



به نام خدا



موضوع: گزارش پروژه درس تئوری کدینگ

استاد: دکتر صباغیان

دانشجو: سجاد علیخانی

شماره دانشجویی: ۸۱۰۱۹۹۲۲۶

نیمسال اول ۱۳۹۹-۱۴۰۰

سوال ۱ (+ امتیازی) :

بخش اول:

در این بخش با وارد کردن هر آرایه حاوی بیت به طول مضربی از ۱۱ در خروجی یک آرایه با طول مضربی از ۱۵ دریافت میکنیم که همان پیام کد شده با استفاده از کدینگ Hamming(15,11) می باشد. با توجه به ماتریس پرتی چک این کد، می دانیم حداقل فاصله همینگ برابر ۳ می باشد بنابراین با توجه به رابطه ی $t = \left\lfloor \frac{d_{min}-1}{2} \right\rfloor$ حداکثر قابلیت تصحیح خطای این کد برابر یک بیت در بین ۱۱ بیت ورودی است. شکل ۱-۱ نمونه ای از یک آرایه بیت ورودی و خروجی را نشان می دهد.

```
ENTER AN ARRAY OF BITS WITH LENGTH OF MULTIPLE OF 11 TO ENCODE: [1,0,0,1,0,1,1,1,0,1,0,1,0,0,0,1,1,0,1]
ENCODED MESSAGE: [0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1]
```

شکل ۱-۱

بخش دوم:

در این بخش با وارد کردن هر آرایه حاوی بیت به طول مضربی از ۱۵ در خروجی در صورت عدم وجود هیچ خطای قابل آشکارسازی، پیام دیکد شده به نمایش در می آید و در صورت بروز حداقل یک خطای قابل آشکارسازی، هشدار وجود خطا ظاهر می شود. شکل ۱-۲-۱ و ۲-۲-۱ نمونه ای از کارکرد این کد را نشان می دهد.

```
ENTER AN ARRAY OF BITS TO DECODE: [0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1]
DECODED MESSAGE: [array([1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0]), array([1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1])]
```

شکل ۱-۲-۱

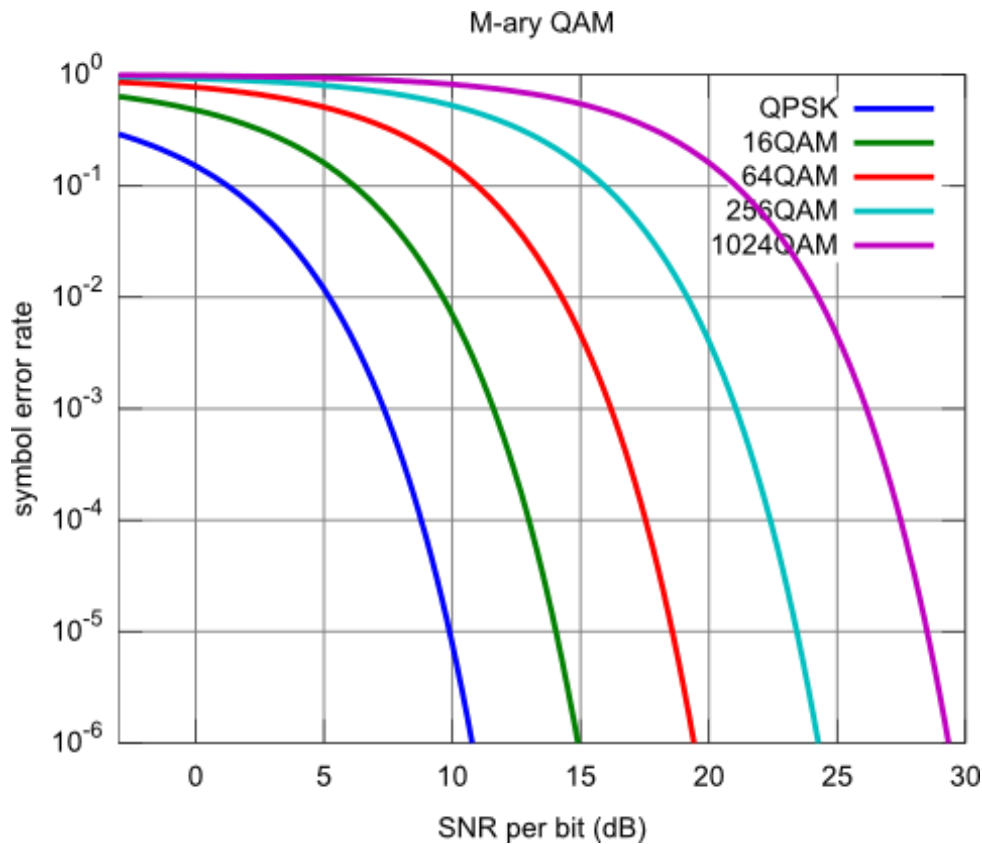
```
ENTER AN ARRAY OF BITS TO DECODE: [0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0]
ERROR: THE MESSAGE IS CORRUPTED DUE TO CHANNEL NOISE
```

شکل ۲-۲-۱

بخش سوم:

در شبیه سازی انتشار موج های رادیویی میتوان نتایج را با راه های متفاوت مدل کرد. معیارهایی نظیر نرخ خطای بیت (Bit Error Rate)، پهنای باند، مکان انتشار موج، فرکانس و دما معیارهای مناسبی برای شبیه سازی هستند. معمولاً معیارهای نرخ خطای بیت و پهنای باند در سیستم های مخابراتی از اهمیت بیشتری برخوردارند و همواره سعی بر این است که BER و پهنای باند کمی داشته باشیم. اما یکی از این دو معیار باید طی یک Trade-off به

دیگری ترجیح داده شود. برای مثال برای کم کردن سطح خطا در یک توان ارسالی مشخص، باید از کدینگ های مختلف استفاده کنیم و این به معنی اضافه کردن بیت های کمکی به رشته بیت مورد نظر است که باعث افزایش پهنای باند اشغالی می شود. اما گاهی اوقات مخصوصا زمانی که برای آشکارسازی سیگنال های ارسالی یک حد آستانه ی توان داریم مجبوریم تا حدی پهنای باند را فدای BER کنیم تا سطح خطا از آستانه ی مشخص شده همواره بالاتر باشد یا اینکه در حالت کلی از Erasure و خطا فاصله بگیریم. شکل ۳-۱ نمونه ای از این Trade-off بین پهنای باند و احتمال خطا را با استفاده از مدولاسیون های دیجیتال متفاوت نشان می دهد. در این مدولاسیون ها هرچه بیت های بیشتری را در هر Symbol جای دهیم، در یک SNR یکسان پهنای باند اشغالی کاهش و احتمال خطا بالا می رود. در این سوال تنها معیار BER را برای ارزیابی کد استفاده میکنیم.



شکل ۳-۱

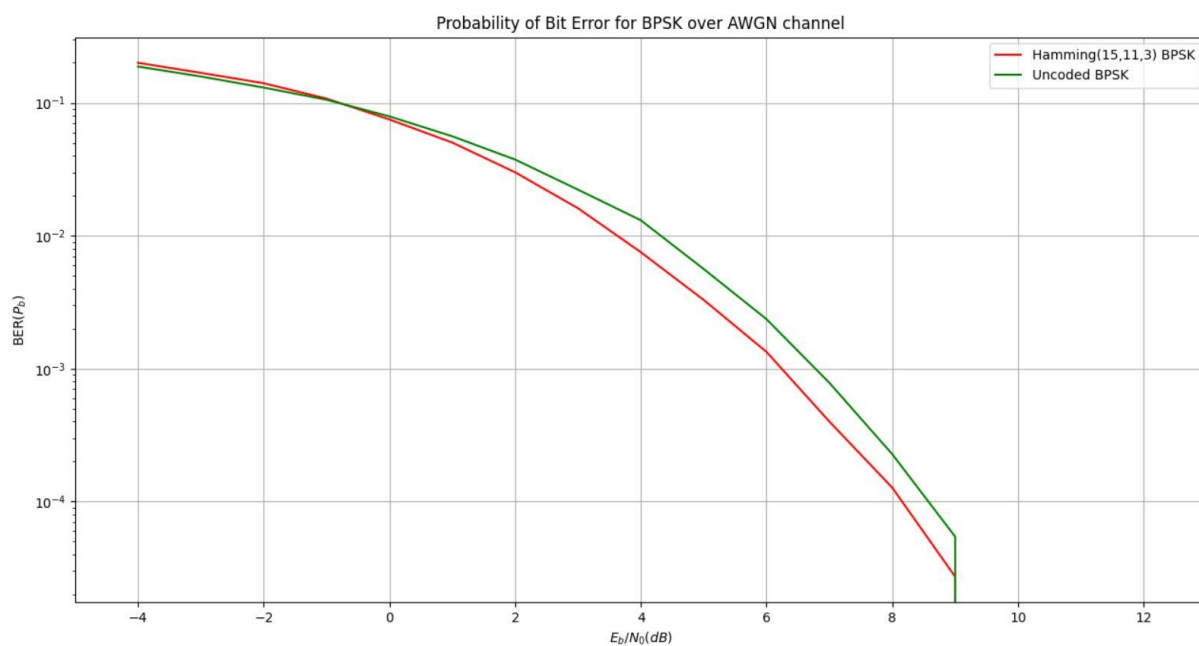
از آنجا که در این سوال از مدولاسیون BPSK استفاده می کنیم، رابطه ی تئوری احتمال خطا بر حسب SNR برای این نوع از مدولاسیون ها در رابطه ۱-۱ آورده شده است که $\frac{Eb}{N0}$ در آن همان SNR یا سطح سیگنال به نویز است.

$$\Pr(e) = 0.5 * \operatorname{erfc}\left(\sqrt{\frac{Eb}{N0}}\right) \quad (\text{رابطه ۱-۱})$$

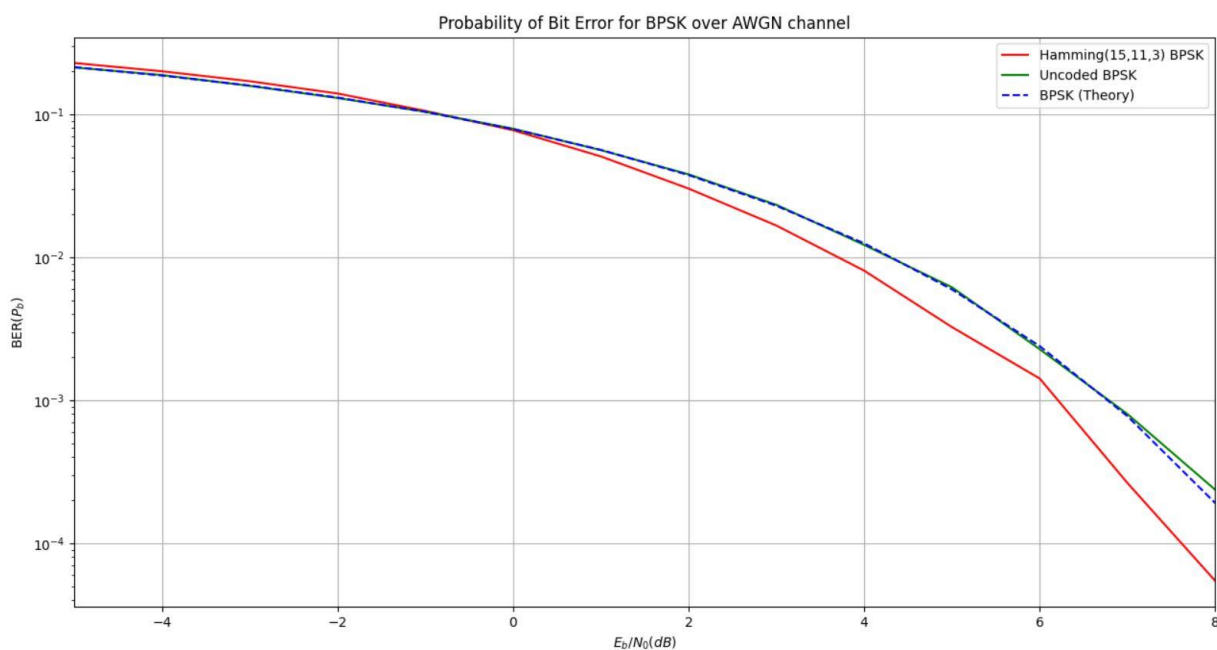
بخش ۴ و ۵ + امتیازی :

در این بخش با برنامه نویسی یک انکدر و دیکدر کد (15,11) Hamming، مدولاتور و دمدولاتور BPSK، نویز گوسی سفید جمع شونده و افزودن آن به کانال و آشکارسازی و محاسبه ی احتمال خطا نتایج شبیه سازی را با نتایج تئوری مقایسه کرده ایم. برای شبیه سازی، ابتدا یک آرایه بیت رندوم با طول ۱۱۰۰۰۰ بیت را با کد همینگ انکد می کنیم و پس از اعمال مدولاسیون بر روی آن، وارد یک کانال AWGN می کنیم. نویز موجود در این کانال با توجه به مقدار SNR در هر مرحله تغییر می کند و با افزایش سطح سیگنال به نویز، تاثیر نویز کانال را بر روی سیگنال مدوله شده کاهش می دهیم. در گیرنده سیگنال را از یک دمدولاتور BPSK عبور می دهیم و پس از دیکد کردن کد همینگ آن، بیت های حاصل را با بیت های اولیه در فرستنده مقایسه می کنیم و با تقسیم تعداد بیت های خطا بر تعداد کل بیت ها احتمال خطا برای هر SNR بدست می آید و در پایان نمودار آن را رسم می کنیم. برای شبیه سازی سیگنال کد نشده نیز مانند روند بالا عمل می کنیم، با این تفاوت که در این مرحله نیازی به انکد کردن و دیکد کردن نمی باشد. برای حالت تئوری نیز با استفاده از رابطه ۱-۱ نمودار را رسم می کنیم و در پایان نتایج را در یک نمودار با یکدیگر مقایسه می کنیم. با توجه به شکل شبیه سازی شده ی ۱-۴-۱، مشاهده می شود که با استفاده از کد همینگ مورد نظر، در یک سطح سیگنال به نویز یکسان، احتمال خطا تا حد خوبی کاهش می یابد. در شکل ۱-۴-۲ رابطه ی تئوری نیز در نمودار اعمال شده است که با مدل شبیه سازی تا حد بسیار زیادی یکسان شده است. شکل ۱-۴-۴ نیز تاثیر کانال رایلی و فیدینگ موجود در آن را که بطور عمده ناشی از چندراهه بودن سیگنال در شهرهای شلوغ و پر از ساختمان است، نشان داده شده است که علاوه بر نویز، سیگنال را در یک ضریب فیدینگ ضرب میکند و باعث میشود عمل دمدولاسیون و آشکارسازی از دقت کمتری برخوردار شوند و خطاهای بیشتری رخ دهد. با مقایسه ی کانال رایلی و AWGN در نمودار های این بخش متوجه BER بیشتر کانال رایلی می شویم.

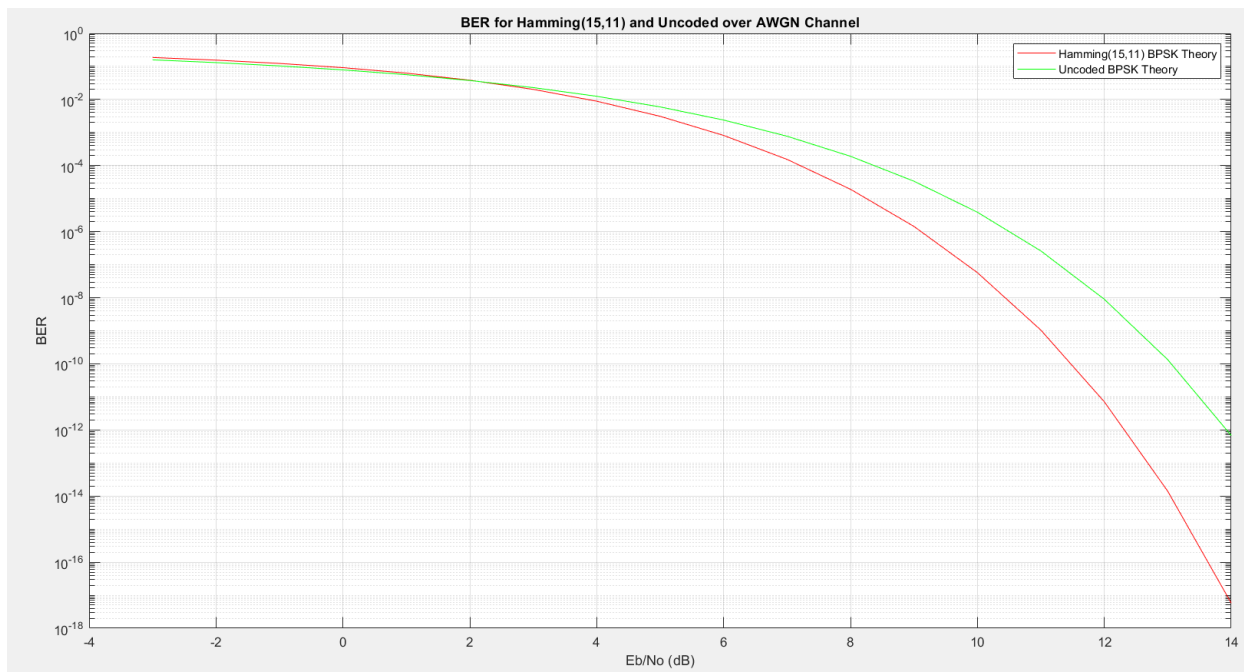
در شکل ۱-۴-۳ نیز به مقایسه‌ی رابطه‌ی BER و SNR در حالت تئوری پرداخته ایم که تا حد زیادی به شبیه سازی نزدیک می باشد.



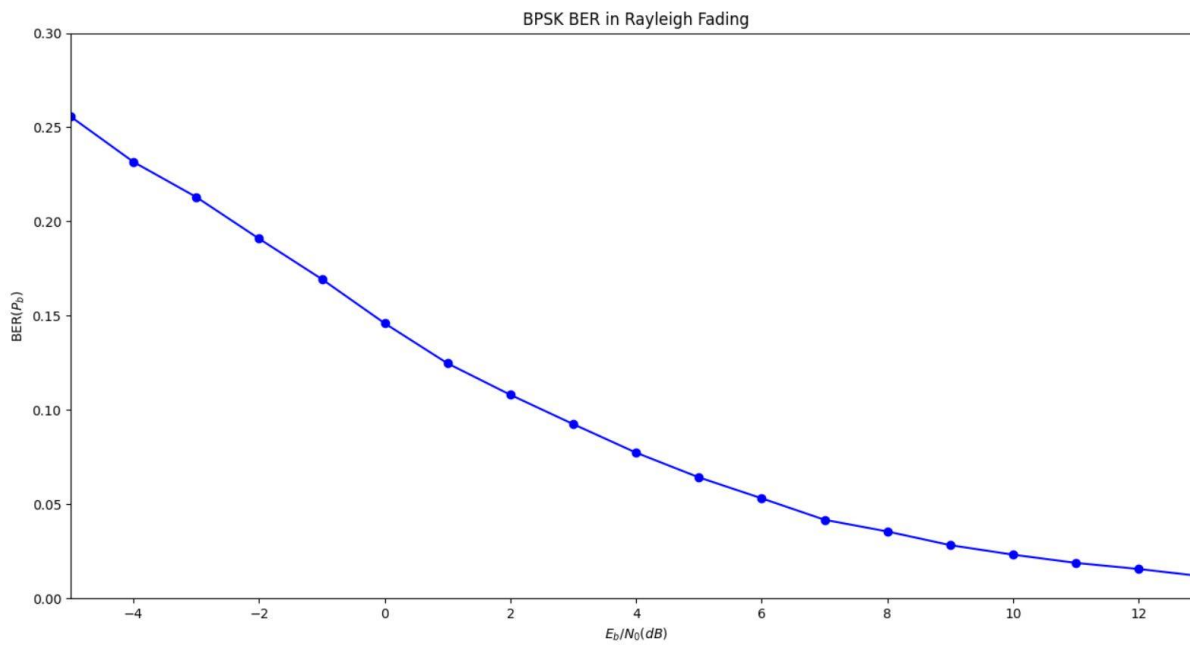
شکل ۱-۴-۱- شبیه سازی همینگ و کد نشده BPSK در کانال AWGN



شکل ۱-۴-۲- مقایسه‌ی شبیه سازی ها با تئوری BPSK در کانال AWGN



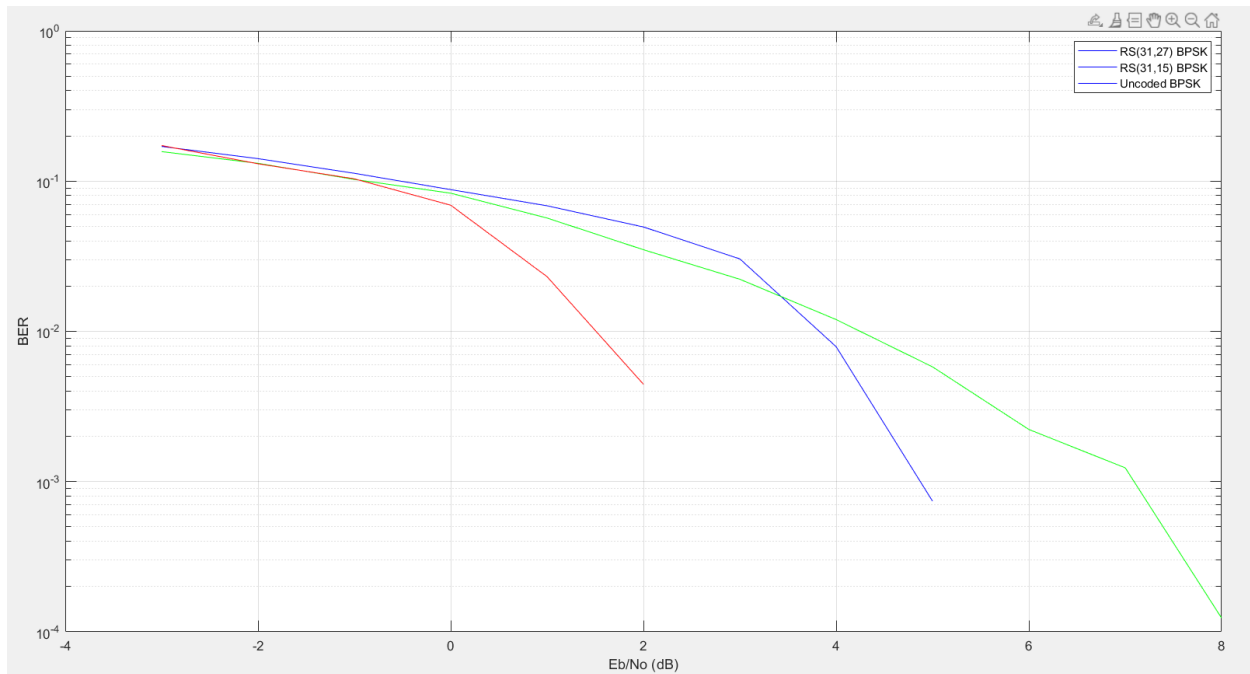
شکل ۱-۴-۳- مقایسه ی حالت تنوری Hamming(15,11) و حالت کد نشده در کانال AWGN



شکل ۱-۴-۴- تاثیر کانال راییلی

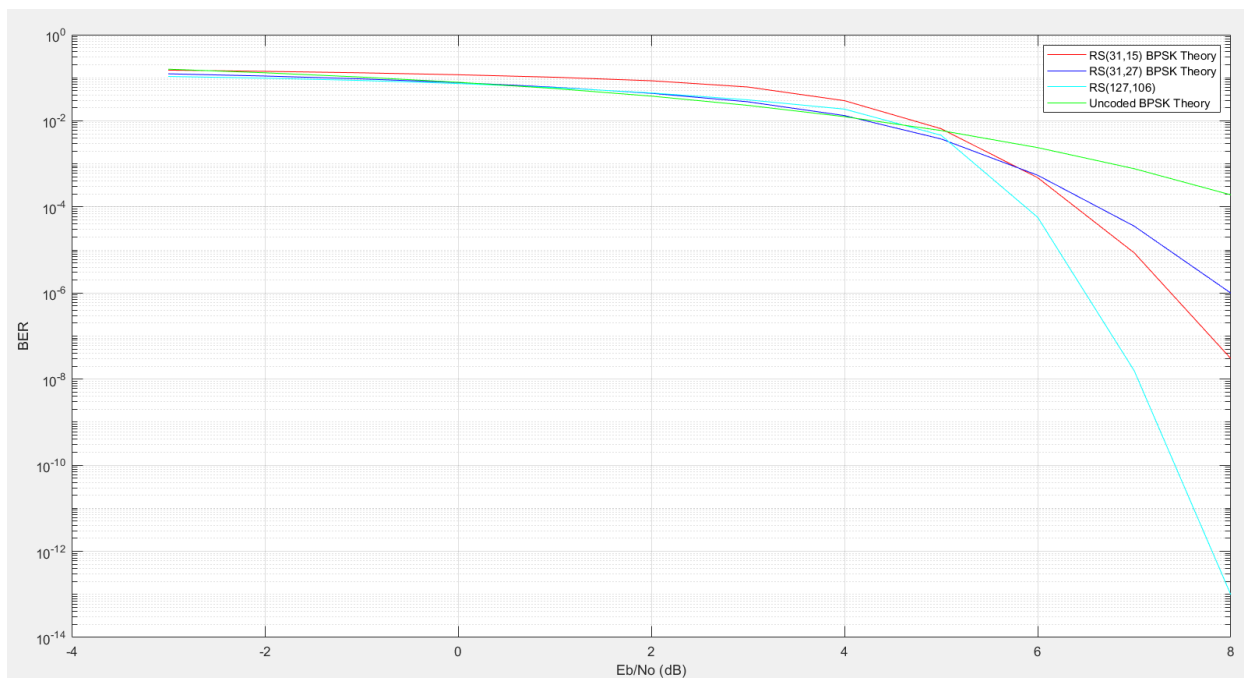
سوال ۲ (+ امتیازی):

در این سوال به انکدینگ و دیکدینگ کدهای رید-سولومون می پردازیم که در صنعت کاربردهای بسیار زیادی نیز دارند. شکل ۱-۲ مقایسه شبیه سازی شده ی BER بر حسب SNR برای کدهای RS(31,15) و RS(31,27) و حالت کد نشده را داریم و مشخص است در $SNR = -1\text{dB}$ کد RS(31,15) از کد RS(31,27) و حالت کد نشده از لحاظ کارایی تصحیح خطاهای موجود در کانال پیشی میگیرد. کد RS(31,15) نیز از $SNR = 3.5\text{dB}$ به بعد از حالت کد نشده پیش میگیرد و احتمال خطای کمتری پیدا میکند. نمودار قرمز RS(31,15)، نمودار آبی بعد از حالت کد نشده پیش میگیرد و احتمال خطای کمتری پیدا میکند. نمودار سبز حالت کد نشده BPSK در کانال AWGN طی شبیه سازی را نشان می دهد.

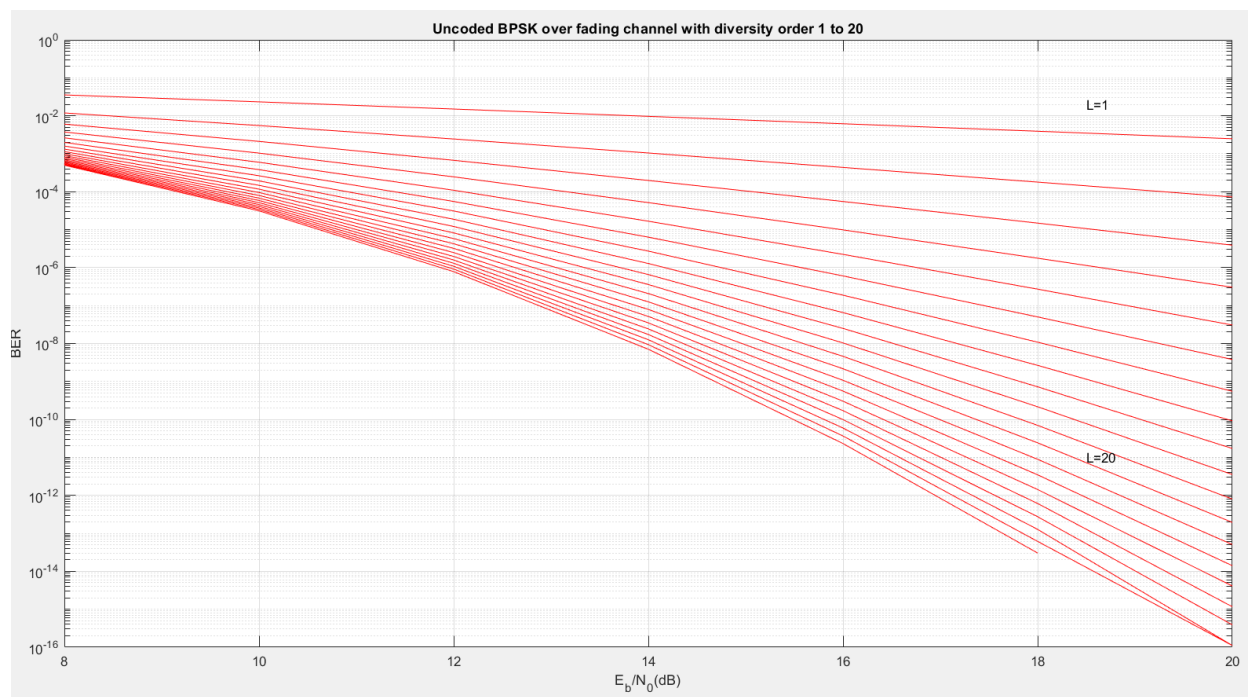


شکل ۱-۲- شبیه سازی و RS(31,15) و RS(31,27) و حالت کد نشده در کانال AWGN

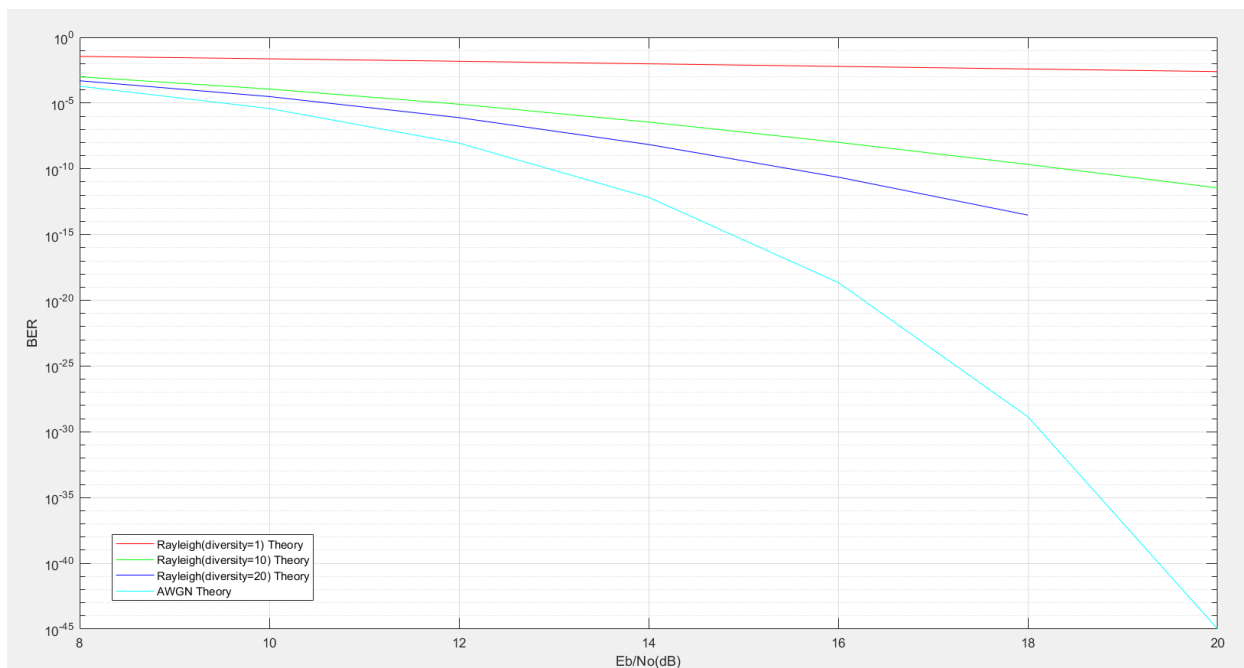
در شکل ۲-۲ به مقایسه ی حالت تئوری سیگنال های RS(31,15) و RS(31,27) و RS(127,63) و حالت کد نشده با مدولاسیون BPSK در کانال AWGN پرداخته ایم. همانطور که انتظار می رود در یک SNR یکسان، به ترتیب RS(127,106) و RS(31,15) و RS(31,27) و حالت کد نشده احتمال خطای ککتری را متحمل می شوند. شایان ذکر است که این برتری از $SNR = 5\text{dB}$ بطور جدی تر اختلاف خود را نشان می دهد.



شکل ۲-۲- مقایسه‌ی تئوری $RS(31,15)$ و $RS(31,27)$ و $RS(127,63)$ و حالت کد نشده با مدولاسیون BPSK در کانال AWGN



شکل ۳-۲- مقایسه‌ی تئوری احتمال خطا در کانال ریلی با دایورسیتی های متفاوت (۱ تا ۲۰)



شکل ۲-۴- مقایسه ی تئوری خطای سیگنال های کد نشده در کانال های راییلی و کانال با نویز سفید گوسی جمع شونده

در شکل ۲-۴ خطای تئوری سیگنال های کد نشده در کانال های راییلی و کانال با نویز سفید گوسی جمع شونده مقایسه شده است که همانطور مشخص است کانال AWGN از تمامی حالات کانال Rayleigh کمتر سیگنال را دچار خطا می کند و دلیل آن وجود ضریب Fading علاوه بر نویز سفید گوسی جمع شونده است که قبل از جمع شدن سیگنال اصلی با نویز آن را در ضریب فیدینگ ضرب می کند و باعث اعوجاج می شود و آشکارسازی را دچار مشکل می کند.

سوال ۳ (+ امتیازی) :

بخش اول:

به ازای ارسال هر بیت، سه بیت در خروجی تحویل میگیریم. در این بخش بطور خودکار کد به هر آرایه بیت ورودی دو صفر در ابتدا و دو صفر در انتها برای ایجاد حالت اولیه و حالت نهایی '00' اضافه می کند.

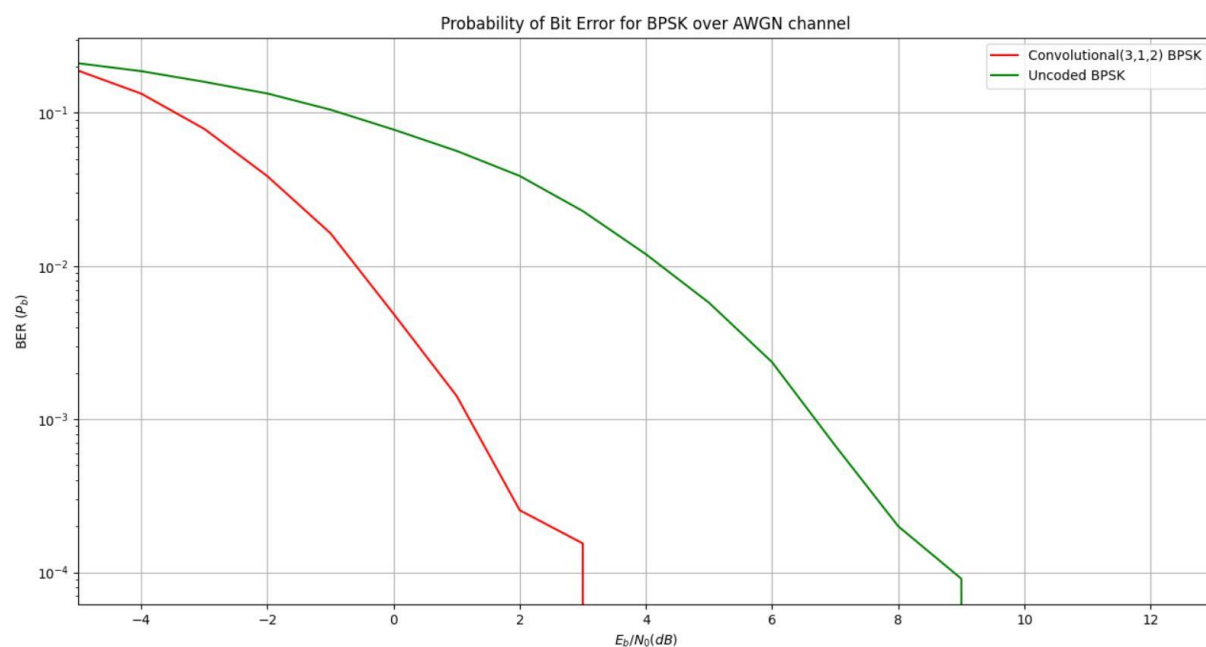
```
ENTER AN ARRAY OF BITS TO ENCODE: [1,0,0,1,0,1,1,1,0,1,0,1,0,1,1,0,0,0,1,1,0,1]
[1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0,
1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1,
0, 0, 1, 0, 1, 0, 1, 1]
```

شکل ۳-۱- یک نمونه ورودی و خروجی از انکدر کانولوشنال موردنظر

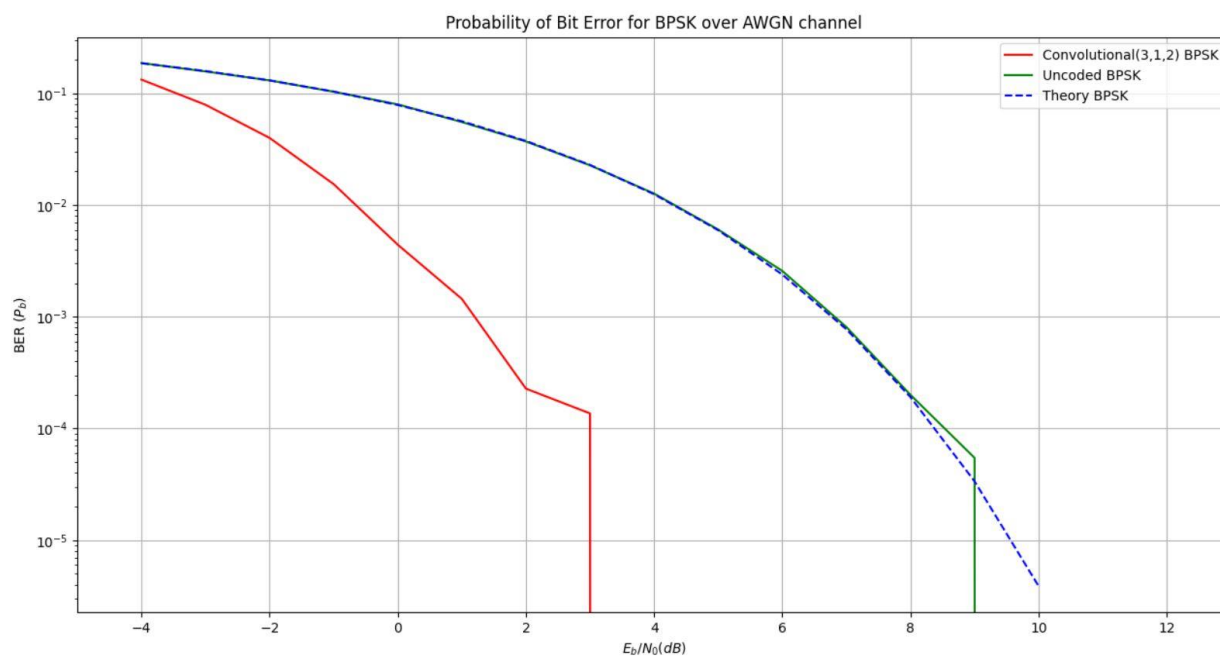
بخش دوم + امتیازی :

برای این سوال انکدر و دیکدر کانولوشنال مربوطه با استفاده از زبان پایتون تماماً نوشته شده است و در دیکدر از الگوریتم ویتربی استفاده شده است. پیاده سازی الگوریتم ویتربی در این برنامه، در بهینه ترین حالت ممکن نمی باشد و به طرق دیگر می توان به سرعت بیشتری در دیکد کردن نیز دست یافت. با این حال الگوریتم به درستی کار می کند و علاوه بر رسم نمودارهای BER بر حسب SNR برای سیگنال های کد شده و کد نشده برای حالت تئوری و شبیه سازی، تمام متریک های نود های دیاگرام ترلیس را نیز ذخیره می کند و بعنوان نمونه متریک آخرین نود دیاگرام ترلیس برای هر SNR که بیانگر مجموع تعداد اصلاحات بیتی و آخرین نود survivor path است را طی یک نمودار رسم می کند. شایان ذکر است که برنامه ی نوشته شده بعنوان ورودی انکدر آرایه بیتی بصورت رندوم به طول ۱۰۰۰۰۰ تولید می کند و شبیه سازی ها را با استفاده از آن انجام می دهد.

در شکل ۳-۲ به مقایسه ی شبیه سازی کد Convolutional(3,1,2) موردنظر سوال و حالت کد نشده با مدولاسیون BPSK در کانال AWGN پرداخته ایم. با توجه به نمودار حاصل شده، مشخص است که احتمال خطا در کد کانولوشنال موردنظر به طرز چشم گیری در تمام SNR ها کاهش می یابد. بعنوان مثال در $SNR = 2dB$ احتمال بروز خطا در کد کانولوشنال به یک صدم حالت کد نشده می رسد.

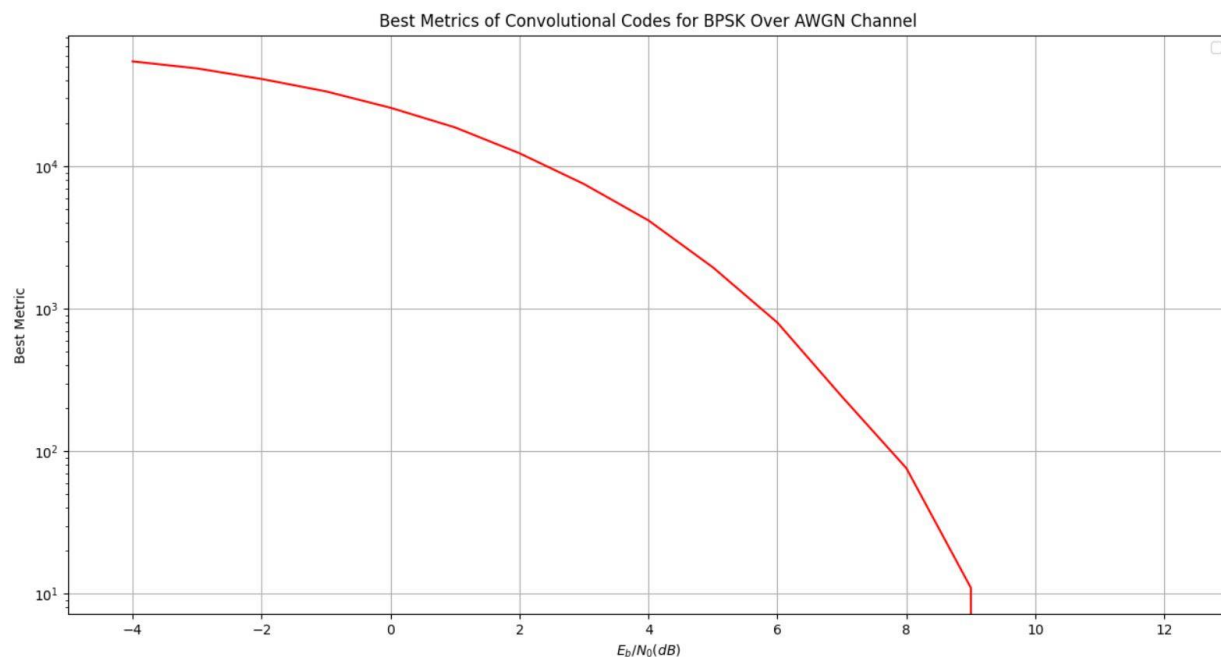


شکل ۳-۲- مقایسه‌ی شبیه‌سازی کد Convolutional(3,1,2) موردنظر سوال و حالت کد نشده با مدولاسیون BPSK در کانال AWGN

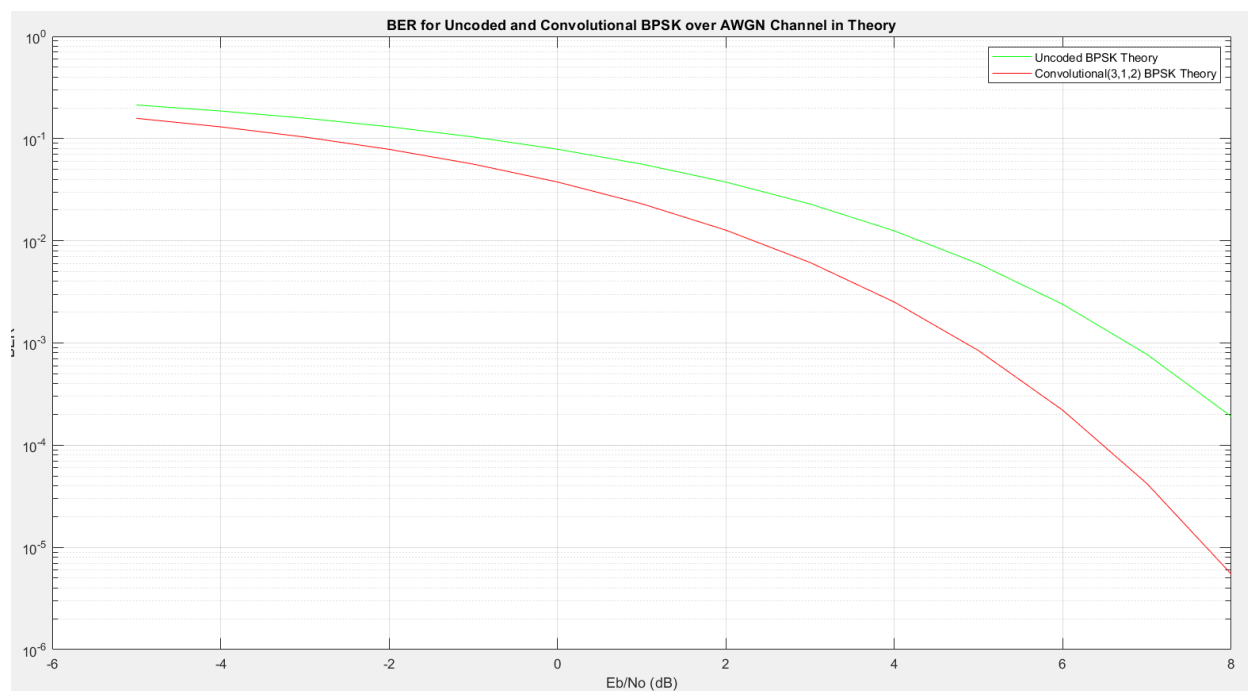


شکل ۳-۳- مقایسه‌ی شبیه‌سازی کد Convolutional(3,1,2) موردنظر سوال و حالت کد نشده و حالت تئوری با مدولاسیون BPSK در کانال AWGN

در شکل ۳-۳ نمودار BER بر حسب SNR های مختلف را نیز به نمودار قبل اضافه نموده ایم. همانطور که ملاحظه می شود با نتیجه‌ی شبیه سازی شباهت بسیار زیادی دارد.

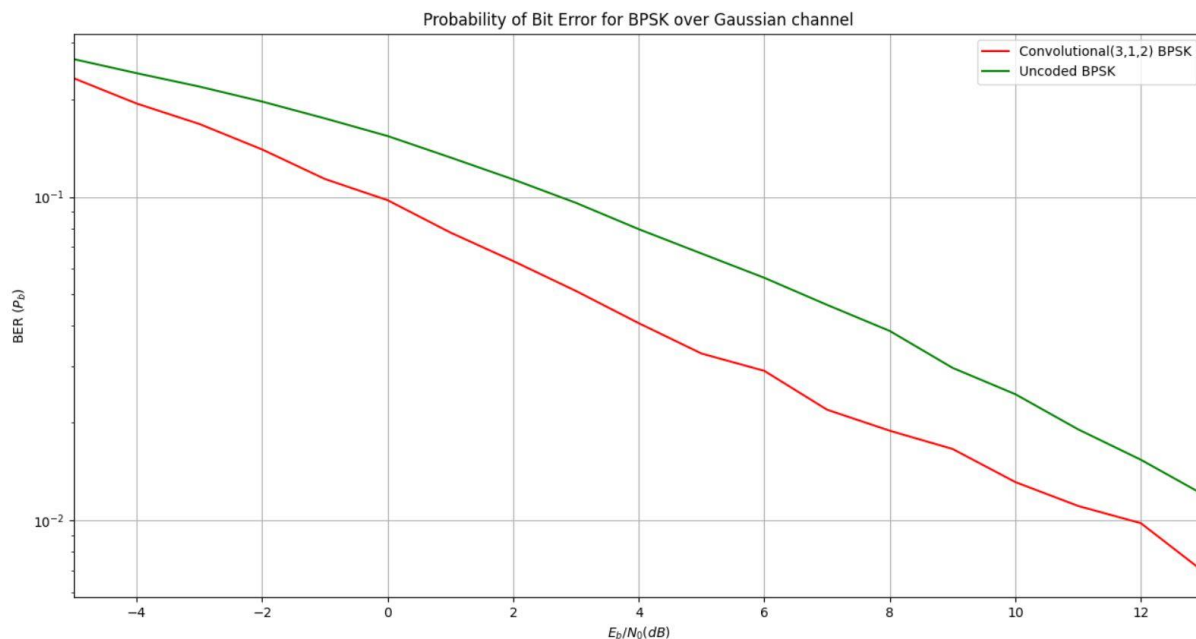


شکل ۳-۴- بهترین متریک کد کانولوشنال موردنظر در مرحله‌ی دیکدینگ به ازای SNR های مختلف



شکل ۳-۵- مقایسه‌ی تئوری کد Convolutional(3,1,2) و حالت کد نشده با مدولاسیون BPSK در کانال AWGN

همانطور که در شکل ۳-۵ پیداست، در حالت تئوری نیز کد کانولوشنال وضعیت بهتری از نظر احتمال رویداد خطا دارد و با نتیجه‌ی حاصل از شبیه‌سازی شباهت بسیار زیادی دارد.



شکل ۳-۶- شبیه‌سازی کد Convolutional(3,1,2) و حالت کد نشده با مدولاسیون BPSK در کانال Gaussian

همانطور که در شکل ۳-۶ مشاهده می‌شود، احتمال خطا برای حالت کد شده نسبت به حالت کد نشده در کانال Gaussian نیز کمتر است اما هر دو BER نسبت به کانال AWGN به دلیل وجود فیدینگ، به شدت افزایش پیدا کرده‌اند.

در پایان شایان ذکر است که در سوال اول و سوم این پروژه، در قسمت‌های شبیه‌سازی بنده از هیچ‌گونه تابع خاص آماده‌ای استفاده نکرده‌ام و تمام قسمت‌های انکدینگ و دیکدینگ و کانال را با استفاده از پایتون نوشته‌ام. اما برای سوال دوم در مرحله‌ی دیکدینگ رید-سولومون در قسمت برنامه‌نویسی الگوریتم برلکم-مسی و تشخیص مقدار خطا به مشکل برخورددم و ناچاراً از تولباکس communication متلب استفاده نمودم.