



پروژه درس سیستم‌های مخابراتی

نام و نام خانوادگی: امیرمهدی جعفری فشارکی

شماره دانشجویی: ۹۸۱۰۹۶۴۵

استاد درس: دکتر محمدرضا پاکروان

تاریخ: ۱۹ بهمن ۱۴۰۰

فهرست مطالب

۲	۱ توضیحات اولیه
۲	۲ پیاده‌سازی بلوک‌ها به صورت مجزا
۳	۳ انتقال دنباله تصادفی صفر و یک
۳	۱.۳ مدولاسیون PAM
۳	۱.۱.۳ شبیه‌سازی فرآیند ارسال دنباله بدون نویز
۶	۲.۱.۳ احتمال خطا بر حسب واریانس نویز
۸	۳.۱.۳ منظومه سیگنال برای ۶ واریانس نویز
۹	۲.۳ مدولاسیون PSK
۹	۱.۲.۳ شبیه‌سازی فرآیند ارسال دنباله بدون نویز
۱۳	۲.۲.۳ احتمال خطا بر حسب واریانس نویز
۱۴	۳.۲.۳ منظومه سیگنال برای ۶ واریانس نویز
۱۵	۳.۳ مدولاسیون FSK
۱۵	۱.۳.۳ شبیه‌سازی فرآیند ارسال دنباله بدون نویز
۱۹	۲.۳.۳ احتمال خطا بر حسب واریانس نویز
۲۰	۳.۳.۳ منظومه سیگنال برای ۶ واریانس نویز
۲۱	۴ انتقال دنباله‌ای از اعداد ۸ بیتی
۲۵	۵ کدینگ منبع
۲۵	۱.۵ مشکل کدهای C_1, C_2, C_3
۲۵	۲.۵ حل مساله بهینه‌سازی برای یافتن طول کلمه کدها برای منبع X
۲۶	۳.۵ کلمه‌کدهای منبع X
۲۶	۴.۵ طول متوسط کلمه‌کدها
۲۷	۵.۵ توابع خواسته شده
۲۷	۶.۵ شبیه‌سازی سیستم و رفتار $H_n(X)$ با بزرگ شدن X
۲۹	۷.۵ کدینگ بی‌اتلاف

۱ توضیحات اولیه

کدهای این پروژه در پوشه codes واقع شده اند. همچنین لازم به ذکر است که این گزارش با استفاده از L^AT_EX تهیه شده است و تمام محتویات مربوط به این گزارش در پوشه report و نمودارها و تصاویر استفاده شده در پوشه pics که خود در همین پوشه واقع است، قرار دارند.

۲ پیاده سازی بلوک ها به صورت مجزا

برای این بخش تمامی توابع خواسته شد در همان پوشه codes، مطابق با نامگذاری خواسته شده قرار گرفته اند.

۱. تابع Divide یک دنباله از صفر و یک را به عنوان ورودی می گیرد و در خروجی، این دنباله را به دو دنباله با طول نصف دنباله ابتدایی تقسیم می کند. این تقسیم بندی به این شکل انجام شده است که به ترتیب یکی از بیت ها را به دنباله اول و بعدی را به دنباله دوم و دوباره همین روند را تکرار می کند. به عبارت دیگر مقادیر مرتبط به n های زوج را به یک دنباله و مقادیر مربوط به n های فرد را به دنباله دوم می دهد. دلیل این نوع تقسیم نیز برای شباهت سیستم به یک سیستم real-time می باشد چرا که در صورتی که بخواهیم مثلاً نصف اول دنباله را در دنباله اول و نصف دوم را در دنباله دوم قرار بدهیم، در این صورت سیستم باید در عمل ابتدا کامل یک دنباله را تشکیل بدهد و بفرستد و سپس دنباله بعدی را بفرستد که عملاً از لحاظ منطقی، فلسفه پشت این نصف کردن داده ها که مخابره موازی برای افزایش سرعت می باشد را نقض می کند. همچنین تابع Combine دقیقاً معکوس این سیستم می باشد و دو دنباله را می گیرد و به ترتیب یک بیت از دنباله اول برداشته و یکی از دنباله دوم و همین روند را ادامه می دهد تا دنباله خروجی را بسازد.

۲. تابع PulseShaping در ورودی خود یک دنباله از صفر و یک ها، یک رشته نمایش دهنده سیگنال متناظر با صفر و یکی متناظر با یک به عنوان ورودی می گیرد. همچنین برای تولید این رشته ها ابتدا به صورت آرایه ای از اعداد حقیقی این مقادیر ذخیره و سپس با استفاده از تابع num2str به شکل رشته در آورده شده اند. همچنین برای تبدیل آن ها به عدد نیز از تابع str2num استفاده شده است.

۳. تابع AnalogMod تنها کاری که می کند این است که سیگنال

$$x_c(t) = x_1(t) \cos(2\pi f_c t) + x_2(t) \sin(2\pi f_c t)$$

را تولید و خروجی می دهد.

۴. برای پیاده سازی تابع Channel از تابع bandpass استفاده شده است تا رفتار فیلتری کانال را مدل سازی کنیم.

۵. برای تابع AnalogDemod ابتدا سیگنال در $\sin(2\pi f_c t)$ و $\cos(2\pi f_c t)$ ضرب شده و سپس با استفاده از تابع lowpass بخش پایین گذر آن فیلتر شده و در نهایت دو خروجی تولید شده است.

۶. در تابع MatchedFilt پاسخ ضربه برای هر کدام از پالس ها در حقیقت flip شده هر کدام از پالس ها می باشد. سپس با کانوالو کردن این دو پاسخ ضربه در سیگنال اصلی، سیگنال جدیدی به دست می آید. در نهایت از هر کدام از این سیگنال ها در لحظاتی که متناظر با وسط پالس می باشند، نمونه برداری شده و برای پیش بینی ۰ یا ۱ بودن بیت فرستاده شده کافی است مقایسه شود که کدام سیگنال به دست آمده از Matched Filter دارای مقدار بیشتری می باشد و به این شکل به راحتی ۰ یا ۱ بودن سیگنال تشخیص داده می شود. همچنین لازم به ذکر

است که برای دو سیگنال دیگر که به عنوان خروجی خواسته شده، خروجی پس از نمونه برداری نرمالیزه نیز می‌شود یعنی بر ماکسیم دامنه تقسیم می‌شود تا مقادیر به دست آمده بین منفی ۱ و ۱ باشند.

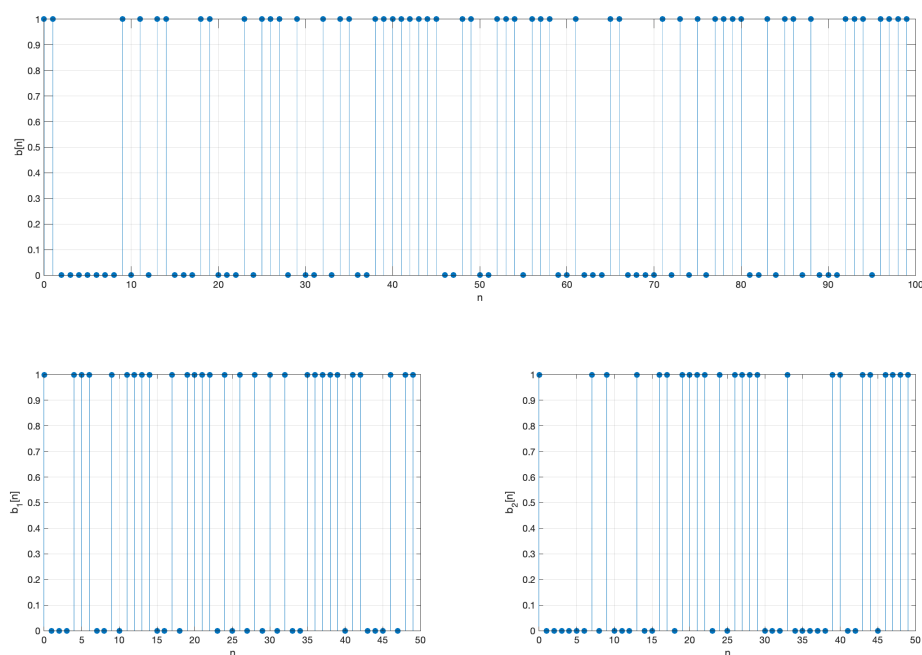
۳ انتقال دنباله تصادفی صفر و یک

۱.۳ مدولاسیون PAM

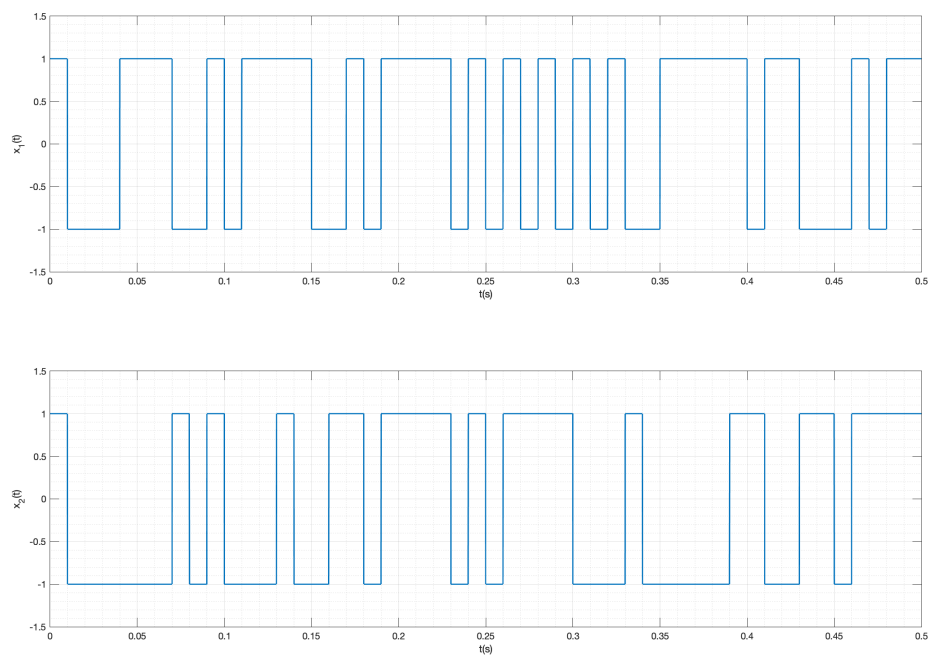
برای این مدولاسیون، برای هر پالس متناظر با بیت یک از پالس ثابتی با دامنه یک و برای بیت صفر از پالسی با دامنه منفی یک استفاده شده است. همچنین طول دنباله فرستاده شده ۱۰۰۰ بیت می‌باشد که در قسمت ب دقت خوبی برای احتمال خطا بدهد.

۱.۱.۳ شبیه‌سازی فرآیند ارسال دنباله بدون نویز

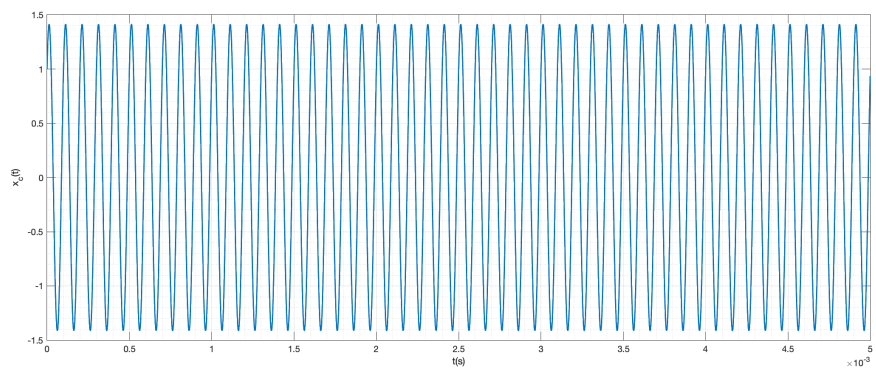
کد این بخش در فایل Part1_PAM_no_noise.m قرار دارد. در ادامه نمودارهای خروجی هر کدام از بلوک‌ها رسم شده است. لازم به ذکر است که برخی از نمودارها به شکل بزرگنمایی شده و فقط بخشی از آنها رسم شده است.



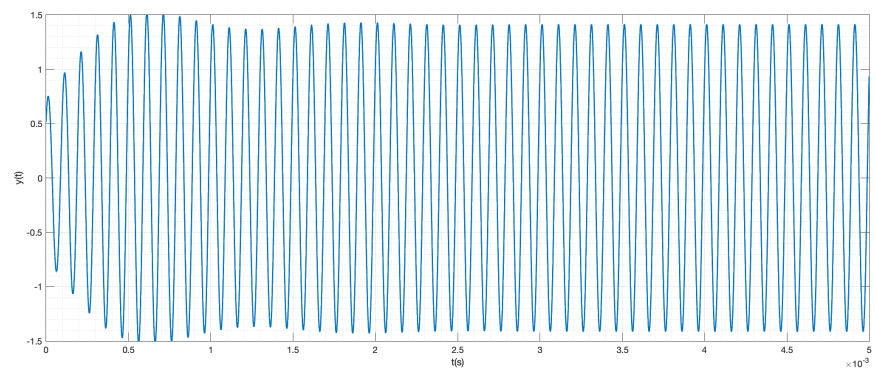
شکل ۱: سیگنال اولیه و سیگنال‌های تقسیم شده



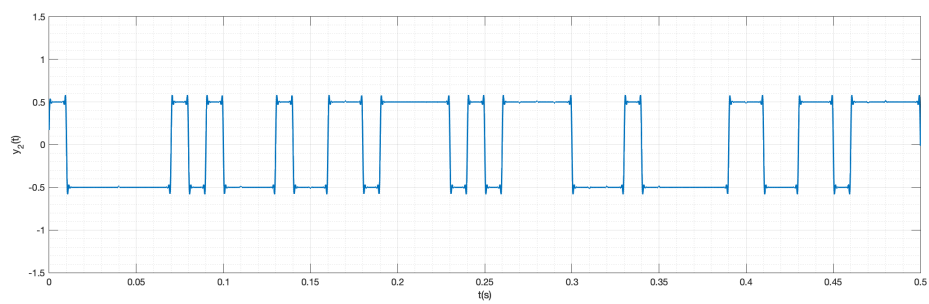
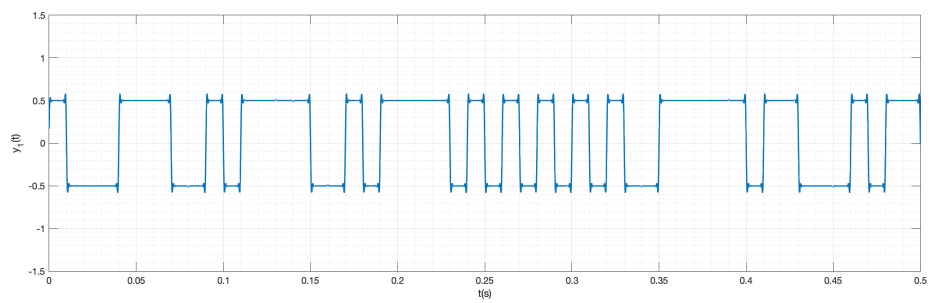
شکل ۲: خروجی های بلوک PulseShaping



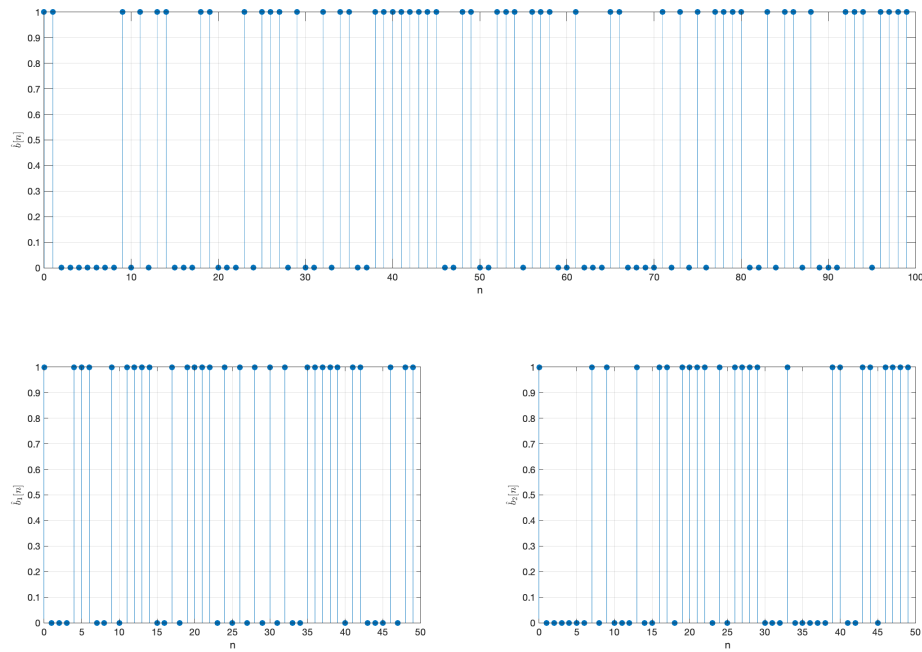
شکل ۳: خروجی بلوک AnalogMod



شکل ۴: خروجی بلوک Channel



شکل ۵: خروجی بلوک AnalogDemod

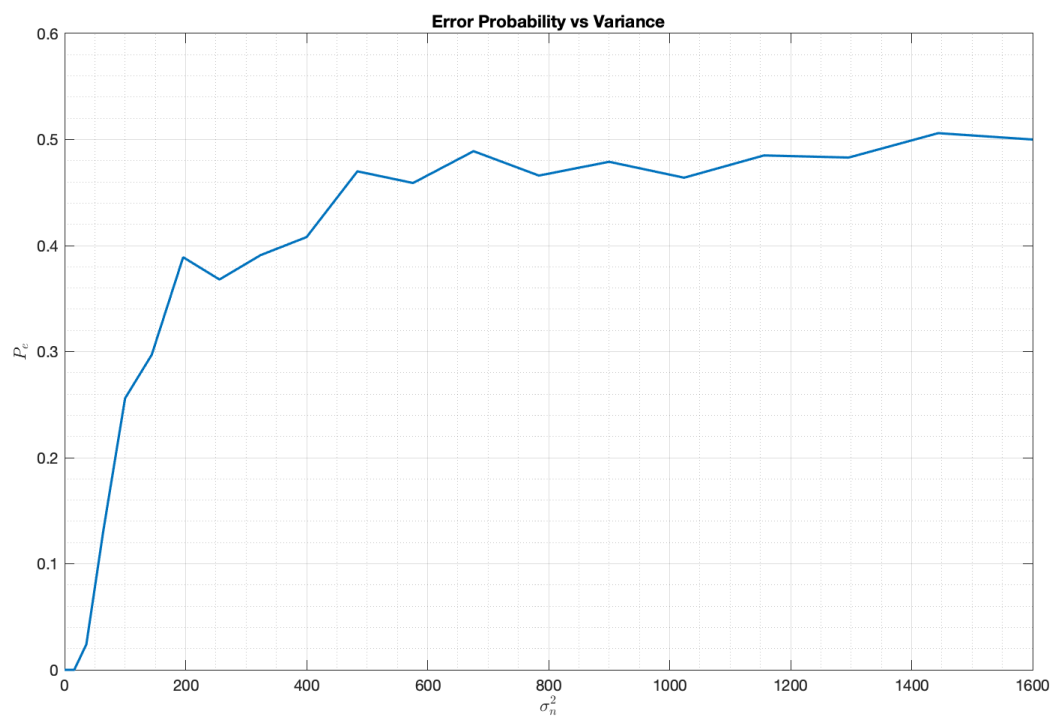


شکل ۶: خروجی بلوک MatchedFilt و دنباله پیشبینی شده نهایی

۲.۱.۳ احتمال خطا بر حسب واریانس نویز

کد این بخش در فایل Part1_PAM_noise.m قرار دارد. برای این بخش در تولید نویز از تابع awgn استفاده شده است و واریانس نویز را در بازه ۰ تا ۴۰ با پله ۲ تغییر داده و در نهایت برای محاسبه احتمال خطا، تعداد بیت‌هایی که غلط تشخیص داده شده است به تعداد کل بیت‌ها تقسیم شده است و به این طریق احتمال خطا به دست آمده است. همچنین لازم به ذکر است که با افزایش واریانس، از جایی به بعد مقدار واقعی بیت در عمل بی تاثیر شده (چرا که توان نویز بسیار بیشتر از توان داده شده) و عملاً تنها عامل تاثیرگذار در تشخیص بیت، نویز ما می‌شود. در نتیجه علامت نویز در پاسخ نهایی موثر شده و به همین دلیل در نیمی از حالات بیت نهایی یک تشخیص داده شده و در نیمی دیگر صفر و از آن جا که احتمال فرستاده شدن بیت یک و صفر نیز برابر می‌باشد، احتمال خطا در واریانس‌های بالا برابر با 0.5 می‌شود.

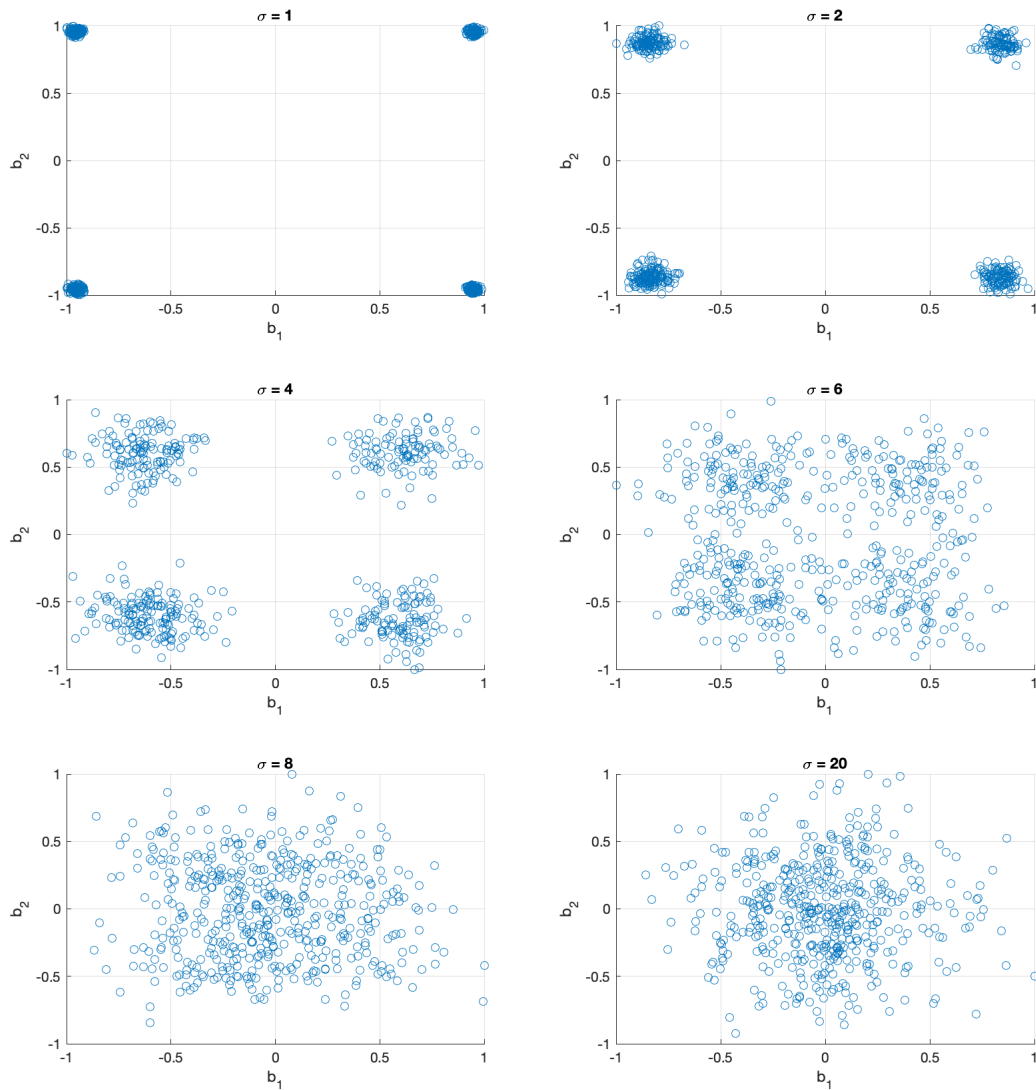
$$\begin{aligned}
 P\{b \neq \hat{b}\} &= P\{b = 1\}P\{b = 1, \hat{b} = 0\} + P\{b = 0\}P\{b = 0, \hat{b} = 1\} \\
 &= \frac{1}{2} \frac{1}{2} + \frac{1}{2} \frac{1}{2} = \frac{1}{2}
 \end{aligned}
 \tag{1}$$



شکل ۷: نمودار احتمال خطا بر حسب واریانس نویز

۳.۱.۳ منظومه سیگنال برای ۶ واریانس نویز

کد این بخش در فایل Part1_PAM_scatter.m قرار دارد. در شکل زیر، نمودارهای منظومه سیگنال برای ۶ واریانس متفاوت رسم شده است. همان‌طور که مشاهده می‌شود، با افزایش واریانس نویز، کم‌کم دقت تشخیص کاهش می‌یابد تا جایی که در واریانس‌های بالا عملاً الگوی مشخصی برای تشخیص بیت‌ها وجود نداشته و منظومه سیگنال مستقل از بیت‌های فرستاده شده می‌شود.



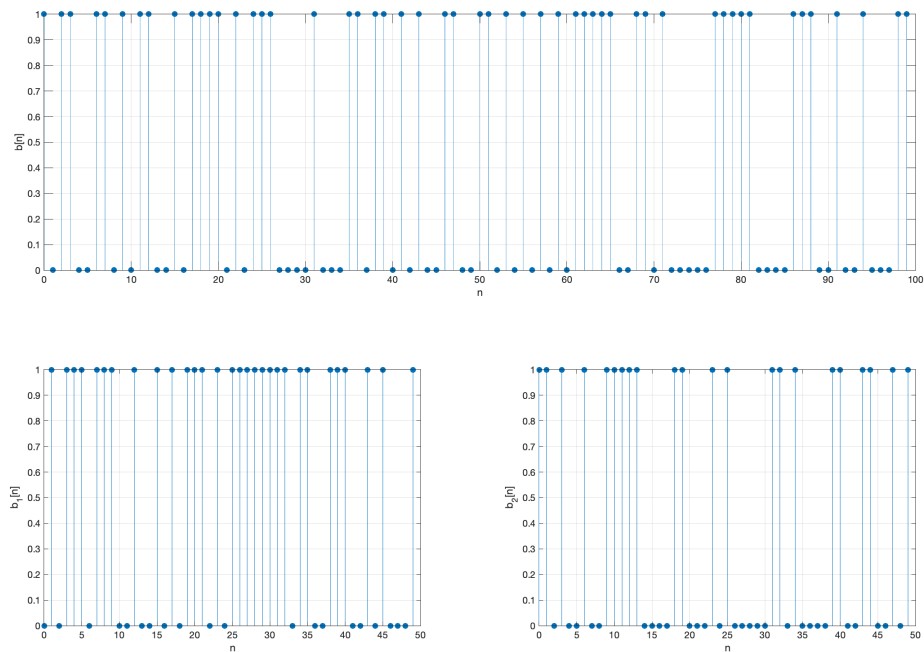
شکل ۸: منظومه سیگنال برای ۶ واریانس نویز

۲.۳ مدولاسیون PSK

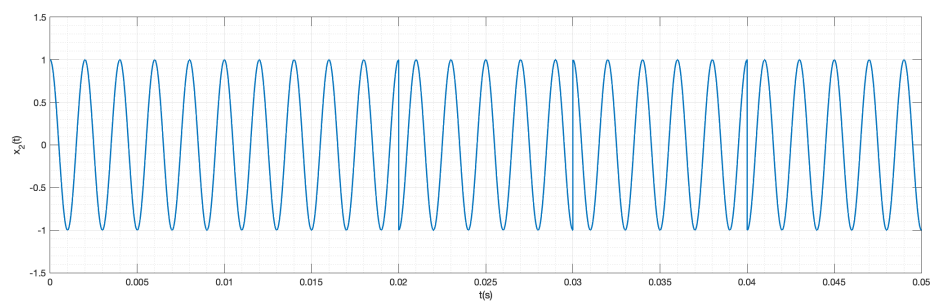
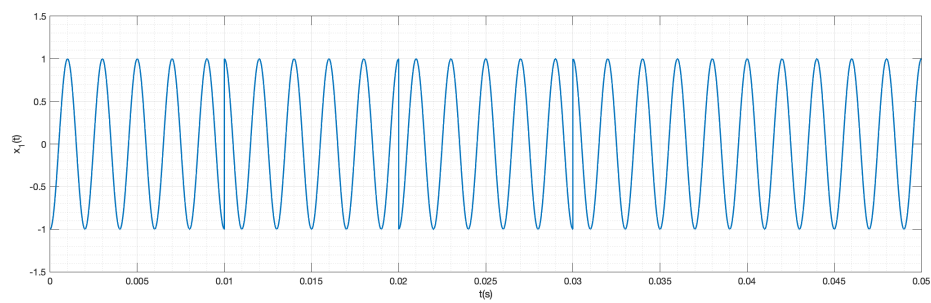
برای این مدولاسیون، برای هر پالس متناظر با بیت یک از یک سیگنال سینوسی با فرکانس ۵۰۰ هرتز و برای صفر منفی همین سیگنال استفاده شده است. همچنین طول دنباله فرستاده شده ۱۰۰۰ بیت می‌باشد که در قسمت ب دقت خوبی برای احتمال خطا بدهد.

۱.۲.۳ شبیه‌سازی فرآیند ارسال دنباله بدون نویز

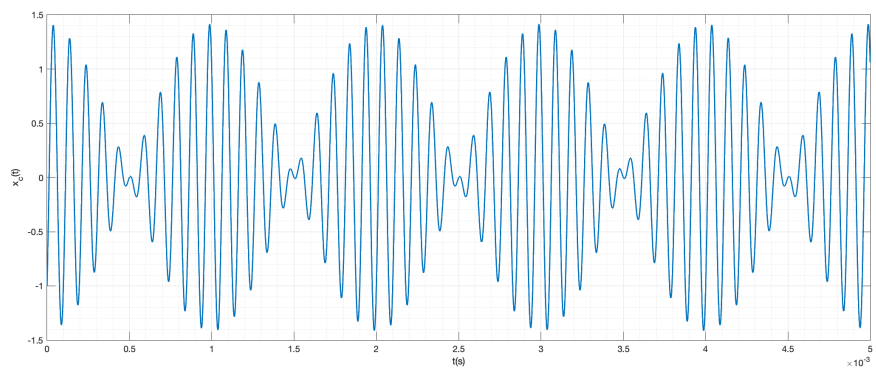
کد این بخش در فایل Part1_PSK_no_noise.m قرار دارد. در ادامه نمودارهای خروجی هر کدام از بلوک‌ها رسم شده است. لازم به ذکر است که برخی از نمودارها به شکل بزرگنمایی شده و فقط بخشی از آن‌ها رسم شده است.



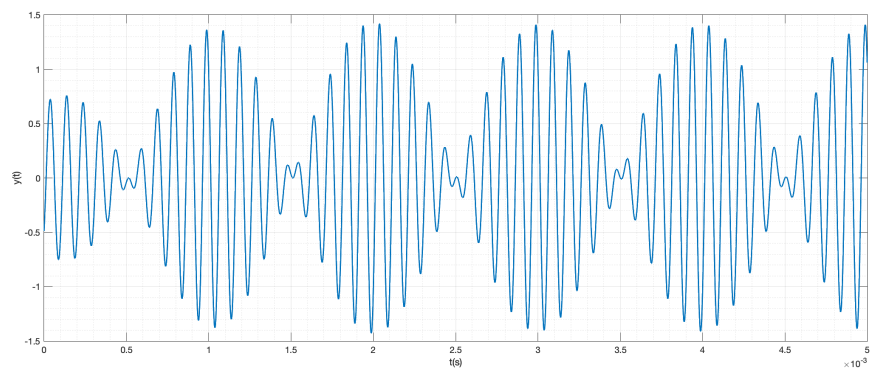
شکل ۹: سیگنال اولیه و سیگنال‌های تقسیم شده



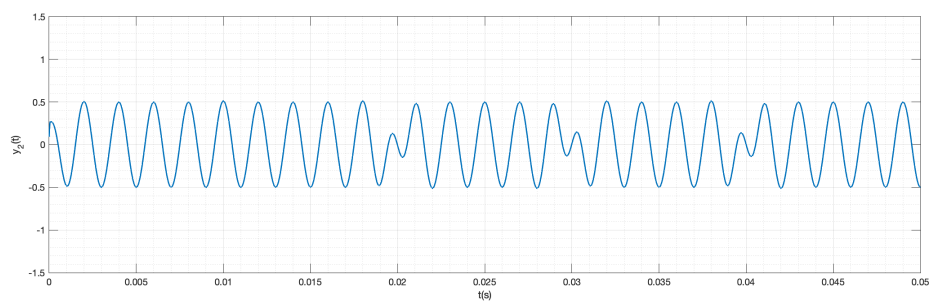
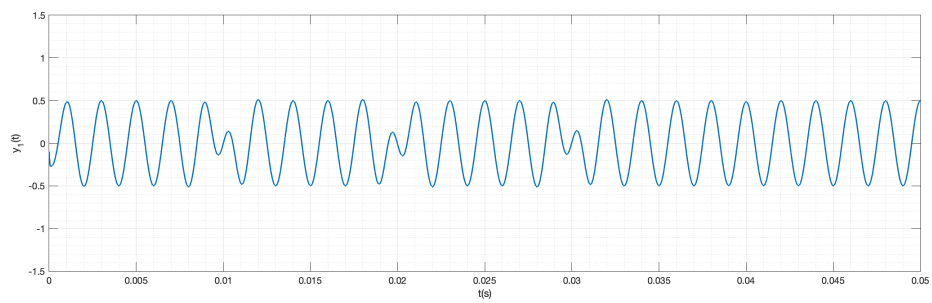
شکل ۱۰: خروجی های بلوک PulseShaping



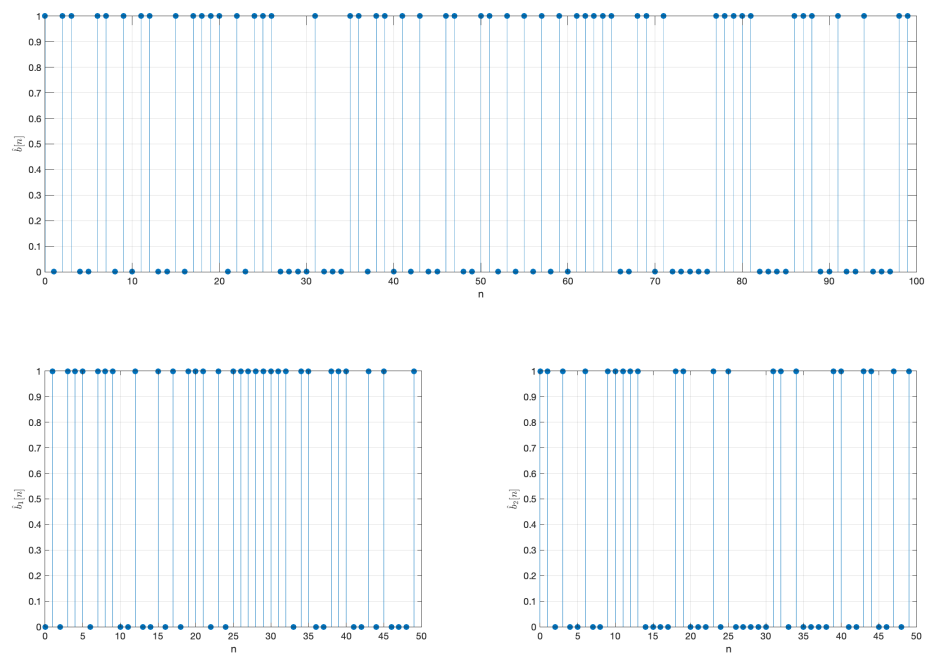
شکل ۱۱: خروجی بلوک AnalogMod



شکل ۱۲: خروجی بلوک Channel



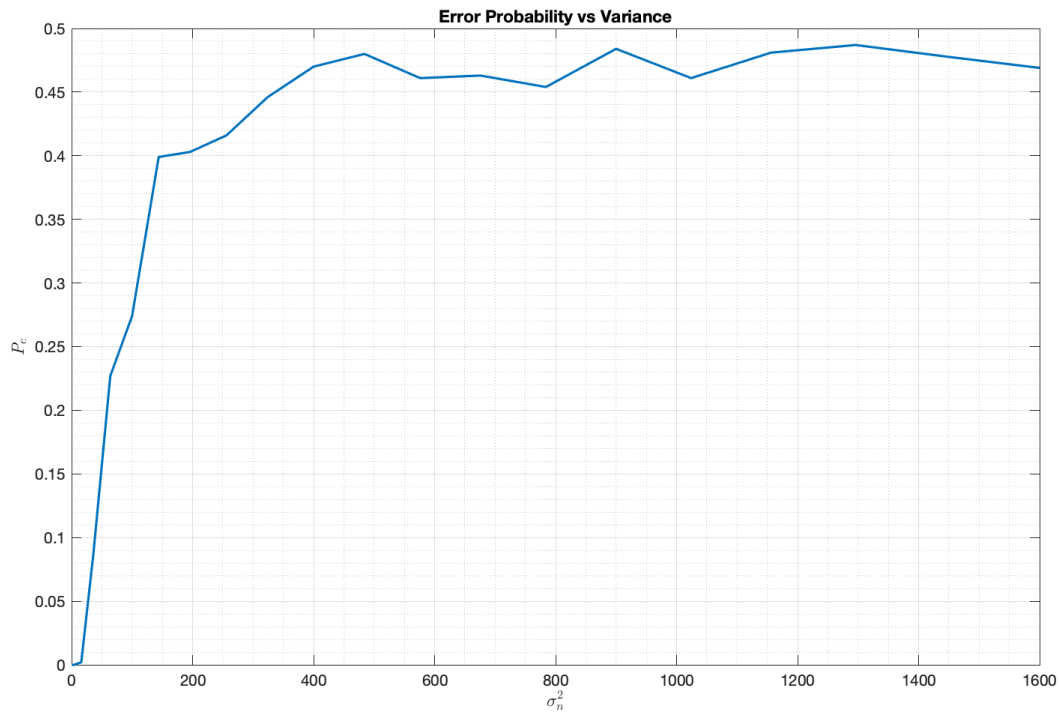
شکل ۱۳: خروجی بلوک AnalogDemod



شکل ۱۴: خروجی بلوک MatchedFilt و دنباله پیشبینی شده نهایی

۲.۲.۳ احتمال خطا بر حسب واریانس نویز

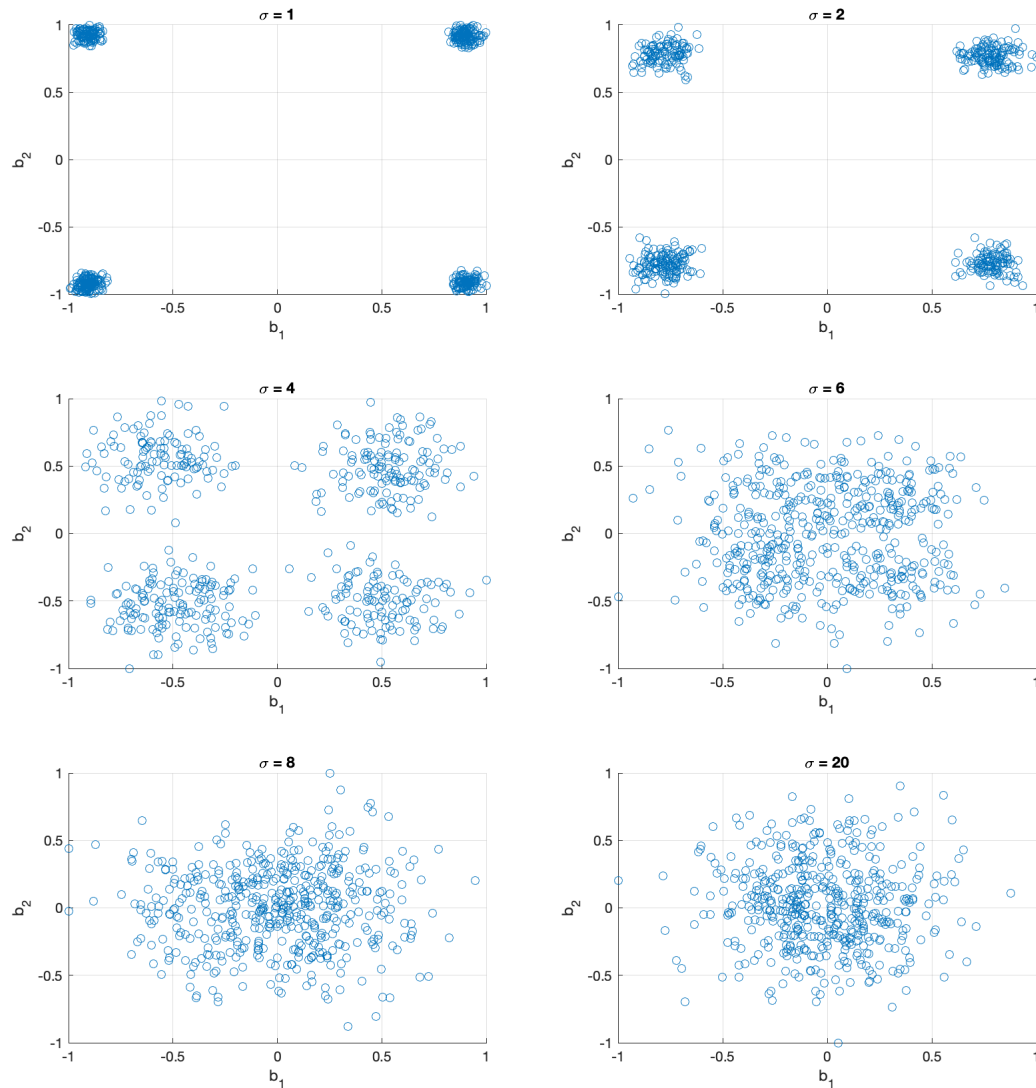
کد این بخش در فایل Part1_PSK_noise.m قرار دارد. در این بخش نیز همه چیز دقیقاً مشابه مدولاسیون PAM بوده و باز هم در واریانس‌های زیاد، احتمال خطا به نیم میل می‌کند. البته نکته لازم به ذکر این است که در این مدولاسیون، همان‌طور که در مقایسه دو نمودار نیز قابل مشاهده است، حساسیت احتمال خطا به واریانس نویز بیشتر می‌باشد و به عبارت دیگر، با افزایش واریانس نویز، احتمال خطا در مدولاسیون PSK بیشتر می‌شود که به این معنی است که برای داشتن یک خطای یکسان، در مدولاسیون PSK باید انرژی بیشتری نسبت به مدولاسیون PAM صرف کرد.



شکل ۱۵: نمودار احتمال خطا بر حسب واریانس نویز

۳.۲.۳ منظومه سیگنال برای ۶ واریانس نویز

کد این بخش در فایل Part1_PSK_scatter.m قرار دارد. در شکل زیر، نمودارهای منظومه سیگنال برای ۶ واریانس متفاوت رسم شده است. همان‌طور که مشاهده می‌شود، با افزایش واریانس نویز، کم‌کم دقت تشخیص کاهش می‌یابد تا جایی که در واریانس‌های بالا عملاً الگوی مشخصی برای تشخیص بیت‌ها وجود نداشته و منظومه سیگنال مستقل از بیت‌های فرستاده شده می‌شود. همچنین در مقایسه با مدولاسیون PAM به ازای یک واریانس یکسان، پراکندگی بیشتری در داده‌ها وجود دارد که همان نتیجه بخش قبل را نشان می‌دهد.



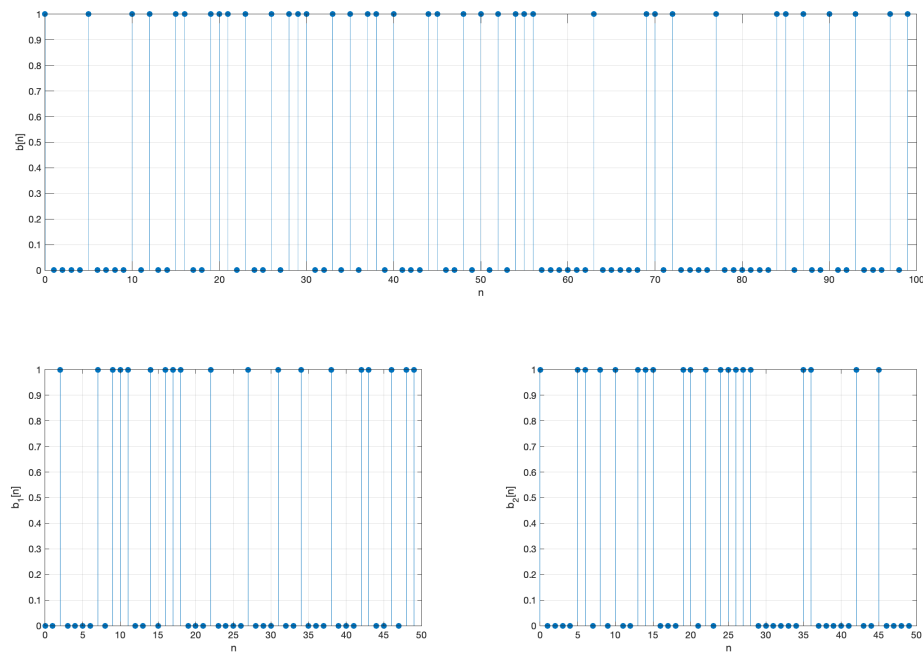
شکل ۱۶: منظومه سیگنال برای ۶ واریانس نویز

۳.۳ مدولاسیون FSK

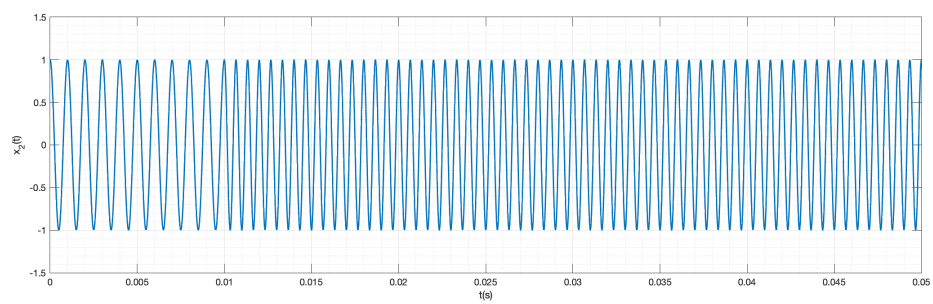
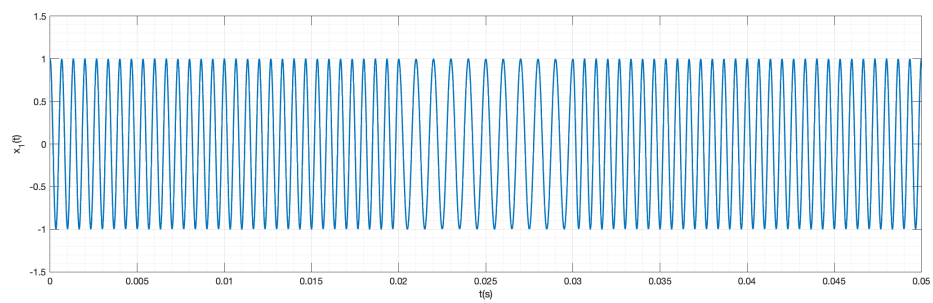
برای این مدولاسیون، برای هر پالس متناظر با بیت یک از یک سیگنال سینوسی با فرکانس ۱۰۰۰ هرتز و برای صفر از سیگنال سینوسی با فرکانس ۱۵۰۰ هرتز استفاده شده است. همچنین طول دنباله فرستاده شده ۱۰۰۰ بیت می‌باشد که در قسمت ب دقت خوبی برای احتمال خطا بدهد.

۱.۳.۳ شبیه‌سازی فرآیند ارسال دنباله بدون نویز

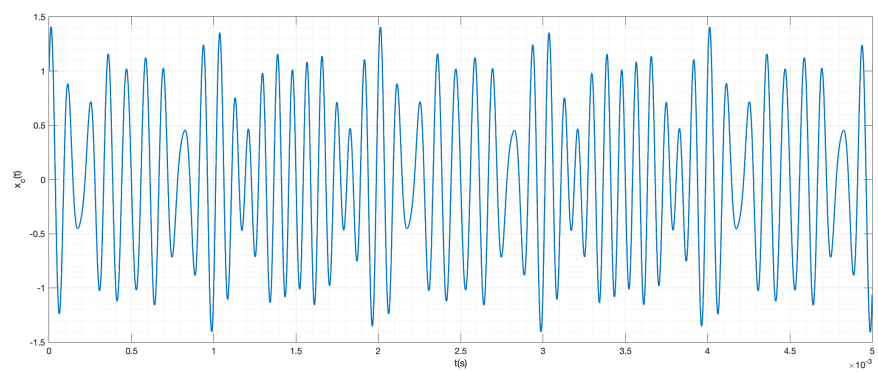
کد این بخش در فایل Part1_FSK_no_noise.m قرار دارد. در ادامه نمودارهای خروجی هر کدام از بلوک‌ها رسم شده است. لازم به ذکر است که برخی از نمودارها به شکل بزرگنمایی شده و فقط بخشی از آنها رسم شده است.



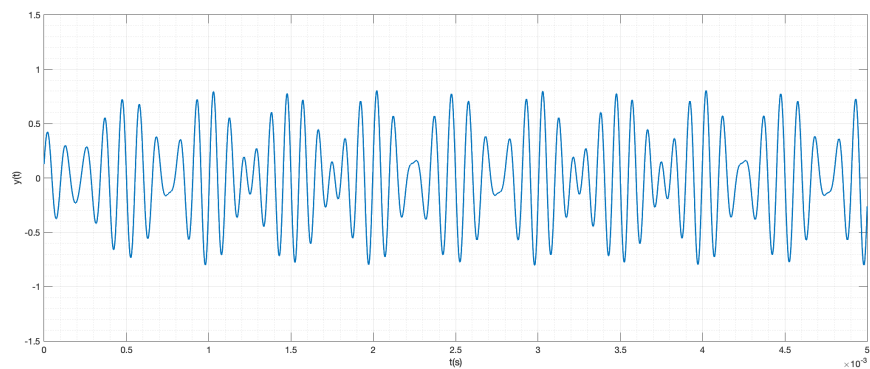
شکل ۱۷: سیگنال اولیه و سیگنال‌های تقسیم شده



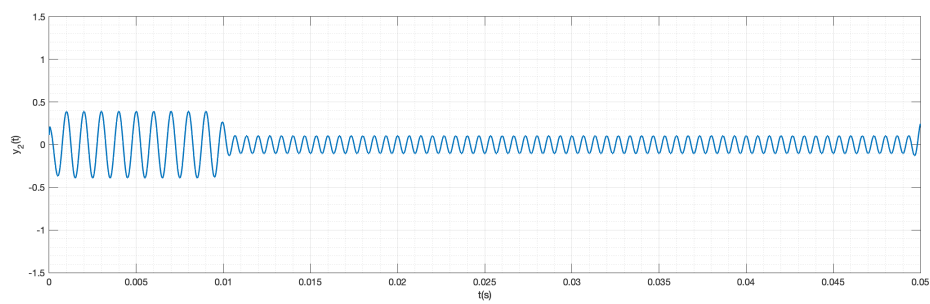
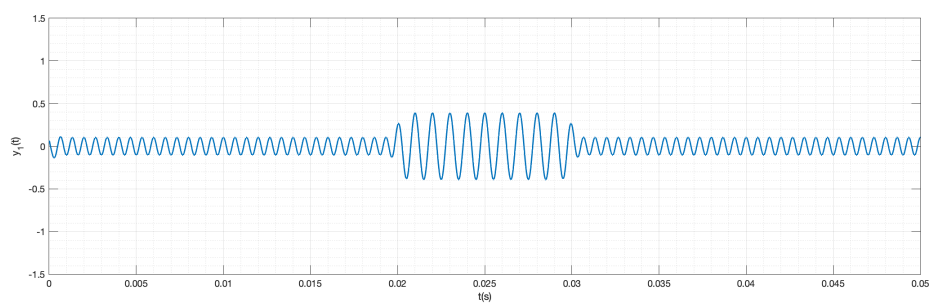
شکل ۱۸: خروجی های بلوک PulseShaping



شکل ۱۹: خروجی بلوک AnalogMod

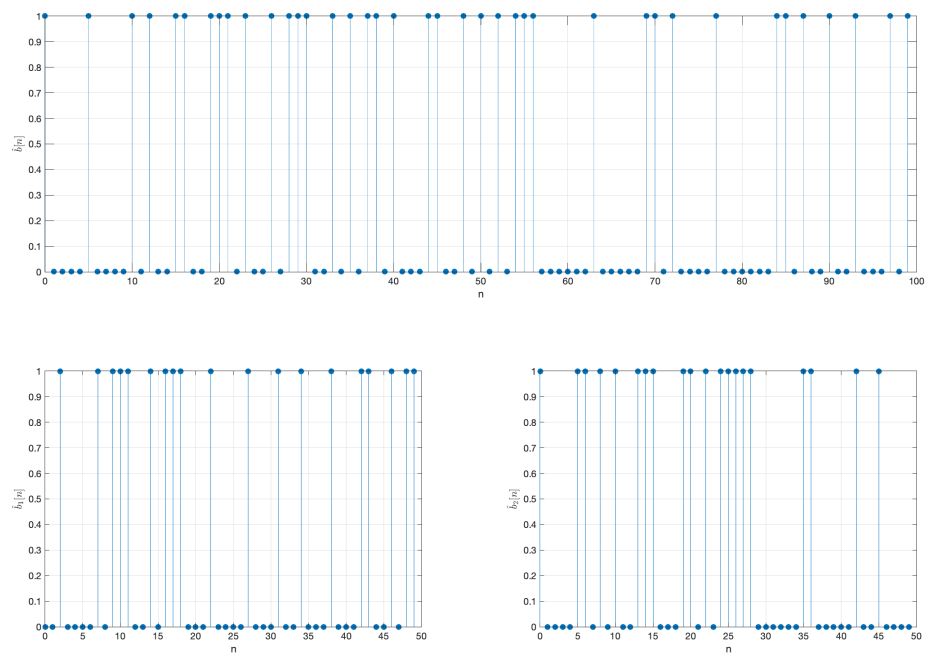


شکل ۲۰: خروجی بلوک Channel



شکل ۲۱: خروجی بلوک AnalogDemod

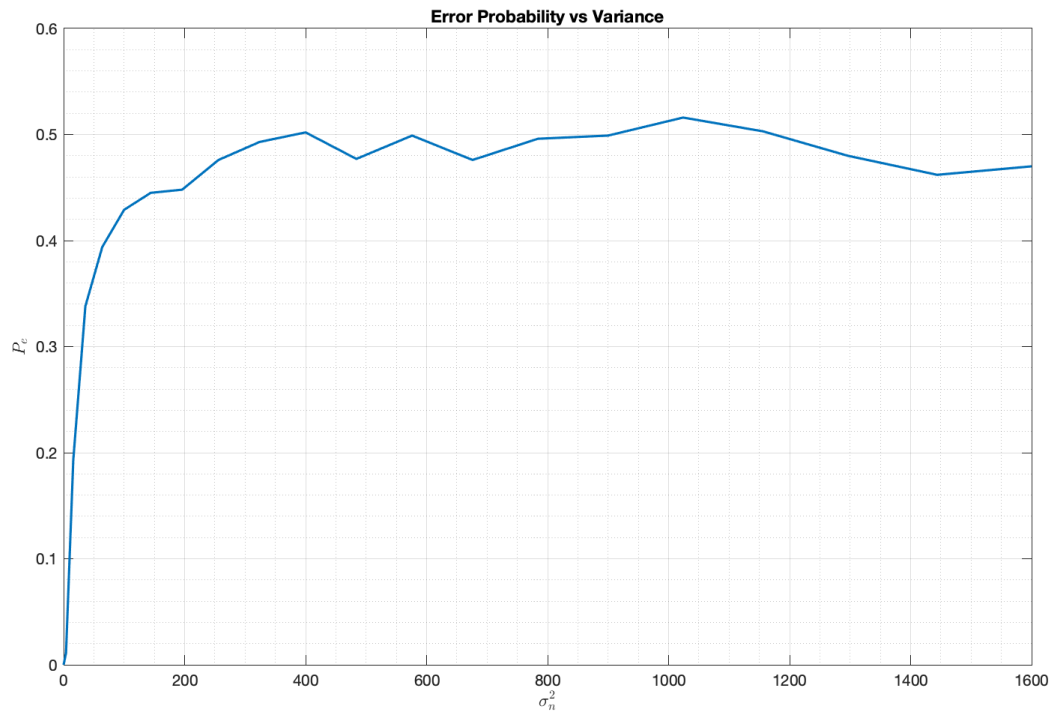
در شکل بالا مشاهده می‌شود که برای سیگنال متناظر با بیت یک، دامنه سیگنال چند برابر سیگنال متناظر با بیت صفر می‌باشد. دلیل این موضوع نیز پهنای باند کانال است که باعث تضعیف سیگنال‌های متناظر با بیت صفر می‌شود.



شکل ۲۲: خروجی بلوک MatchedFilt و دنباله پیش‌بینی شده نهایی

۲.۳.۳ احتمال خطا بر حسب واریانس نویز

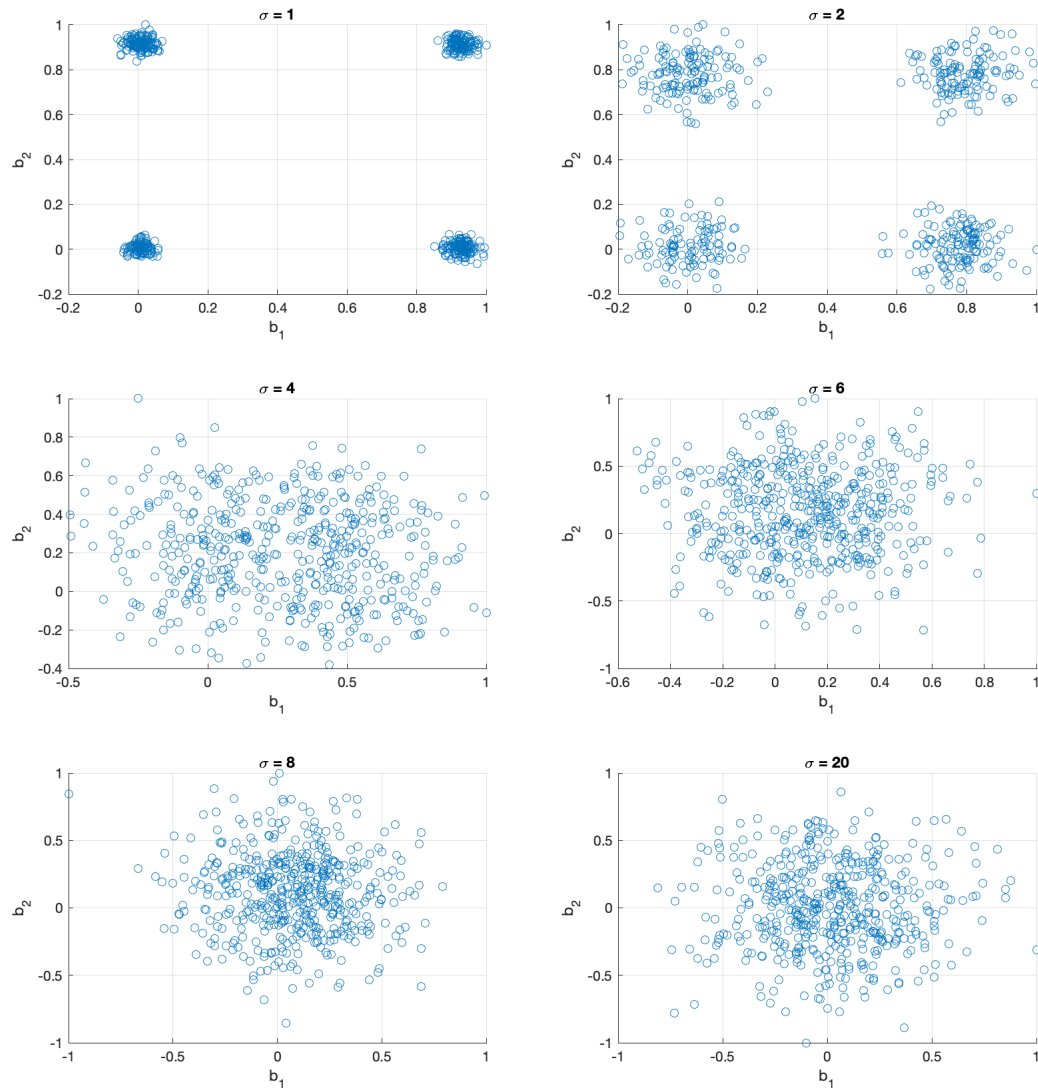
کد این بخش در فایل Part1_FSK_noise.m قرار دارد. در این بخش نیز همه چیز دقیقاً مشابه مدولاسیون PAM بوده و باز هم در واریانس‌های زیاد، احتمال خطا به نیم میل می‌کند. همچنین لازم به ذکر است که در این مدولاسیون، حساسیت به واریانس نویز نسبت به هردو مدولاسیون قبلی بیشتر بوده و به همین دلیل شیب نمودار در ابتدا بسیار بیشتر می‌باشد. این به این معنی است که برای آشکارسازی صحیح برای یک واریانس نویز یکسان، در این مدولاسیون به انرژی سیگنال فرستاده شده بیشتری به نسبت دو مدولاسیون قبلی نیاز است.



شکل ۲۳: نمودار احتمال خطا بر حسب واریانس نویز

۳.۳.۳ منظومه سیگنال برای ۶ واریانس نویز

کد این بخش در فایل Part1_FSK_scatter.m قرار دارد. در شکل زیر، نمودارهای منظومه سیگنال برای ۶ واریانس متفاوت رسم شده است. همان‌طور که مشاهده می‌شود، با افزایش واریانس نویز، کم‌کم دقت تشخیص کاهش می‌یابد تا جایی که در واریانس‌های بالا عملاً الگوی مشخصی برای تشخیص بیت‌ها وجود نداشته و منظومه سیگنال مستقل از بیت‌های فرستاده شده می‌شود. همچنین در مقایسه با دو مدولاسیون پیشین، به ازای یک واریانس یکسان، پراکندگی بیشتری در داده‌ها وجود دارد که همان نتیجه بخش قبل را نشان می‌دهد.



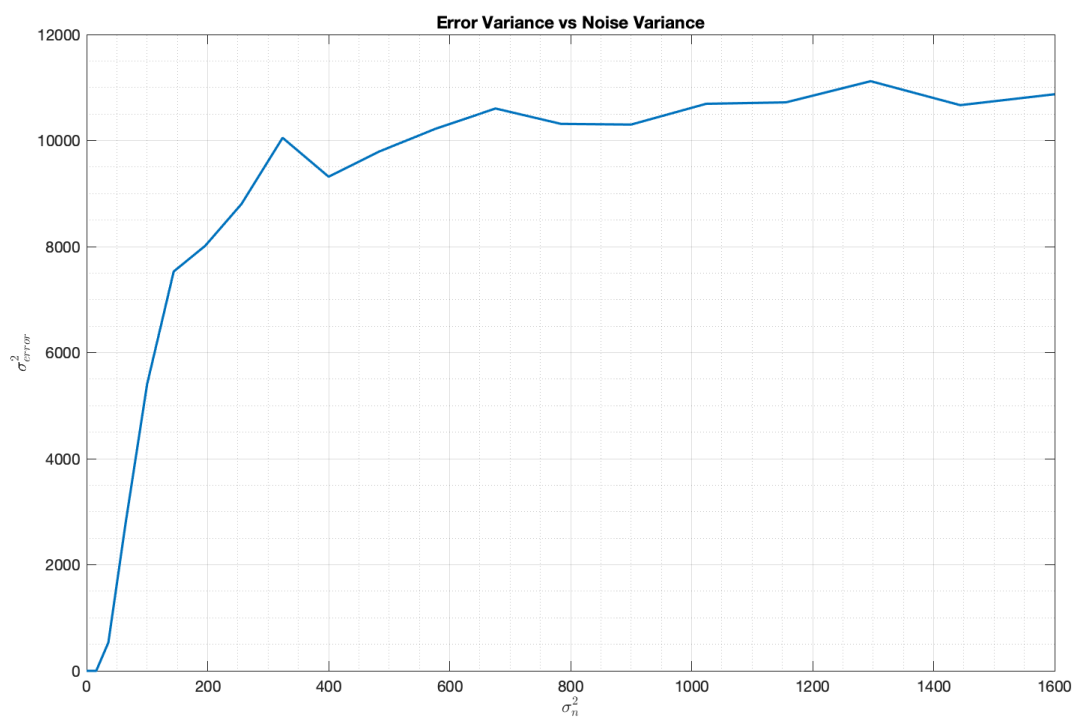
شکل ۲۴: منظومه سیگنال برای ۶ واریانس نویز

۴ انتقال دنباله‌ای از اعداد ۸ بیتی

کد این بخش در فایل Part2.m قرار دارد.

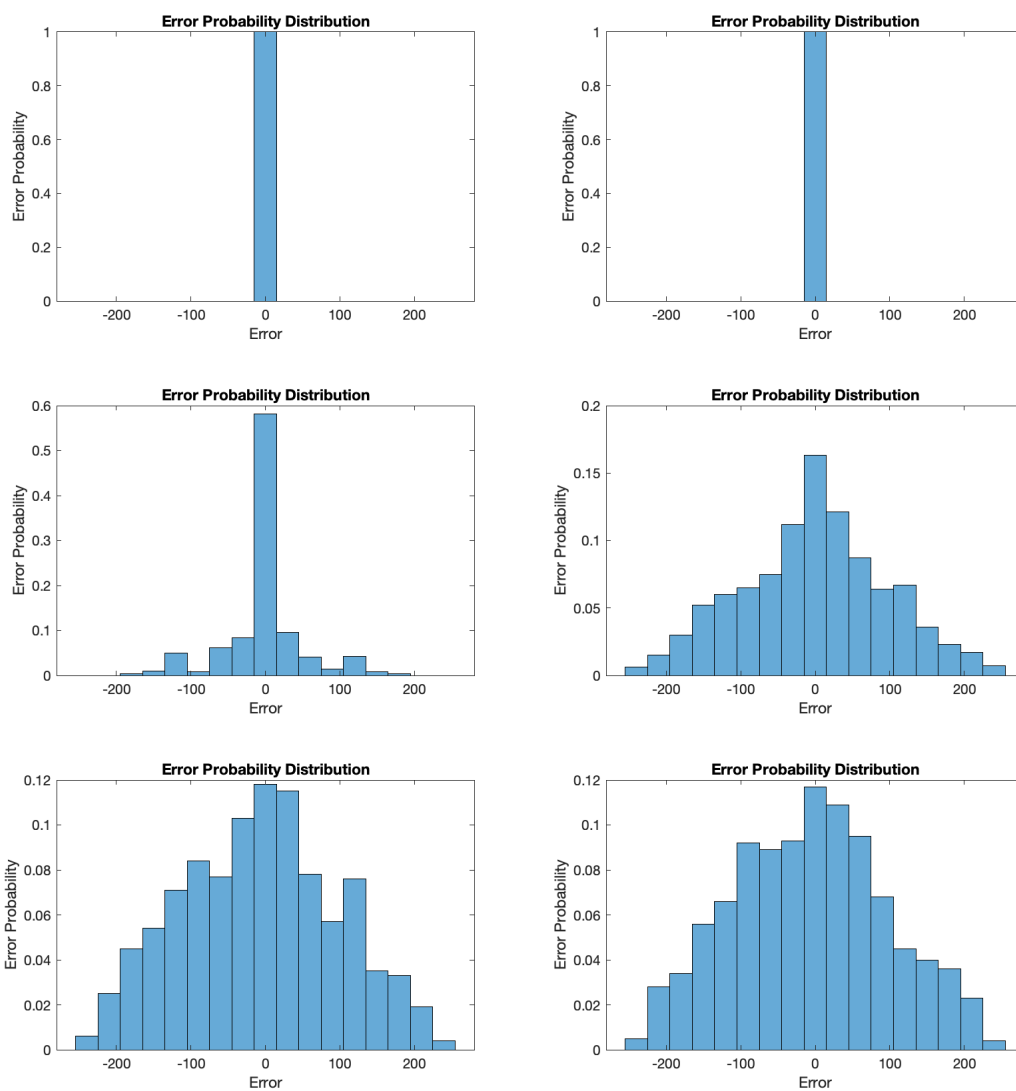
۱. توابع خواسته شده برای این قسمت در فایل‌های SourceGenerator.m و OutputDecoder.m قرار دارند و برای پیاده‌سازی این توابع از توابع de2bi و bi2de استفاده شده است.

۲. نمودار واریانس خطا بر حسب واریانس نویز مطابق شکل زیر می‌باشد. همچنین برای به دست آوردن دقت بالا، از دنباله‌ای با ۱۰۰۰ عدد استفاده شده است که معادل با ۸۰۰۰ بیت می‌باشد.



شکل ۲۵: نمودار واریانس خطا بر حسب واریانس نویز

۳. برای پیاده‌سازی این بخش، از ۶ انحراف معیار نویز ۱، ۲، ۸، ۱۶، ۳۲، ۴۰ به ترتیب استفاده شده است. نمودارهای احتمال به دست آمده مطابق شکل زیر می‌باشد. در دو واریانس اول، همان‌طور که مشاهده می‌شود احتمال خطای صفر برابر با ۱ می‌باشد به این معنی که هیچ خطایی رخ نمی‌دهد.



شکل ۲۶: نمودارهای توزیع احتمال

برای حالت حدی داریم:

$$\begin{cases} x = 2^7 b_7 + 2^6 b_6 + \dots + b_0 \\ y = 2^7 \hat{b}_7 + 2^6 \hat{b}_6 + \dots + \hat{b}_0 \end{cases} \quad (2)$$

از طرفی در حالت حدی همان طور که پیش تر نیز گفتیم، بیت دریافت شده عملاً مستقل از بیت ارسال شده خواهد بود و توزیع آن برابر با یک توزیع برنولی با احتمال برابر می شود. همچنین بیت های فرستاده شده نیز خود با احتمال برابر یا صفر و یا یک می باشند. از طرفی، برای هر X و یا Y ، دقیقاً یک حالت وجود دارد که $X = x$ و یا $Y = y$ باشد. به عبارت دیگر برای مثال داریم:

$$P\{X = 4\} = P\{b_7 = 0, \dots, b_2 = 1, b_1 = 0, b_0 = 0\} = 2^{-8} \quad (3)$$

و برای بقیه x ها نیز باز احتمال برابر با همین مقدار می شود چرا که فقط یک حالت وجود دارد که این مقدار برای X را داشته باشیم و از طرفی هر بیت نیز متغیر تصادفی مستقلی از بیت دیگر می باشد. پس در نهایت داریم:

$$P\{X = x\} = \begin{cases} 2^{-8}, & 0 \leq x \leq 255 \\ 0, & O.W \end{cases} \quad (4)$$

و

$$P\{Y = y\} = \begin{cases} 2^{-8}, & 0 \leq y \leq 255 \\ 0, & O.W \end{cases} \quad (5)$$

که این دو متغیر تصادفی X و Y کاملاً مستقل از یکدیگر می باشند. حال برای محاسبه احتمال خطا داریم:

$$E = X - Y \implies P\{E = e\} = P\{X - Y = e\} \quad (6)$$

در این جا، حالت هایی که در آن $E = e$ اگر e مثبت باشد برابر اند با $(e, 0), (e + 1, 0), (e + 2, 0), \dots, (255, 255 - e)$ و در حالتی هم که e منفی باشد، حالت های ممکن برابرند با $(0, |e|), (1, |e + 1|), \dots, (255 - |e|, 255)$ که در هر دو نتیجه به صورت کلی تعداد حالات برابر با $|e| + 1$ می باشد. دقت کنید که در هر حالت، X و Y باید هر کدام مقادیری مشخص داشته باشند پس احتمال هر حالت برابر است با $2^{-16} = 2^{-8} \times 2^{-8}$ و در نهایت برای احتمال خطا خواهیم داشت:

$$P\{E = e\} = \begin{cases} \frac{256 - |e|}{2^{16}}, & |e| < 256 \\ 0, & O.W \end{cases} \quad (7)$$

۴. همان‌طور که در قسمت قبل نیز مشاهده کردیم، داریم:

$$\begin{cases} x = 2^7 b_7 + 2^6 b_6 + \dots + b_0 \\ y = 2^7 \hat{b}_7 + 2^6 \hat{b}_6 + \dots + \hat{b}_0 \end{cases} \quad (8)$$

و با استفاده از این معادله برای واریانس داریم:

$$\mathbb{E}((X - Y)^2) = 2^{14} \mathbb{E}((b_7 - \hat{b}_7)^2) + 2^{12} \mathbb{E}((b_6 - \hat{b}_6)^2) + \dots + \mathbb{E}((b_0 - \hat{b}_0)^2) - 2 \sum_{i>j}^7 \mathbb{E}((b_i - \hat{b}_i)(b_j - \hat{b}_j)) \quad (9)$$

به دلیل مستقل بودن \hat{b} و b ها از یکدیگر، کل جمله آخر برابر با صفر می‌شود و فقط ۸ جمله که به صورت واریانس $b_i - \hat{b}_i$ می‌باشند، باقی می‌مانند. پس در ادامه باید این واریانس خطای هر بیت را محاسبه کنیم. از طرفی در حالت حدی که واریانس بسیار بزرگ می‌شود، عملاً بیت خروجی مستقل از ورودی می‌شود و همان‌طور که پیشتر گفتیم، با احتمال برابر یا صفر و یا یک می‌شود. در نتیجه برای محاسبه واریانس خواهیم داشت.

$$\begin{aligned} \mathbb{E}((b - \hat{b})^2) &= \mathbb{E}(b^2) + \mathbb{E}(\hat{b}^2) - 2\mathbb{E}(b)\mathbb{E}(\hat{b}) \\ &= \frac{1}{2}1 + \frac{1}{2}0 + \frac{1}{2}1 + \frac{1}{2}0 - 2\frac{1}{2}\frac{1}{2} \\ &= \frac{1}{2} \end{aligned} \quad (10)$$

و در نتیجه برای واریانس کل خواهیم داشت:

$$\begin{aligned} \mathbb{E}((X - Y)^2) &= \frac{2^{14}}{2} + \frac{2^{12}}{2} + \dots + \frac{1}{2} \\ &= \frac{2^{16} - 1}{2^2 - 1} \frac{1}{2} = 10922.5 \end{aligned} \quad (11)$$

که بر روی نمودار بخش دوم نیز مقداری در همین حدود مشاهده می‌شود.

۵ کدینگ منبع

۱.۵ مشکل کدهای C_1, C_2, C_3

آ) در این کدینگ ایرادی که وجود دارد که برای هر کد، یک حرف یکتا وجود ندارد چرا که برای دو حرف b و d از کد 10 استفاده شده است که همین موضوع باعث ایجاد اشکال در تشخیص می شود و در این صورت در صورتی که چنین کدی دریافت شود، نمی توان تشخیص داد که حرف فرستاده شده b یا d بوده.

ب) در این نوع کدینگ، مشکل کد قبل حل شده است یعنی دیگر به ازای هر کد، یک حرف یکتا وجود دارد اما مشکلی که در این کد وجود دارد این است که در صورت قرار گرفتن چند حرف کنار یکدیگر، می تواند کدهای یکسانی تولید شود که قابلیت تشخیص را از بین ببرد. برای مثال فرض کنید یکبار کلمه ab فرستاده شود و بار دیگر حرف d فرستاده شود. در این صورت هر دو این پیام ها به 010 کد می شوند و به همین دلیل اگر چنین کدی دریافت شود، نمی توان تشخیص داد که از طرف فرستنده، چه کلمه ای فرستاده شده است.

ج) ایراد موجود در این کد این است که به محض دریافت یک کد، نمی توان به صورت آنی حرف مورد نظر را تشخیص داد. بگذارید با مثالی به صورت دقیق تر به این مساله بپردازیم. فرض کنید یکبار می خواهیم حرف a را بفرستیم و بار دیگر می خواهیم حرف b را بفرستیم. در این صورت در صورت دریافت یک صفر درگیرنده، هنوز نمی توان تشخیص داد که حرف مورد نظر a بوده و یا b. به همین دلیل برای تشخیص باید صبر کنیم تا یک بیت دیگر نیز دریافت شود و سپس در صورت صفر بودن آن بیت متوجه می شویم که بیت ارسال شده a بوده است و در غیر این صورت اگر بیت بعدی یک باشد، این یعنی حالا حرف مورد نظر می توان b, c, d, e یا f باشد. به عبارت دیگر در این نوع کدینگ، اگر کلمه کد فرستاده دارای طول n باشد، برای آشکارسازی آن باید n+1 بیت را دریافت کرد که همین موضوع باعث غیر آنی بودن این کدینگ می شود.

۲.۵ حل مساله بهینه سازی برای یافتن طول کلمه کدها برای منبع X

برای حل مساله بهینه سازی ابتدا داریم:

$$\mathcal{L} = \sum_{i=1}^M p_i l_i + \lambda \left(1 - \sum_{i=1}^M 2^{-l_i}\right) \quad (12)$$

حال با گرفتن مشتق جزئی از معادله بالا نسبت به l_i ها و همچنین λ و برابر قرار دادن آن ها با صفر خواهیم داریم:

$$\frac{\partial \mathcal{L}}{\partial l_i} = 0 \quad (13)$$

$$\begin{aligned} \Rightarrow p_i - \lambda \ln\left(\frac{1}{2}\right) 2^{-l_i} &= 0 \\ \Rightarrow 2^{-l_i} &= \frac{p_i}{\lambda \ln(0.5)} \end{aligned} \quad (14)$$

همچنین داریم:

$$\frac{\partial \mathcal{L}}{\partial \lambda} = 0 \quad (15)$$

$$\begin{aligned} \implies 1 - \sum_{i=1}^M 2^{-l_i} &= 0 \\ \stackrel{14}{\implies} 1 &= \sum_{i=1}^M \frac{p_i}{\lambda \ln(0.5)} \\ \implies 1 &= \frac{1}{\lambda \ln(0.5)} \\ \implies \lambda &= \frac{1}{\ln(0.5)} \end{aligned} \quad (16)$$

حال با ترکیب دو معادله (۱۴) و (۱۶) داریم:

$$\begin{aligned} 2^{-l_i} &= p_i \\ \implies l_i &= \log_{\frac{1}{2}}(p_i) \end{aligned} \quad (17)$$

۳.۵ کلمه‌کدهای منبع X

یک پاسخ مناسب برای کلمه‌کدهای منبع X مطابق جدول زیر می‌باشد.

X	a	b	c	d	e	f
$C(X)$	1	01	001	0001	00001	00000

۴.۵ طول متوسط کلمه‌کدها

طول متوسط کلمه‌کدها به شکل زیر به دست می‌آید.

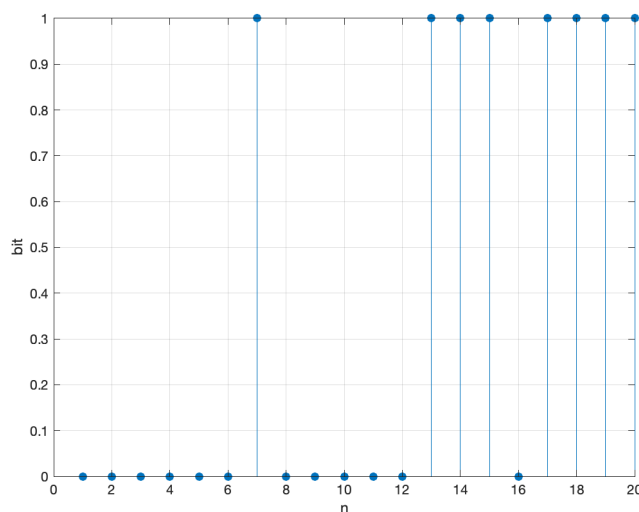
$$\mathbb{E}(l) = \sum_{i=1}^6 p_i l_i = \frac{1}{2} + 2\frac{1}{4} + 3\frac{1}{8} + 4\frac{1}{16} + 5\frac{1}{32} + 5\frac{1}{32} = 1.9375 \quad (18)$$

۵.۵ توابع خواسته شده

کدهای توابع خواسته شده در فایل‌های InformationSource.m، SourceEncoder.m و SourceDecoder.m قرار دارند و تابع InformationSource به احتمال‌های داده شده، سمبل‌های a، b، c، d، e و f را تولید می‌کند.

۶.۵ شبیه‌سازی سیستم و رفتار $H_n(X)$ با بزرگ شدن X

کد این بخش در فایل Part3.m قرار دارد. در ابتدا برای یک دنباله از سمبل‌ها با طول ۱۰، دنباله کد شده مطابق شکل زیر می‌شود.



شکل ۲۷: دنباله کد شد برای یک شبیه‌سازی تصادفی

با محاسبه $H_n(X)$ و رسم آن بر حسب n، نمودار صفحه بعد به دست می‌آید. همچنین لازم به ذکر است که برای n های بزرگ داریم:

$$H_n(X) = \frac{L_B(n)}{n} = \sum_{i=1}^n \frac{l_i}{n} \quad (19)$$

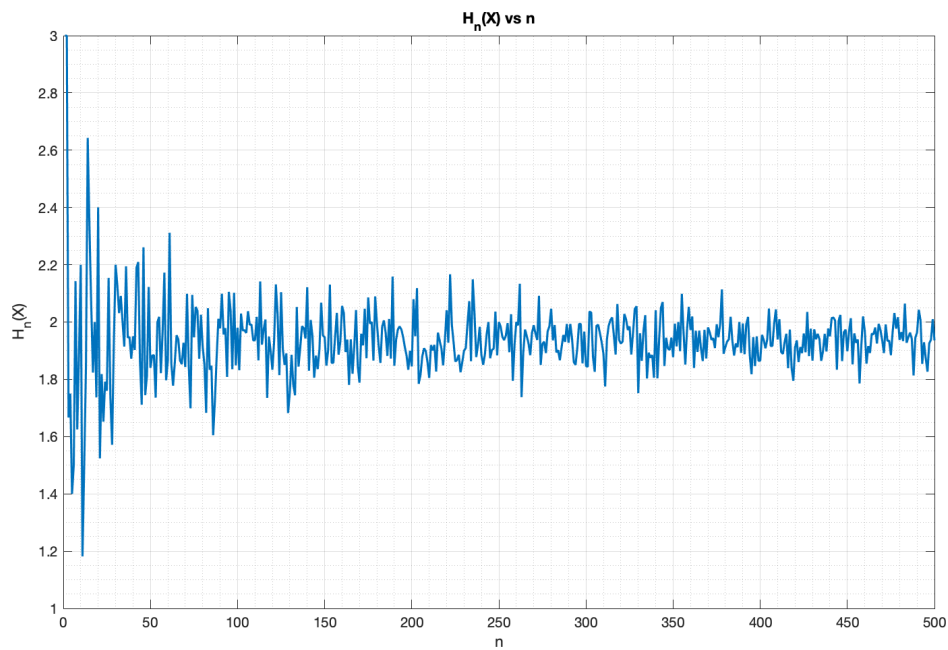
که l_i متغیر تصادفی طول هر کلمه کد می‌باشد که بر اساس چیزی که قبل تر به دست آوردیم، توزیعی مطابق زیر دارد.

$$P\{l\} = \begin{cases} \frac{1}{2}, & l = 1 \\ \frac{1}{4}, & l = 2 \\ \frac{1}{8}, & l = 3 \\ \frac{1}{16}, & l = 4 \\ \frac{1}{16}, & l = 5 \end{cases} \quad (20)$$

حال از آنجا که L_i ها متغیرهای تصادفی مستقل از یکدیگر می‌باشند، با بزرگ تر کردن n طبق قانون اعداد بزرگ خواهیم داشت:

$$\lim_{n \rightarrow \infty} \frac{L_B(n)}{n} = \mathbb{E}(l) = 1.9375 \quad (21)$$

همچنین، نمودار زیر نیز همین موضوع را نشان می‌دهد.



شکل ۲۸: نمودار $H_n(X)$ بر حسب n

۲.۵ کدینگ بی اتلاف

فرض می‌کنیم که یک مرز تصمیم‌گیری λ برای انتخاب کد وجود دارد و داریم:

$$\hat{X} = \begin{cases} \hat{x}_1, & X > \lambda \\ \hat{x}_2, & X < \lambda \end{cases} \quad (22)$$

در این صورت باید λ ، \hat{x}_1 و \hat{x}_2 را به گونه‌ای تعیین کنیم که $D = \mathbb{E}((X - \hat{X})^2)$ کمینه شود.

$$D = \mathbb{E}((X - \hat{X})^2) = \sigma^2 + \mathbb{E}(\hat{X}^2) - 2\mathbb{E}(X\hat{X}) \quad (23)$$

$$\mathbb{E}(\hat{X}^2) = \hat{x}_1^2 \int_{\lambda}^{\infty} f_X(x) dx + \hat{x}_2^2 \int_{-\infty}^{\lambda} f_X(x) dx \quad (24)$$

$$\mathbb{E}(X\hat{X}) = \hat{x}_1 \int_{\lambda}^{\infty} x f_X(x) dx + \hat{x}_2 \int_{-\infty}^{\lambda} x f_X(x) dx \quad (25)$$

$$\begin{aligned} \frac{\partial D}{\partial \lambda} &= -\hat{x}_1^2 f_X(\lambda) + \hat{x}_2^2 f_X(\lambda) + 2\hat{x}_1 \lambda f_X(\lambda) - 2\hat{x}_2 \lambda f_X(\lambda) = 0 \\ \Rightarrow \lambda &= \frac{\hat{x}_1^2 - \hat{x}_2^2}{2\hat{x}_1 - 2\hat{x}_2} = \frac{\hat{x}_1 + \hat{x}_2}{2} \end{aligned} \quad (26)$$

$$\frac{\partial D}{\partial \hat{x}_1} = 0 \Rightarrow \hat{x}_1 = \frac{\int_{\lambda}^{\infty} x f_X(x) dx}{\int_{\lambda}^{\infty} f_X(x) dx} = \frac{\sigma}{\sqrt{2\pi}} \frac{e^{-\frac{\lambda^2}{2\sigma^2}}}{Q(\frac{\lambda}{\sigma})} \quad (27)$$

$$\frac{\partial D}{\partial \hat{x}_2} = 0 \Rightarrow \hat{x}_2 = \frac{\int_{-\infty}^{\lambda} x f_X(x) dx}{\int_{-\infty}^{\lambda} f_X(x) dx} = -\frac{\sigma}{\sqrt{2\pi}} \frac{e^{-\frac{\lambda^2}{2\sigma^2}}}{Q(\frac{-\lambda}{\sigma})} \quad (28)$$

$$\Rightarrow 2\lambda = \frac{\sigma e^{-\frac{\lambda^2}{2\sigma^2}}}{\sqrt{2\pi}} \left(\frac{1}{Q(\frac{\lambda}{\sigma})} - \frac{1}{1 - Q(\frac{\lambda}{\sigma})} \right) \quad (29)$$

پاسخ \circ λ در معادله (۲۹) صادق است، در نتیجه با قرار دادن آن برابر با صفر و به استفاده از معادلات (۲۷) و (۲۸) داریم:

$$\begin{cases} \hat{x}_1 = \sigma \sqrt{\frac{2}{\pi}} \\ \hat{x}_2 = -\sigma \sqrt{\frac{2}{\pi}} \\ \lambda = 0 \end{cases} \quad (30)$$