

گزارش پروژه درس مخابرات دیجیتال

ابتدا مطابق با توضیحات دستور کار پروژه، فایل با پسوند WAV ایجاد می‌کنیم. سپس با استفاده از دستور `audioread()` فایل مورد نظر را در نرم‌افزار متلب باز می‌کنیم.

از آن جایی که برای ارسال دیتای باینری از یک سیستم مخابراتی دیجیتال با مدولاسیون BFSK و یک کانال AWGN با پهنای باند میانی ۱۶۰ KHz استفاده می‌کنیم، در بهینه‌ترین حالت استفاده از پهنای باند، فاصله دو فرکانس مرکزی برابر است با:

$$\Delta f = f_{c2} - f_{c1} = \frac{160\text{KHz}}{2} = 80\text{KHz}$$

حال به طور دلخواه فرکانس مرکزی کانال را برابر ۱۲۰ کیلوهرتز در نظر می‌گیریم. در نتیجه داریم:

$$f_{c1} = 80\text{KHz}$$

$$f_{c2} = 160\text{KHz}$$

در این حالت ماکسیمم فرکانس عبوری از کانال برابر ۲۰۰ کیلوهرتز است. در نتیجه اگر فرکانس نمونه‌برداری را برابر ۴۰۰ کیلوهرتز در نظر بگیریم، سیگنال پیوسته از روی نمونه‌های آن به طور یکتا قابل بازسازی است (فرض می‌کنیم ماکسیمم فرکانس سیگنال ورودی کوچک‌تر مساوی ۲۰۰ کیلوهرتز است و در نتیجه نیازی به عبور سیگنال ورودی از فیلتر پایین‌گذر با فرکانس قطع ۲۰۰ کیلوهرتز قبل از نمونه‌برداری از آن وجود ندارد).

الف) از آن جایی که گیرنده ناهمدوس فرض شده است، برای آن که سیگنال‌ها متعامد باشند، باید داشته باشیم:

$$\Delta f = \frac{1}{T} \rightarrow T = \frac{1}{\Delta f} = \frac{1}{80\text{KHz}} = \frac{1}{80\text{K}}\text{s} \rightarrow R_b = \frac{1}{T} = 80\text{Kbps}$$

فرکانس نمونه‌برداری در فایل WAV خوانده شده، برابر ۴۴.۱ کیلوهرتز است. در نتیجه ابتدا با استفاده از تابع `resample()` تعدادها را به نسبت $\frac{100}{441}$ کاهش می‌دهیم تا فرکانس نمونه‌برداری برابر ۱۰ کیلوهرتز شود. حال با توجه به این که نرخ ارسال برابر ۸۰ کیلوبیت در هر ثانیه است، حداکثر می‌توان هر نمونه را با ۸ بیت کوانتیزه کرد. برای این کار ابتدا با محاسبه مقادیر ماکسیمم و مینیمم سیگنال ورودی، دامنه سیگنال ورودی را به بازه $-1 \leq x < 1$ فشرده می‌کنیم یا بسط می‌دهیم. سپس با توجه به تعداد بیت‌های کوانتیزاسیون (k بیت)، دامنه مقادیر سیگنال ورودی را به بازه $0 \leq x \leq 2^k - 1$ منتقل می‌کنیم. در آخر با استفاده از توابع `cast()` و `de2bi()` بیت‌های لازم را بدست می‌آوریم.

در ادامه فرض می‌کنیم سیگنال پیوسته ارسال موجود است. از این سیگنال با فرکانس ۴۰۰ کیلوهرتز نمونه‌برداری می‌کنیم. حال با استفاده از تابع `awgn()` به نمونه‌های سیگنال پیوسته ارسالی، نویزی گوسی با مقدارهای مختلف SNR اضافه می‌کنیم. سپس با استفاده از ساختار دمدولاسیون مربوط به گیرنده ناهمدوس، بیت‌های ارسالی را تشخیص می‌دهیم و با استفاده از میانگین آماری، احتمال خطا را محاسبه می‌کنیم.

احتمال خطا برای حالات مختلف به صورت زیر بدست می‌آید:

$$[0.0532, 0.0035, 1.7884 \times 10^{-6}, 0]$$

نتایج بدست‌آمده تقریباً با نتایج تئوری برابر است. البته شایان ذکر است احتمال خطاهای بدست‌آمده از شبیه‌سازی کمی از نتایج تئوری بهتر است.

با مقایسه فایل‌های صوتی در می‌یابیم برای $SNR = 0dB$ تشخیص کلمات تقریباً غیر ممکن است؛ اما برای SNR های بیشتر کیفیت فایل صوتی به قدری است که بتوان متن پیام را متوجه شد. هرچه مقدار SNR افزایش می‌یابد، کیفیت صوت در خروجی نیز زیاد می‌شود. کیفیت در دو حالت $SNR = 10dB$ و $SNR = 15dB$ تقریباً برابر است؛ زیرا در این دو حالت احتمال خطا تقریباً صفر است و تنها عامل تضعیف کیفیت تعداد بیت‌های کوانتیزاسیون است که در دو حالت یکسان است.

ب) در این حالت بر اساس دستور کار پروژه، ۲ بیت از تعداد بیت‌های کوانتیزاسیون کم می‌کنیم. در نتیجه نرخ تولید بیت برابر است با:

$$R_b = 10 \frac{Ksymbol}{s} * 6 \frac{bit}{symbol} = 60Kbps$$

در نتیجه می‌توانیم از یک BCH مناسب با نرخ بزرگ‌تر مساوی $\frac{3}{4}$ استفاده کنیم. در این راستا بهترین انتخاب ممکن کد (127,99) با قدرت تصحیح ۴ بیت است.

با استفاده از این کد نرخ ارسال برابر است با:

$$R_c = \frac{n}{k} * R_b = \frac{127}{99} * 60kbps = 76.96kbps$$

لازم به ذکر است در این حالت مقداری از پهنای باند بدون استفاده باقی می‌ماند.

احتمال خطا برای حالات مختلف به صورت زیر بدست می‌آید:

$$[0.0509, 0.0001, 0, 0]$$

مشاهده می‌شود در این حالت برای $SNR = 5dB$ ، احتمال خطا به طور چشم‌گیری کاهش می‌یابد. علت این است که در این حالت احتمال خطای بیت آنقدر کوچک است که در اکثر مواقع کد BCH قدرت تصحیح خطاهای به وجود آمده را دارد. در مورد $SNR = 0dB$ این موضوع برقرار نیست. زیرا در این حالت تعداد خطاها در بیشتر مواقع بیشتر از قدرت تصحیح کد BCH است؛ در نتیجه اگرچه در بعضی موارد خطاهای به وجود آمده تصحیح می‌شوند، در اکثر مواقع تعداد خطاها به علت تصحیح اشتباه افزایش می‌یابد. میانگین این موارد باعث می‌شود احتمال خطا تغییر چندانی نکند.

ج) در این حالت پس از اینکه دامنه سیگنال ورودی را به بازه $-1 \leq x < 1$ فشرده کردیم یا بسط دادیم، سیگنال ورودی را از تابع $companioner()$ عبور می‌دهیم و سپس خروجی این تابع را به بازه $0 \leq x \leq 2^k - 1$ منتقل می‌کنیم. در نتیجه این کار به ازای تعداد بیت کوانتیزاسیون یکسان، مقدار SQNR به شدت کاهش می‌یابد و کیفیت صوت خروجی افزایش می‌یابد.