



بسمه تعالیٰ

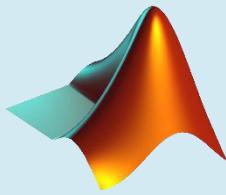
گام‌های تکمیل آزمایشگاه مخابرات دیجیتال

استاد: دکتر علی الفت

ویرایش ۱/۳ (آزمایشی)

فهرست مطالب آزمایشگاه مخابرات دیجیتال

۳	آزمایش اول: آشنایی با نرم افزار MATLAB
۷	آزمایش دوم: پردازش سیگنال دیجیتال با نرم افزار MATLAB
۱۶	آزمایش سوم: رادیونرم افزار و پایش طیف سیگنال های رادیویی
۱۸	آزمایش چهارم: معرفی مدولاسیون دیجیتال خطی
۲۱	آزمایش پنجم: کاوش در مدولاسیون دیجیتال خطی
۳۲	آزمایش ششم: مدولاسیون FSK همدوس
۴۲	آزمایش هفتم: آشکارسازی ناهمدوس
۴۶	آزمایش هشتم: انتقال دیجیتال از درون کانال باند محدود AWGN



آزمایش اول

آشنایی با نرم افزار MATLAB

شرح آزمایش

آزمایش ۱-۱: محاسبات جبری و گرافیک

۱. مقادیر زیر را محاسبه نماید.
۲.

تقریب کسرها:

2.645507812500000
2.645750000000000
2.645751633986928

عدد $\sqrt{7}$:

2.645751311064591

کمینه‌ی فاصله‌ی کسرها تا عدد $\sqrt{7}$:

3.229223373146795e-07

اندیس عدد با کمترین فاصله:

۳

- ۳^{۳۰۱} را یک بار به صورت یک عدد اعشاری تقریبی با ۱۵ رقم اعشار (نمایش به صورت نماد علمی) و بار دیگر به صورت یک عدد صحیح دقیق محاسبه نمایید. (راهنمایی: از دستور `format` و دستور `vpa` با ورودی سمبولیک استفاده نمایید).

نمایش به صورت نماد علمی:

4.106744371757652e+143

نمایش به صورت عدد صحیح دقیق:

410674437175765127973978082146264947899391086876012309414440570235106991532
497229781400618467066824164751453321793982128440538198297087323698003

۸.۸۸۱۷۸۴۱۹۷۰۰۱۲۵۲e-16 ت.
۰ ث.

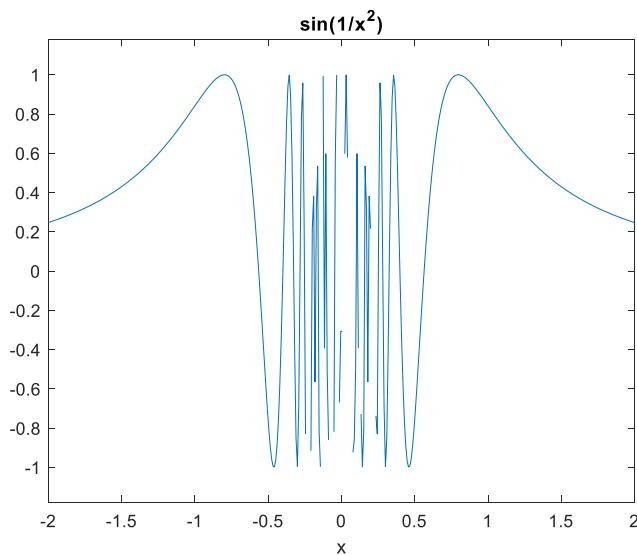
.۲

۱.۰۰۵۰۰۴۱۶۸۰۵۵۸۰۴ أ.
۰.۶۹۳۱۴۷۱۸۰۵۵۹۹۴۵ ب.
۰.۴۶۳۶۴۷۶۰۹۰۰۰۸۰۶ ت.

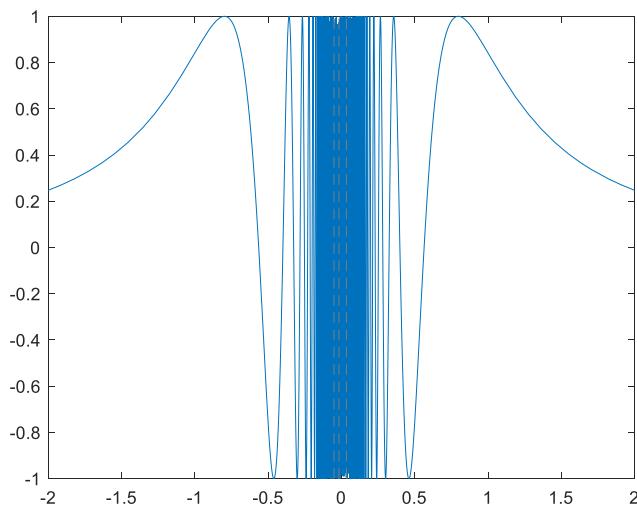
۳

۱۰

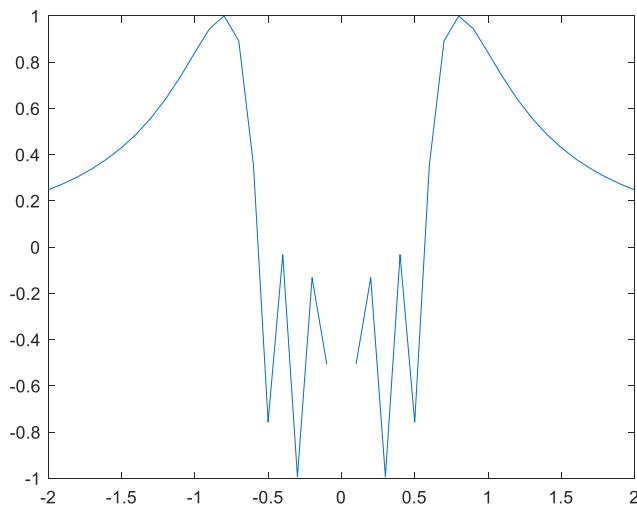
استفاده از دستور :ezplot



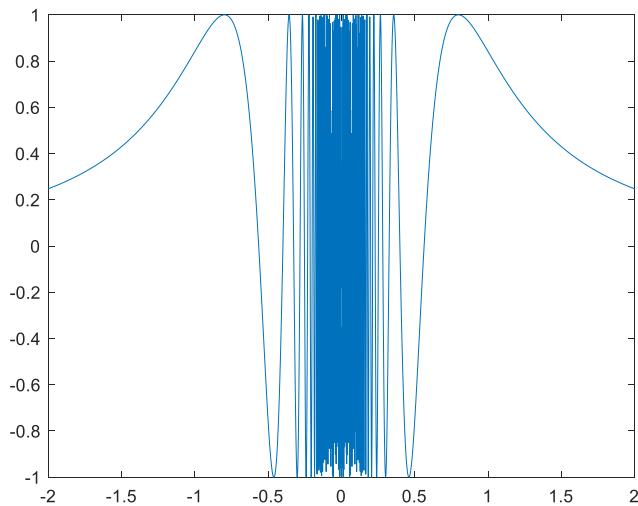
استفاده از دستور :fplot



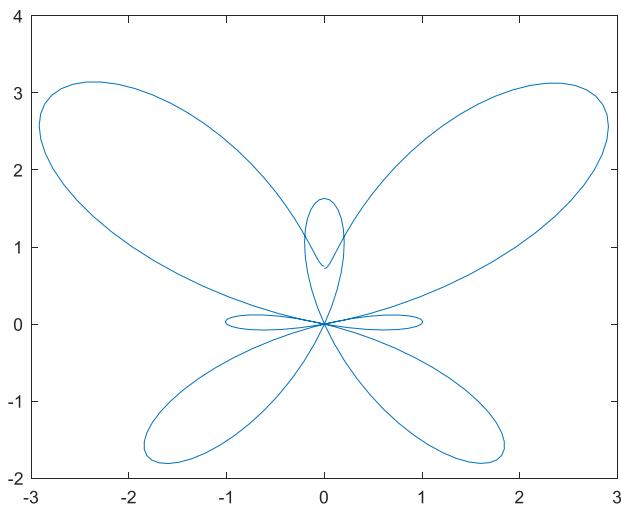
استفاده از دستور plot با گام 0.1



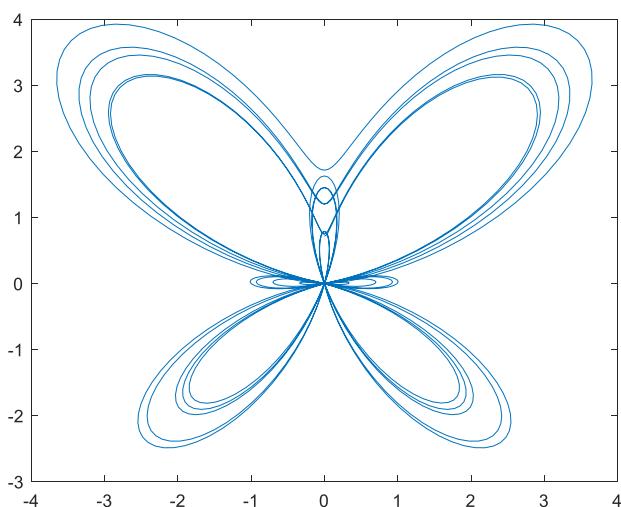
استفاده از دستور `plot` با گام 0.001 :



منحنی پروانه را با استفاده از معادله‌های پارامتری زیر رسم نمایید.
منحنی پروانه برای $0 \leq t \leq 2\pi$.



منحنی پروانه برای $0 \leq t \leq 10\pi$



آزمایش ۲-۱: ریاضیات و برنامه‌نویسی

۱

۰	.ا.
۱	.ب.
۱	.ت.
۰	.ث.

۲

۱	۰	۰	۰	۰	۰	۰
۱	۱	۰	۰	۰	۰	۰
۱	۲	۱	۰	۰	۰	۰
۱	۳	۳	۱	۰	۰	۰
۱	۴	۶	۴	۱	۰	۰
۱	۵	۱۰	۱۰	۵	۱	۰
۱	۶	۱۵	۲۰	۱۵	۶	۱

۲.۳۷۶۹ .۳

۴. اگر اول کد خود بنویسید (1) mg خروجی شما به صورت زیر خواهد شد.

18 26 10 16 30 16 24 28 18 28 18 24 14 28 10 24 18 10 12 14

The Vector Generated After 3 Iteration(s).

۵

خروجی برای 60 :mylcm(4, 5, 6)

خروجی برای 420 :mylcm([6 7 12 15])

خروجی برای :mylcm(4.5, 6)

```
Error using mylcm (line  )
Arguments must be positive integers.
```

آزمایش دوم

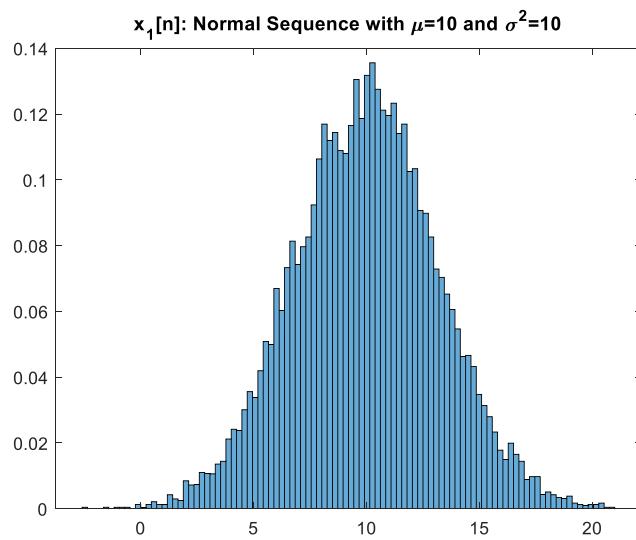


پردازش سیگنال دیجیتال با نرم افزار MATLAB

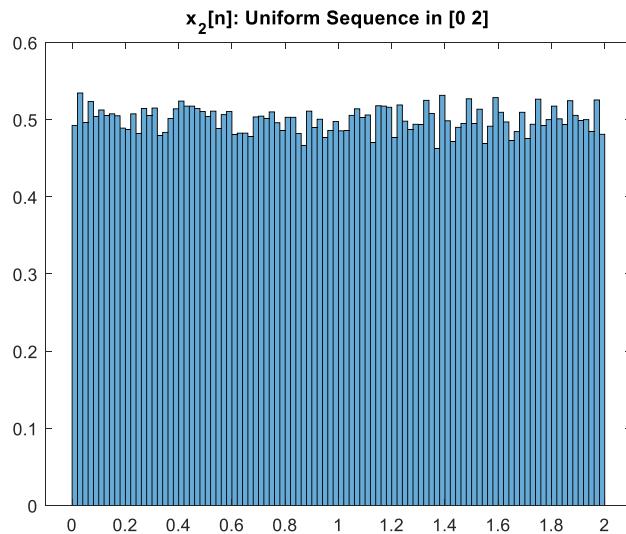


آزمایش ۱-۲: تولید سیگنال دیجیتال و عملیات بر روی آن

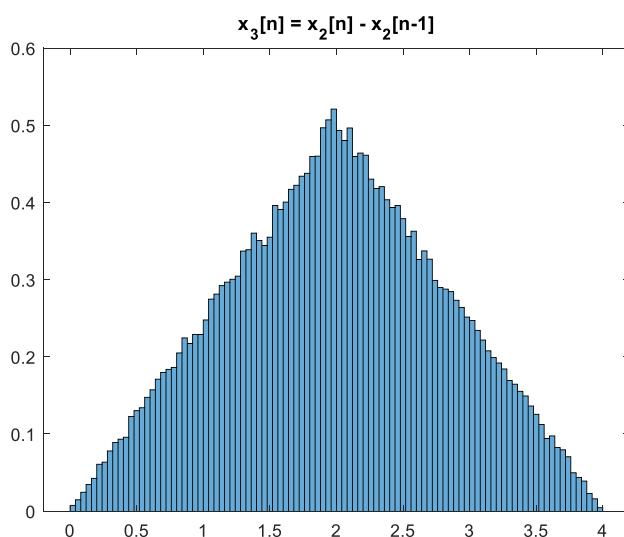
۱. دنباله های تصادفی: در ابتدای تولید همه بخش ها rng(1) قرار دهید تا نمودارها یکسان باشد.



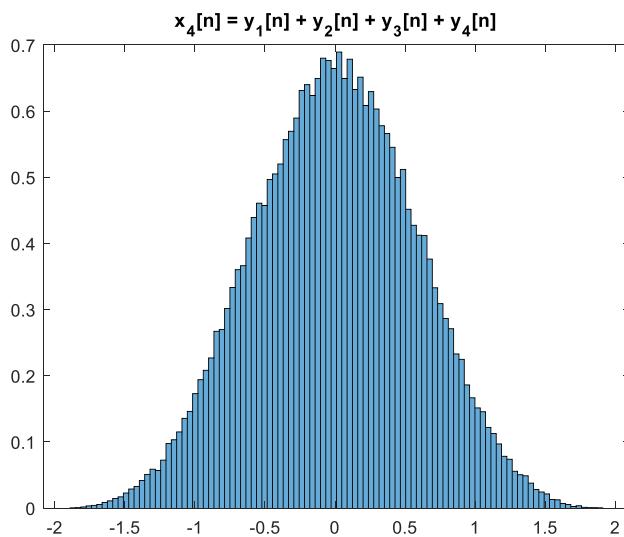
.ا.



.ب.



ت.

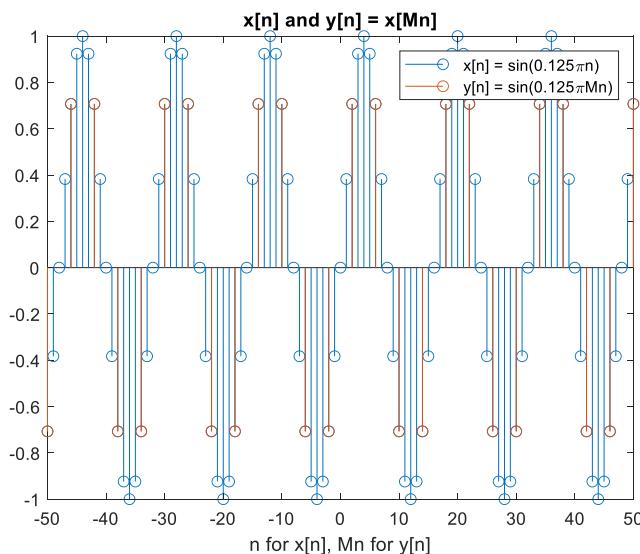


۲. کاهش نرخ نمونه برداری:

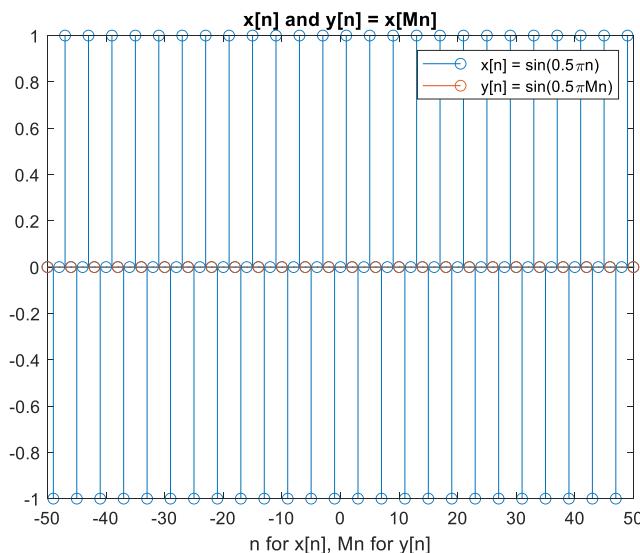
اً. خروجی برای مثال صورت سوال:

$y = 4 \times 1$
-2
3
5
8

ب.



عملیات فوق را برای تابع $x[n] = \sin[0.5\pi n]$ و در بازه $-50 \leq n \leq 50$ تکرار نمایید.



۳. افزایش نرخ نمونهبرداری:

أ.

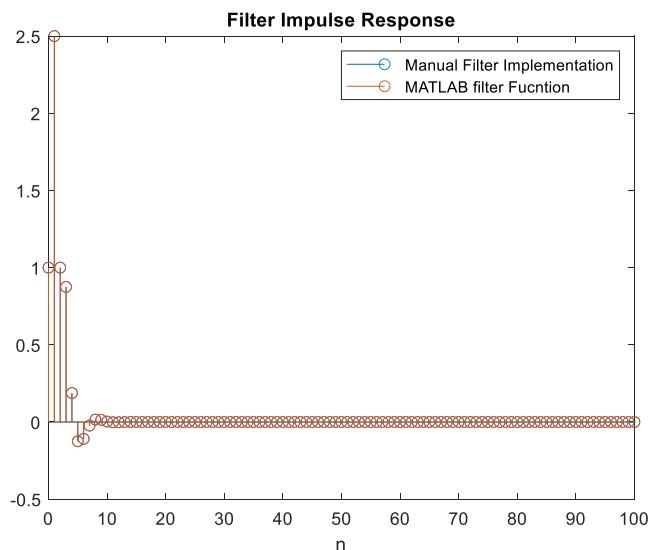
ب.

```
y = 7×1
-2
0
3
0
5
0
8
```

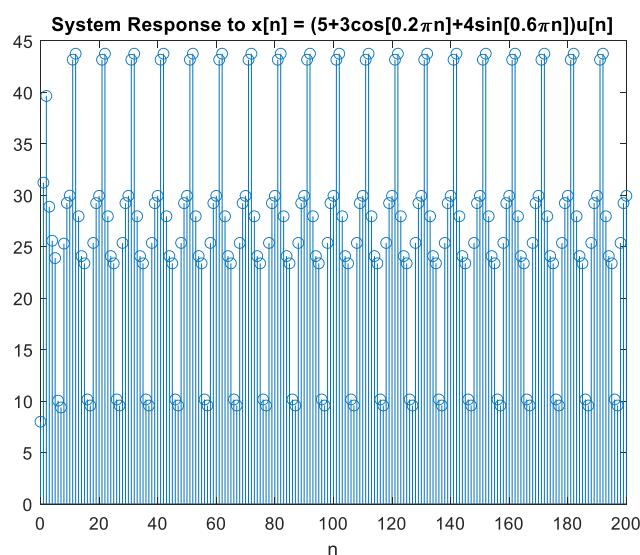
آزمایش ۲-۲: عبور سیگنال گسسته از یک سیستم

۱. حل معادله تفاضلی:

أ.



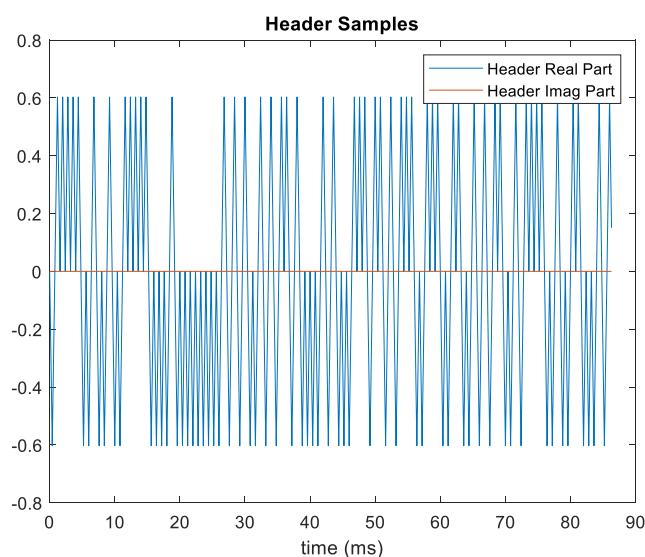
ب.

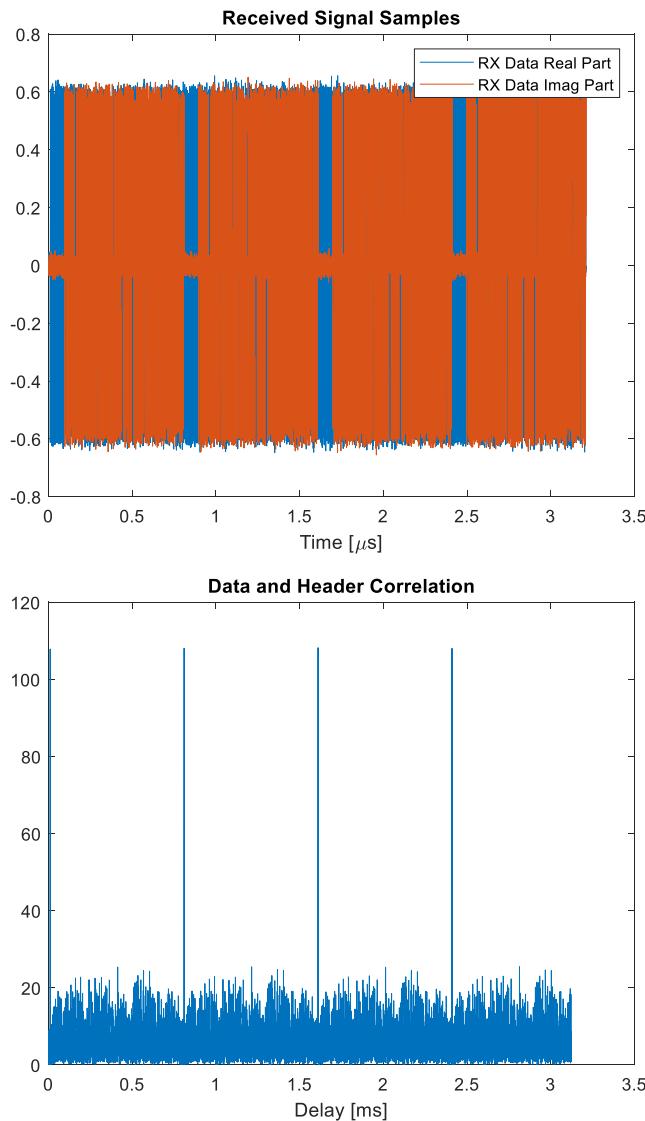


۲. تابع همبستگی و کاربرد آن:

ا.

ب.



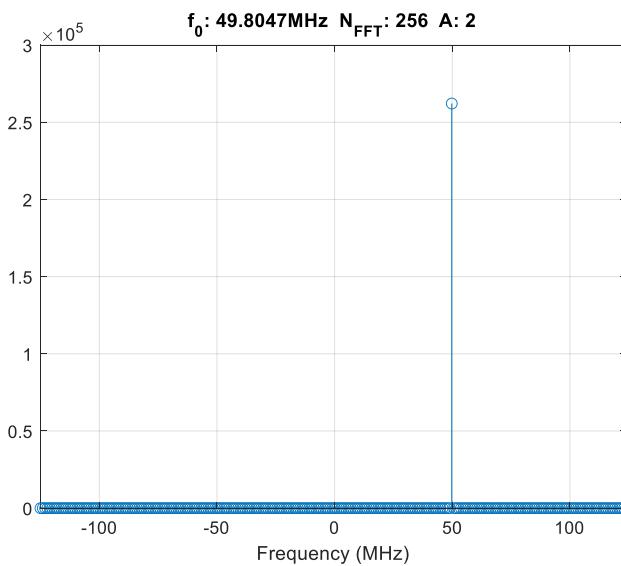


نیازی نیست برای به دست آوردن فاصله و دوره تناوب تکرار پالس از توابع MATLAB برای محاسبه پیک استفاده کنید. با بزرگنمایی نمودار سوم و قرایت نقطه‌ی قله می‌توان فاصله و طول هر بسته را به دست آورد.

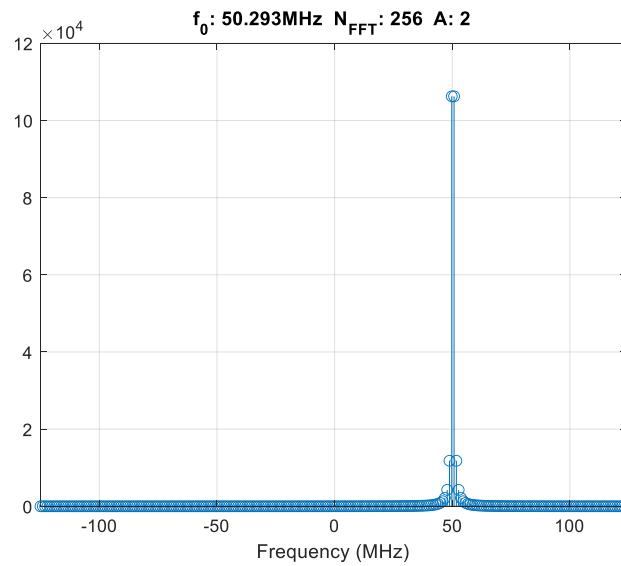
آزمایش ۲-۳: تحلیل حوزه‌ی فرکانس

۱. محاسبه‌ی طیف سیگنال:

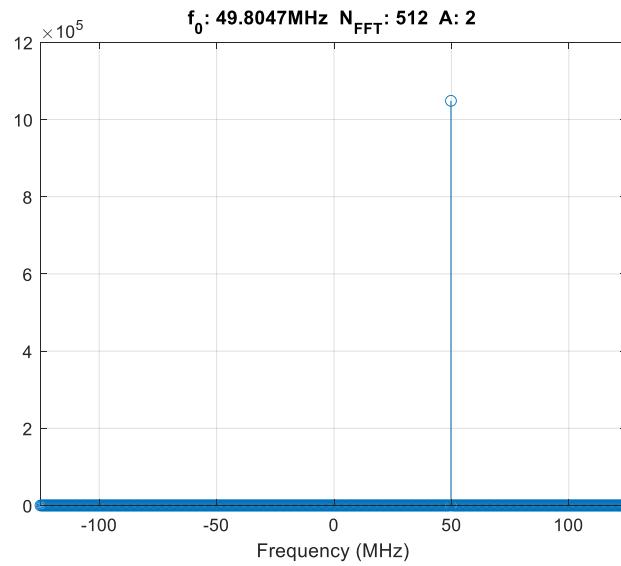
$$\text{f}_0 = \frac{250 \times 51}{256} \text{ MHz}, A = 2$$



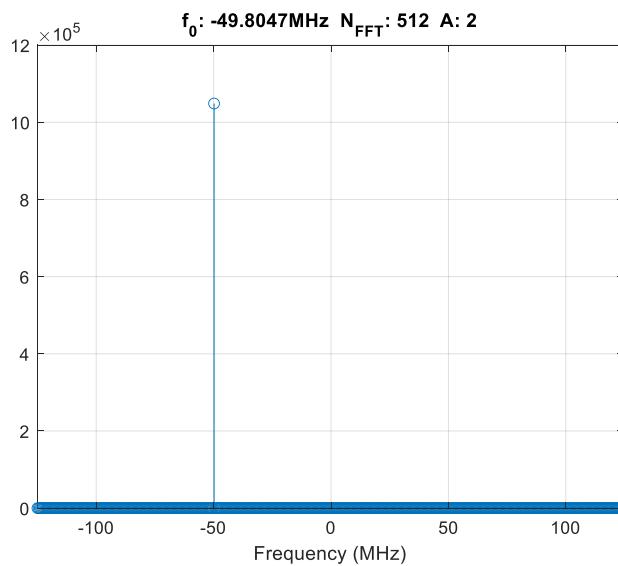
ب. و تعداد نقاط FFT برابر ۲۵۶ نقطه است.



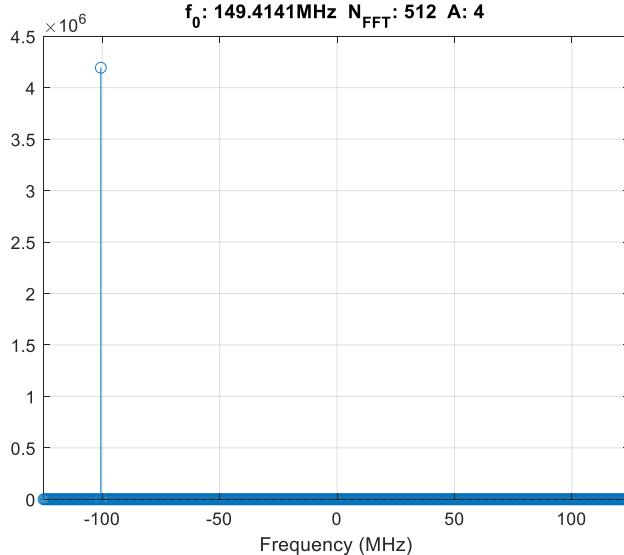
ت. و تعداد نقاط FFT برابر ۵۱۲ نقطه است.



ث. و تعداد نقاط FFT برابر ۵۱۲ نقطه است.



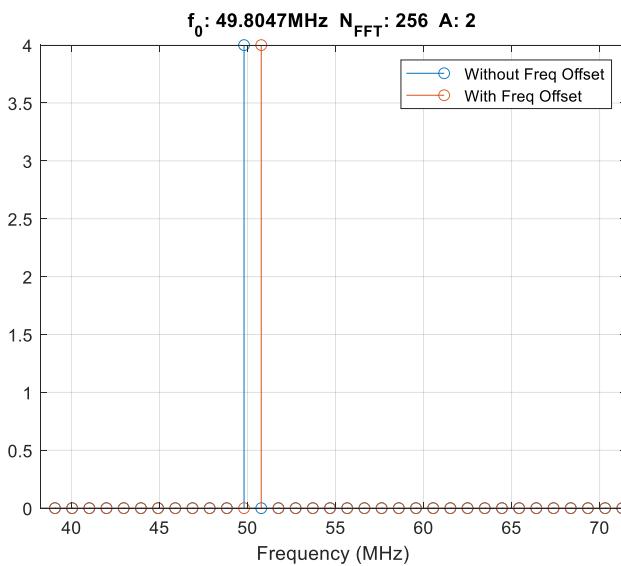
ج. f_0 و تعداد نقاط FFT برابر ۱۲ نقطه است.



ح. نسبت طول زمانی یک فریم FFT ($N_{\text{FFT}}T_s$) به دوره تناوب سیگنال ($1/f_0$) برای بخش‌های قبل.

```
ratio1 = 51
ratio2 = 51.5
ratio3 = 102
ratio4 = -102
ratio5 = 306
```

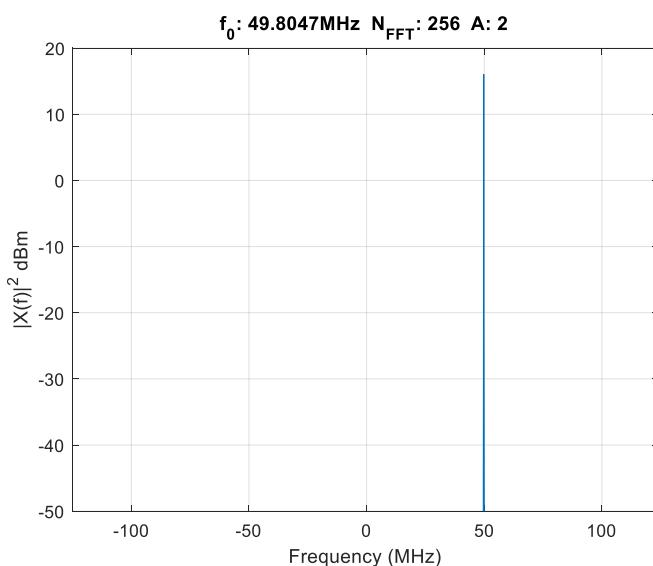
خ. در اینجا بخش الف در نظر گرفته شده است. فرکانس بخش الف اگر به اندازه‌ی رزولوشن جابه‌جا شود بر روی خانه‌ی FFT بعد می‌افتد و هیچ نشتی نیز وجود ندارد. نمودار بزرگنمایی شده‌ی آن در زیر آمده است. دقت نمایید ضریبی که در بخش خ نیز خواسته شده یود در اینجا اعمال شده است.



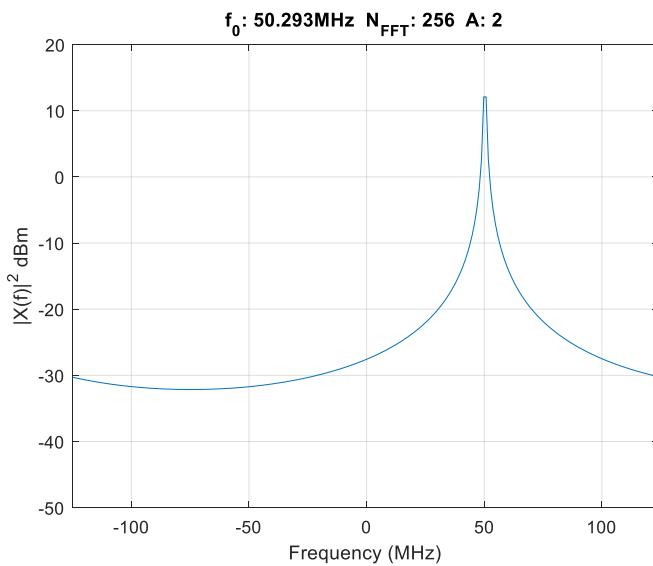
د. توان

```
time_power = 4
freq_power = 4
```

نمودار مربوط به بخش الف

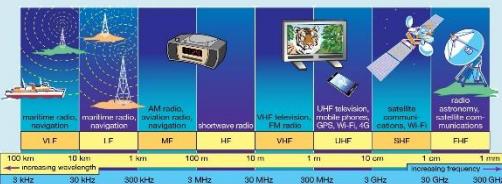


نمودار مربوط به بخش ب





آزمایش سوم



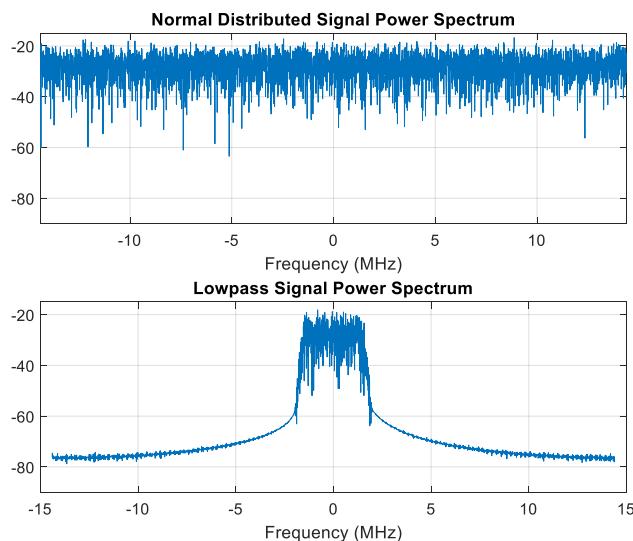
رادیو نرم افزار و پایش طیف سیگنال های رادیویی



آزمایش ۱-۳: معادل پایین گذر و میان گذر سیگنال

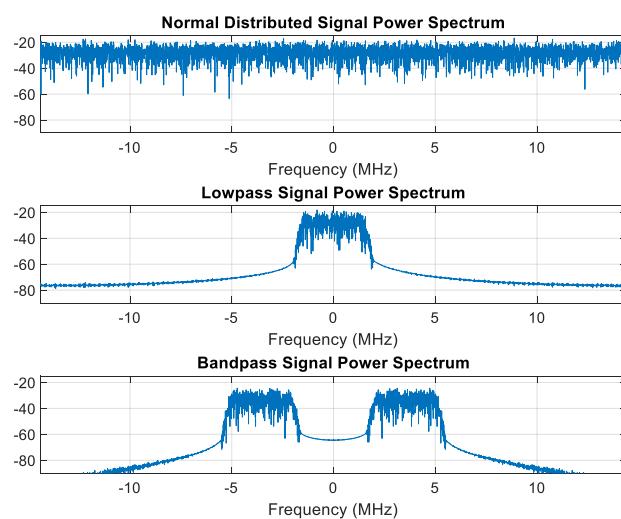
۱. تولید فرآیند میان گذر و پایین گذر:

۲. تولید فرآیند پایین گذر:



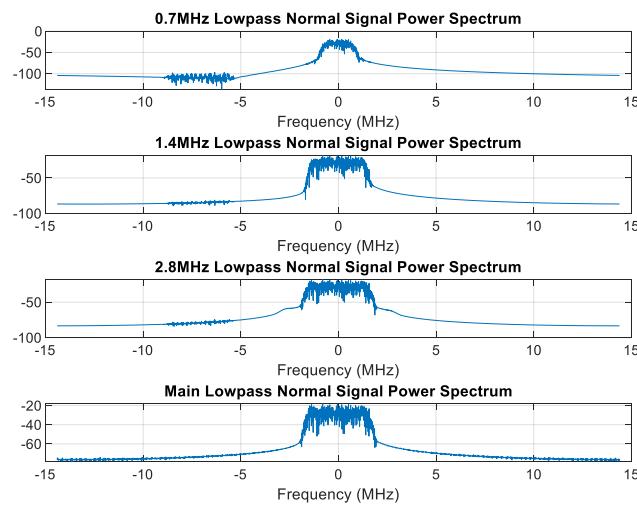
```
x_power = 1.0122
x_baseband_power = 0.1139
bw_noise = 3.3081e6
```

ب. تولید فرآیند میان گذر معادل یک فرآیند پایین گذر:



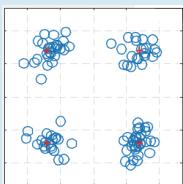
```
x_power = 1.0122
x_baseband_power = 0.1139
x_bandpass_power = 0.0570
```

تولید فرآیند پایین گذر از یک سیگنال میان گذر:



```
x_baseband_power = 0.1139
x_baseband_0p7_power = 0.0455
x_baseband_1p4_power = 0.0954
x_baseband_1p4_power = 0.0954
```

آزمایش چهارم



معرفی مدولاسیون دیجیتال خطی



آزمایش ۴-۱: پیاده‌سازی فرستنده مدولاسیون PAM

محاسبات و خروجی‌های زیر بر اساس این است که ابتدای کد خود (۱) `rng(1)` گذاشته باشد.

گام ۱. تولید بیت: می‌بایست ۲۰ عدد در قالب ماتریس زیر تولید شود.

```
b_tx =
0      0
1      1
0      0
0      1
0      0
0      1
0      0
0      1
0      0
1      0
```

گام ۲. نگاشت بیت به سمبول:

گام ۱. تولید دنباله‌ی کد گری به صورت باینری: می‌بایست تابعی به صورت زیر داشته باشد.

```
function [b_gray] = gray_code(k)
```

گام ۲. کدگذاری گری بیت‌های تولیدی:

```
b_gray =
0      0
0      1
1      1
1      0
```

گام ۳. تولید سمبول‌های ارسالی:

```
sym_idx =
0
2
0
1
0
1
0
1
0
3
```

```
mod_sym =
-1.3416
0.4472
-1.3416
-0.4472
-1.3416
```

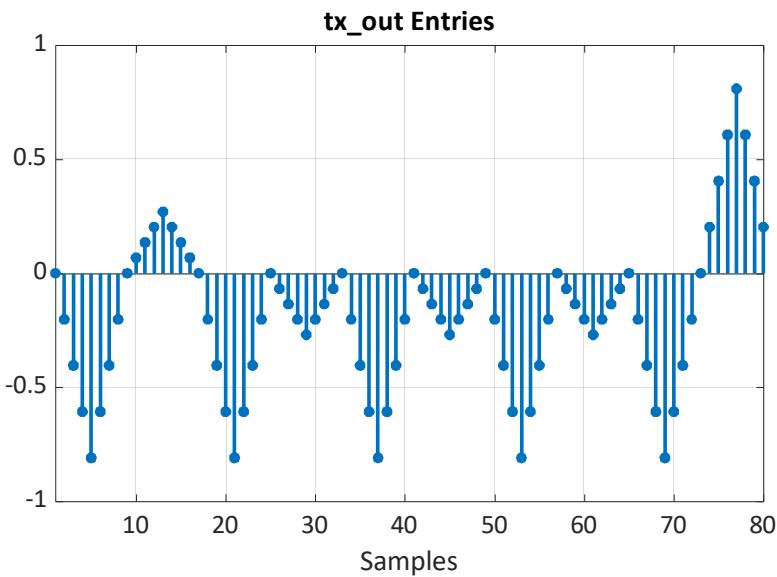
```
-0.4472
-1.3416
-0.4472
-1.3416
1.3416
```

گام ۳. شکل دهی پالس ارسالی: ابتدا می‌بایست تابعی به صورت زیر داشت.

```
function [tx_smpl, cons] = pulse_modulation(sym_idx, modulation, M, fs,
smpl_per_symbl, pulse_name, pulse_shape_mode, varargin)
```

عدادهای زیر به ترتیب در سطر درایه‌های بردار tx_smpl می‌باشد.

0.0000	-0.2023	-0.4045	-0.6068	-0.8090	-0.6068	-0.4045
-0.2023	0.0000	0.0674	0.1348	0.2023	0.2697	0.2023
0.1348	0.0674	0.0000	-0.2023	-0.4045	-0.6068	-0.8090
-0.6068	-0.4045	-0.2023	0.0000	-0.0674	-0.1348	-0.2023
-0.2697	-0.2023	-0.1348	-0.0674	0.0000	-0.2023	-0.4045
-0.6068	-0.8090	-0.6068	-0.4045	-0.2023	0.0000	-0.0674
-0.1348	-0.2023	-0.2697	-0.2023	-0.1348	-0.0674	0.0000
-0.2023	-0.4045	-0.6068	-0.8090	-0.6068	-0.4045	-0.2023
0.0000	-0.0674	-0.1348	-0.2023	-0.2697	-0.2023	-0.1348
-0.0674	0.0000	-0.2023	-0.4045	-0.6068	-0.8090	-0.6068
-0.4045	-0.2023	0.0000	0.2023	0.4045	0.6068	0.8090
0.6068	0.4045	0.2023				



آزمایش ۴-۲: پیاده‌سازی گیرنده‌ی مدولاسیون PAM

۱. تشخیص قدرت سیگنال در راستای سیگنال‌های پایه: می‌بایست تابعی به صورت زیر در یک فایل جداگانه داشته باشد.

```
function [rx_sym] = corr_match(rx_smpl, p, smpl_per_symbl, rx_mode)
```

۲. گیرنده‌ی حداقل فاصله: می‌بایست تابعی به صورت زیر در یک فایل جداگانه داشته باشد.

```
function [det_sym] = min_dist_detector(rx_sym, constellation);
```

۳. دمودولاسیون یا آشکارسازی پالس: می‌بایست تابعی به صورت زیر در یک فایل جداگانه داشته باشد.

```
function [det_sym_idx, rx_sym] = pulse_demodulation(rx_smpl, modulation, M, fs,
smpl_per_symbl, pulse_name, rx_mode, varargin)
```

```
det_sym_idx =
1
```

```

3
1
2
1
2
1
2
1
4

rx_sym =
-1.3416
0.4472
-1.3416
-0.4472
-1.3416
-0.4472
-1.3416
-0.4472
-1.3416
1.3416

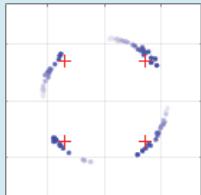
det_bit_dec =
0
3
0
1
0
1
0
1
0
2

det_bit =
0 0
1 1
0 0
0 1
0 0
0 1
0 0
0 1
0 0
1 0

```



آزمایش پنجم



کاوش در مدولاسیون دیجیتال خطی

شرح آزمایش

ابتدا فایل شبیه‌سازی از (1) `rng` استفاده نمایید.

آزمایش ۵-۲: پیاده‌سازی فرستنده مدولاسیون

فرض می‌کنیم `pkt_size` برابر با 10 باشد و مدولاسیون 4PSK است.

گام ۱. تولید بیت: صحبت‌سنگی ماتریس `b_tx` با اندازه‌ی $\text{pkt_size} \times k$ می‌باشد.

`b_tx =`

```

0      0
1      1
0      0
0      1
0      0
0      1
0      0
0      1
0      0
1      0

```

گام ۲. نگاشت بیت به سمبول:

کدگذاری گری بیت‌های تولیدی:

صحبت‌سنگی ماتریس `b_gray`

`b_gray =`

```

0      0
0      1
1      1
1      0

```

صحبت‌سنگی بردار `sym_idx`

`sym_idx =`

```

0
2
0
1
0
1
0
1
0
3

```

کدگذاری طبیعی بیت‌های تولیدی:

صحبت‌سنگی بردار `sym_idx` با کدگذاری باینری

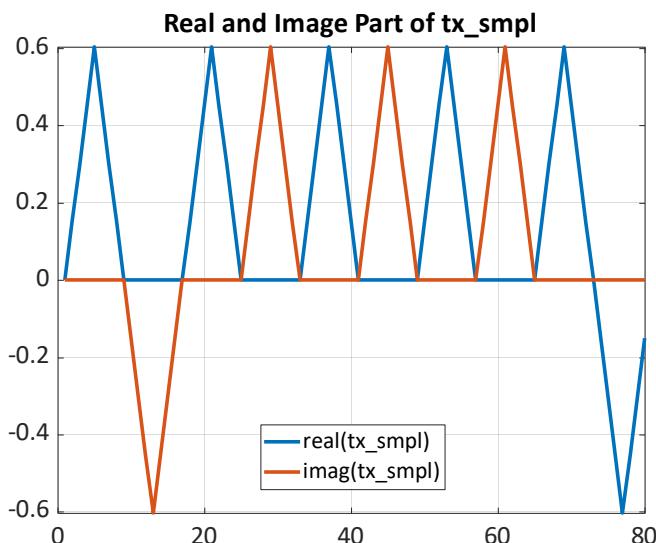
```
sym_idx =
0
3
0
1
0
1
0
1
0
2
```

تولید سمبول‌های ارسالی:

mod_sym بردار صحت‌سنگی

```
mod_sym =
1.0000 + 0.0000i
-0.0000 - 1.0000i
1.0000 + 0.0000i
0.0000 + 1.0000i
1.0000 + 0.0000i
-1.0000 + 0.0000i
```

گام ۳. شکل‌دهی پالس ارسالی: صحت‌سنگی بردار **tx_smp1**



گام ۴. شبیه‌سازی کانال با نویز سفید گاوی

افزودن نویز به سیگنال: صحت‌سنگی بخشی از بردار **rx_smp1_noise**

```
rx_smp1_noise =
```

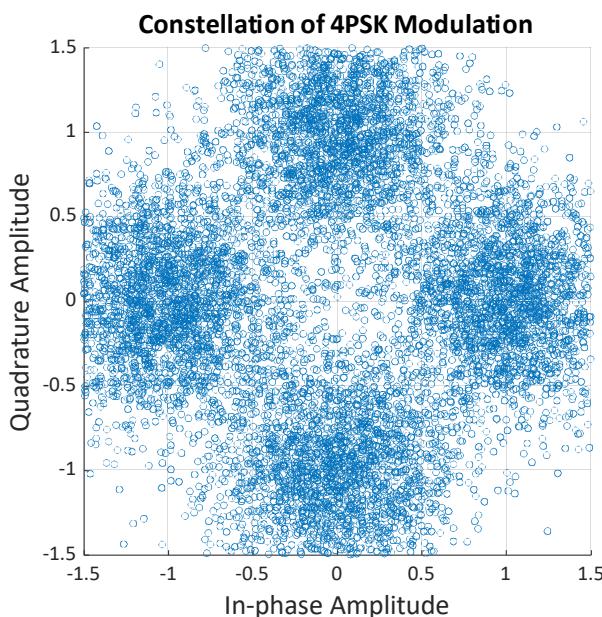
```
0.4160 - 0.1504i
0.8690 + 0.5969i
0.2041 + 0.1294i
0.6659 - 0.3883i
1.2335 + 0.6832i
1.0801 + 0.2205i
-0.3587 + 0.2327i
-0.2213 - 0.5180i
```

```

-0.1477 + 0.3023i
0.4963 + 0.0547i
-0.4841 - 0.6267i
-0.1035 - 0.1709i
...

```

گام ۵. دمودولاسیون یا آشکارسازی پالس: صحبت‌سنگی سمبل‌های دریافتی (`rx_sym`) با استفاده از نمودار. در اینجا تعداد سمبل‌ها برابر ۱۰۰۰ در نظر گرفته می‌شود.



گام ۶. محاسبهٔ خطای سمبل و خطای بیت:
BER و SER با اعمال کدگذاری گری:

SER: 0.0454 BER: 0.0228

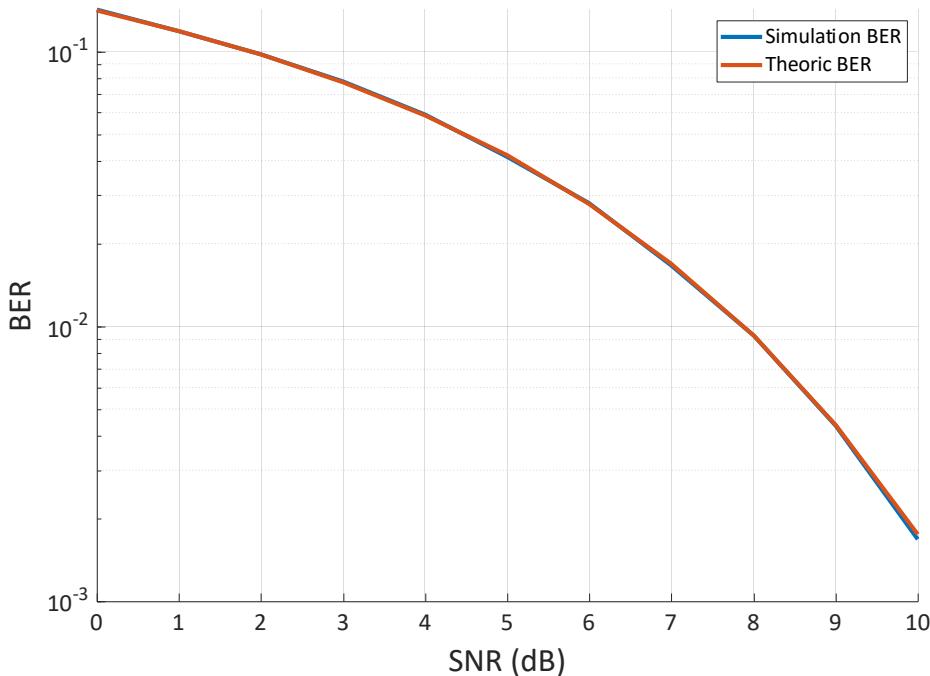
بدون اعمال کدگذاری گری: SER و BER

SER: 0.0447 BER: 0.0336

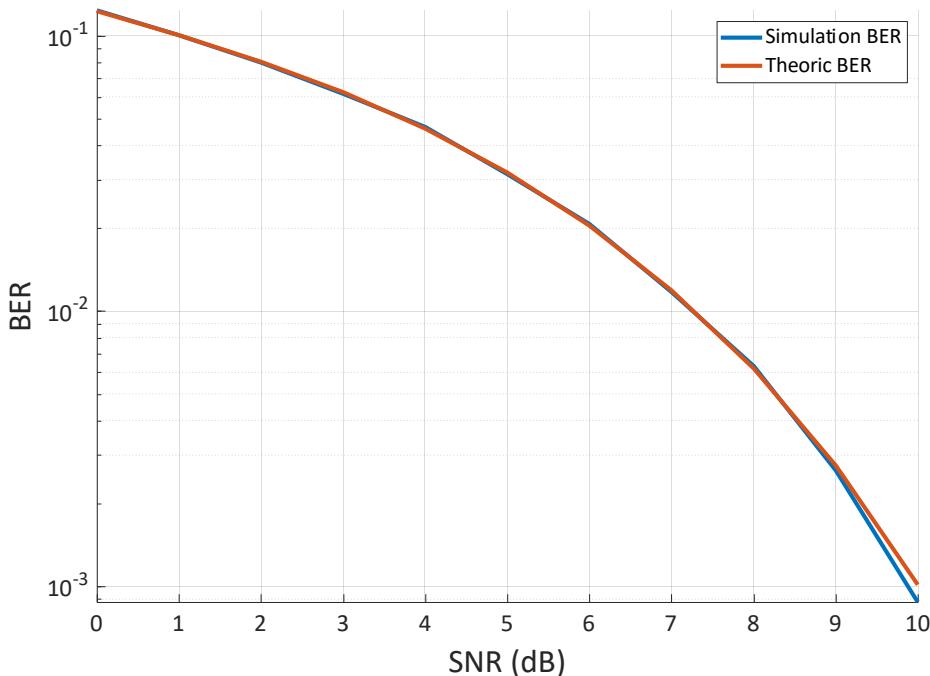
آزمایش ۵-۵: خواسته‌های کلی

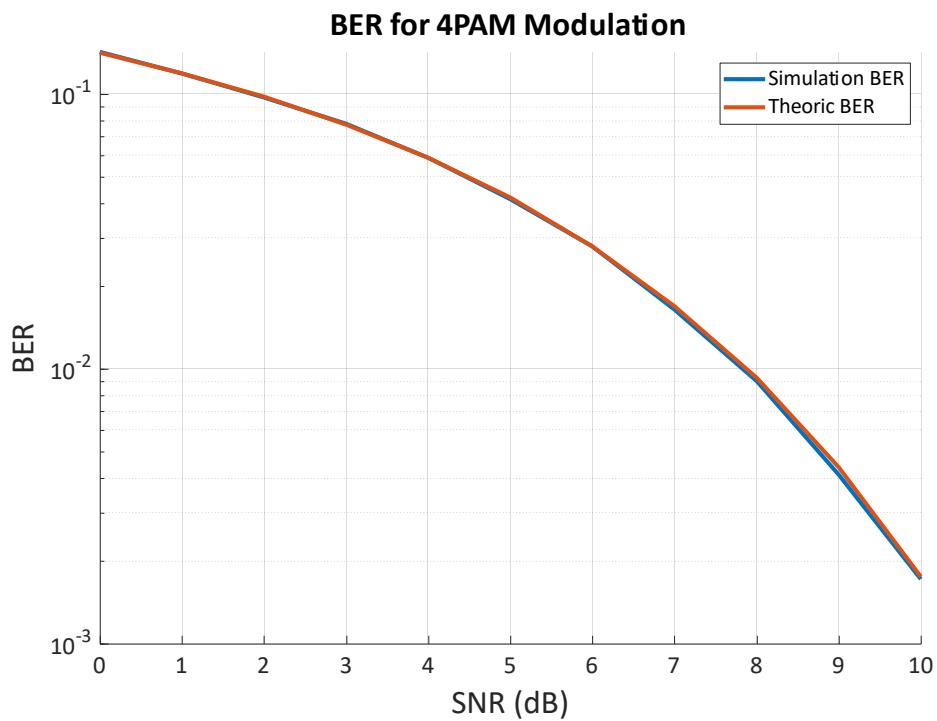
در نمودارهایی که در ادامه امده است ابتدا بروگرام (rng) می‌باشد و تعداد سمبول‌ها ۱۰۰۰۰۰ می‌باشد. منظومه‌های سیگنالی نیز در نسبت سیگنال به نویز ۱۰dB رسم شده است.
گام ۷. رسم نمودار نرخ خطای بیت با اعمال کدگذاری گردید.

BER for 16QAM Modulation

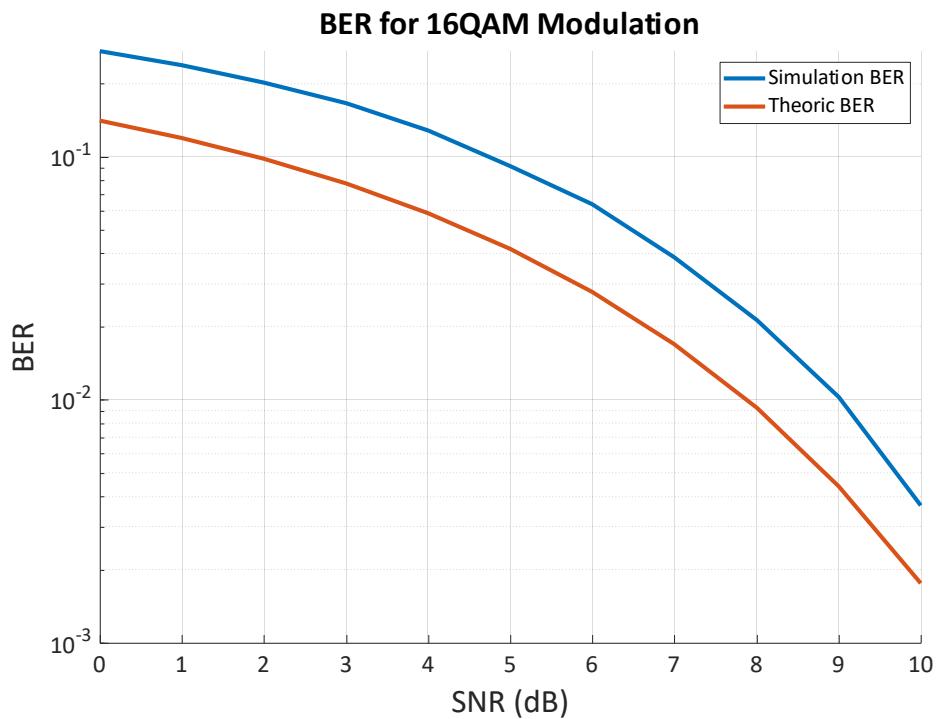


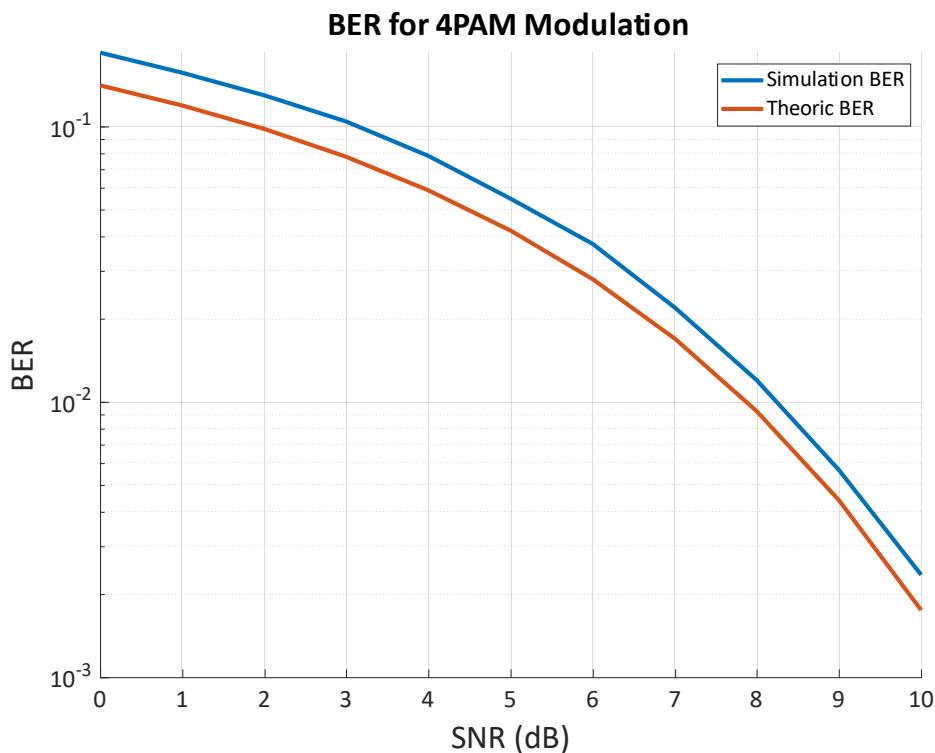
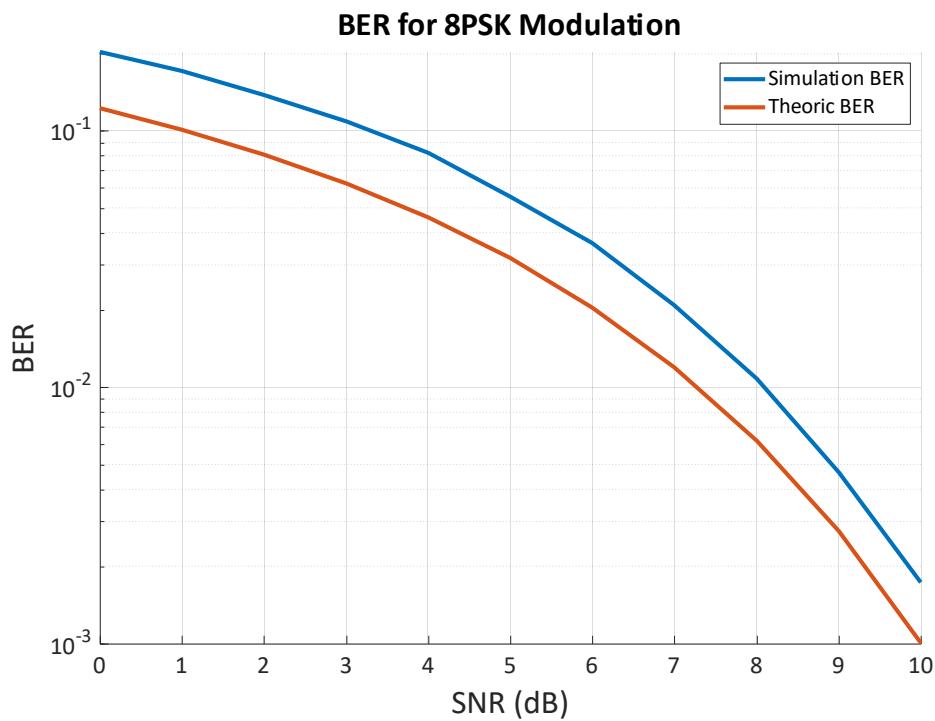
BER for 8PSK Modulation

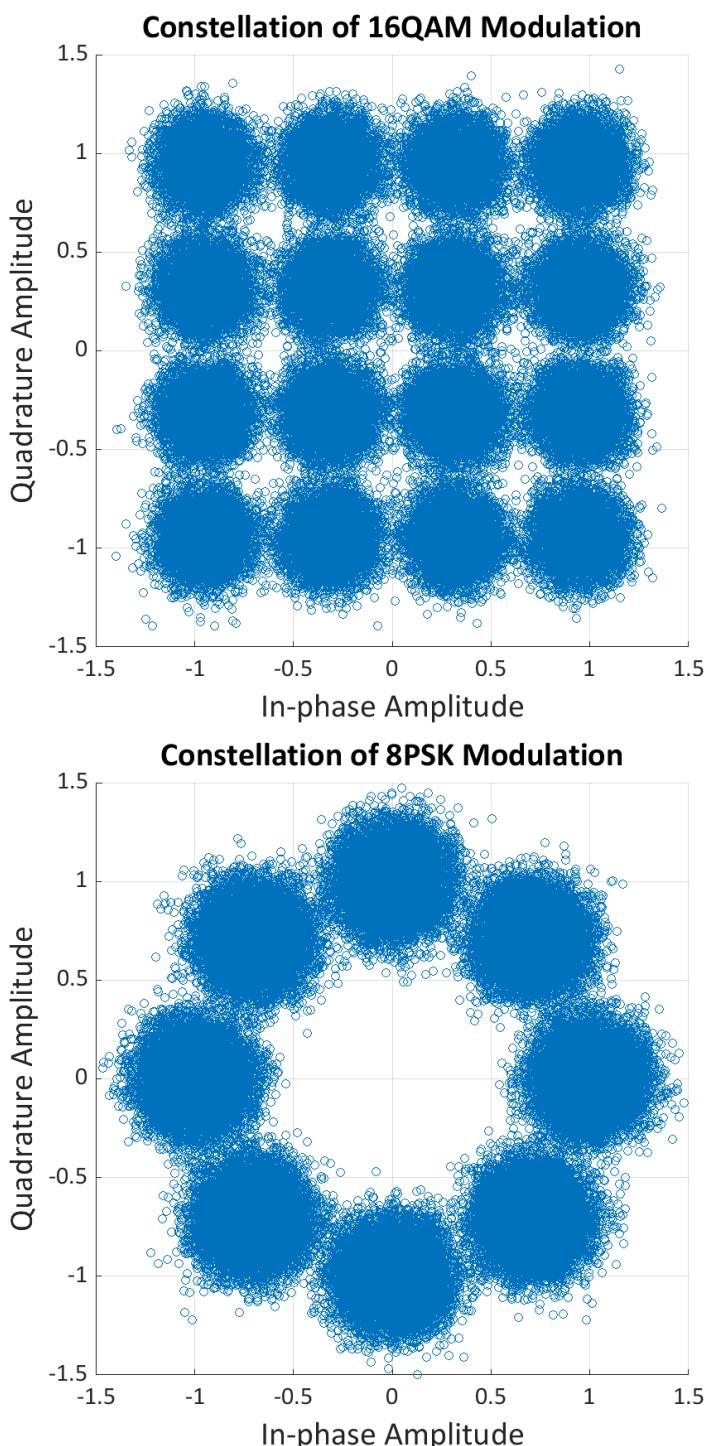


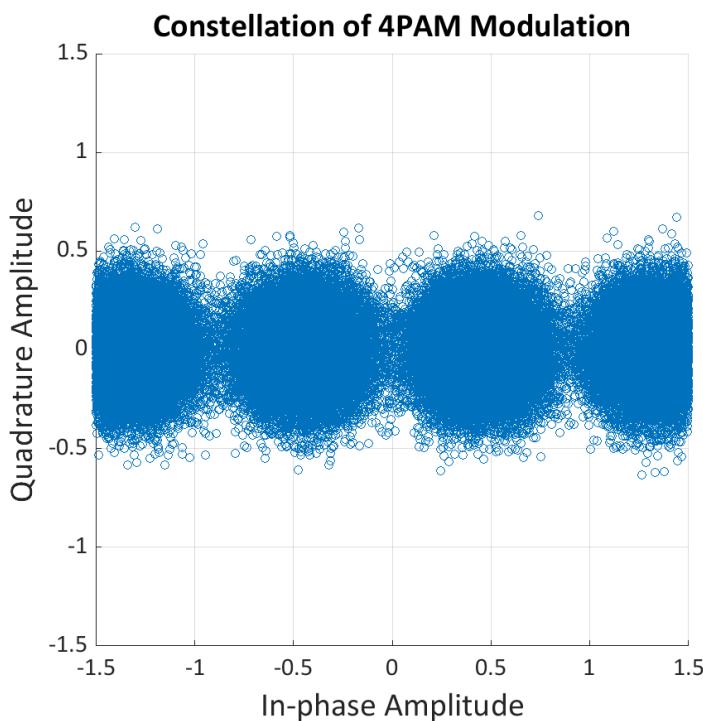


گام .۸ نمودارهای نرخ خطأ بدون اعمال کدگذاری گردید





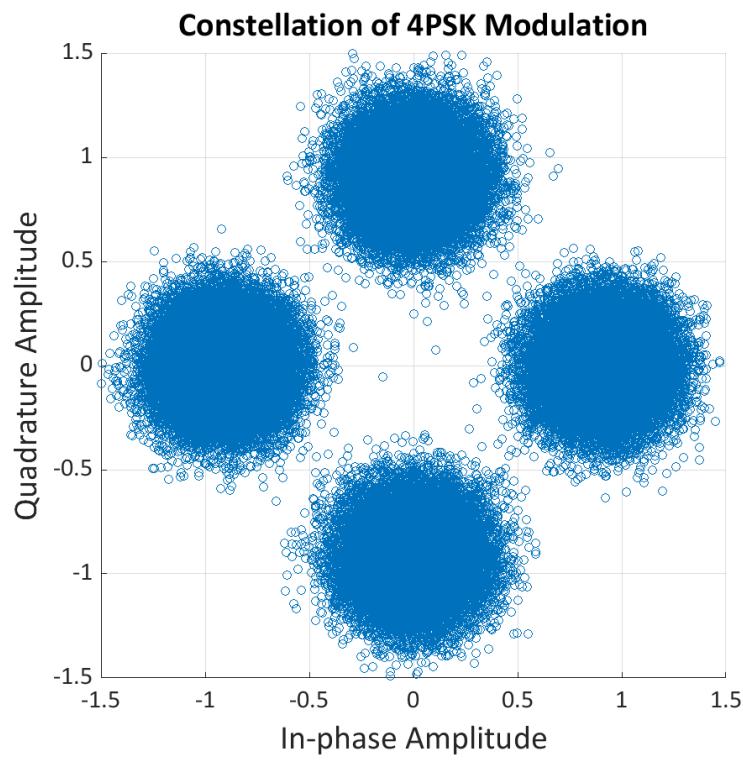




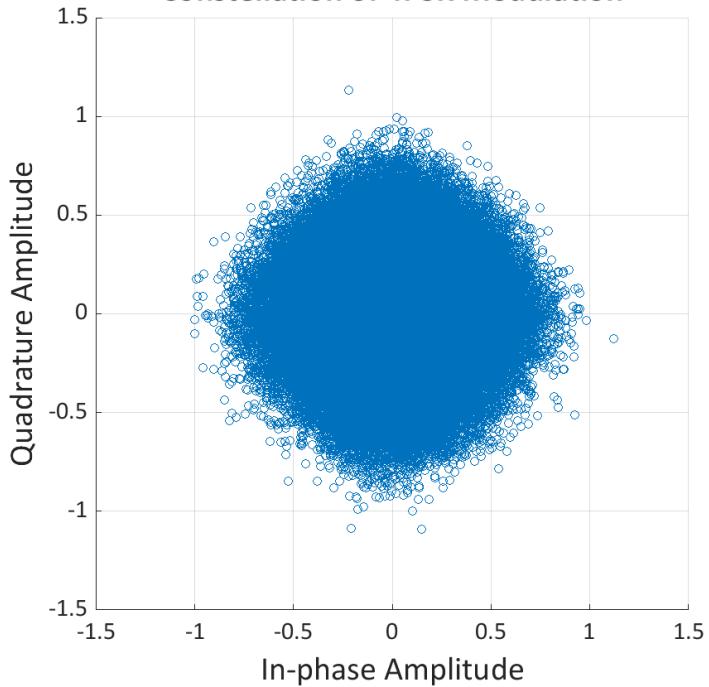
گام ۱۰. صحبت‌سنگی یکسانی تایگ با دو شکل موج متنفسی و root raised cosine

گام ۱۱. اثر تأخیر در کانال:

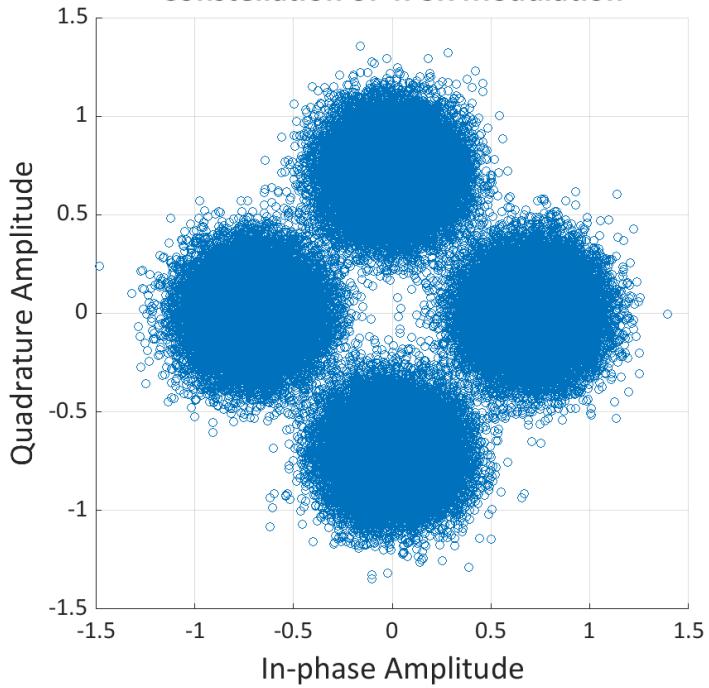
نمودارها و نرخ خطای بیت به ترتیب برای تأخیر کانال برابر با گرد شده‌ی ۰.۱، ۰.۵ و ۰.۸ برابر طول یک سمبول می‌باشد.
مدولاسیون 4PSK و نسبت سیگنال به نویز (E_b/N_0) برابر با ۱۰dB است



BER : 2.5×10^{-5}

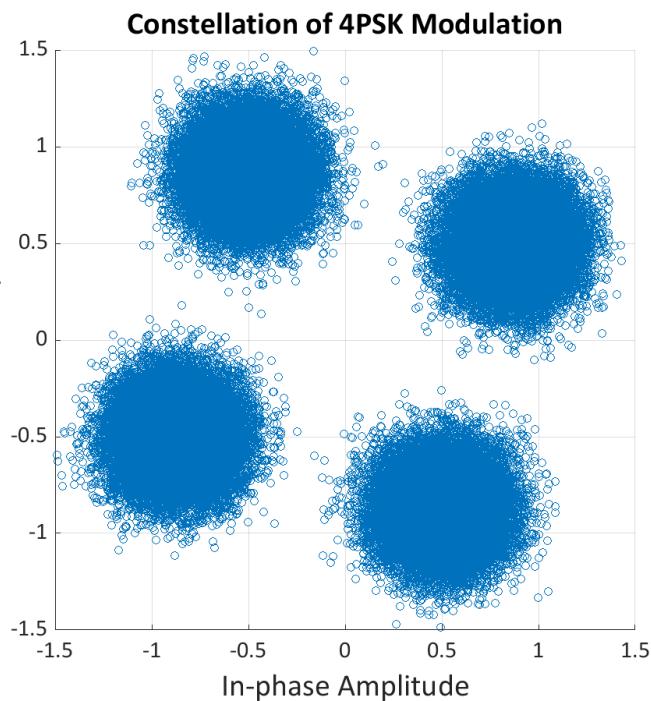
Constellation of 4PSK Modulation

BER : 0.3230

Constellation of 4PSK Modulation

BER : 0.5019

گام ۱۲ . اثر فاز اضافی کانال بر نرخ خطای بیت:

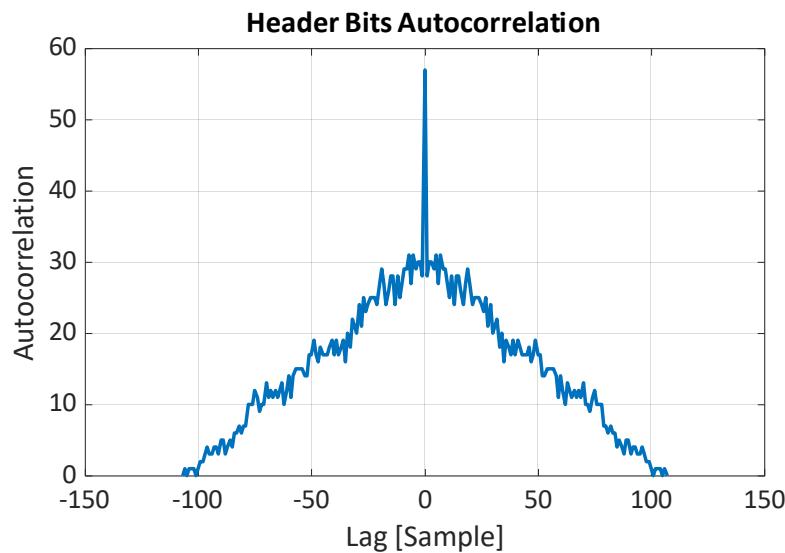


BER : 0.0382

گام ۱۳. اضافه کردن هدر

ستون اول ماتریس **hdr_bit** و خودهمبستگی آن

1	0	0	0	0	0	1	1	0	1	1	0	1
1	0	0	0	0	0	1	1	1	1	0	1	1
1	1	1	1	1	1	1	0	1	0	1	0	1
1	0	1	0	1	0	0	1	0	1	1	1	1
0	1	0	1	1	0	0	0	0	1	0	0	1
0	1	0	0	0	1	0	0	0	0	1	1	0
0	1	0	0	0	1	1	0	0	0	1	1	0
0	0	0	0	1	1	0	1	1	0	0	1	1
1	0	1	0	0	0	1	0	1	0	0	1	1



مقدار قلهی خودهمبستگی نمونه‌های زمانی هدر: 108 با فاز صفر

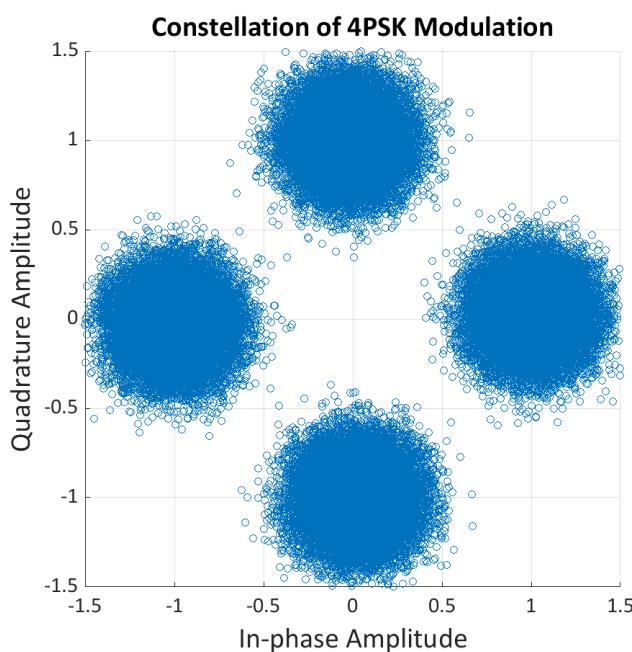
hdr_corr_max =

```
1.0800e+02
```

خروجی همبستگی متقابل بین نمونه‌های هدر و نمونه‌های سیگنال دریافتی:

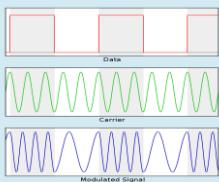
```
corr_out_rx_max =
96.8562 +52.5040i
idx_max =
5
```

جبران‌سازی فاز و تأخیر
منظومه‌ی سیگنالی بعد از جبران‌سازی ۱۴ گام



در SNR برابر با 10dB با تعداد نمونه‌های ۱۰۰۰۰۰ صفر می‌شود.

آزمایش ششم



مدولاسیون FSK همدوس

گام‌های تکمیل آزمایش

ابتدا فایل شبیه‌سازی از (1) `rng` استفاده نمایید.

آزمایش ۲-۶: پیاده‌سازی فرستنده مدولاسیون FSK

فرض می‌کنیم `pkt_size` برابر با ۱۰ باشد.

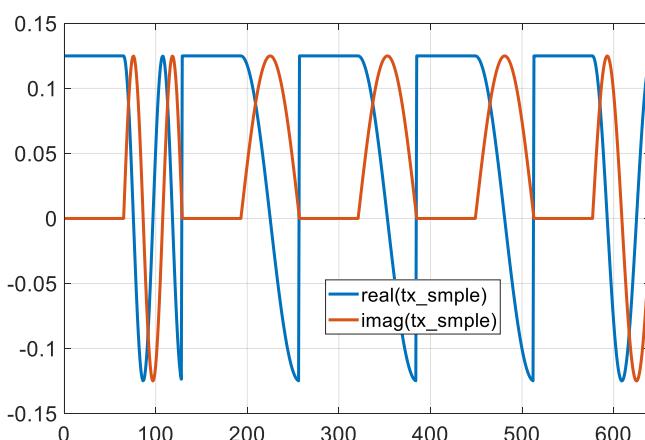
گام ۱. تولید بیت: صحت سنجی ماتریس `b_tx`:

```
b_tx =
0      0
1      1
0      0
0      1
0      0
0      1
0      0
0      1
0      0
1      0
```

گام ۲. نگاشت بیت به شماره‌ی سمبول: صحت سنجی بردار `sym_idx`

```
sym_idx =
0
3
0
1
0
1
0
1
0
2
```

گام ۳. تولید نمونه‌های زمانی سمبول ارسالی: صحت سنجی تولید نمونه‌های ارسالی `tx_smp1`



آزمایش ۳-۶: مدل سازی کانال

گام ۴. شبیه‌سازی کانال با نویز سفید گاوی: صحت سنجی بردار نویز تولیدی برای E_b/N_0 برابر با ۱۰dB (بخشی از بردار نویز تولیدی)

```
noise_smp1 =
0.1858 + 0.1383i
0.3208 + 0.4851i
-0.0435 - 0.0979i
0.0954 + 0.1198i
0.2816 + 0.1626i
0.2804 + 0.0084i
-0.2949 + 0.0333i
-0.1662 + 0.1822i
-0.0660 - 0.0557i
0.2217 + 0.0815i
-0.2163 + 0.1521i
-0.0463 + 0.0612i
0.2009 - 0.0356i
0.0104 + 0.0394i
0.0714 - 0.3029i
-0.0509 + 0.1125i
0.1247 + 0.2471i
0.1468 - 0.1305i
-0.0776 - 0.0898i
0.2842 - 0.1144i
0.0934 - 0.1595i
```

آزمایش ۴-۶: پیاده‌سازی گیرندهٔ مدولاسیون FSK

گام ۵. دمودولاسیون یا آشکارسازی پالس: صحت سنجی مراحل میانی دمودولاسیون

ماتریس خروجی کروماتورهای چهارگانه:

1.0071 + 0.2678i	0.2301 - 0.5272i	0.1354 + 0.0768i	0.1316 - 0.2025i
-0.0779 + 0.3464i	0.1913 + 0.0275i	-0.1206 + 0.6427i	1.1417 + 0.1540i
1.0595 - 0.0011i	-0.0914 - 0.8211i	-0.1949 + 0.2063i	0.2696 - 0.1893i
0.3165 + 0.4536i	0.9049 - 0.1232i	0.1223 - 0.8286i	-0.2220 - 0.0933i
0.9524 - 0.1912i	-0.0836 - 0.5960i	0.1236 - 0.1153i	-0.0791 - 0.3385i
-0.0162 + 0.6201i	1.1771 - 0.0727i	-0.0886 - 0.8867i	-0.0930 + 0.0668i
1.0855 - 0.0804i	-0.1852 - 0.5615i	0.0401 + 0.1100i	-0.1268 - 0.1138i
-0.0933 + 0.7522i	1.0438 + 0.0047i	-0.1195 - 0.4640i	0.2294 + 0.1137i
0.8772 + 0.2424i	0.1505 - 0.5703i	-0.1465 - 0.0123i	-0.0230 + 0.0648i
-0.1540 + 0.0173i	0.1698 + 0.8738i	1.1034 - 0.3322i	-0.3168 - 0.5730i

```
det_sym_idx =
1
4
1
2
1
2
1
2
1
3
```

بسته به نحوهٔ کد نویسی می‌تواند بردار فوق منهای ۱ باشد.

```
rx_sym =
1.0071    0.2301    0.1354    0.1316
-0.0779    0.1913   -0.1206    1.1417
1.0595   -0.0914   -0.1949    0.2696
0.3165    0.9049    0.1223   -0.2220
0.9524   -0.0836    0.1236   -0.0791
```



```

-0.0162    1.1771   -0.0886   -0.0930
 1.0855   -0.1852    0.0401   -0.1268
-0.0933    1.0438   -0.1195    0.2294
 0.8772    0.1505   -0.1465   -0.0230
-0.1540    0.1698    1.1034   -0.3168

```

در واقع خروجی ای است که می‌توان از آن برای رسم منظمه مدولاسیون 4FSK در چهار بعد استفاده نمود.

گام ۶.

محاسبه خطای سمبل: صحت‌سنجی خطای سمبل با `pkt_size` برابر با 10^5

`ser =`

`1.0000e-05`

تبديل سمبل به بیت و محاسبه خطای بیت: صحت‌سنجی خطای بیت با `pkt_size` برابر با 10^5

گام ۷.

`ber =`

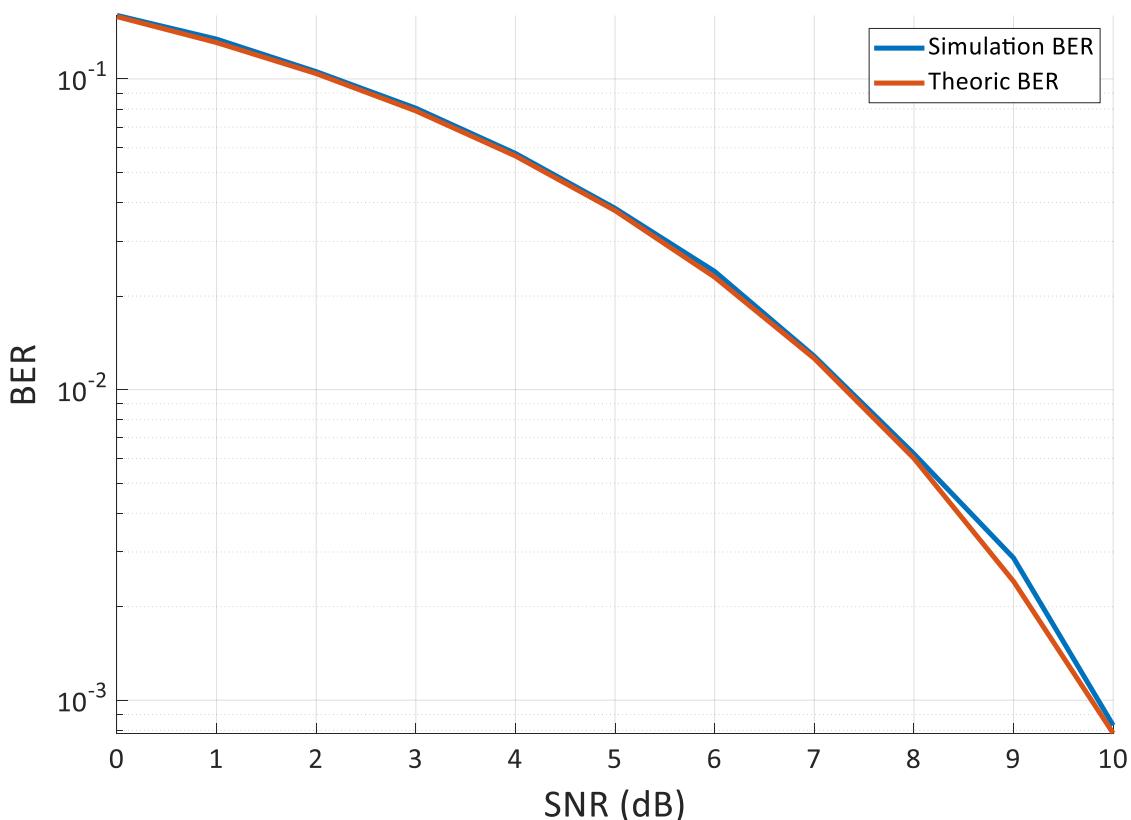
`5.0000e-06`

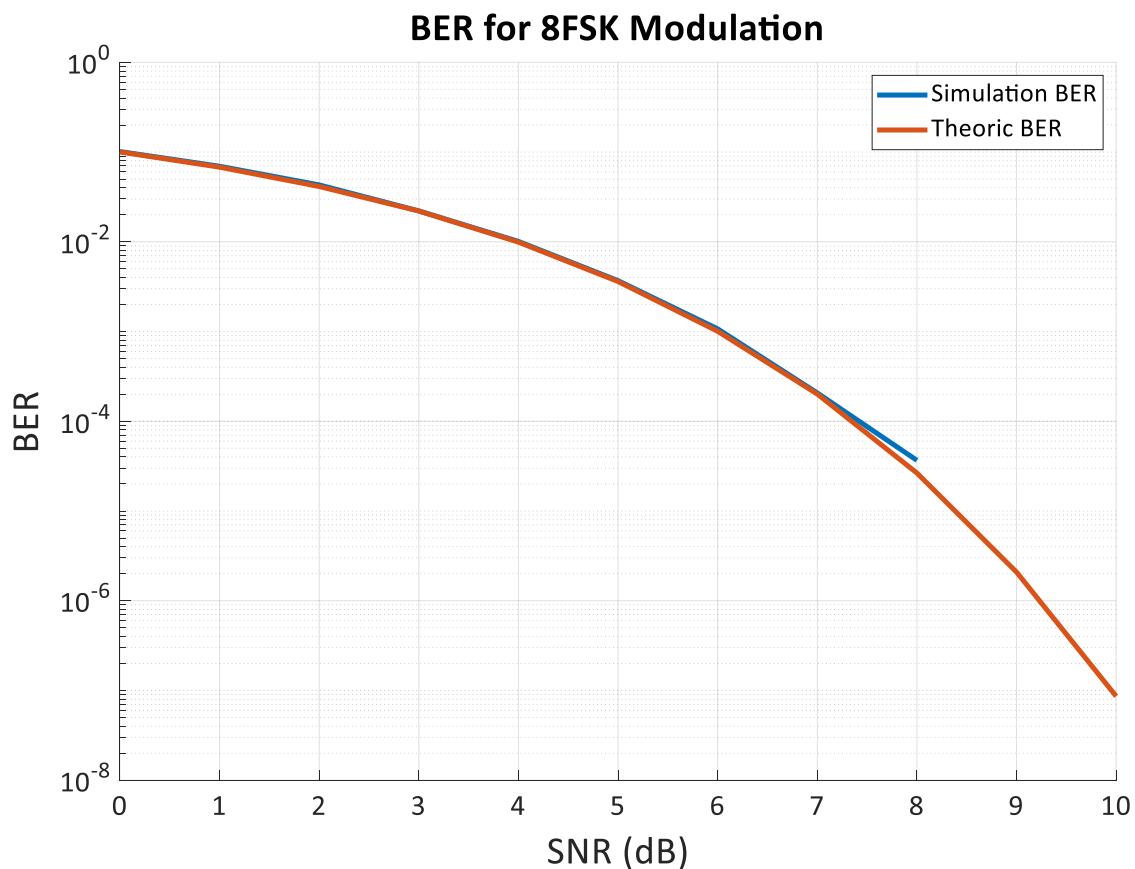
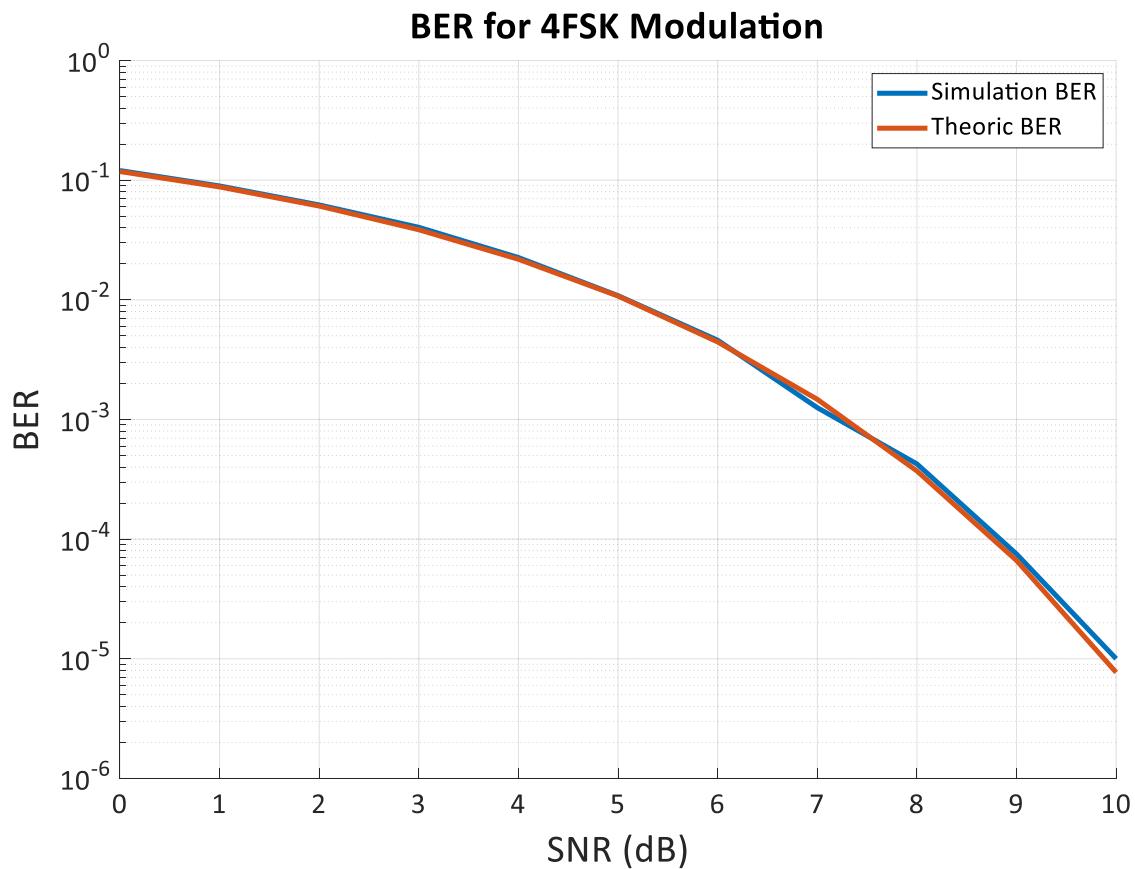
آزمایش ۶-۵: خواسته‌های کلی

گام ۸.

رسم نمودار نرخ خطای بیت: صحت‌سنجی نمودار خطای بیت با `pkt_size` برابر با 10^5

BER for 2FSK Modulation



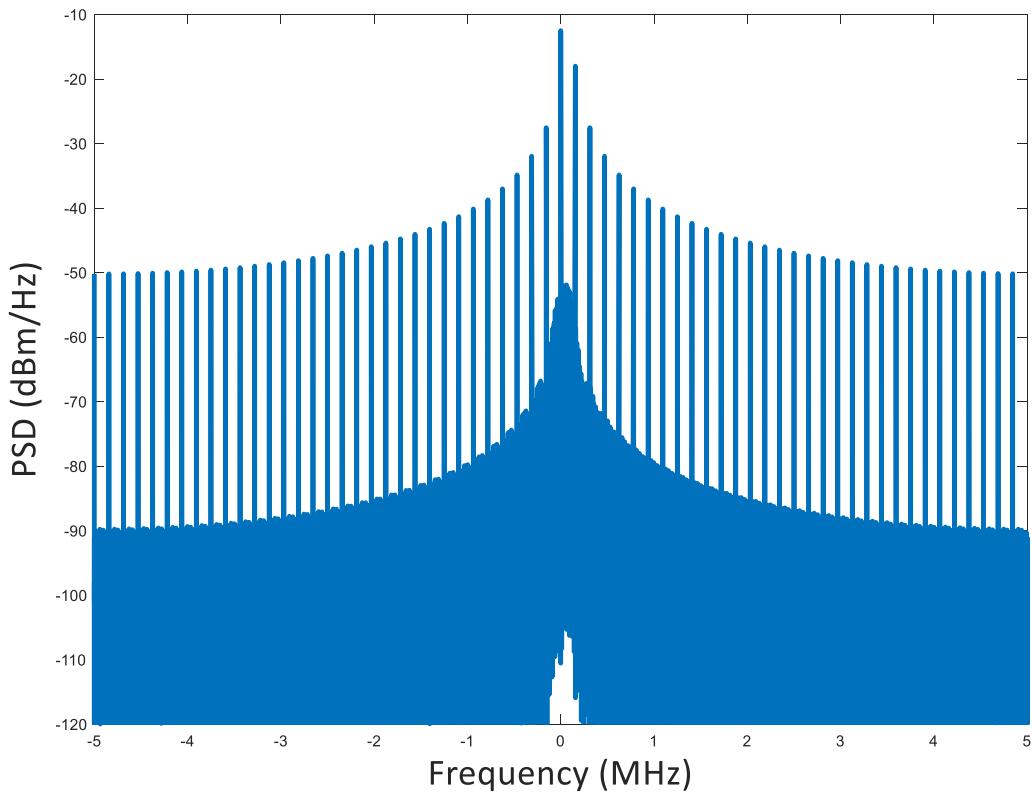


آزمایش ۵-۶: خواسته‌های کلی

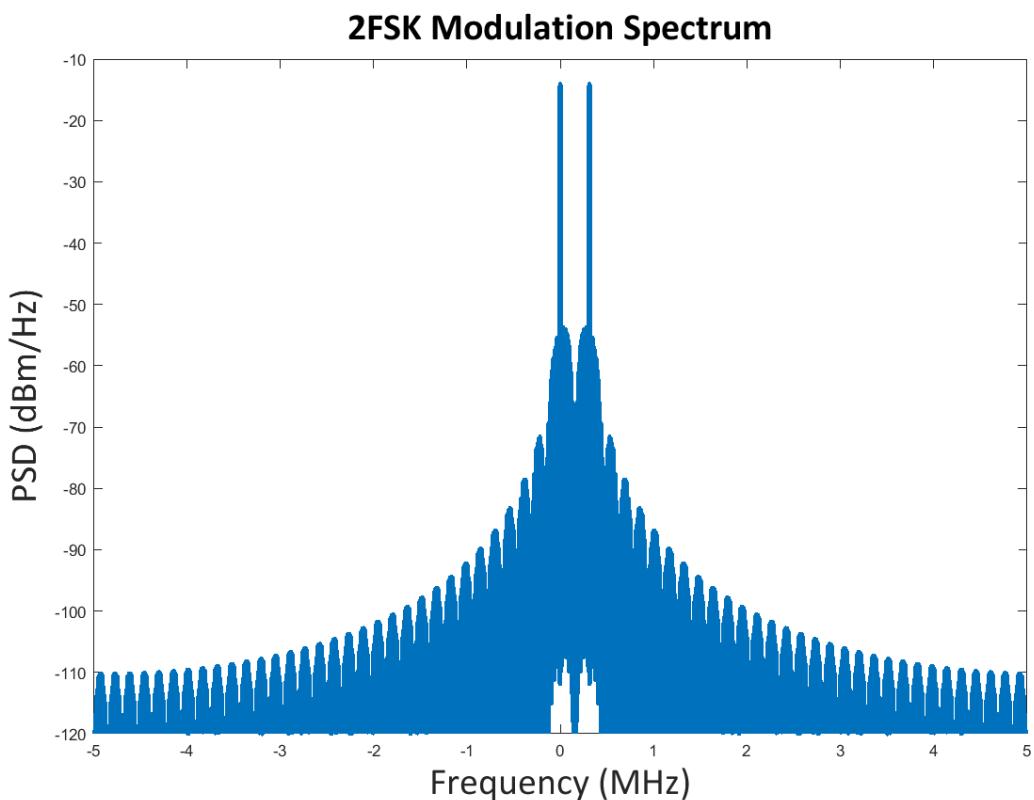
گام ۹. اثر Δf بر روی پهنای باند ارسالی:

$$\Delta f = \frac{1}{2T_s}$$

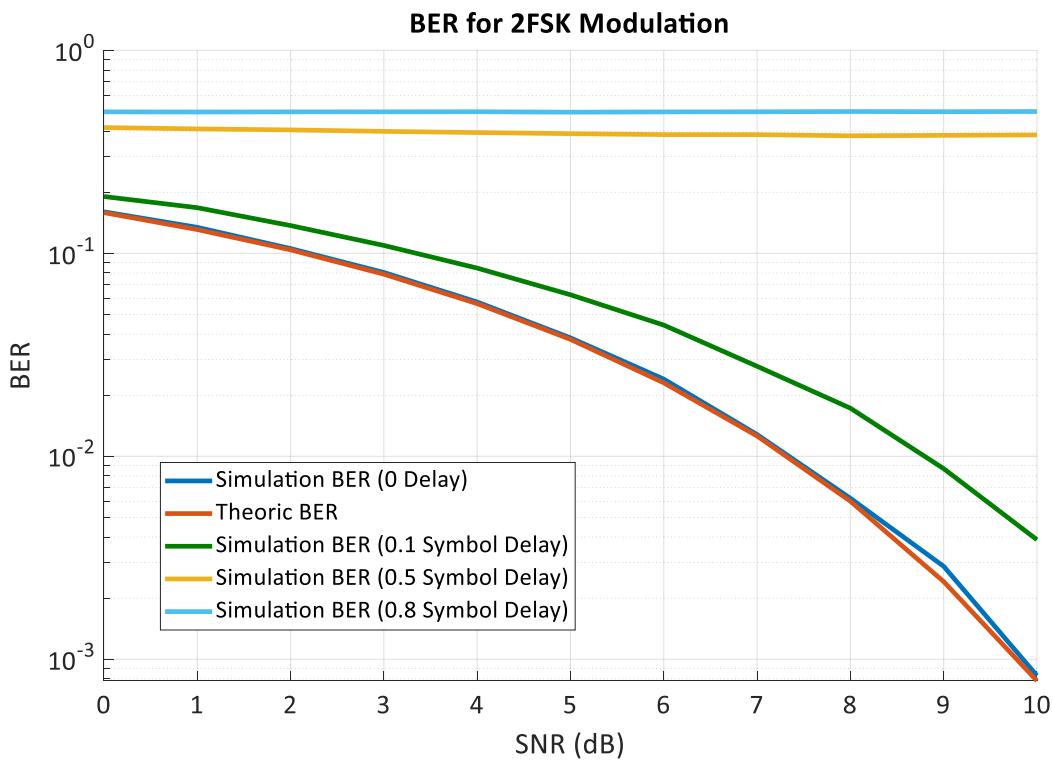
2FSK Modulation Spectrum



$$\Delta f = \frac{2}{T_s}$$



۱۰. گام اثربخشی در کانال بر روی نمودار نرخ خطای بیت: صحبت‌سنجی نمودار خطای بیت با `pkt_size` برابر با 10^5

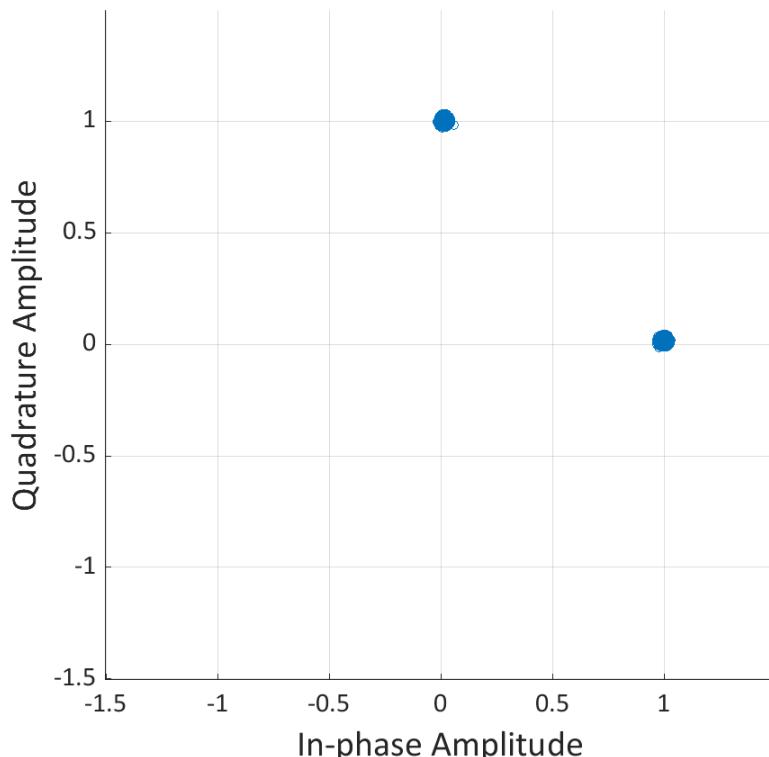


گام ۱۱. اثر تاخیر در کانال بر روی منظومه سیگنالی مدولاسیون 2FSK: منظومه‌ها با `pkt_size` برابر با 10^5 و نسبت سیگنال به نویز

40dB می‌باشد

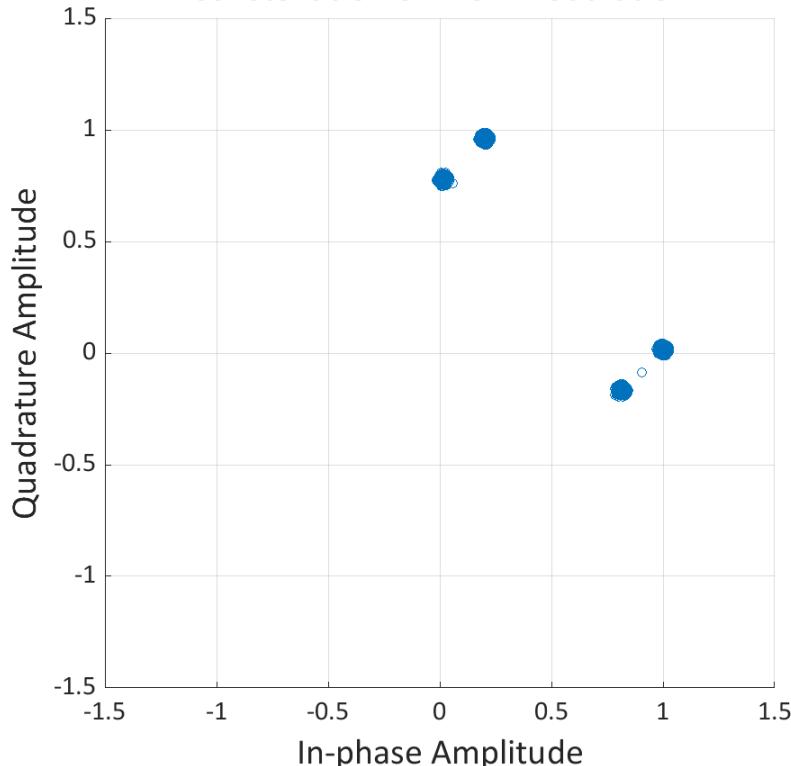
منظومه با تاخیر صفر

Constellation of 2FSK Modulation



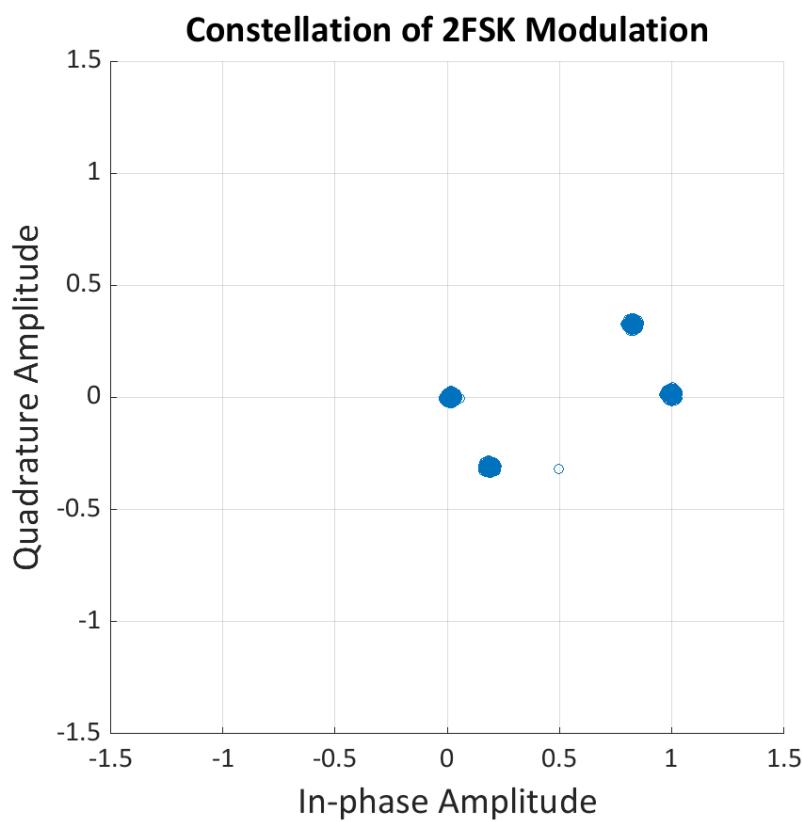
منظومه با تاخیر ۱، طول سمبل

Constellation of 2FSK Modulation

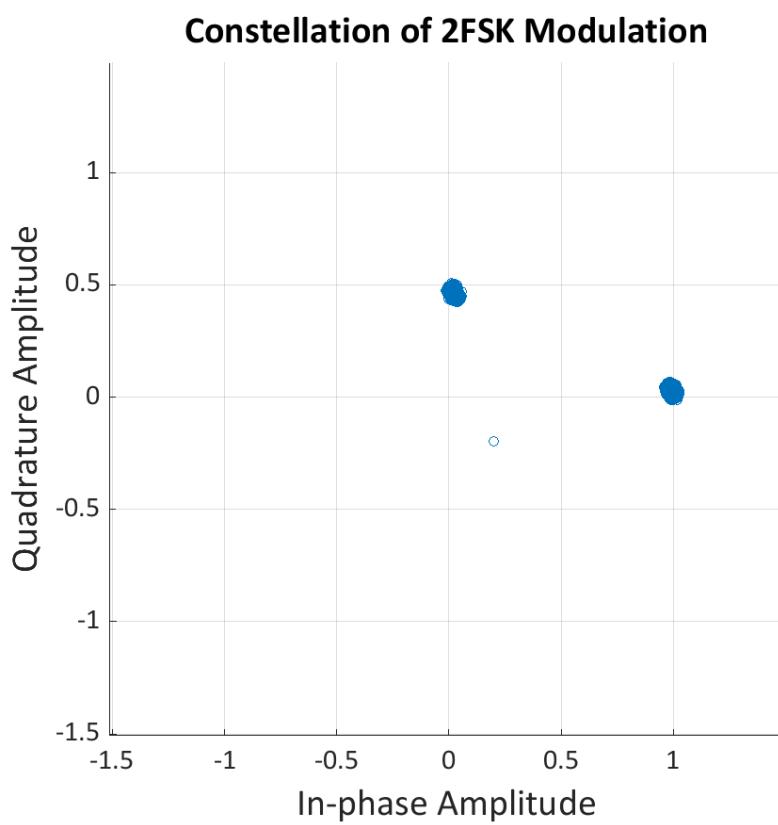


منظومه با تاخیر ۵، طول سمبل



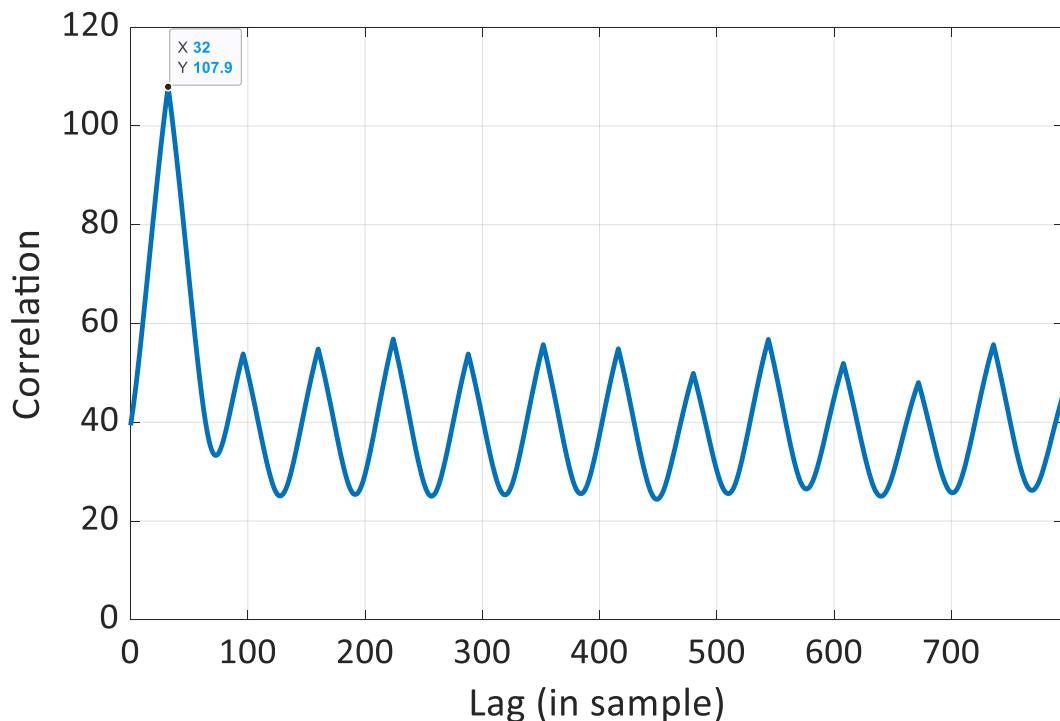


منظومه با تاخیر، طول سمبل



گام ۱۲. بیت‌های هدر:

گام ۱۳. خروجی همبستگی با سیگنال هدر: بخشی از خروجی همبستگی برای تأخیر برابر با ۰.۵ طول سمبل

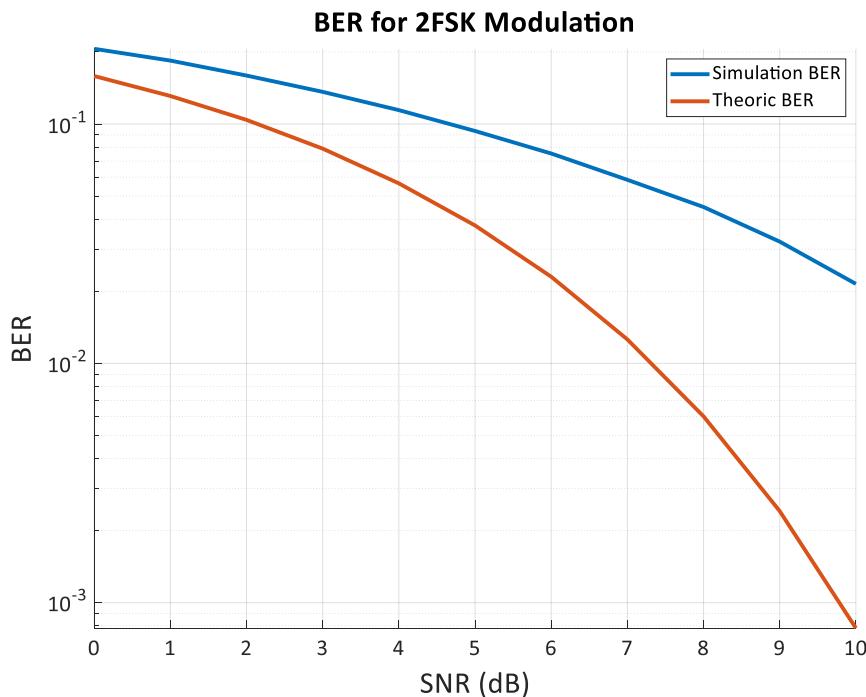


گام ۱۴.

اطمینان از یک بودن نمودار نرخ خطای بیت بعد از جبران سازی تأخیر کanal

اثر فاز کanal بر روی نمودار نرخ خطای بیت: صحبت‌سنجی نمودار خطای بیت با فاز کanal ۳۰ درجه با `pkt_size` برابر با 10^5

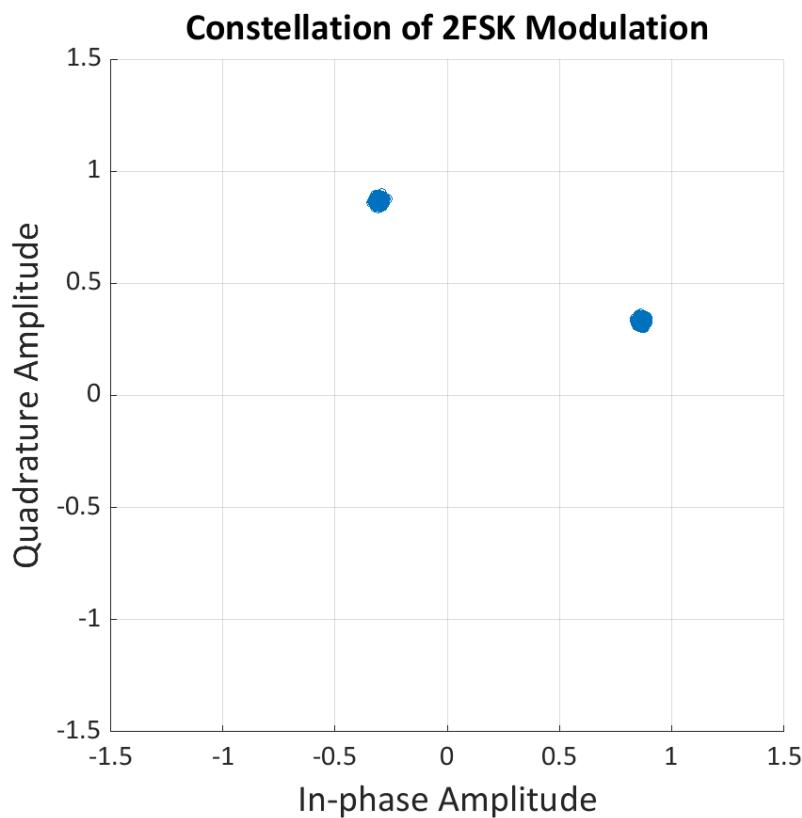
گام ۱۵.



گام ۱۶.

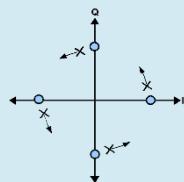
اثر فاز کanal بر روی منظومه سیگنالی مدولاسیون FSK: منظومه‌ها با `pkt_size` برابر با 10^5 و نسبت سیگنال به نویز

می‌باشد



گام ۱۷. اطمینان از یک بودن نمودار نرخ خطای بیت بعد از جبران سازی فاز کانال

آزمایش هفتم



آشکارسازی ناهمدوس



آزمایش ۷-۱: مقداردهی‌های اولیه

در این آزمایش بنا داریم تعدادی بسته‌ی داده را برای ارسال از طریق مدولاسیون FSK ناهمدوس و DBPSK آماده نماییم. از آن جا که در این آزمایش بناست این فرستنده با استفاده از رادیو نرم افزار ADALM-PLUTO پیاده‌سازی شود، می‌بایست با استفاده از پارامترها و ملاحظات عملی، این کار انجام شود.

پارامترهایی که در این پیاده‌سازی اهمیت دارد شامل نرخ نمونه‌برداری (`fs`)، تعداد نمونه‌های هر سمبل (`smpl_per_symbol`)، تعداد سمبل‌های ارسالی در بسته‌ی داده (`pkt_size`) و مدت زمان ارسال داده (`stop_time`) می‌باشد. در این آزمایش مقدار پارامترهای مورد نیاز به صورت **Error! Reference source not found.** می‌باشد.

جدول ۱ پارامترهای آزمایش ۷

پارامتر	<code>fs</code>	<code>smpl_per_symbol</code>	<code>pkt_size</code>	<code>stop_time</code>
مقدار	10MHz	FSK: 64 DBPSK: 8	1000000 1000	۱۰۰ ثانیه

دققت نظر داشته باشید که در این آزمایش نیز تنها می‌خواهیم برنامه‌هایی که در آزمایش قبل نوشته شده است را تکمیل نماییم.

۱. تولید M-File مربوط به مقدار دهی اولیه: قالب کلی پارامترهای مورد نیاز در این آزمایش، مانند آزمایش قبل است و فایل `mod_det_opt.m` می‌بایست پارامتری به نام `dcl_init.m` داشته باشد که نوع آشکارسازی را که می‌تواند همدوس (noncoherent) یا ناهمدوس (coherent) باشد را مشخص می‌نماید. برنامه با توجه به این گزینه نوع گیرنده و نحوه ارسال فرستنده مدیریت می‌شود. همچنین پارامتر `phase_amb_opt` برای نوع اعمال ابهام فاز در نظر گرفته شود. این فایل می‌بایست از برنامه‌ی اصلی فراخوانی نمایید.

آزمایش ۷-۲: پیاده‌سازی مدولاسیون FSK ناهمدوس

۱. شبیه‌سازی کanal با فاز تصادفی: برای شبیه‌سازی کanal با فاز تصادفی می‌بایست بردار `tx_smpl_delayed` را در `tx_smpl` ضرب `exp(1i*chnl_phase_offset_amb)` کرد. حاصل را در بردار `tx_smpl` قرار دهید. `chnl_phase_offset_amb` را به صورت یک بردار تصادفی بین 0 و 2π تولید کنید. طول این بردار برابر با طول بردار `tx_smpl` می‌باشد. دقتش کنید که فاز اضافه شده در طول هر سمبل ثابت باشد. در این حالت `phase_amb_opt` برابر با 0 است. (راهنمایی: از دستور `kron` می‌توان برای این کار استفاده نمود).

۲. شبیه‌سازی مدولاسیون FSK ناهمدوس: برای پیاده‌سازی این مدولاسیون، همانند FSK همدوس که در آزمایش قبل پیاده شد عمل نمایید. نمودار احتمال خطای بیت را برای مدولاسیون 4FSK و 2FSK به دست آورده و نتیجه‌ی را با دستور `berawgn` صحبت‌سنگی نمایید. دقت نمایید در حالت آشکارسازی ناهمدوس باید حداقل فاصله‌ی فرکانسی $T_s/1$ باشد. آشکارساز این مدولاسیون نیز به صورت $\arg \max |r_{\ell, m} - S_{\ell, m}|$ می‌باشد.

۳. پیاده‌سازی سخت‌افزاری مدولاسیون FSK ناهمدوس: با فرض این که توان فرستنده‌ی ADALM-PLUTO برابر با 0dBm و بهره‌ی گیرنده برابر با 20dB باشد و از آن‌تن به منظور ارتباط فرستنده و گیرنده استفاده می‌شود، خطای آشکارسازی با استفاده از آشکارسازی ناهمدوس را برای مدولاسیون 2FSK محاسبه نمایید. هم‌چنین در این حالت منظومه‌ی سیگنالی این مدولاسیون را نیز رسم نمایید. در این جا از هدر تنها برای به دست آوردن ابتدای بسته‌ی ارسالی استفاده می‌شود و هیچ گونه جبران‌سازی فازی صورت نمی‌پذیرد.

آزمایش ۷-۳: پیاده‌سازی مدولاسیون DBPSK

۱. تولید بیت: با استفاده از تابع `bit_gen` تعدادی بیت ۰ و ۱ را متناسب با طول بسته‌ی داده و با در نظر گرفتن پارامترهای مدولاسیون DBPSK تولید نموده و آن را درون ماتریس `b_tx` ذخیره نمایید. ماتریس `b_tx` یک ماتریس با اندازه‌ی `pkt_size × 1` می‌باشد.

۲. کد کردن بیت‌ها: با استفاده از **Error! Reference source not found.** رشته بیت‌های `e_n` را تولید نمایید و آن را در رون بردار `enc_b_tx` قرار دهید.

۳. نگاشت بیت به سمبل:

گام ۱. کدگذاری گری بیت‌های تولیدی: با وجود این که در این جانیازی به کدگذاری گری نیست ولی به منظور کلی بودن برنامه این مرحله نیز انجام می‌شود. با استفاده از تابع `gray_code`، ماتریس مربوط به کدگذاری گری مدولاسیون DBPSK را تولید کنید و آن را در ماتریس `b_gray` ذخیره کنید. بردار جدیدی متناظر با ماتریس `enc_b_tx` و با نام `sym_idx` تولید نمایید و نگاشتی یک‌به‌یک بین سطرهای ماتریس `b_gray` و ماتریس `enc_b_tx` برقرار کنید. به عبارتی سطر ایام بردار `sym_idx`، شماره‌ی سطروی از `b_gray` را نشان می‌دهد که برابر با سطر ایام ماتریس `enc_b_tx` باشد.

گام ۲. تولید سمبل‌های ارسالی: با استفاده از تابع `constellation`، تمامی سمبل‌های ارسالی مدولاسیون DBPSK را تولید کرده و سمبل‌های متناظر با هر سطر بردار `sym_idx` را تولید کنید. این سمبل‌ها را در بردار `mod_sym` ذخیره نمایید.

۴. شکل‌دهی پالس ارسالی: با استفاده از تابع `pulse_modulation` نمونه‌های ارسالی مربوط به یک بسته‌ی داده را آماده‌ی ارسال کرده و حاصل را در بردار `tx_smp1` ذخیره نمایید. تابع شکل‌دهی پالس را مثلثی در نظر بگیرید.

آزمایش ۷-۴: مدل‌سازی کانال

۱. افزودن تأخیر در کانال: برای شبیه‌سازی تأخیر در کانال به اندازه‌ی پارامتر `chnl_delay_in_smp1` به ابتدای بردار `tx_smp1` صفر اضافه نمایید. حاصل را درون بردار `tx_smp1_delayed` قرار دهید. دقت نمایید که طول بردار افزایش می‌باید و می‌بایست در ادامه‌ی برنامه اثر این افزایش لحاظ شود. در این جا فعلاً مقدار تأخیر برابر با صفر قرار دهید.

۲. اعمال ابهام فاز کانال: مدولاسیون DPSK به تعییرات شدید فاز حساس است و بنابراین باید ابهام فاز ایجاد شده خیلی زیاد نباشد. ابهام فاز ایجاد شده برای سمبل‌ها را به صورت زیر مدل می‌کنیم
`chnl_phase_offset_amb = 0:phase_step:(pkt_size-1)*phase_step`



پارامتر `phase_step`، مقدار ابهام فاز ایجاد شده برای یک سمبل نسبت به سمبل قبلی را مشخص می‌کند. حال با استفاده از دستور `kron`، فاز تولید شده را برای کل نمونه‌های هر سمبل تکرار کنید. با ضرب کردن عبارت `(tx_smp1_delayed * exp(1j * chnl_phase_offset_amb))` اضافه نمایید و حاصل را در `tx_smp1` ذخیره نمایید. در اینجا `phase_step` را برابر با 10° درجه قرار دهید.

۳. شبیه‌سازی کانال با نویز سفید گاووسی

گام ۱. تعیین واریانس نویز بر اساس نسبت سیگنال به نویز (E_b/N_0): ابتدا بر اساس خروجیتابع `constellation`، مقدار متوسط انرژی سمبل را به دست آورید (`Es_avg`). حال با استفاده از متوسط انرژی سمبل، مقدار انرژی متوسط بیت را به دست آورده و برابر متغیر `Eb` قرار دهید. مقدار E_b/N_0 را برابر با 10dB در نظر بگیرید (به عبارتی پارامترهای `snr_max` و `snr_min` برابر 10 تنظیم می‌شوند). سپس واریانس نویز را بر اساس نسبت E_b/N_0 به دست آورید و آن را درون متغیر `var_noise` قرار دهید.

گام ۲. افزودن نویز به سیگنال: با استفاده از تابع `randn` یک بردار نویز مختلط با واریانس `var_noise` و ابعاد برابر با `tx_smp1` تولید نمایید و آن را `noise_smp1` نامگذاری نمایید. سپس این بردار را با بردار `tx_smp1` جمع نموده و حاصل را `tx_smp1_noise` بنامید.

آزمایش ۷-۵: پیاده‌سازی گیرنده‌ی مدولاسیون DBPSK

در ادامه‌ی فایل آزمایش قبل می‌خواهیم گیرنده‌ی مدولاسیون DBPSK را پیاده‌سازی نماییم.

۱. دمودولاسیون یا آشکارسازی پالس: نمونه‌های ارسال شده به همراه نویز (`tx_smp1_noise`) را به عنوان نمونه‌های دریافتی درون بردار `rx_smp1` قرار دهید. با استفاده از تابع `pulse_demodulation` با ورودی‌های نمونه‌های سیگنال دریافتی (`rx_smp1`)، نام مدولاسیون (`modulation`، مرتبه‌ی مدولاسیون (`M`)، نرخ نمونه‌برداری (`fs`)، تعداد نمونه‌های هر سمبل (`smp1_per_symbol`)، نام تابع شکل‌دهنده‌ی پالس (`pulse_name`)، روش تشخیص قدرت سیگنال دریافتی در راستای سیگنال‌های پایه (`mode`) و شبیه‌ی آشکارسازی (`det_opt`، سمبل‌های دریافتی (`rx_sym`) و اندیس سمبل‌های آشکارشده (`det_sym_idx`) را به دست آورید.

۲. محاسبه‌ی خطای سمبل: با مقایسه‌ی تعداد اختلاف‌های بردارهای `sym_idx` و `det_sym_idx` و محاسبه نسبت اختلاف این دو بردار به تعداد کل سمبل‌ها، خطای سمبل را به دست آورده و آن را در متغیر `ser` قرار دهید.

۳. تبدیل سمبل و بیت و تبدیل از کدگذاری گری به کدگذاری باینری: بیت‌های متناظر با سطرهای `det_sym_idx` را به دست آورید و آن را درون متغیر `det_bit_gray` قرار دهید. با وجود این‌که در این‌جا عکس تبدیل گری مطرح نیست به منظور کلی ماندن برنامه بیت‌های گری را به کدگذاری باینری تبدیل نماید و آن را درون بردار `det_bit` قرار دهید.

۴. محاسبه‌ی خطای بیت: با مقایسه‌ی تعداد اختلاف ماتریس‌های `tx_b` و `det_bit` و محاسبه نسبت تعداد اختلاف این دو ماتریس به تعداد کل بیت‌ها، خطای بیت را به دست آورده و آن را در متغیر `ber` قرار دهید.

آزمایش ۷-۶: خواسته‌های کلی

۱. رسم نمودار نرخ خطای بیت: با تغییر نسبت سیگنال به نویز بین 0 تا 10dB نرخ خطای بیت را برای مدولاسیون‌های DBPSK به دست آورده و در یک نمودار رسم نمایید و آن را با خروجی تابع `berawgn` مقایسه نمایید. این کار را برای شکل موج مثلثی انجام دهید.

۲. اثر تغییرات فاز بر روی عملکرد: با تغییر پارامتر `phase_step`، تأثیر تغییرات فاز را بر روی عملکرد این سامانه ارزیابی نمایید.



آزمایش ۷-۷: پیاده‌سازی سخت‌افزاری مدولاسیون DBPSK

۱. ارسال نمونه‌ها با استفاده از ADALM-PLUTO: بردار `tx_smp1` مربوط به مدولاسیون DBPSK (هدر اضافه شود) را با استفاده

از دستور `transmitRepeat` به مدت زمان `stop_time` ثانیه به صورت پی‌درپی در فضا ارسال نمایید. این دستور داده‌ها را از طریق USB به رادیو نرم‌افزار ارسال می‌نماید و در حافظه‌ی سخت‌افزار ذخیره‌می‌نماید. فرستنده داده‌ها را از حافظه‌ی رادیونرم‌افزار قرایت کرده و مدام ارسال می‌نماید. برای این منظور ابتدا می‌بایست پس از پیکربندی رادیو نرم‌افزار یک شیء فرستنده ایجاد نموده و فرکانس مرکزی آن را بر روی 2400MHz تنظیم کرده و بهره‌ی آن را برابر با 0dB تنظیم نمایید.

۲. دریافت سیگنال با استفاده از ADALM-PLUTO: ابتدا شیء مربوط به گیرنده را تعریف نمایید. فرکانس مرکزی آن را بر روی

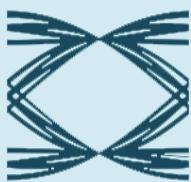
2400MHz تنظیم کرده و بهره‌ی آن را به صورت Manual و برابر 20dB تنظیم نمایید. به علت اطمینان از دریافت یک بسته‌ی کامل، تعداد نمونه‌های زمانی دریافتی را برابر با دو برابر تعداد نمونه‌های زمانی بسته درنظر بگیرید. به منظور دریافت داده، شیء گیرنده را فراخوانی نمایید و حاصل را درون متغیر `rx_smp1` قرار دهید.

۳. دمودولاسیون و مشاهده‌ی منظومه‌ی سیگنالی: مشابه قبل عمل دمودولاسیون را انجام داده و منظومه‌ی سیگنالی مربوط به

را رسم نمایید و مشاهدات خود را یادداشت نمایید. منظومه‌ی سیگنالی باید بتواند به صورت به‌لحظه به‌روز شود.



- [1] [1] R. W. Stewart, K. Barlee, L. Crockett, and D. Atkinson, *Software Defined Radio using MATLAB & Simulink and the RTL-SDR*. 2015.



آزمایش هشتم

انتقال دیجیتال از درون کانال باند محدود AWGN



آزمایش ۸-۱: مقداردهی‌های اولیه

در این آزمایش بنا داریم تعدادی بسته‌ی داده را برای ارسال از طریق مدولاسیون 2PAM آماده نماییم. از آن‌جا که در این آزمایش بناست این فرستنده با استفاده از رادیو نرم‌افزار ADALM-PLUTO پیاده‌سازی شود، می‌بایست با استفاده از پارامترها و ملاحظات عملی، این کار انجام شود.

پارامترهایی که در این پیاده‌سازی اهمیت دارد شامل نرخ نمونه‌برداری (`fs`)، تعداد نمونه‌های هر سمبل (`smp1_per_symbol`)، تعداد سمبل‌های ارسالی در بسته‌ی داده (`pkt_size`) و مدت زمان ارسال داده (`stop_time`) می‌باشد. در این آزمایش مقدار پارامترهای مورد نیاز به صورت **Error! Reference source not found.** می‌باشد.

جدول ۲ پارامترهای آزمایش ۷

<code>stop_time</code>	<code>pkt_size</code>	<code>smp1_per_symbol</code>	<code>fs</code>	پارامتر
۱۰۰ ثانیه	شبیه‌سازی: ۱۰۰۰۰۰ سخت‌افزار: ۱۰۰۰	32	10MHz	مقدار

دقت نظر داشته باشید که در این آزمایش نیز تنها می‌خواهیم برنامه‌هایی که در آزمایش قبل نوشته شده است را تکمیل نماییم.

آزمایش ۸-۲: پیاده‌سازی مدولاسیون 2PAM

۱. شبیه‌سازی مدولاسیون 2PAM: برای پیاده‌سازی این مدولاسیون، همانند آزمایش ۴ یا ۵ عمل نمایید. احتمال خطای بیت را

برای نسبت $\frac{E_b}{N_0}$ برابر با 10dB به دست آورید. این عمل را برای دو شکل پالس مستطیلی و Root Raised Cosine ($\beta = 0.9$) و

(`span_in_symbol = 6`) انجام دهید. (حالت عملکردی دمدولاتور را بر روی فیلتر منطبق تنظیم نمایید).

۲. رسم نمودار چشمی: با استفاده از دستور `eyediagram` نرم‌افزار MATLAB نمودار چشمی مدولاسیون 2PAM را برای دو

شكل پالس گفته شده و در دو نسبت E_b/N_0 برابر با 10dB و 40dB رسم نمایید. پارامترهای بر روی **Error! Reference source not found.** را به دست آورید.

آزمایش ۸-۳: مدل‌سازی کانال باندباریک

۱. تولید یک فیلتر FIR: با استفاده از دستور `fir1` نرم‌افزار MATLAB یک فیلتر با پهنای باند 300kHz و با تعداد ۱۰۰ تپ تولید

نمایید. پاسخ فرکانسی و پهنای باند نویز فیلتر را به دست آورید و با پهنای باند سیگнал ارسالی مقایسه نمایید.

۲. کانال باند محدود: پس از اعمال تأخیر و اعمال ابهام فاز، سیگنال ارسالی حاصل را با استفاده از عمل کانولوشن از فیلتر FIR مرحله‌ی قبل عبور دهید. مجدد نمودار چشمی را در دو نسبت E_b/N_0 برابر با ۱۰dB و ۴۰dB و ۴۰dB نمایید. احتمال خطای بیت را نیز به دست آورید. این عمل برای هر دو شکل پالس گفته شده انجام شود.

آزمایش ۸-۴: پیاده‌سازی کانال باند محدود

۱. کار با ADALM-PLUTO: موارد گفته شده را با استفاده از رادیونرم افزار ADALM-PLUTO نیز انجام دهید.

آزمایش ۸-۵: خواسته‌های کلی

۱. اثر طول شکل پالس: با تغییر پارامتر `span_in_symbol`: تأثیر آن را بر روی نمودار چشمی و خطای بیت مدولاسیون ۲PAM با شکل پالس `root raised cosine` را مشاهده نمایید.



- [1] R. W. Stewart, K. Barlee, L. Crockett, and D. Atkinson, *Software Defined Radio using MATLAB & Simulink and the RTL-SDR*. 2015.