

Politechnika Wrocławska
Wydział Elektroniki
Kierunek Informatyka

Projekt NiDUC

MODULACJA PSK

Autorzy:

JAKUB LABRYSZEWSKI
(251001),
FILIP SOŁDON (249454)

Termin zajęć:

Środa, 7:30-9:00 TN

Prowadzący:

Dr hab. inż.

Henryk Maciejewski

22 czerwca 2020

Spis treści

1	Etap I - założenia projektowe	2
2	Etap II - symulator	2
	2.1 PSK	2
	2.2 APSK	4
3	Etap III - Organizacja eksperymentu symulacyjnego i wyniki	5
	3.1 Wyniki - PSK	6
	3.2 Wyniki - APSK	8
	3.3 Wyniki - APSK vs PSK	16
4	Analiza wyników	22
5	Wnioski	23

1 Etap I - założenia projektowe

Zaczynając pracę nad projektem, postawiliśmy sobie za cel zrealizowanie programu symulującego modulację PSK w środowisku MATLAB. Założyliśmy, że stworzymy tablicę, która będzie przechowywać dane wejściowe - zera i jedynki - i zostanie wypełniona losowymi wartościami dzięki użyciu funkcji `rand()` dostępnej w środowisku MATLAB. Założyliśmy także, że gdy `rand()` przyjmie wartość większą niż 0.5, w tablicy umieścimy 1, a gdy mniejszą niż 0.5, to 0. Dane miały być modulowane i przepuszczane przez kanał z losowymi zakłóceniami, a sygnał miał bazować na funkcji cosinus i być przesuwany w zależności od wartości próbek:

Dla BSK:

o 180° dla 0 oraz o 0° dla 1

Dla QPSK:

45° - (1,0)

135° - (0,0)

225° - (0,1)

315° - (1,1)

Następnie sygnał miał zostać zdemodulowany, aby odczytać odebrane bity. Założyliśmy, że demodulator zacznie demodulację w momencie pierwszego przeskoku fazowego (z 0 na 1 lub odwrotnie), a dokładniej wtedy, gdy różnica dwóch kolejnych próbek będzie większa od amplitudy.

2 Etap II - symulator

Na potrzeby projektu stworzyliśmy 2 symulatory. Jeden do modulacji PSK, a drugi do modulacji APSK.

2.1 PSK

W naszym symulatorze ustawienie liczby punktów w konstelacji sygnału następuje poprzez zmianę wartości M w linii 2. Program wyświetla 4 wykresy wskazowe przedstawiające wyjście kanału transmisyjnego dla odpowiednich wartości (20, 10, 1, 0.1 dB) SNR (Ratio of signal power to noise power). Dla każdego wykresu program wyświetla w Command Window wartość SNR, liczbę przesłanych bitów, liczbę błędnych bitów a także poziom błędów BER (bit error rate). Do utworzenia danych wejściowych użyliśmy funkcji `randi()`, którą losujemy wartości binarne, co można zobaczyć w linii 14. na rysunku nr 2. W celu zmodulowania danych tworzymy modulator za pomocą klasy `PSKModulator`, która jest dostępna w bibliotece `comm` środowiska MATLAB. Podobnie jest w przypadku demodulatora, gdyż korzystamy z tej samej biblioteki, ale tym razem tworzymy obiekt `PSKDemodulator`. W linii 12. tworzymy także instancję klasy `ErrorRate` zawartej w bibliotece `comm` w celu śledzenia statystyk BER. Zmodulowane dane będą przepuszczone przez kanał transmisyjny z biblioteki `comm` - `AWGNChannel`, w którym dodawany jest biały szum

Gaussa. W linii 17. ustawiamy także metodę zaszumiania w kanale na SNR. W pętli w liniach 24. - 55. ustawiamy odpowiednie wartości SNR w kanale, przepuszczamy zmodulowane dane przez kanał, prezentujemy sygnał po wyjściu z kanału na wykresie wskazowym, demodulujemy dane i zliczamy wartość BER.

```

1  %Dla BSK podaj M=2 , Dla QPSK podaj M=4
2  M=64; %Ilość punktów w konstelacji sygnału
3  PHASE = pi/M; %Przesunięcie fazy
4  bitInput = true; %Zaznaczenie czy dane wchodzące do modulatora to wartości binarne czy całkowitoliczbowe
5  bitOutput = true; %Zaznaczenie czy dane wychodzące z demodulatora to wartości binarne czy całkowitoliczbowe
6  amountOfBits = 10000; %Ilość bitów przechodzących przez kanał transmisyjny
7  bitsPerSymbol = log2(M);

```

Rysunek 1: Część pierwsza programu

```

8  %Modulator i demodulator
9  pskModulator = comm.PSKModulator(M,PHASE,'BitInput',bitInput); %Stworzenie obiektu modulatora
10 pskDemodulator = comm.PSKDemodulator(M,PHASE,'BitOutput',bitOutput); %Stworzenie obiektu demodulatora
11 %Stworzenie obiektu ErrorRate
12 errorRate = comm.ErrorRate; %Stworzenie obiektu errorRate do śledzenia statystyk BER(bit error rate)
13 %Dane
14 data = randi([0 1],bitsPerSymbol*amountOfBits,1);
15 %Kanał transmisyjny
16 awgnChannel = comm.AWGNChannel('BitsPerSymbol',bitsPerSymbol); %Stworzenie obiektu reprezentującego kanał transmisyjny.
17 awgnChannel.NoiseMethod = 'Signal to noise ratio (SNR)'; %SNR -stosunek mocy sygnału do mocy szumu
18 %modulujemy dane
19 modulatedData = pskModulator(data);
20 display('Szybkość transmisji (ilość bitów przesyłanych w jednym okresie fali nośnej): ');
21 display((bitsPerSymbol));

```

Rysunek 2: Część druga programu

```

22 %przepuszczenie przez kanał,demodulacja oraz wykresy wskazowe dla SNR=20,10,1,0.1
23 vec = [ 0.1 1 10 20];
24 for i=1:length(vec)
25     release(errorRate);
26     awgnChannel.SNR = vec(i); %Wartość SNR w decybelach
27     channelOutput = awgnChannel(modulatedData);
28     demodulatedData = pskDemodulator(channelOutput);
29     scatterplot(channelOutput);
30     hold on
31     grid
32     plot(modulatedData,'r+')
33     xlabel('In-Phase Amplitude')
34     ylabel('Quadrature Amplitude')
35     if i==1
36         legend('channelOutput, SNR = 0.1');
37         display('SNR = 0.1');
38     elseif i==2
39         legend('channelOutput, SNR = 1');
40         display('SNR = 1');
41     elseif i==3
42         legend('channelOutput, SNR = 10');
43         display('SNR = 10');
44     else
45         legend('channelOutput, SNR = 20');
46         display('SNR = 20');
47     end

```

Rysunek 3: Część trzecia programu

```

48 - berSNR = errorRate(data,demodulatedData);
49 - display('Ilość bitów: ');
50 - display(berSNR(3));
51 - display('Ilość błędów: ');
52 - display(berSNR(2));
53 - display('BER: ');
54 - display(berSNR(1));
55 - end

```

Rysunek 4: Część czwarta programu

2.2 APSK

W drugim symulatorze odpowiedzialnym za modulację APSK ustawiamy ilość punktów w konstelacji dla jednej amplitudy - w naszym programie używamy dwóch amplitud, dlatego ogólnie punktów w konstelacji będzie 2 razy więcej. W linii 4. ustawiamy wartości amplitud, czyli promień okręgów na wykresie wskazowym. W następnej linii ustawiamy też przesunięcie fazowe dla każdej amplitudy. Podobnie jak w symulatorze PSK do generowania binarnych danych wejściowych wykorzystujemy funkcję `randi()`, natomiast, w odróżnieniu od wcześniej opisywanego programu, do modulacji i demodulacji danych nie tworzymy obiektów, lecz wykorzystujemy gotowe funkcje - odpowiednio `apskmod` i `apskdemod`. Do zliczania statystyk BER używamy klasy `ErrorRate` z biblioteki `comm`. W linii 12. tworzymy wektor, który przechowuje badane wartości SNR (0.1, 1, 10, 20, 40, 80, 160 dB), które w linii 17. - w zależności od obiegu pętli - są odpowiednio ustawiane. Zmodulowane dane będziemy przepuszczać przez kanał transmisyjny, do czego wykorzystujemy funkcję `awgn`, a przepuszczony sygnał prezentujemy na wykresie wskazowym. Następnie dane są demodulowane i zliczane są błędy BER. Program wyświetla 7 wykresów, a w Command Window dla każdej wartości SNR wypisywana jest ilość przesłanych bitów, ilość błędnie odebranych bitów oraz wartość BER.

```

1 - N = 32; %Ilość punktów dla jednej amplitudy
2 - M = [N N];
3 - modOrder = sum(M);
4 - randii = [0.5 1.5]; %amplitudy
5 - phOff = [pi/N pi/(N)]; %przesunięcia fazowe
6 - bitsPerSymbol = log2(modOrder); %ilość bitów na przesyłany symbol
7 - amountOfBits = 10000; %ilość przesyłanych liczb
8 - data = randi([0 1],bitsPerSymbol*amountOfBits,1); %Wygenerowanie losowych danych
9 - modulatedData = apskmod(data,M,randii,phOff,'inputType','bit'); %zmodulowanie danych
10 - errorRate = comm.ErrorRate; %utworzenie obiektu do zliczania statystyk BER
11 - sigPow = var(modulatedData); %ustawienie mocy sygnału jako wariancji zmodulowanych danych
12 - vecSNR = [ 0.1 1 10 20 40 80 160]; %Wektor wartości SNR
13 - display('Szybkość transmisji (ilość bitów przesyłanych w jednym okresie fali nośnej): ');
14 - display((bitsPerSymbol));
15 - for i=1:length(vecSNR)
16 -     release(errorRate); %zwolnienie obiektu errorRate
17 -     snr = vecSNR(i); %ustawienie wartości SNR
18 -     channelOutput = awgn(modulatedData,snr,sigPow,'linear'); %przepuszczenie zmodulowanych danych przez kanał transmisyjny
19 -     demodulatedData = apskdemod(channelOutput,M,randii,phOff,'OutputType','bit'); % demodulacja sygnału który wyszedł z kanału
20 -     scatterplot(channelOutput); %zaprezentowanie wyników na wykresie wskazowym
21 -     hold on
22 -     grid
23 -     plot(modulatedData,'r+')
24 -     xlabel('In-Phase Amplitude')
25 -     ylabel('Quadrature Amplitude')

```

Rysunek 5: Początek programu

```

26 -         if i==1
27 -             legend('channelOutput, SNR = 0.1');
28 -             display('SNR = 0.1');
29 -         elseif i==2
30 -             legend('channelOutput, SNR = 1');
31 -             display('SNR = 1');
32 -         elseif i==3
33 -             legend('channelOutput, SNR = 10');
34 -             display('SNR = 10');
35 -         elseif i==4
36 -             legend('channelOutput, SNR = 20');
37 -             display('SNR = 20');
38 -         elseif i==5
39 -             legend('channelOutput, SNR = 40');
40 -             display('SNR = 40');
41 -         elseif i==6
42 -             legend('channelOutput, SNR = 80');
43 -             display('SNR = 80');
44 -         else
45 -             legend('channelOutput, SNR = 160');
46 -             display('SNR = 160');
47 -         end
48 -         berSNR = errorRate(data,demodulatedData);
49 -         display('Ilość bitów: ');
50 -         display(berSNR(3));
51 -         display('Ilość błędów: ');
52 -         display(berSNR(2));
53 -         display('BER: ');
54 -         display(berSNR(1));
55 -     end

```

Rysunek 6: Koniec programu

3 Etap III - Organizacja eksperymentu symulacyjnego i wyniki

Wyniki eksperymentu symulacyjnego były generowane za pomocą dwóch wyżej opisanych programów - pierwsza tabela to wyniki modulacji PSK a druga APSK. Eksperyment został zorganizowany w ten sposób, że dla modulacji PSK zostały wykonane badania dla 2, 4, 8, 16, 32 i 64 punktów na wykresie a dla każdej ilości punktów badany został różny poziom SNR (20,10,1,0.1). Badania zostały także wykonane dla dwóch różnych ilości liczb - 1000 i 10000. Przesunięcie fazowe dla każdego przypadku jest równe wartości liczby pi podzielonej przez ilość punktów w konstelacji wykresu. Szybkość transmisji określa ilość bitów przesyłanych w jednym okresie fali nośnej co bezpośrednio wynika z ilości punktów na wykresie. Sposób badania modulacji APSK był podobny do tego z PSK. Badania zostały wykonane dla 8, 16, 32 i 64 punktów w konstelacji wykresu, badane były różne wartości SNR (160, 80, 40, 20, 10, 1, 0.1) a dla każdej wartości SNR i każdej ilości punktów w konstelacji wykresu testy przeprowadziliśmy dla dwóch ilości liczb - 1000 i 10000. Dodatkowo jako, że w tej modulacji dochodzi opcja zmiany amplitudy pomiary były przeprowadzone dla dwóch sytuacji - amplitudy 0.5 i 1 - oraz druga - amplitudy 0.5 i 1.5 - odpowiednio dla mniejszego i większego okręgu na wykresie. Przesunięcia fazowe dla każdej ilości punktów w konstelacji to dla mniejszego i większego okręgu

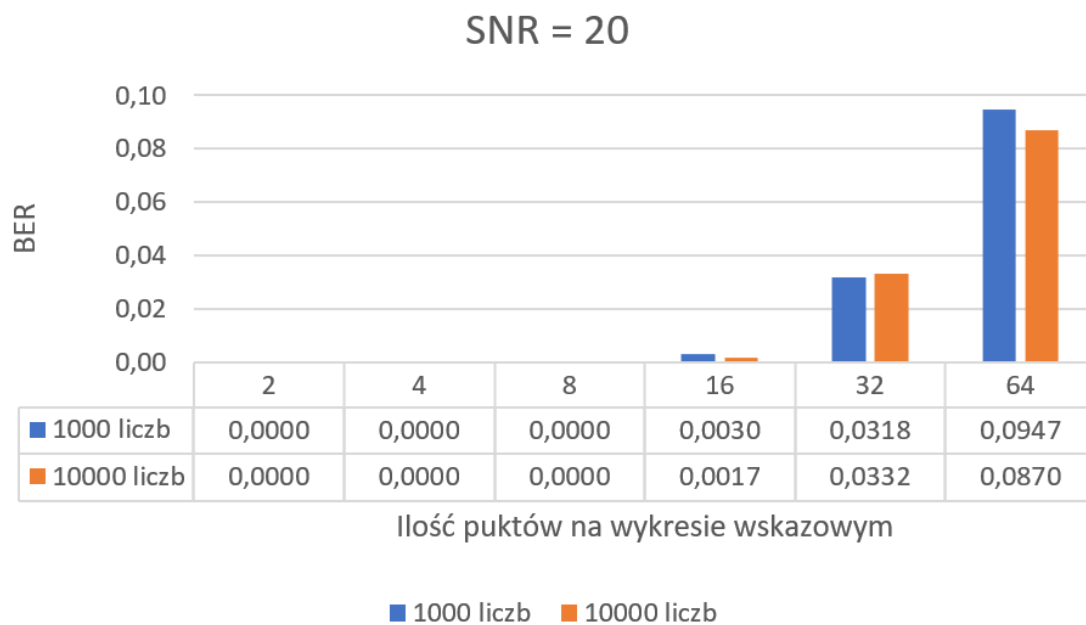
na wykresie π/M , gdzie M to ilość punktów w dla jednego okręgu - czyli połowa wszystkich punktów na wykresie. Przeprowadziliśmy także pomiary dla 16 punktów na wykresie gdzie przesunięcia fazowe ustawiliśmy na $\pi/8$ i $\pi/16$ odpowiednio dla okręgu o mniejszym i większym promieniu. Dla każdej możliwej opcji zliczaliśmy poziom błędów BER zarówno dla modulacji PSK jaki i APSK.

3.1 Wyniki - PSK

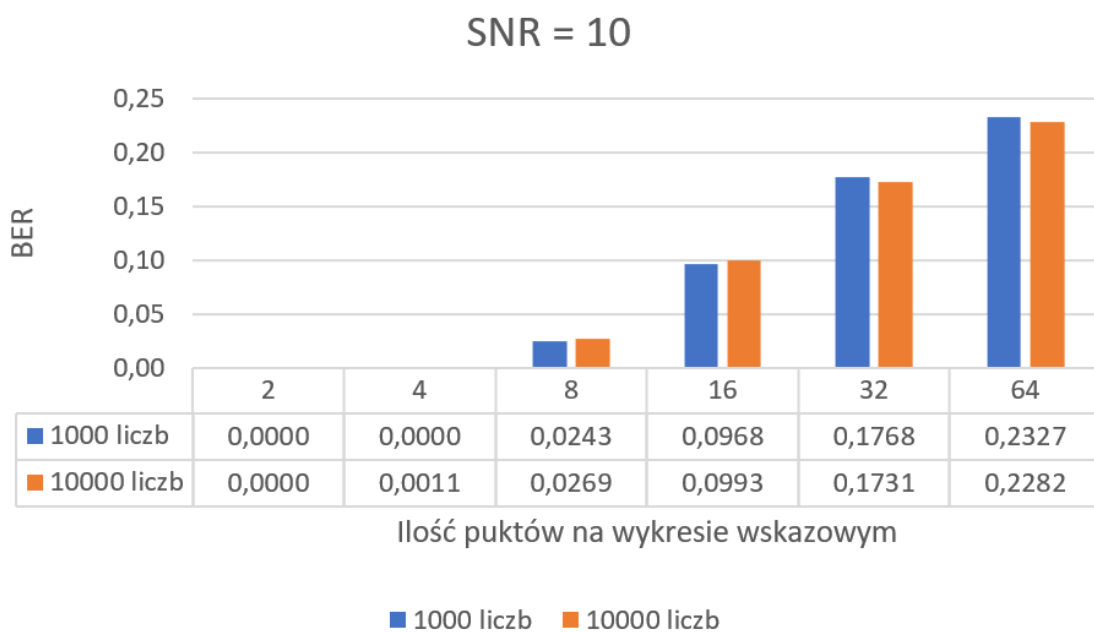
Poniżej umieszczamy wyniki w postaci tabeli oraz wykresów kolumnowych. Każdy wykres odpowiada osobnemu poziomowi SNR.

PSK					
M	SNR	BER		Przesunięcie fazowe	Szybkość transmisji (b/T)
		ilość liczb			
		1000	10000		
2	20	0,0000	0,0000	$\pi/2$	1
	10	0,0000	0,0000		
	1	0,0670	0,0584		
	0,1	0,0700	0,0759		
4	20	0,0000	0,0000	$\pi/4$	2
	10	0,0000	0,0011		
	1	0,1305	0,1268		
	0,1	0,1455	0,1573		
8	20	0,0000	0,0000	$\pi/8$	3
	10	0,0243	0,0269		
	1	0,2170	0,2130		
	0,1	0,2273	0,2413		
16	20	0,0030	0,0017	$\pi/16$	4
	10	0,0968	0,0993		
	1	0,2815	0,2886		
	0,1	0,3015	0,3041		
32	20	0,0318	0,0332	$\pi/32$	5
	10	0,1768	0,1731		
	1	0,3276	0,3283		
	0,1	0,3496	0,3465		
64	20	0,0947	0,0870	$\pi/64$	6
	10	0,2327	0,2282		
	1	0,3620	0,3585		
	0,1	0,3892	0,3716		

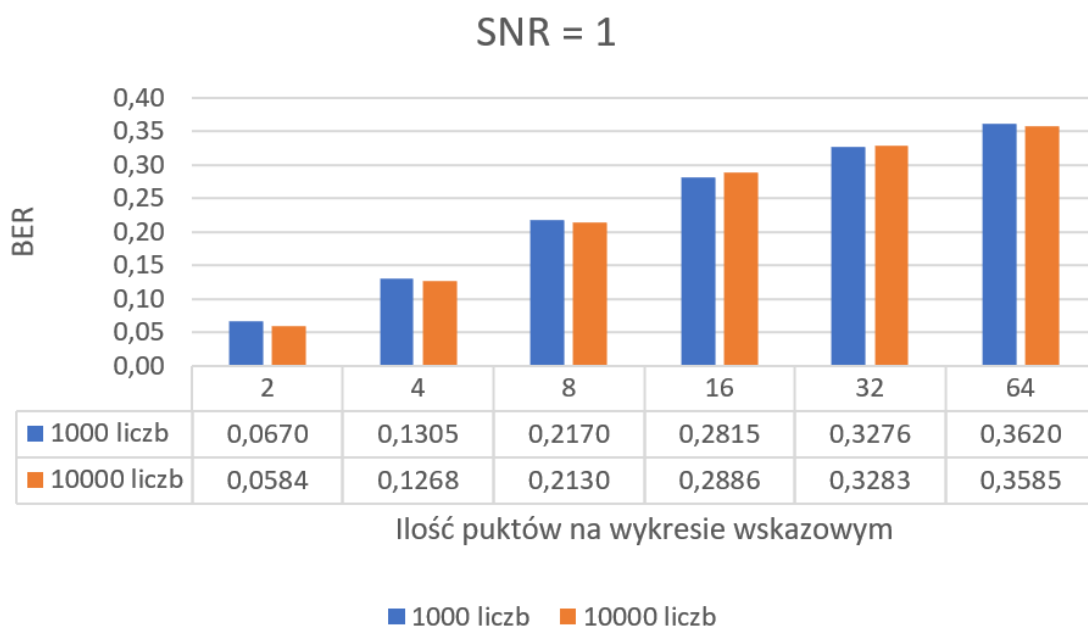
Rysunek 7: Wyniki dla modulacji PSK



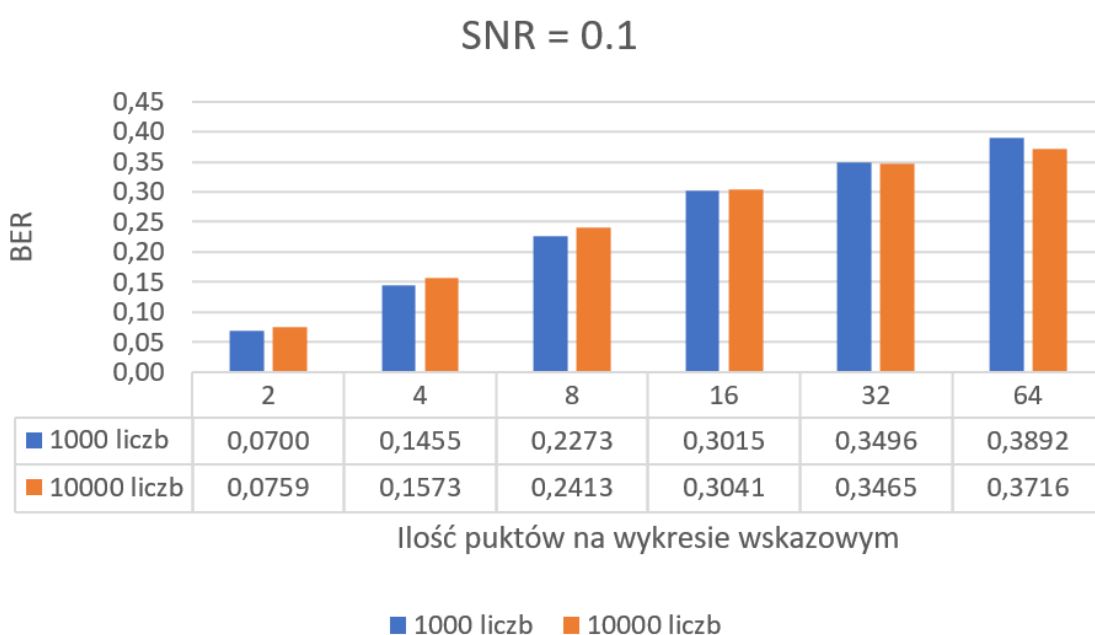
Rysunek 8: Wykres kolumnowy dla modulacji PSK - SNR = 20



Rysunek 9: Wykres kolumnowy dla modulacji PSK - SNR = 10



Rysunek 10: Wykres kolumnowy dla modulacji PSK - SNR = 1



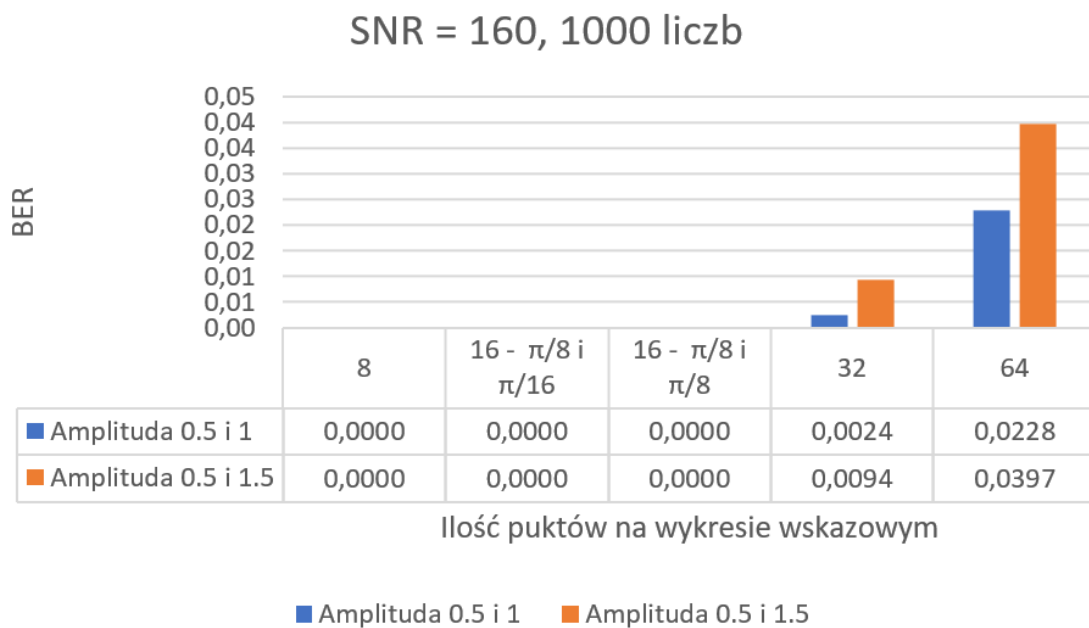
Rysunek 11: Wykres kolumnowy dla modulacji PSK - SNR = 0.1

3.2 Wyniki - APSK

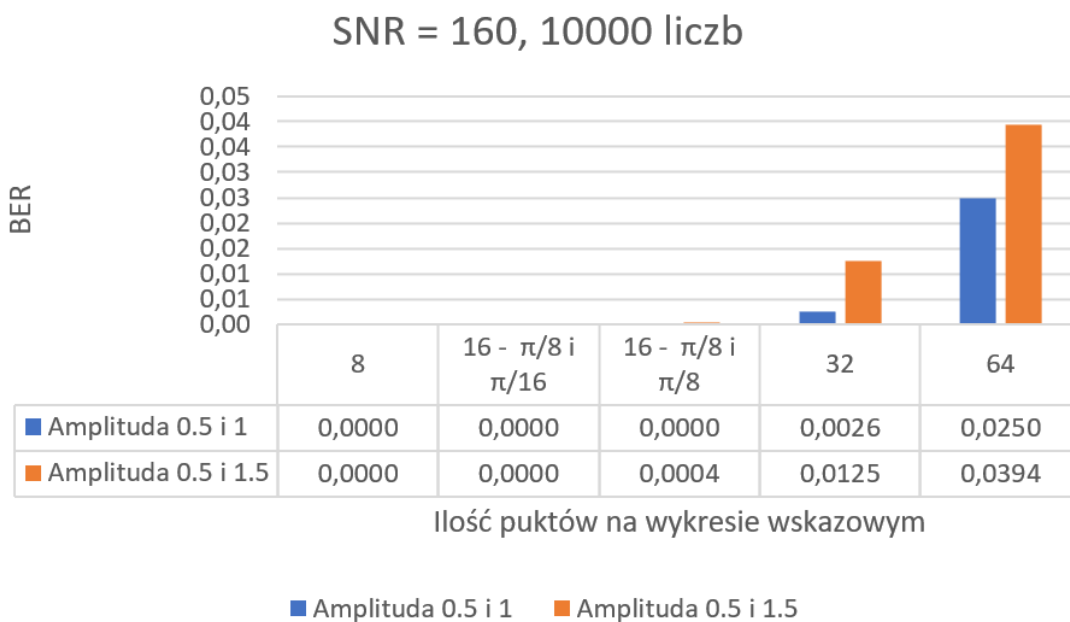
Poniżej umieszczamy wyniki w postaci tabeli oraz wykresów kolumnowych. Każdy wykres odpowiada innemu poziomowi SNR.

APSK								
M	ilość punktów w konstelacji	SNR	BER				Przesunięcie fazowe	Szybkość transmisji (b/T)
			Amplitudy 0.5 i 1		Amplitudy 0.5 i 1.5			
			1000	10000	1000	10000		
4	8	160	0,0000	0,0000	0,0000	0,0000	$\pi/4$ i $\pi/4$	3
		80	0,0000	0,0000	0,0000	0,0000		
		40	0,0007	0,0010	0,0000	0,0000		
		20	0,0090	0,0085	0,0007	0,0008		
		10	0,0353	0,0369	0,0103	0,0087		
		1	0,2473	0,2526	0,1783	0,1762		
		0,1	0,4250	0,4225	0,1147	0,3974		
8	16	160	0,0000	0,0000	0,0000	0,0000	$\pi/8$ i $\pi/16$	4
		80	0,0003	0,0004	0,0005	0,0003		
		40	0,0073	0,0053	0,0040	0,0039		
		20	0,0290	0,0301	0,0190	0,0169		
		10	0,0750	0,0820	0,0440	0,0372		
		1	0,3165	0,3126	0,2530	0,2470		
		0,1	0,4338	0,4435	0,4310	0,4190		
8	16	160	0,0000	0,0000	0,0000	0,0004	$\pi/8$ i $\pi/8$	4
		80	0,0000	0,0002	0,0022	0,0034		
		40	0,0032	0,0037	0,0115	0,0152		
		20	0,0215	0,0226	0,0375	0,0359		
		10	0,0620	0,0611	0,0725	0,0725		
		1	0,2807	0,2903	0,2830	0,2947		
		0,1	0,4390	0,4383	0,4387	0,4456		
16	32	160	0,0024	0,0026	0,0094	0,0125	$\pi/16$ i $\pi/16$	5
		80	0,0114	0,0123	0,0286	0,0283		
		40	0,0306	0,0295	0,0496	0,0474		
		20	0,0622	0,0619	0,0792	0,0808		
		10	0,1150	0,1124	0,1260	0,1248		
		1	0,3226	0,3314	0,3320	0,3324		
		0,1	0,4458	0,4554	0,4654	0,4552		
32	64	160	0,0228	0,0250	0,0397	0,0394	$\pi/32$ i $\pi/32$	6
		80	0,0525	0,0478	0,0698	0,0651		
		40	0,0772	0,0805	0,0988	0,0986		
		20	0,1180	0,1209	0,1445	0,1357		
		10	0,1735	0,1719	0,1792	0,1777		
		1	0,3723	0,3584	0,3527	0,3635		
		0,1	0,4578	0,4602	0,4662	0,4607		

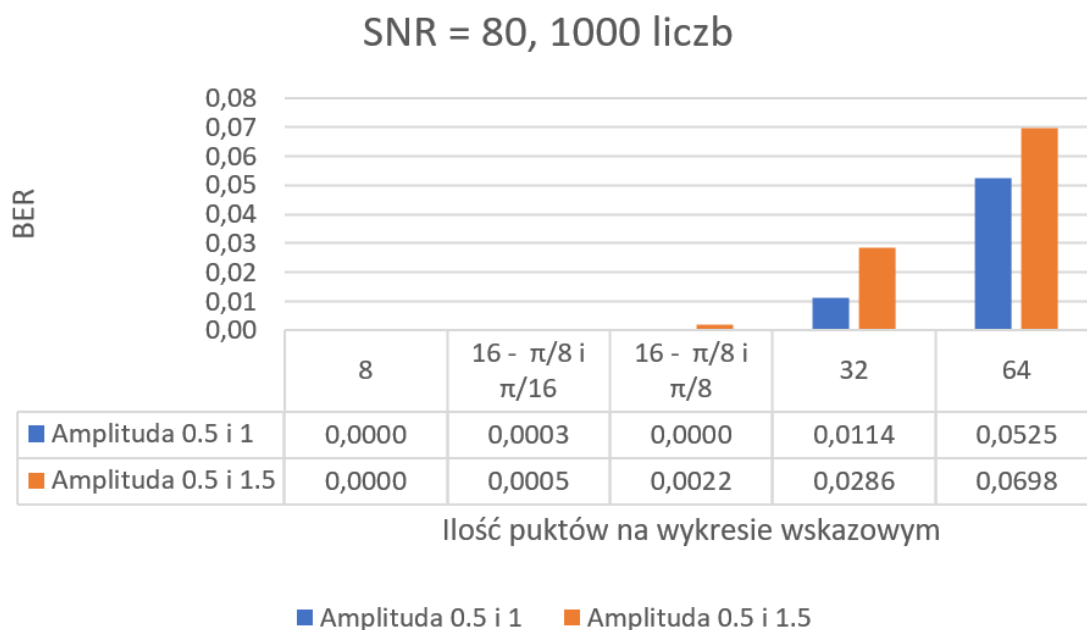
Rysunek 12: Wyniki dla modulacji APSK



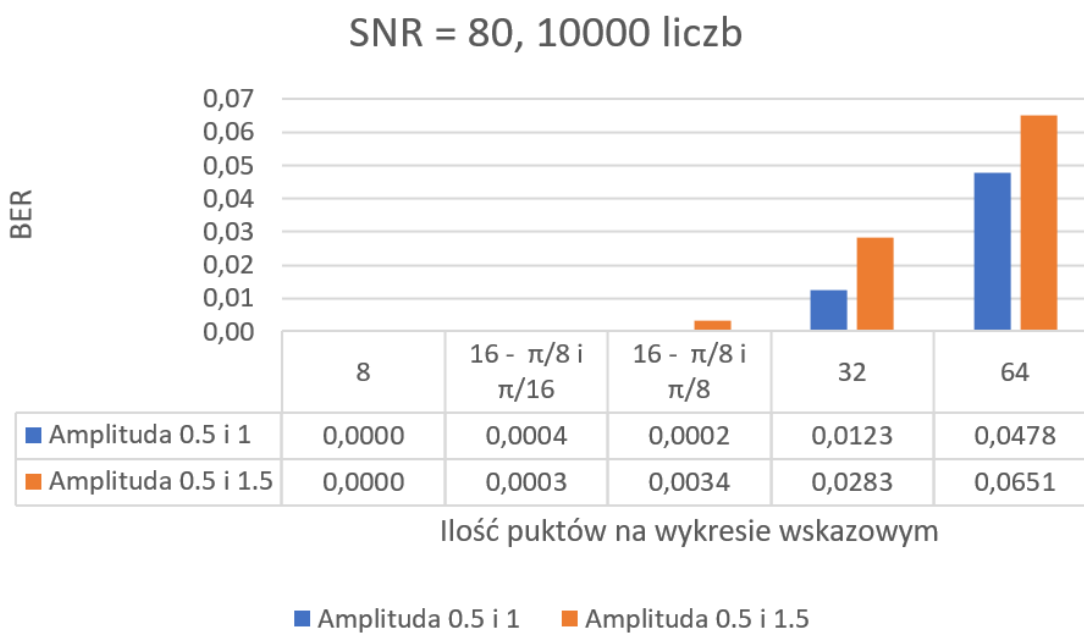
Rysunek 13: Wykres kolumnowy dla modulacji APSK dla 1000 liczb - SNR = 160



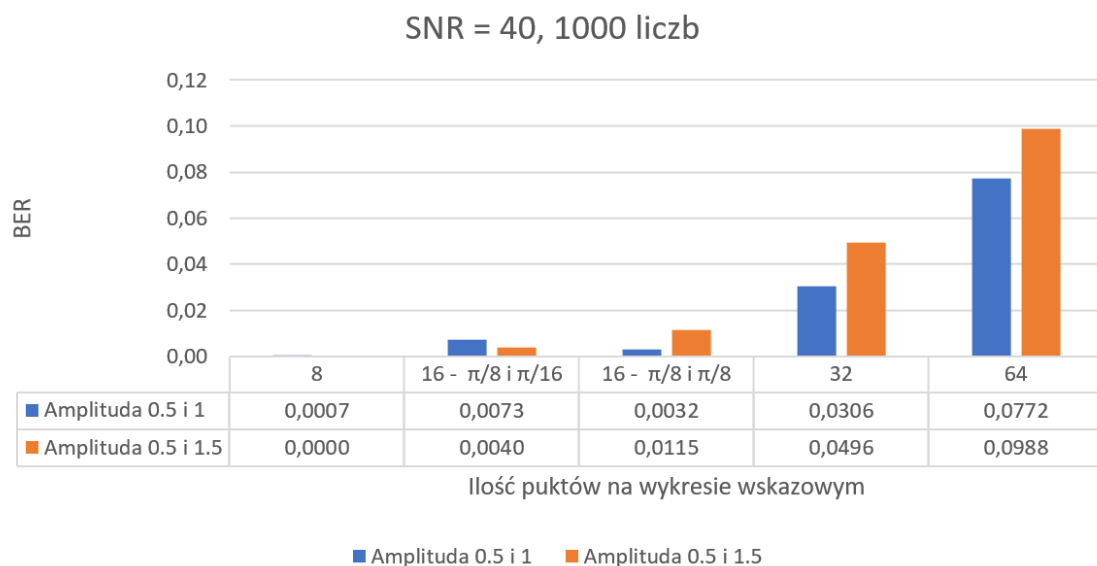
Rysunek 14: Wykres kolumnowy dla modulacji APSK dla 10000 liczb - SNR = 160



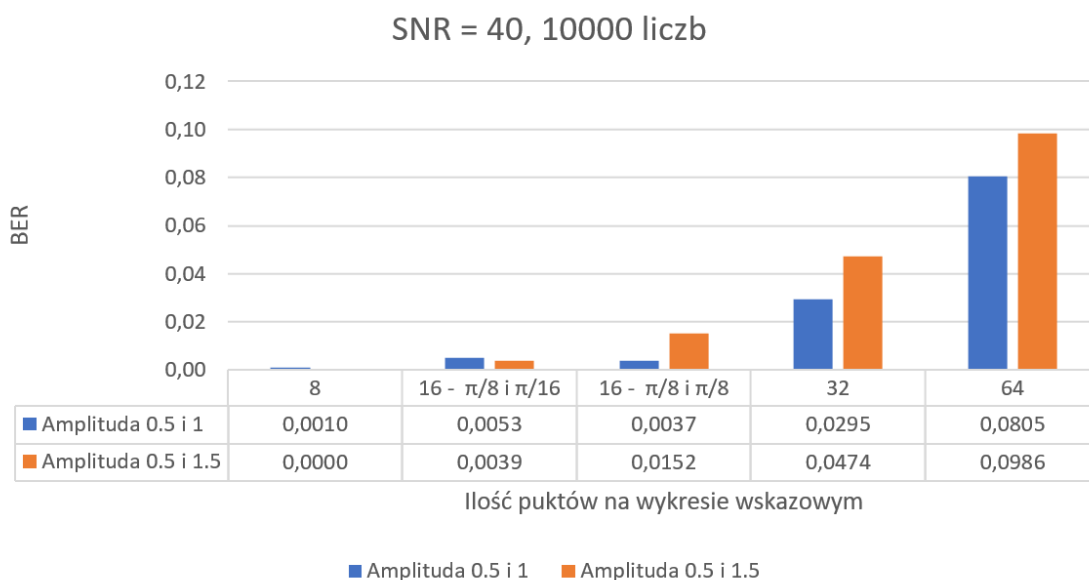
Rysunek 15: Wykres kolumnowy dla modulacji APSK dla 1000 liczb - SNR = 80



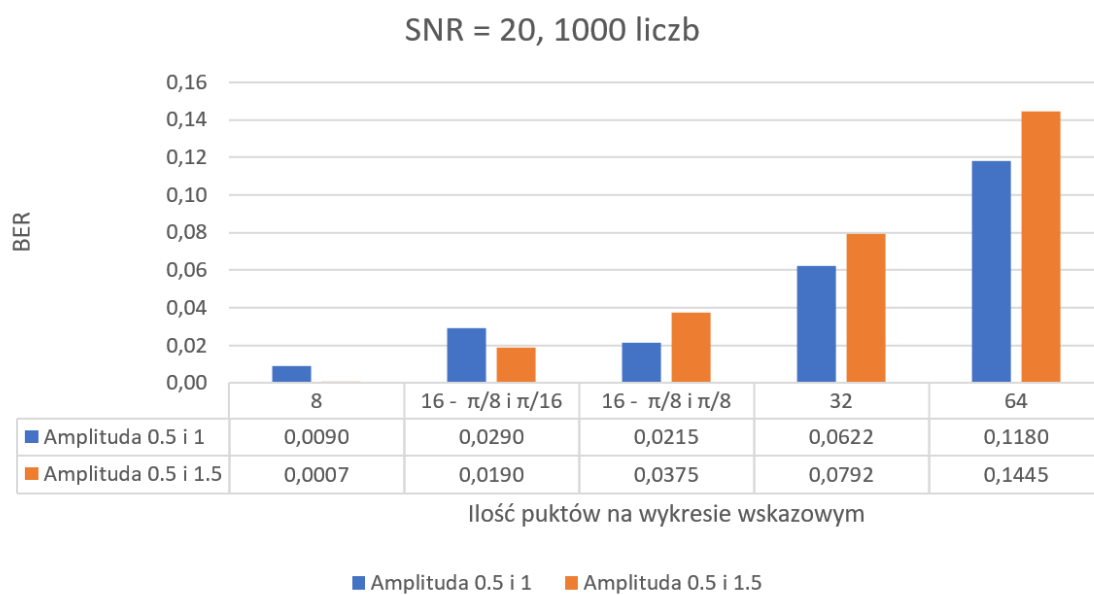
Rysunek 16: Wykres kolumnowy dla modulacji APSK dla 10000 liczb - SNR = 80



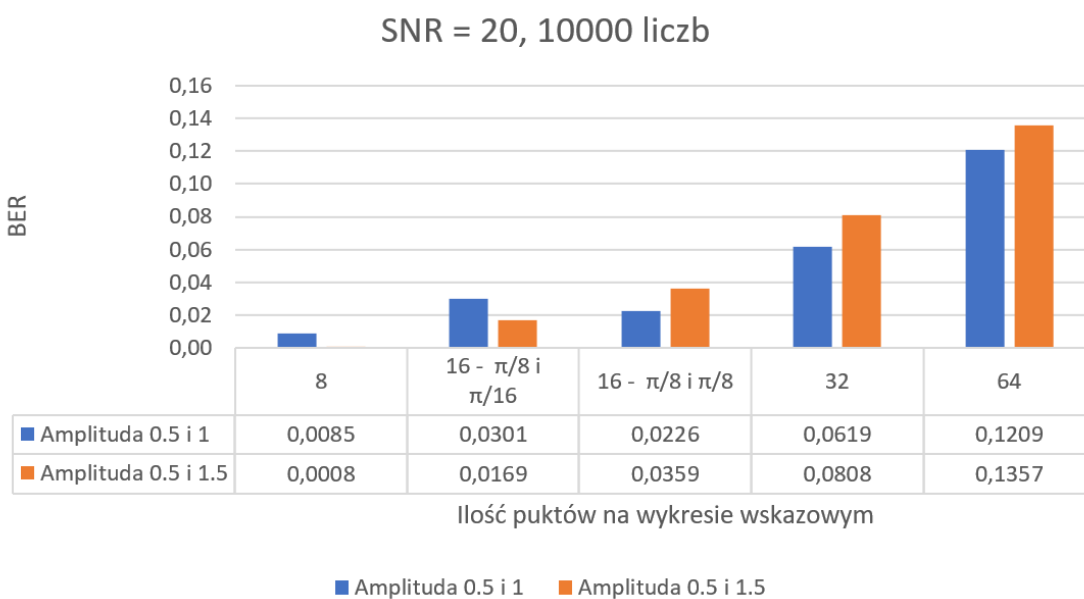
Rysunek 17: Wykres kolumnowy dla modulacji APSK dla 1000 liczb - SNR = 40



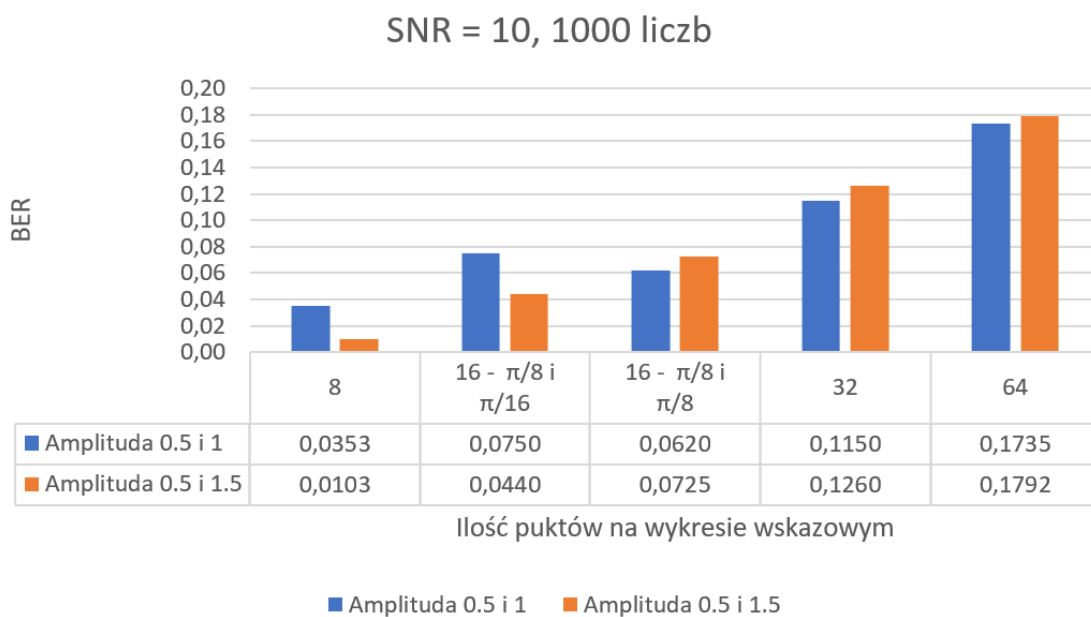
Rysunek 18: Wykres kolumnowy dla modulacji APSK dla 10000 liczb - SNR = 40



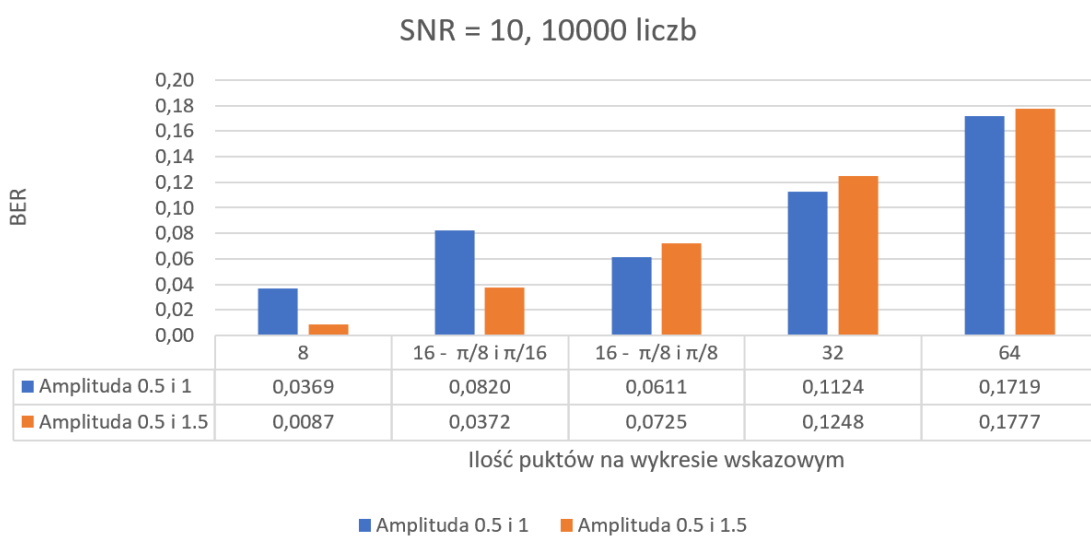
Rysunek 19: Wykres kolumnowy dla modulacji APSK dla 1000 liczb - SNR = 20



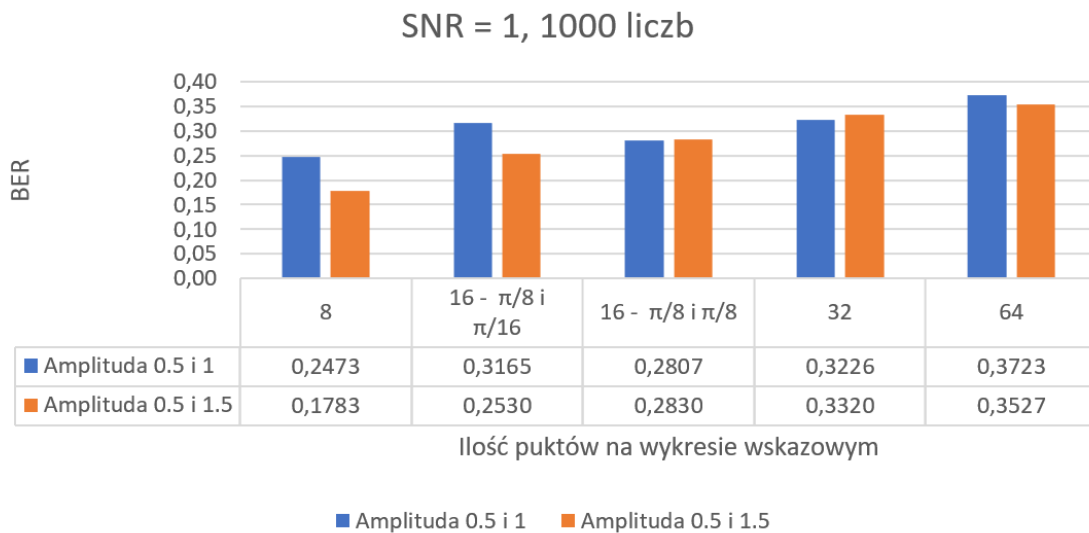
Rysunek 20: Wykres kolumnowy dla modulacji APSK dla 10000 liczb - SNR = 20



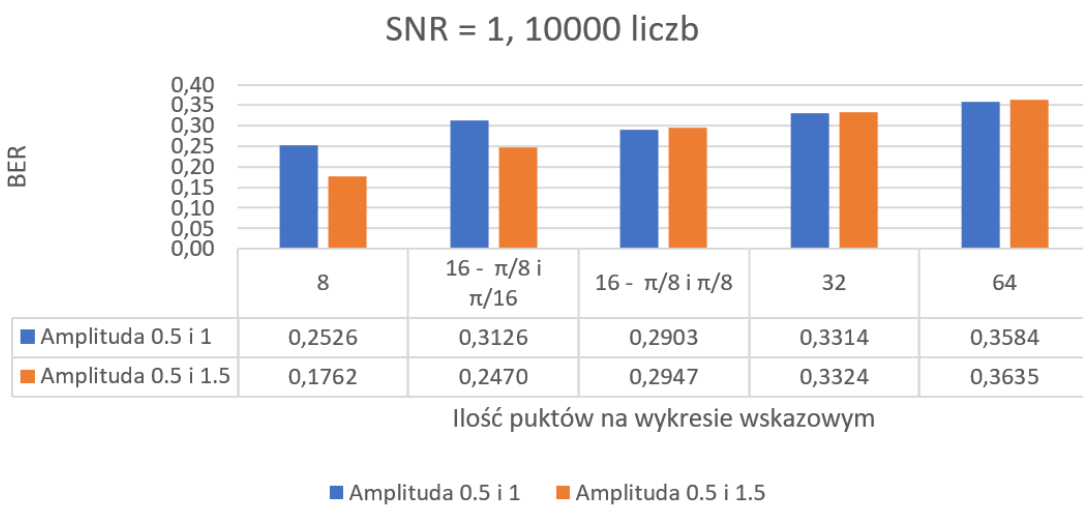
Rysunek 21: Wykres kolumnowy dla modulacji APSK dla 1000 liczb - SNR = 10



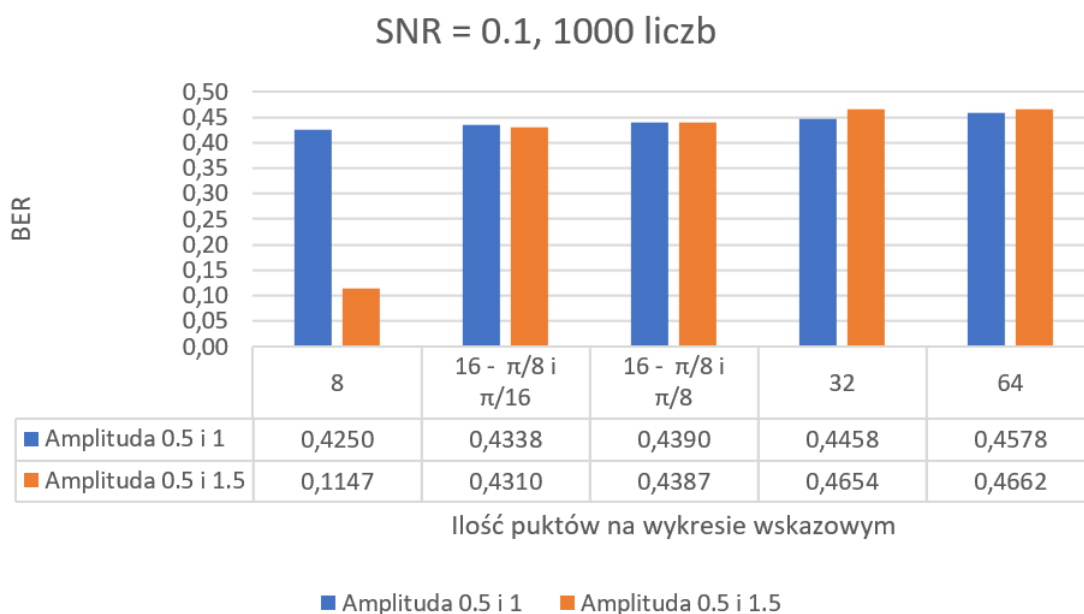
Rysunek 22: Wykres kolumnowy dla modulacji APSK dla 10000 liczb - SNR = 10



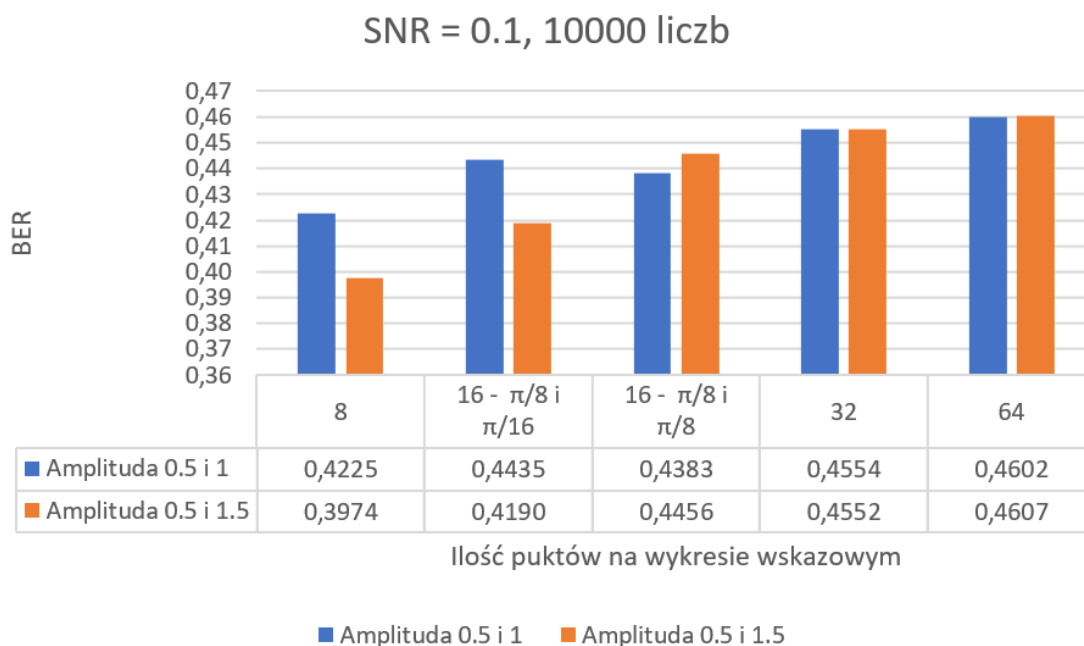
Rysunek 23: Wykres kolumnowy dla modulacji APSK dla 1000 liczb - SNR = 1



Rysunek 24: Wykres kolumnowy dla modulacji APSK dla 10000 liczb - SNR = 1



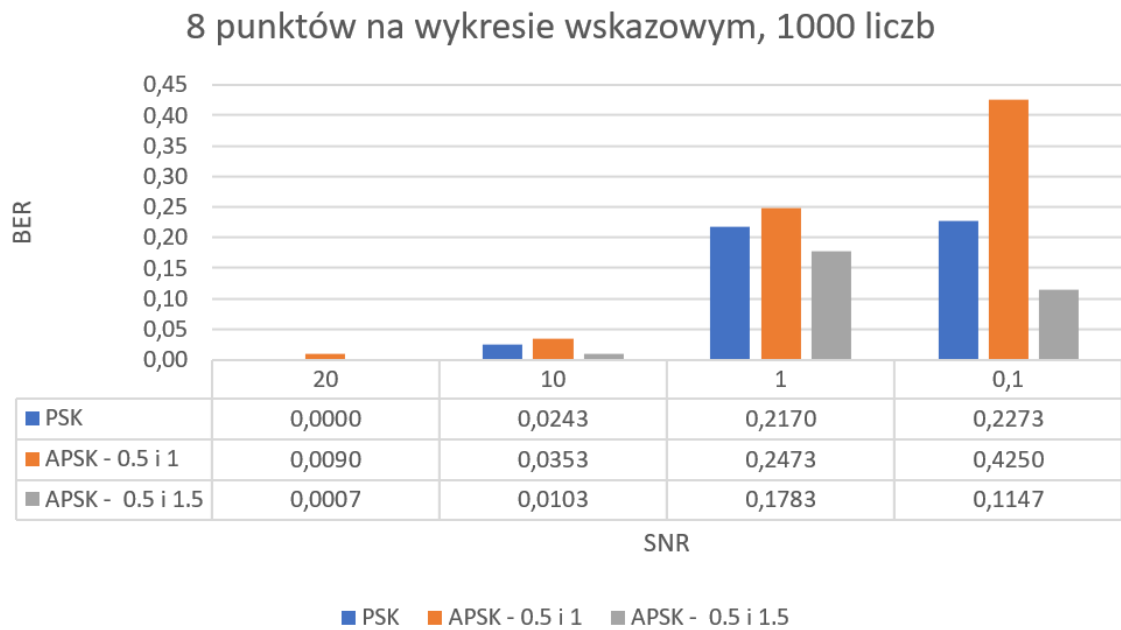
Rysunek 25: Wykres kolumnowy dla modulacji APSK dla 1000 liczb - SNR = 0.1



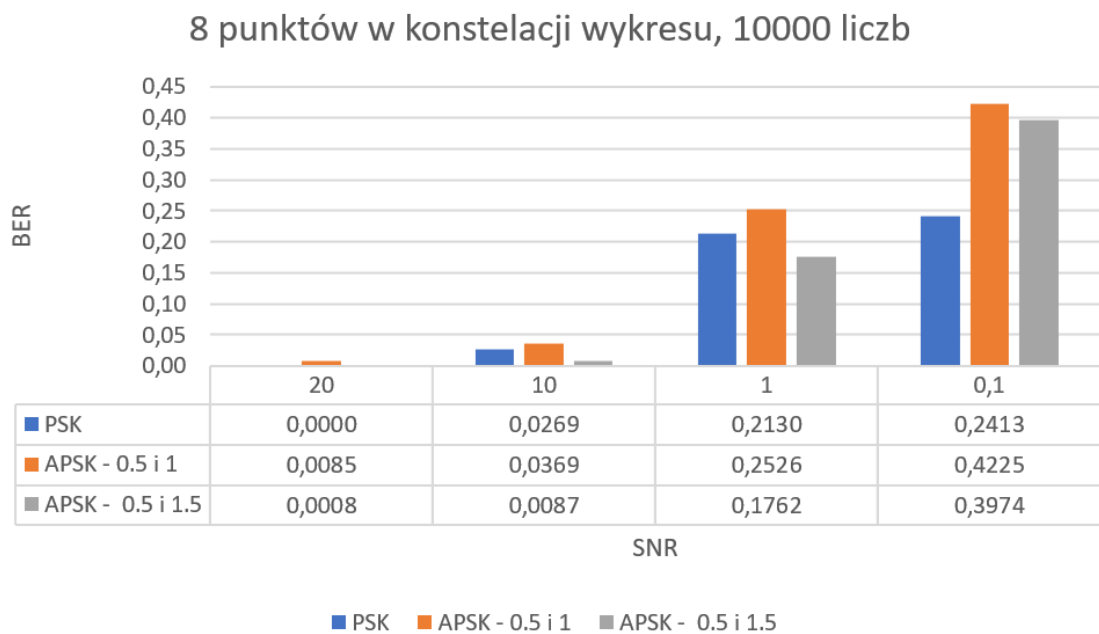
Rysunek 26: Wykres kolumnowy dla modulacji APSK dla 10000 liczb - SNR = 0.1

3.3 Wyniki - APSK vs PSK

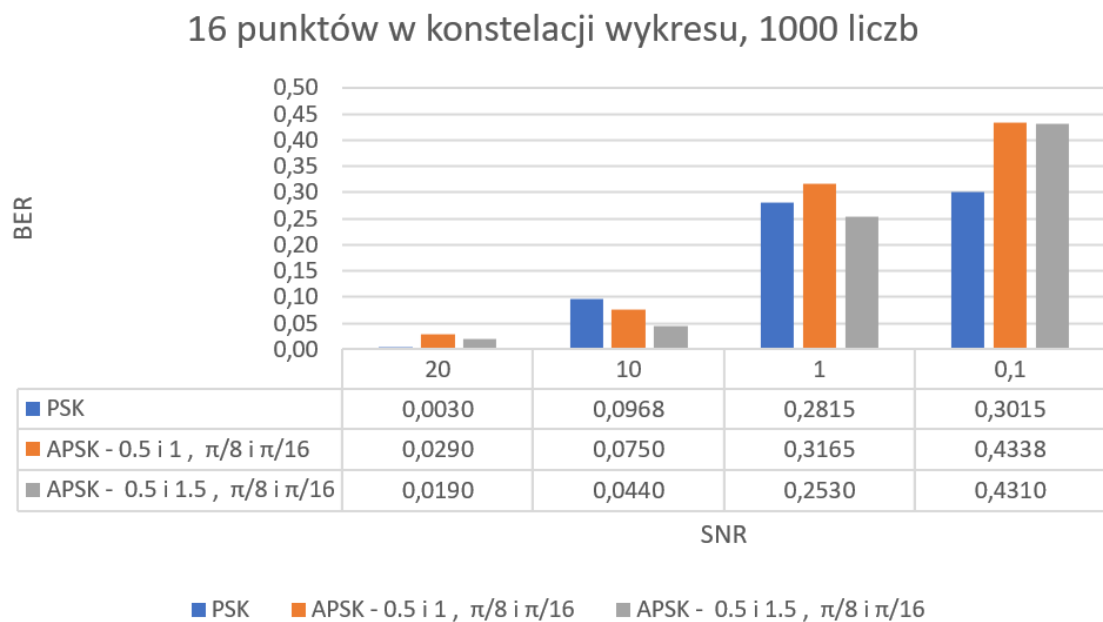
Poniżej umieszczamy wyniki w postaci tabeli oraz wykresów kolumnowych. Każdy wykres odpowiada ilości punktów w konstelacji wykresu wskazowego - czyli ilości bitów przesłanych w trakcie transmisji - i liczby przesłanych liczb.



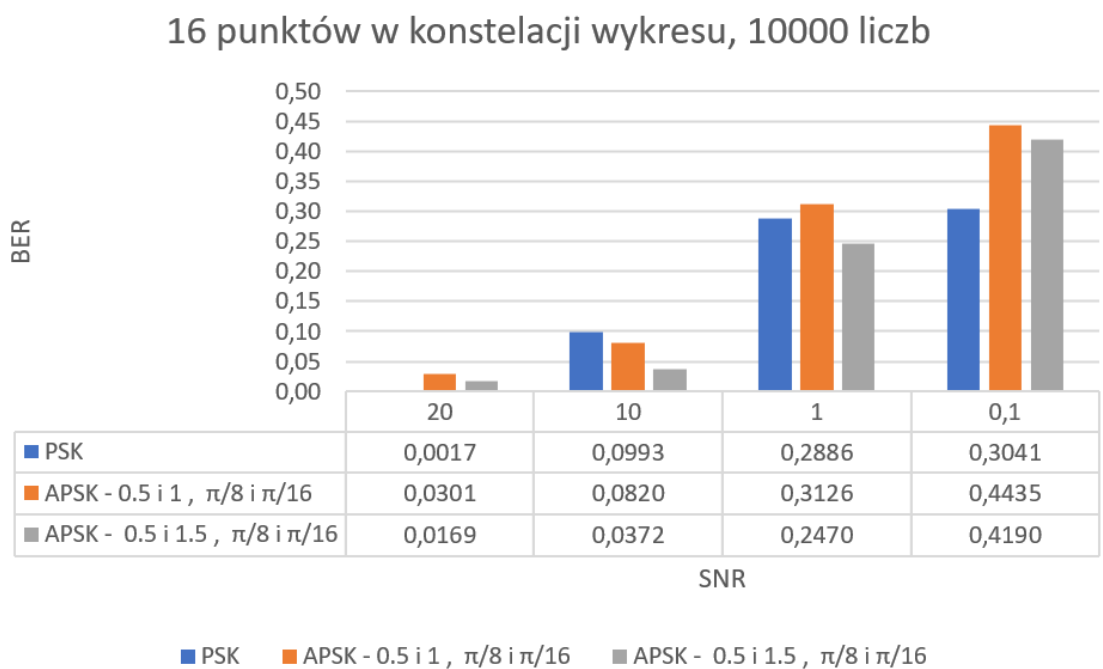
Rysunek 27: APSK vs PSK dla 8 punktów na wykresie i 1000 liczb



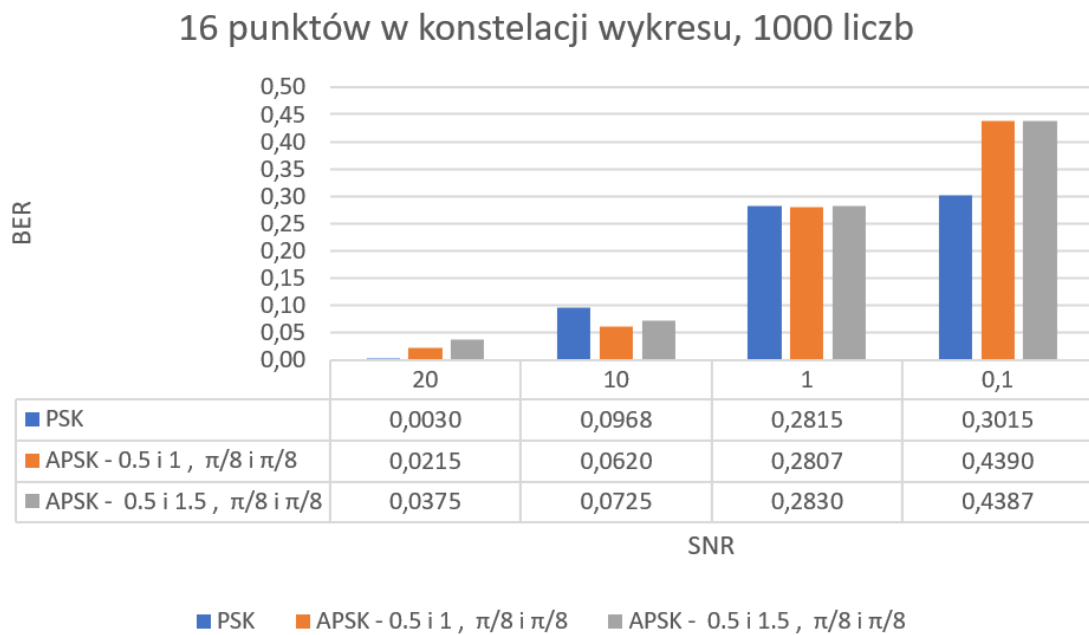
Rysunek 28: APSK vs PSK dla 8 punktów na wykresie i 10000 liczb



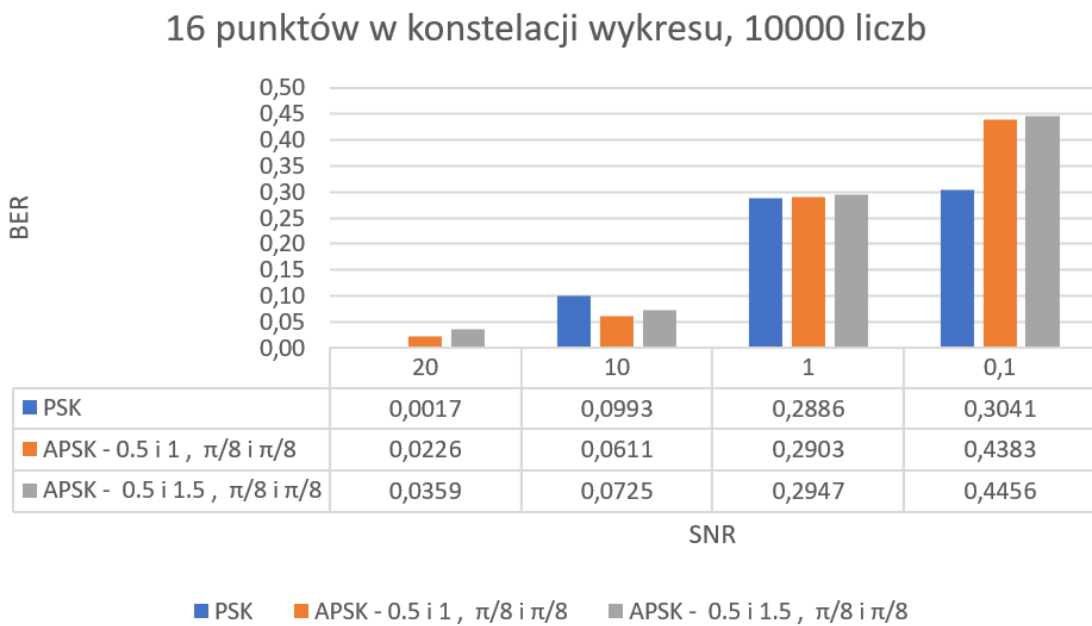
Rysunek 29: APSK vs PSK dla 16 punktów na wykresie, 1000 liczb i różnych przesunięć fazowych dla okręgów na wykresie wskazowym



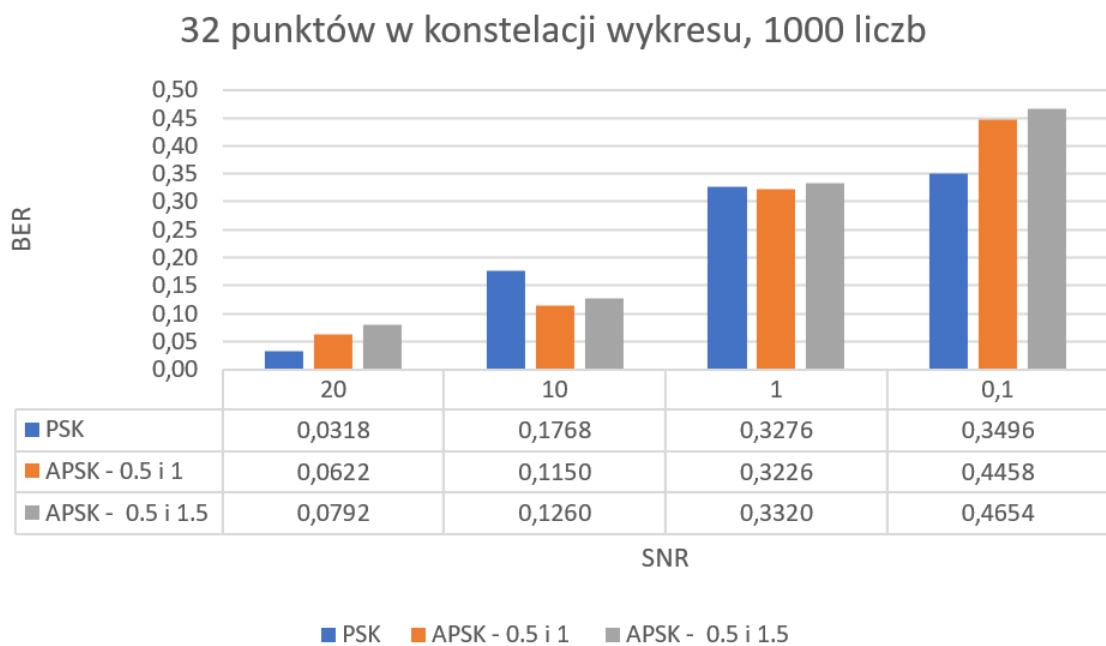
Rysunek 30: APSK vs PSK dla 16 punktów na wykresie, 10000 liczb i różnych przesunięć fazowych dla okręgów na wykresie wskazowym



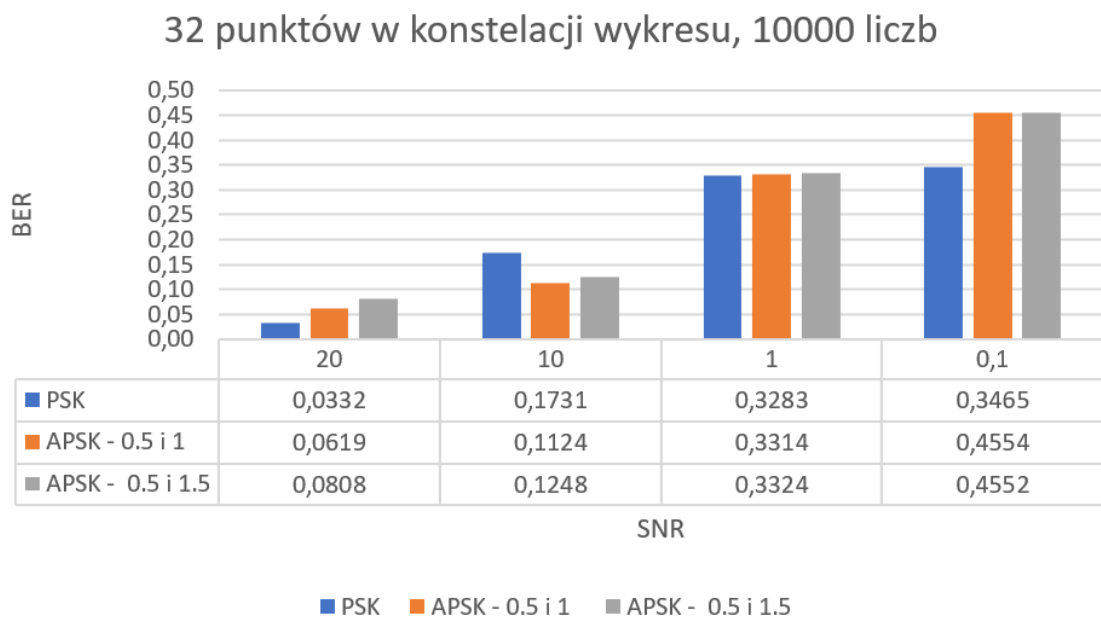
Rysunek 31: APSK vs PSK dla 16 punktów na wykresie, 1000 liczb i równych przesunięć fazowych dla okręgów na wykresie wskazowym



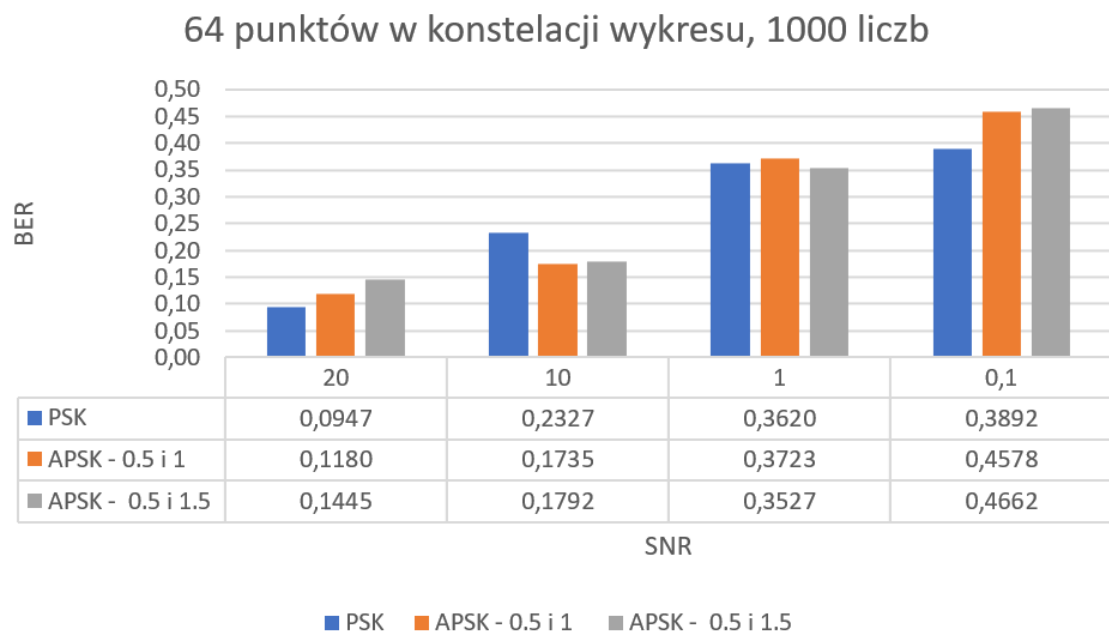
Rysunek 32: APSK vs PSK dla 16 punktów na wykresie, 10000 liczb i równych przesunięć fazowych dla okręgów na wykresie wskazowym



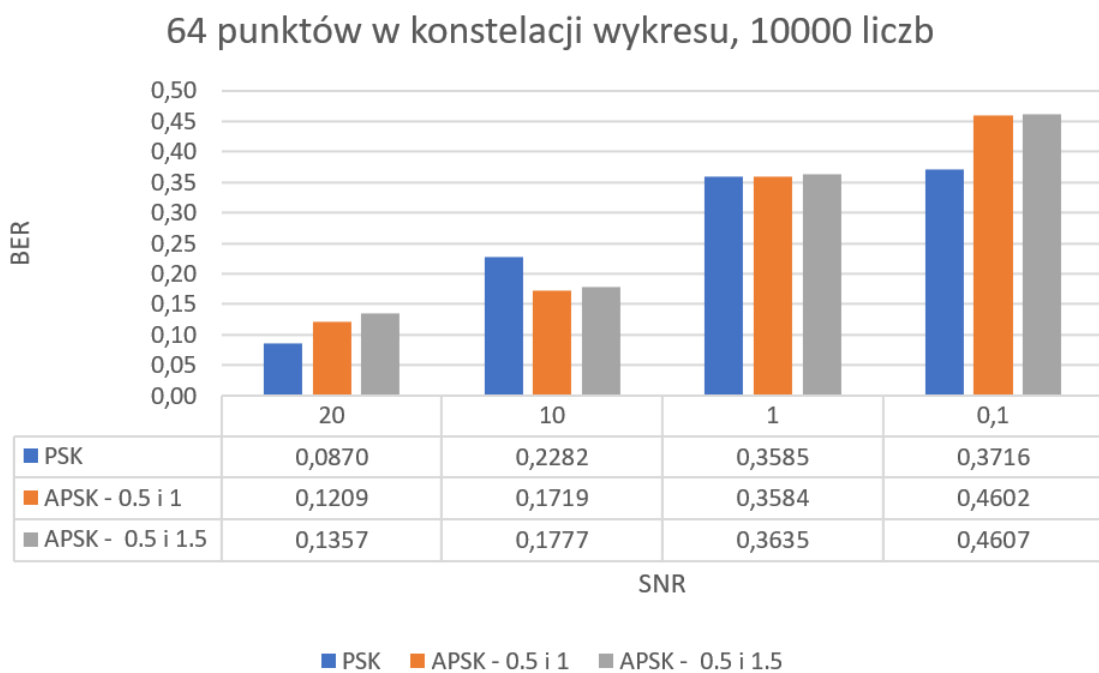
Rysunek 33: APSK vs PSK dla 32 punktów na wykresie i 1000 liczb



Rysunek 34: APSK vs PSK dla 32 punktów na wykresie i 10000 liczb



Rysunek 35: APSK vs PSK dla 64 punktów na wykresie i 1000 liczb



Rysunek 36: APSK vs PSK dla 64 punktów na wykresie i 10000 liczb

4 Analiza wyników

Analizę zaczniemy od modulacji PSK. Na zamieszczonych przez nas wykresach widać dobrze to, czego można było się spodziewać – wraz ze spadkiem wartości stosunku mocy sygnału do mocy szumu (zwiększenie zakłóceń w kanale transmisyjnym) ilość błędnie odebranych bitów, a co za tym idzie współczynnik BER, rośnie. Można to zauważyć dla każdej ilości bitów przesyłanych w jednym okresie fali nośnej czy dla różnych ilości przesyłanych bitów ogółem. Przykładem popierającym to zjawisko może być wartość dla modulacji BSK (1000 liczb), dla której wartość współczynnika BER rośnie, zaczynając od 0 poprzez 0, 0.067 aż do 0.07 wraz ze spadkiem wartości SNR, co widać na rysunkach 8 – 11. Należy też zauważyć, że wartości dla tych samych poziomów SNR, ale różnych ilości liczb są bardzo podobne, a różnice można uznać za błąd pomiarowy. Ważnym spostrzeżeniem jest to, że mniejsza szybkość transmisji zapewnia większą dokładność odbioru, tak więc najlepiej w tym przypadku wypada modulacja BSK.

Podobnie jak w przypadku modulacji PSK, tak i gdy mówimy o modulacji APSK widać to, co oczywiste – wraz ze spadkiem poziomu SNR ilość błędnie odebranych bitów rośnie. Na większą uwagę zasługuje fakt dość nietypowy. Mianowicie wartości BER dla amplitud 0.5 i 1.5 przyjmują regularnie wyższe wartości niż dla amplitud 0.5 i 1 – niejednokrotnie są to wartości kilkukrotnie wyższe. Jednak wraz ze spadkiem wartości SNR na kolejnych wykresach różnica ta coraz bardziej maleje, osiągając wartości błędu pomiarowego i tylko dla modulacji 8 punktowej przy małych wartościach SNR utrzymywana jest znacząca różnica współczynnika BER. Oczekiwane rezultaty widać dopiero w przypadku modulacji 16 punktowej, gdzie przesunięcie fazy ustawiliśmy na różne wartości dla okręgu wewnętrznego i zewnętrznego, odpowiednio $\pi/8$ i $\pi/16$. W tym przypadku widać, że niemal dla każdej wartości SNR poziom błędów dla amplitud 0.5 i 1 jest wyższy niż ten dla amplitud 0.5 i 1.5, a dla niektórych jest nawet znacznie wyższy. Wspomniane wcześniej przesunięcie fazowe wypada też lepiej w porównaniu z przesunięciem $\pi/8$ i $\pi/8$ dla amplitudy 0.5 i 1.5. Tej zależności nie można jednak zauważyć w przypadku amplitud 0.5 i 1, porównując wspomniane przesunięcia fazowe, co więcej sytuacja wydaje się niemal przeciwna. Następną obserwacją jest to, że dla małych wartości SNR – czyli dużych zakłóceń w kanale – poziom błędów BER dla wszystkich szybkości transmisji przyjmuje bardzo podobną wartość, jednakże dla wartości największych przedstawionych w naszym zestawieniu różnice te są bardzo duże.

Przejdziemy teraz do porównania modulacji PSK i APSK. Jak widać na rysunkach 27, 28 dla szybkości trzech bitów przesyłanych w jednym okresie fali nośnej – 8 punktów w konstelacji wykresu wskazowego – najmniejszym poziomem błędów BER może się pochwalić modulacja APSK dla amplitud 0.5 i 1.5, z wyjątkiem wartości SNR 0.1 dla 10000 liczb. Jeśli zbadamy wartości dla 16 punktów na wykresie i przesunięć fazowych $\pi/8$ i $\pi/16$ zobaczymy, że przewaga APSK dla amplitud 0.5 i 1.5 jest obecna dla wszystkich wartości oprócz SNR = 0.1 – zarówno dla 1000 jak i 10000 liczb. Sytuacja zmienia się jeszcze bardziej dla 16 punktów na wykresie, gdzie dla modulacji APSK przesunięcia fazowe są równe $\pi/8$ i $\pi/8$, tutaj dla wyższych wartości SNR (20 i 10) „wygrywa” modulacja APSK z amplitudami 0.5 i 1, dla SNR = 1 różnice zawierają się w błędzie pomiarowym, a dla SNR = 0.1 najmniejszą wartość przyjmuje modulacja PSK – ponownie dla 1000 jak i 10000 liczb. Dla SNR = 1 utrzymują się równe wartości BER dla reszty wykresów (33 – 36), natomiast jeśli

chodzi o resztę wartości SNR przewagę ma modulacja PSK, która przyjmuje najniższe wartości dla $\text{SNR} = 10$ i 0.1 , a ustępuje tylko modulacji APSK dla amplitud 0.5 i 1 , gdy wartość SNR wynosi 10 .

5 Wnioski

Najbardziej oczywisty wniosek nasuwający się po analizie wyników mówi nam, że zapewnienie mniejszych zakłóceń w kanale zapewnia mniej pomyłek przy odbiorze bitów. Z analizy wynika także, że najdokładniejszą modulacją - jeśli chodzi o PSK - okazuje się najwolniejsza, czyli modulacja BSK. Sytuacja ma się podobnie w przypadku modulacji APSK, gdzie transmisja o prędkości 3 bitów na okres (najwolniejsza) wypada najlepiej. Analiza pokazała, że różne przesunięcia fazowe dla wewnętrznego i zewnętrznego okręgu dla prędkości transmisji 4 b/T i dla amplitud 0.5 i 1.5 zwiększa dokładność przesyłu. Z naszych badań wychodzi także inny wniosek. Mianowicie modulacja APSK dla amplitud 0.5 i 1 jest bardziej skuteczna niż dla amplitud 0.5 i 1.5 dla wysokiego poziomu SNR, a gdy poziom jest coraz niższy obie te modulacje okazują się tak samo skuteczne. Z porównania modulacji PSK i APSK wynika, że w większości przypadków modulacja APSK wypada lepiej, jednak wraz ze wzrostem ilości punktów na wykresie coraz częściej zaczyna wygrywać modulacja PSK, przez co na podstawie naszych badań nie można jednoznacznie określić, która z tych modulacji jest lepsza.

Gdy spojrzymy na założenia projektowe zobaczymy, że ilość zagadnień opracowanych przez nas w trakcie pracy nad projektem jest znacznie większa niż to co sobie założyliśmy, co pokazuje ogrom wiedzy zdobytej podczas pracy nad projektem oraz ilość włożonej pracy. Możemy śmiało powiedzieć, że wywiązaliśmy się z założeń projektowych, a prace wykonaliśmy rzetelnie.

Bibliografia

- [1] Materiały pomocnicze ze strony środowiska MATLAB
- [2] Richard G. Lyons "Wprowadzenie do Cyfrowego Przetwarzania Sygnałów", 2010
- [3] Materiały pomocnicze do projektu ze strony Zakładu Systemów Komputerowych i Dyskretnych Politechniki Wrocławskiej