به نام خدا

درس: شبکه مخابرات داده‌ها

استاد: دکتر محمدرضا پاکروان

گزارش پروژه شماره ۱

سیّدمحمّدامین منصوری طهرانی

۹۴۱۰۵۱۷۴

۱. یکی از تکنیک‌ها $Space\ Division\ Multiplexing(SDM)$ است. در این روش محیط انتقال داده به صورت فیزیکی به چند قسمت تقسیم می‌شود. در هر کانال که به صورت فیزیکی از سایر کانال‌ها جدا شده‌است می‌توان تعدادی کانال انتقال داده به یکی از روش‌های FDM, TDM, CDM داشت. در واقع برای مثال در مخابرات سیمی در این روش از رساناهای الکتریکی نقطه به نقطه جدا از هم برای هر کانال استفاده می‌شود. در مخابرات بی‌سیم این روش به وسیله آنتن‌های جدا از هم که تشکیل آرایه فازی می‌دهند پیاده‌سازی می‌شود. در واقع این جدایی فیزیکی برای رساندن چند رشته مختلف داده به صورت همزمان به کار می‌رود و تلاشی برای دستیابی به سیستم‌های $Multiple\ input\ multiple\ output(MIMO)$ است که ظرفیت را با افزایش تعداد آنتن‌ها بهبود می‌بخشد. از این روش هنگامی استفاده می‌شود که بخواهیم حجم زیادی از داده را در فاصله اندک یا متوسط منتقل کنیم. مثال این حالت در $Data\ center$ ها و یا اتصال شبکه‌های کوچک درون یک شهر است. این روش به استفاده از فیبرهای خاصی نیاز دارد و به همین دلیل برای کاربردهای فواصل دور مناسب نیست(چون باید زیرساخت‌های زیادی را تغییر داد.)

یک روش دیگر $Frequency\ Dvision\ Multiplexing(FDM)$ می‌باشد که در واقع یک تکنولوژی آنالوگ به حساب می‌آید. در این روش محیطی که برای انتقال تقسیم می‌شود «فرکانس» است و سیگنال‌ها و رشته‌های داده مختلف در بازه‌های فرکانسی جدا از هم بر روی یک کانال فرستاده می‌شوند. در FDM سیگنال‌ها **الکتریکی** هستند. از کاربردهای بسیار متداول این تکنیک می‌توان از رادیو و تلویزیون نام برد. به محل سکونت مشترکان فقط یک کابل وارد می‌شود ولی در فرکانس‌های مختلف کانال‌های مختلف قرار دارند و گیرنده مشترک با تنظیم فرکانس مورد نظر خود می‌تواند کانال دلخواه را گوش دهد یا ببیند.

یک تکنیک در مخابرات نوری، $Wavelength\ Division\ Multiplexing(WDM)$ بوده و در این تکنیک از حامل‌های($carrier$) اپتیکی استفاده می‌شود. این سیگنال‌ها که بر روی حامل‌های اپتیکی سوارند توسط مالتیپلکسرهایی بر روی یک فیبر نوری فرستاده می‌شوند. این عملیات مالتی‌پلکس در واقع با تفاوت طول موج حامل یا متناظراً رنگ آن محقق می‌شود. در این روش هم‌چنین دو جهته بودن ارسال اطلاعات نیز برآورده می‌شود. در واقع تفاوت آن با FDM در فیزیک متفاوت ناشی از تفاوت محدوده فرکانسی است. اگر در محدوده‌های رادیویی باشیم از FDM و اگر در محدوده نوری باشیم WDM را به کار می‌بریم. از علل محبوبیت آن امکان افزایش ظرفیت شبکه بدون استفاده از فیبر بیشتر است. در واقع فقط با ارتقاء دادن مالتی‌پلکسرها و دی‌ماکس‌های دو سر می‌توان به این هدف دست یافت. هم‌چنین ظرفیت و سرعت شبکه‌های فیبر نوری بسیار زیاد بوده و این باعث پیش رفتن به سمت این تکنیک می‌شود.

یک تکنیک متداول دیگر $Time\ Division\ Multiplexing(TDM)$ است که در آن برای جدا کردن رشته‌های داده به جای تقسیم فرکانس یا فضای فیزیکی از تقسیم «زمان» استفاده می‌شود و معمولا دیجیتال است. در این روش در بازه‌های زمانی جدا، از هر فرستنده تعدادی بیت گرفته می‌شود و با رشته‌های دریافتی از سایر فرستنده‌ها در بازه‌های زمانی بعدی ترکیب شده و منتقل می‌شود و اگر این عملیات و به همراه آن جداسازی این رشته‌های برای گیرنده‌های مربوط به هر فرستنده با سرعت کافی انجام شود، گیرنده‌ها متوجه این تخصیص بازه‌های زمانی نخواهند شد. از کاربردهای آن می‌توان به PCM(انتقال چند تماس تلفنی روی یک کابل) و یا سیستم تلفن GSM اشاره کرد.($TDMA$)

از دیگر تکنیک‌های متداول نیز می‌توان به $Code\ Division\ Multiplexing(CDM)$ اشاره کرد. در این روش رشته‌های مختلف داده بر روی یک کانال و یک فرکانس منتقل می‌شوند ولی پیش از فرستاده شدن کد شده و کلید این کد به همراه سیگنال برای گیرنده فرستاده می‌شود و امکان بازیابی سیگنال برای گیرنده فراهم می‌شود. از کاربردهای بزرگ $CDMA$ می‌توان به GPS اشاره کرد.[۱]

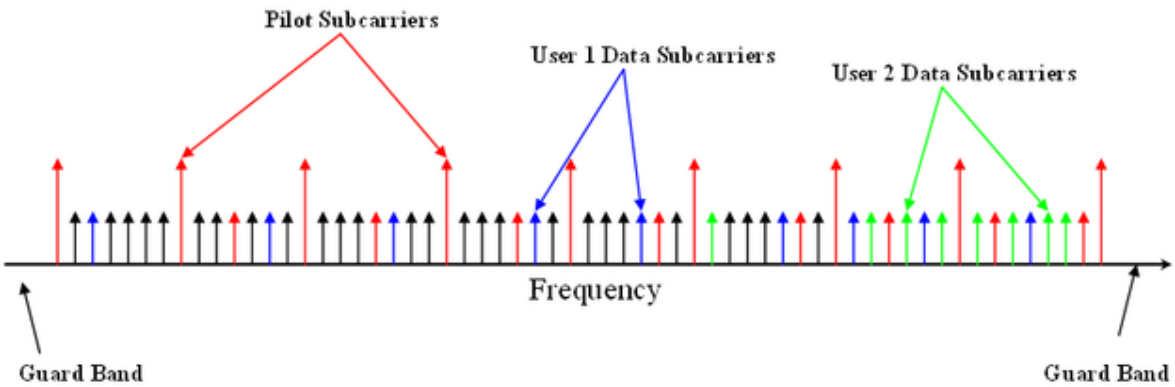
۲.

Digital Audio Broadcasting (DAB) – Digital Television – Wireless LAN – WiMAX – ADSL – LTE – Wireless personal area network (PAN)

۳. از مزایای اصلی این تکنیک نسبت به روشی که در آن از یک تک حامل(FDM) برای ارسال استفاده می‌شود توانایی مقابله با شرایط بد کانال از جمله تضعیف فرکانس‌های بالا در سیم‌های مسی بلند، تداخل باریک‌باند و اثرات مخرب ناشی از رسیدن سیگنال از چند مسیر است. این اثرات کانال بدون فیلترهای متعادل‌ساز چندان پیچیده قابل رفع شدن هستند و علت این امر این است که به جای یک سیگنل پهن‌باند و سریع، از تعداد زیاد سیگنال کندتر و باریک‌باند به جای آن استفاده می‌شود. هم‌چنین نرخ پایین $symbol$ها امکان استفاده از بازه محافظ را به‌وجود می‌آورد. به این ترتیب از خطای ناشی از تداخل بین سمبلی جلوگیری می‌شود(ISI). این خطا ناشی از تاخیرهای احتمالی و پژواک است. از دیگر مزیت‌های مهم آن بازدهی طیفی بیشتر این روش نسبت به دیگر تکنیک‌های $double\ sideband$ می‌باشد چون تعامد حامل‌ها اثر تداخل را از بین می‌برد و امکان استفاده از حامل‌های بیشتر در یک پهنای باند فراهم می‌شود. هم‌چنین پیاده‌سازی این روش به وسیله الگوریتم FFT بسیار کارآمد است.

۴. خیر. اگر یک $throughput$ مشخص مد نظر باشد، در این روش با ترکیب تعداد زیادی از $sub-carrier$ های $low\ symbol\ rate$ به همان $throughput$ حاصله از FDM دست می‌یابیم. هم‌چنین از آنجا که فاصله $sub-carrier$ ها یا همان $\Delta f = 1/T_U$ است، اگر تعداد این زیرحامل‌ها در یک پهنای باند مشخص زیاد شود، فاصله آن‌ها کم شده و با رابطه اخیر طول مفید هر $symbol$ زیاد شده و $symbol\ rate$ کاهش می‌یابد. تعداد زیرحامل‌ها و این نرخ رابطه عکس دارند.

۵. $OFDMA$ استفاده از تکنیک $OFDM$ برای امکان‌پذیر نمودن ارتباط چند کاربر از یک کانال است و با اختصاص پویای $sub-carrier$ ها به کاربرهای مختلف محقق می‌شود. بنابراین می‌توان آن‌را ترکیبی از $FDMA$ و $TDMA$ دانست. هم‌چنین برای حمایت از $Quality\ of\ Service(Qos)$ ممکن است تعداد زیرحامل‌های اختصاص‌یافته به کاربران مختلف متفاوت باشد. تصویر زیر نحوه پیاده‌سازی $OFDMA$ به وسیله $OFDM$ را نشان می‌دهد.



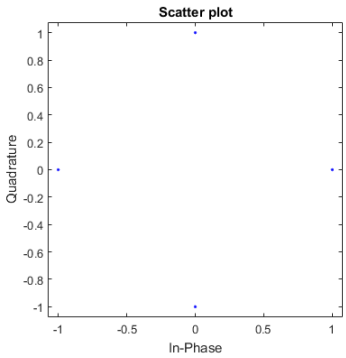
شکل ۱

۶. به وسیله دستور گفته‌شده $100KB$ داده تولید می‌کنیم و این اعداد با احتمال مساوی صفر یا یک هستند. میانگین و واریانس آن‌ها در کد محاسبه می‌شود و با دو متغیر $Mean$ و Var قابل مشاهده هستند. در اینجا گزارش نمی‌شوند زیرا با هر بار اجرای برنامه تغییرات اندکی محتمل است. اما میانگین بسیار به 0.5 و واریانس نیز بسیار به 0.25 نزدیک است.($\frac{\sum(x-\bar{x})^2}{N} = 0.5^2 = 0.25$)

۷. برای مدوله‌کردن باید دقت کنیم که در روش $QPSK$ هر دو بیت به عنوان یک $symbol$ در نظر گرفته می‌شوند و سپس مدوله می‌شوند. به همین منظور داده‌ها را دو بیت دو بیت جدا کرده و با تابع $convertBase$ که نوشته‌ایم به عددی بین 0 تا 3 می‌نگاریم. حال این بردار جدید که شامل 50000 عدد است را به عنوان آرگومان ورودی اول $pskmod$ استفاده می‌کنیم. آرگومان دوم همان پارامتر نقاط $constellation$ است که در این تمرین آن را ۴ انتخاب می‌کنیم.

آرگومان سوم هم زاویه $offset$ نقاط $constellation$ است که ما آن را صفر برمی‌گزینیم. آرگومان آخر هم جهت چرخش در نقاط $constellation$ را نشان می‌دهد و ما آن را $Gray$ گذاشته‌ایم و بنابراین در هر تغییر فقط یکی از دو بیت مجاز است تغییر کند.

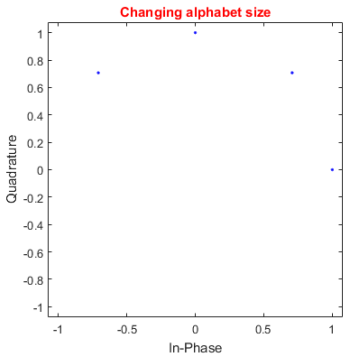
خروجی $pskmod$ به عنوان ورودی تابع $scatterplot$ داده می‌شود و نتیجه در تصویر زیر رسم می‌شود که مطابق انتظار چهار نقطه در زاویه‌های به فاصله 90 هستند.



شکل ۲

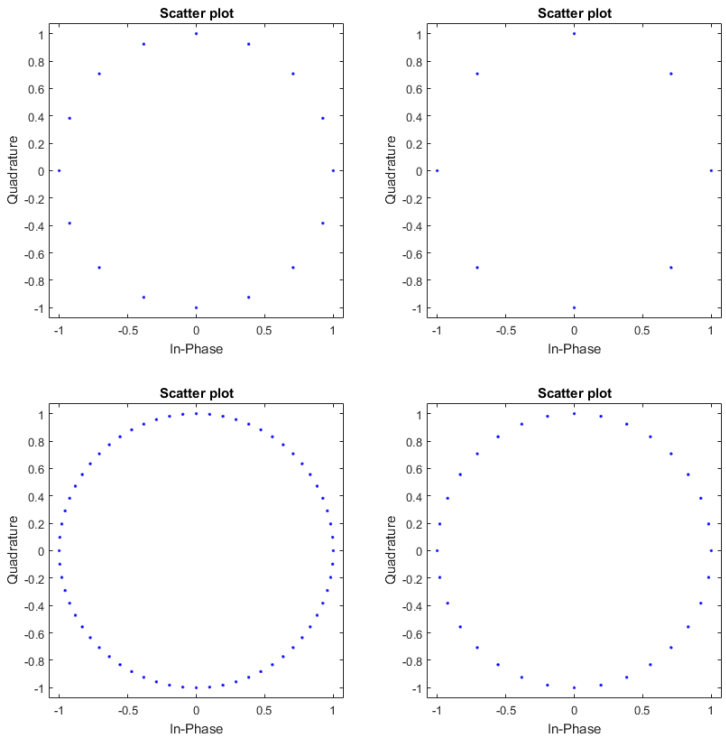
۸. عملیات دمدولاسیون به وسیله دستور $pskdemod$ انجام می‌شود و آرگومان‌های ورودی آن کاملا مشابه $pskmod$ هستند و باید برای بازیابی صحیح با آن‌ها برابر قرار داده شوند. در کد نتیجه دمدولاسیون در بردار $outDemod$ ذخیره می‌شود.

۹. اگر تعداد $constellation\ points$ را عوض کنیم ولی هم‌چنان دو بیت دو بیت جدا کنیم کار درستی نیست و در واقع فقط به ۴ نقطه اول از تعداد نقاط $constellation\ points$ مدوله می‌کنیم.(مطابق شکل زیر)



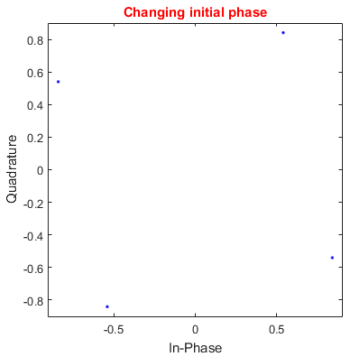
شکل ۳

در شکل فوق تعداد نقاط ۸ انتخاب شده و برای نگاشت مناسب باید هشت‌تا هشت‌تا جدا کنیم و در صورتی‌که این کار را انجام دهیم به تصاویر زیر می‌رسیم.(با تغییر پارامتر $alphabetSize$ در ابتدای کد سوال ۷ امکان مشاهده تصاویر زیر به ازای مقادیر مختلف $alphabetSize$ وجود دارد. در تصاویر زیر این مقدار از ۸ تا ۶۴ تغییر کرده است.)



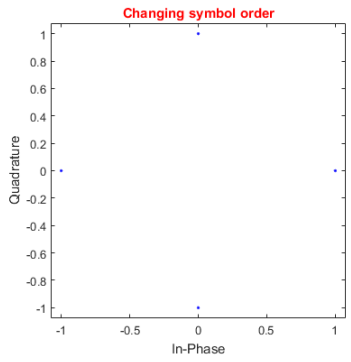
شکل ۴

اگر پارامتر سوم یا زاویه فاز اولیه را تغییر دهیم شکل می‌چرخد.(مطابق تصویر زیر)



شکل ۵

اگر پارامتر چهارم یا ترتیب کد کردن را به $bin(binary)$ تغییر دهیم در شکل تغییری حاصل نمی‌شود و فقط مقداردهی و مدولاسیون تغییر می‌کند. به این معنی که اعداد متوالی در زاویه‌های متوالی قرار می‌گیرند نه اعدادی که باینری آن‌ها تنها در ۱ بیت تفاوت دارد.(تصویر زیر)



شکل ۶

۱۰. این دستور با روشی غیر از محاسبه مستقیم با فرمول DFT آن‌را محاسبه می‌کند. خود DFT نیز در واقع یک پریود سری فوریه یک سیگنال گسسته در زمان است. در واقع وقتی یک سیگنال گسسته در زمان محدود به $MATLAB$ داده می‌شود فرض می‌کند این سیگنال متناوب است و در خروجی یک پریود از سری فوریه آن‌را بدست می‌دهد. در حالت عادی برای محاسبه ضرایب سری فوریه یک دنباله به عنوان مثال هشت‌تایی(متناوب فرض شده) باید ضرایب سمت چپ در عبارت زیر با ضرب ماتریسی بدست آیند که $8 \times 8 = 64$ ضرب را ایجاب می‌کند.

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{bmatrix} = \begin{bmatrix} W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 \\ W_8^0 & W_8^1 & W_8^2 & W_8^3 & W_8^4 & W_8^5 & W_8^6 & W_8^7 \\ W_8^0 & W_8^2 & W_8^4 & W_8^6 & W_8^0 & W_8^2 & W_8^4 & W_8^6 \\ W_8^0 & W_8^3 & W_8^6 & W_8^1 & W_8^4 & W_8^7 & W_8^2 & W_8^5 \\ W_8^0 & W_8^4 & W_8^0 & W_8^4 & W_8^0 & W_8^4 & W_8^0 & W_8^4 \\ W_8^0 & W_8^5 & W_8^2 & W_8^7 & W_8^4 & W_8^1 & W_8^6 & W_8^3 \\ W_8^0 & W_8^6 & W_8^4 & W_8^2 & W_8^0 & W_8^6 & W_8^4 & W_8^2 \\ W_8^0 & W_8^7 & W_8^6 & W_8^5 & W_8^4 & W_8^3 & W_8^2 & W_8^1 \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \\ x[4] \\ x[5] \\ x[6] \\ x[7] \end{bmatrix}$$

$$W_N = e^{-j\frac{2\pi}{N}}$$

در یک روش این دنباله را به دو دنباله 4 نقطه‌ای تبدیل می‌کنیم و ضرایب c_k را نیز به دو جزء $d_k + e_k$ تجزیه می‌کنیم. مطابق شکل‌های زیر این ضرایب حساب می‌شوند.(دنباله‌های با شماره زوج و فرد از هم جدا می‌شوند.)

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} W_4^0 & W_4^0 & W_4^0 & W_4^0 \\ W_4^0 & W_4^1 & W_4^2 & W_4^3 \\ W_4^0 & W_4^2 & W_4^0 & W_4^2 \\ W_4^0 & W_4^3 & W_4^2 & W_4^1 \end{bmatrix} \begin{bmatrix} x[1] \\ x[3] \\ x[5] \\ x[7] \end{bmatrix} \qquad \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} W_4^0 & W_4^0 & W_4^0 & W_4^0 \\ W_4^0 & W_4^1 & W_4^2 & W_4^3 \\ W_4^0 & W_4^2 & W_4^0 & W_4^2 \\ W_4^0 & W_4^3 & W_4^2 & W_4^1 \end{bmatrix} \begin{bmatrix} x[0] \\ x[2] \\ x[4] \\ x[6] \end{bmatrix}$$

تا به این‌جا به $2 \times (4 \times 4) = 32$ ضرب نیاز بوده است. حال رابطه این ضرایب با ضرایب ماتریس در زیر مشخص می‌شود:

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{bmatrix} = \begin{bmatrix} d_0 + e_0 \\ d_1 + e_1 \\ d_2 + e_2 \\ d_3 + e_3 \\ d_4 + e_4 \\ d_5 + e_5 \\ d_6 + e_6 \\ d_7 + e_7 \end{bmatrix}$$

$$\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \end{bmatrix} = \begin{bmatrix} W_8^0 & W_8^0 & W_8^0 & W_8^0 \\ W_8^0 & W_8^2 & W_8^4 & W_8^6 \\ W_8^0 & W_8^4 & W_8^0 & W_8^4 \\ W_8^0 & W_8^6 & W_8^4 & W_8^2 \\ W_8^0 & W_8^0 & W_8^0 & W_8^0 \\ W_8^0 & W_8^2 & W_8^4 & W_8^6 \\ W_8^0 & W_8^4 & W_8^0 & W_8^4 \\ W_8^0 & W_8^6 & W_8^4 & W_8^2 \end{bmatrix} \begin{bmatrix} x[0] \\ x[2] \\ x[4] \\ x[6] \end{bmatrix}$$

$$\begin{bmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \end{bmatrix} = \begin{bmatrix} W_8^0 & W_8^0 & W_8^0 & W_8^0 \\ W_8^1 & W_8^3 & W_8^5 & W_8^7 \\ W_8^2 & W_8^6 & W_8^2 & W_8^6 \\ W_8^3 & W_8^1 & W_8^7 & W_8^5 \\ W_8^4 & W_8^4 & W_8^4 & W_8^4 \\ W_8^5 & W_8^7 & W_8^1 & W_8^3 \\ W_8^6 & W_8^2 & W_8^6 & W_8^2 \\ W_8^7 & W_8^5 & W_8^3 & W_8^1 \end{bmatrix} \begin{bmatrix} x[1] \\ x[3] \\ x[5] \\ x[7] \end{bmatrix}$$

در اینجا ماتریس بزرگ را به دو زیرماتریس تقسیم کردیم. قدم بعدی یافتن رابطه ضرایب d_k, e_k با ضرایب بدست آمده a_k, b_k است:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} W_4^0 & W_4^0 & W_4^0 & W_4^0 \\ W_4^0 & W_4^1 & W_4^2 & W_4^3 \\ W_4^0 & W_4^2 & W_4^0 & W_4^2 \\ W_4^0 & W_4^3 & W_4^2 & W_4^1 \end{bmatrix} \begin{bmatrix} x[0] \\ x[2] \\ x[4] \\ x[6] \end{bmatrix} = \begin{bmatrix} W_8^0 & W_8^0 & W_8^0 & W_8^0 \\ W_8^0 & W_8^2 & W_8^4 & W_8^6 \\ W_8^0 & W_8^4 & W_8^0 & W_8^4 \\ W_8^0 & W_8^6 & W_8^4 & W_8^2 \end{bmatrix} \begin{bmatrix} x[0] \\ x[2] \\ x[4] \\ x[6] \end{bmatrix}$$

$$\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \end{bmatrix} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} W_8^0 & W_8^0 & W_8^0 & W_8^0 \\ W_8^0 & W_8^2 & W_8^4 & W_8^6 \\ W_8^0 & W_8^4 & W_8^0 & W_8^4 \\ W_8^0 & W_8^6 & W_8^4 & W_8^2 \\ W_8^0 & W_8^0 & W_8^0 & W_8^0 \\ W_8^0 & W_8^2 & W_8^4 & W_8^6 \\ W_8^0 & W_8^4 & W_8^0 & W_8^4 \\ W_8^0 & W_8^6 & W_8^4 & W_8^2 \end{bmatrix} \begin{bmatrix} x[0] \\ x[2] \\ x[4] \\ x[6] \end{bmatrix}$$

در این‌جا رابطه ضرایب a_k و d_k مشخص شد. در ادامه رابطه b_k ها و e_k ها مشخص است:

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} W_4^0 & W_4^0 & W_4^0 & W_4^0 \\ W_4^0 & W_4^1 & W_4^2 & W_4^3 \\ W_4^0 & W_4^2 & W_4^0 & W_4^2 \\ W_4^0 & W_4^3 & W_4^2 & W_4^1 \end{bmatrix} \begin{bmatrix} x[1] \\ x[3] \\ x[5] \\ x[7] \end{bmatrix} = \begin{bmatrix} W_8^0 & W_8^0 & W_8^0 & W_8^0 \\ W_8^0 & W_8^2 & W_8^4 & W_8^6 \\ W_8^0 & W_8^4 & W_8^0 & W_8^4 \\ W_8^0 & W_8^6 & W_8^4 & W_8^2 \end{bmatrix} \begin{bmatrix} x[1] \\ x[3] \\ x[5] \\ x[7] \end{bmatrix}$$

$$\begin{bmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \end{bmatrix} = \begin{bmatrix} W_8^0\, b_0 \\ W_8^1\, b_1 \\ W_8^2\, b_2 \\ W_8^3\, b_3 \\ W_8^4\, b_0 \\ W_8^5\, b_1 \\ W_8^6\, b_2 \\ W_8^7\, b_3 \end{bmatrix} = \begin{bmatrix} W_8^0 & W_8^0 & W_8^0 & W_8^0 \\ W_8^1 & W_8^3 & W_8^5 & W_8^7 \\ W_8^2 & W_8^6 & W_8^2 & W_8^6 \\ W_8^3 & W_8^1 & W_8^7 & W_8^5 \\ W_8^4 & W_8^4 & W_8^4 & W_8^4 \\ W_8^5 & W_8^7 & W_8^1 & W_8^3 \\ W_8^6 & W_8^2 & W_8^6 & W_8^2 \\ W_8^7 & W_8^5 & W_8^3 & W_8^1 \end{bmatrix} \begin{bmatrix} x[1] \\ x[3] \\ x[5] \\ x[7] \end{bmatrix}$$

بنابراین ضرایبی که به دنبال آن‌ها بودیم به ترتیب زیر بدست آمدند:

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{bmatrix} = \begin{bmatrix} d_0 + e_0 \\ d_1 + e_1 \\ d_2 + e_2 \\ d_3 + e_3 \\ d_4 + e_4 \\ d_5 + e_5 \\ d_6 + e_6 \\ d_7 + e_7 \end{bmatrix} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} + \begin{bmatrix} W_8^0\, b_0 \\ W_8^1\, b_1 \\ W_8^2\, b_2 \\ W_8^3\, b_3 \\ W_8^4\, b_0 \\ W_8^5\, b_1 \\ W_8^6\, b_2 \\ W_8^7\, b_3 \end{bmatrix}$$

علاوه بر 32 ضرب قبلی، 8 ضرب نیز برای ضرب W_8^i ها لازم است که مجموعاً 40 ضرب به جای 64 ضرب را نتیجه می‌دهد. با این مثال علت کاهش یافتن تعداد عملیات‌ها روشن می‌شود و به طور کلی اگر $M(N)$ تعداد ضرب‌های لازم برای انجام عملیات بالا برای N نقطه باشد، داریم:

$$\begin{aligned} M(1) &= 0 \\ M(2) &= 2M(1) + 2 = 2 \\ M(4) &= 2M(2) + 4 = 2 \times 4 \\ M(8) &= 2M(4) + 8 = 3 \times 8 \\ M(16) &= 2M(8) + 16 = 4 \times 16 \\ M(32) &= 2M(16) + 32 = 5 \times 32 \\ M(64) &= 2M(32) + 64 = 6 \times 64 \\ M(128) &= 2M(64) + 128 = 7 \times 128 \\ &\cdots \\ M(N) &= (\log_2 N) \times N \end{aligned}$$

۱۱. خروجی این دو هیچ تفاوتی نداشته و در واقع FFT روشی سریع‌تر برای محاسبه DFT است. همان‌طور که در قسمت قبل مشاهده شد، در حالتی که از رابطه اصلی برای محاسبه DFT استفاده کنیم تعداد محاسبات از مرتبه $O(n^2)$ خواهد بود که n تعداد اعضای دنباله ورودی است. در صورتی‌که از FFT استفاده کنیم این تعداد از مرتبه $O(nlogn)$ خواهد شد که برای n های بزرگ تعداد محاسبات را به طور بسیار قابل ملاحظه‌ای کاهش می‌دهد. FFT یک الگوریتم ندارد و در واقع به هر روشی که تعداد محاسباتش از مرتبه فوق باشد FFT گفته می‌شود.

۱۲. تفاوت این دو در یک تغییر علامت نمایی مختلط از $-j$ به j و همزمان یک فاکتور $\frac{1}{n}$ است که در FFT حضور نداشت. پیاده‌سازی هر دو کاملا مشابه بوده و تنها تفاوت، دو مورد اخیر می‌باشد و زمان اجرای آن‌ها نیز یکسان است. البته تفاوت مفهومی آن‌ها محرز است و FFT از حوزه زمان به حوزه فرکانس منتقل می‌کند حال آن‌که $IFFT$ از حوزه فرکانس به حوزه زمان منتقل می‌کند.

۱۳. دستور $fftshift$ جای دو نیمه چپ و راست یک بردار را عوض می‌کند. وقتی از دستور fft تبدیل فوریه یک سیگنال را محاسبه کنیم، اگر بخواهیم فرکانس DC یا صفر در مرکز طیف باشد باید از دستور $fftshift$ استفاده کنیم.

۱۴. یک مدل پایه‌ای و کلی مورد پذیرش برای مدل کردن نویز حرارتی در کانال‌های مخابراتی فرض‌های زیر را در نظر می‌گیرد:

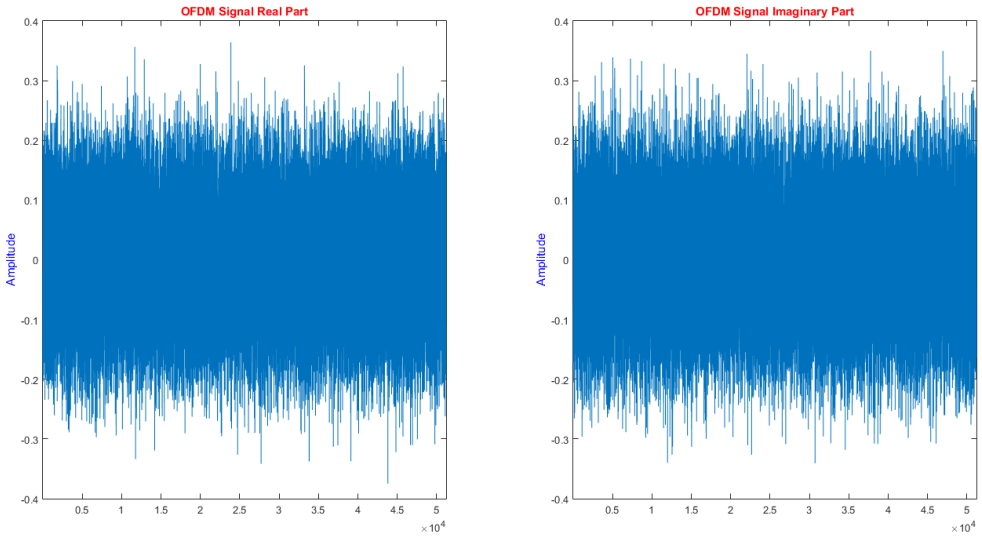
- نویز جمع‌شونده است به این معنی که سیگنال دریافتی در گیرنده حاصل جمع سیگنال ارسالی از فرستنده با یک نویز مستقل از سیگنال است.
- نویز سفید است به این معنی که چگالی طیف توان آن در همه فرکانس‌ها برابر است یا هم‌ارز آن اتوکوریلیشن نویز یک ضربه بوده و در هر شیفتی غیر از صفر، مقدار اتوکوریلیشن صفر است.
- نمونه‌های فرآیند تصادفی این نویز از چگالی توزیع گاوسی تبعیت می‌کنند.

معمولا فرض می‌شود که این کانال خطی و مستقل از زمان نیز می‌باشد.

۱۵. SNR به نسبت توان سیگنال به توان نویز گفته می‌شود که برحسب دسیبل تعیین می‌شود. این نسبت را می‌توان در ورودی یا خروجی گیرنده محاسبه کرد. BER یا همان $bit\ eror\ rate$ به نسبت تعداد بیت‌های ارسالی که در گیرنده دچار خطا شده‌اند به تعداد کل بیت‌های ارسالی گفته می‌شود. در صورتی که تعداد بیت‌های ارسالی زیاد باشد انتظار داریم این مقدار با $BEP(bit\ error\ proabability)$ تقریبا برابر شود. رابطه BER با SNR به صورت نزولی انتظار می‌رود زیرا انرژی هر بیت ارسالی زیاد شده و نسبت به نویز مقاوم‌تر می‌شود پس با احتمال کمتری دچار خطا شده و BER کاهش می‌یابد. رابطه ریاضی آن برای QAM به صورت زیر است.(E_b انرژی در هر بیت و N_0 چگالی طیف توان نویز است.)

$$BER = \frac{1}{2} erfc\left(\sqrt{\frac{E_b}{N_0}}\right), erfc(x) = 1 - \mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-t^2}\, dt$$

۱۶. مطابق بلوک دیاگرام عمل می‌کنیم. ابتدا دو بیت دو بیت مدولاسیون دیجیتال را انجام می‌دهیم. سپس این اعداد مدوله شده را به ۶۴ زیرحامل می‌فرستیم(با دستور $reshape$). یعنی اعداد ۱ تا ۶۴ دیجیتالی مدوله شده به ۶۴ زیرحامل رفته و به همین ترتیب تا انتها. در ادامه از هر کدام از این ستون‌های ۶۴ تایی تبدیل فوریه وارون می‌گیریم. نتیجه در $OFDMtimeDomain$ ذخیره می‌شود. این ماتریس ۸۰۰ ستون دارد که هر کدام ۶۴ نمونه زمانی سیگنال مختلط است. با دستور $reshape$ دوباره این داده‌ها را کنار هم می‌گذاریم(مطابق بلوک دیاگرام). این سیگنال در زیر رسم شده است.(با توجه به نبود اطلاعاتی برای D/A محور افقی شماره اندیس‌ها است و مفهوم زمانی ندارد.) ضمناً با توجه به ضرایب مختلط سیگنال زمانی نیز مختلط خواهد بود و برای رسم آن از دو نمودار مجزا برای قسمت‌های حقیقی و موهومی آن استفاده می‌کنیم.



شکل ۷

$PAPR$ برابر نسبت مجذور مقدار ماکزیمم موج به مجذور مقدار rms موج است. این مقدار به سادگی قابل محاسبه است. در هر بار اجرا طبیعتاً این مقادیر اندکی تغییر می‌کنند ولی حدود آن‌ها نزدیک مقادیر زیر است که در یک بار اجرا بدست آمدند.

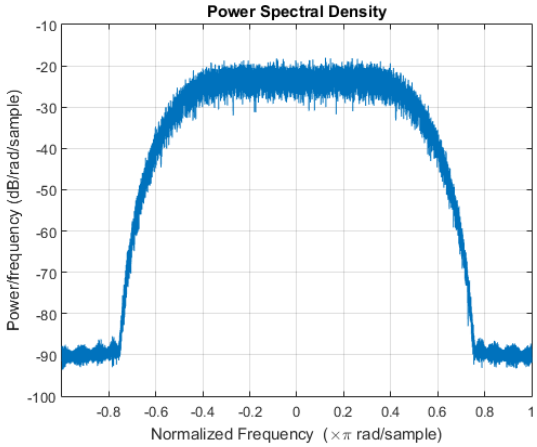
$$PAPR = 11.52 \equiv 10.61dB$$

$$PAPR\ for\ real\ part = 16.98 \equiv 12.30dB$$

$$PAPR\ for\ imag\ part = 15.66 \equiv 11.95dB$$

۱۷. تقویت‌کننده‌های خطی توان خروجی ماکزیممی می‌توانند تحویل دهند که در توان بیشتر از آن خطی عمل نمی‌کنند. بنابراین وجود پیک‌های بزرگ نسبت به متوسط سیگنال در صورتی‌که تقویت‌کننده برای متوسط تنظیم شده باشد که منطقی است، باعث بروز اثرات غیرخطی شده که مشخصا مطلوب نیستند. هم‌چنین بالا بودن $PAPR$ هم‌ارز کم بودن توان متوسط است و چون تعداد بیت‌های ارسالی در واحد زمان متناسب با توان متوسط سیگنال ارسالی است، این بالا بودن به منزله نرخ ارسال کمتر است که مطلوب نیست.

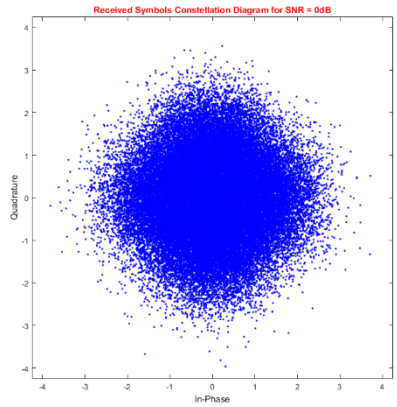
۱۸. نتیجه در تصویر زیر مشاهده می‌شود:

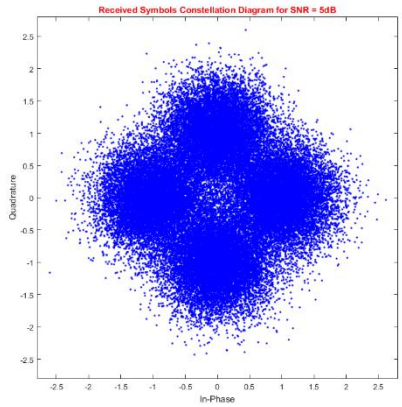


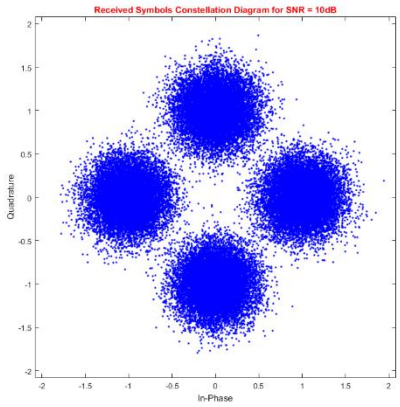
شکل ۸

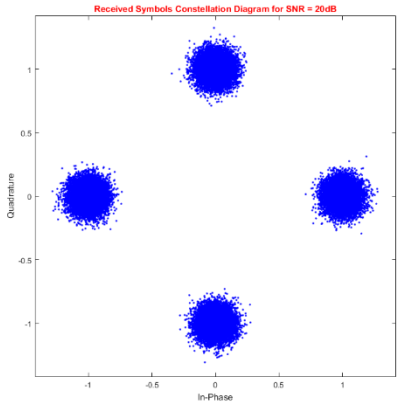
۱۹. همان‌طور که شکل پریودیک نشان می‌دهد در فرکانس‌های بالا توان افت می‌کند و یک روش معقول یافتن فرکانس نرمالایز شده‌ای است که انتگرال نمودار فوق تا آن در دو طرف فرکانس صفر برابر نصف انتگرال روی کل بازه فرکانسی شود(تعریف فرکانس قطع $3dB$). این فرکانس نرمال در صورتی‌که بدانیم فاصله زمانی نمونه‌ها چقدر است قابل تبدیل به فرکانس غیر نرمال و واقعی بوده و دو برابر آن پهنای باند استفاده شده را بدست می‌دهد.(معکوس فاصله زمانی نمونه‌برداری باید در فرکانس نرمال ضرب شود.)

۲۰. نمودار $constellation$ مشخصاً به SNR ربط دارد و ما برای چند SNR خاص آن را رسم می‌کنیم. ابتدا باید سایر مراحل را در گیرنده پیاده‌سازی کنیم. در ابتدا اثر کانال که نویز سفید جمع‌شونده گاوسی می‌باشد را پیاده‌سازی می‌کنیم. در ورودی اول آن سیگنال رد شده از کانال، در ورودی دوم مقدار SNR و در ورودی سوم کلمه $measured$ را می‌گذاریم زیرا در غیر این صورت مقدار پیش‌فرضی برای توان سیگنال در نظر می‌گیرد در حالی‌که توان سیگنال ارسالی عدد ثابتی نیست. سپس داده‌های سریال ورودی را با دستور $reshape$ مانند قبل به صورت موازی در می‌آوریم و از آن‌ها fft گرفته و سپس دوباره آن‌ها را سریال می‌کنیم. با دستور $scatterplot$ مانند قبل نمودار $constellation$ را رسم می‌کنیم. به ازای چند SNR مختلف نتیجه در زیر آورده شده‌است.(شکل ۹)



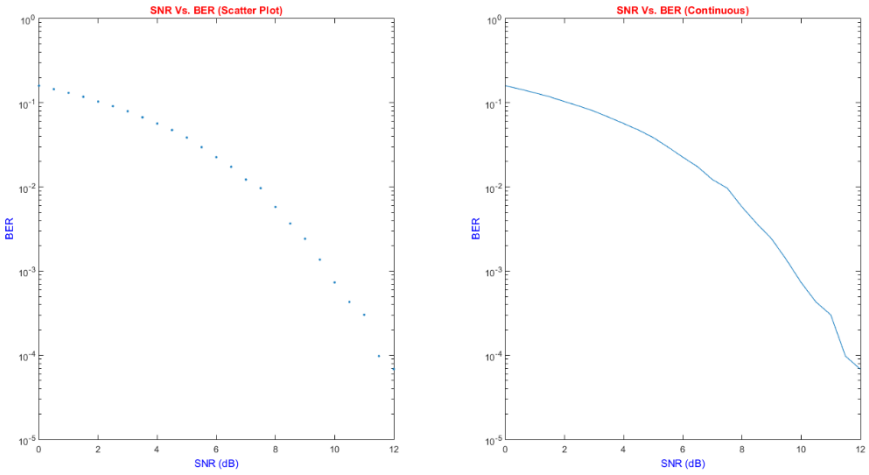






شکل ۹

۲۱. باید داده‌های رسیده به گیرنده را دمدوله کنیم. با دستور $pskdemod$ و با آرگومان‌های مشابه مدولاسیون آن این‌کار را انجام می‌دهیم. بعد از آن هر عدد را به دو بیت تبدیل می‌کنیم(کد را به گونه‌ای نوشته‌ام که برای M های غیر از ۲ نیز همین عملیات انجام شود.). سپس کل بیت‌های بازیابی شده را با بیت‌های ارسالی مقایسه می‌کنیم و تعداد بیت‌هایی که دچار خطا شده تقسیم بر کل تعداد بیت‌های ارسالی BER را بدست می‌دهد. تعداد بیت‌های دچار خطا شده با xor خروجی نهایی و داده‌های ارسالی اولیه بدست می‌آید. مجموع یک‌های این بردار همان تعداد خطاهاست. نتیجه در تصویر زیر مشخص است(شکل ۱۰):



شکل ۱۰

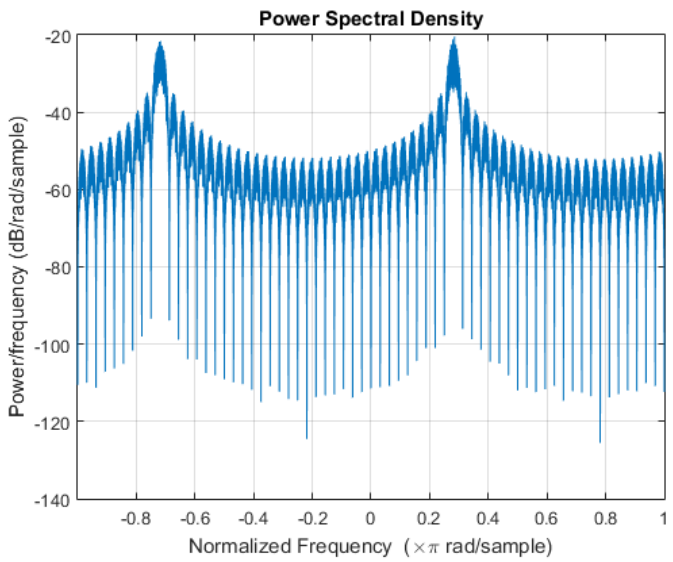
۲۲. برای Cable DVB-C: مقدار معمول SNR برابر $9dB$ و مقدار معمول BER 10^{-8} است. برای Terrestrial DVB-T هم همان مقادیر قبلی است. هم‌چنین جدول زیر مقادیر نوعی را نشان می‌دهد.

Required Base Band SNR		
SNR Requirements Versus Coding Rate and Modulation Scheme		
Modulation	Code Rate	SNR [dB]
QPSK	1/8	-5.1
	1/5	-2.9
	1/4	-1.7
	1/3	-1.0
	1/2	2.0
	2/3	4.3
	3/4	5.5
	4/5	6.2
16 QAM	1/2	7.9
	2/3	**11.3**
	3/4	12.2
	4/5	12.8
64 QAM	2/3	15.3
	3/4	17.5
	4/5	18.6

جدول 1

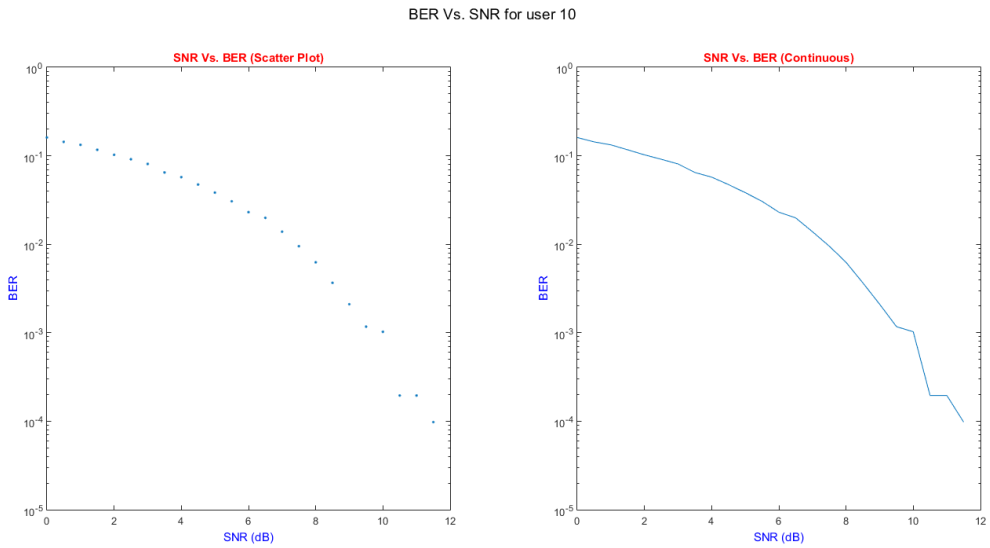
۲۳. در صورتی‌که تعامد کامل نباشد یا از کامل بودن انحراف زیادی پیدا کند BER زیاد می‌شود چون احتمال خطا به خاطر هم‌پوشانی بالا می‌رود. هم‌چنین اگر $Gaurd\ Interval$ استفاده نشود یا به اندازه کافی نباشد اثرات تاخیر و پژواک باعث تداخل شده و BER را افزایش می‌دهند. تداخل‌های ناشی از امواج ناخواسته دیگر نیز باعث افزایش BER می‌شوند که با محدود کردن پهنای باند ارسالی می‌توان آن‌را کاهش داد(که البته ممکن است باعث کاهش $throughput$ شود.) افزایش توان سیگنال ارسالی باعث افزایش توان هر بیت شده اما باید مراقب بود باعث تداخل با کاربران دیگر نشویم و هم‌چنین این افزایش توان بر تقویت‌کننده‌هایی که استفاده می‌کنیم نیز اثر مستقیم دارد. هم‌چنین بر باتری و مصرف و توان. مدولاسیون‌های مرتبه پایین BER بهتری دارند اما باید توجه داشت در صورت استفاده از آن‌ها $throughput$ متناظراً کم می‌شود. کاهش پهنای باند برای کم کردن اثر نویز هم در کاهش BER مفید است اما مشابه قبل $throughput$ کاهش می‌یابد.

۲۴. در این حالت برای کند نشدن بیش از حد برنامه تعداد داده‌ها را به $۲۰KB$ کاهش می‌دهیم. سیگنال $OFDM$ را بدست می‌آوریم. در واقع پس از مدوله کردن داده‌های هر کاربر بین آن‌ها صفر می‌گذاریم تا عملیات قسمت‌های قبل قابل انجام شود. برای مثال برای کاربر ۱۰ یک بردار جدید می‌سازیم که همه درایه‌های آن صفرند و طول ۳۲۷۶۸۰(هر دو سمبل در یک ستون ۶۴ تایی قرار می‌گیرند و کل سمبل‌ها برای حالتی که با دو بیت مدوله کنیم ۱۰۲۴۰ تا هستند پس کل طول این بردار جدید باید ۳۲۷۶۸۰ = ۳۲×۱۰۲۴۰ باشد.) خواهد بود. البته کد به گونه‌ای نوشته شده که با هر مقدار M سازگار باشد و کار کند. این بردار بزرگ با دستور $reshape$ به ستون‌های ۶۴ تایی تبدیل می‌شود که فقط سطر ۱۰ و ۴۲ در هر ستون مقدار غیر صفر دارد. سپس عملیات قسمت‌های قبل به آن اعمال می‌شود.($IFFT, parallel\ to\ serial$) در نهایت برای هر کاربر یک سیگنال سریال نهایی $OFDMtimeDomain$ داریم که این‌ها باید با هم جمع شده و از کانال عبور کنند. سیگنال $OFDMtimeDomainFinal$ حاصل جمع این‌ها در هوا است. برای این قسمت چگالی طیف توان سیگنال $OFDM$ برای کاربر شماره ۱۰ رسم می‌شود.(شکل ۱۱)



شکل ۱۱

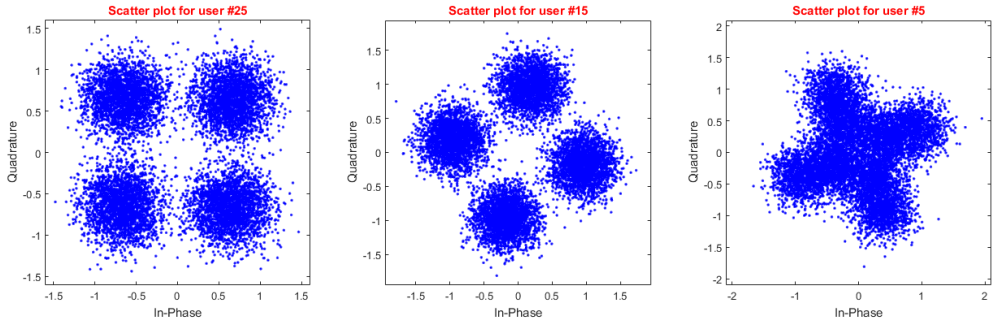
۲۵. همه عملیات مشابه سوال ۲۱ است. سیگنال نهایی قسمت قبل با نویز جمع شده، موازی شده، از آن fft گرفته می‌شود و کانال‌های مربوط به کاربر مورد نظر برداشته می‌شوند و سریال می‌شوند سپس دمدوله شده و به روش قبل BER محاسبه می‌شود. نتیجه در تصویر زیر منعکس شده‌است.(شکل ۱۲)



شکل ۱۲

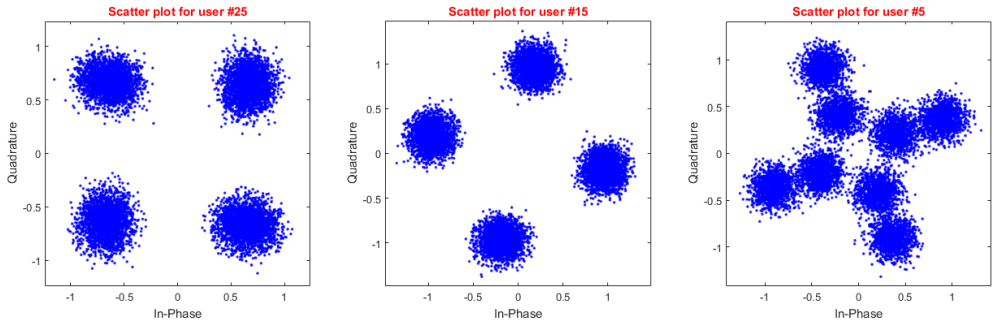
۲۶. با دستور dot مقدار ضرب داخلی را حساب می‌کنیم و این مقدار بسیار کم است و می‌توان با اطمینان زیادی گفت تعامد وجود دارد.

۲۷. در صورتی‌که فرض کنیم نویز هم اضافه می‌شود($SNR = 10$) و این دو اثر را با هم بررسی کنیم نتیجه به صورت زیر خواهد بود:



شکل ۱۳

اگر اثر تاخیر به تنهایی را در نظر بگیریم و نویزی در کانال اضافه نکنیم نتیجه به صورت زیر می‌شود:



شکل ۱۴

۲۸. با در نظر گرفتن $SNR = 10$ و اثر تاخیر همزمان خواهیم داشت:

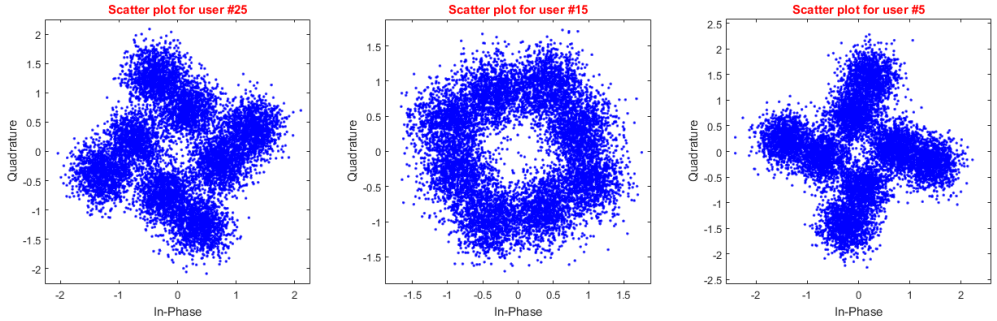
$$BER\ user\ \#5 = 0.4389\ , BER\ user\ \#15 = 0.4987, BER\ user\ \#25 = 0.4652$$

بدون در نظر گرفتن اثر نویز کانال، تاثیر تاخیر به صورت زیر خواهد بود:

$$BER\ user\ \#5 = 0.4713\ , BER\ user\ \#15 = 0.5000, BER\ user\ \#25 = 0.4390$$

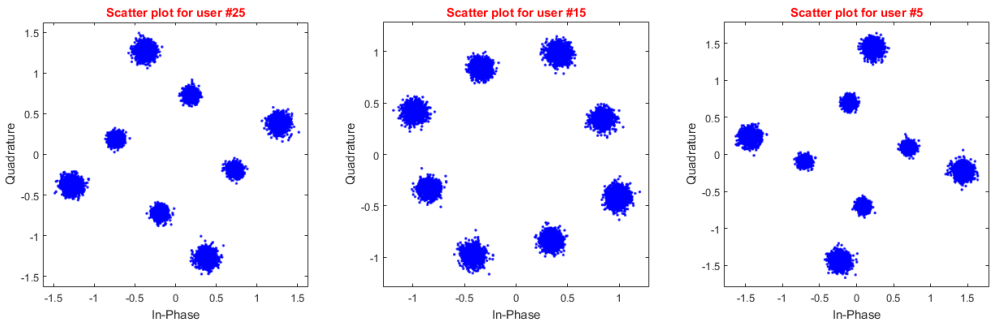
۲۹. به صورت کلی از تخمین‌گرهای آفست زمانی و فرکانسی با اضافه کردن پیشوندهای چرخه‌ای استفاده می‌شود. در بعضی سیستم‌های بی‌سیم به پایلوت هم نیاز است. روش کلی در واقع تخمین کانال است و جبران اثر آن است.

۳۰. با در نظر گرفتن $SNR = 10$ و اثر ISI همزمان خواهیم داشت:



شکل ۱۵

بدون در نظر گرفتن اثر نویز کانال، تاثیر ISI به صورت زیر خواهد بود:



شکل ۱۶

۳۱. با در نظر گرفتن $SNR = 10$ و اثر ISI همزمان خواهیم داشت:

$$BER\ user\ \#5 = 0.0111\ , BER\ user\ \#15 = 0.0319, BER\ user\ \#25 = 0.0176$$

بدون در نظر گرفتن اثر نویز کانال، تاثیر ISI به صورت زیر خواهد بود:

$$BER\ user\ \#5 = 0\ , BER\ user\ \#15 = 0, BER\ user\ \#25 = 0$$

۳۲. یکی از روش‌ها که ما در این تمرین از آن استفاده نکردیم استفاده از بازه‌های نگهبان یا $Gaurd\ Interval$ ها بین سمبل‌ها است. به این ترتیب احتمال تداخل سمبل‌ها را کاهش می‌دهیم. هم‌چنین برای جبران این خطا از فیلترهای جبران‌ساز خطی یا غیرخطی استفاده می‌شود. البته نوع خطی باعث افزایش نویز شده و معمولا در کاربردها از نوع غیرخطی استفاده می‌شود و معروف ترین آن‌ها $Descision - feedback\ equalization(DFE)$ می‌باشد. برای کانال‌هایی که SNR پایینی دارند روش بهینه $maximum\ likelihood\ sequence\ estimation(MLSE)$ است که البته پیچیدگی آن با افزایش دنباله تاخیر به طور نمایی افزایش می‌یابد. هم‌چنین این فیلترها به دو دسته $symbol - by - symbol$ و یا $sequence\ estimator(SB)$ دسته‌بندی می‌شوند.