



به نام خدا



۸۱۰۱۹۶۶۸۸

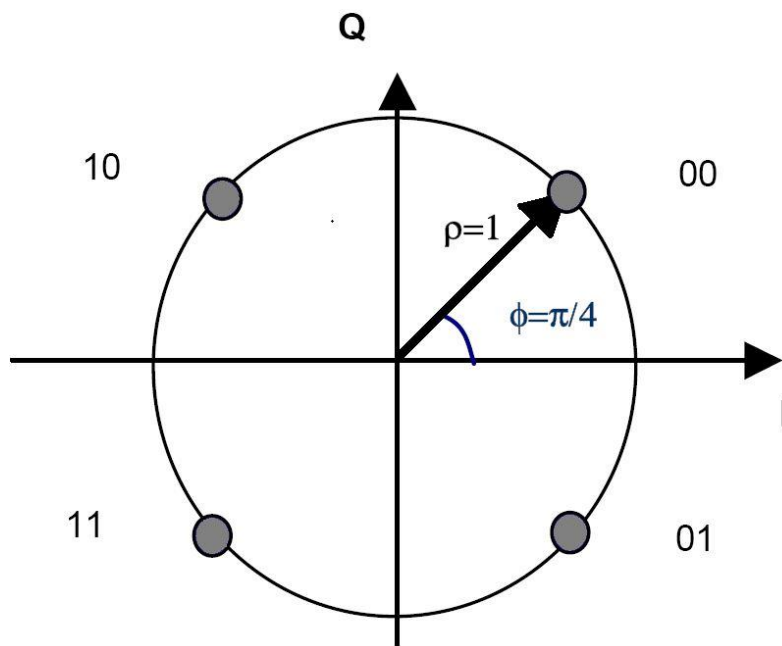
تمرین کامپیوتری اول

علی بهاری

در این تمرین کامپیوتری به شبیه سازی کانال بیسیم می پردازیم در ابتدا با مدولاسیون QPSK پیش می رویم و بعد از بدست آوردن نتایج با این مدولاسیون از مدولاسیون دیگر به نام 16QAM و همچنین روش کدینگ کانال Hamming (7,4) نیز استفاده خواهیم کرد.

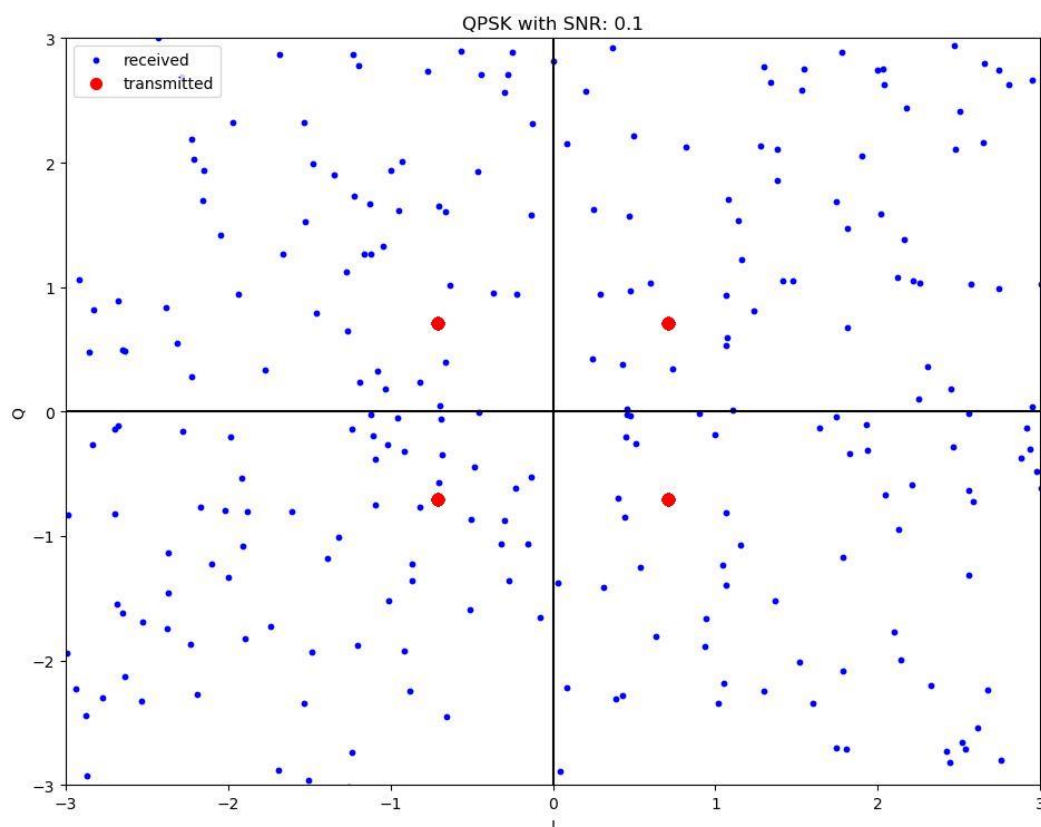
QPSK

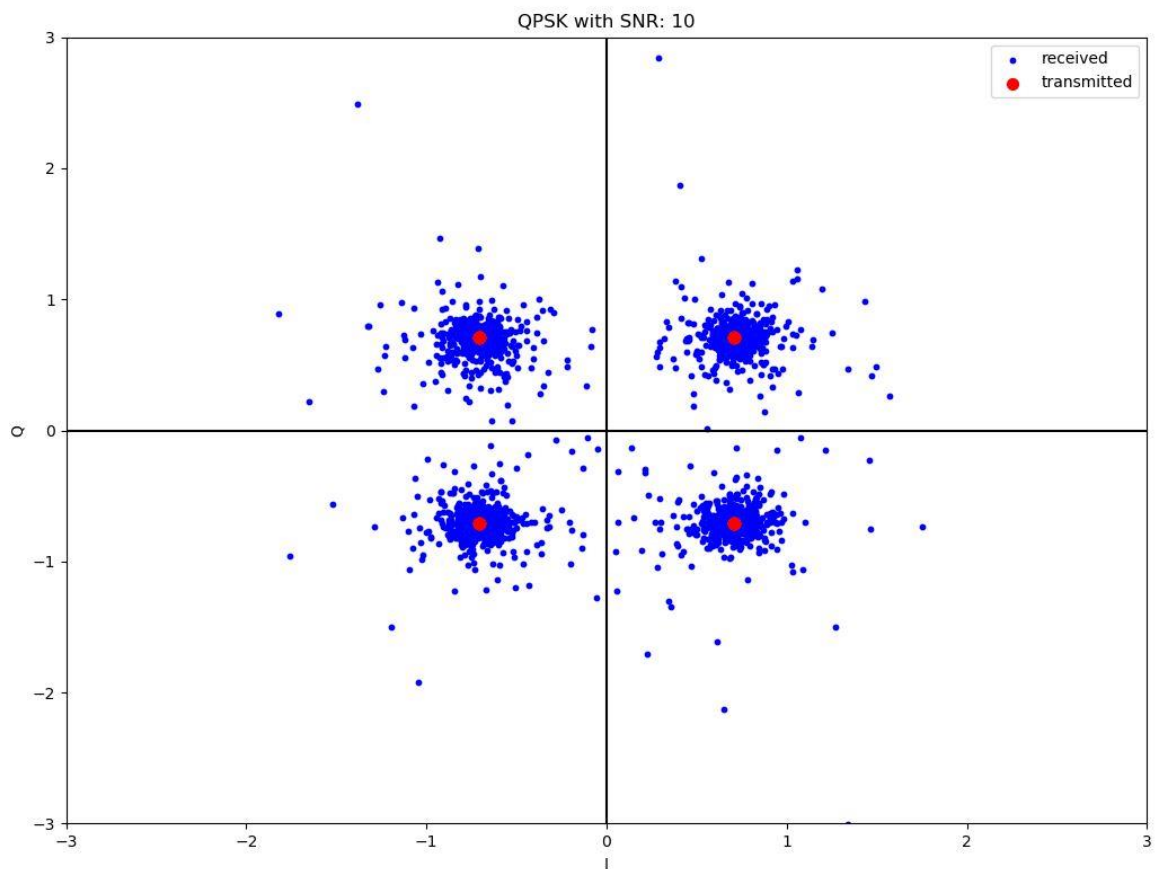
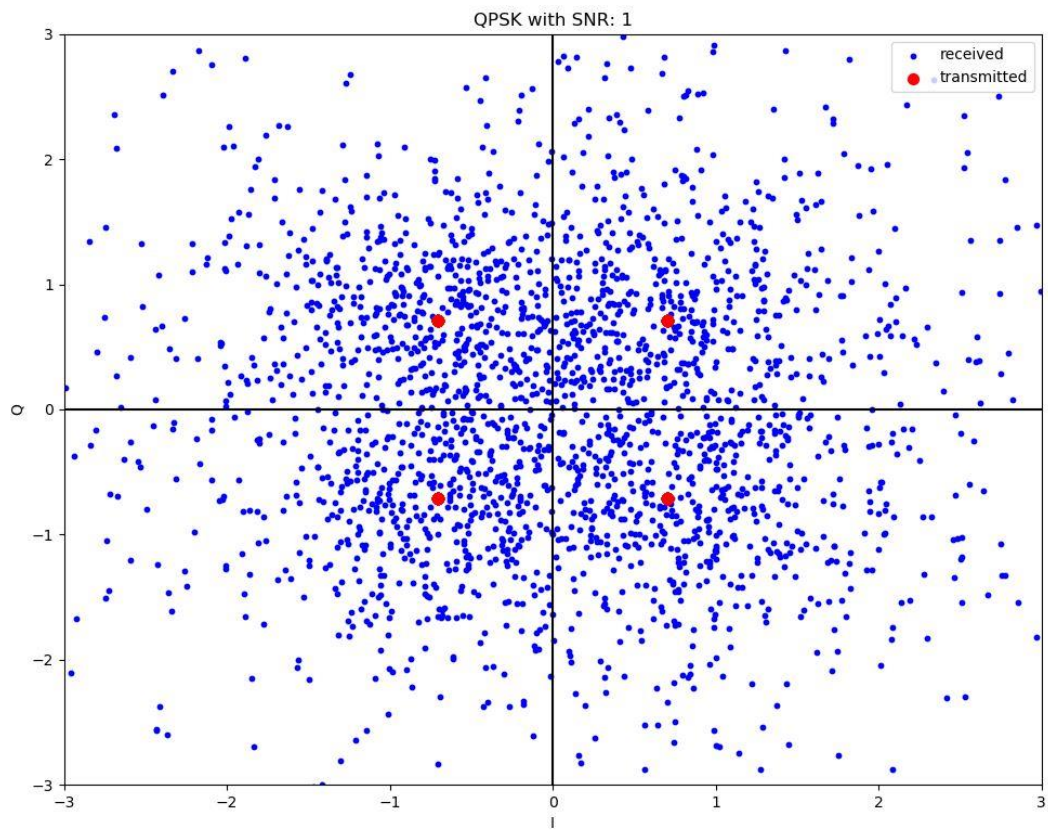
در این مدولاسیون ابتدا دیتای ارسالی که ۵۰۰۰ بیت در نظر گرفته شده را دو بیت دو بیت جدا کرده و طبق شکل زیر به هر دو بیت عدد مختلط متناظر را نسبت می دهیم.

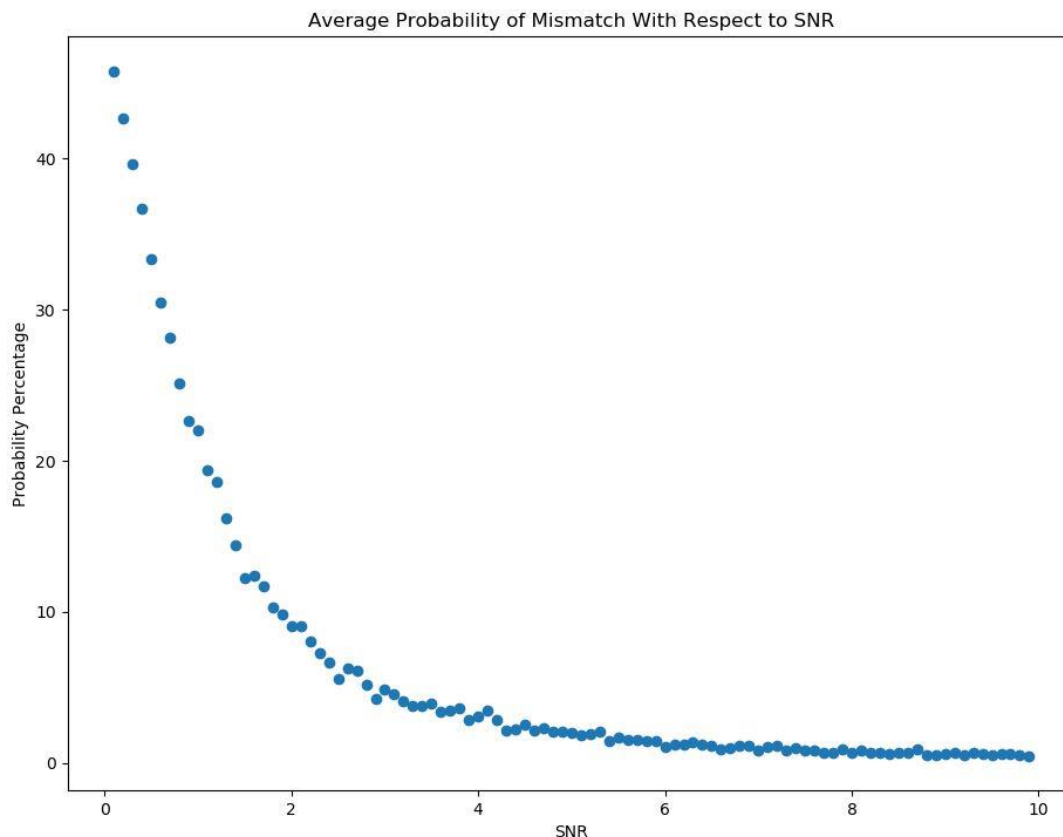


با این کار بیت های ما به فرمت مورد نیاز در آمده اند و آماده ارسال از فرستنده می باشند. با ارسال، gain کانال به این سیگنال با ضرب کردن مقدار gain اضافه می شود (generateTransmitterOutputSignal) حاصل تحت تاثیر نویز نیز قرار می گیرد و در نهایت سیگنال به گیرنده می رسد (generateReceiverInputSignal). در گیرنده ابتدا با تقسیم gain

کانال بر سیگنال دریافتی اثر آن را خنثی می کنیم (removeGainFromReceiverInputSignal) سپس سیگنال دریافتی را decode کرده و دیتا را به دست می آوریم (generateDecodedSignal). برای اختصاص دادن هر دو بیت به عدد مختلط در ابتدای کار و همچنین decode کردن سیگنال نهایی تغییراتی بر روی داده و سیگنال دریافتی داده شده است که در کد قابل مشاهده است. برای ML در قسمت demodulate حدود در نظر گرفته شده است که اگر از آن مقدار کم تر بود بیت صفر می شود در غیر این صورت یک می شود. دلیل استفاده از این موضوع این است که نویز تاثیر خود را بر روی سیگنال می گذارد و این موضوع باعث می شود تقسیم کردن بر gain کانال دقیقا همان مقدار فرستاده شده را ندهد برای همین از ML به شکلی که گفته شد یعنی با تعیین حدود و مقداردهی به بیت ها با توجه به آن ها استفاده شده است. در نهایت نیز داده دریافتی و ارسالی در نمودارهایی با SNR های مختلف رسم شده اند. همچنین برای احتمال خطا نیز تعداد عدم تطابق ها بین داده دریافتی و ارسالی چک شده و درصد خطا در نموداری نشان داده شده است.

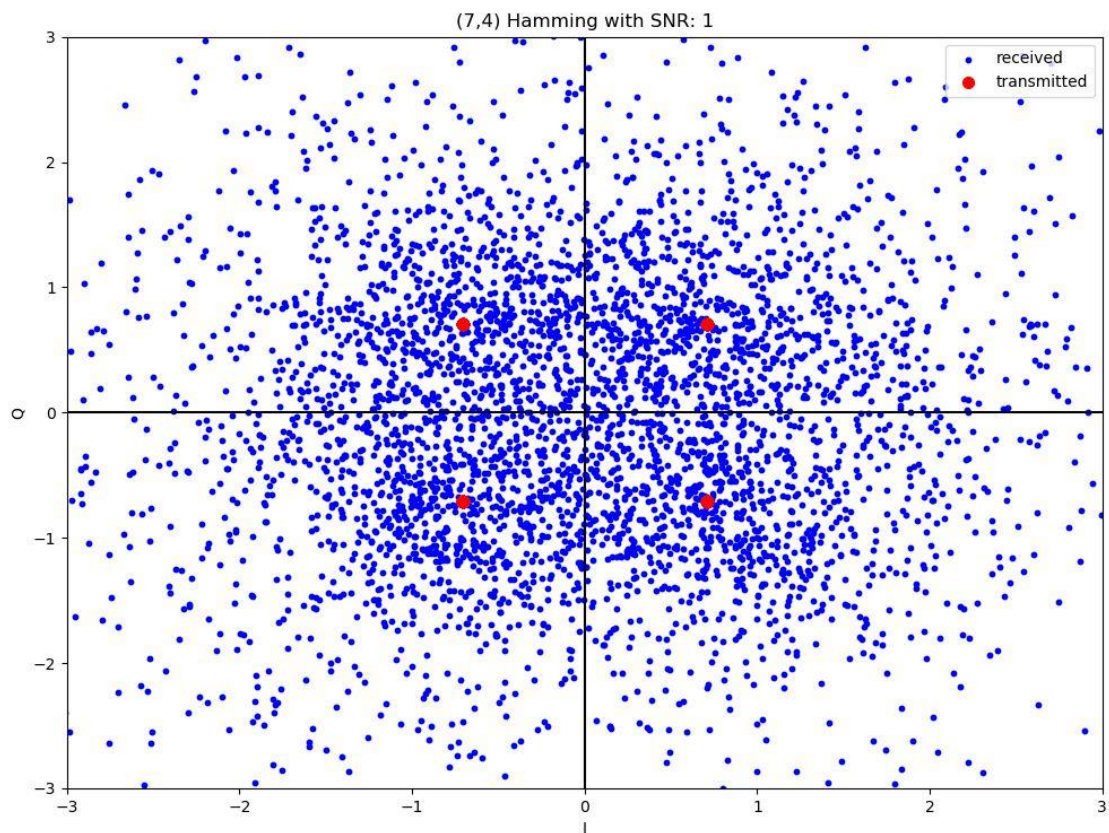
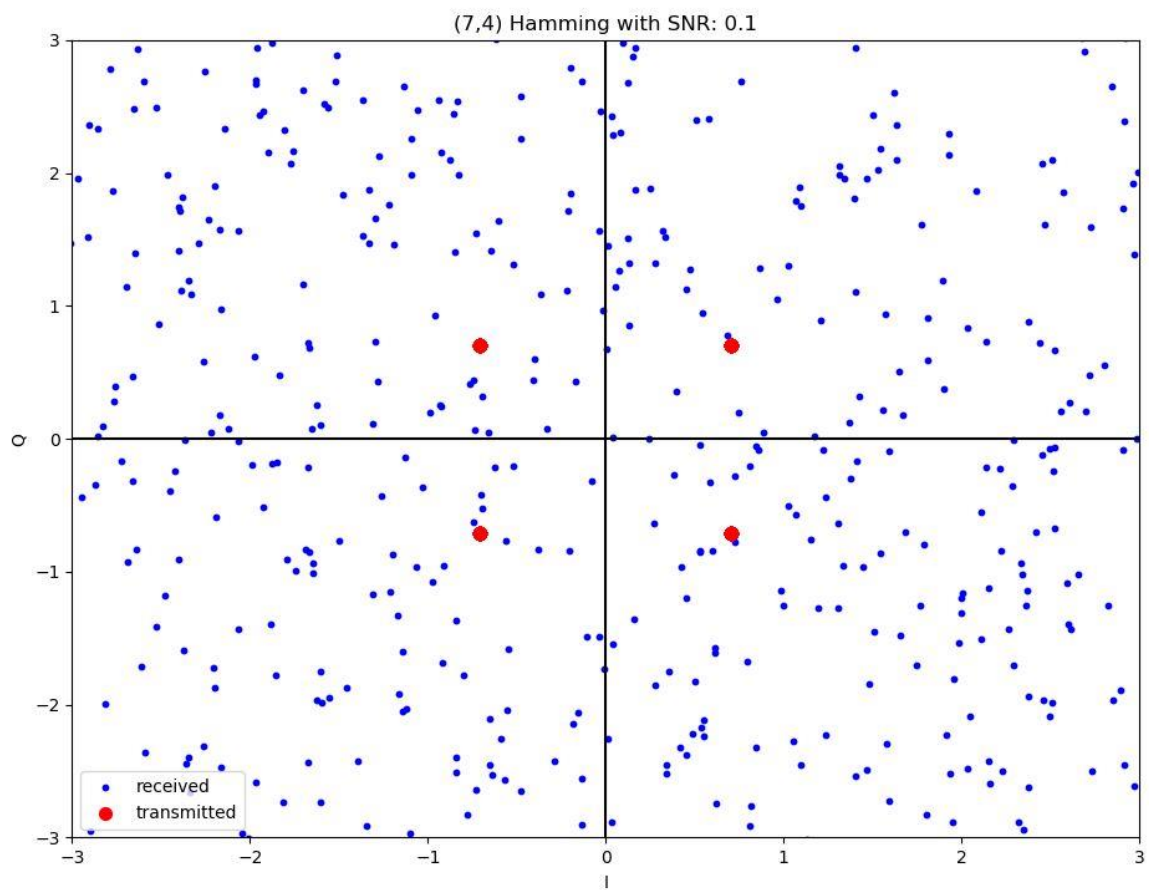


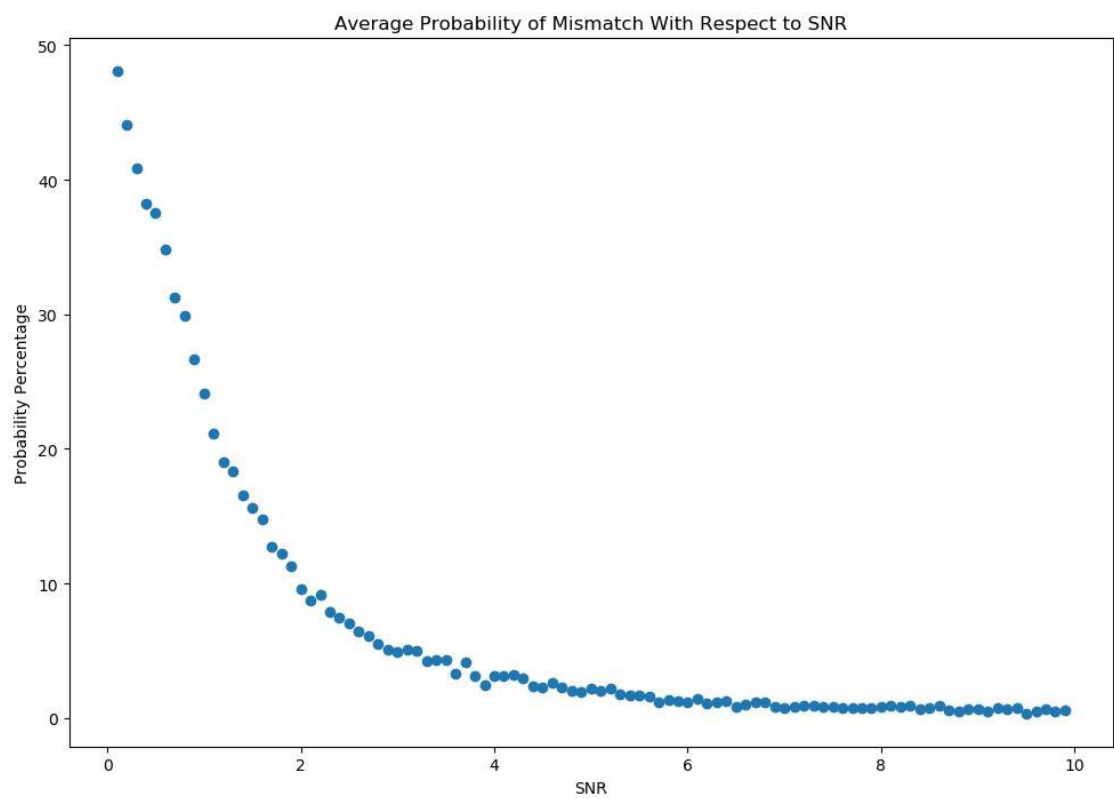
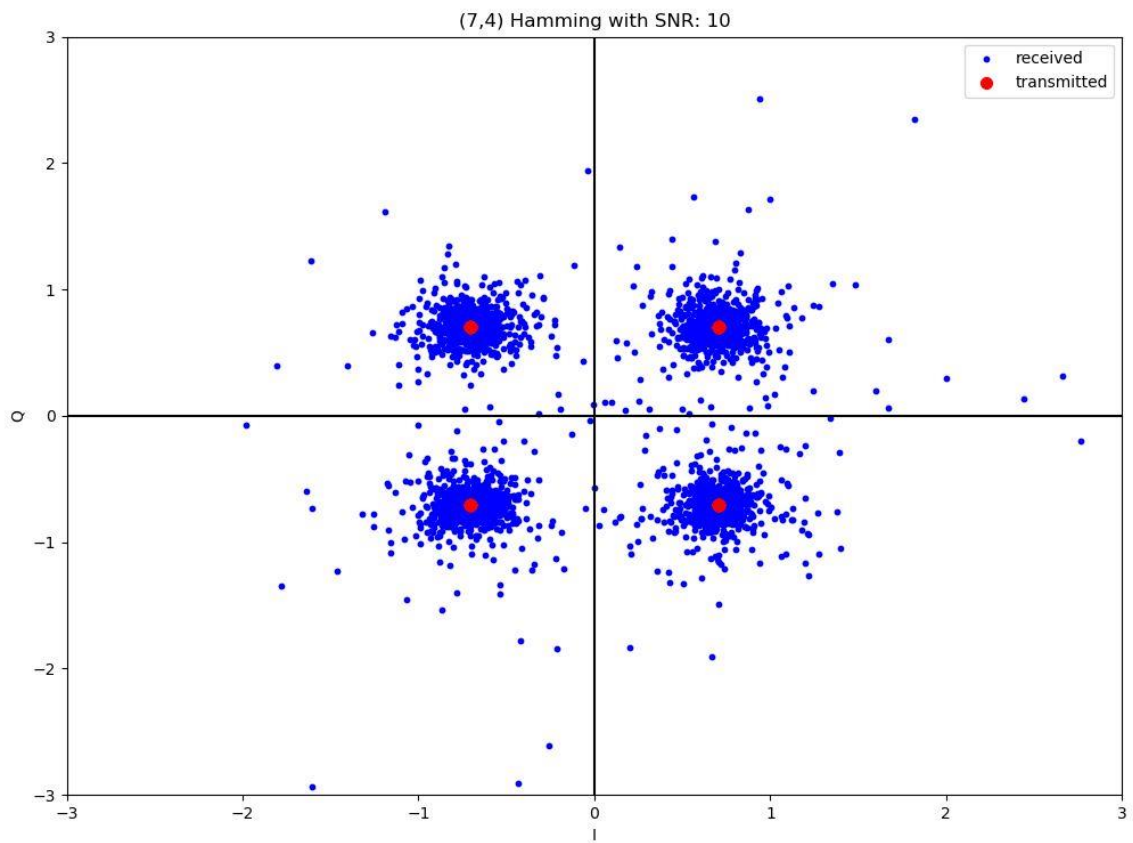




(7, 4) Hamming

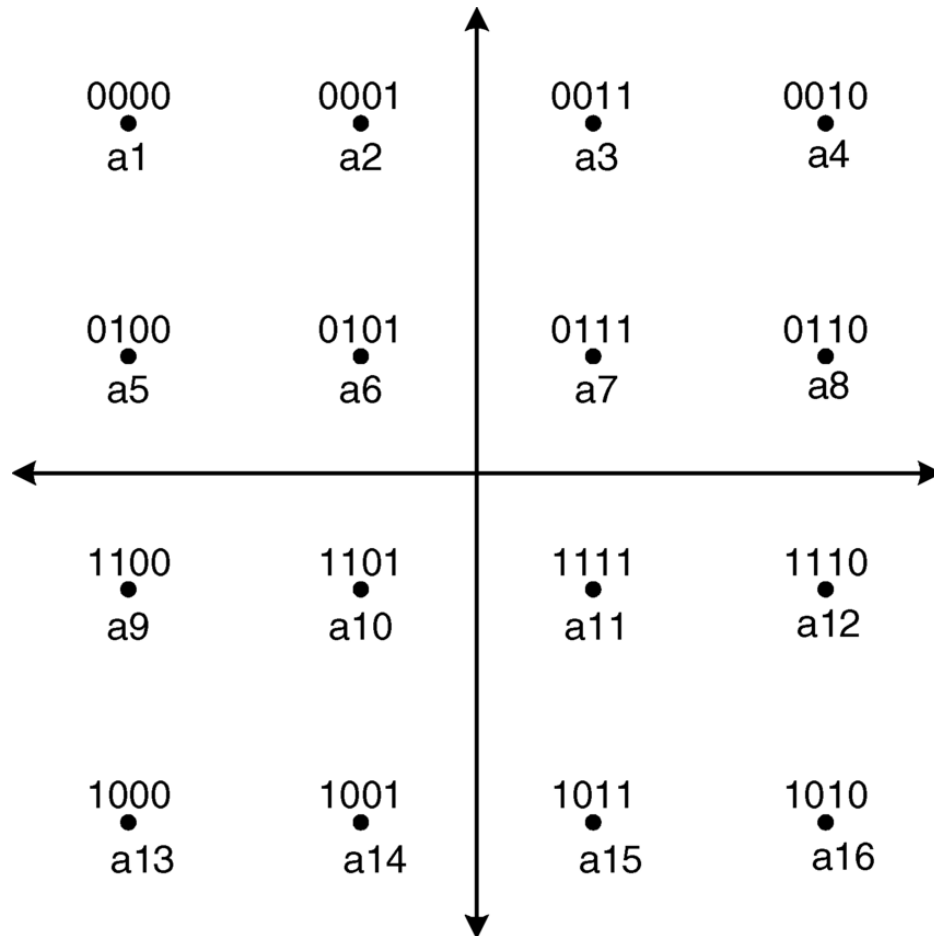
در این مرحله مانند QPSK عمل می کنیم فقط قبل از دادن داده اولیه به روند قبل hamming encoding و بعد از demodulate کردن در گیرنده hamming decoding انجام می دهیم برای encoding و decoding از روش گفته شده در لینک گفته شده در انتهای گزارش استفاده شده است. پیاده سازی آن نیز در کد قابل مشاهده است. با این روش بیت های دارای خطا پیدا شده و درست می شوند همچنین هر ۴ بیت داده در نهایت ۷ بیت ارسال می شود و یعنی ۳ بیت به عنوان redundancy مورد استفاده قرار گرفته است و در decode کردن نیز هر ۷ بیت به ۴ بیت تبدیل خواهد شد. در نهایت مانند قبل داده های ارسالی و دریافتی به ازای SNR های گفته شده در نمودار هایی نمایش داده شده اند و همچنین احتمال خطا با توجه به مقدار SNR های مختلف نیز در نمودار نمایش داده شده است.



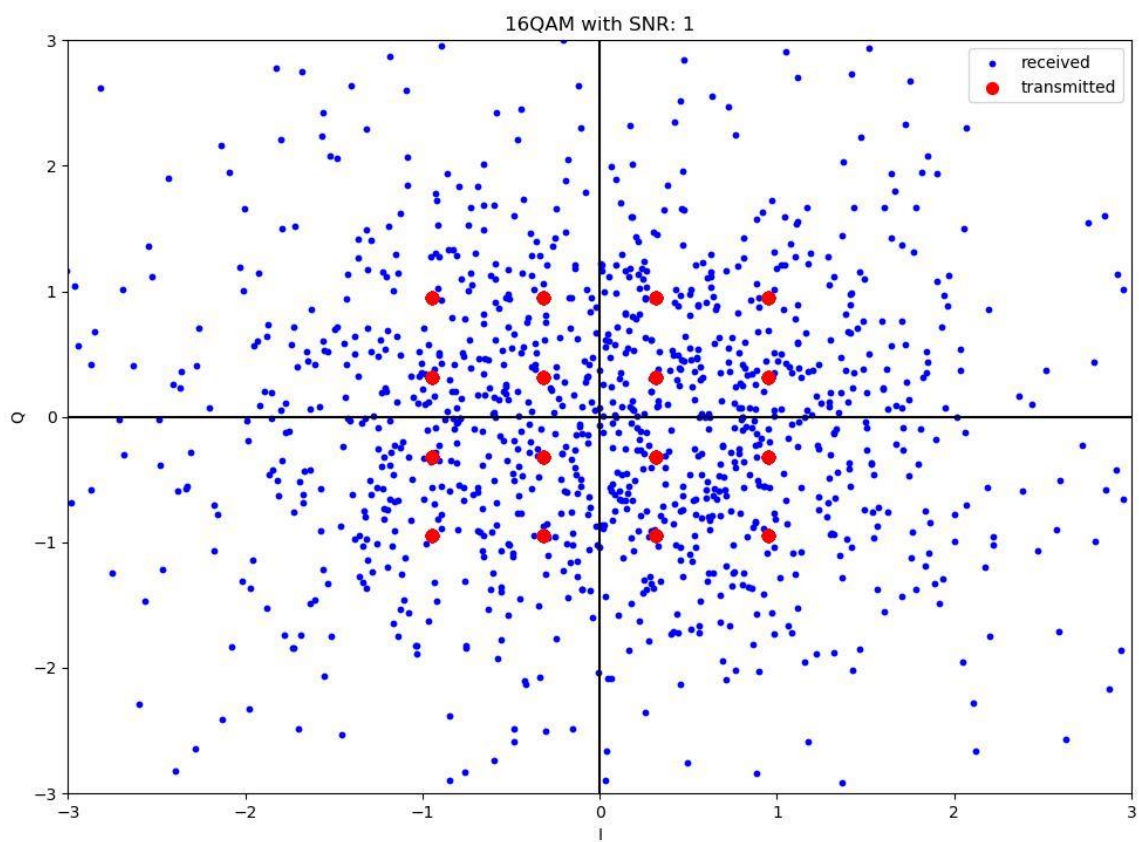
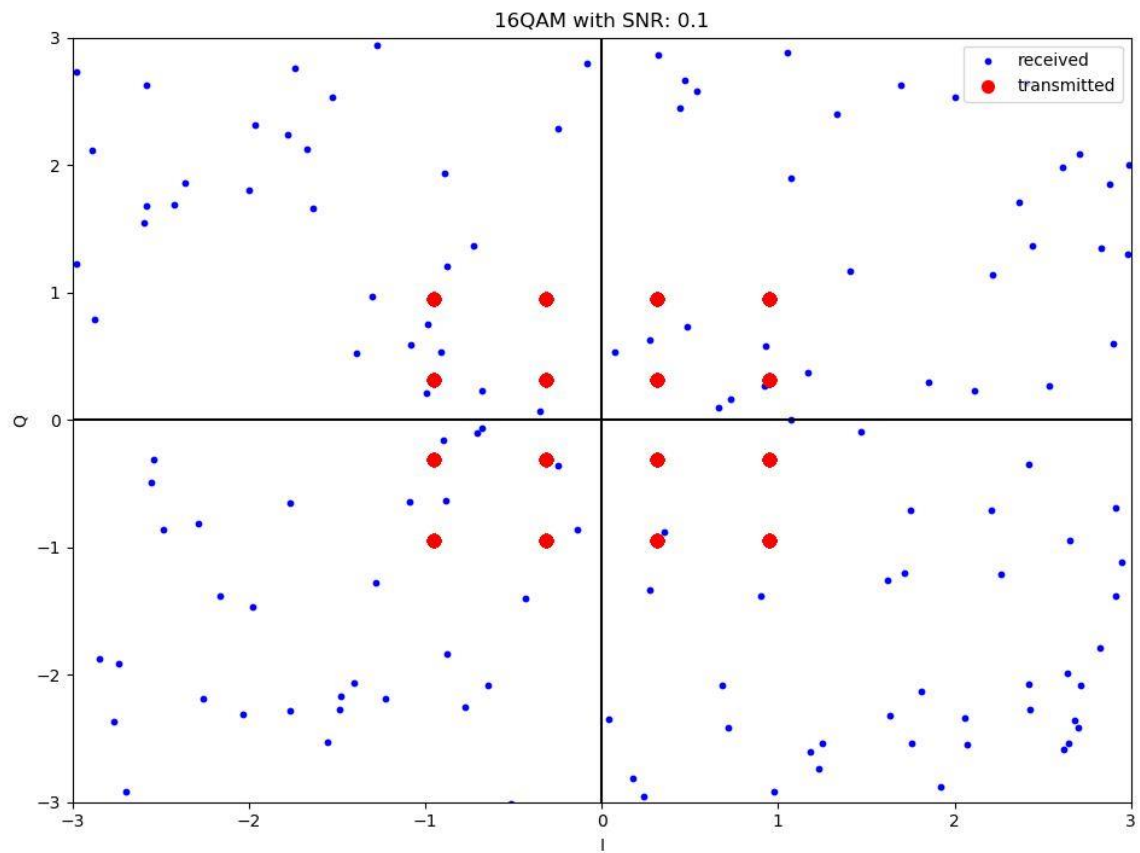


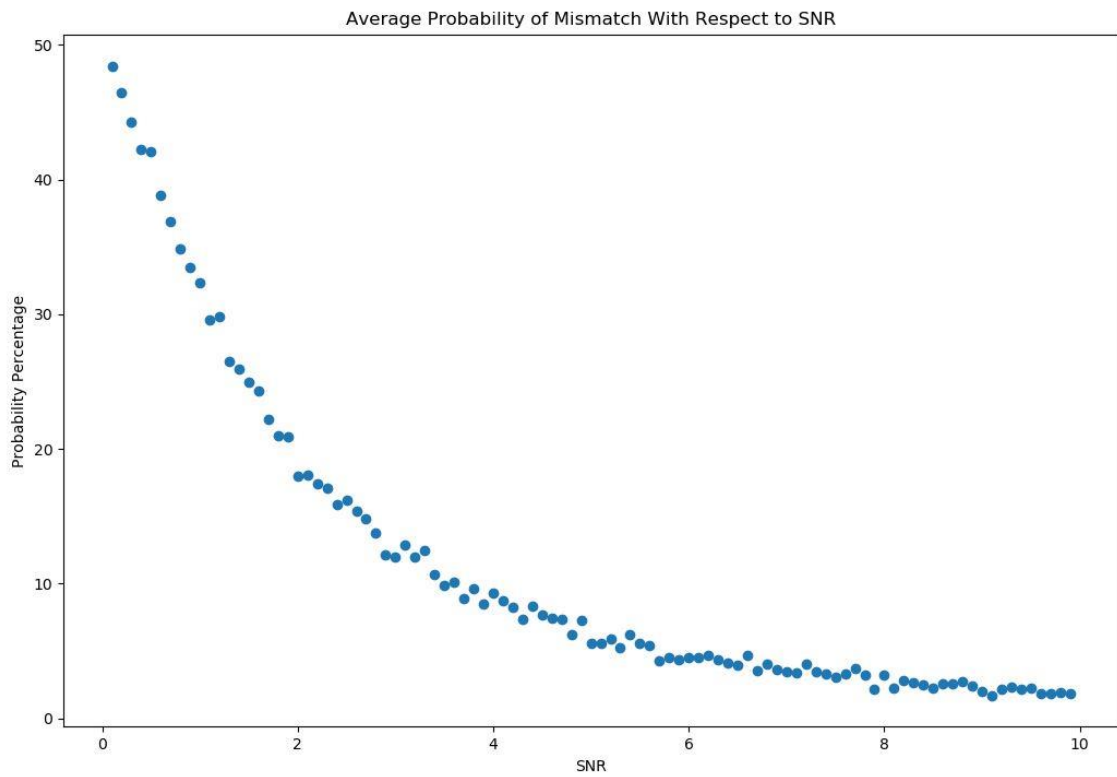
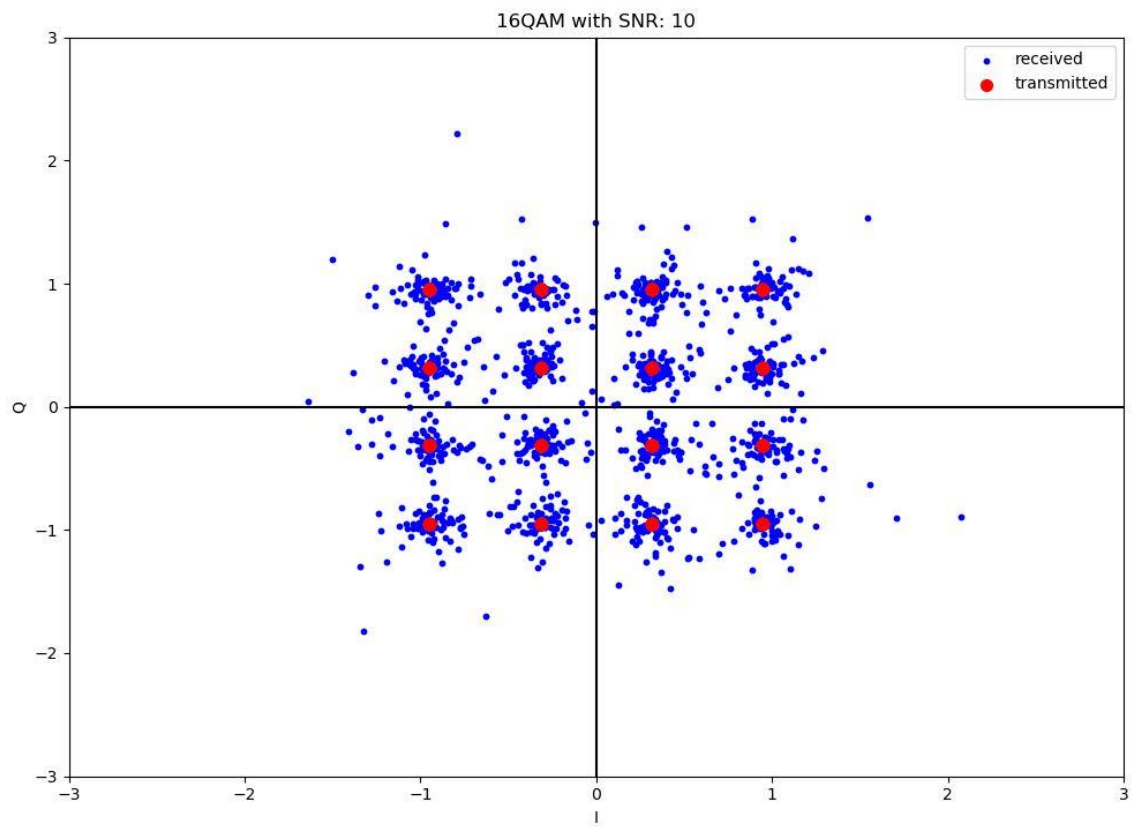
16QAM

این مدولاسیون حدوداً مانند مدولاسیون QPSK می باشد. در این جا از ۱۶ نقطه استفاده می شود که این بار هر ۴ بیت به یک نقطه نسبت داده می شود. این نسبت دادن در شکل زیر نمایش داده شده است.



در این جا هم تغییراتی بر روی داده و سیگنال دریافتی برای نسبت دادن هر نقطه که یک عدد مختلط می باشد به هر ۴ بیت انجام شده است که در کد قابل مشاهده می باشد. روند این تغییرات تا حدودی مانند قسمت QPSK می باشد. نتایج این قسمت هم مانند قسمت های دیگر در نمودار ها نمایش داده شده است. همان طور که در شکل بالا قابل مشاهده است در این روش برای مقادیر حقیقی و موهومی بر خلاف QPSK به جز ۱ و ۱- از ۳ و ۳- نیز استفاده می شود.





برای تولید متغیرهای رندوم نرمال از کتابخانه `numpy` استفاده شده است.

به دلیل این که SNR با واریانس متغیرهای نرمال رابطه عکس دارد با افزایش مقدار SNR واریانس کمتر می شود و این به این معناست که پراکندگی در نویز و `gain` کانال کمتر می شود بنابراین داده دریافت شده در حالتی که SNR بیشتر است به داده ارسال شده نزدیکتر است و همان طور که در نمودارها نیز دیده می شود مقدار عدم تطابق بین داده ارسالی و دریافتی در SNR های بالاتر کمتر می باشد.

منبع استفاده شده برای `encode` و `decode` کردن قسمت Hamming :

[https://en.wikipedia.org/wiki/Hamming\(7,4\)](https://en.wikipedia.org/wiki/Hamming(7,4))