



دانشگاه تهران  
پردیس دانشکده‌های فنی  
دانشکده برق و کامپیوتر

---

## مخابرات دیجیتال

---

گزارش تمرین کامپیوتری ۲

سید علیرضا جاوید

۸۱۰۱۹۸۳۷۵

استاد

دکتر ربیعی

۱ اردیبهشت ۱۴۰۱

# فهرست مطالب

فهرست مطالب	
۱	تولید پالس Raised-Cosine . . . . .
۲	تولید سیگنال ارسالی . . . . .
۳	مدل سازی کانال AWGN . . . . .
۴	آشکارسازی سمبلها . . . . .
۵	محاسبه احتمال خطا . . . . .
۶	محاسبه نتایج نهایی برای $\beta$ های دیگر . . . . .
۷	مقایسه و نتیجه گیری . . . . .

## ۱ تولید پالس Raised-Cosine

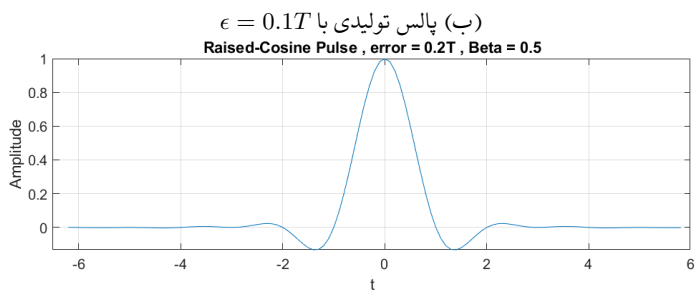
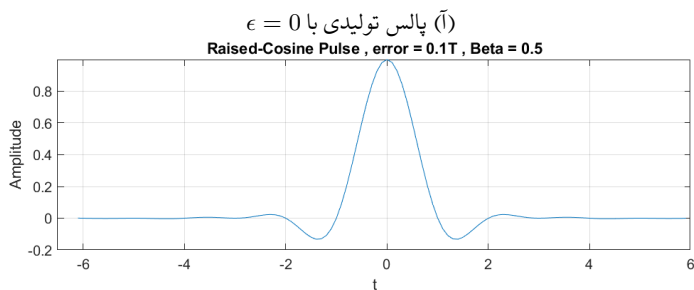
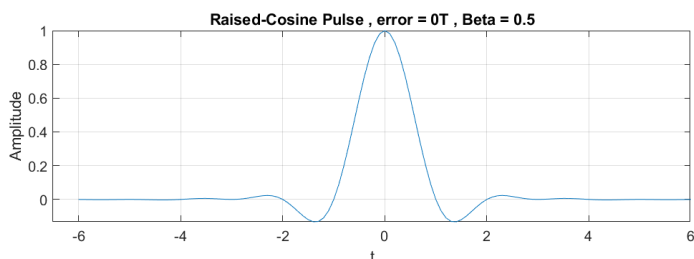
ابتدا برای پارامترهای به شرح زیر نتایج را بدست آورده و سپس مطابق خواسته انتهایی به ازای  $\beta$  های دیگر نیز تکرار کرده و مقایسه و نتیجه گیری می کنیم.

$$\beta = 0.5, T = 1, interval = 12T, F_s = 10, \epsilon = [0 \ 0.1T \ 0.2T]$$

با استفاده از تابع کمکی RC\_pulse.m مطابق پیوست و استفاده از تعریف زیر برای پالس

$$p_R(t) = \begin{cases} \text{sinc}(\frac{t}{T}) \times \frac{\cos(\frac{\pi\beta t}{T})}{1 - (\frac{2\beta t}{T})^2} & t \neq \pm \frac{T}{2\beta} \\ \frac{\pi}{4} \times \text{sinc}(\frac{1}{2\beta}) & t = \pm \frac{T}{2\beta} \end{cases} \quad (۱)$$

شبیه سازی را در متلب انجام داده و نتیجه زیر را بدست می آوریم.



(ج) پالس تولیدی با  $\epsilon = 0.2T$

شکل ۱: پالس های تولید شده برای  $\beta = 0.5$

## ۲ تولید سیگنال ارسالی

برای تولید modulated\_symbols ابتدا از الگوریتم زیر بهره می‌بریم.

```
1 for i=1:length(bits)
2   if ( bits(1,i) == 1 )
3     modulated_symbols(1, i) = 1 ;
4   else
5     modulated_symbols(1, i) = -1;
6   end
7 end
```

سپس با upsample کردن به کمک راهنمایی سوال در انتها با

```
1 for i = 1:3
2   transmitted_signal(i,:) = conv(upsampled_symbols, pulse(i,:)) ;
3 end
```

می‌توانیم سیگنال ارسالی (transmitted\_signal) را بدست آوریم.

## ۳ مدل سازی کانال AWGN

مطابق فرض سوال داریم:

$$SNR = \frac{1}{\eta} \rightarrow \eta = \frac{1}{SNR}$$

همچنین باید نویز تولید شده توسط تابع randn را در  $\sqrt{\frac{\eta}{2}}$  ضرب کنیم. از آنجایی که سائز بردار نویز و  $\eta$  متناسب نیست ابتدا باید به کمک توابع مناسب متلب این دو را هم سائز کنیم. برای این کار از کد زیر بهره می‌بریم

```
1 noise = randn(1,length(transmitted_signal(1,:)));
2 noise = diag(sqrt(eta./2))*repmat(noise,11,1);
3 received_signal1 = repmat(transmitted_signal(1,:),11,1)+noise;
4 received_signal2 = repmat(transmitted_signal(2,:),11,1)+noise;
5 received_signal3 = repmat(transmitted_signal(3,:),11,1)+noise;
```

به این شکل توانستیم سیگنال دریافتی (received\_signal) را برای هر ۳ حالت بدست آوریم.

## ۴ آشکارسازی سمبلها

ابتدا به کمک راهنمایی سوال زمان های مشخص برای نمونه برداری را پیدا کرده و در متغیر samples ذخیره می‌کنیم. همچنین در این قسمت از تابع کمکی symbol\_detection.m مطابق پیوست استفاده می‌کنیم. این تابع با مقایسه مقادیر سیگنال دریافتی با آستانه بهینه مقادیر ۱ و -۱ را به سمبل ها نسبت می‌دهد.

## ۵ محاسبه احتمال خطا

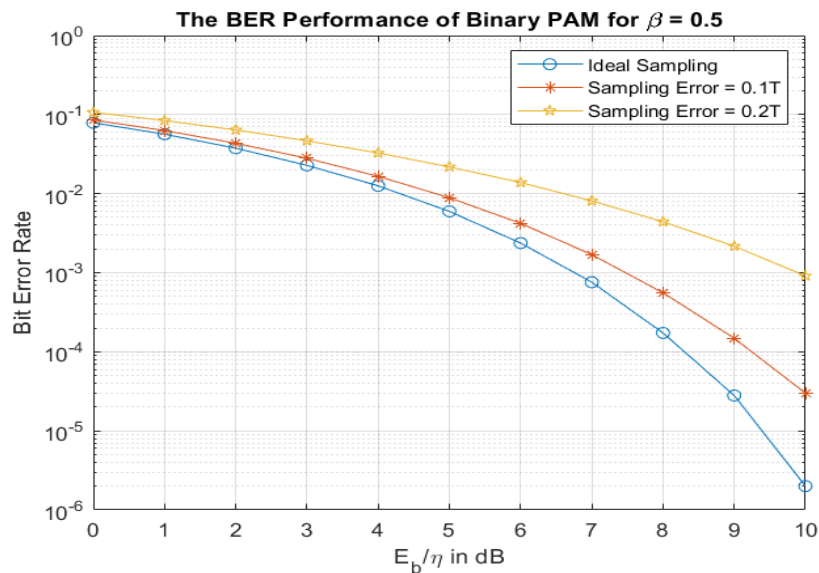
در نهایت باید احتمال خطا را مطابق تعریف زیر بدست آوریم.

$$P_e = \frac{\text{number of errors}}{\text{total number of symbols } (N)}$$

ابتدا باید سائز modulated\_symbols را متناسب کنیم برای این موضوع از تابع repmat استفاده کرده و تابع بدست آمده را در repeated\_symbols ذخیره می کنیم. همچنین در این بخش از تابع کمکی Error\_probability.m مطابق پیوست استفاده می کنیم. این تابع با دریافت repeated\_symbols و detected\_symbols با معیار ارزیابی

repeated\_symbols  $\neq$  detected\_symbols

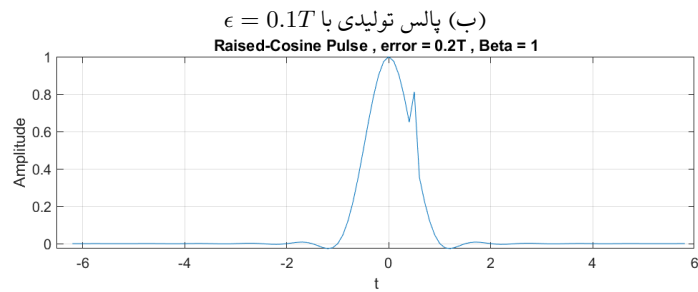
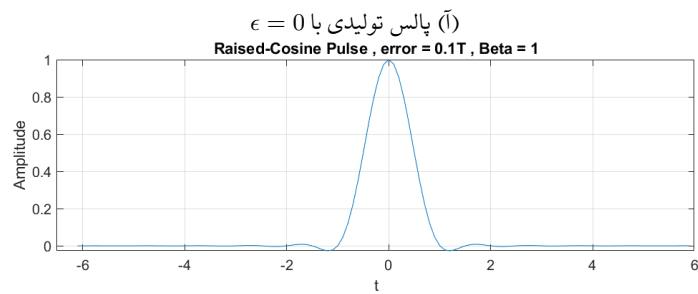
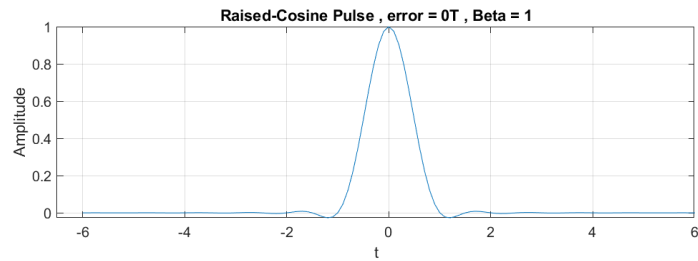
تعداد خطای هر ستون را پیدا می کند و در انتها در  $N$  تقسیم می کند. با بدست آوردن احتمال خطا ( $P_e$ ) در نهایت می توانیم احتمال خطا را به صورت زیر بر حسب مقادیر مختلف SNR\_dB رسم نماییم.



شکل ۲: احتمال خطا برای مقادیر مختلف SNR\_dB برای  $\beta = 0.5$

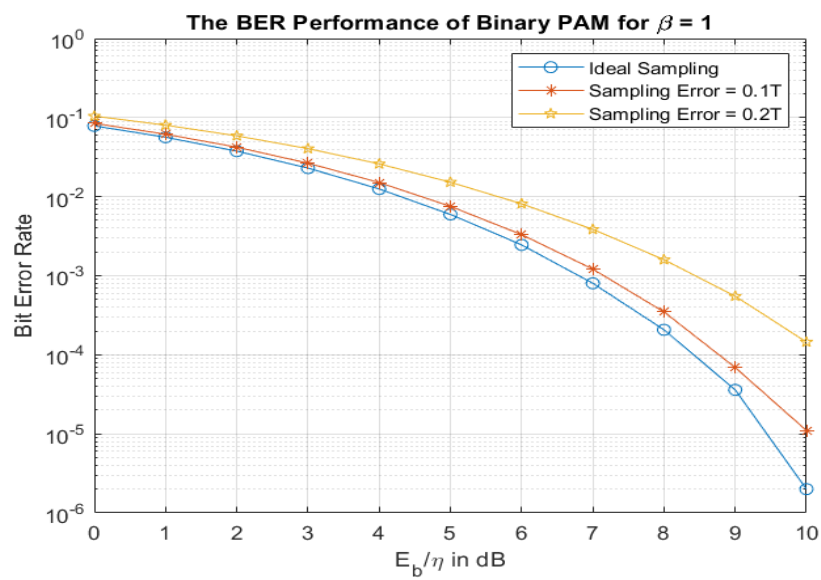
## ۶ محاسبه نتایج نهایی برای $\beta$ های دیگر

برای  $\beta = 0$  و  $\beta = 1$  مراحل بالا را تکرار میکنیم اما صرفاً نتایج نهایی را در این بخش ذکر می‌کنیم.  
برای  $\beta = 1$ :



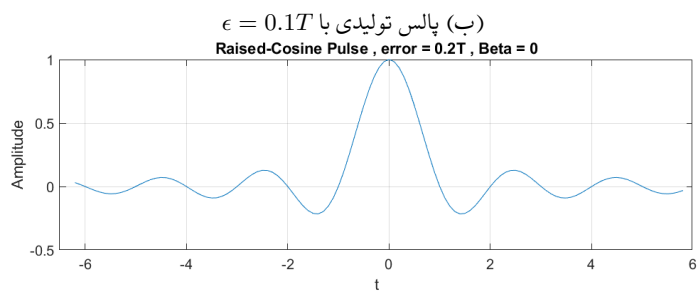
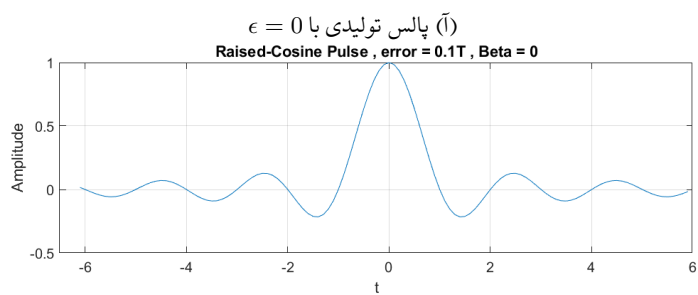
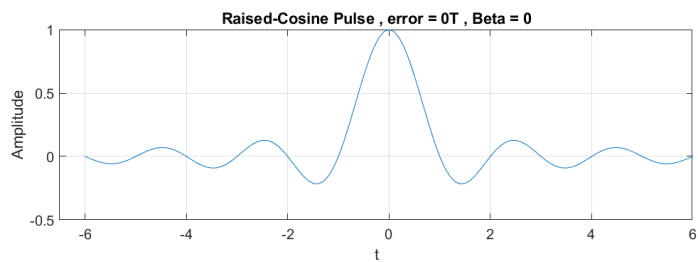
(ج) پالس تولیدی با  $\epsilon = 0.2T$

شکل ۳: پالس های تولید شده برای  $\beta = 1$



شکل ۴: احتمال خطا برای مقادیر مختلف SNR\_dB برای  $\beta = 1$

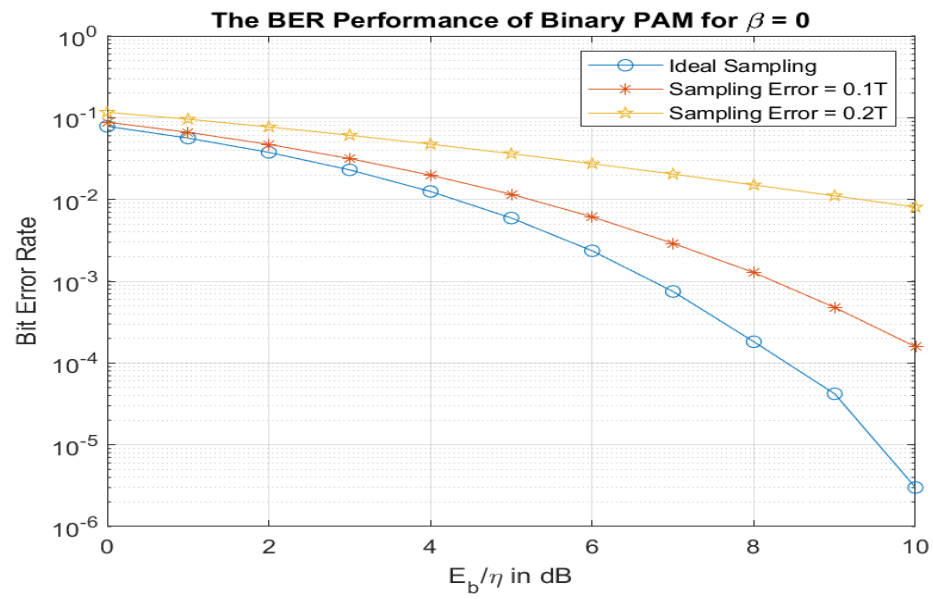
برای  $\beta = 0$ :



(ج) پالس تولیدی با  $\epsilon = 0.2T$

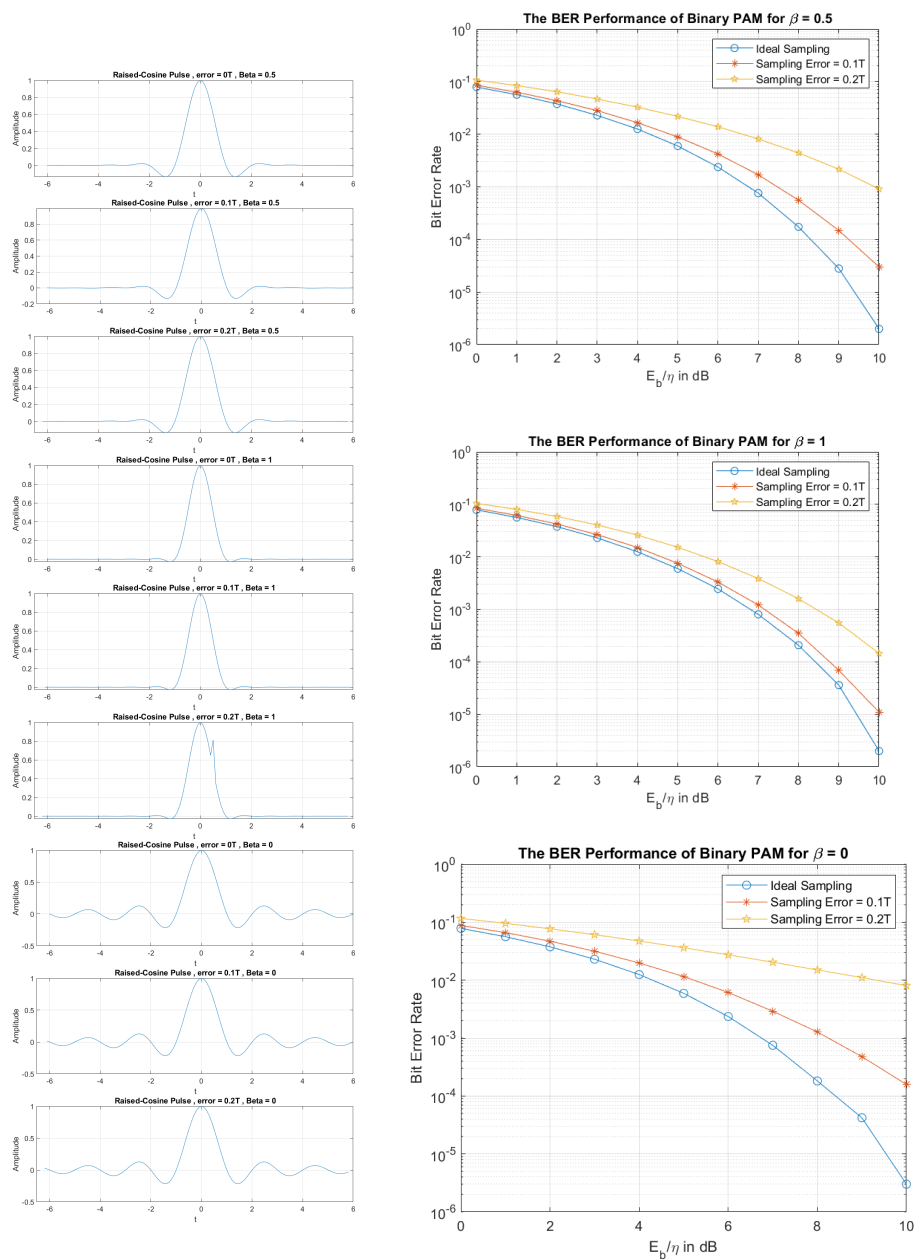
شکل ۵: پالس های تولید شده برای  $\beta = 0$





شکل ۶: احتمال خطا برای مقادیر مختلف SNR\_dB برای  $\beta = 0$

## ۷ مقایسه و نتیجه گیری



شکل ۷: تمام احتمال خطا و پالس های تولید شده

با دقت در شکل ۷ می توان مشاهدات زیر را بدست آورد:

۱. ابتدا مشاهده می شود برای Roll-off factors یا  $\beta$  کوچکتر، ریبیل و اعوجاج های بیشتری در شکل پالس RC داریم.

۲. برای Roll-off factors یا  $\beta$  کوچکتر حساسیت نسبت به خطای نمونه برداری بیشتر است. به صورتی که برای  $\beta = 0$  در  $\epsilon = 0.2T$  احتمال خطا تقریباً  $10^4$  برابر شده که در مقایسه با حالت  $\beta = 1$  که احتمال خطا در  $\epsilon = 0.2T$  نهایتاً ۱۰۰ برابر می شود، مقدار بسیار بزرگتری است. علت این موضوع در قسمت ۱ نیز اشاره شد، بدلیل ریبیل های بیشتر پالس RC در حالت با  $\beta = 0$  تابع دارای تغییرات تیز تری است و کمی خطای نمونه برداری می تواند مقدار ISI را به مقدار زیادی افزایش داده و موجب احتمال خطای بزرگی شود.

• فایل متلب مربوط به این تمرین در پیوست به نام `ca2_simulation.mlx` موجود می باشد. همچنین تابع های کمکی معرفی شده در گزارش نیز با نام های گفته شده در انتهای این فایل پیوست وجود دارد.