



# طراحی و پیاده سازی سیستم OFDM با متلب

گزارش پروژه پایانی مخابرات بی سیم

سهند خوشدل - ۸۱۰۱۹۶۶۰۷

استاد: دکتر صباغیان - تیر ۱۴۰۰

## فهرست مطالب

---

۱- مقدمه ای بر سیستم OFDM	۲
۲- توضیح بلوک های استفاده شده	۲
۲-۱. فرستنده	۲
۲-۲. کانال	۸
۲-۳. گیرنده	۸
۳- نتایج و پاسخ به خواسته ها	۱۱

## ۱. مقدمه ای بر سیستم OFDM

هدف این پروژه طراحی یک سیستم OFDM به کمک پیاده سازی بلوک به بلوک آن بر اسا بلوک دیاگرام داده شده در دستور کار، مقایسه کانال های AWGN و رایلی، تاثیر وجود همسان ساز بر احتمال خطای آشکارسازی بیت (BER) و در نهایت آشکار سازی محدوده پیام OFDM به کمک سیگنال های معین شده Pilot، می باشد.

کلیت کار این سیستم بر مبنای تقسیم بندی یک پیام به چندین فریم و تقسیم کردن هر فریم به بخش هایی که هر یک بر روی یک زیر حامل قرار می گیرد است، بنا نهاده شده. به همین دلیل سیستم OFDM ، زیرمجموعه سیستم های multicarrier می باشد. ویژگی اصلی این سیستم تعیین کردن فرکانس حامل ها به ترتیبی است که طیف فیلتر هر یک از زیرکانال ها در حوره زمان در محل فرکانس مرکزی زیرکانال مجاور از صفر عبور کند. برای تحقق شرط فوق زیر حامل ها باید حداقل به میزان  $1/T$  از هم فاصله داشته باشند. به همین دلیل هم به این سیستم Orthogonal Frequency Division Multiplexing یا به اختصار OFDM گفته می شود.

## ۲. توضیح عملکرد بلوک های استفاده شده

بلوک های ارائه شده در بلوک دیاگرام آمده در صورت پروژه، نسبت به پیاده سازی واقعی سیستم OFDM برخی ماژول های قسمت آنالوگ را ندارند (مثل مبدل های ADC, DAC و اسیلاتور هایی که وظیفه تولید فرکانس های مرکزی را دارند). با این وجود، اکثر این سیستم به صورت تماما دیجیتال تقریبا کامل پیاده سازی شده و در این بخش به شرح عملکرد هر کدام از بلوک های استفاده شده در فرستنده و گیرنده و همچنین مدلسازی کانال های AWGN و Rayleigh خواهیم پرداخت.

### ۲-۱. فرستنده

در ابتدا داده ها به صورت تصادفی با استفاده از تابع randi تولید می شوند و به فرم یک رشته بیت در می آیند که هر دو بیت آن کنار هم قرار داده شده (در اصل به صورت یک ماتریس ذخیره می شوند). این شیوه ذخیره سازی آن هارا برای تبدیل شدن به سمبل های متناظر با فضای سیگنال QPSK آماده می کند.

### 2-1-1. QPSK Modulator

این ماژول وظیفه تصویر کردن زوج بیت ها به اعداد decimal در محدوده 0-3 را دارد. سمبل های فضای تصویر در این نگاشت بر اساس Gray Coding کنار هم قرار میگیرند تا به ازای خطای سمبل معین، خطای بیت کمینه شود. Gray Coding مربوط به فضای سیگنال ۴ تایی در مدولاسیون QPSK به صورت زیر است.

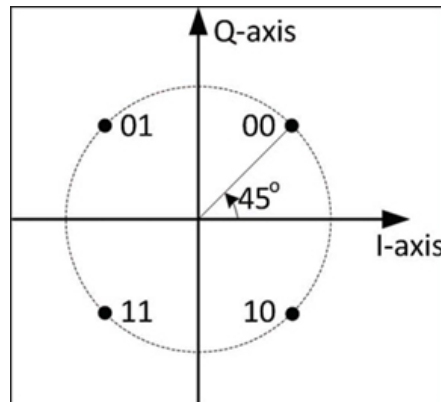


Figure 1. Constellation View of QPSK

شکل ۱-۲- فضای سیگنال در مدولاسیون QPSK

در صورت پروژه عنوان شده که سائز ورودی داده شده به سیستم یک پیام با طول  $10^7$  بیت است که با توجه به اینکه مدولاسیون ۴ تایی استفاده می کنیم و هر دو بیت به صورت یک سمبل در فضای مختلط نشان داده می شوند، طول پیام برابر با  $5 \times 10^6$  سمبل خواهد بود.

پس از آنکه پیام ورودی در این ماژول به صورت یک ماتریس با ابعاد  $500000 \times 2$  درآمد. هر یک از ستون های این ماتریس نشان دهنده زوج بیت هایی هستند که در پیام اصلی کنار هم قرار داده شده. این زوج بیت ها پس از عبور از این ماژول با توجه به Gray Coding تخصیص داده شده، به عدد متناظر با خود نگاشت می شوند.

بنابراین خروجی این ماژول یک بردار به طول  $5 \times 10^6$  خواهد بود که شامل اعداد 0-3 است.

### 2-1-2. Frame Divider

از آنجایی که ارسال تمام پیام به صورت یکجا مناسب نیست ( به دلیل مسائل ارسال دوباره در اثر خطا و ...) در سیستم OFDM، هر پیام را به چندین فریم تقسیم می کنند و فریم ها را جدا گانه ارسال می کنند ( هر چند این جا فقط ساختار داده عوض می شود و به منظور راحتی همه فریم ها با هم ارسال می شود).

تعداد فریم ها را باید بر اساس تعداد سمبل هایی که در هر فریم قرار دارند (ظرفیت یک فریم) محاسبه کنیم. این تعداد با توجه به اینکه چند زیر حامل در اختیار داریم (nc) و هر زیرحامل ظرفیت چند سمبل را دارد (sfc). حاصلضرب این دو عدد ظرفیت کلی هر فریم را بر اساس سمبل نشان خواهد داد.

$$frame\ size = sfc \times n_c$$

با جایگذاری مقادیر داده شده در صورت پروژه ظرفیت هر فریم مطابق محاسبات زیر بدست می آید:

$$frame\ size = 21 \times 400 = 8400 \text{ (سمبل)}$$

بدین ترتیب با جدا کردن هر دسته ۸۴۰۰ تایی از بردار خروجی ماژول قبل، ماژول *Frame divider*، ۵۹۵ فریم کامل و یک فریم ناقص را از هم جدا خواهد کرد. به انتهای فریم آخر صفر اضافه میکنیم تا ۵۹۶ فریم داده برای بلوک های بعدی خروجی بدهیم.

$$Total\ frames = ceil \left[ \frac{Total\ symbols}{Frame\ size} \right] = 596$$

خروجی این ماژول تعداد فریم ها یا سائز هر کدام خواهد بود ( کافی است خروجی معیاری باشد تا ماژول بعدی که ساختار داده رو عوض می کند با توجه به آن معیار داده ها را از هم جدا کند).

### 2-1-3. Serial to Parallel

در نهایت فرم ذخیره سازی این فریم ها می تواند به دو صورت باشد.

۱- هر فریم به صورت یک ماتریس ۲۱ در ۴۰۰ ذخیره شود و ماتریس ها کنار هم قرار گرفته و ارسال شوند

۲- هر فریم به عنوان یک لایه میان دو فریم مجاور خود به صورت یک ماتریسی قرار بگیرد تا کل داده به فرم یک تانسور در بیاید که ابعاد آن ۲۱ در ۴۰۰ در ۵۹۶ است

در این پیاده سازی از روش دوم برای ذخیره سازی داده به دلیل فرم مرتب تر آن و مزایایی که ساختار داده تانسوری برای *reshape* کردن در بلوک های بعدی دارد، استفاده شده. به همین دلیل هم به این فرمت ذخیره سازی *parallel* کردن فریم ها گفته می شود.

### 2-1-4. DPSK Modulation

برای آنکه تغییرات فاز در یک مدولاسیون فاز محور مثل BPSK به حداقل برسد و ابهام در آشکار سازی فاز به حداقل در صورت خطای فاز به حداقل برسد، معمولاً از مدولاسیون های differential مثل مدولاسیون

$$y_i = y_{i-1} \oplus x_i,$$

DPSK استفاده می کنند. این مدولاسیون بر مبنای یک رابطه بازگشتی بین سمبل های ارسال شده در هر زیرحامل (ستون) عمل میکند که به شرح زیر است:

بدین ترتیب، به نوعی تابع فاصله سمبل های متوالی در دنباله ورودی خواهد بود. هر رابطه بازگشتی نیاز به یک شرط اولیه (initialization) برای برقرار بودن دارد. که در اینجا یک سمبل به ابتدای هر ستون (به عنوان سمبل اولیه برای رشته سمبل های هر زیرحامل) اضافه می شود که این مقدار سمبل تاثیری در بازیابی نهایی ندارد بنابراین آن را به صورت رندوم انتخاب میکنیم.

بنابراین مثل آن است که یک سطر رندوم به عنوان reference row، به ابتدای هر لایه (ماتریس، فریم) از این تانسور اضافه کرده ایم یا به طور معادل یک طبقه به بالای این تانسور به عنوان طبقه reference برای مدولاسیون DPSK اضافه کرده ایم.

خروجی این ماژول با توجه به توضیحات فوق یک تانسور با ابعاد ۲۲ در ۴۰۰ در ۵۹۶ خواهد بود.

## 2-1-5. IFFT bins allocation

سیستم OFDM، بر مبنای ارسال سمبل ها در حوزه زمان طراحی شده اما از آنجایی که برخی فیلتر ها مثل فیلتر های Equalizer در حوزه فرکانس عملکرد بهتری دارند باید ماژول هایی قبل و بعد ماژول های مربوط به حوزه زمان در حوزه فرکانس قرار بگیرند و مدولاسیون اولیه و ساختار بندی پیام در آن ها صورت بگیرد. ماژولی هم برای تبدیل سمبل ها از حوزه فرکانس به حوزه زمان مورد نیاز است که همان ماژول IFFT است که تعداد نقاط آن با دقت IFFT گرفته شده متناسب است. ماژول متناظر که در گیرنده سمبل ها را به حوزه فرکانس بر میگرداند ماژول FFT است. در این پروژه تعداد نقاط این دو ماژول ۱۰۲۴ نقطه در نظر گرفته شده است. با توجه به اینکه هر زیرحامل نماینده یک فرکانس مرکزی در بازه فرکانسی مربوط به خود است IFFT و FFT باید به صورت سطری گرفته شوند. فلذا تعداد ستون های ماتریس ورودی به این ماژول و ماتریس خروجی از ماژول FFT باید ۱۰۲۴ تا باشد. برای آنکه تغییری در نتیجه رخ ندهد و خروجی هم در حوزه زمان به صورت حقیقی باشد، بلوک مزدوج هر فریم را کنار آن قرار می دهیم و سایر ستون ها را به صورت مساوی با صفر پر کرده و در سمت چپ، وسط و راست قرار می دهیم. شماره ستون هایی که باید اصل فریم و مزدوج آن، از آنجا شروع بشوند به صورت زیر محاسبه می شود:

$$1024 - 2 \times 400 = 224 = \text{تعداد ستون های تمام صفر}$$

$$\text{تعداد ستون های راست و چپ} = \frac{225}{3} = 75$$

۷۶ ستون تمام صفر هم برای قراردادن در وسط باقی می ماند. بنابر این از ستون شماره ۷۶ باید فریم اصلی قرار بگیرد و از ستون ۵۵۰ هم ماتریس مزدوج را قرار دهیم.

## 2-1-6. IFFT (1024-point)

این ماژول از تانسور ورودی رو هر سطر IFFT می گیرد و در خروجی در سطر های متناظر قرار می دهد. برای سرعت بیشتر به جای آنکه IFFT را به صورت for روی تمام سطر ها پیاده سازی کنیم، آن را مستقیماً بر روی تانسور و نسبت به بعد اول (سطر ها) اعمال می کنیم.

## 2-1-7. CP Addition

برای آنکه بتوان از خواص کانولوشنی سیستم در حوزه زمان به خاصیت ضربی در حوزه فرکانس رسید باید کانولوشن انجام شده معادل یک کانولوشن حلقوی باشد. و شرط مذکور در صورتی که یکی از سیگنال های کانوالو شده به صورت حلقوی باشد (بخش انتهایی آن در ابتدا تکرار شده باشد و یا برعکس)، خود به خود برقرار می شود.

برای آنکه چنین فضایی برای قرار گیری بخشی از سیگنال در ابتدای فریم ها محیا شود در واقعیت **guard band** هایی در محور زمان در نظر گرفته می شود که در آن داده واقعی ارسال نشود و همین بخش پیشوندی یا به صورت **zero prefix** و یا به صورت **cyclic prefix** در این بازه ارسال شود. هدف اصلی قرار دادن **guard band** ها اما همانطور که در طول درس به آن اشاره شده جلوگیری از تداخل بین سمبل های زیر حامل های یکسان از بلوک های مجاور (**inter-block interference**) و همچنین جلوگیری از تشعشع خارج از باند (**out of band radiation**)

با توجه به **CP ratio** تعیین شده چندین ستون باید به ابتدای هر فریم (ماتریس) اضافه بشود که در اینجا **cp ratio** برابر با ۱/۴ است، چون ۲۵ درصد انتهایی هر ردیف باید به ابتدای آن اضافه می شود و تانسور خروجی این ماژول ابعادی به شکل ۲۲ در ۱۲۸۰ در ۵۹۶ خواهد داشت. (۲۵۶ ستون به ابتدای هر فریم اضافه شده است)

می توان **CP ratio** را که مستقیماً نسبت **overhead** سیستم نیز هست، به صورت یک متغیر تعریف کرد و آن را کم و زیاد کرد.

### 2-1-8. Parallel to Serial

پس از آنکه عملیات IFFT روی داده ها انجام شد و پیشوند گردشی ابتدای هر فریم اضافه شد، با توجه به اینکه فریم ها را قرار است پشت سر هم ارسال کنیم. ابتدا هر ماتریس (فریم) را به صورت یک بردار در می آوریم. لازم به ذکر است که باید ستون های این ماتریس پشت هم قرار بگیرند چرا که تمام بخش CP در صورت پروژکشن کنار هم قرار دارد (با عبارت *frame guard*) مشخص شده است.

هر یک از فریم ها پس از عبور از این ماژول به صورت بردار هایی خواهند بود که به شکل سطری زیر هم چیده می شوند. (تبدیل تانسور به ماتریس)

### 2-1-9. Cascade Frames

در این مرحله، همه ۵۹۶ فریم ای که هر کدام صورت بردار در آمده اند و ابتدای هر کدام از آن ها را پیشوند گردشی تشکیل داده، پشت سر هم قرار گرفته و در یک بردار آماده ارسال قرار می گیرند. (تبدیل ماتریس به بردار)

علاوه بر این این ماژول مسئول اضافه کردن دو header به ابتدا و انتهای داده ها است. طول header ها ۸ برابر طول هر فریم (البته اصل فریم و نه به همراه *frame guard*) خواهد بود که معادل است با:

$$\text{طول header بر حسب سمبل} = 22 \times 1024 \times 8 = 180'224$$

محتوای **هدر** ها برای بخش امتیازی می تواند یک سیگنال پایلوت دانسته شده باشد (البته فقط ابتدای هدر را اشغال کند)

### 2-1-10. Clipper

به دلیل استفاده از تقویت کننده هایی که در دامنه های نسبتا بالا وارد ناحیه اشباع می شوند، سیگنال ارسالی قبل از قرار گرفتن روی کانال دچار اعوجاج غیر خطی نسبت به دامنه های مختلف می شود. انی اثر غیر خطی معمولا به صورت clip کردن دامنه های موجود در سیگنال در صورتی که از یک آستانه خاص بالاتر باشند، اتفاق می افتد. این آستانه به صورت نسبی با توجه به ماکزیمم دامنه سمبل های ارسالی در صورت پروژکشن 3dB داده شده که به صورت یک متغیر در انتهای تابع transmitter بکار رفته است و می توان آن را عوض کرد.



## ۲-۲. کانال

برای مدل کردن کانال، یا از یک نویز ساده AWGN و یا از ترکیب نویز AWGN در ابتدای گیرنده و ضرب کردن یک تصادفی رایلی، قبل از آن استفاده می کنیم. متغیر تصادفی رایلی با واریانس ۱ را به عنوان ضریب fading کانال در بردار متناظر با ضرایب fading (همان بردار  $H_k$ ) ذخیره می کنیم و در سیگنالی که از خروجی فرستنده بیرون آمده ضرب می کنیم.

## ۲-۳. گیرنده

ماژول های به کار رفته در گیرنده، به این صورت چیده شده اند تا عکس عمل ماژول های فرستنده را به ترتیب بر روی سیگنال نویزی و تضعیف شده انجام بدهند.

### 2-3-1. Frames Detection

ابتدا بردار سمبل های دریافت شده به فریم هایی که در فرستنده ساخته شده بود تقسیم می شود (تعداد فریم ها که ۵۹۶ تا هستند در گیرنده هم مشخص است (بر اساس  $sfc, nc$ ) و بدین ترتیب، هر کدام از قسمت های جدا شده به عنوان یک فریم زیر دیگری قرار میگیرند (تقسیم بندی سطری). بنابراین خروجی این ماژول یک ماتریس ۵۹۶ در  $(1280 \times 22)$  می باشد.

لازم به ذکر است که قبل از جدا سازی باید به اندازه طول هدر سمبل ها را دور بریزیم. طول هدر مشابه محاسبه ای که در فرستنده گفته شد می تواند محاسبه شود و یا به صورت یک عدد مستقیم به گیرنده نیز داده شود.

### 2-3-2. Serial to Parallel

سطر های ماتریس ورودی به این ماژول هر کدام یک فرمی هستند و باید دوباره به صورت ماتریس در بیایند تا عملیات بعدی مثل حذف پیشوند و FFT روی آن به صورت سطری (همانطور که ماژول های متناظر در گیرنده عمل کردند) اعمال شود تا بازیابی هر سمبل به درستی انجام شود (در نهایت مقایسه هر سمبل با تخمین خود آن سمبل انجام شود و نه یک سمبل دیگر)

بنابراین این ماژول هر یک از سطر ها را ۲۲ دسته تقسیم می کند ( $SFC + 1(ref)$ ) و انی دسته ها را به صورت یک ماتریس پشت هم می چیند. علاوه بر این هر یک از ماتریس های بدست آمده که معرف یک فریم هستند را به صورت لایه لایه پشت هم قرار می دهد تا داده شکل تانسور به خود بگیرد (تبدیل ماتریس به تانسور)

### 2-3-3. CP Removal

در این قسمت ۲۵۶ سمبل اضافه شده به هر سطر ( در مجموع ۲۵۶ ستون اولیه) هر فریم کنار گذاشته می شوند و اصل فریم ها به همان شکل تانسوری در ادامه مسیر قرار خواهند گرفت. بنابراین خروجی این ماژول یک تانسور با ابعاد ۲۲ در ۱۰۲۴ در ۵۹۶ خواهد بود.

### 2-3-4. FFT

در این مرحله از فریم ها به صورت سطری تبدیل فوریه ۱۰۲۴ نقطه ای گرفته می شود و خروجی برای جداسازی اصل فریم به ماژول بعدی داده می شود.

### 2-3-5. Extract Carriers from FFT bins

ستون هایی که اصل فریم را تشکیل می دادند با توجه به شیوه قرار گیری شان در ماژول متناظر در گیرنده جدا می شوند. (از ستون ۷۶ تا ۴۷۵)

### 2-3-6. DPSK Demodulation

عملیات مهمی که باید در این ماژول در ابتدا انجام شود آشکار سازی سمبل ها است. چرا که ورودی DPSK Demodulation همان سمبل های مختلطی را می شناسد که ماژول متناظر آن تولید کرده و یک سیگنال نویزی طبیعتاً در فضای سیگنال فاصله مشخصی از این نقاط منظومه (constellation points) خواهد داشت. پس یا باید فاصله ها را تا نقاط حساب کنیم و نقطه ای که کمترین فاصله را با تصویر سیگنال نویزی در فضای سیگنال (فضای مختلط) خواهد داشت به عنوان تخمین آشکار ساز انتخاب می شود. و یا باید با استفاده از تعیین نواحی تصمیم گیری این آشکار سازی را انجام داد که برای مدولاسیون های DPSK اصولی ترین کار تبدیل به مختصات قطبی و تصمیم گیری بر اساس محدوده ای است که فاز در آن قرار می گیرد. سپس در این ماژول عکس مدولاسیون DPSK انجام می شود. هر ستون که یک سمبل reference در بالای خود به همراه داشت به همان شکل XOR دو سمبل متوالی را با هم جمع می کند و حاصل را جایگزین می کند. بدین ترتیب تانسور خروجی ابعاد ۲۱ در ۴۰۰ در ۵۹۶ خواهد داشت. بدین ترتیب ردیف reference اضافه شده برداشته می شود.

### 2-3-7. Parallel to Serial

یکبار دیگر در این مرحله تانسور داده را به ماتریس تبدیل می کنیم. به عبارت دیگر هر کدام از فریم هارا flatten می کنیم و زیر هم در ردیف های یک ماتریس با ابعاد ۵۹۶ در ۸۴۰۰ قرار می دهیم.

### 2-3-8. Cascade Frames

برای آنکه بتوان تخمین آشکار ساز را به صورت نظیر به نظیر با سمبل های اصلی مقایسه شوند باید تمام به فرم یک بردار درآید ، بنابراین یکبار دیگر، این بار در گیرنده ردیف (فریم) های ماتریس داده را پشت هم می چینیم. خروجی ماژول یک رشته سمبل با طول  $5 \times 10^6$  خواهد بود.

### 2-3-9. QPSK Demodulation

در این مرحله سمبل هایی که به صورت decimal تعریف شده اند را مطابق همان gray coding به زوج بیت های اولیه برگردانیم. بردار خروجی یک رشته بیت با همان طول  $10^7$  خواهد بود که برای مقایسه و محاسبه BER آماده است.

### 2-3-10. BER Calculation

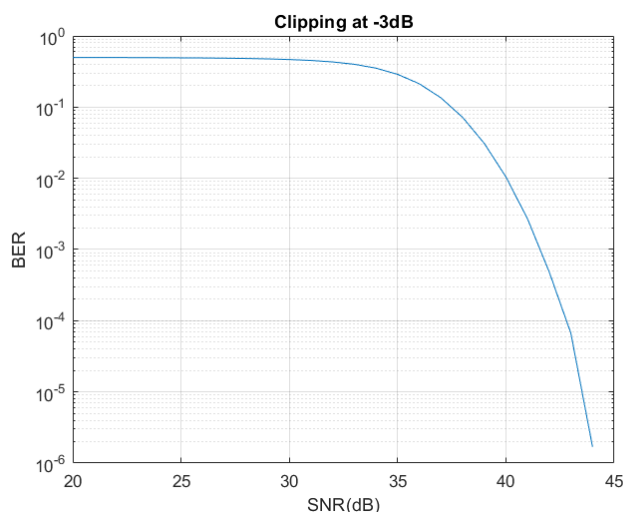
در قسمت آخر کافی است تا بردار evaluation را تشکیل دهیم. برداری که هر عنصر آن در صورت برابر عناصر متناظر بردار تخمین و بردار رشته بیت اولیه ۱ و در غیر این صورت ۰ خواهد بود. در نهایت مجموع تعداد ۰ ها، تعداد آشکار سازی های اشتباه بیت را نشان می دهد و اگر این مقدار را بر طول کل رشته بیت تقسیم کنیم BER متناظر با رشته بیت تولیدی را بدست خواهیم آورد. بهتر است طول رشته بیت تولیدی را همانند اینجا نسبتا بزرگ بگیریم تا از نوسانات نتایج جلوگیری شود (stochasticity). همچنین می توان برای طول های کوچک میانگین BER چندین epoch را گزارش داد تا نتایج قابل اعتماد تر و قابل تحلیل تر باشند.

### ۳. نتایج و پاسخ به خواسته ها

الف) احتمال خطا برای  $SNR = 20dB$  تقریباً برابر با ۵۰ درصد و عملاً معادل تشخیص کور (نویزی) است و وضعیت مناسبی ندارد چون نویز تولید شده به اندازه ای کوچک نیست که باعث شود سیگنال در ناحیه متناظر با خود باقی بماند.

$$BER = 0.4981$$

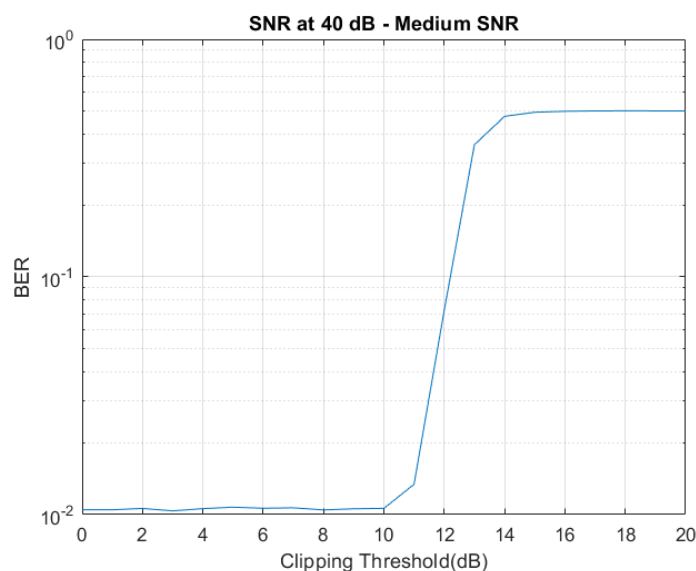
ب) اگر نمودار احتمال خطای بیت بر حسب  $SNR$  را برای  $SNR$  های بالای 20 dB را رسم کنیم، رفتار نزولی کاملاً مشخص است.



ج) اگر نمودار احتمال خطای بیت بر حسب سطح clipping که ناشی از اثر غیر خطی (اشباع شونده) تقویت کننده های توان فرستنده است را رسم کنیم، یک رفتار صعودی مشاهده میکنیم که مطابق انتظار هم هست چرا که با افزایش اندازه این clipping (یا به عبارتی کاهش ترشولد) سمبل های با دامنه های پایین تر هم دچار اعوجاج می شوند و از آنجا که تعداد کل سمبل هایی که دار اعوجاج شده اند و سپس روی کانال قرار گرفته اند، در کل بیشتر شده، احتمال آشکار سازی غلط و خطای بیت هم بالاتر خواهد رفت.

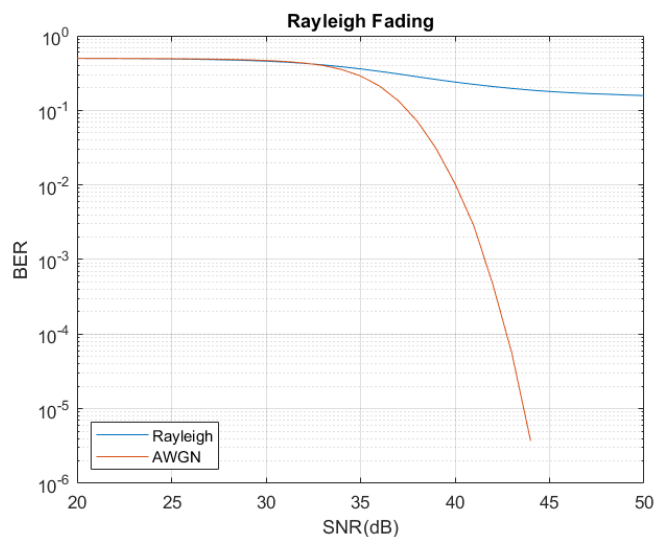
\* نکته بسیار مهم این قسمت آن است که اگر این رفتار را برای  $SNR$  های کوچک مثل ۲۰ دسی بل که بدون clipping هم وضعیت بدی دارند و یا برای  $SNR$  های بزرگ مثل ۷۰ دسی بل که عملاً سطح نویز بسیار پایین است (نسبت به سیگنال) رسم کنیم، متوجه میشویم که حساسیت BER نسبت به آستانه clipping بسیار کاهش می یابد و رفتار افزایشی به شکل واضحی قابل مشاهده نخواهد بود.

\* بنابراین مهمترین نکته این بخش انتخاب یک SNR متوسط برای افزایش حساسیت BER به پارامتر دوم که همان Clipping Threshold است می باشد. برای اینکه رفتار افزایشی به طرز مناسبی دیده شود من SNR = 40dB که بدون clipping عملکرد نسبتاً متوسطی داشت را انتخاب کردم.



#### (د) مدلسازی کانال رایلی:

برای مدلسازی این نوع کانال یک متغیر تصادفی رایلی با واریانس ۱ تولید کرده و از آن نمونه می گیریم. به اندازه تعداد سمبل های ارسالی این کار را تکرار می کنیم و آن را داخل یک بردار که معرف پاسخ ضربه کانال است ذخیره می کنیم. (کانال را برای سادگی هم طول با اصل سیگنال در نظر میگیریم)

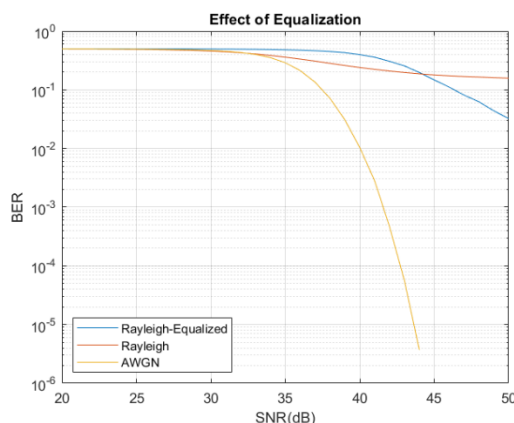


همانطور که انتظار می رفت خطای بیت متناظر با کانال رایلی بالاتر از کانال AWGN است. همچنین به نظر می رسد که خطای کانال رایلی به یک حد پایین می رسد و با افزایش SNR از یک جا بعد کاهش محسوسی پیدا نمی کند.

- نکته مهم آن است که ابتدا ضریب رایلی را ضرب کرده و سپس نویز AWGN را به صورت یک فرایند ایستادن در ابتدای گیرنده (انتهای کانال) مدل کنیم و نه برعکس.

#### ه) مدلسازی همسان ساز MMSE:

با توجه به رابطه ای که در درس برای این محاسبه ضرایب این همسان ساز گفته شده ضرایب آن را در ابتدای گیرنده در یک بخش مربوط به همسان ساز، محاسبه میکنیم و سپس در بردار سمبل هایی که روی کانال قرار گرفته نظیر به نظیر ضرب میکنیم.



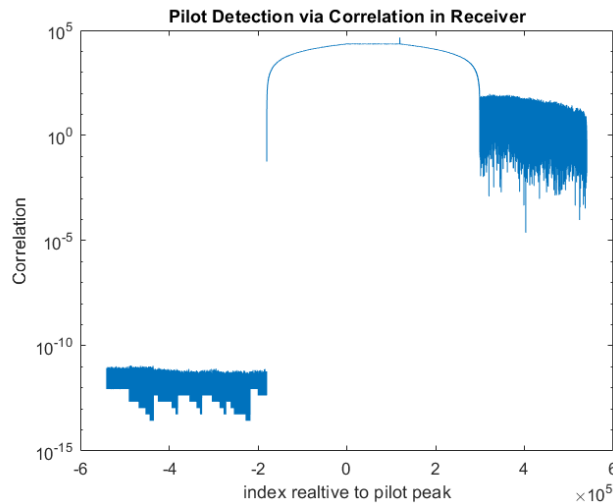
همانطور که مشخص است احتمال خطای بیت برای گیرنده دارای همسان ساز MMSE که ضرایب آن در SNR های بالا به عکس Hk تبدیل می شود، نسبت به SNR کاهشی است و دیگر آن حد پایینی که در محوشدگی رایلی به نظر می رسد، مشاهده نمی شود.

همچنین همانطور که انتظار میرفت عملکرد این حالت چیزی بین کانال AWGN و کانال رایلی است.

#### و) بخش امتیازی: آشکار سازی سیگنال پایلوت

سیگنال پایلوت به عنوان بخشی از ابتدای هدر های ابتدایی و انتهایی در پیام OFDM قرار میگیرند و با توجه به مشخص بودن این سیگنال ها در گیرنده (به صورت پیش فرض و یا پاس دادن آن به عنوان آرگومان) گیرنده دائما در حال کورلیشن گرفتن بین سیگنال دریافتی و سیگنال پایلوت است. اگر پیامی ارسال نشود این کورلیشن به دلیل دریافت نویز رندوم به صورت یک مقدار کم و غیر قابل توجه و نسبتا ثابت باقی خواهد ماند.

اما در صورتی که سیگنالی فرستاده شود که ابتدای آن با سیگنال پایلوت مشخص شده است با توجه به پیک زدن این کورلیشن می توان اندیسی که کورلیشن در آن پیک زده شده به عنوان ابتدای هدر ( و عملا ابتدای پیام) مشخص کرد. همین موضوع برای انتهای سیگنال نیز صادق است.



نمودار احتمال خطا مشابه قسمت الف و ب خواهد بود چرا که ابتدای سیگنال بدون خطای زمانی تشخیص داده می شود و سمبل ها نظیر به نظیر مقایسه می شوند.

