



به نام خدا



دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده مهندسی برق و کامپیوتر

مخابرات 2

استاد: دکتر ربیعی

پروژه شماره 2

محمدحیدری

810197494

اردیبهشت ماه 1400

تولید پالس Raised_Cosine

در این قسمت قصد داریم برای $B=0.5$ پالس Raised_Cosine را در متلب شبیه سازی کنیم.

برای اینکار 3 سناریو مختلف را پیش میگیریم : حالت اول بدون داشتن خطا در بازه زمانی (ایده آل) و حالت دوم با داشتن خطایی به اندازه $\epsilon = 0.1T$ و در حالت سوم با خطایی به اندازه $\epsilon = 0.2T$ پیاده سازی پالس را انجام خواهیم داد.

$$t1 = (-6 * T : T / fs : 6 * T)$$

$$\epsilon = 0$$

$$t2 = (-6.1 * T : T / fs : 5.9 * T)$$

$$\epsilon = 0.1T$$

$$t3 = (-6.2 * T : T / fs : 5.8 * T)$$

$$\epsilon = 0.2T$$

برای پیاده سازی کد دقت شود که از ضابطه مطرح شده برای Raised_Cosine در جزوه استفاده کرده ام.

$$P_R(t) = \text{sinc}\left(\frac{t}{T}\right) \frac{\cos\left(\frac{\pi B t}{T}\right)}{1 - \left(\frac{\pi B t}{T}\right)^2}$$

تنها نکته ای که در پیاده سازی این تابع در متلب میبایست رعایت کرد این است که به ازای مقادیری که بینهایت میشود بجای مقادیر برداری پالس مقدار $\frac{\pi}{4} \text{sinc}\left(\frac{1}{2B}\right)$ قرار گیرد. تمامی نکات مطرح شده به صورت کامل در کد پیاده سازی شده اند. منحنی های خواسته شده در ادامه ضمیمه شده اند:

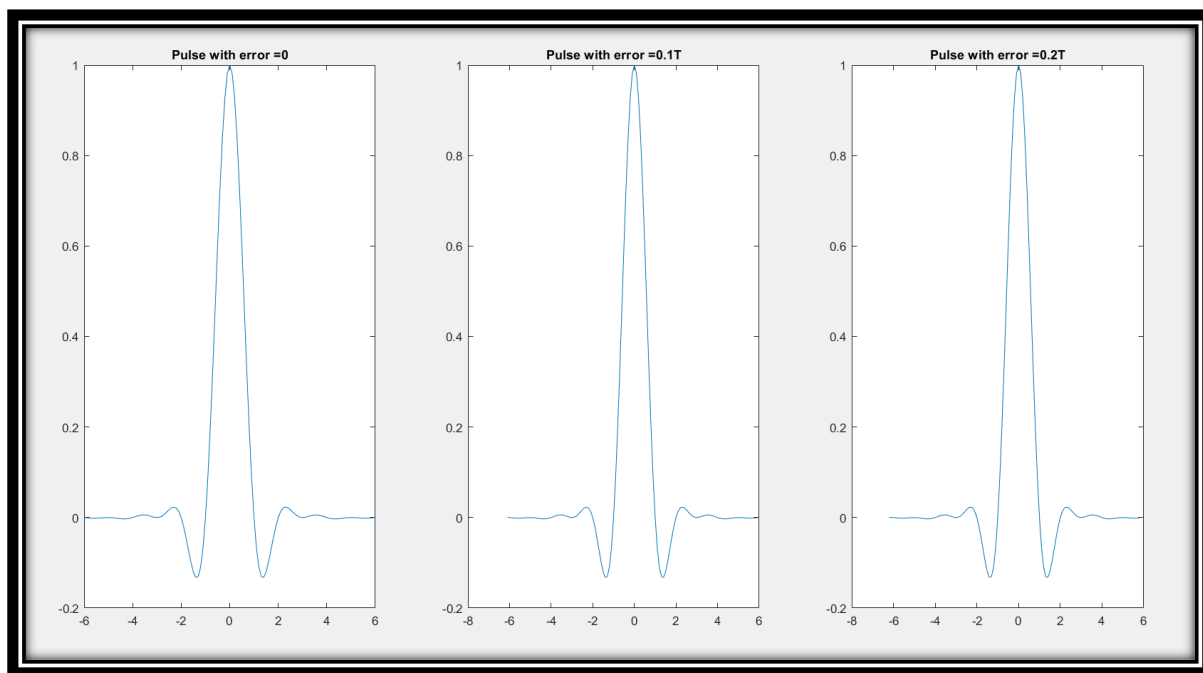


Figure 1

Figure1 نمودارپالس های پیاده سازی شده را بازای $B=0.5$ و برای هر 3 حالت خطا نشان میدهد.

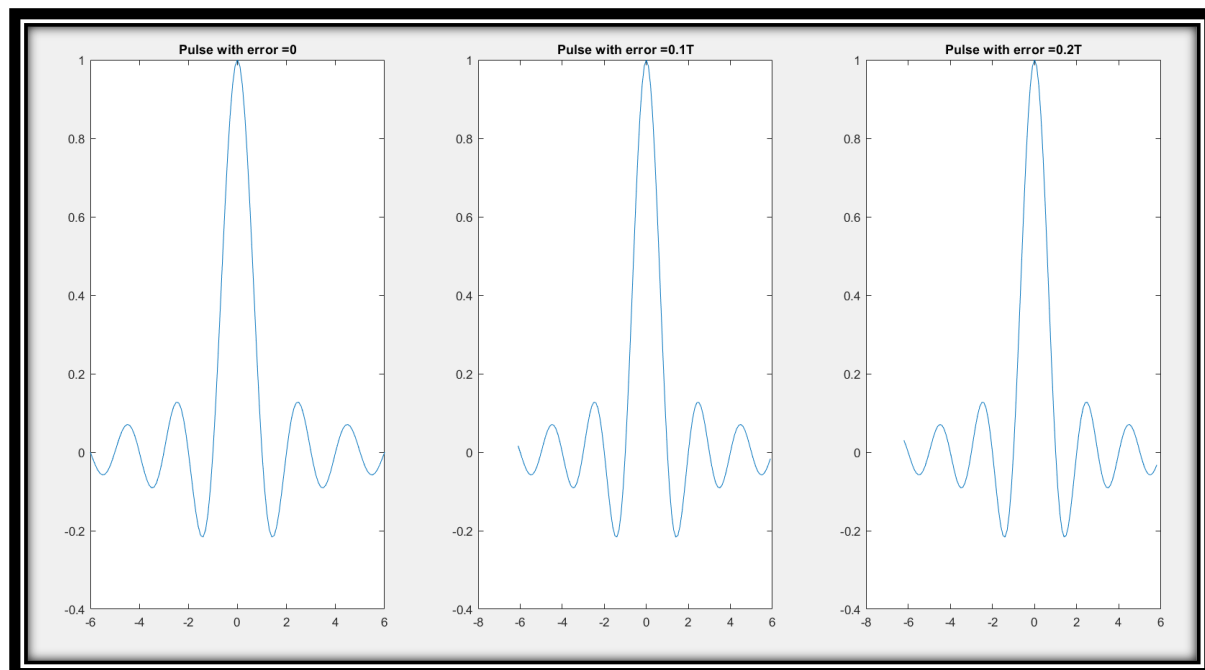


Figure 2

Figure2 نمودارپالس های پیاده سازی شده را بازای $B=0$ و برای هر 3 حالت خطا نشان میدهد.

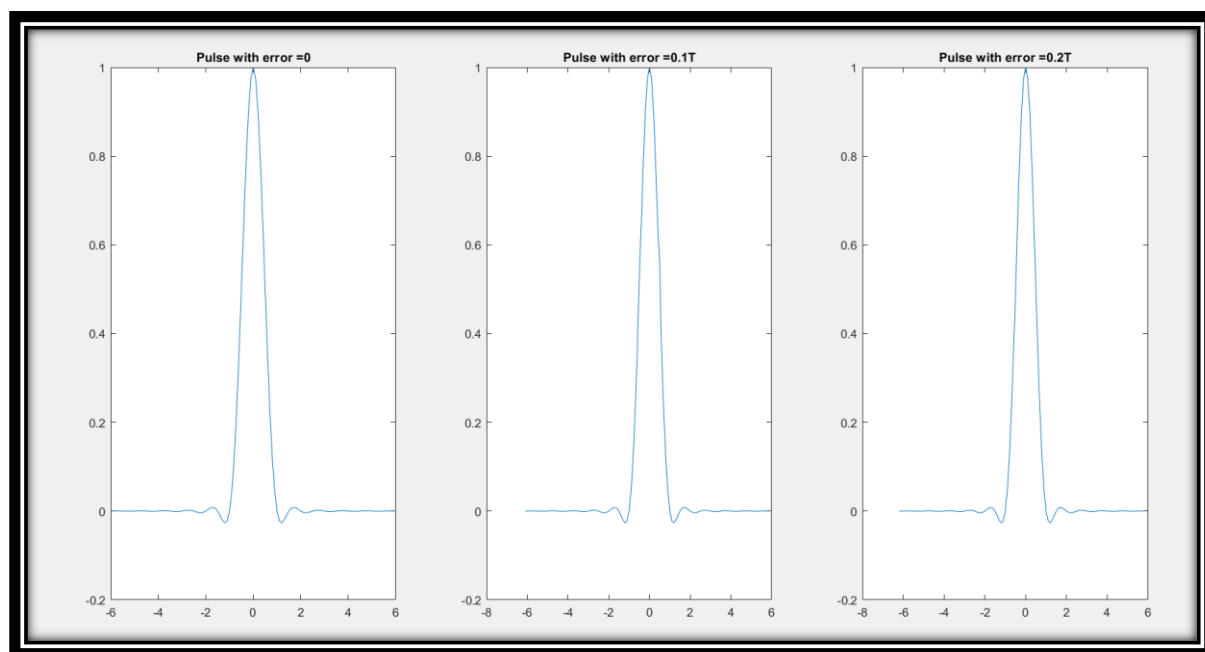


Figure 3

Figure3 نمودارپالس های پیاده سازی شده را بازای $B=1$ و برای هر 3 حالت خطا نشان میدهد.

تحلیل منحنی های بالا:

همانطور که منحنی های بالا به وضوح نشان میدهند هرچقدر مقدار B کمتر میشود و به مقدار 0 نزدیک تر میشود شکل موجی تر میشود و درواقع isi بیشتر خودش را نشان میدهد و ملموس تر است. درواقع نتیجه ای که از این بخش میگیریم این است که حساسیت isi نسبت به بتاهای کوچک تر بیشتر است و برای مقدار صفر بیشترین مقدار اعوجاج را خواهیم داشت. (از این نتیجه در سوال آخر استفاده شده است)

تولید سیگنال ارسالی

در این قسمت میبایست دنباله ارسالی را تولید کنیم در ابتدا با استفاده از دستور `randn` یک مجموعه از 0 و 1 ها تولید میکنیم و سپس متناظر با هر 0 یک -1 و متناظر با هر 1 یک 1 در دنباله قرار میدهیم و دنباله ارسالی را خواهیم ساخت که این آرایه `modulated_symbol` نامیده میشود.

برای پیاده سازی سیگنال ارسالی همانطور که در درس اشاره شد میبایست ابتدا از دنباله ورودی (گسسته) نمونه برداری شود و به حوزه پیوسته برده شود و سپس با پالس `raised_cosine` که در قسمت قبلی بدست آمد کانالو شود تا به سیگنال `transmitted_signal` برسیم.

در مطلب چون ماتریس های خود را به صورت گسسته انجام میدهیم درواقع منتظر با حالت تیوری نمونه برداری را به صورت `up_sample` کردن دنباله ورودی انجام میدهیم و بین هر 2 سمبل متوالی $L-1$ صفر قرار میدهیم که درواقع همان اعمال کردن تابع ضربه برای نمونه برداری در حالت پیوسته میباشد. و در انتها نیز با کانالو کردن با `raised_cosine` به سیگنال `transmitted_signal` دست می یابیم.

مدلسازی کانال A

در این قسمت قصد داریم کانال انتقال را مدلسازی کنیم و همانطور که میدانیم بر اثر عبور از کانال یک نویز سفیدگوسی به سیگنال ارسالی اضافه میشود. در این قسمت قصد داریم با استفاده از مقادیر SNR و رابطه آن بتوان نویز به `Implement` کردن نویز کانال برای SNR های مختلف پردازیم.

برای این کار با استفاده از رابطه SNR با توان نویز که در صورت سوال قید شده مقدار η را برای هر یک از مقادیر SNR محاسبه میکنیم و متناظر با آن با دستور `randn` به ساختن بردارهای نویز با توان متناظر میکنیم. در انتها نیز تمام 11 بردار نویز تولید شده را تک به تک به سیگنال ارسالی اضافه میکنیم و

درارایه Received_signal دخیره میکنیم که درمراحل بعدی باید عمل آشکارسازی و محاسبه خطا برای هریک از 11 حالت SNR محاسبه شوند.

آشکارسازی سمبل ها

دراین قسمت قصدداریم سیگنال ارسال شده را آشکارسازی کنیم. برای این کار دربازه مشخص شده درصورت سوال ازسیگنال received_signal نمونه برداری میکنیم و به ازای تمام SNR های مختلف سیگنال samples را بدست می آوریم.

درمرحله آخرنیز با پیمایش آرایه samples توسط یه حلقه for ومقایسه مقدار Threshold که دراینجا باتوجه به تقارن 0 درنظرگرفته شده است بردار سیگنال آشکارسازی detected_symbols را تشکیل میدهم.

محاسبه احتمال خطا

دراین قسمت کافی است که بازای هریک از SNR های مختلف تعداد خطاهای سیگنال آشکارسازی شده را نسبت به modulated_signal بدست بیاوریم و با تقسیم آن برتعداد کل سمبل ها یعنی N بردار خطا را بدست بیاوریم.

درادامه منحنی های لگاریتمی خطا برای $B=0,0.5,1$ بترتیب رسم شده است :

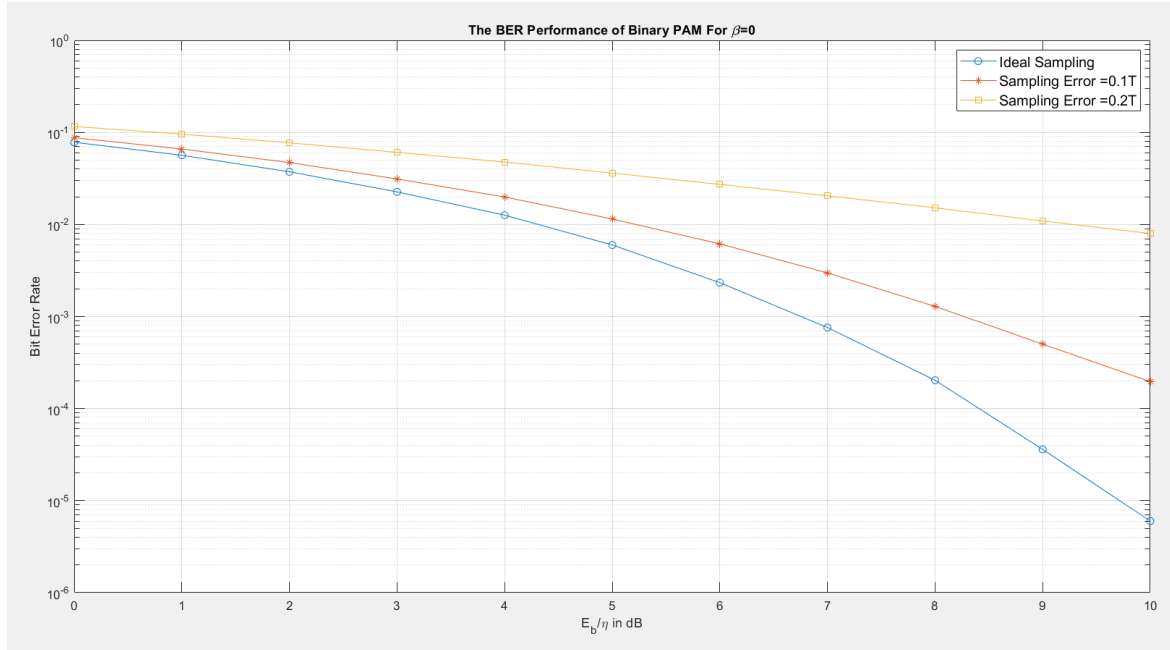


Figure 4

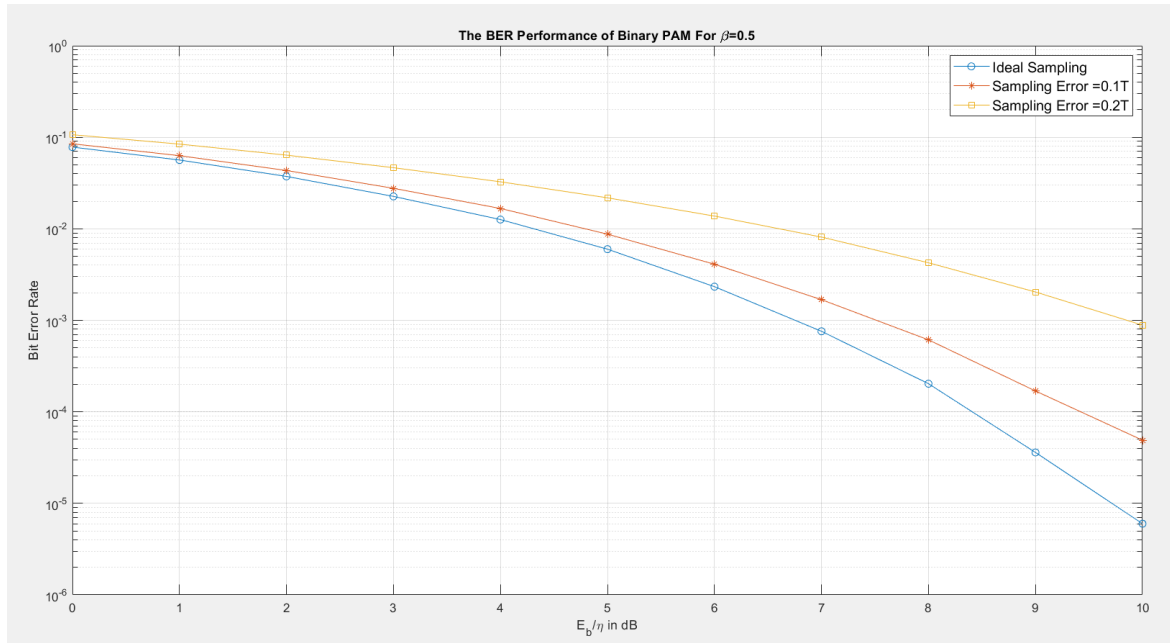


Figure 5

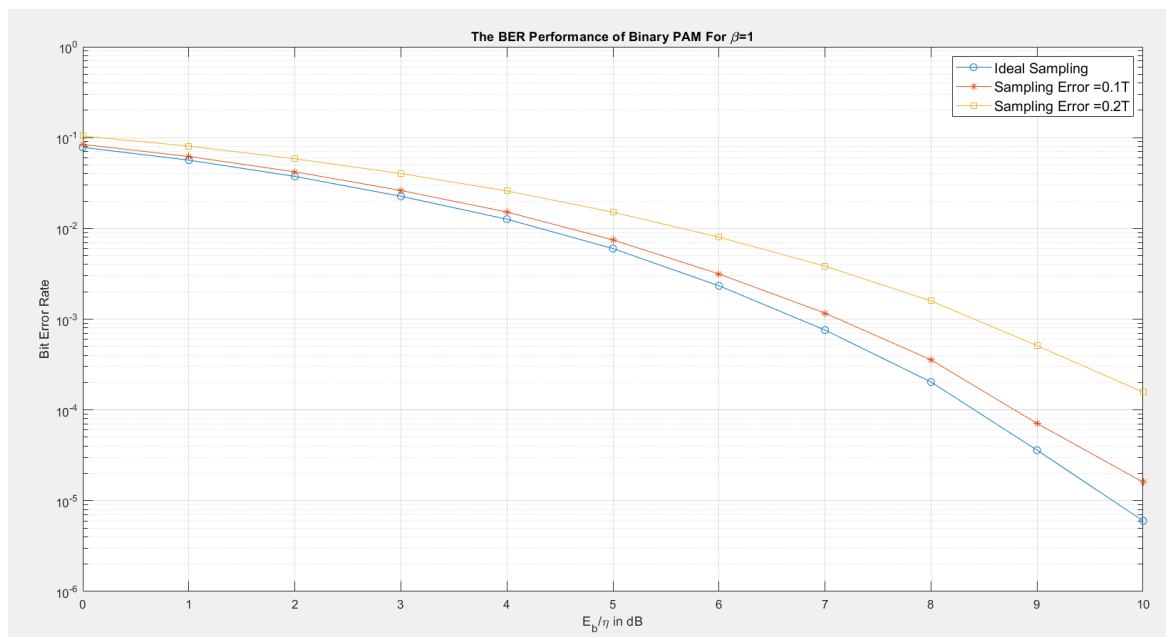


Figure 6

همانطور که میبینیم منحنی آبی که نمایانگر حالت ایده آل میباشد در هر 3 شکل در پایین ترین حالت خود قرار گرفته است و علت آن نیز واضح است زیرا در منحنی های زرد و نارنجی علاوه بر خطای نمونه برداری خطای ISI نیز وارد شده که طبعاً باعث میشود منحنی های زرد و نارنجی که بترتیب 0.2T و 0.1T خطا دارند بترتیب رتبه ی بالاترین خطا و 2دومین خطا را خواهند داشت که کاملاً طبق انتظار میباشد و منحنی ایده آل کمترین خطای را خواهد داشت.

حساسیت B:

همانطور که منحنی ها نشان میدهند در حالت $B=0$ بیشترین حساسیت نسبت به ISI را خواهیم داشت زیرا در این حالت منحنی های زرد و نارنجی با منحنی آبی که نمایانگر حالت ایده آل است بسیار فاصله دارند که نشان دهنده ی این میباشد که در حالت $B=0$ خطای ناشی از ISI بسیار محسوس تر است نسبت به حالت $B=0.5$ و $B=1$ و این امر باعث میشود منحنی های زرد و نارنجی که خطای ISI را نیز دارند نسبت به حالت ایده آل بسیار فاصله بگیرند.

اگر به منحنی های مربوط به $B=0.5$ و $B=1$ نگاه کنیم میبینیم که با افزایش B مقدار فاصله بین منحنی زردنارنجی و آبی نسبت به حالت قبل بسیار کاهش یافته که این نشانگر این مطلب است که بیشترین حساسیت خطای نمونه برداری مربوط به $B=0$ است و متناظرا کمترین حساسیت نیز مربوط به $B=1$ میباشد.

درمورد علت این امر کافی است به نتیجه نمودار های raised_cosine رجوع کنیم همانطور که میدانیم isi ناشی از پیک سیگنال های باند کناری raised_cosine میباشد. همانطور که میبینیم برای $B=1$ در اطراف پیک ماکزیمم پالس کناری تقریبا 0 و ثابتی خواهیم داشت و این موجب میشود که در برخورد 2 پالس مجاور که میخواهد منجر به ایجاد دامنه در خطای isi شود با ضرب 2 مقدار بسیار کوچک نزدیک به صفر مواجه شویم که در نتیجه اثر خطای isi بدلیل این دامنه ناچیز بسیار اندک است. و باعث میشود در مقایسه با حالت ایده ال خیلی تفاوت محسوسی بین منحنی زرد و آبی یا نارنجی و آبی نداشته باشیم و منحنی ها بسیار به یکدیگر نزدیک هستند.

اما با کاهش B و حرکت به سمت بتای 0 مقدار موجی های پالس های کناری پیک ماکزیمم بشدت افزایش می یابد و حاصل ضرب این پیک های کناری موجب ایجاد isi با دامنه محسوس میشود که در نتیجه آن در منحنی لگاریتمی نهایی Isi برتری و مقدار محسوس خود را نشان میدهد و منحنی های نارنجی و زرد با اختلاف از منحنی آبی (حالت ایده ال) فاصله میگیرند.

تحلیل منحنی ها روی لاین ها :

همانطور که میبینیم با حرکت روی هر لاین خطا کاهش می یابد علت این امر آن است که با افزایش SNR مقدار سیگنال به نسبت نویز زیاد شده و مطابق رابطه $\text{SNR} = \frac{1}{\eta}$ توان نویز کاهش می یابد و همانطور که منحنی نشان میدهد بازای SNR های بالا عملا توان نویز در حدی ناچیز است که عملا سیگنال modulated_symbols و detected_symbols بهم شبیه میشوند و در واقع آشکارساز باتوجه به

Threshold که دارد عملاً خطایی شمارش نمیکند که باعث میشود درمقدارهای SNR بالا خطا بسیار به صفر نزدیک گردد.

همانطور که درمنحنی های مربوطه به ازای B های یکسان میبینیم منحنی های مربوط به تاثیر ISI ناشی از نمونه برداریه غیرایده ال خطای بیشتری دارند و درواقع به ازای SNR های یکسان این منحنی ها بالاتر قرار میگیرند زیرا علاوه برخطای نمونه برداری خطای ISI را نیز متحمل شده اند.

توجه شود که بدلیل تکرار تمامی مراحل برای B=0 و B=1 تمامی کد بصورت function نوشته شده است.

توضیحات function ها :

تابع اول با نام pulse_generator.m به ازای B مختلف پالس raised_cosine را برای هر 3 حالت خطا تولید میکند و 3 بردار را به عنوان خروجی پاس میدهد.

و تابع Error_calculator.m با گرفتن بردار raised_cosine مربوط به هر حالت بردار خطا را به ازای SNR های مختلف را به عنوان خروجی پاس میدهد.

نکات پایانی :

تمامی کدها و screenshot علاوه بر متن گزارش درپوشه مربوطه ضمیمه شده اند.