باسمه تعالی

پروژه سیگنال و سیستم
مخابرات طیف گسترده باینری

نام و نام خانوادگی: آرش دهقانی۹۹۱۰۹۱۳۳ آرمان لطفعلی خانی ۹۹۱۰۹۱۶۶

در تمام کد پروژه اندیس ۱ در نام سیگنال ها مربوط به مخابرات عادی و اندیس ۲ در نام سیگنال ها مربوط به مخابرات طیف گسترده است.

تفاوت مخابرات طیف گسترده با مخابرات عادی در این است که در مخابرات طیف گسترده ما داده ها را گسترش میدهیم و بعد از مدولاتور در یک دنباله PN رندوم ضرب میکنیم.

ابتدا داده ها را با دستور $randi$ تولید میکنیم. دنباله رندوم را هم با همین دستور ایجاد میکنیم تا دنباله های -1 و ۱ ایجاد شوند. با توجه به اینکه تفاوت مدولاسیون عادی با مدولاسیون bpsk تنها تغییر فاز کسینوس ضرب شده در بیت‌های صفر به مقدار π است، کافیست بیت‌ها را به ۱و ۱- تبدیل کرده سپس مدولاسیون عادی را روی آنها انجام دهیم.

قبل از رسیدن به بلوک $modulator$ ما داده ها را برای طیف گسترش میدهیم. یعنی در بردار داده ها از هر داده، M نمونه متوالی وجود دارد.(در مخابرات عادی در بردار داده ها، هر خانه مربوط به یک داده جدید است.) گسترش را به کمک ضرب $outproduct$ در متلب انجام میدهیم.

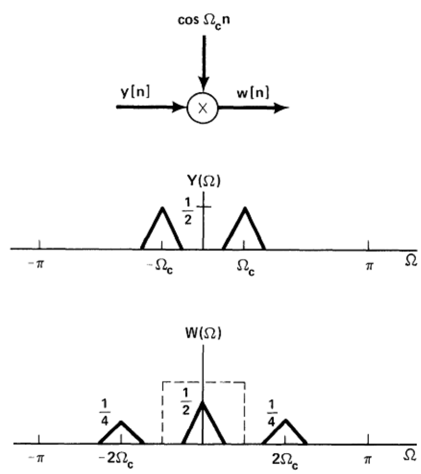
از این به بعد بردار دامنه توابع مربوط به دو روش دارای بازه ی متفاوتی هستند. n_1(مخابرات عادی) از ۰ تا N(تعداد داده ها) و (مخابرات طیف گسترده)n_2 از ۰ تا NM است.

۱ مدولاتور

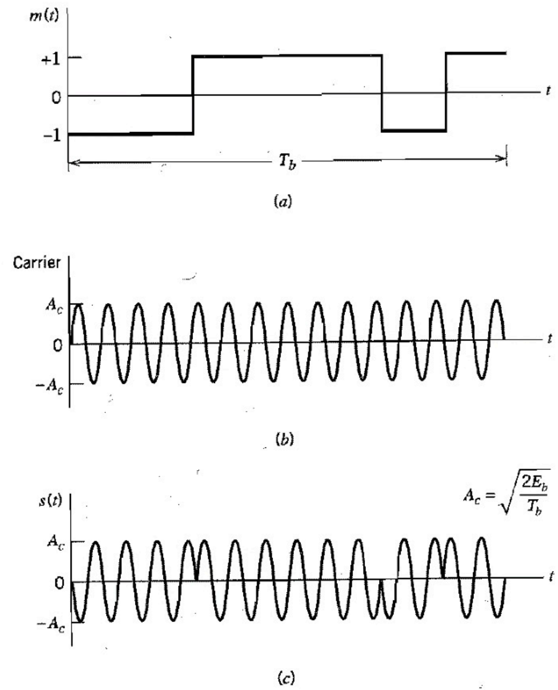
در این بخش داده های هر دو بخش را در $cos(2\pi f_m n)$ ضرب میکنیم. که f_m فرکانس مدولاتور و n دامنه مربوطه است. این کار باعث شیفت در حوزه فرکانس به اندازه fm و $-fm$ میشود. اگر $b[n]$ را بیت های داده بنامیم، تعریف میکنیم:

$$x_2[n] = b_2[n]cos(2\pi f_m n)$$

خروجی این بلوک در کد modulated نام دارد.



شکل ۱تغییرات نمودار تبدیل فوریه سیگنال بعد از مدولاسیون و دمدولاسیون



(*a*) **Product signal $m(t) = c(t)b(t)$.** (*b*) Sinusoidal carrier. (*c*) DS/BPSK signal.

شکل ۲مدولاسیون BPSK در حوزه زمان

با توجه به نمودار، برای همپوشانی نداشتن بخش‌های مختلف تبدیل فوریه سیگنال بعد از دمدولاسیون، دو شرط زیر باید برقرار باشند (پهنای باند سیگنال اصلی برابر ω_b فرض شده است):

$$\begin{aligned} ۲\Omega_c - \omega_b \geq \omega_b \Rightarrow \Omega_c \geq \omega_b \Rightarrow f_c \geq f_b \\ ۲\Omega_c + \omega_b \leq ۲\pi - \omega_b \Rightarrow f_c + f_b \leq \frac{۱}{۲} \end{aligned} \quad (۱)$$

هر دو شرط نیز در طراحی فیلتر پایین گذر و تعیین فرکانس مدولاسیون در کد لحاظ شده‌اند.

۲ ضرب PN

این بخش فقط برای طیف گسترده است و دنباله PN که در اینجا $p[n]$ مینامیم در خروجی مدولاتور ضرب میشود. در واقعیت این رشته اعداد توسط تعدادی شیفت رجیستر با مشخصات خاص ساخته می‌شوند تا به تقریب اعداد تولید شده، رندوم باشند ولی ما اینجا مستقیما از دستورات تولید عدد رندوم استفاده کردیم تا نیازی به این تقریب نباشد.

خروجی این بلوک در کد $modulatedp۲$ نام دارد.

۳ کانال

حال سیگنال سینوسی($i[n]$) و نویز گاوسی ($g[n]$) به سیگنال عادی و طیف گسترده اضافه میشود. برای نویز گاوسی از تابع wgn استفاده میکنیم. برای نویز سینوسی تابع نوشتیم و در یک $Mfile$ کنار فایل های اصلی قرار دادیم. دامنه‌ی سیگنال سینوسی تداخلی بزرگ است. ($A > M$) در فایل تابع میتوان ضریب $\frac{A}{m}$ را تغییر داد. حالت ایده آل برای jammer این است که فرکانس بیت تداخلی برابر فرکانس مدولاتور باشد تا بیشینه تداخل را داشته باشد. در کد به طور پیش فرض این دو فرکانس مساوی اند.

خروجی این بلوک در کد noisedsignal نام دارد.

۴ ضرب دوباره PN

در این قسمت دوباره همان دنباله ای را که در بخش ۲ ضرب کردیم، ضرب میکنیم. اگر خروجی این بخش را $y[n]$ بنامیم، داریم:

$$y[n] = p[n]^۲ x[n] + p[n]i[n] + p[n]g[n]$$

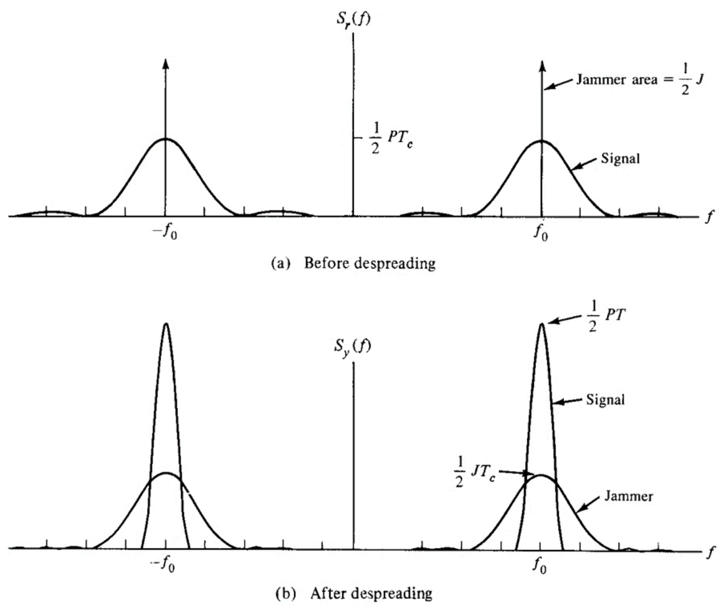
که چون $p[n]$ ۱ یا ۱− است، $p[n]^۲ = ۱$

$$y[n] = x[n] + p[n]i[n] + p[n]g[n]$$

اثر pn روی داده ها از بین میرود اما ضرب pn روی نویز و بیت تداخلی باعث گسترش طیف آن میشود. در شکل زیر میبینیم که بیت تداخلی سینوسی(که δ در حوزه فوریه است) به واسطه ضرب pn در حوزه فوریه پخش شده است. که بعدا با گذاشتن فیتر مناسب بخش کمتری از طیف آن عبور میکند. همچنین توجه داریم که ضرب $p[n]$ بر نویز گاوسی اثری ندارد. با توجه به اینکه نویز $g[n]$ توسط یک فرایند تصادفی تولید شده است و با فرض تصادفی بودن و همچنین iid بودن مقادیر $p[n]$ در یک دوره داریم:

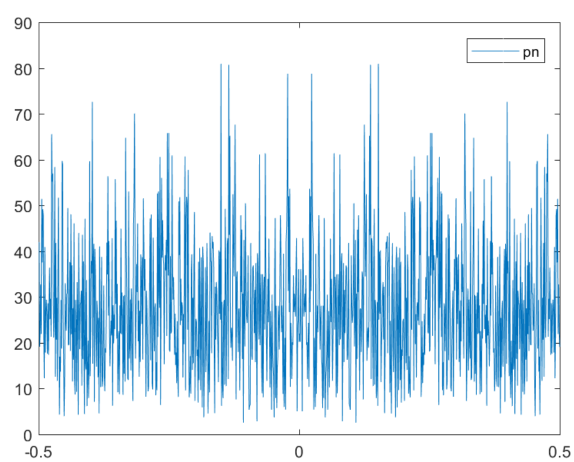
$$f_G(g) = f_{G|p=۱}(g)\mathbb{P}(p=۱) + f_{G|p=-۱}(-g)\mathbb{P}(p=۱) = \frac{۱}{۲}(f_G(g) + f_G(-g)) = f_G(g) \tag{۲}$$

توجه داریم که در رابطه بالا فقط از فرض زوج بودن توزیع $f_G(g)$ استفاده دش بنابراین رابطه برای توزیع‌های دیگری بجز توزیع گاوسی نیز می‌تواند صادق باشد. از این رابطه نتیجه می‌شود که خواص فرکانسی $p[n]g[n]$ با $g[n]$ یکسان است و فرقی بین مخابرات طیف گسترده و عادی از این نظر وجود ندارد.



شکل ۳سیگنال اصلی و تداخلی : بالایی قبل ضرب دوباره و پایینی بعد ضرب دوباره

البته شکل بالا در تحلیل پیوسته زمان است ولی در گسسته زمان نیز همین گسترده شدن وجود دارد. اما یک فرق بین تحلیل پیوسته زمان و گسسته زمان وجود دارد و آن این است که PN در تحلیل پیوسته زمان محدود باند است و در تحلیل گسسته زمان پهن باند.



شکل PN۴ در حوزه فرکانس

خروجی این بلوک در کد pnremoved۲ نام دارد.

۵ دمودولاسیون

در این مرحله دوباره سیگنال را در $cos(۲\pi f_m n)$ ضرب میکنیم یعنی در کسینوس با همان فرکانس مدولاتور. با این کار برعکس شیفت مدولاسیون را اعمال مبکنیم و بعد آن با یک فیلتر پایین گذر تنها طیف های مورد نظر را عبور میدهیم. برای فیلتر پایین گذر از کانولوشن با تابع $sinc(۲\pi f_{sinc} n)$ بهره میبریم. کار دیگری که میتوان کرد. با توجه به اینکه فرکانس بیت تداخلی هم نزدیک فرکانس مدولاتور است، طیف هایی از بیت تداخلی از فیلتر عبور میکند.

خروجی دمودولاسیون در کد demodulated نام دارد. خروجی پایین گذر filtered نام دارد.

۶ detector

در این بخش M نمونه را جمع میکنیم. بعد از دمدولاتور و فیلتر ما حاصل ضرب بیت تداخلی سینوسی در کسینوس دمودولاتور را داریم. با استفاده از اتحاد مثلثاتی، این حاصل ضرب در واقع دو تا سینوس با فرکانس های متفاوت ایجاد میکند. با جمع کردن M نمونه، در واقع حاصل جمع نمونه های بیت های داده اصلی و چند شکل سینوسی را حساب میکنیم. ابن جمع در بدترین حالت، روی چند دوره تناوب تابع سینوسی و بخشی از دوره‌ی آن است. پس در حالی که حاصل جمع نمونه ها در حال افزایش است، حاصل جمع بیت سینوسی تداخلی تغییر چندانی نمیکند. اگر s_k را جمع متناظر با بیت k ام بدانیم، داریم:(بخش زیادی از نویز گوسی در فیلتر پایین گدر ار بین میرود و با توجه به کوچکی اثرش نسبت به بیت تداخلی، نگرانی ما در این بخش نیست)

$$s_k = \sum_{(k-۱)M}^{kM} (b_۲[n] + A sin(\Omega n) p[n])$$

$$s_k = \pm M + \alpha A$$

که α ضریبی کوپک تر از ۱ است.

هنوز اثر بیت تداخلی تا حدی وجود دارد.(با توجه به $A > M$) هر چند در طیف گسترده، بخشی از طیف های آن گسترده و حذف شده است. ما سه راه برای شناسایی بیت های خروجی داریم:

یکی اینکه اگر s_k مثبت شد، بیت را ۱ و اگر منفی شد -۱ اعلام کنیم. اما با توجه به روابط بالا روش بهتری نیز وجود دارد. فرض کنید بیت اطلاعاتی مثبت ۱ باشد: شاید اثر بیت تداخلی باعث منفی شدن بیت مثبت شود اما یک عدد منفی با اندازه نسبتا کوچک ایجاد کند. یا برای بیت اطلاعاتی -۱ اتفاقی مشابه رخ دهد. به عبارت دیگر، ممکناست نویز خارجی یک عدد ثابت منفی بزرگی ایجاد کند که سیگنال ورودی detector را به مقدار زیادی به بالا یا پایین شیفت دهد. برای رفع این مشکل نیز دو راه داریم: یکی اینکه میانگین همه داده‌ها را از آن کم کنیم، که موجب از دست رفتن مولفه dc سیگنال اولیه می‌شود و می‌تواند خطا ایجاد کند. راه دوم این است که $thereshold$ را میانگین ماکزیمم و مینیموم s_K ها بگذاریم تا از خطای گفته شده تا حد امکان جلوگیری کنیم. در این حالت، مولفه $f = ۰$ سیگنال اولیه نیز دچار تغییر نشده تمام اثرات نامطلوب حداکثر به فرم ضرب عدد ثابت در تبدیل فوریه سیگنال اولیه ظاهر می‌شوند.

روش اول نیز در ابتدا آزمایش شده و از لحاظ عملی نیز خطای بسیار بالای آن (بین ۴۰ تا ۵۰ درصد) مشاهده شده است، بنابراین استفاده از روش سوم تنها راه قابل قبول است.

$$thereshold = \frac{\min(s_k) + max(s_k)}{۲}$$

به عنوان مثال اگر بردار داده ها −۱۳۰،−۱۲۰،−۵ و ۲۰ بود، منطقی تر آن است که −۵ را ۱ بگیریم. در حالی که در روش اول(با $thereshold$ صفر) آن را −۱ تشخیص خواهیم داد.
خروجی در بردار data موجود است.

۷ نتایج

۴ فایل وجود دارد که خطای مخابرات عادی و طیف گسترده را به ازای M های ۵۰،۱۰۰،۵۰۰، ۱۰۰۰ بیان میکند. ۴ عکس اول برای $\frac{A}{M} = ۱/۲$ است(نسبت ثابت A به M). با تغییر در تابع sinenoise میتوان A را تغییر داد. به طور پیش فرض $۱/۲M$ است. (برای نشان دادن عملکرد خوب طیف گسترده در نویز های منطقی)

```
error_normal =

    0.4250


error_spread_spectrum =

    0.0100
```

$$M = ۵۰$$

```
error_normal =

    0.4200


error_spread_spectrum =

    0.0050
```

$$M = ۱۰۰$$

```
error_normal =

    0.4350


error_spread_spectrum =

     0
```

$M = ۵۰۰$

```
error_normal =

    0.4750


error_spread_spectrum =

     0
```

$M = ۱۰۰۰$

۴ عکس بعدی برای $A = ۱۱۰۰$ است و دامنه در آن به ازای ۴ M مختلف ثابت است. پیش فرض برای A در تابع $۱٫۲M$ است. برای دیدن نتایج زیر کافی است مقدار A را در تابع sinnoise مساوی ۱۱۰۰ بگذارید.

```
error_normal =

    0.4500


error_spread_spectrum =

    0.4650
```

$M = ۵۰$

```
error_normal =

    0.4700


error_spread_spectrum =

    0.1600
```

$$M = ۱۰۰$$

```
error_normal =

    0.4700


error_spread_spectrum =

     0
```

$$M = ۵۰۰$$

```
error_normal =

    0.5300


error_spread_spectrum =

     0
```

$$M = ۱۰۰۰$$

همانطور که مشاهده میکنیم، در A ثابت با افزایش M خطای ما بسیار کاهش میابد.

۸ مزایا و پاسخ به سوالات

همان طور که در توضیحات نیز ذکر شد، مزیت طیف گشترده یکی در پخش کردن سیگنال تداخلی در حوزه فرکانس است که باعث میشود بخشی از طیف های آن از فیلتر عبور کند. دیگر مزیت آن در detector و

هنگام جمع زدن است که اثر بیت تداخلی را در مقابل بیت داده کمتر مبکند. این را در بند اول detector توضیح داده ایم. jammer سعی میکتد بیت تداخلی با دامنه ای بزرگتر از اندازه داده ها ارسال کند اما در بخش جمع کنندهی detector با جمع کردن نمونه ها، توان و اثر داده ها زیاد خواهد شد و به مرتبه M خواهد رسید. همان طور که مشاهده شد،با بیشتر شدن M نتیجه بهتری خواهیم گرفت چون نمونه ها بیشتر هستند.

مزیت دیگر آن در دسترسی چندگانه از کانال است که تنها فرستنده و گیرنده مورد نظر از PN اطلاع دارند. و ضرب دو PN متفاوت در هم یک سینگال نویز مانند تولید میکند پس بیت های فرستنده های دیگر مانند نویز میشوند و تنها سیگنال مورد نظر (بعد از دمودوله و ...) به عنوان داده به گیرنده میرسد.

با افزایش N این مزیت ها بیشتر میشود.

میتوانید با کپی کردن کدها از فایل fourier.txt و پیست کردن آن در پایان هر یک از Mfile ها تبدیل فوریه مراحل را به ترتیب مشاهده فرمایید.