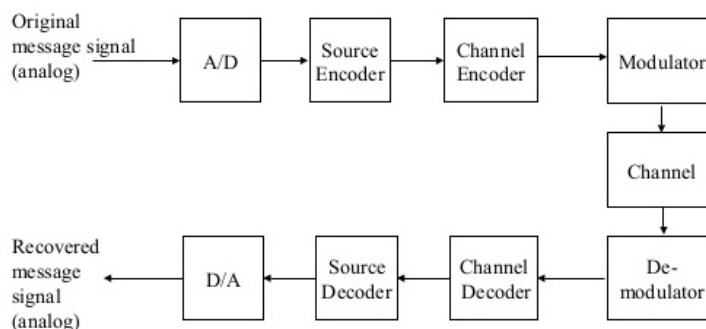


در این پروژه هدف انتقال تصویر توسط سیستم مدولاسیون دیجیتال است. بلوک دیاگرام یک سیستم مدولاسیون دیجیتال به صورت شکل زیر است:



در این پروژه به علت دیجیتال بودن تصاویر ورودی، بلوک‌های A/D و D/A پیاده‌سازی نمی‌شوند. همچنین برای سادگی بلوک‌های Channel Encoder و Channel Decoder نیز بررسی نمی‌شوند؛ لذا سیستم نهایی قابلیت تصحیح خطا ندارد و در صورت بروز خطا در ارسال و دریافت حتی یک بیت، عملکرد سیستم مختل می‌شود. یعنی در مواردی که بازبایی تصویر ارسالی در گیرنده مد نظر است، میزان نویز شونده با سیگنال ارسالی باید کمتر از مقدار مشخصی باشد تا اطمینان حاصل شود که عملکرد سیستم دچار اختلال نخواهد شد.

در ابتدا تصاویر دیجیتال داده شده را توسط دستور `imread` می‌خوانیم، سپس با استفاده از دستور `imresize` ابعاد آن را از یک ماتریس  $512 \times 512$  به یک ماتریس  $64 \times 64$  تقلیل می‌دهیم. در هر زمانی که به نمایش تصویر نیاز باشد، با استفاده از دستور `imshow`، آن تصویر را می‌کشیم.

## فاز اول

### بلوک‌های Source Decoder و Source Encoder

نقش این بلوک تبدیل دنباله سمبل‌ها به دنباله باینری است. در این پروژه از الگوریتم کدگذاری هافمن استفاده می‌شود. توضیحات مربوط به این روش در دستور کار آمده است.

اولین قدم برای پیاده سازی بلوک‌های Source Decoder و Source Encoder، تشکیل درخت هافمن و ساختن جدول کد طول متغیر از روی آن است.

با استفاده از `Data type` استراکت (`struct`) ابتدا درخت هافمن را با شروع از برگ‌ها و حرکت به سمت ریشه، بر اساس احتمال رخداد سمبل‌ها تشکیل می‌دهیم. سپس با جاروب کردن درخت ایجاد شده از ریشه به سمت برگ‌ها، جدول کد طول متغیر مربوط به آن را می‌سازیم. در این پروژه سمبل‌ها اعداد ۰ تا ۲۵۵ هستند.

به عنوان مثال برای تصویر شماره ۱، جدول کد طول متغیر برای ۱۶ سمبل پر تکرار به صورت زیر خواهد بود:

	1 Symbol	2 Code
1	155	"010101"
2	158	"1110100"
3	162	"1110000"
4	126	"1101001"
5	104	"1101000"
6	146	"1100111"
7	123	"1100110"
8	119	"1011111"
9	113	"1011110"
10	107	"1011101"
11	161	"1011011"
12	103	"1011010"
13	159	"1010101"
14	148	"1010100"
15	106	"1010011"
16	100	"1010010"

پس از تهیه جدول کد طول متغیر، با حرکت روی ماتریس پیکسل‌های تصویر دیجیتالی ارسالی و ساختن رشته‌ای که در آن کدهای مربوط به پیکسل‌های تصویر به ترتیب و بدون فاصله، بلوک Source Encoder را پیاده‌سازی می‌کنیم.

در بلوک Source Decoder نیز عکس این عمل را انجام می‌دهیم؛ یعنی با حرکت روی رشته دریافتی و مطابقت دادن آن جدول کد طول متغیر، مقدار پیکسل‌های تصویر را تشخیص می‌دهیم.

لازم به ذکر است که اگر در فرآیند ارسال و دریافت رشته، حتی یک بیت دچار خطا شود، عملکرد سیستم مختل می‌شود و دیگر توانایی بازیابی درست و کامل تصویر را ندارد. (پیکسل‌هایی که کد آن‌ها قبل از محل بروز اولین بیت خطا باشد، به درستی بازیابی خواهند شد)

تصویر شماره ۱

تصویر ارسالی:



تصویر دریافتی:





با استفاده از تصویرهای مختلف، صحت عملکرد بلوک‌های Source Encoder و Source Decoder به طور کامل بررسی شد.

### بلوک‌های Modulator و Demodulator

در این پروژه برای مدولاسیون و دمدولاسیون از روش M-ary PSK استفاده شده است. برای توضیح این روش فرض کنید در ورودی دنباله‌ای از سمبل‌ها داشته باشیم و تعداد حالت‌های ممکن برای سمبل‌ها M است. در این صورت به هریک از این حالت‌ها یک سیگنال خاص را نسبت می‌دهیم. این سیگنال‌ها عبارت‌اند از:

$$S_m = \begin{cases} \sqrt{\frac{2}{T_s}} \cos\left(2\pi f_c t + \frac{(2m+1)\pi}{M}\right) & 0 \leq t \leq T_s \\ 0 & \text{otherwise} \end{cases} \quad m = 0, \dots, M-1$$

تفاوت این سیگنال‌ها در فاز آن‌ها است. بدین صورت که بازه  $[0, 2\pi]$  را به M بازه مساوی تقسیم می‌کنیم و میانگین هر بازه را به یکی از حالت‌ها نسبت می‌دهیم.

بعد از تعیین سیگنال مربوط به هر یک از حالت‌ها، باید از روی رشته خروجی بلوک Source Encoder، سیگنال مدوله شده برای ارسال از طریق کانال را تولید کنیم. برای این کار باید M توانی از ۲ باشد. تا بتوان هر  $v = \log_2 M$  را یک سمبل با M حالت ممکن در نظر گرفت. سپس از روی دنباله سمبل‌های بدست آمده، سیگنال مدوله شده را ایجاد می‌کنیم.

در گیرنده با فرض ایده‌آل بودن کانال، سیگنال مدوله شده بدون تغییر دریافت می‌شود. حال برای دمدولاسیون این سیگنال ابتدا باید سیگنال دریافتی در بازه‌هایی به طول  $T_s$  بریده شود. هر یک از این برش‌ها معرف یک سمبل ارسالی است. برای تشخیص سمبل ارسالی از یک همبستگی سنج استفاده می‌کنیم. نحوه عملکرد بدین صورت است که همبستگی سیگنال دریافتی با دو پایه متعامد یک‌ه یعنی سینوس و کسینوس با فرکانس برابر با فرکانس حامل محاسبه می‌شود. سپس با محاسبه فاز مربوط به این دو مولفه، در مورد فاز سیگنال ارسالی تصمیم‌گیری می‌شود.

بلوک‌های Modulator، Demodulator و Detector بر این اساس پیاده‌سازی می‌شوند.

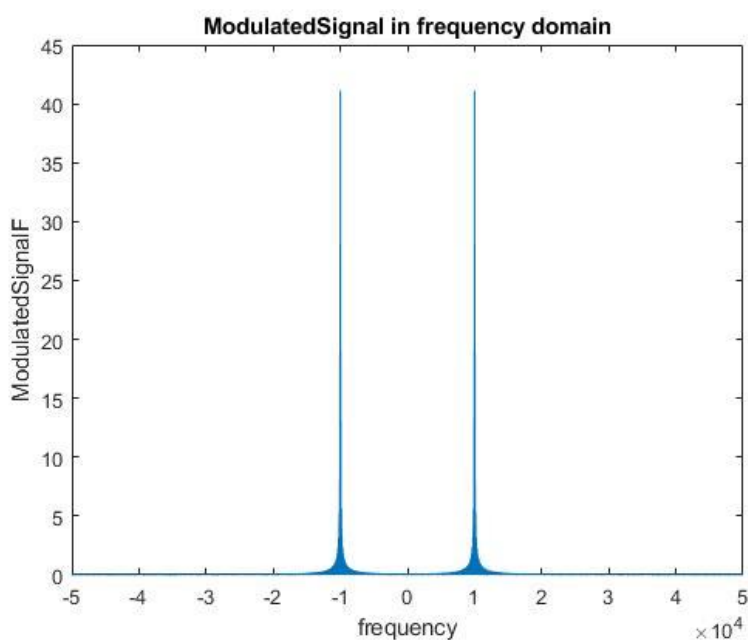
## فاز دوم

الف) با اتصال بلوک‌های ساخته شده در فاز اول، سیستم انتقال تصویر را کامل می‌کنیم.

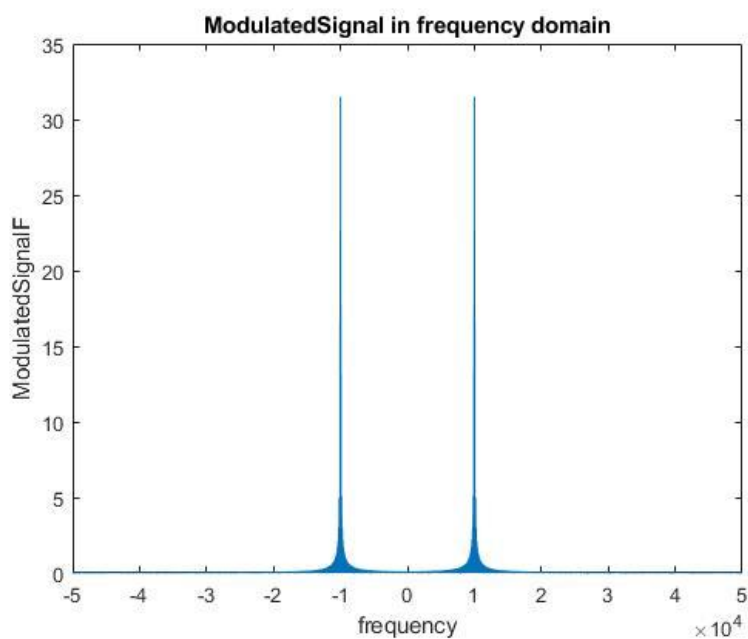
ب) برای محدود کردن پهنای باند سیگنال عبوری، از دستور `bandpass` استفاده می‌کنیم. این فیلتر روی سیگنال مدوله شده و قبل از دمدولاسیون اعمال می‌شود.

پ) با استفاده از دستورهایی `fft` و `fftshift`، طیف سیگنال عبوری از کانال را رسم می‌کنیم.

تصویر شماره ۱



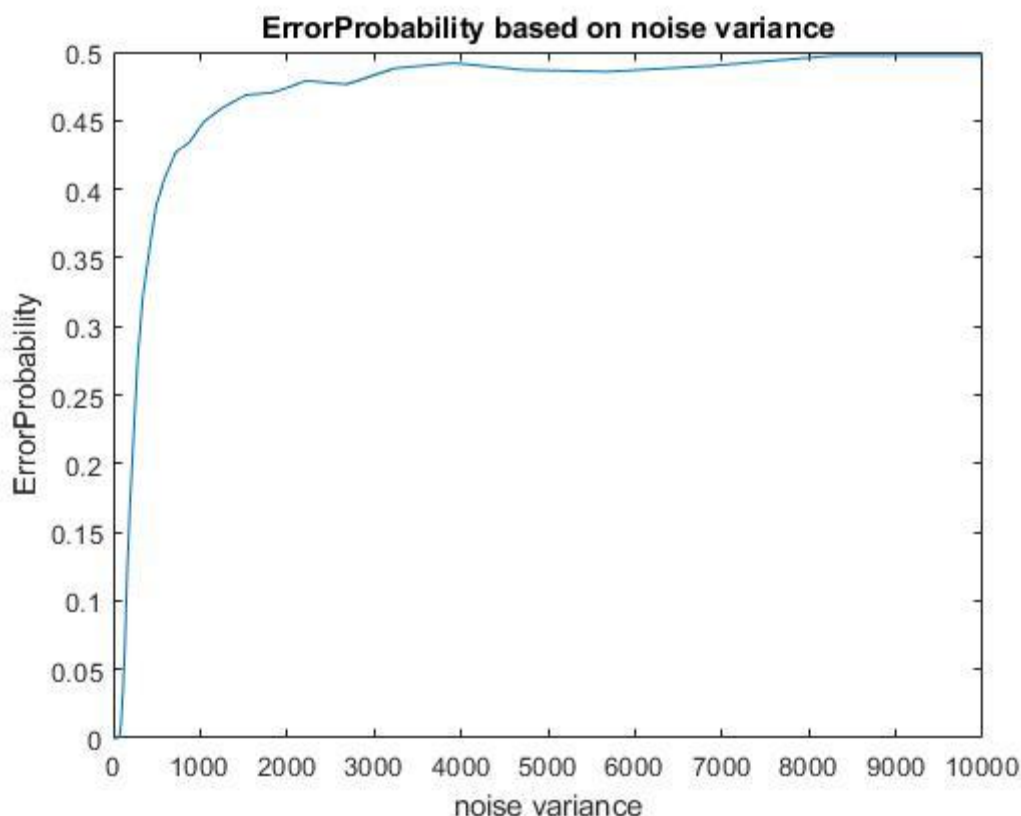
تصویر شماره ۲



برای محاسبه پهنای باند میانگین، پهنای باند مربوط به هر تصویر را با استفاده از تقریب ۹۹ درصد انرژی، بدست می‌آوریم و سپس با گرفتن میانگین از این مقادیر، پهنای باند میانگین را محاسبه می‌کنیم.

$$\text{average BW} = 909.45 \text{ Hz}$$

ت) فرض می‌کنیم نویز جمع شونده از نوع AWGN است و با استفاده از دستور `normrnd` آن را تولید می‌کنیم. باید در نظر داشت که با افزایش مقدار واریانس نویز شونده احتمال خطا افزایش می‌یابد و در صورت بروز خطا دیگر امکان بازیابی تصویر ارسالی وجود ندارد. اما بر اساس تعداد بیت‌های متفاوت در رشته‌های ارسالی و دریافتی می‌توان احتمال خطا را بدست آورد. برای نشان دادن احتمال خطا بر حسب واریانس نویز، با استفاده از دستور `logspace`، واریانس نویز را از ۱ تا ۱۰۰۰۰ با فاصله‌های نابرابر تغییر می‌دهیم.



بر اساس نمودار رسم شده، احتمال خطا در حالت حدی به ۵۰ درصد میل می‌کند. زیرا با فرض  $M = 4$  و بسیار بزرگ بودن واریانس نویز، صفحه تشخیص به چهار قسمت با احتمال تشخیص برابر تقسیم می‌شود. اگر ناحیه مربوط به سیگنال به درستی تشخیص داده شود، خطا نداریم. اما اگر این ناحیه به اشتباه تشخیص داده شود، در دو ناحیه یک بیت خطا و در یک ناحیه دو بیت خطا داریم. در این صورت احتمال غلط بودن دو بیت تشخیصی در ارسال دو بیت برابر است با:

$$P(\text{خطا در ارسال و دریافت یک بیت}) = \frac{1}{4} * 0 + \frac{1}{4} * \frac{1}{2} + \frac{1}{4} * \frac{1}{2} + \frac{1}{4} * 1 = \frac{1}{2}$$

پس در حال حدی، احتمال خطا به ۵۰ درصد باید میل کند.

ث) برای محاسبه SNR قبل از دمدولاسیون، سیگنال مدوله شده و نویز جمع شونده AWGN را به صورت جداگانه از کانال عبور می‌دهیم و سپس با محاسبه توان مربوط به هر کدام در گیرنده، SNR قبل از مدولاسیون را بدست می‌آوریم.

برای محاسبه SNR بعد از دمدولاسیون، فرض می‌کنیم واریانس نویز جمع‌شونده به صورتی است که با تقریب خوبی می‌توان، توان سیگنال قبل و بعد از دمدولاسیون را یکسان در نظر گرفت. سپس برای محاسبه توان نویز بعد از دمدولاسیون، سیگنال دمدوله شده را دوباره به مدوله می‌کنیم و اختلاف آن با سیگنال مدوله شده اصلی را به عنوان سیگنال نویز در نظر می‌گیریم و با محاسبه توان نویز، SNR بعد از مدولاسیون را بدست می‌آوریم.

برای مقادیر مختلف واریانس نویز، خروجی‌های این بخش متفاوت خواهد بود. لذا در ادامه واریانس نویز را برابر ۱۰۰ در نظر می‌گیریم.

با گرفتن میانگین از مقادیر بدست آمده برای هر تصویر داریم:

$$\text{average SNR}_i = -10.74 \text{ dB}$$

$$\text{average SNR}_o = 12.90 \text{ dB}$$

در صورتی که واریانس به مقدار کافی کوچک باشد، دمدولاسیون می‌تواند نویز را به طور کامل از بین ببرد و می‌توان تصویر ارسالی را به درستی بازیابی کرد.

برای مثال اگر واریانس نویز را برابر ۱ در نظر بگیریم، داریم:

تصویر شماره ۱

تصویر ارسالی:



تصویر دریافتی:



تصویر شماره ۲

تصویر ارسالی:

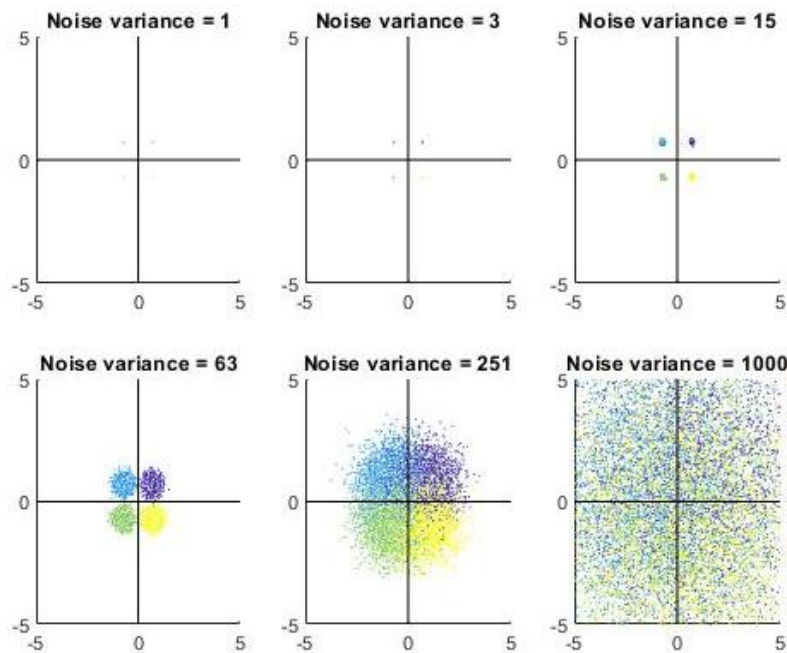


تصویر دریافتی:



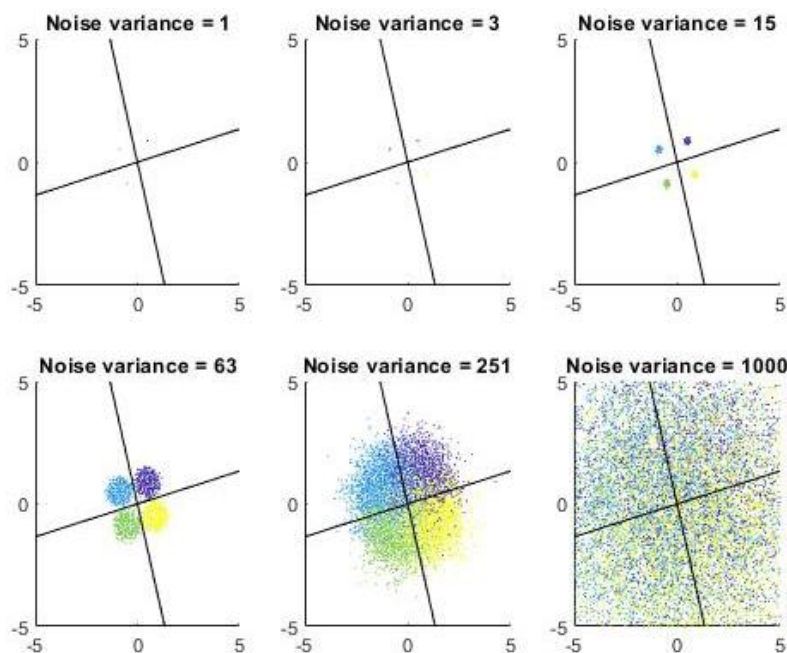
اما اگر واریانس نویز از حدی بزرگتر باشد، سیستم دمدولاسیون دچار مشکل می‌شود و امکان بازیابی تصویر ارسالی وجود نخواهد داشت.

ج) با استفاده از دستور `logspace`، ۶ مقدار برای واریانس نویز بین ۱ تا ۱۰۰۰ با فاصله‌های نابرابر انتخاب می‌کنیم. به وضوح برای  $M = 4$ ، محورهای  $x$  و  $y$ ، آستانه‌های تصمیم‌گیری بهینه `Detector` هستند.

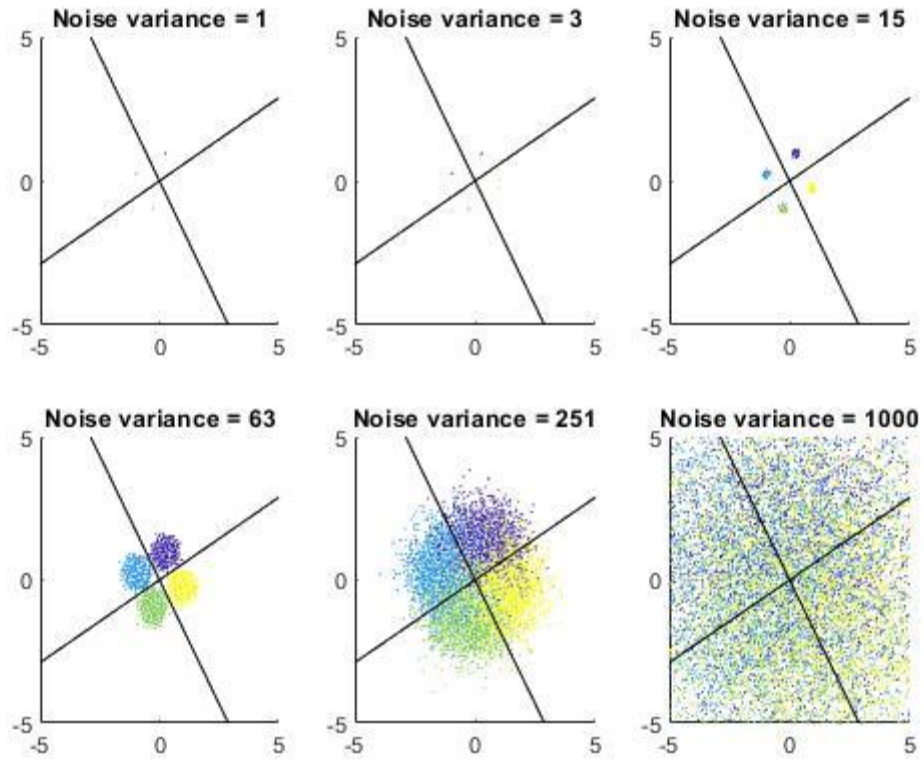


همان طوری که از نمودارهای رسم شده مشخص است، با افزایش واریانس نویز جمع شونده، مکان‌های تشخیص داده شده از مکان‌های اصلی فاصله می‌گیرند. به طوری که تقریباً برای واریانس نویز برابر ۸۰، امکان تشخیص درست از بین می‌رود. (چ) در این حالت مکان‌های تشخیص داده شده با زاویه  $\varphi$  حول مبدا مختصات دوران می‌کنند. لذا خطوط مربوط به آستانه تصمیم‌گیری نیز باید با همین زاویه دوران کنند.

$$\varphi = \frac{\pi}{12}$$



$$\varphi = \frac{\pi}{6}$$



$$\varphi = \frac{\pi}{3}$$

