



تمرین کامپیوتری «مخابرات دیجیتال»

تهیه کننده: حامد شریفی، استاد درس: دکتر مهدی مهدوی - نیم سال 4032

مباحث پروژه: ارسال داده از کانال شبیه سازی شده

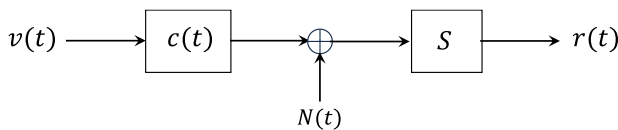
هدف این پروژه، کامل کردن مباحث تئوری با تجربه واقعی است. آنچه در این پروژه انجام می دهیم، شباهت زیادی با آنچه یک مهندس مخابرات برای شبیه سازی کامل یک سیستم انجام می دهد، دارد.

مراحل کلی

یک فایل text که حاوی یک جمله است در اختیار شما قرار می گیرد. شما باید این جمله را در بخش فرستنده به یک سیگنال تبدیل کنید. سیگنال را از تابعی که رفتار کانال را شبیه سازی می کند، عبور داده و سپس در گیرنده، سیگنال دریافتی را به فایل text تبدیل کنید.

مدل کانال و اثرات آن

سیگنال خروجی کانال، $r(t)$ ، مطابق بلوک دیاگرام زیر به سیگنال ورودی مرتبط می شود.



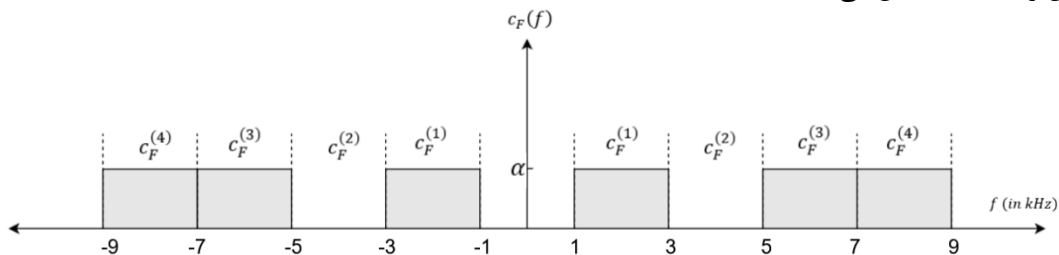
که c یک فیلتر می انگذرد علی (causal passband filter)، S یک برش دهنده (clipper) می باشد. ضمناً $N(t) \sim \mathcal{N}(0, \sigma^2)$

همچنین فرض می کنیم که c به صورت زیر تصادفی است: هر زمان که شما داده ای را به صورت کامل به کانال ارسال می کنید، $C(f)$ فرم زیر را می گیرد:

$$C(f) = \left[\sum_{k=1}^4 C^k(f) \right] - C^i(f)$$

$$C^k(f) = \begin{cases} \alpha & |f - 2000k| \leq 1000 \\ 0 & \text{otherwise} \end{cases}$$

که i به صورت تصادفی و با احتمال یکسان از مجموعه $\{1, 2, 3, 4\}$ انتخاب می شود. $\alpha \in (0, 1]$ ثابت است. شکل زیر یک فیلتر ممکن را با $i = 2$ نشان می دهد:



تابع $S(x)$ نیز خروجی را به صورت زیر برش می‌دهد:

$$S(x) = \begin{cases} 1 & x \geq 1 \\ x & -1 < x < 1 \\ -1 & x \leq -1 \end{cases}$$

کد مربوط به پیاده‌سازی کانال برای راحتی شما به همراه فایل پروژه در سامانه درس با عنوان `simulate_channel_project.m` قرار داده شده است ولی بهتر است به طور دقیق آن را بررسی کنید تا در طراحی گیرنده و فرستنده دچار مشکل نشوید. دقت کنید که در تمام مراحل پروژه، بجز قسمت اختیاری 5 مربوط به پرسش‌های زیر، شما مجاز به تغییر کد متلب مربوط به کانال نیستید.

پرسش‌های مربوط به پیاده‌سازی کانال:

- (1) در کدام قسمت، یکی از چهار باند به صورت تصادفی انتخاب و حذف می‌شود؟ به نظر شما، چگونه می‌توان با این اثر کانال در فرستنده و گیرنده مقابله کرد؟ (راهنمایی: می‌توانید پیام را در فرستنده، در چند باند مختلف ارسال کنید)
- (2) در کدام قسمت، باند بالا نیز به دلیل حفظ تقارن سیگنال ورودی به کانال حذف می‌شود؟
- (3) در کدام قسمت، نا همزمانی (asynchrony) یا عدم تطابق سمپلینگ زمانی بین فرستنده و گیرنده پیاده‌سازی شده است؟ تاثیر اضافه کردن این خاصیت به کانال در حوزه زمان و فرکانس سیگنال خروجی کانال چیست؟ راه حل پیشنهادی شما برای مقابله با این اثر کانال چیست؟
- (4) در کدام قسمت، کانال سیگنال خروجی را برش (clip) می‌دهد؟
- (5) (اختیاری) می‌خواهیم یک محدودیت جدید بر روی انرژی متوسط سیگنال ورودی به کانال اضافه کنیم. چنانچه سیگنال ورودی $v(t)$ باشد، و انرژی متوسط آن E_{avr} باشد، کد کانال را به نحوی تغییر بدهید که اگر $E_{avr} > 1$ باشد، آنگاه دامنه سیگنال ورودی $v(t)$ بدون تغییر باقی بماند و اگر $E_{avr} < 1$ ، دامنه سیگنال ورودی $v(t)$ ، \sqrt{S} برابر شود و به عبارتی نویز $N(t)$ به اندازه \sqrt{S} اسکیل پیدا کند. این کار برای حفظ نسبت SNR انجام می‌شود. شما باید مقدار مناسب \sqrt{S} را با تابع مناسب در متلب بیابید.

طراحی فرستنده و گیرنده برای این کانال

مرحله اول: شما باید فرستنده ای را پیاده سازی کنید که فایل `target.txt` را بخواند. برای خواندن فایل `text` و تبدیل محتوا به کد اسکی، می‌توانید پس از باز کردن آدرس مورد نظر با دستور `fopen()` در متلب، از دستور `fread([1 info], "uint8")` استفاده بکنید تا متلب، محتوای فایل را به صورت اعداد اسکی در یک متغیر ذخیره کند و در انتها با دستور `fclose()`، فایل را ببندید. اکنون می‌توانید اعداد اسکی را با استفاده از توابع `dec2bin` و `reshape` به اعداد باینری تبدیل کنید.

مرحله دوم (اختیاری): سپس باید متن باینری به دست آمده از مراحل قبل را توسط کدینگ منبع هافمن (Huffman Source Encoding)، آماده ارسال به بلوک کدینگ کانال کنید. برای آشنایی با نحوه کدنویسی در کدینگ هافمن، می‌توانید از [لینک](#) استفاده بکنید. برای سادگی در این پروژه، می‌توانید از توابع آماده `huffmanenco` و `huffmandict` در متلب، استفاده کنید.

مرحله سوم: روش های Channel Coding، به دو قسمت Error Detection و Error Correction تقسیم می‌شوند. برای تشخیص خطا در گیرنده می‌توانید از روش کدینگ همینگ (15,11) یا (7,4) ([Hamming Code](#)) استفاده کنید. توضیحات این [لینک](#) می‌تواند در مورد تصحیح خطا مفید باشد. ماتریس G را به صورت دلخواه و مناسب، اختیار کنید. در سایت github نیز می‌توانید نمونه هایی از کدینگ همینگ را مشاهده کنید.

(برای مطالعه بیشتر با روش های تشخیص خطا در گیرنده، می‌توانید به لینک های [Parity Bit,Checksum](#) یا [CRC](#) مراجعه کنید).

مرحله چهارم: حال دنباله بدست آمده آماده ارسال بر روی کانال می‌کنیم. ابتدا با استفاده از فیلتر کسینوسی افزایش یافته (Raised Cosine Filter)، پهنای باند سیگنال را محدود و مقاوم به رخدادن ISI احتمالی کنید. نرخ ارسال را به عنوان ورودی در نظر بگیرید (چگونه اینکار را انجام می‌دهید؟). در این قسمت، می‌توانید از مدولاسیون های QAM یا PSK یا FSK با درجه دلخواه M استفاده نمایید. سایر پارامتر های مدولاسیون را به نحوی انتخاب کنید که در نرم افزار متلب قابل پیاده سازی باشد. (مثلا حافظه سریع پر نشود یا متلب هنگ نکند).

تذکره 1) همانطور که در بخش مدل سازی کانال بیان شد، کانال به صورت تصادفی یکی از چهارباند در دسترس را حذف می‌کند. فرستنده را طوری طراحی کنید که به این اثر کانال مقاوم باشد و مطمئن باشیم سیگنال پیام ما به دست گیرنده خواهد رسید.

تذکره 2) مطابق مدل کانال ارائه شده، مدولاسیون شما باید در محدوده فرکانسی 1 kHz تا 9 kHz باشد تا بتواند به درستی از کانال عبور کنند.

تذکره 3) با توجه به نا همزمانی (asynchrony) یا تاخیر زمانی نامشخص بین ورودی و خروجی کانال، راه حل خود را برای مقابله با این تاخیر تصادفی کانال بیان کنید. **راه حل پیشنهادی:** دو دنباله قراردادی را به عنوان سیگنال **پایلوت** به اندازه کافی بزرگ در فرستنده و گیرنده به نام های `pilot_strat` و `pilot_end` تعریف کنید. این دو سیگنال قبل از ارسال به کانال، باید به ابتدا و انتهای سیگنال ارسالی به کانال چسبانده شوند. درگیرنده، هنگامی که سیگنال خروجی از کانال بیشترین شباهت را به سیگنال `pilot_strat` پیدا کرد، این لحظه همان لحظه شروع سیگنال `pilot_strat` است و می‌فهمیم در ادامه آن، سیگنال پیام قرار دارد و سپس `pilot_end` قابل مشاهده است. بررسی میزان شباهت دو سیگنال را می‌توانید با استفاده از محاسبه همبستگی متقابل دو سیگنال بدست آورید.

مرحله پنجم: اکنون سیگنال خروجی فرستنده را به تابع `simulate_channel_project` بدهید تا سیگنال از کانال عبور کند و خروجی کانال را به گیرنده بدهید.

مرحله ششم: حال در گیرنده، ابتدا باید تاخیر ایجاد شده در ابتدا و انتهای سیگنال ارسالی را با دانستن `pilot_strat` و `pilot_end` در گیرنده، تخمین بزنید. سپس سیگنال مدوله شده را از `pilot_strat` و `pilot_end` جدا کنید. چنانچه پیام خود را در چند باند مختلف ارسال کرده‌اید، بررسی کنید در کدام باندها سیگنال از دست رفته و در کدام باندها، سیگنال همچنان وجود دارد. حال دمدولاسیون را بر روی باندی از پیام که همچنان سیگنال در آن وجود دارد، انجام بدهید تا سیگنال در باند پایه قرار بگیرد.

(اختیاری): چنانچه روش پیاده‌سازی مدولاسیون انتخاب شده نیاز به PLL در گیرنده دارد، می‌توانید PLL مورد نیاز را به صورت کد متلب در لپتاب گیرنده پیاده سازی کنید.

مرحله هفتم: پس از دمدولاسیون، بلوک مورد نیاز برای `Detection` بیت های دریافت شده را که با نویز جمع شده اند، به صورت کد متلب پیاده سازی کنید.

مرحله هشتم: طبق روش `Channel Decoding` استفاده شده در مرحله سوم، خطاهای ایجاد شده را تشخیص یا تصحیح بکنید. می‌توانید از روش `Hard – Decision Decoding` استفاده کنید.

مرحله نهم (اختیاری): با استفاده از دیکدینگ منبع هافمن (`Huffman Source Decoding`) که در مرحله دوم کد شده بود، و استفاده از تابع آماده `huffmandeco` در متلب، رشته باینری را دیکد کنید تا آماده تبدیل به متن شود.

مرحله دهم: در انتها رشته باینری را به کد اسکی و سپس اعداد دسیمال اسکی را با استفاده از دستور `char()` به متن تبدیل کرده و فایل `txt` را مجدداً تولید کرده و با فایل ارسالی در فرستنده مقایسه کنید.

خروجی پروژه:

الف) سیگنال خروجی کانال در زمان، سیگنال خروجی کانال بعد از حذف پیلوت‌ها در زمان و فرکانس، سیگنال دمدوله شده در زمان و فرکانس، و دنباله باینری قبل از تبدیل شدن به متن در گیرنده (خروجی مرحله نهم و اول) را رسم کنید.

ب) منحنی خطا بر حسب نرخ ارسال را رسم کنید. برای محاسبه خطا می‌توانید تعداد بیت دارای خطا را به کل بیت ارسال، تقسیم کنید. برای این منظور محور افقی را نرخ ارسال (bps) قرار بدهید و آن را تغییر دهید. همچنین این منحنی را برای هنگامی که کدینگ کانال انجام نمی‌شود رسم کنید. هر دو منحنی را یکجا رسم نمایید.

پ) با تغییر پارامتر α به عنوان می‌زان تضعیف کانال در باندهای گذر، مقدار SNR را می‌توان تغییر داد. تأثیر α را بر روی SNR بدست بیاورید.

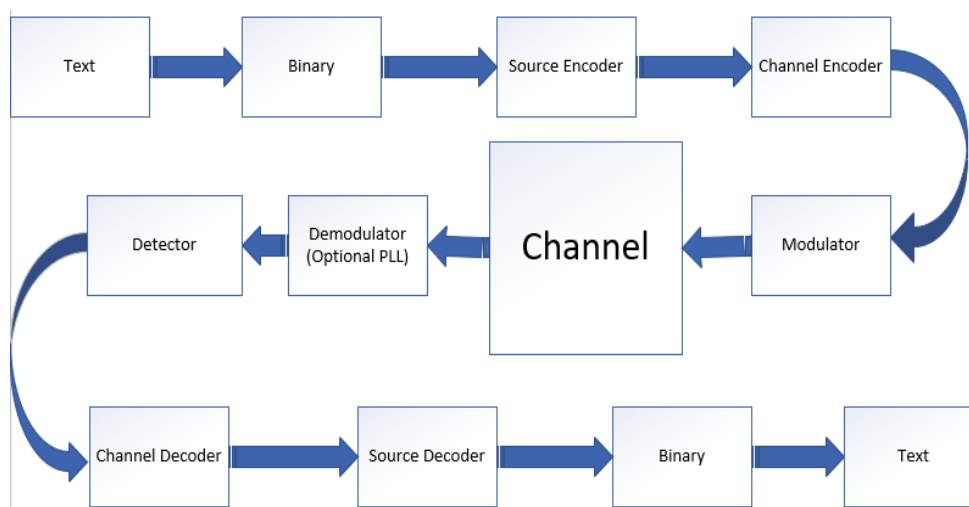
ت) با تغییر پارامتر σ به عنوان انحراف معیار نویز در کانال، SNR سیگنال خروجی کانال در گیرنده تغییر می‌کند. نمودار می‌زان خطا را بر حسب σ رسم کنید. (برای محاسبه خطا، می‌توانید اختلاف اعداد دسیمال اسکی متن اصلی و متن دریافت شده را محاسبه کرده و یا خطا را به گونه ای دیگر تعریف کنید.) واضح است ارسال صحیح متن با σ بیشتر، نشان از عملکرد بهتر سیستم در قسمت های Source Coding, Modulation, Channel Coding و Data Rate است.

ث) (اختیاری) راهکاری پیشنهاد بدهید که بتوان ظرفیت کانال را بطور عملی تخمین زد.

نکات:

این پروژه در گروه ها دو تا سه نفره قابل انجام است.

تمام کد ها در این پروژه باید در محیط متلب نوشته شود.



بلوک دیاگرام پروژه

گزارش نهایی:

متن گزارش نهایی باید فایل pdf و word شامل نام و نام خانوادگی و شماره ی دانشجویی اعضای گروه، توضیح مختصر درمورد پیاده سازی های انجام شده، شکل های خواسته شده (ذخیره شده در فایل word گزارش) و پاسخ به سوالات خواسته شده به همراه کد های استفاده شده باشد. گزارش نوشته شده را به همراه فایل های اجرایی مربوطه به صورت یک فایل zip ارسال نمایید. جلسه ارائه نیز اطلاع رسانی خواهد شد.