

آزمایشگاه مخابرات دیجیتال پیشرفته

فصل سه

کد گذاری کانال

(شامل ۲ جلسه آزمایشگاه مجازی)

۱۳۹۹-۱۴۰۰ (نیم سال اول)

نسخه دانشجو

فصل ۳: کدگذاری کانال

۱- اهداف:

هدف اصلی آزمایش آشنایی مقدماتی با بلوک های کدگذار و کد بردار کانال، نحوه عملکرد، انواع آن و تأثیر آن در بهبود نرخ خطای بیت و مقابله با خطا در کانال مخابراتی است. در انجام این آزمایش اهداف زیر دنبال می شود:

- آشنایی با مفهوم کدگذاری کانال، افزونگی و چندگانگی زمانی
- آشنایی با کدهای بلوکی
- آشنایی با کدهای کانولوشن به عنوان کدهای پرکاربرد در سیستمهای مخابراتی و کدبردار ویتربی به عنوان یک روش معمول کدگذاری کانولوشن، آشنایی با پیاده سازی نرم افزاری کدینگ کانولوشن
- درک تفاوت موجود بین کدهای بلوکی و کانولوشن و معایب و مزایای آنها بر یکدیگر
- آشنایی با کد همینگ به عنوان یک کد بلوکی نمونه

۲- تجهیزات و نرم افزارهای موردنیاز:

نرم افزار: Matlab

در این آزمایش از GUI به عنوان ابزاری مؤثر در مشاهده نتایج آزمایش در نرم افزار Matlab استفاده می کنیم.

۳- پیش نیازها

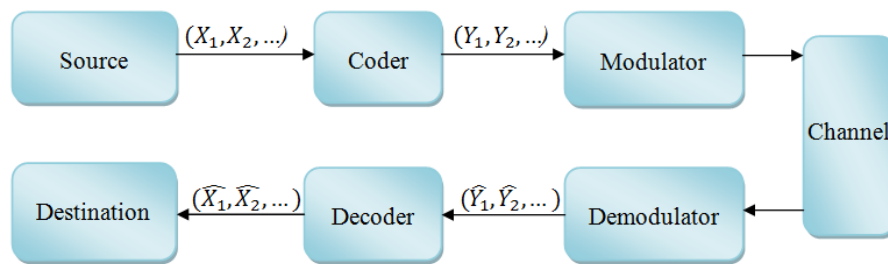
آشنایی با مبانی سیستمهای مخابراتی دیجیتال، آشنایی مقدماتی با نرم افزار MATLAB.

۴- پیش آزمایش

۴-۱- مطالعه

در یک سیستم مخابراتی دیجیتال، سمبل های فرستنده، پس از تبدیل به دنباله صفر و یک، توسط یک پالس مناسب شکل دهی شده و (در صورت نیاز پس از مدوله شدن) بر روی کانال ارسال می گردند. سیگنال ارسالی بر روی کانال، معمولاً در معرض اغتشاش، اعوجاج و تداخل قرار دارد و حین عبور از کانال دچار تغییر می شود. بنابراین گیرنده در تشخیص و آشکارسازی (برخی از) سمبل های ارسالی توسط گیرنده دچار اشتباه خواهد شد. اما این خطاها می تواند در گیرنده تصحیح شود. این عمل توسط کدگذاری انجام می پذیرد. کدگذاری کنترل خطا عبارتست از انجام عملیاتی بر روی سمبل های ارسالی به منظور مقابله با خطاهای ایجاد شده در کانال. به همین دلیل به آن کدگذاری کانال نیز گفته می شود. کدگذاری کانال کاربردهای بسیاری از جمله در مخابرات ماهواره ای، بی سیم و شبکه های کامپیوتری دارد.

هدف از کدگذاری کانال، قرار دادن افزونگی‌هایی بر روی اطلاعات ورودی است تا آن را در برابر تغییرات کانال مقاوم‌تر سازد. واضح است که این عمل نرخ ارسال اطلاعات مفید را کاهش می‌دهد، ولی در عوض قابلیت اطمینان سیستم را افزایش خواهد داد. شکل (۱) بلوک دیاگرام یک سیستم مخابراتی نوعی و جایگاه کدگذار و کدبردار کانال را در آن نمایش می‌دهد.



شکل (۱): جایگاه کدگذار و کدبردار کانال در یک سیستم مخابراتی

کدگذار کانال، اطلاعات ورودی را دریافت $(X_1 X_2 \dots)$ و با انجام عملیاتی آن را به دنباله دیگری $(Y_1 Y_2 \dots)$ تبدیل می‌کند. همانطور که گفته شد، معمولاً طول دنباله خروجی بلوک کدگذار کانال، بلندتر از طول دنباله ورودی آن است. این افزونگی به گیرنده کمک می‌کند که خطاهای این دنباله را تشخیص داده و یا تصحیح نماید. به مثال زیر دقت کنید؛ فرض کنید فرستنده‌های سمبل‌های A, E, I و O را می‌فرستد. برای ارسال هر کدام از این سمبل‌ها به دو بیت نیاز است. همچنین فرض کنید کدگذار کانال در هر لحظه دو بیت را بعنوان ورودی دریافت و سه بیت را به عنوان خروجی تحویل می‌دهد. به عبارت دیگر، خروجی کدگذار کانال سمبل‌ها (کلمات کد) سه بیتی هستند. از هر سه بیت خروجی، دو بیت اول همان دو بیت ورودی کدگذار و بیت سوم به گونه‌ای انتخاب می‌شود که تعداد یک‌ها در سمبل خروجی کدگذار زوج باشد. به این بیت، بیت توازن^۱ گفته می‌شود. عملیات فوق یک نوع کدگذاری کانال است که به آن کدگذاری بررسی توازن^۲ نیز می‌گویند. جدول (۱) نحوه عملکرد این کدگذار را نمایش می‌دهد.

جدول (۱): کدگذاری بررسی توازن

<i>Input Vowel</i>	<i>Input Binary</i>	<i>Output Binary</i>
<i>A</i>	<i>00</i>	<i>000</i>
<i>E</i>	<i>01</i>	<i>011</i>
<i>I</i>	<i>10</i>	<i>101</i>
<i>O</i>	<i>11</i>	<i>110</i>

^۱ Parity bit

^۲ Parity check coding

بلوک کدبردار در گیرنده، پس از دریافت هر یک از کلمات کد، تعداد یک‌های موجود در آن را می‌شمارد. در صورتی که این تعداد زوج باشد، گیرنده بنا را بر این می‌گذارد که کلمه کد بدون خطا دریافت شده و بنابراین با حذف بیت آخر (بیت توازن) سمبل دو بیتی را تحویل می‌دهد. اما در صورتی که تعداد یک‌ها در کلمه کد سه بیتی دریافتی فرد باشد، گیرنده تشخیص می‌دهد که در کلمه کد مزبور خطایی رخ داده است، ولی تعیین اینکه کلمه کد صحیح بوده است (و یا عبارت دیگر خطا در کدام بیت اتفاق افتاده) از عهده آن خارج است. بنابراین، کدگذاری بررسی توازن تنها امکان تشخیص خطا را دارد و از تصحیح خطا عاجز است.

کدهای مختلف در چگونگی افزودن بیت‌های افزونگی متفاوت هستند. بطور کلی، معیارهایی که کدهای مختلف توسط آنها مقایسه می‌شوند عبارتند از :

۱- دقت کدگذاری

۲- راندمان کدگذاری

۳- بهره کدگذاری

الف- دقت کدگذاری :

وظیفه بلوک کدبردار، بازیابی اطلاعات است. هرچه این اطلاعات بازیابی شده به اطلاعات ارسالی فرستنده نزدیکتر باشد، دقت کدگذاری (و مقابله آن با خطا) بیشتر بوده است. این معیار با احتمال خطای بیت (BER) سنجیده می‌شود :

$$P_{BER} = \frac{1}{K} \sum_{i=1}^k \Pr(\hat{W}_i - W_i)$$

ب- راندمان کدگذاری :

کدهای مختلف به منظور مقابله با خطا، بیت‌های افزونگی به اطلاعات اصلی فرستنده اضافه می‌کنند. هرچه تعداد این بیت‌های افزونگی در هر سمبل فرستنده کمتر باشد، کارایی کد بالاتر خواهد بود. راندمان کد توسط پارامتری به نام نرخ کد تعیین می‌شود. اگر هر سمبل ورودی کدگذار حاوی k بیت و هر سمبل خروجی آن حاوی n بیت باشد (یعنی $k = n$ + بیت‌های افزونگی) آنگاه نرخ کد عبارتست از :

$$R = \frac{k}{n}$$

هرچه نرخ کد بیشتر باشد، راندمان آن بالاتر است. بنابراین با کاهش تعداد بیت‌های افزونگی می‌توان راندمان یک کد را افزایش داد ولی در عوض قدرت آن در مقابله با خطا کاهش خواهد یافت. به عبارت دیگر، بین کارایی

کد و قدرت تشخیص و یا تصحیح خطای آن، یک تعادل وجود دارد و نرخ یک کد، به نوعی نشان‌دهنده قدرت تشخیص و یا تصحیح آن کد نیز می‌باشد.

ج- بهره کد گذاری :

بهره کد معیاری است از میزان کاهش نسبت سیگنال به نویز (SNR) در حالتی که از کد گذاری استفاده می‌شود، نسبت به حالتی که از آن استفاده نمی‌شود، برای دستیابی به کارایی یکسان. به عنوان مثال، فرض کنید در یک سیستم که از کد گذار استفاده نمی‌کند، با SNR برابر $3dB$ ، احتمال خطای بیت 10^{-5} بدست می‌آید. حال فرض کنید استفاده از نوعی کد گذاری کانال باعث شود که احتمال خطای بیت 10^{-5} در SNR برابر $1.4 dB$ به دست آید. تفاوت دو SNR مورد نیاز برای رسیدن به احتمال خطای بیت یکسان (در این جا $1.6 dB$) بهره کد نامیده می‌شود.

در حالت کلی، دو راه متمایز برای طراحی روش مقابله با خطا در یک سیستم مخابراتی دیجیتال وجود دارد :

۱- درخواست تکرار خودکار (ARQ):

در این حالت کد مورد استفاده تنها قادر به تشخیص خطا است. گیرنده با استفاده از آن، در صورت تشخیص خطا در اطلاعات دریافتی، از فرستنده درخواست ارسال مجدد اطلاعات را می‌نماید. این عمل معمولاً به صورت عکس انجام می‌شود، یعنی گیرنده در صورت دریافت صحیح اطلاعات با ارسال پیغامی (ACK) فرستنده را از آن با خبر می‌سازد. فرستنده در صورت عدم دریافت پیغام ACK ، اطلاعات مربوطه را مجدداً ارسال می‌کند.

۲- تصحیح خطا (FEC):

در این حالت کد مورد استفاده علاوه بر تشخیص خطا، توانایی تصحیح آن را نیز دارد. گیرنده اطلاعات دریافتی را به نزدیکترین کلمه کد صحیح د کد می‌کند. در این صورت دیگر گیرنده پیغامی را به فرستنده ارسال نخواهد کرد.

کدهای کانال به طور عمده به دو دسته کدهای بلوکی و کدهای کانولوشن تقسیم‌بندی می‌شوند.

الف- کدهای بلوکی

کدهای بلوکی بر پایه جبر مدرن^۱ و جبر میدان محدود^۲ ساخته می‌شوند. این کدها می‌توانند هم برای تشخیص و هم برای تصحیح خطا مورد استفاده قرار گیرند. تعدادی از کدهای بلوکی متداول عبارتند از کد همینگ، کد BCH ، کد

$Golay$ و کد $R-S^3$.

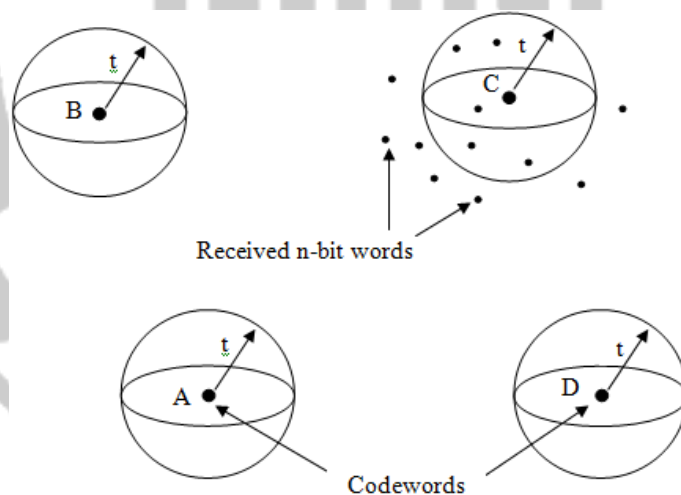
^۱ Modern algebra

^۲ Finite element algebra

^۳ Reed-Solomon codes

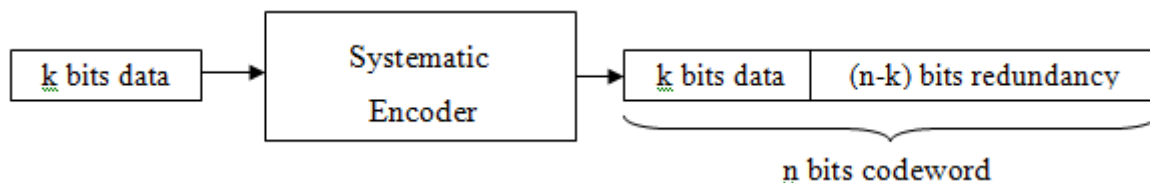
به دست آوردن درک کاملی از نحوه عملکرد کدهای بلوکی نیاز به دانش جبری بالایی دارد. اما به طور کلی می‌توان کار کرد یک کد بلوکی را این گونه بیان کرد: کدگذار بلوکی، یک بلوک k بیتی را به عنوان ورودی دریافت و در خروجی یک بلوک n بیتی به عنوان کلمه کد تحویل می‌دهد. در حالت باینری، بلوک k بیتی، 2^k حالت مختلف خواهد داشت، بنابراین 2^k کلمه کد متفاوت وجود دارد. اما این کلمات کد n بیتی پس از عبور از کانال دچار خطا می‌شوند و در گیرنده کلمات کد هر یک از 2^n بلوک ممکن می‌تواند باشد که تنها 2^k کلمه از آنها مجاز خواهد بود. وظیفه کدبردار کانال، پیدا کردن نزدیک کلمه کد مجاز به کلمه کد n بیتی دریافتی است.

توضیح چگونگی انجام این فرآیند نیاز به بیان روابط ریاضی نسبتاً پیچیده‌ای دارد که در اینجا از آوردن آن خودداری می‌نمایم. ولی برای به دست آوردن یک درک شهودی از مسأله، فرض کنید کدبردار از دایره‌های موجود در شکل (۲) برای انجام عملیات کدبرداری استفاده کند. در شکل (۲)، هر کلمه کد مجاز با یک نقطه نشان داده شده است، که این نقاط توسط دوایری با شعاع t احاطه شده‌اند. t برابر تعداد خطاهایی است که کد مورد نظر می‌تواند آن را تصحیح کند. واضح است که هر چه فاصله این نقاط از یکدیگر بیشتر باشد بهتر است، زیرا کد می‌تواند تعداد خطاهای بیشتری را تصحیح نماید (و یا تشخیص دهد).



شکل (۲): دایره‌های کدبرداری در گیرنده

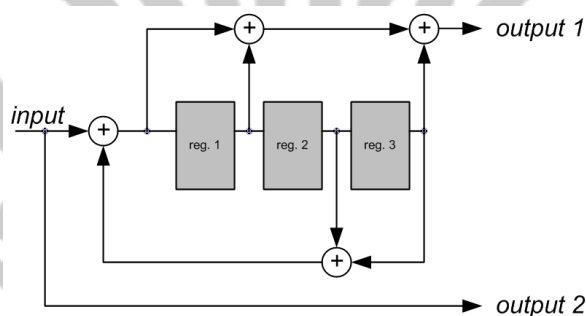
- کد بلوکی سیستماتیک: همان گونه که بیان شد، کدگذار بلوکی یک بلوک k بیتی در ورودی دریافت می‌کند و با افزودن $(n-k)$ بیت افزونگی، یک بلوک n بیتی در خروجی تحویل می‌دهد. در حالت کلی، بلوک n بیتی خروجی می‌تواند از لحاظ ظاهری شباهتی به بلوک k بیتی ورودی نداشته باشد. ولی در برخی کدها، بلوک k بیتی ورودی عیناً در بلوک n بیتی خروجی ظاهر شده و $(n-k)$ بیت افزونگی در ادامه آن می‌آید (شکل (۳)).



شکل (۳): کدگذار سیستماتیک

ب- کدهای کانولوشن

کدهای کانولوشن از پرکاربردترین انواع کدهای مقابله با خطا در سیستم های مخابراتی هستند. این کدها نیز دارای ساختار ریاضی محکم و پیچیده ای هستند و اغلب برای تصحیح خطا در سیستم های بلادرنگ به کار می روند. در این کدها، کل اطلاعات ورودی به صورت یک بلوک واحد در نظر گرفته می شود. با استفاده از یک کد کانولوشن، رشته بیت ورودی به یک شیفت رجیستر b بیتی داده می شود. به ازای هر شیفت از این شیفت رجیستر، k بیت جدید به آن وارد و n بیت کد شده از آن خارج می شود. بنابراین نرخ کد عبارت است از $\frac{k}{n}$. بیت های کد شده در یک کد کانولوشن نه تنها به k بیت ورودی در آن لحظه وابسته هستند، بلکه به ورودی های قبلی نیز بستگی دارند. شکل (۳) نمونه ای از یک کد کانولوشن ساده را نشان می دهد. در این کدگذار، k برابر ۱ و n برابر ۲ است، بنابراین، نرخ کد عبارتست از: $\frac{1}{2}$. یعنی با هر بار شیفت، این کدگذار یک بیت ورودی دریافت و دو بیت خروجی تحویل می دهد. دو بیت خروجی ترکیبی از بیت ورودی و بیت های موجود در خانه های حافظه (ورودی های قبلی) است.



شکل (۴): یک کدگذار کانولوشن ساده با نرخ $\frac{1}{2}$

قدرت یک کد کانولوشن در مقابله با خطا بستگی مستقیم به طول حافظه شیفت رجیستر (b) دارد. به عنوان مثال این طول در شکل (۳) برابر ۳ است. هرچه این طول بیشتر باشد، قدرت کد نیز بیشتر خواهد بود. اما افزایش طول حافظه کدگذار موجب افزایش پیچیدگی کدبردار نیز خواهد شد. برای کدبرداری کد کانولوشن راه های متعددی وجود دارد

که از آن میان الگوریتم ویتربی^۱ محبوبیت بیشتری دارد. این روش توسط آندرو ویتربی (Andrew Viterbi) در سال ۱۹۶۷ ارائه شد. مزیت عمده کدبردار ویتربی سادگی پیاده‌سازی آن بر روی مدارات مجتمع^۲ است. هر چند ساخت کدبردارهای ویتربی با طول حافظه بزرگ ($b > 9$) از نظر عملی غیرممکن (و یا حداقل بسیار دشوار) است. بطور کلی، کدهای کانولوشن بهره کدگذاری بالاتری نسبت به کدهای بلوکی دارند (با پیچیدگی مشابه).

۴-۲- طراحی و شبیه سازی

این آزمایش نیاز به طراحی و شبیه سازی پیش از آزمایش ندارد. ولی آشنایی مقدماتی با مفاهیم کدینگ، کانال دیجیتال و BER می‌تواند مفید باشد.

۴-۳- پرسش ها

(Q.1) کدگذاری بررسی توازن در چه حالاتی ممکن است دچار اشتباه شود؟
 (Q.2) دو کلمه کد از یک کد باینری را در نظر بگیرید. فاصله بین این دو کلمه کد عبارتست از تعداد بیت‌هایی که این دو کلمه با یکدیگر متفاوت هستند.^۳ به عنوان مثال فاصله بین 100110111 و 101110001 برابر سه است. زیرا این دو کلمه در سه بیت (بیت‌های سوم، هفتم و هشتم) با یکدیگر تفاوت دارند. قدرت تشخیص (تعداد خطاهای قابل تشخیص) و قدرت تصحیح (تعداد خطاهای قابل تصحیح) توسط یک کد، چه ارتباطی با حداقل فاصله بین کلمات آن کد دارد؟

message	codeword
000	000000
001	100100
010	010010
011	110110
100	001001
101	101101
110	011011
111	111111

(Q.3) کد زیر را در نظر بگیرید.

الف) کد مقابل چه نوع کدی است؟

ب) آیا این کد یک کد سیستماتیک است؟

ج) نرخ کد و قابلیت تشخیص و تصحیح خطای آن را به دست آورید.

¹ Viterbi

² VLSI

^۳ به این فاصله، فاصله همینگ (Hamming Distance) نیز گفته می‌شود.

۵- دستور کار:

قسمت اول: کدهای بلوکی

۵-۱- آزمایش ۱

کد تکرار: در این قسمت یکی از ساده ترین انواع کدهای بلوکی خطی را مورد آزمایش قرار می دهیم. فرض کنید در یک سیستم مخابراتی، هدف ارسال سمبل های ۰ و ۱ باشد. همچنین فرض کنید کانال مورد استفاده یک کانال باینری متقارن (bsc) باشد. بدین ترتیب هر بیت ارسال شده با احتمال مشخص p دچار خطا می شود. یک راه مناسب برای مقابله با این خطا (و کاهش آن) تکرار ارسال است. بدین معنی که فرستنده برای ارسال هر بیت (۰ یا ۱) آنرا n بار ارسال کند. بدین ترتیب، با فرض بدون حافظه بودن کانال در ارسال های متوالی، احتمال خطا به شکل قابل توجهی کاهش می یابد (چرا؟ $(R.1)$). به این عمل کد تکرار^۱ گفته می شود.

در محیط MATLAB فایل "repetition_code.m" را بنویسید که دو پارامتر تعداد تکرار (m) و نسبت سیگنال به نویز (SNR) را گرفته، به کمک تابع شبیه ساز مدولاتور/دمدلاتور BPSK (نوشته شده در فصل قبل)، ارسال داده به روش BPSK روی کانال AWGN را در حضور کد تکرار شبیه سازی نموده و نرخ خطای بیت را محاسبه کند. در صفحه command تعداد تکرار هر بیت را برابر ۵ انتخاب نمایید. آزمایش را در مقادیر مناسب SNR انجام داده و نتیجه حاصل را مشاهده و در قالب یک منحنی مناسب نرخ خطای بیت بر حسب SNR ذخیره و در گزارش خود ثبت کنید. $(R.2)$

بار دیگر m-file فوق را اجرا نموده و این بار تعداد تکرار هر بیت را برابر ۱ انتخاب کنید. به عبارت دیگر رشته بیت ورودی عیناً بر روی کانال ارسال می شود. در این صورت نتیجه بدست آمده برای حالت بدون کد گذاری خواهد بود. شکل مورد نظر را ذخیره و از آن مقایسه کارایی این کد استفاده نمایید. $(R.3)$

بهره کدینگ تکرار را در حالت های مختلف آزمایش شده بدست آورید. $(R.4)$

۵-۲- آزمایش ۲

کد همینگ: کد همینگ یکی دیگر از کدهای معروف بلوکی است. این کد از جمله کدهای سیستماتیک می باشد. کد همینگ توانایی تشخیص حداکثر ۲ بیت خطا و تصحیح ۱ بیت خطا را در هر کلمه کد دارد. به همین دلیل به آن کد SECDED^۲ می گویند.

با استفاده از ابزارهای MATLAB و توابه از پیش آماده آن یک واسط گرافیکی (GUI) با تنظیمات زیر ایجاد کنید:

- ۱- امکاناتی برای تنظیم نوع کدینگ مورد نیاز شامل کد همینگ یا کد RS یا ... ، که در صورت نیاز به استفاده از چند کد بصورت موازی و همزمان، می توانید چند کد را با هم انتخاب کنید. توجه کنید که

^۱ Repetition code

^۲ Single Error Correction-Double Error Detection

در کنار قسمت انتخاب برای هر کد، دکمه ای وجود دارد که تنظیمات مربوط به آن کد را می توانید در آن قسمت وارد کنید. (برای حالت بدون کد پارامتر خاصی وجود ندارد. واسطه گرافیکی برای کد تکرار پارامتر تعداد تکرار (k) و برای کد همینگ دو پارامتر n و k و برای کد کانولوشن دو پارامتر n و k و دو ماتریس طول های مقید و ضرائب را به شکل مناسب دریافت می کند.)

۲- دو فیلد برای گرفتن مقدار پایین و بالای SNR مورد نیاز برای ارسال روی کانال AWGN، که می توانید بر حسب dB مقادیر را وارد کنید.

۳- دکمه Run که باید بعد از انجام همه تنظیمات فشار داده شود تا شبیه سازی اجرا شود. بعد از اتمام شبیه سازی که با توجه به تعداد کدهای انتخابی و محدوده ی SNR داده شده، مدتی (شاید طولانی) طول خواهد کشید و منحنی احتمال خطا بر حسب SNR رسم خواهد شد.

فایل GUI تهیه شده را به ضمیمه گزارش خود تحویل دهید. (R.5)

در قسمت تنظیم پارامترهای کد همینگ، در ابتدا یک کد همینگ (7,4) را انتخاب کنید. با ترسیم منحنی BER کارایی آن را در مقابله با خطا در مقایسه با حالت بدون کدینگ مقایسه کرده و نتایج را ثبت نمایید. (R.6)

با تغییر پارامترهای کد همینگ، بند قبلی را به ازای سه مقدار مختلف از این پارامترها تکرار نموده نتایج را در گزارش خود منعکس نمایید. برداشت خود را از نتایج بدست آمده بیان کنید. (R.7) بهره کد را بدست آورید. (R.8)

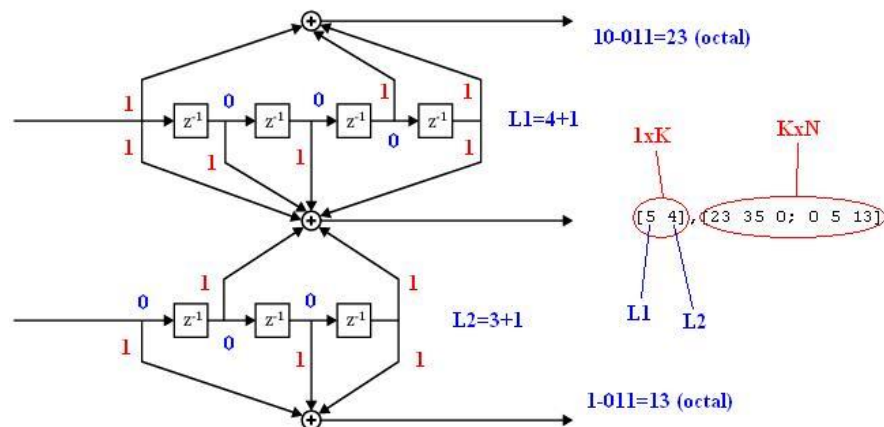
توجه: در انتخاب مقادیر پارامترهای کد همینگ دقت شود به دلیل ساختار ریاضی کد شرایط زیر باید لحاظ شود:

$$\begin{cases} n = 2^m - 1 \\ k = 2^m - 1 - m \end{cases}$$

قسمت دوم: کدهای کانولوشن

۳-۵- آزمایش

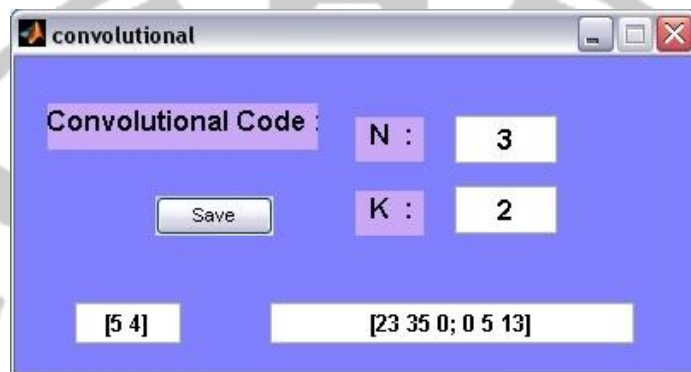
کد کانولوشن: کدهای کانولوشن، کدهایی هستند که کلمات کد خود در خروجی را با استفاده از جمع در مبنای ۲ شیفته یافته های زمانی داده ورودی بدست می آورند. به شکل (۵) که برای نمونه از نرم افزار MATLAB برداشته شده نگاه کنید.



شکل (۵): یک نمونه کد کانولوشن با k=2 و n=3

شکل ۵ یک نمونه کد کانولوشن را نشان می دهد. همان طور که ملاحظه می کنید دو ردیف حافظه وجود دارد، یعنی برای هر ورودی یک خط حافظه، به همین دلیل $k=2$ می باشد. پس k در کد های کانولوشن نشان دهنده تعداد ورودی است. هر خط ورودی می تواند چند حافظه داشته باشد، مثلاً در مثال بالا خط یک ۴ حافظه و خط دو ۳ حافظه دارد. اما برای استفاده از دستورات MATLAB برای کد کانولوشن باید از طول مقید کد استفاده کنیم. طول مقید کد^۱ به یکی بیشتر از تعداد حافظه ها گفته می شود. در مثال بالا L_1 و L_2 طول های مقید خط اول و دوم کد به ترتیب برابر ۵ و ۴ می باشند. برای استفاده از محیط گرافیکی ایجاد شده برای کدینگ باید پارامتر های n و k و مشخصات شیر^۲ های گرفته شده در کد به برنامه داده شود. (به شکل ۶ نگاه کنید)

فرمت استفاده شده در GUI برای دریافت اطلاعات شیر هابه صورت نشان داده شده در شکل ۵ و ۶ باشد.



شکل (۶): محیط گرافیکی برای دریافت اطلاعات کد کانولوشن

با توجه به خط اول خروجی و خط اول حافظه مشاهده می کنید که اگر در مکان هایی که شیر قرار دارند یک و جاهایی که شیر قرار ندارد صفر قرار بدهیم، برای خط اول خروجی از ردیف حافظه اول عدد ۲۳ در مبنای ۸ بدست می آید. به همین ترتیب برای بقیه خطوط خروجی و خطوط حافظه عمل می کنیم. پس واضح است که ابعاد ماتریس های ورودی بصورت $1 \times K$ و $K \times N$ می باشد. مانند مثال بالا که در آن ابعاد به صورت 1×2 و 2×3 است.

اکنون به بررسی کارایی این کدها در مقابله با خطا در کانال می پردازیم:

- در محیط گرافیکی آزمایش قبل (GUI)، کد کانولوشن را انتخاب و پارامترهای آن را همان پارامترهای پیش فرض قرار دهید.
 - با ترسیم منحنی BER کارایی این کد در مقابله با خطا و در مقایسه با حالت بدون کدینگ مشاهده کرده و نتایج را ثبت کنید. (R.13)
 - با تغییر طول مقید یکی از شاخه های ورودی، یک کد جدید (دلخواه) طراحی و بند فوق را تکرار و نتایج را با یکدیگر مقایسه کنید. تأثیر طول مقید در قدرت کد چیست؟ برداشت شما از این نتایج چیست؟ (R.14)
 - یک کد کانولوشن طراحی کنید که ریت آن برابر یکی از کدهای همینک مورد استفاده در آزمایش ۲ باشد. با ترسیم منحنی BER، قدرت این دو کد را با یکدیگر مقایسه نمایید. (R.15)
- بهره کد چقدر است؟ (R.16)

¹ Constraint Length

² Tap