

پروژه

مهلت تحویل ۱۴۰۴/۱۱/۱۷

مهلت تحویل پروژه تمدید نخواهد شد.

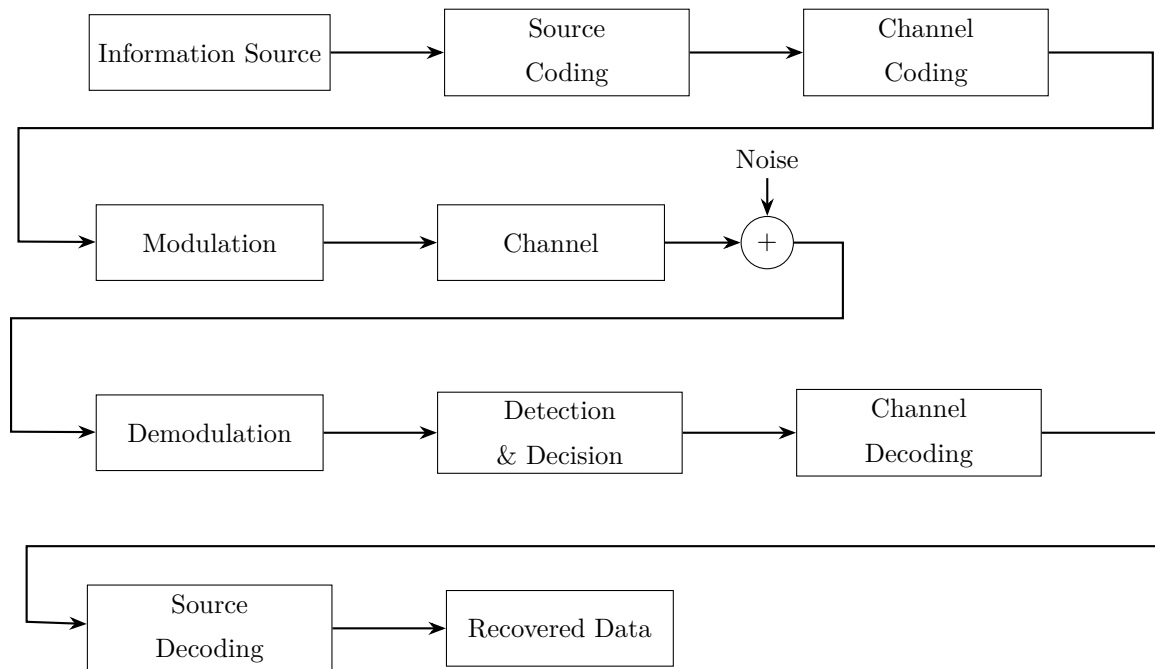
لطفاً به نکات زیر توجه کنید:

۱. این پروژه امتیازی و اختیاری است.
۲. شبیه سازی ها را باید با کمک Python انجام دهید.
۳. گزارش پروژه ی خود را در قالب یک فایل pdf. تحویل دهید. در گزارش لازم است تمامی خروجی ها و نتایج نهایی، پرسش های متن پروژه و توضیح مختصری از فرآیند حل مسئله ی خود در هر قسمت را ذکر کنید.
۴. کد کامل پروژه را در قالب یک فایل ipynb/.pyw/.py. تحویل دهید. لازم است بخش های مختلف پروژه در section های مختلف تفکیک شوند و کد تحویلی منظم و دارای کامنت گذاری مناسب باشد. بدیهی است آپلود کردن کدی که به درستی اجرا نشود، به منزله ی فاقد اعتبار بودن نتایج گزارش شده نیز می باشد.
۵. توابعی را که (در صورت لزوم) نوشته اید، حتما در انتهای کد ضمیمه کنید و از ایجاد فایل جدای py. خودداری کنید.
۶. نام گذاری فایل های تحویلی را به صورت .../.rar/.zip/.py/Project_StudentNumber.pdf انجام دهید.
۷. معیارهای نمره دهی عبارتند از: ساختار مرتب و حرفه ای گزارش، استفاده از توابع و الگوریتم های مناسب، پاسخ به سوالات تئوری و توضیح روش های مطلوب سوال، کد و گزارش خروجی کد برای خواسته های مسأله.
۸. در صورت داشتن هرگونه ابهام با آیدی تلگرام @ahmadara25 یا ایمیل قرار گرفته در سایت درس ارتباط برقرار کنید.
۹. اصولاً مشورت دانشجویان در حل تمرین ها نه تنها نکوهیده نیست، بلکه شدیداً توصیه می شود؛ اما توجه کنید که مشورت و رونویسی متفاوتند! لذا نتایج متفاوتی نیز دارند. همچنین برای مشورت در حل تمرین ها، از گروه درس استفاده کنید تا حق کسی در این روند ضایع نشود.
۱۰. شرافت انسانی ارزشی به مراتب والاتر از تعلقات دنیوی دارد. رونویسی تمارین، زیر پا گذاشتن شرافت خویشتن است؛ به کسانی که شرافتشان را زیر پا می گذارند هیچ نمره ای تعلق نمی گیرد.

۱ سیستم مخابراتی

این پروژه شامل سه بخش می‌باشد. در بخش اول به پیاده سازی یک سیستم مخابراتی دیجیتال از مفاهیم پایه کدینگ منبع، کدینگ کانال، مدولاسیون، شبیه سازی کانال، دی‌مدولاسیون، دی‌کدینگ کانال، دی‌کدینگ منبع، فیلتر تطبیقی، شناسایی ورودی ارسالی و تحلیل نویز می‌پردازیم.

۲ ساختار سیستم



شکل ۱: بلوک دیاگرام کلی سیستم مخابراتی دیجیتال

یک نمودار بلوکی از سیستم ارتباطی در شکل بالا نشان داده شده است. در قسمت های بعد به تشریح و پیاده سازی هر بلوک می‌پردازیم.

۱.۲ منبع اطلاعات

در مرحله ی نخست، یک منبع اطلاعات گسسته با الفبایی مطابق جدول (۱) در نظر گرفته می‌شود. احتمال وقوع هر یک از نمادهای این الفبا به عنوان ورودی سیستم نیز در همان جدول مشخص شده است. همان گونه که در مباحث درس بیان شد، برای ارسال این اطلاعات از طریق یک سیستم مخابراتی دیجیتال، لازم است نمادهای منبع ابتدا به یک دنباله از بیت ها تبدیل شوند تا امکان مدولاسیون و ارسال آن ها از طریق کانال فراهم گردد.

بر اساس نظریه ی شانون، بیشترین نرخ قابل دستیابی برای انتقال اطلاعات از یک منبع، توسط انتروپی آن منبع تعیین می‌شود. به بیان دیگر، یک سیستم کدینگ منبع ایده آل می‌تواند نرخ نزدیک به انتروپی منبع ایجاد کند. شایان ذکر است که نظریه ی شانون روشی عملی برای دستیابی به این نرخ ارائه نمی‌دهد و صرفاً وجود چنین کدینگی را به صورت نظری اثبات می‌کند.

انتروپی منبع برای یک توزیع احتمال مشخص، از رابطه ی زیر محاسبه می‌شود:

$$H(X) = - \sum_i p_i \log_2 p_i,$$

که در این رابطه، $H(X)$ انتروپی منبع (بر حسب بیت بر نماد)، X متغیر تصادفی متناظر با منبع اطلاعات، و p_i احتمال وقوع نماد i از الفبا می‌باشد.

برای مطالعه‌ی بیشتر درباره‌ی نظریه‌ی شانون و مفاهیم مرتبط با انتروپی و نرخ اطلاعات، می‌توانید از [این لینک](#) استفاده کنید.

Symbol	Probability
A	0.25
B	0.20
C	0.15
D	0.10
E	0.08
F	0.07
G	0.05
H	0.04
I	0.04
J	0.02

جدول ۱: سمبل‌های ورودی و احتمال وقوع هر یک از آنها

خواسته‌ها:

۱. به صورت تئوری نرخ آنتروپی منبع را محاسبه کنید.

۲. تابعی با عنوان `entropy_rate` بنویسید که با دریافت جدول الفبای ورودی نرخ آنتروپی را محاسبه کند و با نتیجه تئوری مقایسه کنید.

۲.۲ کدینگ منبع

هدف اصلی کدینگ منبع، نمایش اطلاعات تولیدشده توسط منبع با کمترین تعداد بیت ممکن و بدون از دست رفتن اطلاعات است. در سیستم‌های مخابراتی دیجیتال، این مرحله نقش مهمی در افزایش بهره‌وری پهنای باند ایفا می‌کند، زیرا با کاهش نرخ بیت، امکان ارسال اطلاعات با منابع محدود کانال فراهم می‌شود. همان‌طور که در بخش قبل بیان شد، حد پایین نرخ قابل دستیابی برای نمایش یک منبع اطلاعات، توسط انتروپی آن منبع تعیین می‌شود.

یکی از مهم‌ترین روش‌های عملی برای نزدیک شدن به این حد نظری، استفاده از کدهای با طول متغیر است. در این میان، کد هافمن به عنوان یک روش بهینه برای کدینگ منبع‌های گسسته و بدون حافظه شناخته می‌شود. ایده‌ی اصلی در کد هافمن آن است که نمادهایی با احتمال وقوع بیشتر، با کدهای کوتاه‌تر و نمادهایی با احتمال کمتر، با کدهای بلندتر نمایش داده شوند. این کار باعث کاهش طول متوسط کد و نزدیک شدن آن به انتروپی منبع می‌شود.

کد هافمن با ساخت یک درخت دودویی بر اساس توزیع احتمال نمادها به دست می‌آید. در این درخت، هر برگ متناظر با یک نماد از الفبای منبع بوده و مسیر طی شده از ریشه تا برگ (با تخصیص بیت‌های ۰ و ۱ به شاخه‌ها) کد نهایی آن نماد را تعیین می‌کند. برای اطلاعات بیشتر راجب کد هافمن می‌توانید از [این لینک](#) استفاده کنید.

برای ارزیابی عملکرد کدینگ منبع، معیار مهمی به نام **طول متوسط کد** تعریف می‌شود که از رابطه‌ی زیر به دست می‌آید:

$$\bar{L} = \sum_i p_i l_i,$$

که در آن p_i احتمال وقوع نماد i و l_i طول کد اختصاص یافته به آن نماد است.

خواسته‌ها:

۱. ابتدا الفبای جدول (۱) را با استفاده از یک کد ساده (naive) با طول ثابت ۴ بیت برای هر نماد مدل‌سازی کرده و طول متوسط کد را محاسبه کنند.

۲. سپس با استفاده از الگوریتم هافمن، کد بهینه‌ی منبع را طراحی کرده و طول متوسط کد هافمن را به دست آورید.

۳. در نهایت، طول متوسط هر دو کد را با نرخ انتروپی منبع که در بخش قبل محاسبه شده است، مقایسه کرده و میزان بهینگی کد هافمن را تحلیل کنید.

۴. تابعی با نام Huffman_code بنویسید که جدول الفبا و احتمالات را دریافت کرده و کد هافمن مربوط به الفبا را در خروجی به عنوان یک آرایه به ما بدهد. سپس یک تابع با نام source_coder بنویسید که با دریافت یک رشته از الفبا و آرایه خروجی تابع Huffman_code یک رشته بیت به عنوان خروجی بدهد. (سیستم دیکودر این تابع را هم نوشته و از صحت عملکرد آن اطمینان حاصل کنید).

۵. تابعی با نام generate_data بنویسید که با دریافت طول رشته، رشته ای از حروف با توزیع داده شده در جدول شماره ۱ تولید کند.

۶. رشته تولیدی از تابع بخش ۵ را به تابعی که در بخش ۴ نوشتید به عنوان ورودی بدهید و رشته بیت خروجی را به همراه ورودی تابع پرینت کنید و در گزارش خود بیاورید.

۳.۲ کدینگ کانال

کدینگ کانال یکی از مراحل کلیدی در سیستم‌های مخابراتی دیجیتال است که هدف آن افزایش قابلیت اطمینان انتقال داده‌ها از طریق کانال‌های مخابراتی است. در دنیای واقعی، کانال‌های ارتباطی معمولاً تحت تأثیر نویز، تداخل و اختلالات مختلف قرار دارند که می‌توانند باعث بروز خطا در داده‌های ارسال شده شوند. کدینگ کانال با اضافه کردن بیت‌های اضافی و ساختارمند به داده‌ها، امکان شناسایی و تصحیح خطاها در سمت گیرنده را فراهم می‌کند.

به عبارت دیگر، کدینگ کانال به ما اجازه می‌دهد که بدون افزایش توان سیگنال یا پهنای باند، داده‌های ارسالی را در برابر خطاهای ناشی از نویز مقاوم کنیم. این کدها می‌توانند به دو دسته کلی تقسیم شوند:

- کدهای شناسایی خطا: که فقط امکان شناسایی وقوع خطاها را فراهم می‌کنند، مانند کدهای چک پاریته parity codes.

- کدهای اصلاح خطا: که علاوه بر شناسایی، قابلیت تصحیح خطاها را نیز دارند، مانند کد همبستگی بلوکی و کد کانولوشنال.

یکی از مشهورترین کدهای اصلاح خطا، کد هامینگ (۷, ۴) است. این کد ۴ بیت اطلاعاتی را به ۷ بیت کد تبدیل می‌کند و می‌تواند هر خطای تک‌بیتی را شناسایی و اصلاح کند.

- تعداد بیت‌های اطلاعاتی: ۴

- تعداد بیت‌های کد: ۷

- نرخ کد: $R = \frac{4}{7}$

- قابلیت تصحیح خطا: تک‌بیتی

در کد Hamming (۷, ۴)، سه بیت کنترل یا parity bits به چهار بیت اطلاعات اضافه می‌شود. ترتیب قرارگیری بیت‌ها معمولاً به صورت $(p_1, p_2, d_1, p_3, d_2, d_3, d_4)$ است، که در آن p_i بیت‌های کنترل و d_i بیت‌های داده هستند. گیرنده می‌تواند با بررسی بیت‌های کنترل، موقعیت خطای رخ داده را شناسایی و آن را اصلاح کند.



شکل ۲: نمودار بلوکی ساده Hamming(4,7)

برای مطالعه‌ی بیشتر درباره کد هامینگ و کاربردهای آن، می‌توانید به [این لینک](#) مراجعه کنید.

خواسته‌ها:

۱. با ذکر یک مثال توضیح دهید که کد Hamming(4,7) چگونه تا یک بیت خطا را تصحیح می‌کند.

۲. تابعی با عنوان Hamming_coder بنویسید که با دریافت رشته بیت ورودی بیت‌های کد شده را در خروجی خود قرار دهد.

۳. خروجی تابع بخش ۶ سوال قبل را به عنوان ورودی به تابع Hamming_coder بدهید و خروجی حاصل را به همراه ورودی پرینت کنید.

۴.۲ کانال

کانال یک فیلتر با پاسخ ضربه‌ی مشخص است که محیط انتقال مانند سیم مسی، کابل یا هوا را توصیف می‌کند. کانال به عنوان یک سیستم خطی تغییرناپذیر با زمان (LTI) با پاسخ ضربه $h(t)$ شبیه سازی می‌شود که در این پروژه کانال ایده آل در نظر گرفته می‌شود. دقت کنید که برای نوشتن تابع کانال پارامتری تحت عنوان length به عنوان ورودی به تابع داده می‌شود و شما باید تضعیف کانال با مقدار $att = 1 \frac{dB}{Km}$ را براساس این پارامتر شبیه سازی کنید.

۵.۲ نویز

همانطور که در درس با تاثیر نویز کانال بر سیگنال آشنا شدید در این پروژه هم کانال را به صورت یک کانال AWGN شبیه سازی می‌شود که نویز ورودی نیز باید به صورت متغیر به تابع شما داده شود. بنابراین سیگنال دریافتی به صورت زیر نوشته می‌شود:

$$r(t) = s(t) * h(t) + n(t), \quad (1)$$

که * عملگر کانولوشن، $s(t)$ سیگنال مدوله شده و $n(t)$ نویز افزوده شده است. برای شبیه سازی این نویز یک ماتریس نویز تصادفی گاوسی با دستور `sigma * np.random.randn(.,.)` تولید کنید، که در آن σ ریشه‌ی دوم توان نویز است. آرگومان‌های دستور `randn` بستگی به اندازه‌ی جریان ورودی در پیاده‌سازی شما دارد. این نویز گاوسی را به خروجی کانال اضافه کنید.

۶.۲ فیلتر تطبیق یافته Mathced Filter

گیرنده سعی می‌کند سیگنال ارسال شده را که توسط کانال فیلتر شده و با نویز افزوده خراب شده است، بازیابی کند. یک بلوک در گیرنده که با اثر نویز مقابله می‌کند، فیلتر سازنده پالس دریافتی است که با فیلتر سازنده پالس ارسال شده مطابقت دارد:

$$g_R(t) = g_T(T - t). \quad (2)$$

فیلتر تطبیق یافته بهترین فیلتر گیرنده برای مقابله با نویز سفید است، زیرا در نقطه نمونه برداری بالاترین نسبت سیگنال به نویز (SNR) را ایجاد می‌کند.

۷.۲ مدولاسیون

برای ارسال بیت‌های دیجیتال از طریق یک کانال فیزیکی نیاز داریم که هر بیت را با یک موج نمایش دهیم. در این پروژه قصد داریم چند مورد از مدولاسیون‌های دیجیتال که در درس با آنها آشنا شدید به صورت عملی پیاده سازی کنیم.

۱. تابعی با عنوان PulseShaping بنویسید که در ورودی، دنباله‌ای از صفر و یک و شکل پالس‌های متناظر (بر اساس مدولاسیون در هر بخش شکل پالس‌ها متفاوت هستند) و در خروجی یک سیگنال پیوسته با طول برابر مجموع طول پالس‌ها به ما تحویل دهد.

۲. تابعی با عنوان AnalogMod بنویسید که ورودی آن، شکل موج نمونه‌برداری شده و فرکانس حامل را دریافت کرده و در خروجی سیگنال $x_c(t)$ را بدهد.

۳. تابعی با عنوان AnalogDemod بنویسید که در ورودی سیگنال $x_c(t)$ ، فرکانس نمونه‌برداری، پهنای باند سیگنال و فرکانس حامل را بگیرد و در خروجی دو شکل موج demodulated شده (یعنی $y_I(t)$ و $y_Q(t)$) را تحویل دهد (ترکیب ضرب و فیلتر پایین‌گذر).

برای این قسمت، مشخصات بلوک‌ها را به این صورت در نظر بگیرید:

متغیر	مقدار
فرکانس نمونه‌برداری	۱ MHz
طول هر پالس	۱۰ ms
فرکانس حامل	۱۰ kHz
فرکانس مرکزی کانال	۱۰ kHz
پهنای باند کانال	۱ kHz

جدول ۲: مشخصات بلوک‌های سیستم

در این قسمت می‌خواهیم با استفاده از مدولاسیون FSK سیگنال‌ها را ارسال کنیم. دقت کنید که سیگنال‌ها در این پروژه به صورت نمونه بیت‌های دوتایی (۰، ۱، ۱۰، ۱۱) ارسال می‌شوند. ۴ فرکانس متعامد را با توجه به مشخصات مدولاسیون انتخاب کنید. (الف) دنباله‌ای به طول L (به اندازه‌ی کافی بلند) از الفبای جدول ۱ تولید کرده و با استفاده از توابع source_coder و Hamming_coder به دنباله‌ای از ۰ و ۱ تبدیل کنید.

(ب) با استفاده از توابع pulse_shaping و استفاده از ۴ سیگنال متعامد این رشته بیت را به یک سیگنال آماده برای مدوله شدن توسط Analog_mod تبدیل کنید. حال با فرض اینکه نویز AWGN به سیگنال بعد از عبور از کانال آن جمع می‌شود، احتمال خطا بر حسب واریانس نویز رسم کنید. بازه‌ی محوری را به اندازه‌ی کافی بزرگ بگیرید. به رفتار احتمال خطا توجه کنید. (واریانس نویز را از $\sigma^2 = ۱۵$ تا $\sigma^2 = ۹۰$ با استپ‌های ۱۵ تایی متغیر در نظر بگیرید).

(ج) با توجه به بخش قبل، ۶ مقدار مختلف برای واریانس نویز انتخاب کنید. برای هر کدام از این واریانس‌ها، scatter plot دوبعدی خروجی چهار Matched Filter را رسم کنید. حال اگر مقدار واریانس نویز کم باشد، نقاط $(\hat{b}_I[i], \hat{b}_Q[i])$ به درستی تخمین زده می‌شوند، اما با افزایش واریانس نویز، این نقاط دچار پراکندگی می‌شوند. این نقاط را روی یک نمودار رسم کنید و برای واریانس‌های مختلف، این نمودارها را مقایسه کنید. این مسئله، خواسته‌ی مسئله‌ی سیگنال (به عبارت مفهومی Signal Constellation) را محقق می‌کند.

(د) eye diagram سیگنال دریافتی را به ازای واریانس نویزهای مختلف رسم و مولفه‌های اساسی آن را تحلیل کنید.

در این بخش می‌خواهیم مدولاسیون خود را به QPSK تغییر دهیم. به صورت تئوری بیان کنید که سیگنال‌های مورد استفاده برای هر جفت دوتایی از بیت‌ها را چگونه انتخاب می‌کنید.

(الف) دنباله‌ای به طول L (به اندازه‌ی کافی بلند) از الفبای جدول ۱ تولید کرده و با استفاده از توابع source_coder و Hamming_coder به دنباله‌ای از ۰ و ۱ تبدیل کنید.

(ب) با استفاده از توابع pulse_shaping و استفاده از ۴ سیگنال رشته بیت را به یک سیگنال آماده برای مدوله شدن توسط Analog_mod تبدیل کنید. حال با فرض اینکه نویز AWGN به سیگنال بعد از عبور از کانال آن جمع می‌شود، احتمال خطا را بر حسب واریانس نویز رسم کنید. بازه‌ی محوری را به اندازه‌ی کافی بزرگ بگیرید. به رفتار احتمال خطا توجه کنید. (واریانس نویز را از $\sigma^2 = 15$ تا $\sigma^2 = 90$ با استپ های ۱۵ تایی متغیر در نظر بگیرید)

(ج) با توجه به بخش قبل، ۶ مقدار مختلف برای واریانس نویز انتخاب کنید و Signal constellation را رسم کنید.

(د) eye diagram سیگنال دریافتی را به ازای واریانس نویز های مختلف رسم و مولفه های اساسی آن را تحلیل کنید.

در این بخش میخواهیم مدولاسیون خود را به 16-QAM تغییر دهیم. با توجه به مطالبی که در درس یاد گرفتید دامنه ۱۶ سیگنال مورد نیاز برای انتقال مجموعه های ۴ تایی بیت ها را به صورت تئوری انتخاب کنید.

(الف) دنباله‌ای به طول L (به اندازه‌ی کافی بلند) از الفبای جدول ۱ تولید کرده و با استفاده از توابع source_coder و Hamming_coder به دنباله ای از ۰ و ۱ تبدیل کنید.

(ب) با استفاده از توابع pulse_shaping و استفاده از ۱۶ سیگنال رشته بیت را به یک سیگنال آماده برای مدوله شدن توسط Analog_mod تبدیل کنید. حال با فرض اینکه نویز AWGN به سیگنال بعد از عبور از کانال آن جمع می‌شود، احتمال خطا را بر حسب واریانس نویز رسم کنید. بازه‌ی محوری را به اندازه‌ی کافی بزرگ بگیرید. به رفتار احتمال خطا توجه کنید. (واریانس نویز را از $\sigma^2 = 15$ تا $\sigma^2 = 90$ با استپ های ۱۵ تایی متغیر در نظر بگیرید)

(ج) با توجه به بخش قبل، ۶ مقدار مختلف برای واریانس نویز انتخاب کنید و Signal constellation را رسم کنید.

(د) eye diagram سیگنال دریافتی را به ازای واریانس نویز های مختلف رسم و مولفه های اساسی آن را تحلیل کنید.

۳ پیاده‌سازی پروژه و تحلیل

۱.۳ نکات پیاده‌سازی

برای پیاده‌سازی پروژه، شما باید هر بلوک عملکردی را به‌طور جداگانه آزمایش کنید قبل از اینکه همه‌ی آن‌ها را به یک سیستم متصل کنید. نتیجه خروجی هر بلوک را در پردازش خود به‌طور جداگانه بیاورید. در نهایت همه بلوک ها را به هم متصل کرده و مطابق شکل ۱ یک دنباله ۱۵۰ حرفی را با کمک سیستم خود مخابره کرده و نتیجه نهایی را با داده اولیه مقایسه کنید و به ازای واریانس نویزهای مختلف مقدار BER را گزارش دهید.

۲.۳ تحلیل

پس از اینکه سیستم مخابراتی کامل را ساختید و آزمایش کردید که به درستی کار می‌کند، اکنون ابزاری قدرتمند در اختیار دارید که می‌توانید اثرات عوامل مختلف را بر عملکرد سیستم آزمایش و تحلیل کنید.

۱. با توجه به نتایج خروجی خود کدام یک از مدولاسیون ها در شرایط مشابه نویز عملکرد بهتری از لحاظ BER نمایش می‌دهد؟

۲. eye diagram مربوطه را به ازای واریانس نویز $\sigma^2 = 60$ برای هر سه مدولاسیون مقایسه کرده و تحلیل کنید.

۳. تاثیر پارامتر length را بر مدولاسیون های مختلف ارزیابی کنید. آیا با تغییر نوع مدولاسیون میتوان نتایج بهتری در طول های بلند بدست آورد؟

۴ شبیه‌سازی سیستم PCM صوت با Companding

در این بخش، هدف پیاده‌سازی و شبیه‌سازی کامل فرآیند تبدیل صدای گفتار انسان به یک استریم دیجیتال با نرخ ثابت 64 kbps می‌باشد. این فرآیند با استفاده از روش Pulse Code Modulation (PCM) و با بهره‌گیری از مفاهیم نمونه‌برداری،

کوانتیزاسیون یکنواخت و غیر یکنواخت، و همچنین استفاده از Comander (شامل فشرده سازی و انبساط دامنه سیگنال) انجام می شود.

ورودی این سیستم، صدای ضبط شده ی هر دانشجو بوده و خروجی، سیگنال دیجیتال PCM و نسخه ی بازسازی شده ی آن در گیرنده است. در طول این بخش، اثر Companding بر کاهش خطای کوانتیزاسیون و بهبود کیفیت سیگنال بازسازی شده مورد بررسی و تحلیل قرار می گیرد.

۱.۴ نمونه برداری از سیگنال صوتی

سیگنال گفتار انسان دارای باند فرکانسی محدود (تقریباً از ۳۰۰ Hz تا ۳۴۰۰ Hz) می باشد. براساس قضیه نمونه برداری نایکوئیست، برای جلوگیری از پدیده ی تداخل طیفی، فرکانس نمونه برداری باید حداقل دو برابر بیشترین فرکانس موجود در سیگنال باشد. از این رو، در این پروژه فرکانس نمونه برداری برابر با

$$f_s = 8 \text{ kHz}$$

در نظر گرفته می شود که مطابق استانداردهای مخابراتی سیستم های صوتی (مانند تلفن) است.

خواسته ها:

۱. صدای خود را با استفاده از میکروفون سیستم و با فرکانس نمونه برداری ۸ kHz و مدت حداقل ۲۰ ثانیه ضبط کنید.

۲. شکل موج سیگنال صوتی نمونه برداری شده را در حوزه زمان رسم کنید.

۳. با استفاده از تبدیل فوریه سریع (FFT) طیف فرکانسی سیگنال را رسم کرده و باند غالب فرکانسی آن را تحلیل کنید.

۲.۴ کوانتیزاسیون یکنواخت و PCM

پس از نمونه برداری، سیگنال پیوسته ی صوتی باید به سطوح گسسته ی دامنه نگاشته شود. در این بخش، از کوانتیزاسیون یکنواخت با دقت ۸ بیت برای هر نمونه استفاده می شود. با این انتخاب، نرخ بیت سیستم برابر خواهد بود با:

$$R = f_s \times N = 8000 \times 8 = 64 \text{ kbps}$$

که یک نرخ استاندارد برای سیستم های PCM صوتی محسوب می شود.

در کوانتیزاسیون یکنواخت، بازه ی دینامیکی سیگنال به تعداد ثابتی از سطوح با فاصله ی مساوی تقسیم می شود و هر نمونه به نزدیک ترین سطح نگاشته می شود.

خواسته ها:

۱. تابعی با نام UniformQuantizer بنویسید که سیگنال ورودی و تعداد بیت کوانتیزاسیون را دریافت کرده و سیگنال کوانتیزه شده را در خروجی تحویل دهد.

۲. سیگنال اصلی و سیگنال کوانتیزه شده را روی یک نمودار رسم کرده و آن ها را مقایسه کنید.

۳. خطای کوانتیزاسیون را محاسبه کرده و شکل موج آن را رسم کنید.

۴. نسبت سیگنال به نویز کوانتیزاسیون (SNR) را محاسبه و گزارش کنید.

۳.۴ Companding و کوانتیزاسیون غیر یکنواخت

سیگنال های گفتار دارای ویژگی مهمی هستند که بخش عمده ای از نمونه ها دارای دامنه های کوچک می باشند. در کوانتیزاسیون یکنواخت، این موضوع باعث می شود که دقت کوانتیزاسیون برای دامنه های کوچک کافی نباشد و نویز کوانتیزاسیون محسوس گردد.

برای رفع این مشکل، از روش Companding استفاده می‌شود که شامل دو مرحله‌ی اصلی است:

• **Compressor**: فشرده‌سازی دامنه سیگنال قبل از کوانتیزاسیون

• **Expander**: بازگردانی دامنه سیگنال پس از دی‌کوانتیزاسیون

در این پروژه می‌توان از یکی از قوانین استاندارد μ -law یا A-law برای پیاده‌سازی Companding استفاده کرد. این روش باعث افزایش تفکیک‌پذیری کوانتیزاسیون در دامنه‌های کم‌انرژی سیگنال می‌شود. برای اطلاعات بیشتر از **این لینک** و **این لینک** استفاده کنید خواهسته‌ها:

۱. تابعی با نام Compressor بنویسید که سیگنال صوتی ورودی را بر اساس قانون μ -law یا A-law فشرده‌سازی کند.

۲. سیگنال فشرده‌شده را با استفاده از کوانتیزاسیون یکنواخت ۸ بیتی کوانتیزه کنید.

۳. تابع Expander را پیاده‌سازی کرده و سیگنال بازسازی‌شده را در گیرنده به دست آورید.

۴. سیگنال بازسازی‌شده با Companding را با حالت بدون Companding از نظر خطای کوانتیزاسیون مقایسه کنید.

۵. مقدار SNR را در هر دو حالت محاسبه کرده و نتایج را تحلیل کنید.

۴.۴ تحلیل و مقایسه نتایج

در این بخش، عملکرد سیستم PCM در حالت کوانتیزاسیون یکنواخت و حالت استفاده از Companding مورد مقایسه قرار می‌گیرد. خواهسته‌ها:

۱. توزیع خطای کوانتیزاسیون را برای هر دو حالت یکنواخت و Companded رسم کرده و مقایسه کنید.

۲. تأثیر Companding را بر کاهش نویز کوانتیزاسیون در دامنه‌های کوچک سیگنال تحلیل کنید.

۳. سیگنال بازسازی‌شده در هر دو حالت را از نظر شنیداری (کیفیت صدا) مقایسه و تحلیل کنید.

۴. توضیح دهید چرا استفاده از Companding در سیستم‌های مخابراتی صوتی (مانند تلفن) رایج است.

۵ شبیه‌سازی فرایند RACH با استفاده از سیگنال‌های Zadoff-Chu

در این بخش، هدف آشنایی عملی با فرایند Random Access Channel (RACH) در سیستم‌های مخابرات سلولی و بررسی نقش سیگنال‌های Zadoff-Chu (ZC) در فرایند دسترسی اولیه کاربر (UE) به شبکه می‌باشد. در این بخش، ابتدا تعریف ریاضی این سیگنال‌ها ارائه می‌شود، سپس خواص همبستگی آن‌ها بررسی شده و در نهایت یک شبیه‌سازی ساده از فرایند RACH شامل ارسال pre-amble، شناسایی آن در گیرنده و تکمیل فرایند اتصال انجام می‌شود.

۱.۵ تعریف سیگنال Zadoff-Chu

سیگنال Zadoff-Chu یک دنباله مختلط با طول N است که به صورت زیر تعریف می‌شود:

$$x_u[n] = \begin{cases} \exp\left(-j\frac{\pi un(n+1)}{N}\right) & \text{اگر } N \text{ فرد باشد} \\ \exp\left(-j\frac{\pi un^2}{N}\right) & \text{اگر } N \text{ زوج باشد} \end{cases}$$

که در آن:

• N طول دنباله

• u ریشه‌ی دنباله (root index) که نسبت به N اول باشد

$$n = 0, 1, \dots, N - 1$$

این دنباله‌ها دارای ویژگی خودهمبستگی ضربه‌ای بوده و همبستگی متقاطع بین دنباله‌هایی با ریشه‌های متفاوت بسیار کوچک است. برای اطلاعات بیشتر به [این لینک](#) مراجعه کنید.

۲.۵ همبستگی متقاطع و خودهمبستگی

در درس با مفاهیم خود همبستگی و همبستگی متقاطع آشنا شدید برای دو سیگنال گسسته $x[n]$ و $y[n]$ ، همبستگی متقاطع به صورت زیر تعریف می‌شود:

$$R_{xy}[k] = \sum_n x[n] y^*[n - k].$$

همچنین، خودهمبستگی یک سیگنال برابر است با:

$$R_{xx}[k] = \sum_n x[n] x^*[n - k].$$

خواسته‌ها

۱. تابعی با نام `cross_correlation` بنویسید که دو دنباله‌ی مختلط را دریافت کرده و همبستگی متقاطع آن‌ها را محاسبه کند.

۲. چهار دنباله Zadoff-Chu تولید کنید:

- دو دنباله با پارامترهای کاملاً یکسان
- دو دنباله با ریشه‌های متفاوت

۳. برای هر حالت:

- نمودار خودهمبستگی
- نمودار همبستگی متقاطع

را رسم کرده و نتایج را تحلیل کنید. تفاوت رفتار دنباله‌های مشابه و متفاوت را توضیح دهید.

۴. با توجه به نمودارها توضیح دهید چرا این سیگنال‌ها برای قراینده Random access procedure در شبکه‌های LTE و 5G استفاده می‌شوند.

۳.۵ شناسایی MIB و فرآیند دسترسی تصادفی

در این بخش، یک سناریوی ساده‌شده از فرآیند دسترسی اولیه UE به شبکه شبیه‌سازی می‌شود. فرض کنید:

- یک رشته‌ی بیت تصادفی با طول ۱۰۰۰ بیت در کانال وجود دارد.
- درون این رشته، یک الگوی ثابت (نماینده‌ی MIB) با ۲۴ بیت قرار داده شده است.
- سیگنال با استفاده از مدولاسیون QPSK (که در بخش‌های قبل پیاده‌سازی شده) ارسال می‌شود.
- کانال و نویز همان مشخصات بخش‌های قبلی را دارند.

گیرنده (UE) باید این الگوی MIB را در سیگنال دریافتی شناسایی کند.

خواسته‌ها

۱. یک رشته‌ی بیت تصادفی با طول ۱۰۰۰ تولید کرده و الگوی MIB را در موقعیتی دلخواه از آن قرار دهید.
۲. سیگنال را با استفاده از مدولاسیون QPSK مدوله کرده و از کانال عبور دهید.
۳. با استفاده از همبستگی، موقعیت الگوی MIB را در سیگنال دریافتی شناسایی کنید.
۴. پس از شناسایی موفق، UE MIB به صورت تصادفی یکی از ۶۴ سیگنال Zadoff-Chu موجود را به عنوان pre-amble RACH انتخاب و ارسال کند.
۵. درگیرنده:

- سیگنال ZC دریافتی را شناسایی کنید.
- شماره‌ی دنباله شناسایی شده را به صورت یک عدد ۶ بیتی مدوله نمایید.
- ۶. اگر دنباله‌ی شناسایی شده با دنباله‌ی ارسال شده یکسان بود، پیام

Attach Completed

را نمایش دهید.

۷. در نهایت:

- سیگنال MIB مدوله شده
 - سیگنال Zadoff-Chu ارسالی
 - خروجی همبستگی درگیرنده
 - سیگنال ۶ بیتی مدوله شده
- را رسم کرده و کل فرآیند RACH را به صورت شماتیک تحلیل کنید.