

به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

# تمرین کامپیوتری

پیاده سازی سیستم OFDM

مخابرات بی سیم

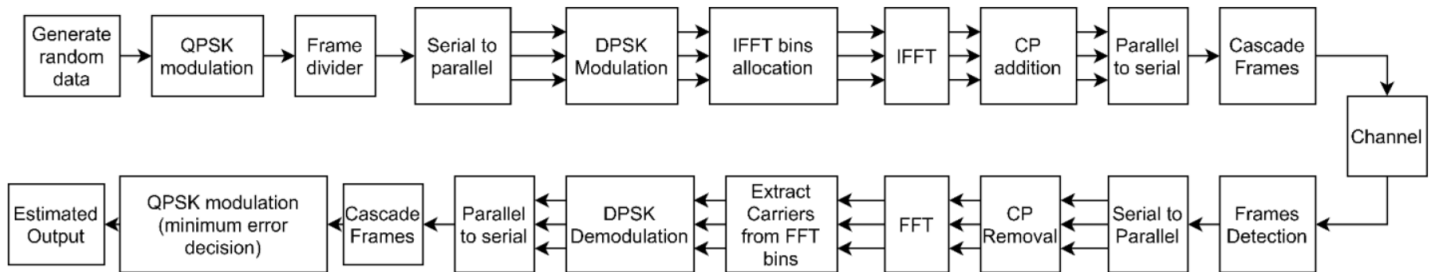
نیم سال دوم 99-00

دکتر صباغیان

علی ساعی زاده | 810196477

3	مقدمه و توضیحات پیاده سازی
3	فرستنده
3	Initialization & Generate Random Bits
3	& QPSK Modulation Generating Symbols
3	Frame Divider & Serial to Parallel
4	Adding Reference Row
4	DPSK Modulation
4	QPSK Signal Replacement
4	IFFT bins Allocation
5	IFFT
5	Adding CP
5	& Cascade Frames Parallel to Serial
5	کانال
5	Clipping
5	AWGN Channel
6	گیرنده
6	Serial to Parallel & Frame Detection
6	CP Removal
6	FFT
6	Extract Carriers from FFT bins
6	DPSK Demodulation & QPSK Demodulation (minimum error decision)
6	Parallel to Serial & Cascade Frames
6	Estimated Output & QPSK Demodulation
6	BER Calculations
6	BER Plot
7	خواسته های پروژه
7	الف.
8	ب.
9	ج.
10	د.
11	ه.
12	و. (امتیازی)

## مقدمه و توضیحات پیاده سازی



شکل 1: دیاگرام بلوکی سیستم OFDM

مقدار دهی اولیه در فایل `wlc_init.m` موجود است.

## فرستنده

### Initialization & Generate Random Bits

با استفاده از معیار `sfc` که تعداد سمبل ها در هر فریم را مشخص می کند داریم؛

$$sfc = \left\lceil \frac{2^{13}}{n_c} \right\rceil$$

این معیار نشان دهنده این است در هر حامل، در هر فریم چند سمبل باید قرار گیرد. سپس سمبل ها در هر فریم به هر زیر حامل اختصاص داده می شود.

برای راحتی پیاده سازی در فضای نرم افزار متلب از ماتریس سه بعدی بهره میگیریم که بعد سوم آن فریم ها خواهند بود. بعد دوم آن (ستون ها) زیرحامل ها خواهند بود و هر سطر نشان دهنده یک بلاک ارسالی در OFDM است.

برای  $10^7$  بیت ارسالی با توجه به معیار های ذکر شده، تعداد سمبل های در هر فریم در هر زیرحامل برابر با 21 خواهد بود. بر این اساس به هر زیر حامل 21 سمبل اختصاص پیدا خواهد کرد. به آخر داده ها صفر اضافه می شود تا تعداد داده ها بر  $n_c$  بخش پذیر باشد. تعداد فریم های مورد نیاز برای انتقال این حجم از داده برابر با 1191 خواهد بود.

### Generating Symbols & QPSK Modulation

سپس داده های تولید شده با توجه به مدولاسیون QPSK ایندکس 0 تا 3 را دریافت می کنند.

### Frame Divider & Serial to Parallel

در این قسمت، برای فیت شدن داده در فریم ها تعدادی صفر به آخر داده ها اضافه می شود (800) سپس با توجه به معیار های انتخاب شده در قسمت های قبل، و با استفاده از دستور `reshape` داده ها در فریم های مختلف قرار می گیرند. ماتریس بدست آمده،

1191 × 400 × 21 خواهد داشت که 1191 مربوط به تعداد فریم‌ها، 400 مربوط به زیرحامل‌ها و 21 مربوط به تعداد سمبل در هر حامل است.

### Adding Reference Row

در این قسمت، با توجه به توضیحات دستور کار، یک سطر مرجع برای اجرای مدولاسیون DPSK اضافه می‌شود.

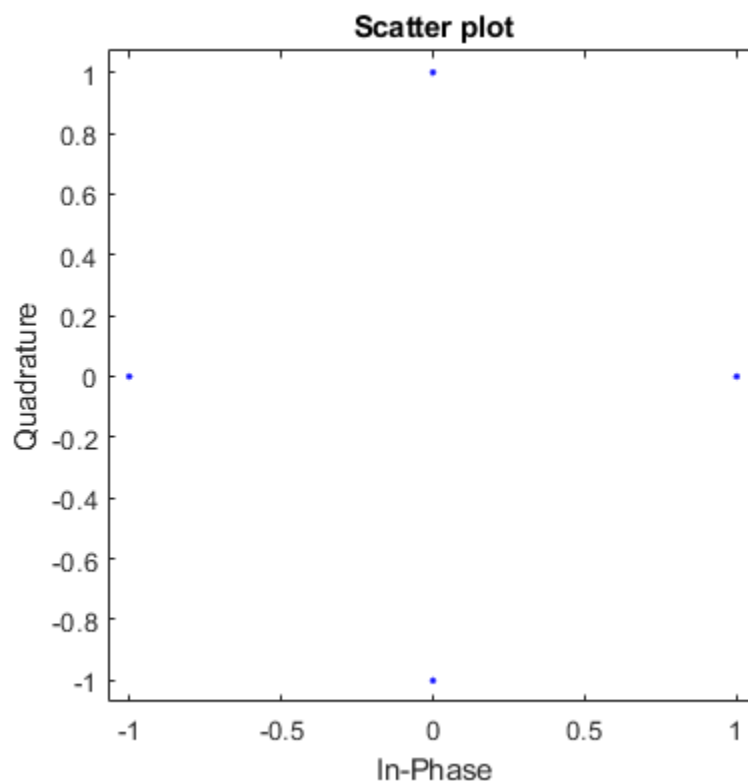
### DPSK Modulation

در این قسمت مدولاسیون DPSK با توجه به روشی که در دستور کار ارائه شد اجرا می‌شود. دلیل اجرای این مدولاسیون، جبران‌سازی تاخیر فاز است.

### QPSK Signal Replacement

در این قسمت، ایندکس‌ها با توجه به عدد مربوط به منظومه سیگنالی خود جایگزین می‌شوند.

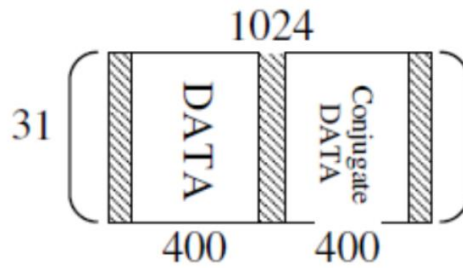
$$\begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 \\ j \\ -1 \\ -j \end{bmatrix}$$



شکل 2 منظومه سیگنالی QPSK

### IFFT bins Allocation

در این مرحله با توجه به دستور کار، ماتریس بدست آمده در قسمت قبل با توجه به طول IFFT گسترش می‌یابد.



شکل 3 گسترش داده با توجه به طول IFFT

## IFFT

با توجه به کارکرد سیستم OFDM، از هر بلاک داده، IFFT گرفته می شوند. سپس تنها قسمت حقیق یاین اتریس ذخیره می شود.

## Adding CP

در این مرحله برای جلوگیری از تداخل بین بلاکی و همچنین تبدیل عملیات convolution به cyclic convolution پیشوند گردشی به داده های ارسالی اضافه می شود. بدین منظور 25٪ ستون های انتهایی ماتریس به ابتدای آن اضافه می شود.

## Parallel to Serial & Cascade Frames

سپس این ماتریس بدست آمده به شکل یک دنباله در می آید تا یک فریم از داده ها حاصل شود. ماتریس کلی ابعاد  $28160 \times 1191$  خواهد داشت که هر سطر آن نشان دهنده یک فریم کامل از داده هاست.

جدول 1 ماتریس ارسالی OFDM از فرستنده

```
ofdm_tx_cp_serial = 28160x1191
-0.0137    0.0391         0    0.0254   -0.0078 ...
-0.0176   -0.0117   -0.0137    0.0039    0.0391
 0.0332    0.0312    0.0195    0.0137    0.0488
 0.0391   -0.0117    0.0059   -0.0137   -0.0254
-0.0195   -0.0332   -0.0078    0.0059   -0.0020
-0.0254    0.0273    0.0293    0.0039   -0.0156
-0.0586   -0.0508   -0.0078   -0.0137    0.0508
-0.0469   -0.0020   -0.0137   -0.0488   -0.0020
 0.0039    0.0313   -0.0391   -0.0254   -0.0313
 0.0078   -0.0098    0.0059    0.0078   -0.0273
      ⋮
```

## کانال

### Clipping

در کانال با توجه به خواسته دستور کار، قبل اضافه شدن نویز بزرگترین دامنه سیگنال ارسالی به اندازه 3dB کاهش می یابد. یعنی لبه بالا سیگنال به اندازه 3dB بریده می شود.

### AWGN Channel

در این قسمت، نویز سفید گاوسی، با واریانس مشخص (با توجه به SNR های مختلف) هم موهومی و هم حقیقی، به سیگنال اضافه می شوند. مدل کانال رایلی در ادامه توضیح داده خواهد شد.

## گیرنده

**Serial to Parallel & Frame Detection**

در این مرحله داده که به شکل دنباله دریافت شده با فرض دانستن ابعاد ماتریس فرستاده شده در گیرنده، به شکل موازی و قبلی خود در می آید.

**CP Removal**

پیشوند گردشی که به داده ها اضافه شد، حذف می شود.

**FFT**

با توجه به ساختار گیرنده OFDM از هر بلاک داده FFT گرفته می شود.

**Extract Carriers from FFT bins**

با توجه به اینکه در فرستنده ماتریس برای گرفتن IFFT گسترش داده شد، در این قسمت برعکس آن عمل می شود و ماتریس داده از داده گسترش یافته استخراج می شود.

**DPSK Demodulation & QPSK Demodulation (minimum error decision)**

در ابتدا با مقایسه هر داده با منظومه سیگنالی QPSK، با استفاده از قانون کمترین فاصله، مقدار (زاویه) هر داده را تخمین می زنیم. سپس ایندکس (سمبل) مربوطه به هر کدام از درایه تخمین شده جایگزین می کنیم.

برای دمدولاسیون DPSK هر درایه از سطر را از سطر قبل خود کم می کنیم و باقیمانده تقسیم آن بر 4 را جایگزین آن درایه می کنیم.

**Parallel to Serial & Cascade Frames**

در این قسمت داده های هر فریم را، در کنار هم می گذاریم تا مجموعه داده های اصلی بدست آید. (هنوز سمبل ها، مدوله شده توسط QPSK هستند).

**Estimated Output & QPSK Demodulation**

در این قسمت هر سمبل با رشته بیت کد شده خود جایگزین می شود. سپس صفر های اضافه شده به انتهای داده ها (برای فیت شدن داده در فریم ها) حذف می شود. (فرض شده تعداد این صفر ها در گیرنده دانسته است. این فرض چندان دور از ذهن نیست این اطلاعات را می توان در هدر اضافه کرد).

**BER Calculations**

در قسمت آخر هم مقدار خطای بیت با توجه به داده دریافتی و داده اولیه محاسبه می شود و در آرایه ای برای رسم نمودار BER ذخیره می شود.

**BER Plot**

در این قسمت هم نمودار خطای بیت بر حسب SNR رسم می شود.

## خواسته های پروژه

### الف.

فایل مربوط به این قسمت با نام **WLC\_CA\_core.mlx** ضمیمه شده است.

$$clipping = 3dB$$

$$SNR = 20dB$$

$$IFFT Length = 1024$$

$$n_c = 400$$

$$sfc = 21$$

مقدار احتمال خطای بیت برای مقادیر فوق برابر با 0.4923 است که مقدار بسیار ضعیفی است و عملاً سیستم درست کار نمی کند. با افزایش SNR رفتار سیستم درست عمل میکند و خطا به صفر میل می کند. برای مثال در  $SNR = 35dB$  خطا به 0.2520 می رسد و در  $SNR = 40dB$  خطا به 1e-4 می رسد که مقدار قابل قبولی است.

یکی از دلایل این خطا در SNR های بالا بنظر می آید مدولاسیون DPSK باشد که برای جبران تاخیر فاز به سیستم اضافه شده است. زیرا زمانی که در یک درایه از داده خطا رخ دهد تمامی درایه های آن ستون، بعد از آن داده خراب می شوند (در دمدولاسیون هر درایه به سطر قبلی خود وابسته است). و به اصطلاح خطا propagate می شود و خطای کل سیستم افزایش می یابد.

همانطور که در قسمت های قبل توضیح داده شد با توجه به مقادیر داده شده در دستور کار، تعداد فریم ها برابر 1191 خواهد بود.

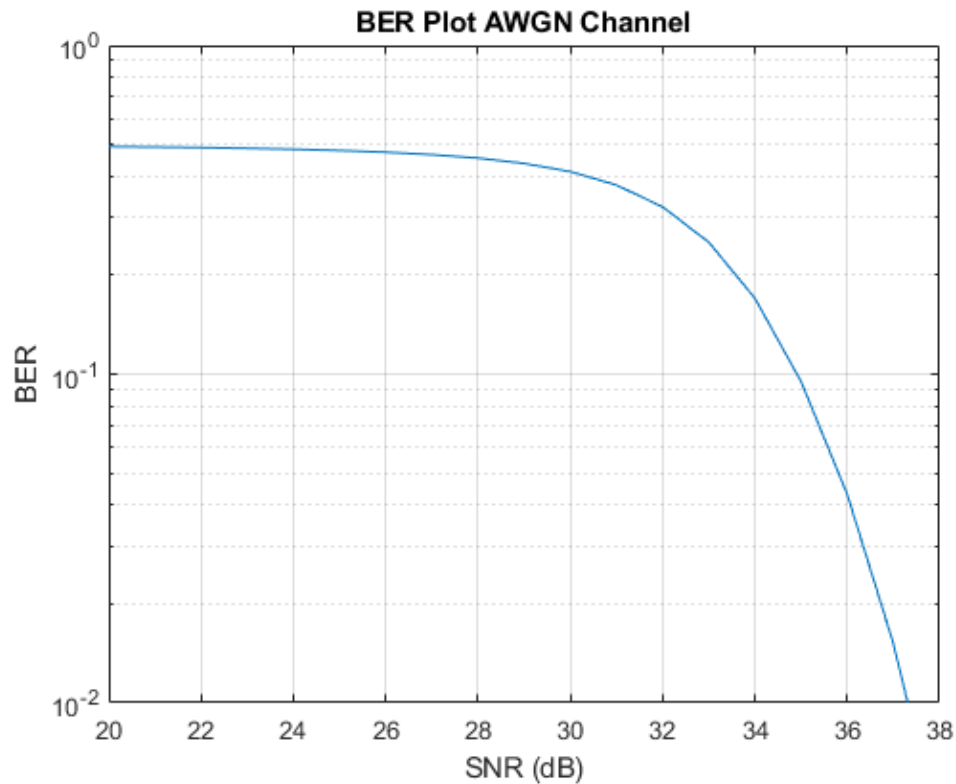
$$Number\ of\ Frames = 1191$$

نحوه عملکرد هر بلوک در قسمت قبل ارائه شد.

ب.

فایل مربوط به این قسمت با نام **WLC\_CA\_AWGN.mlx** ضمیمه شده است.

با توجه به مقادیر قبل، نمودار احتمال خطا از 20 dB الی 40 dB به شکل نمودار 1 است.



نمودار 1 احتمال خطا بر حسب SNR برای کانال AWGN

همانطور که در قست الف توضیح داده شد. رفتار کلی سیستم بر حسب SNR رفتار معقولی است. ( با افزایش SNR خطای سیستم به شدت کاهش می‌یابد.)

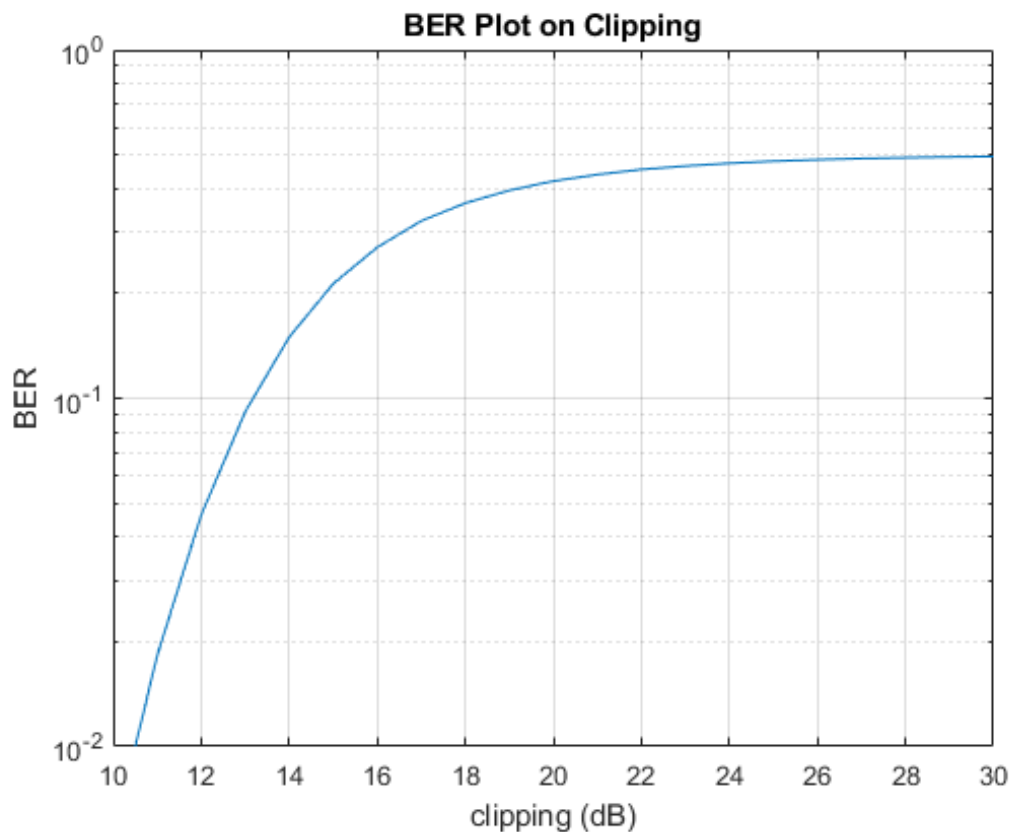


ج.

فایل مربوط به این قسمت با نام **WLC\_CA\_clipping.mlx** ضمیمه شده است.

نمودار خطای سیستم بر حسب مقدار clipping رسم شده این مقدار از 3dB شروع و تا 30 dB بدست آمده است و نمودار 2 بدست آمده که رفتار این قسمت هم معقول است. با افزایش بریدگی سیگنال شکل اصلی سیگنال کم کم از بین می‌رود و خطا سیستم افزایش می‌یابد.

در این قسمت برای عملکرد درست سیستم  $SNR = 40\text{ dB}$  در نظر گرفته شده است.



نمودار 2 احتمال خطای سیستم بر حسب clipping

## د.

فایل مربوط به این قسمت با نام **WLC\_CA\_Rayleigh.mlx** ضمیمه شده است.

برای اضافه کردن مدل کانال رایلی تغییراتی در سیستم بوجود آوردیم که به شرح زیر است؛

یک بلاک به سیستم کانال اضافه شده است. می دانیم در مدل کانال رایلی ضریبی در سیگنال ضرب می شود که از توزیع رایلی تبعیت می کند. خود توزیع رایلی از بخش هم فاز و غیر هم فاز با توزیع نرمال تشکیل شده است. پس کافی است برای مدل کردن این کانال اعداد تصادفی با توزیع نرمال (مانند نویز سفید) را تولید کرده، سپس بخش هم فاز و غیر هم فاز در سیگنال ضرب کنیم.

$$H = R_I + R_Q$$

$$R_I, R_Q \sim N(0, \sigma^2)$$

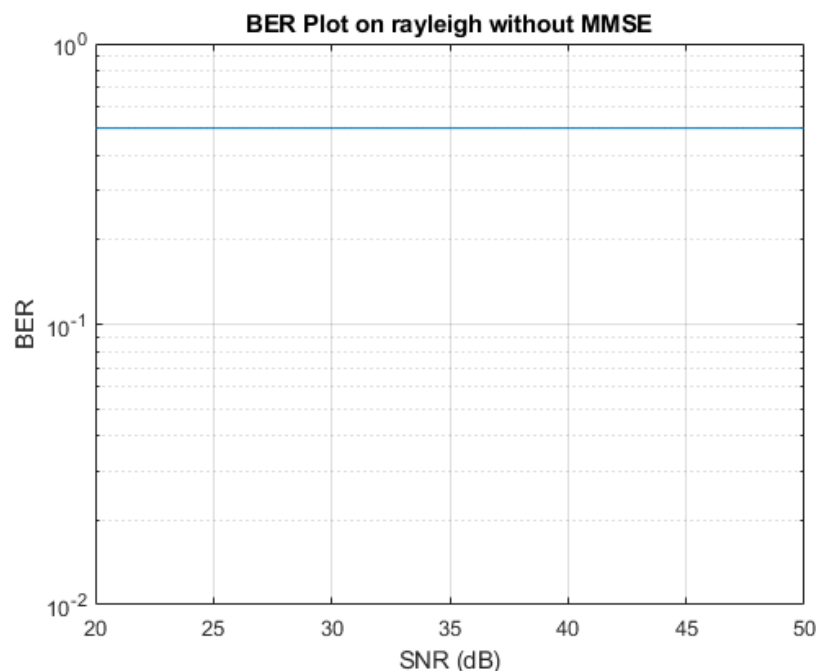
$$H.x$$

و مدل آن در متلب به شکل زیر پیاده سازی شده است؛

```
var_rayleigh = 1;
rx_rayleigh = ofdm_tx_cp_serial_clipped;
h = sqrt(var_rayleigh) .* (randn(size(rx_rayleigh)) + 1i * randn(size(rx_rayleigh)));
rx_rayleigh = h .* rx_rayleigh;
```

شکل 4 پیاده سازی کانال رایلی در متلب

در این قسمت از همسان ساز استفاده نشده است. نمودار احتمال خطا به شکل نمودار 3 است. همانطور که انتظار می رفت سیستم بطور کلی مختل شده است.



نمودار 3 نمودار احتمال خطا در مدل کانال رایلی بدون همسان ساز

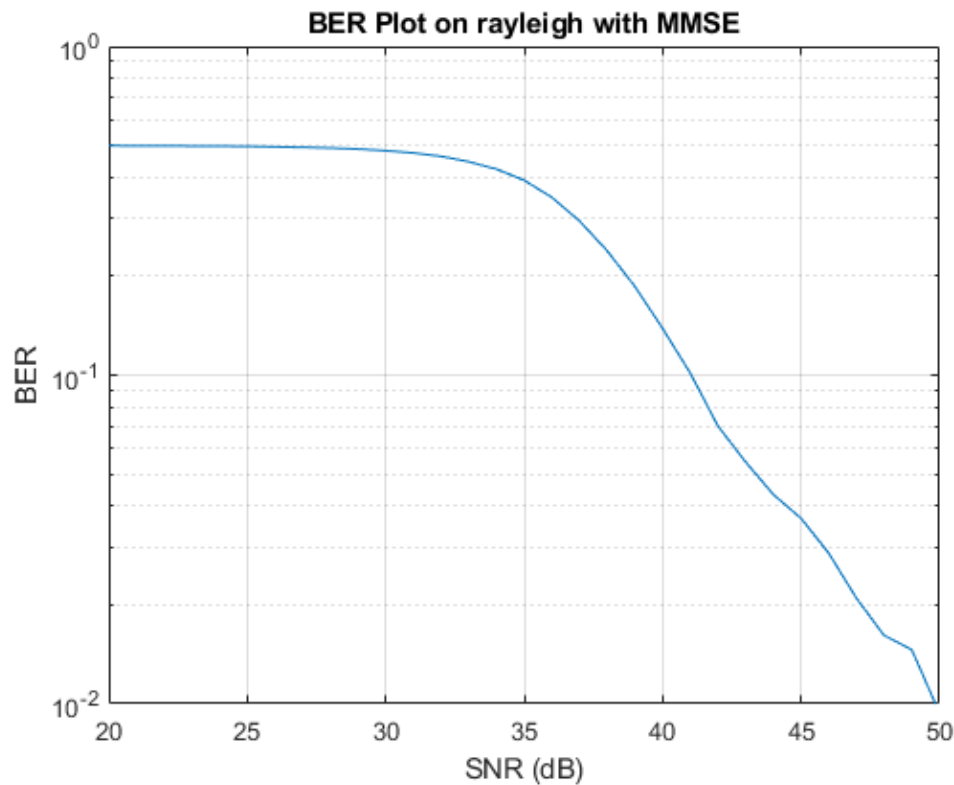
.۵

فایل مربوط به این قسمت با نام **WLC\_CA\_Rayleigh\_MMSE.mlx** ضمیمه شده است.

علاوه بر تغییرات قسمت قبل در این قسمت در گیرنده همسان ساز MMSE اضافه شده است. با توجه به ماتریس ها و عملیات پیچیده در گیرنده، برای راحتی پیاده سازی، این همسان‌ساز قبل از همه بلوک های گیرنده قرار گرفته است (می دانیم در واقع باید بعد از بلوک FFT قرار گیرد.) و شرایط کانال در گیرنده دانسته فرض شده و با توجه به روابط همسان ساز MMSE از رابطه زیر تبعیت می‌کند؛

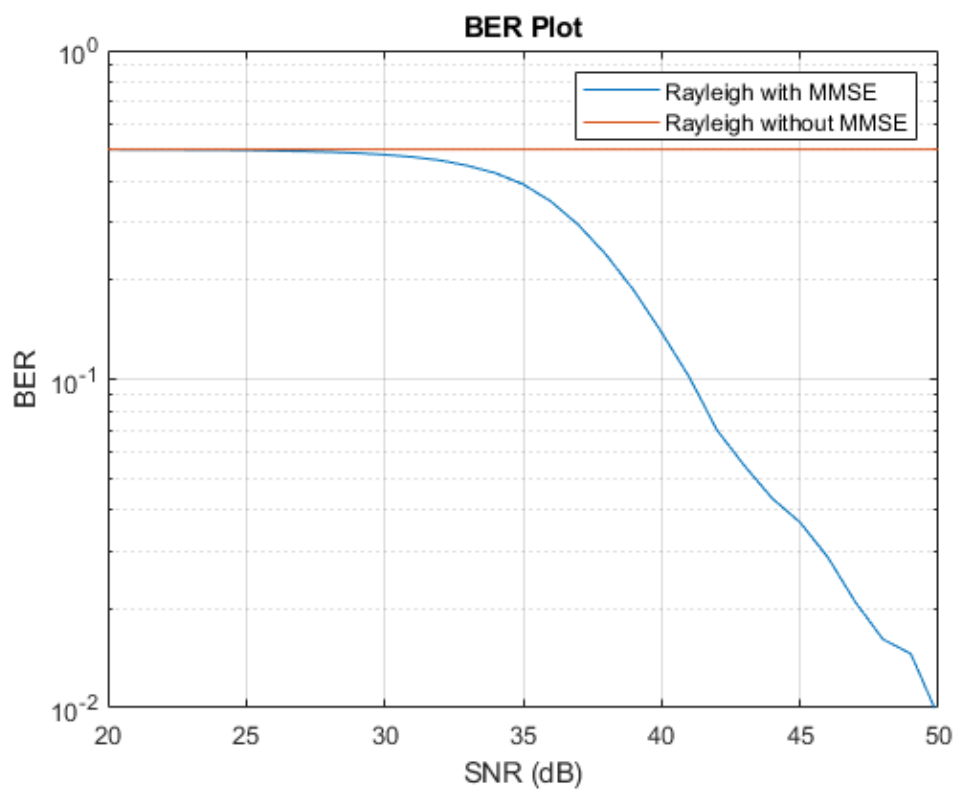
$$W_k = \frac{H_k^*}{|H_k|^2 + \frac{\sigma_n^2}{\sigma_s^2}}$$

نمودار احتمال خطا برای این بخش به شکل نمودار 4 بدست می‌آید. که نشان دهنده آن است که همسان‌ساز به خوبی عمل کرده و تاثیر کانال رایلی را جبران کرده است.

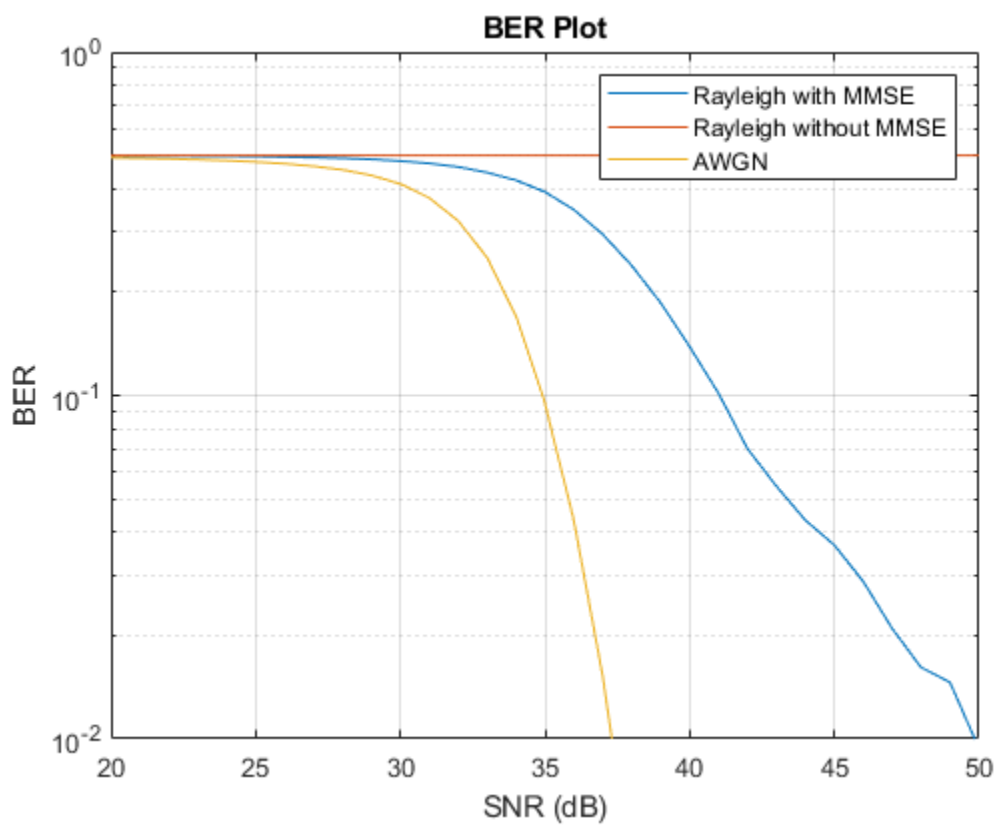


نمودار 4 نمودار احتمال خطا در مدل کانال رایلی با وجود همسان‌ساز

در نمودار 5 عدم وجود و وجود همسان ساز به خوبی مشخص است. در نمودار 5 هم مدل کانال AWGN بدون مدل رایلی آورده شده است همانطور ه مشخص است همسان‌ساز کاملاً اثر کانال را از بین نبرده است.



نمودار 5 مقایسه نمودار احتمال خطا بین سیستم با همسان ساز و بدون آن



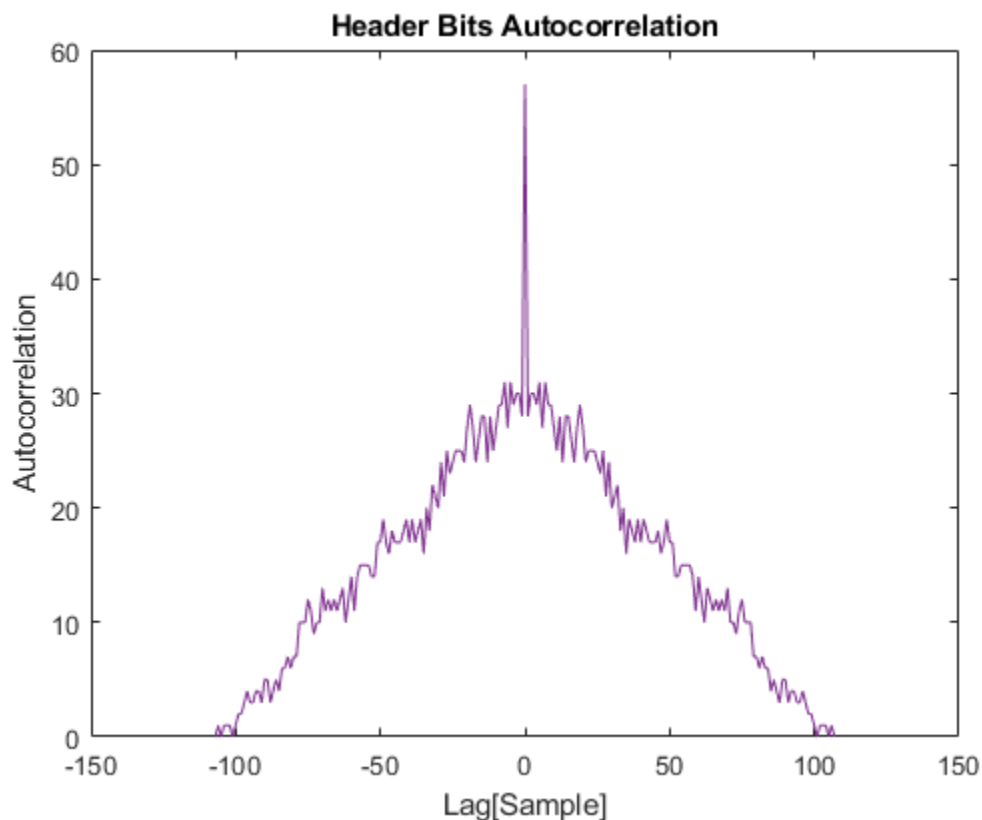
نمودار 6 مقایسه کانال رایلی بدون همسانساز، با همسان ساز و کانال AWGN

**و. (امتیازی)**

فایل مربوط به این قسمت با نام **WLC\_CA\_delay.mlx** و **WLC\_CA\_header.mlx** ضمیمه شده است.

در این قسمت هم تغییراتی در ساختار سیستم بوجود آمده است. در ساختار گیرنده بخشی برای تولید header اضافه شده است. در این قسمت یک کدی که خواص هم‌بستگی خوبی دارد (از آزمایشگاه مخابرات دیجیتال اقتباس شده است) در هدر اضافه شده است. با توجه به اینکه طول هدر هشت برابر طول فریم است بقیه مقادیر آن صفر در نظر گرفته می‌شود. برای کارکرد سریع‌تر متلب و شبیه‌سازی، اضافه کردن بقیه صفرهای هدر را نادیده گرفتیم و طول هدر را برابر با طول یک فریم در نظر گرفتیم. طبعاً می‌توان اطلاعات دیگری نیز در هدر قرار داد. (مانند طول دیتا و تعداد فریم‌ها و ... که در قسمت‌های قبل دانسته در نظر گرفته شد).

hex code: **1C6387FF5DA4FA325C895958DC5**

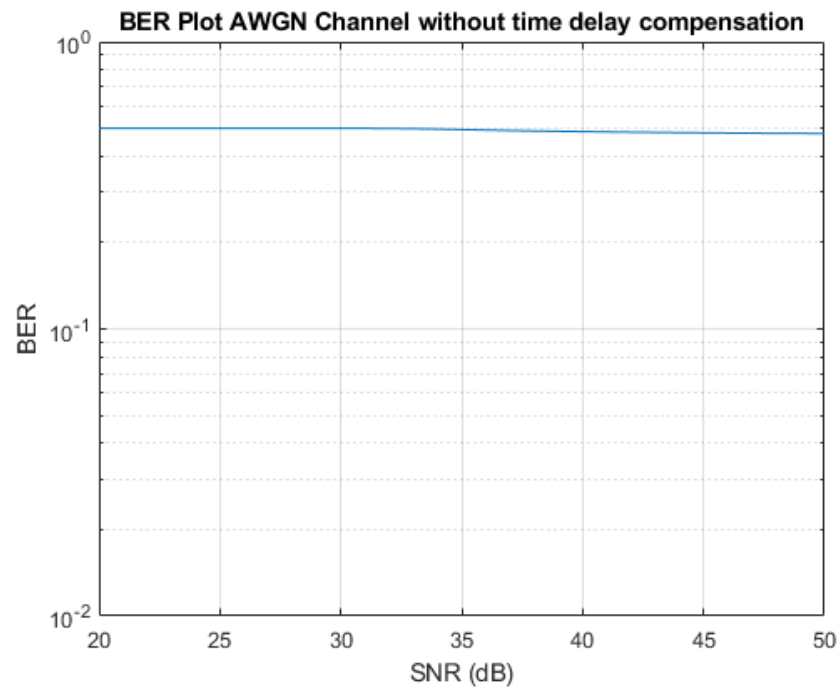


نمودار 7 نمودار خودهمبستگی کد استفاده شده

حال این کد که در هدر اضافه شده در گیرنده نیز دانسته است. در گیرنده با ضافه کردن بلوکی، با استفاده از تابع همبستگی، همبستگی بین سیگنال دریافتی و کد موجود بدست می‌آید. ایندکس نقطه ماکسیمم این همبستگی برابر با مقدار تاخیر زمانی ما در کانال است.

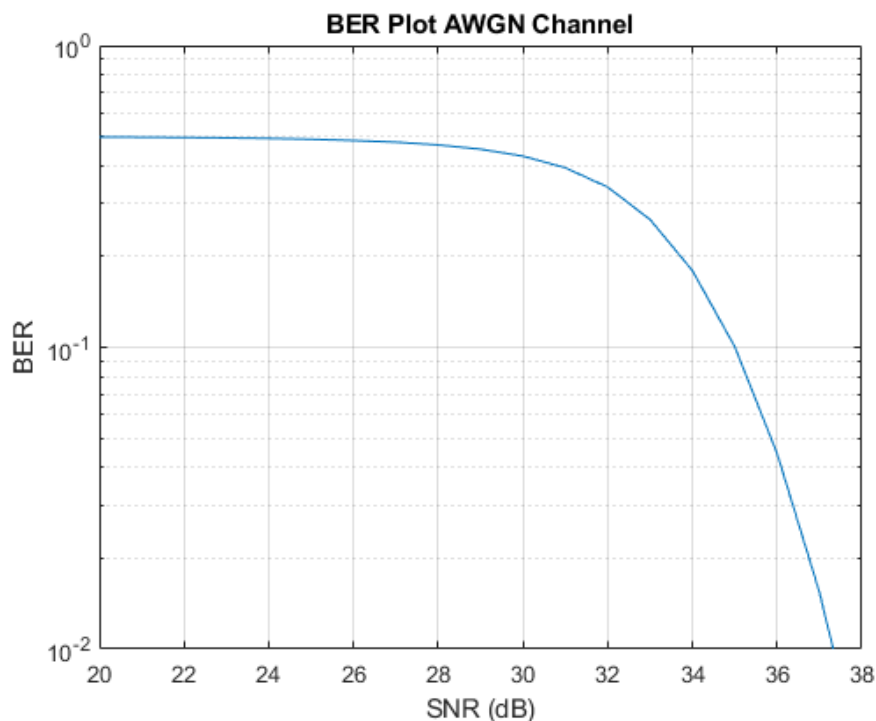
در پیاده‌سازی، تاخیر کانال را 10 سمبل فرض می‌کنیم. (10 سمبل رندوم قبل داده ارسالی اضافه می‌شود).

بدون تخمین تاخیر کانال نمودار احتمال خطای سیستم به شکل نمودار 8 بدست می‌آید که حاکی از مختل شدن کامل سیستم است.



نمودار 8 نمودار خطای سیستم بدون جبران سازی تاخیر کانال

با اضافه کردن بلوک تخمین تاخیر کانال در گیرنده با روش های ذکر شده نمودار 9 بدست می‌آید که نشان از کارکرد درست سیستم دارد.



نمودار 9 نمودار احتمال خطای سیستم با استفاده از جبران ساز تاخیر کانال به کمک header

توجه شود که تخمین کانال در تمامی موارد به درستی انجام شده است. تاخیر کانال با عدد 10 تخمین زده شده که درست است.

```
corr_out_rx_max = 19.6072 - 0.0464i
idx_max = 28180
estimated_time_delay = 10
ans = 1x2
      28160      1192

corr_out_rx_max = 19.4912 - 0.0099i
idx_max = 28180
estimated_time_delay = 10
ans = 1x2
      28160      1192

corr_out_rx_max = 19.6071 - 0.0230i
idx_max = 28180
estimated_time_delay = 10
ans = 1x2
      28160      1192

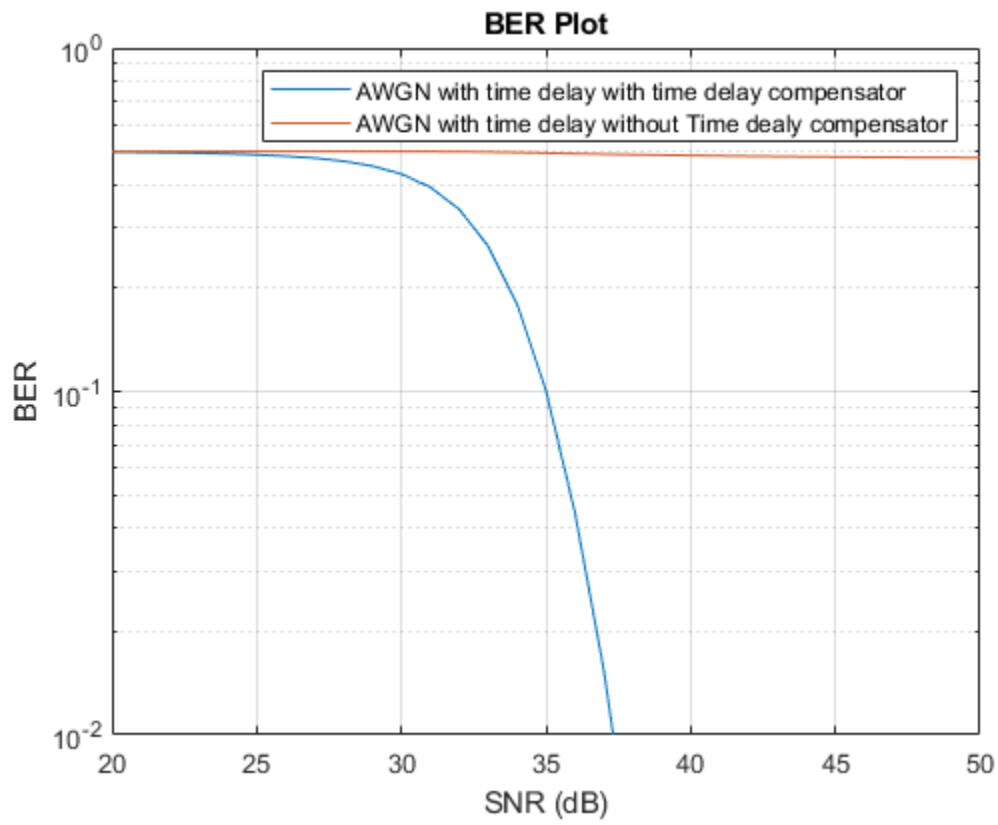
corr_out_rx_max = 19.5375 - 0.1570i
idx_max = 28180
estimated_time_delay = 10
ans = 1x2
      28160      1192

corr_out_rx_max = 19.5749 + 0.0481i
idx_max = 28180
estimated_time_delay = 10
ans = 1x2
      28160      1192

corr_out_rx_max = 19.5532 - 0.0792i
idx_max = 28180
estimated_time_delay = 10
ans = 1x2
```

شکل 5 بخشی از تاخیر تخمین زده شده با استفاده از هدر که در تمامی موارد درست انجام شده و خطایی اعلام نشده

در حالتی که تاخیر کانال وجود دارد، نمودار احتمال خطای سیستم در نمودار 10، در دو حالتی که از header برای تخمین تاخیر کانال استفاده شده و در حالت بدون استفاده از آن مقایسه شده است.



نمودار 10 مقایسه نمودار احتمال خطا در دو حالت استفاده از جبران ساز تاخیر کانال و عدم استفاده از آن

همانطور که انتظار داشتیم اضافه کردن هدر کمی bit rate ما را پایین می آورد اما کمک زیادی به جبران اثرات کانال در گیرنده می کند.