x – wektor wartości sygnału s_x w chwilach t_x ,
- t_x – wektor, którego wartości reprezentują czas akwizycji kolejnych próbek sygnału s_x ,
- f_x – częstotliwość próbkowania sygnału s_x .

Oznaczenia użyte w tekście:

(sygnał) – nazwa funkcji, która powinna zostać użyta w celu rozwiązania danego problemu,

s1 – nazwa zmiennej.

`s1a=s1(1:end/4)` – polecenie MATLAB'a, które należy wpisać w *Okno Poleceń* (ang.: *Command Window*).

$s(t) = A \cdot \sin(2 \cdot \pi \cdot f \cdot t)$ - opis w notacji matematycznej.

1. Projektowanie filtrów cyfrowych

- Uruchomić narzędzie **filterDesigner** służące do projektowania i analizy charakterystyk filtrów cyfrowych. Poprosić prowadzącego o zademonstrowanie możliwości narzędzia.
- Zaprojektować filtry NOI (ang.: IIR): Butterwortha, Chebysheva I rodzaju, Chebysheva II rodzaju, Eliptyczny o następujących parametrach:
 - $f_{przep}=1050$ Hz – częstotliwość graniczna pasma przepustowego,
 - $f_{zap}=1350$ Hz – częstotliwość graniczna pasma zaporowego,
 - $t_l=60$ dB – tłumienie pasma zaporowego,
 - $zaf=1$ dB – zafalowania charakterystyki amplitudowej w paśmie przepustowym,
 - $f_p=5000$ Hz – częstotliwość próbkowania,
- Wykreślić charakterystyki amplitudowe, fazowe oraz opóźnienia grupowego dla każdego w filtrów zaprojektowanych w pkt. 1.b. Dla każdego typu charakterystyki stworzyć jeden rysunek na którym będą przedstawione wykresy dla wszystkich filtrów zaprojektowanych w p. 1.b).
- Zaprojektować filtry SOI (ang.: FIR): o jednorodnych zafalowaniach pasma przepustowego (ang.: equiripple) oraz projektowany oknem Kaisera o następujących parametrach:
 - $f_{przep}=1050$ Hz – częstotliwość graniczna pasma przepustowego,
 - $f_{zap}=1350$ Hz – częstotliwość graniczna pasma zaporowego,

- $tl=60$ dB – tłumienie pasma zaporowego,
 - $zaf=1$ dB – zafalowania charakterystyki amplitudowej w paśmie przepustowym,
 - $fp=5000$ Hz – częstotliwość próbkowania,
- e) Wykreślić charakterystyki amplitudowe, fazowe oraz opóźnienia grupowego dla każdego w filtrów zaprojektowanych w p. 1.d). Dla każdego typu charakterystyki stworzyć jeden rysunek na którym będą przedstawione wykresy dla wszystkich filtrów zaprojektowanych w p. 1.d).

Na podstawie wykresów uzyskanych w p. 1.c) i 1.e) opisz cechy charakterystyczne poszczególnych filtrów oraz różnice pomiędzy filtrami typu NOI i SOI.

2. Identyfikacja systemów liniowych.

- a) Wczytać plik „sygnał_nr_1.mat” do przestrzeni roboczej (ang.: workspace). Sygnał **s1** jest spróbkowanym szumem białym gaussowskim, częstotliwość próbkowania **fp1=1000** Hz, czas trwania akwizycji **T1=300** s.
- b) Wczytać plik „filtr_zagadka.mat” do przestrzeni roboczej. Zmienne **Hd_zagadka1, Hd_zagadka2 ... Hd_zagadka5**, reprezentują struktury nieznanymi filtrów cyfrowych (poprosić prowadzącego o wskazanie, który filtr ma zostać zidentyfikowany).
- c) Przefiltrować sygnał **s1** filtrem **Hd_zagadka (filter)**:
`s1_filt=filter(Hd_zagadka,s1);`
- d) Przeanalizować, a następnie wykonać poniższe instrukcje:
- `okno = hann(length(s1))';` - utworzenie okna Hanna o długości sygnału **s1**,
 - `s_przed = s1.*okno;` - wymnożenie sygnału **s1** przez okno Hanna,
 - `s_po = s1_filt.*okno;` - wymnożenie sygnału **s1_filt** przez okno Hanna,
 - `[f_s1, WA_s1, faza_s1]=WA('jednostronne', s_przed, [], fp1, []);` - wyznaczenie widma amplitudowego i fazowego sygnału **s1** (przed filtracją),
 - `[f_s1,WA_s1_filt,faza_s1_filt]=WA('jednostronne',s_po,[],fp1,[]);` - wyznaczenie widma amplitudowego i fazowego sygnału **s1_filt** (po filtracji),
 - `WA_filtru = 20*log10(WA_s1_filt ./ WA_s1);` - wyznaczenie widma amplitudowego nieznanego filtru w skali decybelowej,
 - `faza_filt = unwrap(faza_s1_filt - faza_s1);` - wyznaczenie widma fazowego nieznanego filtru w radianach.
- e) Wykreślić na jednym rysunku (jeden wykres pod drugim) wyznaczone widmo amplitudowe i fazowe nieznanego filtru (**subplot** – standardowa funkcja pakietu MATLAB'a).

Na podstawie uzyskanych charakterystyk (widmo amplitudowe i fazowe) scharakteryzuj nieznaną filtr. Określ jego: typ ze względu na przebieg charakterystyki amplitudowej (pasmo przenoszonych częstotliwości), wartości częstotliwości granicznych, typ ze względu na sposób implementacji (NOI / SOI). Wyjaśnij metodę zastosowaną do identyfikacji nieznanego systemu liniowego.

3. Opóźnienie grupowe filtrów.

- a) Zaprojektować (*filterDesigner*) dolnoprzepustowy filtr Butterwortha oraz SOI (projektowany oknem Kaisera) o następujących parametrach :
- $f_{przep}=9$ kHz – częstotliwość graniczna pasma przepustowego,
 - $f_{zap}=9.5$ kHz – częstotliwość graniczna pasma zaporowego,
 - $t_l=1000$ dB – tłumienie pasma zaporowego,
 - $z_{af}=0.1$ dB – zafalowania charakterystyki amplitudowej w paśmie przepustowym,
 - $F_p=20$ kHz – częstotliwość próbkowania.

Dla pozostałych parametrów pozostawić wartości domyślne.

- a) Zapisać (np. w formie obrazów bmp) wykresy charakterystyk amplitudowych, fazowych i opóźnienia grupowego obu filtrów.
- b) Wyniki wyeksportować w formie obiektów do zmiennych o następujących nazwach: **Hd_NOI (dla filtru Butterwortha), Hd_SOI (dla filtru SOI).**
- c) Wczytać plik „sygnał_nr_2_3_4.mat” do przestrzeni roboczej (ang.: workspace). Sygnał **s4** jest sumą sygnałów **s2** i **s3** $s_4(t)=s_2(t)+s_3(t)$. Dla wszystkich sygnałów częstotliwość próbkowania $f_{p4}=20$ kHz, czas trwania akwizycji $T_4=0.5$ s.
- d) Na jednym rysunku (na trzech wykresach jeden pod drugim) (*subplot* – standardowa funkcja pakietu MATLAB'a) wykreślić przebiegi sygnałów **s2**, **s3** i **s4** w funkcji czasu (**uwaga: ograniczyć każdą oś czasu tak aby obejmowała tylko niezerową część odpowiedniego sygnału np.: od 0 do 0.025 s.**
- e) Wyznaczyć i wykreślić widma amplitudowe (*WA*) sygnałów **s2**, **s3** i **s4**.
- f) Przefiltrować (*filter*) sygnał **s4** zaprojektowanymi filtrami **Hd_NOI**, **Hd_SOI**:
- `s4_SOI = filter(Hd_SOI, s4);`
 - `s4_NOI = filter(Hd_NOI, s4);`
- g) Na jednym rysunku (na trzech wykresach jeden pod drugim) wykreślić (*subplot* – standardowa funkcja pakietu MATLAB'a) przebiegi sygnałów **s4**, **s4_SOI** i **s4_NOI** w funkcji czasu (**uwaga: ograniczyć każdą oś czasu tak aby obejmowała tylko niezerową część odpowiedniego sygnału**).

Wykorzystując zapisane charakterystyki obu filtrów wyjaśnij różnice w przebiegach sygnałów po filtracji (**s4_SOI** i **s4_NOI**).

4. Autokorelacja sygnału prostokątnego.

- a) Wczytać plik „sygnał_nr_6.mat” do przestrzeni roboczej (ang.: workspace). Sygnał **s6** został uzyskany przez spróbkowanie sygnału ciągłego będącego falą prostokątną o amplitudzie $A_6=0.5$, częstotliwości $f_6=1$ Hz, współczynnika wypełnienia równym 0.25 oraz składowej stałej równej 0.5, częstotliwość próbkowania $f_{p6}=100$ Hz, czas trwania akwizycji $T_6=4$ s.
- b) Wykreślić przebieg sygnału **s6** w funkcji **t6** (*plot* – standardowa funkcja pakietu MATLAB'a).

- c) Wyznaczyć i wyrysować (**korelacja**) przebiegi funkcji autokorelacji sygnału **s6** przy użyciu następujących estymatorów:
- nieobciążonego,
 - obciążonego,
 - współczynnika autokorelacji.

5. Autokorelacja paczek sinusoidy.

- a) Wczytać plik „sygnał_nr_7.mat” do przestrzeni roboczej (ang.: workspace). Sygnał **s7** został uzyskany przez spróbkowanie sygnału ciągłego $s(t)$ opisanego następującym równaniem $s(t) = 2 \cdot \sin(2 \cdot \pi \cdot 8 \cdot t) \cdot s6(t)$, częstotliwość próbkowania $f_{p7} = 100$ Hz, czas trwania akwizycji $T7 = 4$ s.
- b) Wykreślić przebieg sygnału **s7** w funkcji **t7** (**plot** – standardowa funkcja pakietu MATLAB'a).
- c) Wyznaczyć i wyrysować (**korelacja**) przebiegi funkcji autokorelacji sygnału **s7** przy użyciu następujących estymatorów:
- nieobciążonego,
 - obciążonego,
 - współczynnika autokorelacji.

Wykorzystując wyniki uzyskane w punktach 5 i 6 wyjaśnij zasadę wyznaczania wartości funkcji autokorelacji oraz korelacji wzajemnej.

6. Autokorelacja szumu białego.

- a) Wczytać plik „sygnał_nr_5.mat” do przestrzeni roboczej (ang.: workspace). Sygnał **s5** jest próbkowanym szumem białym gaussowskim, częstotliwość próbkowania $f_{p5} = 100$ Hz, czas trwania akwizycji $T5 = 4$ s.
- b) Wyznaczyć i wyrysować (**korelacja**) przebiegi funkcji autokorelacji sygnału **s5** przy użyciu następujących estymatorów:
- nieobciążonego,
 - obciążonego,
 - współczynnika autokorelacji.

Wyjaśnij zaobserwowane różnice i podobieństwa w uzyskanych przebiegach. Czy uzyskany wynik jest zgodny z teorią?

7. Korelacja ze wzorcem.

- a) Wczytać plik „sygnał_nr_8.mat” do przestrzeni roboczej (ang.: workspace). Sygnał **s8** został uzyskany przez spróbkowanie sygnału ciągłego $s(t)$ opisanego następującym równaniem $s(t) = 2 \cdot \sin(2 \cdot \pi \cdot 8 \cdot t)$, częstotliwość próbkowania $f_{p8} = 100$ Hz, czas trwania akwizycji $T8 = 0.25$ s.
- b) Wykreślić przebieg sygnału **s8** w funkcji **t8** (**plot** – standardowa funkcja pakietu MATLAB'a).
- c) Wyznaczyć i wyrysować (**korelacja**) przebieg funkcji współczynnika korelacji

wzajemnej sygnału **s7** i **s8**.

8. Korelacja zaszumionego sygnału ze wzorcem.

- Wczytać plik „sygnał_nr_9.mat” do przestrzeni roboczej (ang.: workspace). Sygnał **s9** został uzyskany przez spróbkowanie sygnału ciągłego $s(t)$ opisanego następującym równaniem $s(t) = s7(t) + szum(t)$, częstotliwość próbkowania $f_{p9} = 100$ Hz, czas trwania akwizycji $T9 = 4$ s.
- Wykreślić przebieg sygnału **s9** w funkcji **t9** (*plot* – standardowa funkcja pakietu MATLAB'a).
- Wyznaczyć i wyrysować (*korelacja*) przebieg funkcji współczynnika korelacji wzajemnej sygnału **s9** i **s8**.

Wyjaśnić różnice pomiędzy wykresami uzyskanymi w punkcie 7.c) i 8.c). Jaki wpływ na przebieg funkcji korelacji wzajemnej ma obecność szumu w jednym z sygnałów.

9. Zadanie.

- Na podstawie analizy widma (*WGM*) sygnału **s8** zaprojektować filtr (*filterDesigner*) w celu usunięcia szumu znajdującego się poza pasmem sygnału użytecznego **s8**. Parametry filtru zapisz w odpowiednich rubrykach *Sprawozdania*.
- Przefiltrować sygnał **s9** za pomocą zaprojektowanego filtru (**s9_filt**).
- Wykreślić przebieg sygnału **s9_filt** w funkcji **t9** (*plot* – standardowa funkcja pakietu MATLAB'a).
- Wyznaczyć i wyrysować (*korelacja*) przebieg funkcji współczynnika korelacji wzajemnej sygnału **s9_filt** i **s8**.

Wyjaśnić różnice pomiędzy wykresami uzyskanymi w punkcie 8.c) i 9.c). Czy filtracja sygnałów może być pomocna przy analizie sygnałów z wykorzystaniem korelacji.

Sprawozdanie

Ćwiczenie nr 3 „Filtracja i korelacja sygnałów dyskretnych”

L.p.	Imię i nazwisko	Grupa	Data	
1				
2				
3				
Punkt ćwiczenia	Wyniki	Liczba punktów	Uzyskana liczba punktów	Uwagi prowadzącego
1	X	0,5		
2	X	2		
3	X	2		
4	X	0,5		
5	X	0,5		
6	X	0,5		
7	X	0,5		
8	X	0,5		
9	Parametry filtru	3		
	częstotliwość graniczna pasma przepustowego [kHz]			
	częstotliwość graniczna pasma zaporowego [kHz]			
	tłumienie pasma zaporowego [dB]			
	zafalowania charakterystyki ampl. w paśmie przepustowym [dB]			
	częstotliwość próbkowania [kHz]			
	rzęd filtru			