

## پروژه

۱۴۰۴/۱۱/۱۷ مهلت تحويل

مهلت تحويل پروژه تمدید نخواهد شد.

### لطفاً به نکات زیر توجه کنید:

۱. این پروژه امتیازی و اختیاری است.

۲. شبیه‌سازی‌ها را باید با کمک Python انجام دهید.

۳. گزارش پروژه‌ی خود را در قالب یک فایل pdf. تحويل دهید. در گزارش لازم است تمامی خروجی‌ها و نتایج نهایی، پرسش‌هایی متن پروژه و توضیح مختصری از فرآیند حل مسئله‌ی خود در هر قسمت را ذکر کنید.

۴. کد کامل پروژه را در قالب یک فایل .ipynb/.pyw/.py. تحويل دهید. لازم است بخش‌های مختلف پروژه در section‌های مختلف تفکیک شوند و کد تحولی منظم و دارای کامنت‌گذاری مناسب باشد. بدیهی است آپلود کردن کدی که به درستی اجرا نشود، به منزله‌ی فاقد اعتبار بودن نتایج گزارش شده نیز می‌باشد.

۵. توابعی را که (در صورت لزوم) نوشته‌اید، حتماً در انتهای کد ضمیمه کنید و از ایجاد فایل جدای py. خودداری کنید.

۶. نام گذاری فایل‌های تحولی را به صورت .../Project\_StudentNumber.pdf/.py/.zip/.rar/.rar انجام دهید.

۷. معیارهای نمره‌دهی عبارتند از: ساختار مرتب و حرفة‌ای گزارش، استفاده از توابع و الگوریتم‌های مناسب، پاسخ به سوالات تئوری و توضیح روش‌های مطلوب سوال، کد و گزارش خروجی کد برای خواسته‌های مسئله.

۸. در صورت داشتن هرگونه ابهام با آیدی تلگرام @ahmadara25 یا ایمیل قرار گرفته در سایت درس ارتباط برقرار کنید.

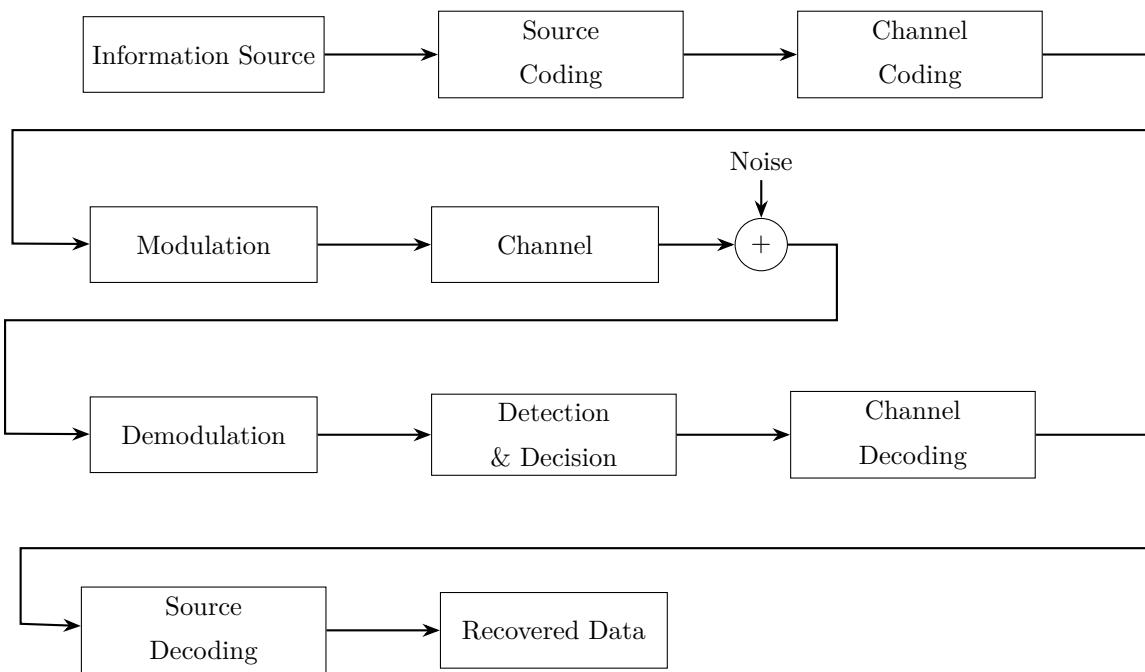
۹. اصولاً مشورت دانشجویان در حل تمرین‌ها نه تنها نکوهیده نیست، بلکه شدیداً توصیه می‌شود؛ اما توجه کنید که مشورت و رونویسی متفاوتند! لذا نتایج متفاوتی نیز دارند. همچنین برای مشورت در حل تمرین‌ها، از گروه درس استفاده کنید تا حق کسی در این روند ضایع نشود.

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 می‌گذارند هیچ نمره‌ای تعلق نمی‌گیرد.

## ۱ سیستم مخابراتی

این پژوهش شامل سه بخش اول به پیاده سازی یک سیستم مخابراتی دیجیتال از مفاهیم پایه کدینگ منبع، کدینگ کanal، مدولاسیون، شبیه سازی کanal، دیکدینگ منبع، فیلتر تطبیقی، شناسایی ورودی ارسالی و تحلیل نویز می پردازم.

## ۲ ساختار سیستم



شکل ۱: بلوک‌دیاگرام کلی سیستم مخابراتی دیجیتال

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

### ۱.۲ منبع اطلاعات

در مرحله‌ی نخست، یک منبع اطلاعات گسسته با الفبایی مطابق جدول (۱) در نظر گرفته می‌شود. احتمال وقوع هریک از نمادهای این الفبا به عنوان ورودی سیستم نیز در همان جدول مشخص شده است. همان‌گونه 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 بیت‌ها تبدیل شوند تا امکان مدولاسیون و ارسال آن‌ها از طریق کanal فراهم گردد.

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

انتروپی منبع برای یک توزع احتمال مشخص، از رابطه‌ی 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

خواسته‌های:

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 پهنانی 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.

خواسته‌های:

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، رشته‌ای 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 نوشته‌ید 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 خطای ارسال شده شوند. کدینگ کانال با اضافه کردن بیت‌های اضافی و ساختارمند به داده‌ها، امکان شناسایی و تصحیح خطاهای 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 خطای اضافی : که فقط امکان شناسایی وقوع خطاهای را فراهم می‌کنند، مانند کدهای چک پاریتی parity codes.

• کدهای اصلاح خطای اضافی : که علاوه بر شناسایی، قابلیت تصحیح خطاهای را نیز دارند، مانند کد همنهشتی بلوکی و کد کانولوشنال.

یکی از مشهورترین کدهای اصلاح خطای اضافی 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.

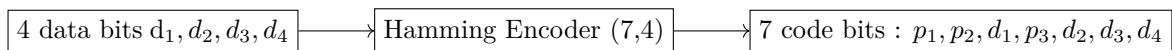
• تعداد بیت‌های اطلاعاتی: ۴

• تعداد بیت‌های کد: ۷

• نرخ کد:  $R = \frac{4}{7}$

• قابلیت تصحیح خطای اضافی : 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 کanal

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

## ۵.۲ نویز

همانطور که در درس با تاثیر نویز کanal بر سیگنال آشنا شدید در این پروژه هم کanal را به صورت یک کanal 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  $s(t)$  سیگنال مدوله شده و  $n(t)$  نویز افزوده شده است. برای شبیه سازی این نویزیک ماتریس نویز تصادفی گاوی با دستور `(.., np.random.randn(.., ..))` تولید کنید، که در آن  $\sigma$  ریشه‌ی دوم توان نویز است. آرگومان‌های دستور randn بستگی به اندازه‌ی جریان ورودی در پیاده‌سازی شما دارد. این نویز گاوی را به خروجی کanal اضافه کنید.

## ۶.۲ فیلتر تطبیق‌یافته Mathced Filter

گیرنده سعی می‌کند سیگنال ارسال شده را که توسط کanal 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 تطبیق‌یافته بهترین فیلتر گیرنده برای مقابله با نویز سفید است، زیرا در نقطه نمونه‌برداری بالاترین نسبت سیگنال به نویز (SNR) را ایجاد می‌کند.

## ۷.۲ مدولاسیون

برای ارسال بیت‌های دیجیتال از طریق یک کanal 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 پالس های متناظر (بر اساس مدولاسیون در هر بخش شکل پالس ها متفاوت هستند) و در خروجی یک سیگنال پیوسته با طول برابر مجموع طول پالس ها به ما تحويل 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  $(t)x_c$ ، 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 تحويل 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 (۱۱، ۱۰، ۰۱، ۰۰) 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 خطأ را بر حسب واریانس نویز رسم کنید. بازه‌ی محوری را به اندازه‌ی کافی بزرگ بگیرید. به رفتار احتمال خطأ توجه کنید. (واریانس نویز را از  $\sigma^2 = 90$  با استپ های ۱۵ تایی متغیر در نظر بگیرید.)

(ج) با توجه به بخش قبل، ۶ مقدار مختلف برای واریانس نویز انتخاب کنید. برای هر کدام از این واریانس‌ها، scatter plot دو بعدی 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 خطرا بر حسب واریانس نویز رسم کنید. بازه‌ی محوری را به اندازه‌ی کافی بزرگ بگیرید. به رفتار احتمال خطرا توجه کنید.

$$(\text{واریانس نویز را از } 15 = \sigma^2 \text{ تا } 90 = \sigma^2 \text{ با استپ های ۱۵ تایی متغیر در نظر بگیرید})$$

(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 ۱۶-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 دنباله‌ای 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 خطرا بر حسب واریانس نویز رسم کنید. بازه‌ی محوری را به اندازه‌ی کافی بزرگ بگیرید. به رفتار احتمال خطرا توجه کنید.

$$(\text{واریانس نویز را از } 15 = \sigma^2 \text{ تا } 90 = \sigma^2 \text{ با استپ های ۱۵ تایی متغیر در نظر بگیرید})$$

(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 به طور جداگانه آزمایش کنید قبل از اینکه همه‌ی آنها را به یک سیستم متصل کنید. نتیجه خروجی هر بلوک را در پردازش خود به طور جداگانه بیاورید. در نهایت همه بلوک‌ها را به هم متصل کرده و مطابق شکل ۱ یک دنباله ۱۵۰ حرکی 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 مدولاسیون‌ها در شرایط مشابه نویز عملکرد بهتری از لحاظ BER نمایش می‌دهد؟

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

۳. تاثیر پارامتر length را بر مدولاسیون‌های 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 فشرده‌سازی و انبساط دامنه سیگنال) انجام می‌شود.

ورودی این سیستم، صدای ضبط شده‌ی هر دانشجو بوده و خروجی، سیگنال دیجیتال PCM و نسخه‌ی بازسازی شده‌ی 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 می‌گیرد.

#### ۱۴ نمونه‌برداری از سیگنال صوتی

سیگنال گفتار انسان دارای باند فرکانسی محدود (تقریباً از ۳۰۰ Hz تا ۳۴۰۰ Hz) می‌باشد. براساس قضیه نمونه‌برداری نایکوئیست، برای جلوگیری از پدیده‌ی تداخل طیفی، فرکانس نمونه‌برداری باید حداقل دو برابر بیشترین فرکانس موجود در سیگنال باشد. از این‌رو، در این پروژه فرکانس نمونه‌برداری برابر با

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

در نظر گرفته می‌شود که مطابق استانداردهای مخابراتی سیستم‌های صوتی (مانند تلفن) است.  
خواسته‌ها:

۱. صدای خود را با استفاده از میکروفون سیستم و با فرکانس نمونه‌برداری ۸ kHz و مدت حداقل ۲۰ ثانیه ضبط کنید.
۲. شکل موج سیگنال صوتی نمونه‌برداری 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.

#### ۲۴ کوانتیزاسیون یکنواخت و PCM

پس از نمونه‌برداری، سیگنال پیوسته‌ی صوتی باید به سطوح گسسته‌ی دامنه نگاشته شود. در این بخش، از کوانتیزاسیون یکنواخت با دقت ۸ بیت برای هر نمونه استفاده می‌شود. 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 سیستم‌های PCM صوتی محسوب می‌شود.

در کوانتیزاسیون یکنواخت، بارهای دینامیکی سیگنال به تعداد ثابتی از سطوح با فاصله‌ی مساوی تقسیم می‌شود و هر نمونه به نزدیک‌ترین سطح نگاشته می‌شود.

خواسته‌ها:

۱. تابعی با نام UniformQuantizer بنویسید که سیگنال ورودی و تعداد بیت کوانتیزاسیون را دریافت کرده و سیگنال کوانتیزه شده را در خروجی تحويل دهد.
۲. سیگنال اصلی و سیگنال کوانتیزه شده را روی یک نمودار رسم کرده و آن‌ها 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.

#### ۳۴ Companding و کوانتیزاسیون غیر یکنواخت

سیگنال‌های گفتار دارای ویژگی مهمی هستند که بخش عمده‌ای از نمونه‌ها دارای دامنه‌های کوچک می‌باشند. در کوانتیزاسیون یکنواخت، این موضوع باعث می‌شود که دقت کوانتیزاسیون برای دامنه‌های 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 پژوهه می‌توان از یکی از قوانین استاندارد  $\mu$ -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 خواسته‌ها:

۱. تابعی با نام Compressor بنویسید که سیگنال صوتی ورودی را بر اساس قانون  $\mu$ -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. خواسته‌ها:

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 فرایнд 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:

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

که در آن:

• طول دنباله  $N$

•  $u$  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 گستته  $x[n]$  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 مختلف 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 قرایند Random access procedure در شبکه‌های LTE و 5G استفاده می‌شوند.

## ۳.۵ شناسایی MIB و فرآیند دسترسی تصادفی

در این بخش، یک ساریوی 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  $1000$  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  $64$  سیگنال Zadoff–Chu موجود را به عنوان pre-amble انتخاب و ارسال کند.
۵. در گیرنده:
  - سیگنال ZC دریافتی را شناسایی کنید.
  - شماره‌ی دنباله شناسایی شده را به صورت یک عدد  $6$  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 در گیرنده
  - سیگنال  $6$  بیتی مدوله شده
- را رسم کرده و کل فرآیند RACH را به صورت شماتیک تحلیل کنید.