



به نام خدا

دانشگاه صنعتی شریف

دانشکده‌ی مهندسی برق

سیستم‌های مخابراتی - گروه دکتر پاک‌روان

نیم‌سال اول ۱۴۰۳-۰۴

پروژه‌ی درس سیستم‌های مخابراتی

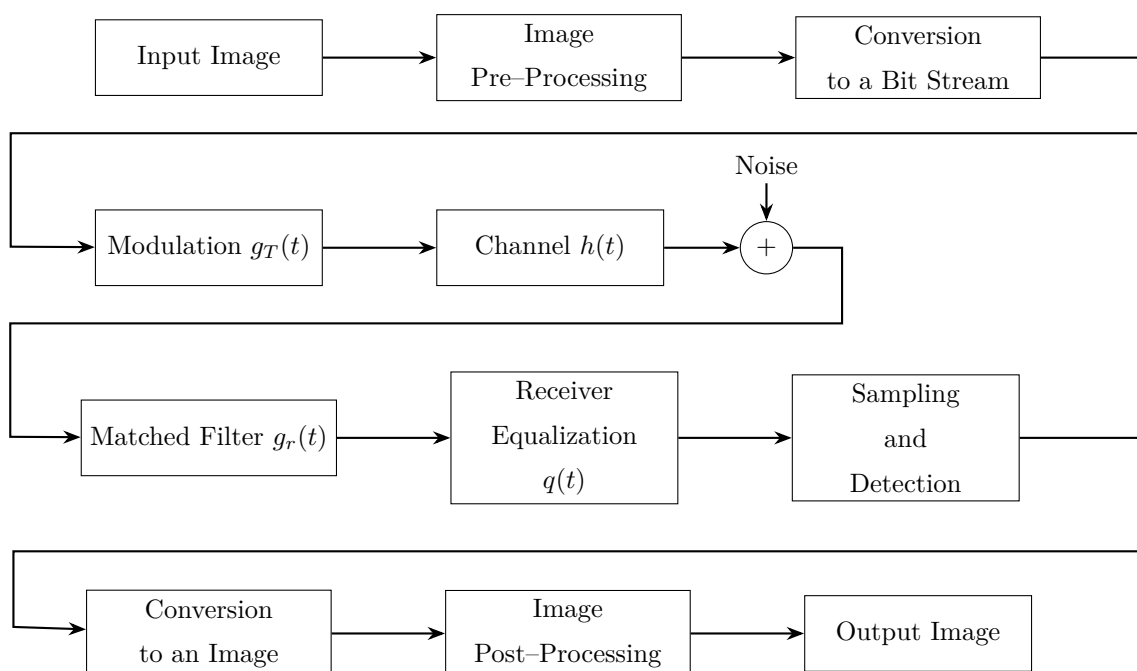
لطفاً به نکات زیر توجه کنید:

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. تحویل دهید. در گزارش لازم است تمامی خروجی‌ها و نتایج نهایی، پرسش‌های 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. تحویل دهید. لازم است بخش‌های مختلف پروژه در section‌های مختلف تفکیک شوند و کد تحویلی 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 تحویلی را به صورت Project_StudentNumber.pdf/.py/.zip/.rar/... انجام دهید.
 ۷. معیارهای نمره‌دهی عبارتند از: ساختار مرتب و حرفه‌ای گزارش، استفاده از توابع و الگوریتم‌های مناسب، پاسخ به سوالات تئوری و توضیح روش‌های مطلوب سوال، کد و گزارش خروجی کد برای خواسته‌های مسأله.
 ۸. در صورت داشتن هرگونه ابهام با آیدی تلگرام [@amir_naghdi_2003](https://t.me/amir_naghdi_2003) 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، زیرا گذاشتن شرافت خویش است؛ به کسانی که شرافتشان را زیر پا می‌گذارند هیچ نمره‌ای تعلق نمی‌گیرد.
 ۱۱. مهلت تحویل: ۱۲ بهمن ۱۴۰۳
- زمان تحویل پروژه تمدید نخواهد شد.

۱ سیستم مخابراتی

در این پروژه قصد داریم مدل یک سیستم مخابراتی مربوط به ارسال عکس را مورد بررسی قرار دهیم که در آن هدف ارسال و دریافت یک عکس بین گیرنده و فرستنده است. برای ارسال یک عکس ما آن را به شکل بلوک‌های 8×8 پیکسل در ۸ پیکسل تبدیل می‌کنیم و سپس تبدیل کسینوسی گسسته^۱ روی هر بلوک اعمال می‌کنیم. اطلاعات فرکانسی DCT هر بلوک می‌تواند کدگذاری و مدوله شده سپس ارسال شود.

۲ ساختار سیستم



شکل ۱: ساختار سیستم

یک نمودار بلوکی از سیستم ارتباطی در شکل بالا نشان داده شده است. این سیستم شامل بلوک‌های زیر است:

۱.۲ پیش‌پردازش تصویر

تصویر به بلوک‌های 8×8 پیکسلی تقسیم و DCT بر روی هر بلوک انجام می‌شود. سپس ضرایب DCT به ۲۵۶ سطح کوانتیزه می‌شوند.

۲.۲ تبدیل به جریان بیت

برای ارسال سریال تصویر کوانتیزه‌شده بر روی کانال، باید تصویر به جریان بیت تبدیل شود. برای این کار، بلوک‌های DCT کوانتیزه شده را به گروه‌هایی با اندازه‌ی N دسته‌بندی می‌کنیم.

۳.۲ مدولاسیون

برای ارسال بیت‌های دیجیتال از طریق یک کانال فیزیکی، نیاز داریم که هر بیت را با یک موج نمایش دهیم، فرآیندی که به آن مدولاسیون گفته می‌شود. در این پروژه از مدولاسیون دامنه PAM با فیلتر سازنده پالس $g_T(t)$ استفاده خواهیم کرد. ما دو فیلتر سازنده پالس مختلف را مطالعه خواهیم کرد تا تأثیر سازنده پالس بر پهنای باند انتقال و عملکرد خطا را مقایسه کنیم. به طور خاص، ما بین یک پالس نیم‌سینوسی و یک پالس SRRC^۲ مقایسه خواهیم کرد که جزئیات آن در بخش پیاده‌سازی ارائه خواهد شد.

^۱Discrete Cosine Transform (DCT)

^۲Square Root Raised Cosine

۴.۲ کانال

کانال یک فیلتر با پاسخ ضربه‌ی مشخص است که محیط انتقال مانند سیم مسی، کابل یا هوا را توصیف می‌کند. کانال به‌عنوان یک سیستم خطی تغییرناپذیر با زمان (LTI) با پاسخ ضربه‌ی $h(t)$ مدل‌سازی خواهد شد.

۵.۲ نویز

در گیرنده، نویز (مانند نویز حرارتی مدار یا تداخل) به سیگنال دریافتی اضافه می‌شود. 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. معمولاً نویز به صورت نویز سفید در نظر گرفته می‌شود.

۶.۲ فیلتر تطبیق‌یافته Mathced Filter

گیرنده سعی می‌کند سیگنال ارسال شده که توسط کانال فیلتر شده و با نویز افزوده خراب شده است را 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. شکل کلی پالس ترکیبی از فیلترهای سازنده پالس فرستنده و گیرنده است. به همین دلیل از فیلتر SRRC در فرستنده استفاده می‌شود زیرا پس از ترکیب با فیلتر گیرنده، شکل کلی پالس به SRRC تبدیل می‌شود.

۷.۲ جبران‌سازی Equalization

جبران‌سازی یک فیلتر خطی است که اثرات کانال را جبران می‌کند. ما دو نوع جبران‌سازی را بررسی خواهیم کرد:

• فیلتر Zero Forcing

• فیلتر حداقل خطای میانگین مربعات (MMSE)

۸.۲ نمونه‌برداری و آشکارسازی

سیگنال خروجی جبران‌ساز در نقاط بهینه نمونه‌برداری می‌شود. در اینجا لازم است همگام‌سازی ساده‌ای انجام شود تا مشخص شود اولین سنبل دریافت شده چه زمانی آغاز می‌شود و زمان بهینه برای نمونه‌برداری چه موقع است. پس از نمونه‌برداری، سیگنال دریافت شده شناسایی می‌شود، به این معنی که بررسی می‌شود آیا فرستنده در ابتدا عدد ۰ یا ۱ ارسال کرده است. در اینجا از یک روش شناسایی بهینه ساده برای بازیابی جریان بیت‌های ارسال شده استفاده خواهیم کرد.

۹.۲ تبدیل به تصویر

این بخش، جریان بیت‌های شناسایی شده را به ترتیب اصلی خود به شکل یک تصویر بازسازی می‌کند. این مرحله، عکس فرآیند تبدیل به جریان بیت است.

۱۰.۲ پردازش پس از تبدیل تصویر

تصویر دمدوله شده به گروه‌های تبدیل کسینوسی گسسته (DCT) با ابعاد ۸ در ۸ تبدیل می‌شود و سپس به حوزه تصویر بازگردانده می‌شود.

۳ پیاده‌سازی بلوکی

۱.۳ پیش‌پردازش تصویر

ابتدا یک تصویر برای استفاده پیدا کنید، پیشنهاد می‌کنیم تصویر خودتان را انتخاب کنید. برای بارگذاری تصویر از دستور `cv2.imread()` استفاده کنید. باید تصویر را با استفاده از دستورات مناسب^۳ به نوع داده `double` تبدیل کنید. فرض می‌کنیم تصویر دارای ابعاد $m \times n$ پیکسل باشد (اطمینان حاصل کنید که m و n بر ۸ قابل تقسیم هستند). تبدیل DCT را روی بلوک‌های 8×8 با استفاده از تابع `skimage.util.view_as_blocks` انجام دهید. برای آماده‌سازی تصویر تبدیل‌شده جهت کوانتیزه‌سازی، مقادیر را به صورت خطی مقیاس‌بندی کنید به طوری که بزرگ‌ترین مقدار ۱ و کوچک‌ترین مقدار ۰ باشد. ثابت‌های مقیاس‌بندی را برای استفاده در مرحله پس‌پردازش ذخیره کنید. سپس تصویر مقیاس‌بندی‌شده را کوانتیزه‌سازی کنید. در نهایت با استفاده از `reshape` و `permute` تصویر مقیاس‌بندی‌شده را به آرایه‌ای سه‌بعدی با ابعاد $8 \times 8 \times \frac{m \times n}{64}$ تبدیل کنید، به طوری که در هر نقطه از بعد سوم، دو بعد اول یکی از بلوک‌های DCT 8×8 باشد.

۲.۳ تبدیل به جریان بیت

بلوک‌های DCT را به صورت گروه‌هایی با اندازه‌ی N ارسال خواهیم کرد. کد خود را به گونه‌ای بنویسید که N یک پارامتر قابل تغییر باشد. با توجه به گروه فعلی شامل N بلوک، آن را به یک بردار ستونی بلند تبدیل کنید. هر عنصر این بردار را به یک سطر تبدیل کنید که شامل ۸ بیت آن عنصر به عنوان ۸ ورودی سطر باشد.

۳.۳ مدولاسیون

ما مدولاسیون را با استفاده از دو تابع سازنده‌ی پالس مختلف پیاده‌سازی خواهیم کرد و عملکرد آن‌ها را از نظر پهنای باند و احتمال خطا مقایسه می‌کنیم. اولین تابع سازنده پالس یک موج سینوسی با نیم‌دوره است که در آن بیت‌های داده روی یک نیم‌دوره از موج سینوسی مدوله می‌شوند. دومین تابع سازنده‌ی پالس، SRRC خواهد بود.

فرض کنید تک بیت b دارای مدت زمان T باشد. برای تابع سازنده‌ی پالس سینوسی با نیم‌دوره، سیگنال ارسال‌شده به صورت زیر است:

$$g_1(t) = \begin{cases} +\sin\left(\frac{\pi}{T}t\right) & \text{if } b = 1 \\ -\sin\left(\frac{\pi}{T}t\right) & \text{if } b = 0 \end{cases} \quad \text{برای } 0 \leq t \leq T. \quad (3)$$

برای تابع سازنده‌ی پالس SRRC، سیگنال ارسال‌شده به صورت زیر است:

$$g_2(t) = \begin{cases} +A \cdot x(t) & \text{if } b = 1 \\ -A \cdot x(t) & \text{if } b = 0 \end{cases} \quad \text{برای } -KT \leq t \leq KT, \quad (4)$$

که در آن A عامل بهنجارسازی است که باعث می‌شود انرژی در پالس SRRC برابر با انرژی در پالس سینوسی با نیم‌دوره باشد و K طول برش است که بعداً مورد بحث قرار خواهد گرفت. پالس SRRC طیف زیر را دارد:

$$X(f) = \begin{cases} \sqrt{T}, & 0 \leq |f| \leq \frac{1-\beta}{4T} \\ \sqrt{T} \cos\left(\frac{\pi T}{4\beta} \left(|f| - \frac{1-\beta}{4T}\right)\right), & \frac{1-\beta}{4T} \leq |f| \leq \frac{1+\beta}{4T} \\ 0, & |f| > \frac{1+\beta}{4T} \end{cases} \quad (5)$$

که در آن β عامل افت است.

^۳ در نهایت خروجی شما مشابه دستور `im2double` در MATLAB باشد

پاسخ ضربه‌ی پالس SRRC به صورت زیر است:

$$x(t) = \frac{\sin\left(\frac{\pi t}{T}(1-\beta)\right) + 4\beta\frac{t}{T}\cos\left(\frac{\pi t}{T}(1+\beta)\right)}{\frac{\pi t}{T}\left(1 - \left(4\beta\frac{t}{T}\right)^2\right)} \quad (6)$$

در نقاط خاصی که صورت کسر صفر می‌شود، مقادیر پاسخ ضربه‌ی SRRC به صورت زیر محاسبه می‌شود:

$$x(t) = \begin{cases} 1 - \beta + 4\frac{\beta}{\pi} & \text{برای } t = 0 \\ \frac{\beta}{\sqrt{1-\beta^2}}\left[\left(1 + \frac{2}{\pi}\right)\sin\left(\frac{\pi}{4\beta}\right) + \left(1 - \frac{2}{\pi}\right)\cos\left(\frac{\pi}{4\beta}\right)\right] & \text{برای } t = \pm\frac{T}{4\beta} \end{cases} \quad (7)$$

از نظر فنی، پالس SRRC به مدت نامحدودی امتداد دارد. برای پیاده‌سازی عملی، باید آن را در بازه $-KT \leq t \leq KT$ برش دهیم که معمولاً $K = 6$ انتخاب می‌شود. در این پروژه، شما باید K را به عنوان یک پارامتر تعریف کنید تا بتوانید آن را بین $2 \leq K \leq 6$ تغییر دهید. به این ترتیب، می‌توانید تأثیر طول برش K را بر عملکرد سیستم مشاهده کنید.

اینکه پالس SRRC در مضارب صحیح مدت زمان بیت صفر نمی‌شود، به این معنی است که در فرستنده حتی قبل از اینکه سیگنال از کانال عبور کند، امواج مدوله‌شده برای بیت‌های مختلف با هم هم‌پوشانی دارند. توجه داشته باشید که پالس SRRC معیار نایکویست را رعایت نمی‌کند زیرا در مضارب صحیح nT صفر نمی‌شود. با این حال، اگر در دریافت‌کننده از فیلتر متناسب استفاده کنیم و در زمان‌های صحیح نمونه‌برداری کنیم (همانطور که در این پروژه خواهیم کرد)، خروجی فیلتر مناسب، طیف کسینوسی بالا رونده‌ای خواهد داشت که معیار نایکویست را رعایت می‌کند و بنابراین سیگنال بدون تداخل بین سمبل‌ها بازبازی می‌شود.

برای تابع سازنده‌ی پالس سینوسی با نیم‌دوره، موج مدوله‌شده $\sin(\pi t)$ برای $0 \leq t \leq 1$ تعریف می‌شود (ما فرض می‌کنیم که مدت زمان بیت $T = 1$ است). پیشنهاد می‌شود که از ۳۲ نمونه استفاده کنید. برای پالس SRRC، $\beta = 0.5$ را انتخاب کنید (آن را به عنوان یک پارامتر تعریف کنید تا بعداً قابل تغییر باشد). هر بیت مدت زمان $2K \times T$ دارد. دو بیت متوالی از سیگنال مدوله‌شده به اندازه $(2K - 1) \times T$ هم‌پوشانی خواهند داشت. سیگنال ارسال‌شده سپس مجموع تمامی پالس‌های SRRC با در نظر گرفتن هم‌پوشانی است. برای هر مدت زمان بیت، پیشنهاد می‌شود که همان تعداد نمونه (۳۲ نمونه) را که برای سینوسی با نیم‌دوره استفاده شده است، استفاده کنید. بنابراین، تعداد کل نمونه‌ها برای هر پالس SRRC، $2K \times 32$ خواهد بود.

Eye Diagram با هم‌پوشانی سیگنال‌های بیت متوالی به دست می‌آید. ناحیه‌ی میانی نمودار Eye Diagram باید باز باشد. عرض عمودی این باز شدن، حاشیه ولتاژ است و عرض افقی آن، حاشیه زمانی است. این‌ها نقش‌های مهمی در نمونه‌برداری و بازسازی سیگنال دارند.

پرسش ۱: تابع شکل هر دو پالس و پاسخ فرکانسی هر کدام را رسم کنید. کدام پالس پهنای باند بیشتری استفاده می‌کند؟ این نتایج چه معنایی دارند؟ اگر طول برش پالس SRRC را افزایش دهید، چه چیزی انتظار دارید که مشاهده کنید؟
نکته: هنگام رسم این پالس‌ها، مطمئن شوید که از همان تعداد نمونه برای هر مدت زمان بیت استفاده کنید، بنابراین پالس SRRC تعداد نمونه‌های بیشتری نسبت به پالس سینوسی با نیم‌دوره خواهد داشت (پالس SRRC، $2K$ برابر تعداد نمونه‌های بیشتری دارد). برای طیف، باید هم نمودارهای دامنه و هم فاز را نشان دهید. دامنه را در دسی‌بل (dB) به جای مقیاس خطی رسم کنید.

پرسش ۲: طیف سیگنال مدوله‌شده (در خروجی فیلتر سازنده‌ی پالس) را رسم کرده و با طیف پالس برای هر پالس مقایسه کنید.
پرسش ۳: برای هر تابع سازنده‌ی پالس، سیگنال مدوله‌شده را در طول زمان برای 10 بیت تصادفی رسم کنید. در مورد آنچه مشاهده می‌کنید، توضیح دهید. پرسش ۴: Eye Diagram سیگنال ارسال‌شده را برای هر دو پالس رسم کنید. آیا باز شدن چشم را می‌بینید؟ در مورد آنچه مشاهده می‌کنید، توضیح دهید.

۴.۳ کانال

از آن جایی که سیگنال را در زمان گسسته پردازش می‌کنیم، فرض می‌کنیم که کانال را با همان نرخ نمونه‌برداری که سیگنال دریافتی را نمونه‌برداری می‌کنیم، که در فواصل زمانی بیت nT است، نمونه‌برداری می‌کنیم. بنابراین، می‌توانیم از پاسخ ضربه‌ی $h[n] = h(nT)$ سیستم معادل در زمان گسسته استفاده کنیم. فرض کنید کانال یک فیلتر FIR (پاسخ ضربه‌ی محدود) با چهار tap است که به صورت زیر داده شده است:

$$h[n] = h(nT) = \delta[n] + \frac{1}{4}\delta[n-1] + \frac{3}{4}\delta[n-2] - \frac{2}{4}\delta[n-3]. \quad (8)$$

این سه پژواک با تأخیر زمانی ممکن است انعکاس‌های تضعیف‌شده سیگنال در محیط را نشان دهند (مثلاً پژواک‌هایی از ساختمان‌ها، اتومبیل‌ها، تپه‌ها، درختان در ارتباطات بی‌سیم). این پژواک‌ها باعث اعوجاج در سیگنال دریافتی می‌شوند، زیرا امواج دریافتی بیت‌های مختلف در زمان هم‌پوشانی خواهند داشت و به این ترتیب باعث ISI می‌شوند.

نمایش $h[n]$ را در Python به‌عنوان یک بردار انجام دهید. توجه داشته باشید که چهار tap باید به‌طور مساوی فضاگذاری شوند تا یک مدت زمان بیت کامل T بین هر دو tap متوالی باشد. بنابراین اگر شما از ۳۲ نمونه برای نمایش هر بیت در قسمت مدولاسیون استفاده کنید، باید ۳۱ صفر بین هر دو tap متوالی قرار گیرد. دقت کنید که تعداد کل نمونه‌ها باید یک عدد توان ۲ باشد تا کد شما سریع‌تر اجرا شود.

حالا جریان بیت مدوله‌شده را از طریق کانال ارسال کنید.

پرسش ۵: پاسخ فرکانسی (هم فاز و هم اندازه) و پاسخ ضربه‌ی کانال را رسم کنید.

پرسش ۶: Eye Diagram خروجی کانال را برای هر تابع سازنده‌ی پالس رسم کنید. در نتایج خود توضیح دهید. همچنین ارتباط بین بازشدگی چشم و ISI را توضیح دهید.

۵.۳ نویز

یک ماتریس نویز تصادفی گاوسی با دستور `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.

پرسش ۷: دوباره نمودارهای چشم خروجی کانال را با نویز اضافه‌شده رسم کنید. نتایج خود را توضیح دهید. (شما می‌توانید با مقادیر مختلف توان نویز σ^2 آزمایش کنید).

۶.۳ فیلتر تطبیق‌یافته Mathced Filter

فیلتر تطبیق‌یافته به گونه‌ای طراحی می‌شود که با فیلتر سازنده‌ی پالس ارسال‌شده (مانند معادله‌ی ۲) تطابق داشته باشد. برای هر یک از دو تابع سازنده‌ی پالس داده شده، تابع پاسخ ضربه‌ی فیلتر تطبیق‌یافته را به همان روشی که تابع سازنده‌ی پالس را می‌سازید، پیاده‌سازی کنید.

پرسش ۸: پاسخ ضربه و پاسخ فرکانسی فیلتر تطبیق‌یافته را برای هر تابع سازنده‌ی پالس رسم کنید.

پرسش ۹: Eye Diagram را برای هر شکل پالس در خروجی فیلتر تطبیق‌یافته رسم کنید. بهترین نقطه نمونه‌برداری طبق Eye Diagram شما کدام است؟ نتایج خود را توضیح دهید.

۷.۳ جبران‌ساز

در این بخش، شما یک فیلتر جبران‌ساز را برای بازبازی سیگنال مدوله‌شده که قبل از عبور از کانال بوده است، اعمال خواهید کرد. در این پروژه، ما دو نوع فیلتر جبران‌ساز مختلف را بررسی خواهیم کرد.

Zero-Forcing Equalizer ۱.۷.۳

پاسخ فرکانسی این فیلتر معکوس پاسخ کانال است. فرض کنید که $H(e^{j\omega})$ پاسخ فرکانسی کانال $h[n]$ باشد، سپس پاسخ فرکانسی فیلتر ZF به شکل زیر است:

$$Q_{ZF}(e^{j\omega}) = \frac{1}{H(e^{j\omega})}. \quad (9)$$

پرسش ۱۰: توضیح دهید که چگونه فیلتر ZF را پیاده‌سازی می‌کنید. پاسخ فرکانسی و پاسخ ضربه این فیلتر را رسم کنید. پاسخ فرکانسی حاصل را بررسی کنید. آیا این فیلتر Zero-Forcing پایدار است؟ آیا می‌توان انتظار داشت که معکوس کانال همیشه پایدار باشد؟

پرسش ۱۱: Eye Diagram ها را برای هر شکل پالس در خروجی فیلتر Equalizer رسم کنید. به یاد داشته باشید که سطح نویز σ^2 را در نمودار خود مشخص کنید. در مورد نتایج خود توضیح دهید.

MMSE Equalizer ۲.۷.۳

اگر نویزی وجود نداشت و کانال معکوس پذیر بود، فیلتر ZF سیگنال مدوله‌شده را به طور کامل بازیابی می‌کرد. اما در عمل، همیشه نویز وجود دارد. علاوه بر این، اغلب نمی‌دانیم که پاسخ کانال دقیقاً چیست چون فقط می‌توانیم اندازه‌گیری‌های پرازنویز از آن داشته باشیم. پاسخ کانال همچنین ممکن است به طور ناگهانی تغییر کند (تصور کنید فردی در حال استفاده از تلفن همراه در حال پیاده‌روی یا سفر با ماشین است) و از این رو اندازه‌گیری‌های کانال باید به طور مکرر به‌روزرسانی شوند که باعث می‌شود کانال حساس‌تر به نویز شود. بنابراین در اینجا یک فیلتر متفاوت بررسی می‌کنیم که هم پاسخ کانال و هم نویز را در نظر می‌گیرد.

امکان طراحی فیلترهایی وجود دارد که همزمان پاسخ کانال را معکوس کرده و سعی در جبران برخی اثرات نویز دارند. یکی از فیلترهای پرکاربرد از این نوع، فیلتر حداقل خطای مربعات (MMSE) است، چون این فیلتر میانگین مربعات اختلاف بین سیگنال ارسالی و سیگنال دریافتی را کمینه می‌کند. فیلتر MMSE نویز را از طریق نسبت سیگنال به نویز (SNR) در نظر می‌گیرد که به‌عنوان نسبت توان سیگنال به توان نویز تعریف می‌شود (که در نقطه نمونه‌برداری درگیرنده محاسبه می‌شود). توان سیگنال از توابع سازنده پالس‌هایی که در مرحله مدولاسیون استفاده می‌شوند، به دست می‌آید. برای پالس half sine، توان متوسط آن برابر با $\frac{1}{2}$ است. برای پالس SRRC، ما ضریب نرمال‌سازی A را انتخاب می‌کنیم به‌طوری‌که انرژی پالس (بر روی مدت زمان کوتاه‌شده‌اش) همانند انرژی پالس half sine باشد، به طوری‌که توان سیگنال در نقطه‌ی نمونه‌برداری درگیرنده برای هر دو تابع سازنده پالس یکسان باشد.

توان نویز برابر با σ^2 است، همانطور که در ایجاد نویز گاوسی افزایشی در مرحله اضافه کردن نویز به سیستم استفاده شده است. پاسخ فرکانسی فیلتر MMSE به صورت زیر داده می‌شود:

$$Q_{MMSE}(e^{j\omega}) = \frac{H^*(e^{j\omega})}{|H(e^{j\omega})|^2 + 2\sigma^2} \quad (10)$$

توجه داشته باشید که وقتی نویز ضعیف است، یعنی وقتی که $\sigma^2 \rightarrow 0$ ، فیلتر MMSE به فیلتر ZF نزدیک می‌شود. بنابراین، زمانی که توان نویز کم است، دو فیلتر باید عملکرد مشابهی داشته باشند، اما وقتی که توان نویز زیاد است، انتظار می‌رود که MMSE عملکرد بهتری نسبت به فیلتر ZF داشته باشد.

پرسش ۱۲: روش استفاده‌شده برای پیاده‌سازی فیلتر MMSE را توضیح دهید. پاسخ فرکانسی و پاسخ ضربه‌ی این فیلتر را رسم کنید. نتایج را تحلیل کرده و آن را با نتایج فیلتر ZF مقایسه کنید.

پرسش ۱۳: Eye Diagram های خروجی جبران‌ساز MMSE را رسم کنید. نتایج خود را تحلیل کنید.

۸.۳ نمونه برداری و شناسایی

در تبدیل سیگنال از حوزه‌ی پیوسته به گسسته، باید سیگنال را نمونه برداری کنیم. در اینجا ما در خروجی فیلتر جبران ساز، در فواصل زمانی از بیت‌ها نمونه برداری می‌کنیم. اما قبل از نمونه برداری، باید زمان شروع اولین بیت دریافت شده را تعیین کنیم و بهترین زمان نمونه برداری را مشخص کنیم. زمان‌های نمونه برداری بعدی فقط ضریبی از فواصل زمانی بیت‌ها از اولین نمونه برداری هستند.

برای این کار، با استفاده از آستانه‌ی صفر، اگر نمونه مثبت باشد، می‌گوییم یک (۱) دریافت کرده‌ایم، در غیر این صورت صفر (۰) دریافت کرده‌ایم. از این داده‌ها یک ماتریس از بیت‌ها بسازید که در ردیف n -ام آن، هشت ورودی وجود داشته باشد که مربوط به هشت بیت از پیکسل n -ام دریافت شده و شناسایی شده است.

۹.۳ تبدیل به تصویر

حال، ماتریس شناسایی شده را به یک بردار از اعداد صحیح تبدیل کنید و با استفاده از دستورات مناسب بلوک‌های 8×8 DCT دریافت شده را در مکان مناسب خود در تصویر کامل قرار دهید.

۱۰.۳ پس پردازش تصویر

بلوک‌های DCT را با استفاده از دستور `skimage.util.view_as_blocks` معکوس کنید و تصویر حاصل را با استفاده از معکوس مقیاس‌بندی خطی از مرحله پیش پردازش مقیاس‌بندی کنید.

پرسش ۱۴: نتیجه را نمایش دهید. آیا تصویر شما به طور کامل بازبازی شده است؟ در مورد آنچه می‌بینید نظر دهید.

- یک تصویر خروجی در سطح نویز که بازبازی کامل را ممکن می‌سازد.
- تصویر در سطح نویزی که باعث ایجاد برخی خطاها در بازبازی می‌شود.
- تصویری که در آن خطاهای زیادی در بازبازی دیده می‌شود.

برای هر تصویر، قدرت نویز را مشخص کنید. این آزمایش را برای هر دو شکل پالس ورودی انجام دهید.

۴ پیاده سازی پروژه و تحلیل

۱.۴ نکات پیاده سازی

برای پیاده سازی پروژه، شما باید هر بلوک عملکردی را به طور جداگانه آزمایش کنید قبل از اینکه همه‌ی آن‌ها را به یک سیستم متصل کنید. به عنوان مثال، ممکن است بخواهید ابتدا فقط ۱۰ بیت تصادفی را از طریق کانال ارسال کنید، بدون نویز، سپس فیلتر ZF را اعمال کنید تا اطمینان حاصل کنید که همه‌ی بیت‌ها به طور کامل بازبازی شده‌اند. پس از آن نویز را اضافه کنید و فیلتر MMSE را آزمایش کنید. سپس بلوک‌های مربوط به تصویر را متصل کنید و با یک تصویر واقعی آزمایش کنید.

۲.۴ 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.

۱.۲.۴ اثر نویز و جبران سازی

شبیه سازی را به ازای مقادیر مختلف قدرت نویز σ^2 تکرار کنید تا مقدار بحرانی را که به طور دقیق منجر به خطا در تصویر بازبازی شده می‌شود، پیدا کنید. این مقدار به چه SNR-ای مربوط می‌شود؟ از آنجا که از گیرنده فیلتر منطبق استفاده می‌کنیم، می‌توانید SNR را با تقسیم انرژی پالس بر قدرت نویز σ^2 محاسبه کنید. انرژی پالس با میانگین مربعات مقادیر نمونه در هر پالس محاسبه می‌شود.

پرسش ۱۵: مقادیر آستانه‌ی بحرانی SNR (به دسی بل) که برای هر دو فیلتر جبران ساز پیدا کردید را گزارش کنید. کدامیک از

انواع فیلترها آستانه‌ی بحرانی کمتری دارد؟ آیا این منطقی است؟ پرسش ۱۶: دو تصویر مختلف را آزمایش کنید. برای هر تصویر، یک مقدار از σ^2 انتخاب کرده و در گزارش خود تصاویر بازبازی شده با هر دو فیلتر جبران‌ساز و SNR مربوطه را شامل کنید. نتایج خود را مورد تحلیل قرار دهید.

۳.۴ اثر تابع سازنده‌ی پالس

ما دو تابع سازنده‌ی پالس مختلف را پیاده‌سازی کردیم، پالس‌های نیمه‌سینوسی و پالس‌های SRRC.

پرسش ۱۷: آیا این پالس‌ها معیار نیکوییست برای عدم تداخل بین نمونه‌ها (Zero-ISI) را برآورده می‌کنند؟ در کدام نقطه از سیستم (پس از فیلتر منطبق یا پس از فیلتر جبران‌ساز) انتظار دارید که Zero-ISI در نقطه‌ی نمونه‌برداری باشد؟ دلیل خود را ارائه دهید. آیا درک شما با مشاهدات شما در این پیاده‌سازی هم‌خوانی دارد؟

پرسش ۱۸: در همان قدرت نویز و با فرض اینکه دو تابع سازنده پالس انرژی یکسانی دارند، آیا انتظار دارید که تفاوتی در عملکرد خطا وجود داشته باشد؟ دلایل خود را بیان کنید. آیا شبیه‌سازی با انتظارتان منطبق است؟ در گزارش خود، تصویری از خروجی برای هر تابع سازنده پالس در همان سطح نویز شامل کنید.

پرسش ۱۹: اثرات این توابع سازنده‌ی پالس را بر پهنای باند انتقال مقایسه و تحلیل کنید. اثر طول پالس در تعداد دوره‌های زمانی بیت‌ها را بر عملکرد سیستم بحث کنید.

پرسش ۲۰: بر اساس تمام نکات بالا، کدام شکل پالس بهتر است؟ همچنین هرگونه معایب آن را بررسی کنید.

۴.۴ اثر کانال

پس از تکمیل سیستم شبیه‌سازی کامل، دو کانال دیگر را با پاسخ ضربه داده شده در زیر آزمایش کنید. این‌ها مدل‌های واقعی کانال بی‌سیم هستند که در طراحی سیستم‌های سلولی استفاده می‌شوند. یک کانال فضای باز به شکل زیر داده شده است:

$$h_1[n] = 0.5\delta[n] + \delta[n-1] + 0.63\delta[n-3] + 0.25\delta[n-8] + 0.16\delta[n-12] + 0.1\delta[n-25].$$

و یک کانال فضای بسته به شکل زیر است:

$$h_2[n] = \delta[n] + 0.4365\delta[n-1] + 0.1905\delta[n-2] + 0.0832\delta[n-3] + 0.0158\delta[n-5] + 0.003\delta[n-7].$$

توجه داشته باشید که تأخیرهای طولانی در این کانال‌ها وجود دارد، به‌ویژه در کانال فضای باز. پس از ترسیم پاسخ فرکانسی این کانال‌ها، باید بتوانید جایی که کانال به طور قابل توجهی تضعیف می‌شود (که به آن "افت عمیق" گفته می‌شود) و تفاوت بین پاسخ کانال فضای باز و فضای بسته را مشاهده کنید.

پرسش ۲۱: آیا ارتباط شما هنگام اتصال یک کانال جدید به‌طور یکپارچه کار کرد؟ در گزارش خود، تصویری از خروجی هر کانال جدید شامل کنید. عملکرد سیستم خود را تحت کانال‌های مختلف تحلیل و بررسی کنید.

تذکر پیاده‌سازی: از آنجا که کانال فضای باز تأخیر بسیار طولانی دارد، معادل‌سازهای آن باید پاسخ ضربه‌ی بسیار طولانی‌تری داشته باشند. همچنین می‌توانید با طول N بلوک انتقال (بخش تبدیل به جریان بیت) که هر بار از کانال و معادل‌ساز عبور می‌کند، آزمایش کنید تا ببینید آیا این امر در کیفیت تصویر خروجی تفاوتی ایجاد می‌کند یا خیر.