

# گزارش کار آزمایش مخابرات ۱۵ تیر

«یک افسانه دیجیتال»»»»

بخش اول: عملیات چریکی و شناسایی دشمن!

## هدف من در بخش اول پروژه

در همان ابتدای کار سعی کردم همه بلوک‌ها را تست کنم تا بتوانم یک دید کلی از کار داشته باشم.

پس به پیاده‌سازی بلوک‌های مختلف پرداختم. اول از همه هم از بلوکی که به نظرم سخت تر و حیاتی تر می‌رسید شروع کردم.

طی این کار گاهی از نرم‌افزار GNU octave به جای متلب استفاده می‌کنم. این برنامه هنوز ناکامل است اما مزایای مهمی نسبت به متلب دارد که من را به استفاده از آن ترغیب می‌کند.

## بلوک دیاگرام

سیستم من که قصد پیاده‌سازی آن را دارم:

فایل ورودی و کدینگ منبع ← رمزنگاری ساده ← کدینگ کانال ← مدولاسیون ← کانال ← دیمدولاسیون ← دیکدینگ کانال ← رمزگشایی ← کدگشایی منبع و گرفتن فایل خروجی

## کارهایی که کردم

### بلوک مدولاسیون

اولین کاری که سعی کردم پیاده‌سازی بلوک مدولاسیون بود. من مدولاسیون PSK را انتخاب کرده بودم پس سعی کردم همان را در حالت باینری پیاده سازی کنم.

مشکلی که در همان ابتدا با آن برخورد کردم پیاده سازی سیگنال آنالوگ بود. راه حل من استفاده از نمونه گیری و پردازش سیگنال دیجیتال شد.

فرکانس سیگنال سینوسی حامل بدین شکل قابل محاسبه است:

یک سیگنال من باید به طور ثابت چند سیکل سینوسی داشته باشد.

اگر سرعت انتقال داده را بیشتر کنم باید طول سیگنال را کمتر کنم. اگر طول سیگنال کمتر شود برای ثابت نگه داشتن تعداد سیکل ها، باید فرکانس آن را افزایش دهم.

پس:

سرعت انتقال داده (هر قطعه سیگنال حامل اطلاعات بر ثانیه) متناسب است با فرکانس سیگنال

طول هر قطعه سیگنال (برحسب فاز) متناسب است با فرکانس سیگنال

## کانال و نویز

پس از بلوک مدولاسیون باید سیگنال حاصل را در کانال ارسال کنیم.

من نویز سفید گوسی با نسبت توان سیگنال به نویز مشخص را به سیگنال اضافه کردم.

مساله دیگر که در کانال پیاده سازی شده است، آزمایش همزمان نبودن ارسال و دریافت سیگنال در فرستنده و گیرنده است. یعنی ممکن است گیرنده روشن باشد و در چند ثانیه اول سیگنالی دریافت نکند. یا اینکه فرستنده در حال ارسال باشد اما گیرنده خاموش باشد و وقتی روشن شد از وسط سیگنال شروع به دریافت کند.

مشخصه دیگر کانال پهنای باند محدود است که برای پیاده سازی فیلتر کانال و فیلتر فرستنده نیاز به پردازش دیجیتال سیگنال ها احساس می شود. این بخش در سیستم هنوز پیاده سازی نشده است.

## دیمدولاسیون

سیگنال خروجی از کانال را به روش هم بستگی آشکارسازی کردم. با توجه به این که کانال من تنها یک نویز سفید روی سیگنال اعمال می کرد، حتی در SNR های خیلی پایین نیز پاسخ قابل قبولی از دیمدولاتور دریافت می کردم. خودم از این تعجب کردم! در SNR برابر ۱۰ - ، (سیگنال ۱۰ بار ضعیف تر از نویز) ، احتمال خطای من کمتر از  $10^{-3}$  بود!

باید کانال را دوباره بررسی کنم که آیا بخش اعمال نویز سفید اشکالی دارد. هم‌چنین تست‌های بعدی نشان خواهد داد که تا چه حد نویز در کار موثر است و تا چه حد پیاده‌سازی ما تاثیر دارد.

## مساله هم‌زمانی

بعد سیگنال را با یک شیفت فاز به دیمدولاتور دادم.

با تغییر دادن، نسبت BER تغییر پیدا می‌کرد چرا که من در سیستم، هم‌زمانی پالس ساعت ندارم.

وقتی پالس ساعت را دستی تنظیم کردم سیستم دوباره به خوبی کار کرد.

البته در ادامه سراغ تست‌های دیگری هم خواهم رفت. مثل ناکامل بودن تایمر دیمدولاتور. اما الان با توجه به این که در حال حاضر نمی‌توانم انجام دهم تست‌ها را متوقف می‌کنم.

**ادیت:** امروز توانستم سیستم را برای این کار بهبود دهم. درواقع برای حل مساله شروع اولیه پالس‌ها، آن را به مقادیر مختلف شیفت می‌دهم و سطح خروجی دیمدولاتور را بررسی می‌کنم. ماکسیمم را انتخاب می‌کنم و همه سیگنال را به همان اندازه شیفت می‌دهم.

در صورتی که بخشی از اول سیگنال از دست برود، این روش کار می‌کند. اما در صورتی که سیگنال دیرتر دریافت شود یعنی بخشی از اول سیگنال خالی باشد، این روش کارا نیست. البته در این صورت می‌توان سیگنال دریافتی را بررسی کرد و اگر بزرگ‌تر از یک حدی بود آن را آشکارسازی کرد. ولی مشکل این روش آن است که از همان اول توان بهینه سیگنال را نمی‌دانیم.

## کدینگ دیفرانسیلی

با توجه به استفاده از مدولاسیون PSK-2 در حدود خاصی از تغییر، همه بیت‌ها معکوس می‌شد. یعنی ۱ ها ۰ و ۰ ها ۱ می‌شد. در این حالت ممکن بود از بین ۱۰۰۰ بیت، ۹۹۹ بیت خطا دریافت شود.

من بلوک کدینگ دیفرانسیلی را قبل از مدولاسیون اضافه کردم.

هم‌چنین دیکدینگ را هم بعد از مدولاسیون.

این مشکل برطرف شد! چرا که دیگر فاز مهم نبود بلکه تغییرات فاز مهم بود.

رابطه من برای محاسبه کیفیت سیستم:

درصد کیفیت = نرمال شده (تعداد بیت‌های خطا - نصف بیت‌های دریافتی)

پس وقتی نصف بیت‌ها خطا دریافت شود کیفیت سیستم من ۰٪ است و اگر بیت خطا نداشته باشیم یا همه بیت‌ها خطا دریافت شود، کیفیت سیستم من ۱۰۰٪ است.

## پردازش سیگنال دیجیتال

همانطور که گفتم، برای پیاده‌سازی بسیاری از بخش‌های سیگنال نیاز به پردازش سیگنال دیجیتال دارم. پیاده‌سازی کامل کانال، فیلترهای فرستنده و فیلتر گیرنده، همزمان‌سازی اتوماتیک یا نیمه اتوماتیک پالس ساعت و غیره.

توانایی زیادی در پیاده‌سازی آن از ابتدا نداشتم هرچند تجربیاتی طی پروژه کسب کردم اما راهی طولانی در پیش دارم.

## کدینگ منبع

داده‌های ارسالی روی کانال باید کد شوند!

اگر طول هر کد ما بیشتر از تعداد بیت‌های هر سیگنال باشد، باید هر قطعه کد را در چند سیگنال ارسال کرد. حال اگر سیستم یکی از سیگنال‌ها را از دست بدهد، مشکل در دیکد کردن به وجود می‌آید. چرا که بیت‌های بالایی یک عدد، جایگزین بیت‌های پایینی یک عدد دیگر می‌شود.

من هنوز در این بخش مشکل دارم.

## رمزنگاری

از همان ابتدا تصمیم بر این بود که یک رمزنگاری ساده پیاده‌سازی شود.

هدف تست کردن این است که ببینیم آیا واقعا می‌توان در این سیستم (که بیشتر برای انتقال است) رمزنگاری را پیاده‌سازی کرد؟

رمزنگاری سزار **Caesar cipher** را تست کردم که در الفبای ۲ حرفی، همان معکوس کردن بیت‌ها است.

هرچند امنیت این رمزنگاری در حد گذاشتن یک تکه چوب جارو به جای کلون در است، ولی در هر حال کار کرد!

## کدینگ کانال و کدینگ خط

در ادامه سعی کردم خطاها را با بیت‌های زوجیت تصحیح کنم.

پس سعی کردم الگوریتمی برای این کار پیدا کنم. از طرفی چون می‌خواستم **line coding** هم داشته باشم ترجیح دادم الگوریتم خودم را طراحی کنم. الگوریتم من ۴ بیت را می‌گرفت و ۸ بیت را خروجی می‌داد. فازهای یکسان متوالی ارسال نمی‌کرد و می‌توانست ۲ بیت خطا را تشخیص دهد و ۱ بیت خطا را تصحیح کند.

اما...!

## دیکدینگ کانال

هنگام دیکدینگ کانال در بخش تصحیح خطا متوجه چیزی شدم که انتظار آن را نداشتم. خطاها اغلب کنار هم اتفاق می‌افتادند. یعنی در بین ۱۰۰۰ بیت داده، من ۲ بیت متوالی خطا داشتم!

برخلاف انتظارم توزیع خطا کاملاً یکنواخت نبود.

در نهایت به جای کدینگ اصلی یک کدینگ Parity قرار دادم. چرا که حس کردم بیهوده است!

با اضافه کردن ۱ بیت زوجیت در انتهای هر بلوک از بیت‌ها می‌توان محل خطا را تشخیص داد. سپس اگر با روش‌هایی مثل hash کردن به خطا خوردیم، می‌توانیم بیت‌ها را تشخیص دهیم. برای این کار البته که لازم است بین هر چند بلوک یک بلوک برای مقدار hash نیز قرار داد.

در حال حاضر از تصحیح خطا صرف نظر کردم و به تشخیص خطا قانع شدم.

## نتیجه‌گیری از این کار

- این کار به دانش زیادی در حوزه پردازش سیگنال‌های دیجیتال احتیاج دارد.
- برای مساله گم شدن بیت‌ها در هنگام دیکد کردن باید راه حلی پیدا کنم.
- اگر مساله گم شدن بیت‌ها برطرف شود می‌توانم آزمایش‌های بیشتری درباره رمزنگاری انجام دهم.

- در پیاده‌سازی پروژه نیز با توجه به زبان‌هایی که کار کرده‌ام مثل پایتون و سی، زبان متلب تفاوت‌هایی با آن‌ها دارد که کار من را سخت می‌کند. هم‌زمان با انجام پروژه در حال دست و پنجه نرم کردن با متلب هم هستم! متلب واقعاً زیبا نیست.
- توزیع بیت‌های خطا برخلاف آنچه که تصور می‌کردم کاملاً یکنواخت نبود! باید نویز روی کانال را هم بررسی کنم.
- در طی این کار وقتی می‌خواستم بلوک جدید را آزمایش کنم باید آن را در شرایط خوب آزمایش می‌کردم. مثلاً وقتی می‌خواستم ببینم بخش هم‌زمانی اولیه ضرب کننده سیگنال خوب کار می‌کند، باید نویز را کاهش می‌دادم یا بلوک‌های کدینگ را حذف می‌کردم تا مطمئن باشم خطایی که رخ می‌دهد به خاطر این بلوک است نه بلوک‌های دیگر.
- با توجه به تجربیاتم میدانم که مساله هم‌زمانی پالس ساعت در فرستنده و گیرنده اهمیت زیادی دارد. هرچند هنوز آزمونی برای آن و راهکاری برای حل این مساله در سیستم تدارک ندیده‌ام.

در نهایت در این بازه زمانی از پیشرفت خودم در این پروژه راضی بودم.