

TRA 2020Z Ćwiczenie 6

Temat: rozkodowanie sygnału DVB-T do postaci konstelacji – korekcja częstotliwości i fazy

W ćw 5 zajęliśmy się zagadnieniem wyodrębnienia sygnału z szerszego widma radiowego oraz wstępną synchronizacją ramek.

W niniejszym ćwiczeniu postaramy się uzyskać obraz tzw. konstelacji QAM-64, czyli dojść do miejsca, w którym kończy się przetwarzanie sygnałów a zaczyna obszar telekomunikacji - protokołów, bitów, kanałów itd..

uwaga redakcyjna: w ćw. 5 opieraliśmy się na artykule, w którym używano pojęcia "frame" na oznaczenie tego, co aktualnie nazywa się "symbol".

Część teoretyczna – jak wygląda sygnał OFDM w środku

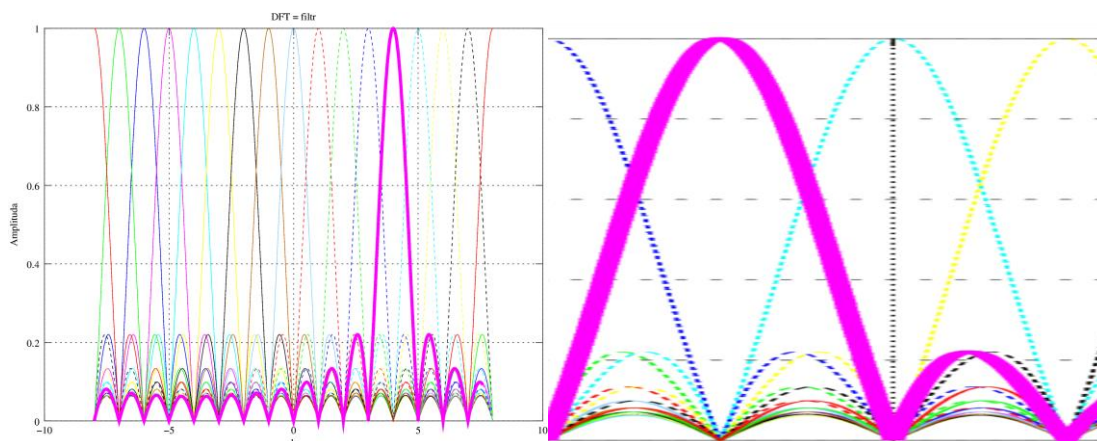
Sens modulacji OFDM to "upchanie" jak najwięcej danych w ograniczonym paśmie przy zachowaniu odporności na transmisję wielodrogową. Osiąga się to transmitując mnóstwo dość powolnych strumieni bitów na różnych częstotliwościach (tzw. podnośnych) - podnośne dobrane są tak, żeby dało się je upakować gęsto, ale aby dały się rozdzielić w odbiorniku.

Modulacja OFDM

Studenci zapewne pamiętają 😊 że DFT ma własność ortogonalności - to znaczy, że sygnał o jednej z częstotliwości $f_s \cdot k_1/N$ nie ma nic wspólnego z sygnałem o innej częstotliwości $f_s \cdot k_2/N$.

Ściślej:

- iloczyn tych sygnałów (o ile $k_1 \neq k_2$) uśredniony za N próbek równy jest zeru
- Filtr dopasowany do N próbek sygnału 1 kompletnie wytnie sygnał 2
- N -punktowa FFT sygnału 1 będzie miała niezerową próbkę tylko dla $k=k_1$, a sygnału 2 - tylko dla $k=k_2$.



Dlatego w OFDM używa się podnośnych o częstotliwościach $f_c + f_s \cdot k/N$, gdzie:

f_c - główna nośna (carrier) - na przykład 522MHz

f_s - częstotliwość próbkowania (w standardzie: $B \cdot 8/7$, gdzie B - szerokość kanału, na przykład 8 MHz)

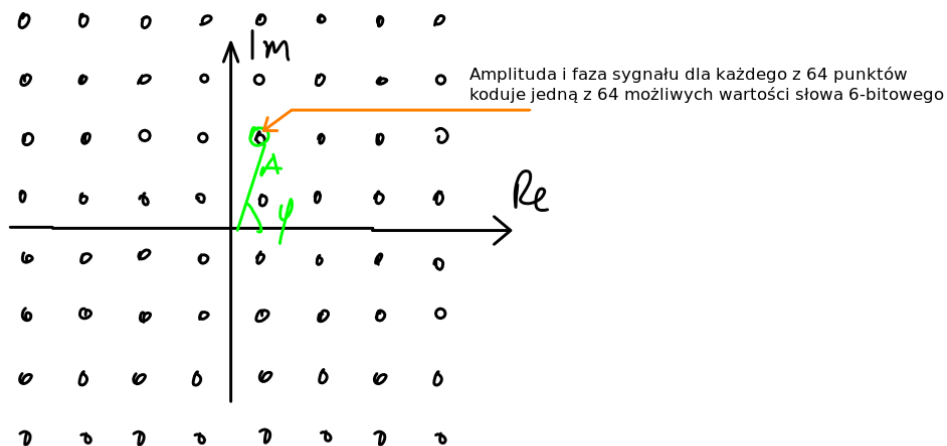
N – liczba podnośnych (na przykład 8192, z czego 6817 jest używanych, a pozostałe, na końcach pasma, nie są nadawane dla zachowania odstępu od następnego kanału)

k – numer podnośnej (od $-N/2$ do $+N/2-1$)

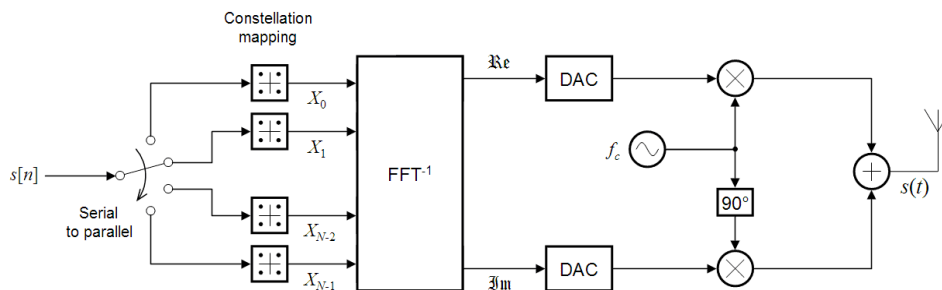
Ortogonalność podnośnych jest zachowana, gdy uśredniamy (filtrujemy, liczymy FFT) za N próbek sygnału. To znaczy, że symbol ma mieć N próbek (i do tego jeszcze dodajemy trochę - np. $1/8 \cdot N$ – na odstęp ochronny, o którym za chwilę).

Na każdej podnośnej jakoś koduje się kilka bitów (od QPSK – 2 bity dla DAB, po 64-QAM – 6 bitów dla DVB-T, a nawet 256-QAM dla DVB-T2) - czyli zmienia się fazę i ew. amplitudę danej podnośnej.

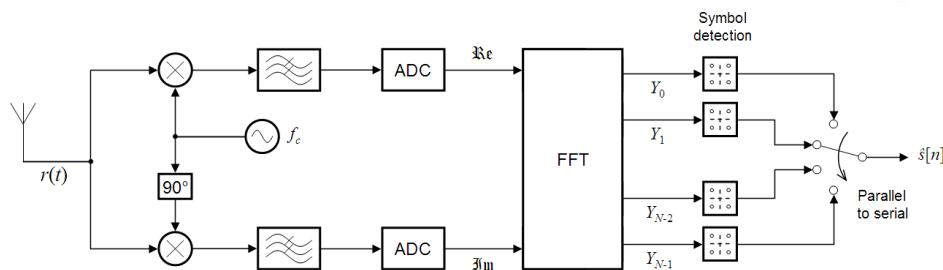
Przykład kodu 64-QAM:



Podsumowując - w nadajniku modulujemy naraz 6817 podnośnych (używając 8192 punktowego IFFT):



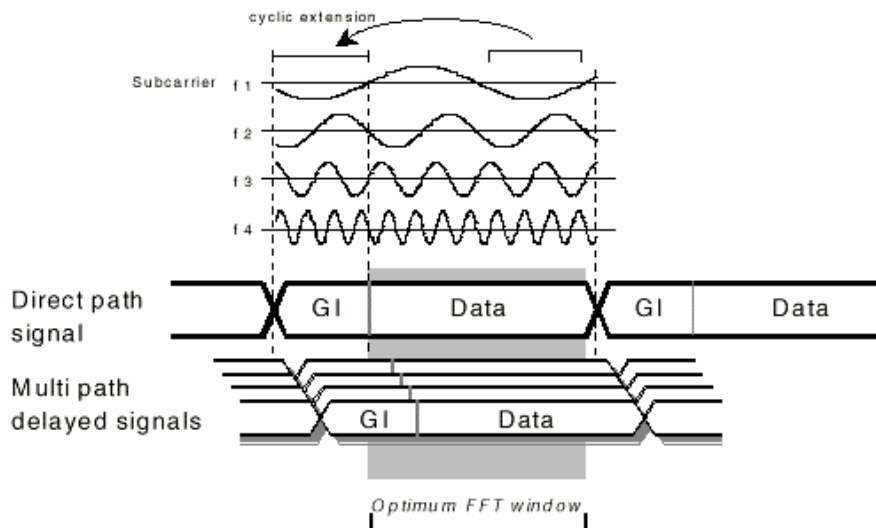
Po czym w odbiorniku demodulujemy je w ten sam sposób - używając 8192 punktowego FFT; dostajemy wtedy fazę i amplitudę każdej podnośnej:



Problem wielodrogowości

(bardzo krótko)

Jeśli sygnał dociera do odbiornika dwoma drogami: np. “na wprost” oraz poprzez odbicie od wieżowca, odbieramy sumę sygnałów o różnych opóźnieniach. To znaczy, że kawałek poprzedniego symbolu nakłada się nam (sumując) na symbol bieżący. Dlatego potrzebny jest odstęp ochronny (GI).



W standardzie OFDM wypełnia się ten odstęp (GI) kopią fragmentu sygnału z końca symbolu, dlatego również nazywamy to “Cyclic Prefix” (CP).

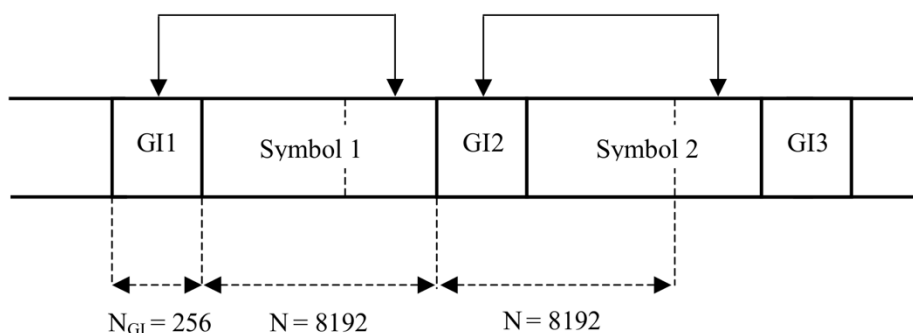
Takie wypełnienie powoduje, że charakter (zespolona sinusoida) sygnału się nie zmienia wskutek zsumowania mnóstwa wielodrogowych kopii – zmienia się tylko amplituda i faza...

Synchronizacja

Aby poprawnie rozkodować sygnał, musimy:

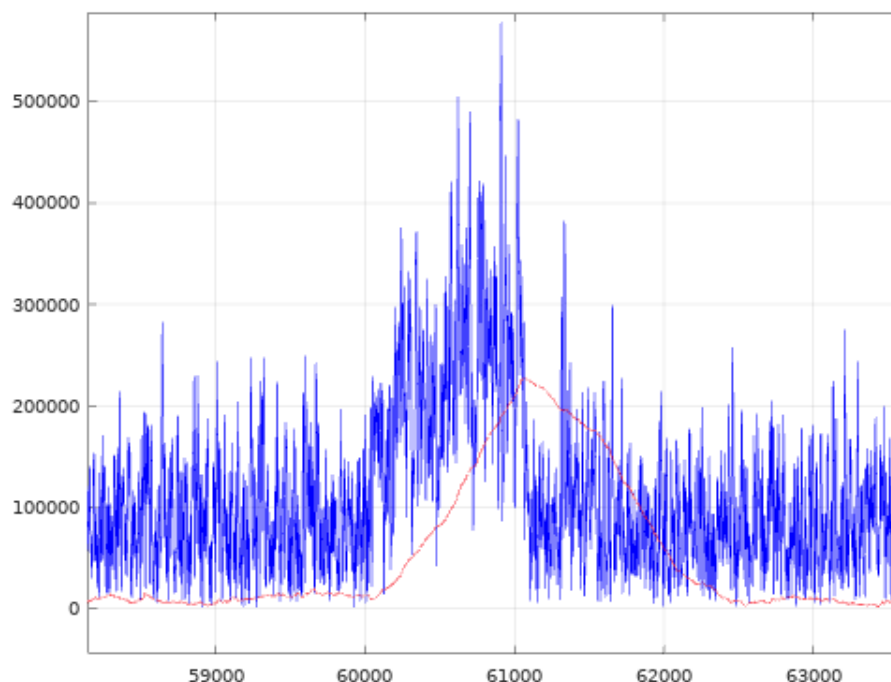
- Skorygować błąd częstotliwości nośnej między nadajnikiem i odbiornikiem (synchronizacja częstotliwości) - to w ćw 6 mamy już przygotowane.
- Odnaleźć początki symboli (synchronizacja czasu) - tak jak robiliśmy to w ćw 5.
- Wyrównać fazy i amplitudy dla różnych podnośnych (8 MHz to już sporo, fazy będą więc różne, i różnie zniekształci się amplituda).

Synchronizacja czasu



W tym celu wymnażamy sygnał ze swoją kopią sprzężoną i przesuniętą o 8192 – kiedy prefiks (GI na rysunku) będzie wymnożony przez fragment, z którego był wzięty, to zadziała ta sama matematyka

którą znamy i stosujemy w filtrze dopasowanym. Uśredniony wynik mnożenia będzie duży (i rzeczywisty).



Rys: wynik mnożenia (moduł) - niebieski; po uśrednieniu za długość GI – czerwony.

Synchronizacja częstotliwości

Jeśli jednak mamy błąd częstotliwości nadajnik-odbioru (zawsze mamy, nikt nie nosi w odbiorniku zegara atomowego), to w/w wynik mnożenia nie będzie rzeczywisty. Jego faza pokaże nam *o ile zdążyła odjechać faza jednego zegara względem drugiego na odcinku od GI do fragmentu, którego on jest kopią*.

Na tej podstawie możemy skorygować częstotliwość - modulując sygnał. **(w tym ćwiczeniu sygnał mamy już skorygowany).**

Mamy natomiast inny problem - sygnał jest próbkowany z częstotliwością 12 MHz zamiast $64/7 = 9.1429$ MHz – o tym kolejna sekcja.

Transformata świergotowa zamiast przepróbkowania

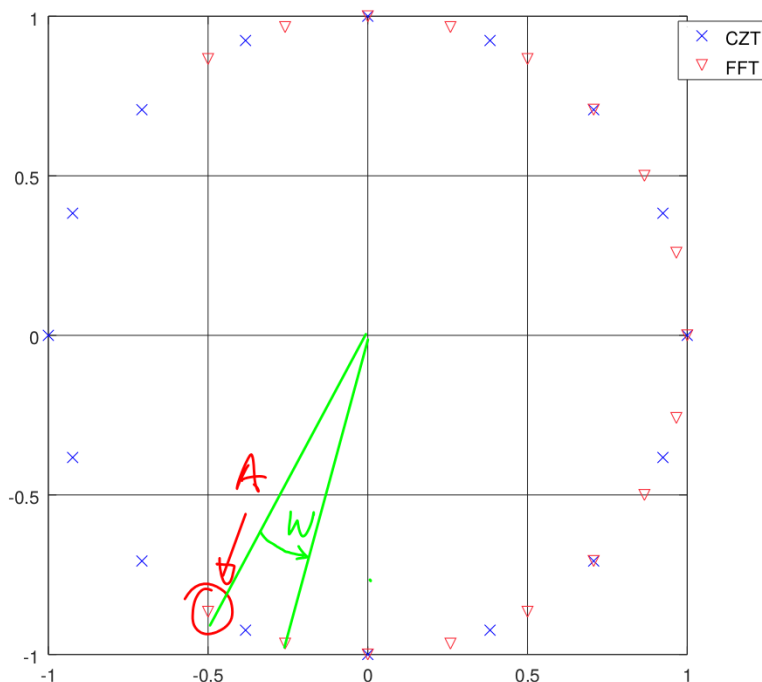
Aby poprawnie zadziałało dekodowanie OFDM (wyizolowanie każdej z N podnośnych), musimy obliczyć widmo dla konkretnych N częstotliwości. Najprościej mieć N próbek sygnału na długości symbolu i obliczyć N-punktową FFT.

Jeśli tak nie jest (my mamy więcej: 10752 zamiast 8192, bo mamy sygnał spróbkowany z wyższą częstotliwością), możemy:

- Przepróbkować sygnał i potem zrobić FFT-8192
- Od razu obliczyć z tych 10752 próbek widmo dla wybranych 8192 częstotliwości

Wykorzystamy opcję drugą i zastosujemy transformatę świergotową (Chirp-Z Transform, CZT).

W skrócie: CZT oblicza to samo co DFT, tylko w innych punktach na płaszczyźnie Z.



Rys: 16 punktowe FFT a 16-punktowe CZT przy 24 próbkach danych
($A = \exp(-j \cdot \pi \cdot 16/24)$, $W = \exp(-j \cdot 2 \cdot \pi \cdot 1/24)$)

Otóż CZT oblicza się poprzez 3 operacje FFT i jedno wymnożenie wektora przez wektor chirp (czyli stosunkowo tanio), i uzyskuje się wartości transformaty Z w punktach określonych wzorem

$$z_k = A \cdot W^{-k}, k = 0, 1, \dots, M - 1$$

gdzie A i W to dowolne liczby zespolone, a M nie musi być równe liczbie próbek sygnału. To znaczy, że możemy m.in. obliczyć transformatę Z na spirali (to nas tu nie interesuje), albo na dowolnym fragmencie okręgu jednostkowego (rysunek) - to przy odpowiednim doborze punktu początkowego A i kroku W daje nam taki sam efekt, jak przepróbkowanie i FFT.

Po to, aby dostać widmo z $N=10752$ próbek sygnału (przy $F_s=12$ MHz) obliczone w $M=8192$ punktach częstotliwości rozłożonych w zakresie od $-\frac{1}{2} \cdot 9.1429$ MHz do $+\frac{1}{2} \cdot 9.1429$ MHz, musimy punkt startowy A ustawić na $\exp(-1i \cdot \dots)$ (i tu wstawić kąt odpowiadający 4.xx MHz przy pi oznaczającym 6 MHz), a krok W na $1/8192$ z 9.1429 MHz / 12 MHz (oczywiście znowu przeliczone na $\exp(-1i \cdot \text{kąt})$).

Warto po obliczeniu W i A kazać Matlabowi wyrysować na płaszczyźnie Z gdzie leżą punkty z_k - można wtedy uniknąć głupich pomyłek.

Synchronizacja (korekcja fazy i amplitudy)

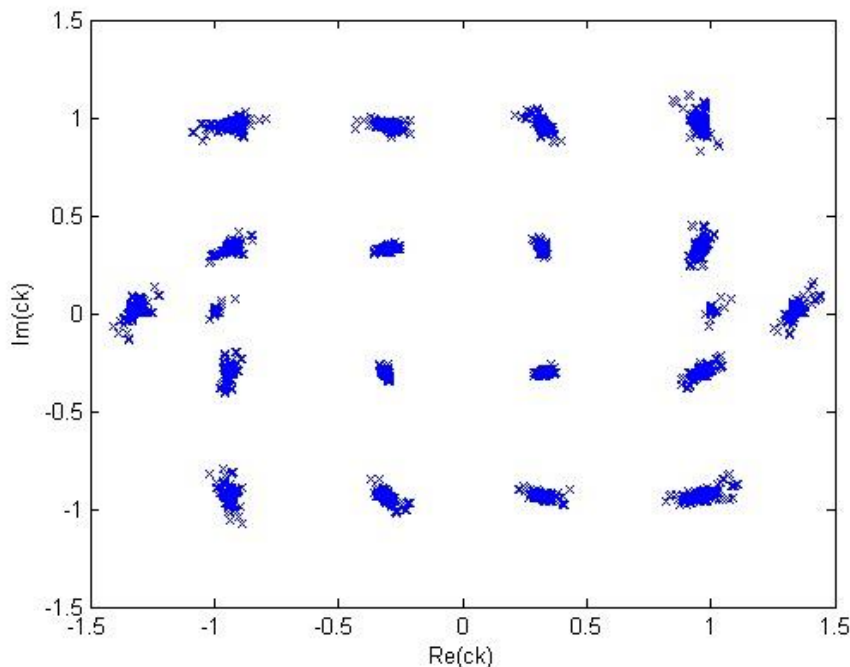
Aby poprawnie zdekodować QAM, trzeba znać fazę początkową sygnału i jego amplitudę. Niestety, na różnych podnośnych jest ona różnie zniekształcona w trakcie propagacji. Aby to skorygować, potrzebny jest jakiś wzorec, punkt odniesienia. W standardzie DVB-T w tym celu na wybranych podnośnych (zmieniających się w czasie według znanego wzorca) nadaje się sygnał o znanej fazie i amplitudzie – tzw. piloty.

Uwaga: W standardzie DAB stosuje się DQPSK – bity zakodowane są w różnicy fazy między m-tym a m-1-szym symbolem. Wtedy korekta fazy nie jest potrzebna.

W ćwiczeniu użyjemy wiedzy o tym, gdzie są piloty stałe. Funkcja `pilots=get_pilots('8k')` poda nam indeksy podnośnych, na którym mają być piloty. Potem:

- Dowiemy się, jakie mają być wartości tych pilotów:
`wk_sequence=get_wk_sequence(6817),`
`pilot_vals=(4/3 * 2 * (1/2 - wk_sequence(pilots)))`
- Obliczymy jaki jest błąd fazy i amplitudy (porównując faktyczne wartości ze wzorcem):
`dane(pilots)./pilot_vals`
- Przeinterpolujemy błąd na te podnośne, na których są dane (więc nie wiemy jaką mają mieć fazę i amplitudę) (`help interp1`)
- Podzielimy dane przez błąd => czyli je skorygujemy.

Wtedy wyświetlone wyniki we współrzędnych Re-Im powinny dać obraz podobny do poniższego - tu jest modulacja QAM-16, a “dodatkowe” ciapki to są nieskompensowane piloty zmienne,



... albo jeszcze ładniejszy, gdy “odgadniemy” że badamy symbol o numerze X i zapytamy, gdzie są **stałe i zmienne** piloty w symbolu nr X (`get_pilots('8k',X)`).

(hint: pierwszy symbol w danych `signal.mat` ma numer 3)

W rzeczywistości można te błędy uśredniać w czasie (warunki propagacji dla telewizora stacjonarnego nie zmieniają się bardzo szybko) i wykorzystać fakt, że wtedy mamy wiedzę o całym mnóstwie punktów do korekty w dziedzinie częstotliwości (ale tego na zajęciach nie zdążymy).

Część wykonawcza

Uwaga: instrukcja jest bardzo uproszczona, rzeczywistość za to jest bogata. Studenci w tym ćwiczeniu jeszcze bardziej niż poprzednio powinni angażować prowadzącego 😊.

6 a) Wczytaj do Matlaba dane z serwera zawierające rzeczywiste sygnały naziemnej telewizji cyfrowej. Wyświetl widmo tego sygnału. Zauważ na widmie strukturę (pojedynczy kanał DVB-T).

(plik signal.mat - sygnał zespolony, który jest już zdemodulowany tak że f_c jest sprowadzona do zera)

6 b) Wykorzystując strukturę sygnału wyznacz długość (w próbkach) pojedynczej ramki dla tego kanału. Użyj funkcji autokorelacji. Wyświetl wynik autokorelacji. Spróbuj znaleźć lokalne maksimum tej funkcji odpowiadające jednej z długości ramki dopuszczonych przez standard (patrz informacje na początku instrukcji). Pamiętaj o różnicy w częstotliwości próbkowania.

6 c) Zaimplementuj funkcję wyznaczającą początki symboli w sygnale:

```
cp_corr = correlate_cp(signal, symbol_length, CP);
```

Wyświetl sygnał `cp_corr`. Wyznacz początki symboli i wytnij jeden cały symbol.

6 d) Zaimplementuj funkcję wyznaczającą transformatę OFDM:

```
symbol_FD = OFDM_transform(symbol_TD, symbol_length, N)
```

wykorzystaj transformatę świergotową (funkcja `cztf()`).

Dobra rada: po zdefiniowaniu parametrów A i W dla `cztf()` wyświetl punkty na płaszczyźnie Z , dla których liczysz transformatę:

```
plot(A*W.^(-[0:M]), '*')
```

Korzystając z tej funkcji przetransformuj symbol do dziedziny częstotliwości.

Wyświetl przetransformowany sygnał.

6 e) Wytnij $K = 6817$ użytecznych środkowych częstotliwości z sygnału (odrzuć pozostałe). Wyświetl sygnał.

6 f) Zaimplementuj funkcję `symbol_out = symbol_filter(symbol_FD, pilots, wk_sequence);`

która na podstawie znanych położenia pilotów, ich faz i amplitud wyznaczy błąd, a następnie skoryguje sygnał.

Uzyskaj sekwencję pseudolosową:

```
wk_sequence = get_wk_sequence(6817);
```

Uzyskaj częstotliwości pilotów:

```
pilots = get_pilots('8k', symbol_number)
```

Sprawdź działanie swojej funkcji dla `symbol_number = 0, 3`. Wyświetl sygnał.