通信理論レポート課題

担当: 前原　文明　教授

1w142016-0
情報通信学科　飯島　涼
提出日 2016/01/08（金）

0. コサインロールオフフィルタの役割

波形をデジタル変調で送る場合、シンボル周期 T_0 ごとに、インパルス信号を用いてサンプリングする。インパルス信号はフーリエ変換すると、無限の周波数帯域を占めることがわかり、送信するには無限の周波数帯域が必要である。しかし、実際に信号を送信する際には、送信者ごとに使って良い周波数帯域は限られている。そこで、2 つのナイキスト基準(伝送帯域を制限/符号間干渉が起こらないよう、シンボル周期 T_0 ごとに零クロス)を満たすコサインロールオフフィルタを用いることで、符号間干渉を起こさないよう、伝送帯域を制限することができる。

また、伝送速度の速さが早くなると、伝送帯域は広くなるため、伝送速度を保ちつつ、伝送帯域を制限するのにも使える。

1. コサインロールオフフィルタのインパルス応答

逆フーリエ変換によってコサインロールオフフィルタを求める方法については、式が多くなり、word で表記するのに時間がかかるため、手書きし、この章の最後に添付した。

コサインロールオフフィルタのインパルス応答を、ロールオフ係数をパラメータにとって表示した結果を図 1 に示す。(c 言語を用いて gnuplot にパイプラインをつなぎ、XQuartz で表示)式をわかりやすくするため、T_0=1 と設定し、x の範囲は[-5,5]とした。

紫が $\alpha = 1$, 緑が α =0.75,青が α =0.5,オレンジが α =0.25,黄色が α =0 とした時のインパルス応答である。α =0 のとき、理想ローパスフィルタである。図をみればわかる通り、α が 0 に近いほど、振動が大きく、収束が遅い。よって、理想ローパスフィルタに近いフィルタではインパルス応答を記録するためのメモリを多く用意しなければいけなくなるため、α の値を適宜設定し、振動を少なくしたコサインロールフィルタが用いられる。

注

このレポート内の図は、(インパルス応答の式に単純に値を挿入しただけであるため、)特異点が欠けてしまっているが、この部分の数値についてはインパルス応答の導出で特異点について導出しているため(T_0=1 だから図 1 では $1/T_0$=1、図 2 以降ではそれに振幅を乗じた値が特異点である)、その値がグラフ内にあるとして補って考えることにする。

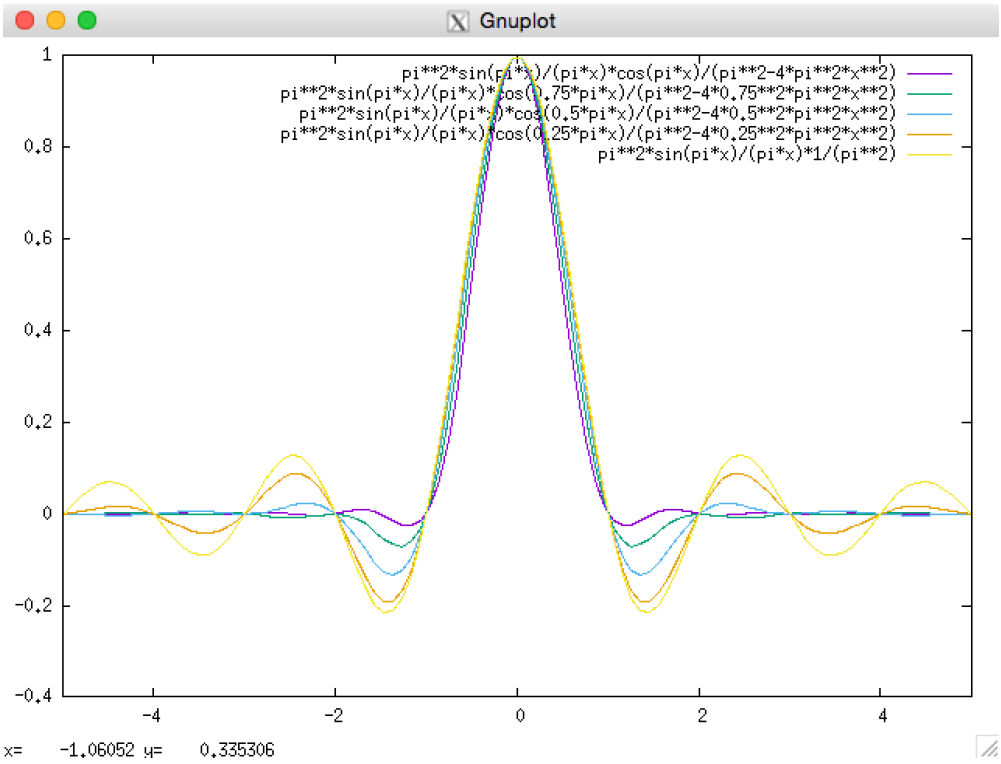

図 1 コサインロールオフフィルタ　インパルス応答の波形（tushin.c）

2. 波形整形後の信号空間ダイアグラムとアイパターン

まず、アイパターンのグラフを生成する

アイパターンは、式をわかりやすくするため、T_0=1、A=1 とし、T_0ごとに現れるインパルス応答を足し合わせた波形を、すべてのパターンを重ねて表示したものである。今回、5 つのインパルスが通ったインパルス応答の合計を求める。QPSK 信号は、I 軸上で考えると、2 桁で表される QPSK 信号の 1 桁目が[0]のとき A/√2,[1]のとき-A/√2 となる。Q 軸では、2 桁のうち 2 桁目で[0]のとき A/√2,[1]のとき-A/√2 となる。よって、桁の場所の違いだけで、[0]のとき A/√2,[1]のとき-A/√2 という条件は変わらないため、I 軸、Q 軸共に同じアイパターンとなる。

同じになるため、ここからは一つの軸 Q 軸に限定して考える(I 軸に対しても同様である)。2 桁目の数値が、00000、00001、00010、……、11111 を取るとき、それぞれのインパルス応答を重ねて表記表記する。C 言語では、2 進数を表すのが難しく、よくわからなかったので、5 つの数がそれぞれ 0 を取るか、1 を取るかは 5 個それぞれのインパルスに配列を割り当て、それぞれに rand()%2 の値を格納することによって決め、波形を多めに 80 個出すことによってすべての種類の波形を出そうと試みた(すべての波形は 32 通りだが、0 か 1 かをランダムに決めているため、32 個の波形を重ね合わせただけでは出ない波形があり、80 個ぐらいにしないと左右対称にならなかった)。

図 2.1 図 2.2 図 2.3 に α=0, α=0.5, α=1 のときのアイパターンを示す。

(図が入りきらないため、次ページから)

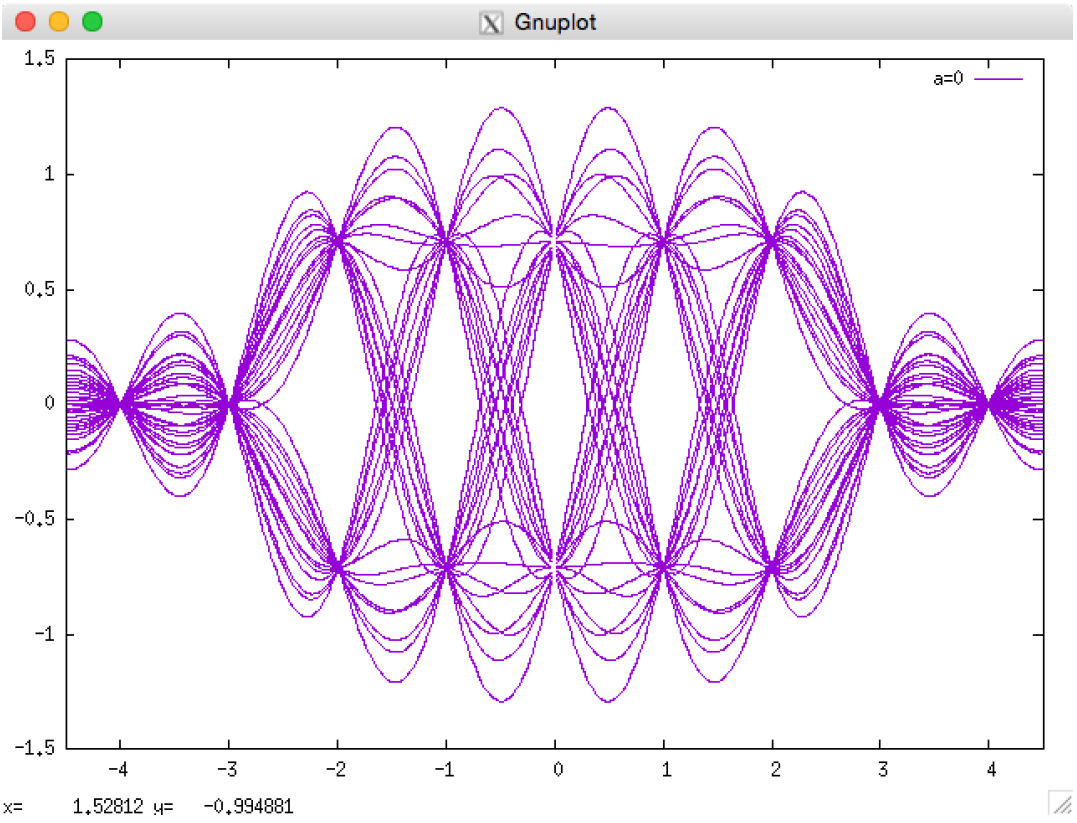

図 2.1 アイパターン $\alpha=0$ (tushin5.c)

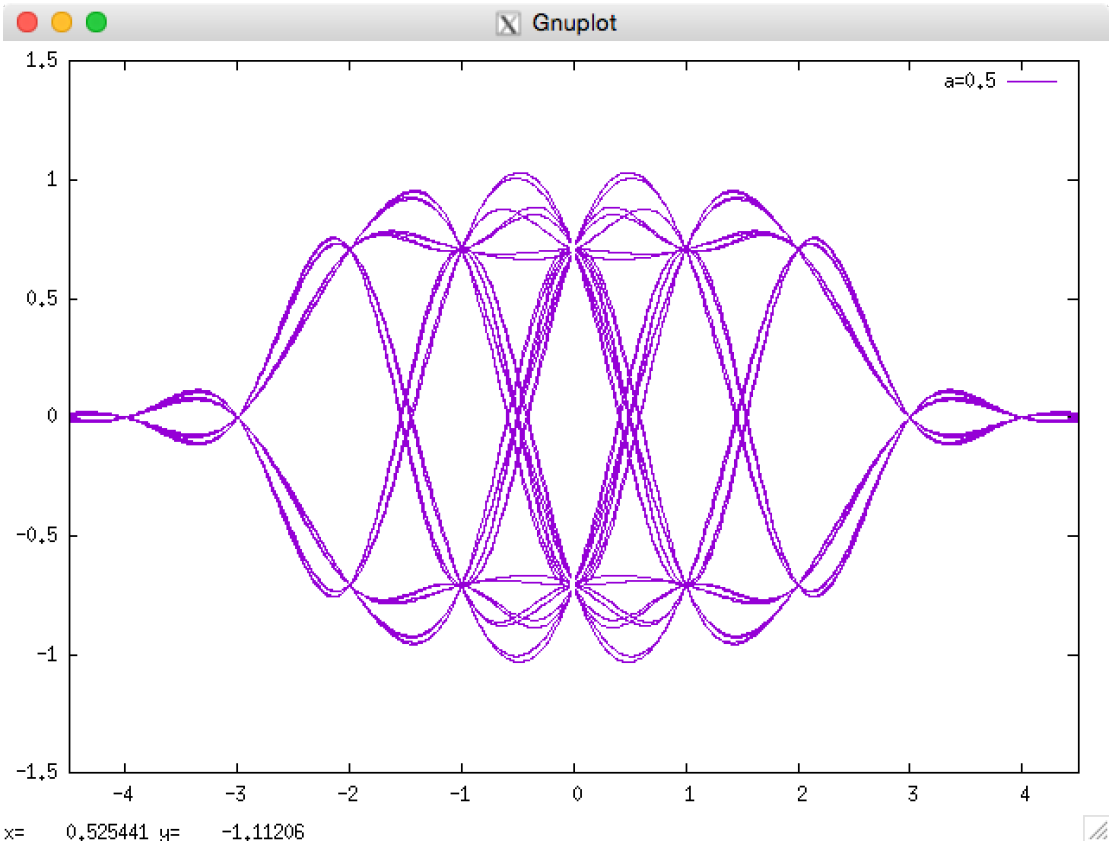

図 2.2 アイパターン　α =0.5 (tushin4.c)

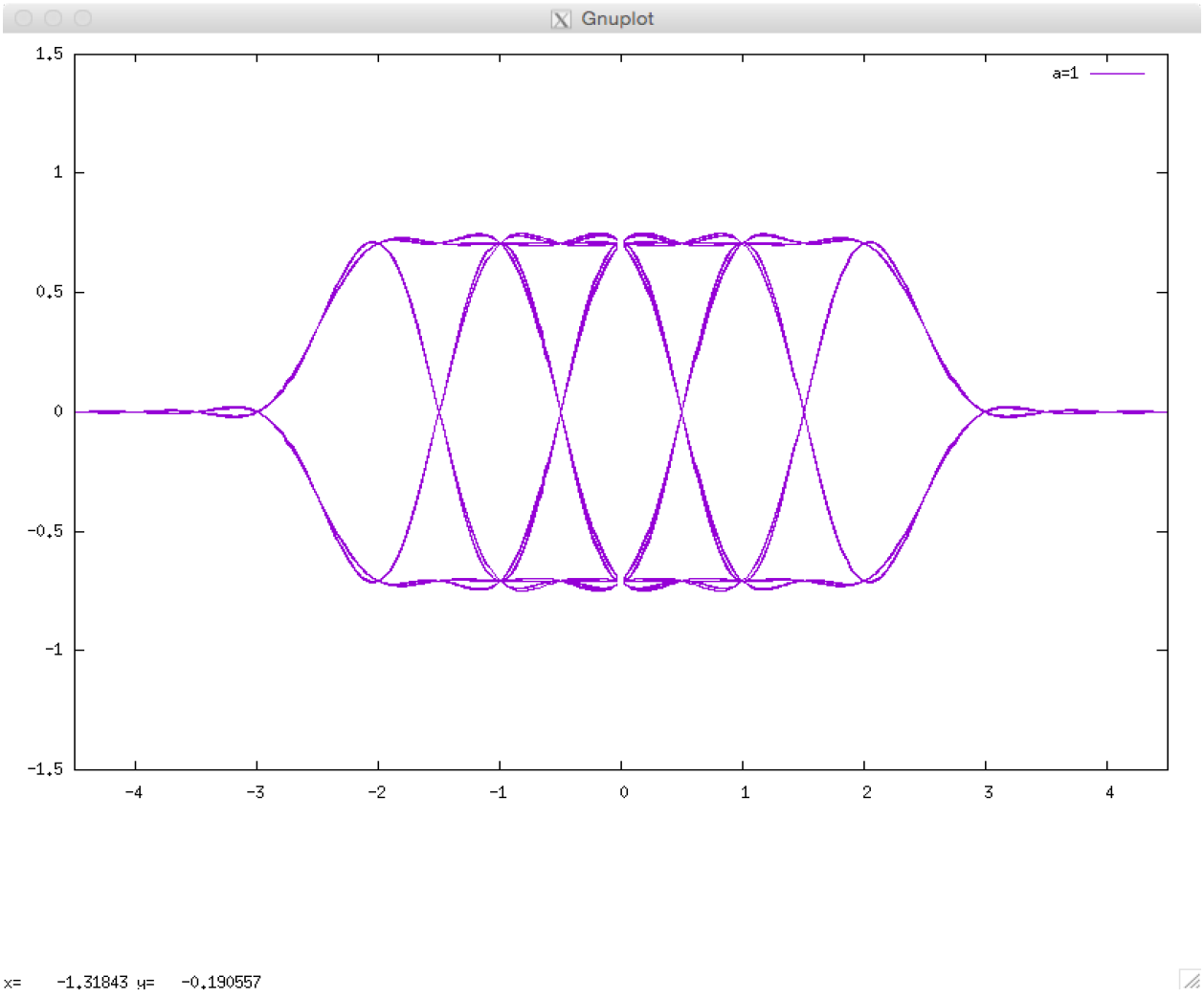

図 2.3 アイパターン　α =1　(tushin3.c)

混雑度は α が大きくなるごとに増しているが、識別タイミングでのアイの開きはどの波形も同じであり、これは、コサインロールオフフィルタが、ナイキスト基準の T_0(ここでは T_0=1)で零クロスという条件を満たしているからである。nT_0 ごとの値は+A/√2, -A/√2 のどちらかに収束し、情報が保てていることを確認できる。

次に信号空間ダイアグラムを示す。上で作った波形の y 成分を縦軸方向と横軸方向に y,y2 として配置する。rand()関数は同一の乱数列を生み出してしまうため、randA,randA2 クラスを作り、randA2()クラスではあらかじめ 2 回 rand();を実行しておくことで、Q 軸、I 軸の波形が常に一致してしまう現象を避けた。32*32 通りで約 900 回の波形を重ねれば信号空間ダイアグラムは目視できる範囲で十分確認できるだろうと考え、波形は 900 回分を重ねあわせている。

図 2.4 図 2.5 図 2.6 に α =0, α =0.5, α =1 のときの信号空間ダイアグラムを示す。

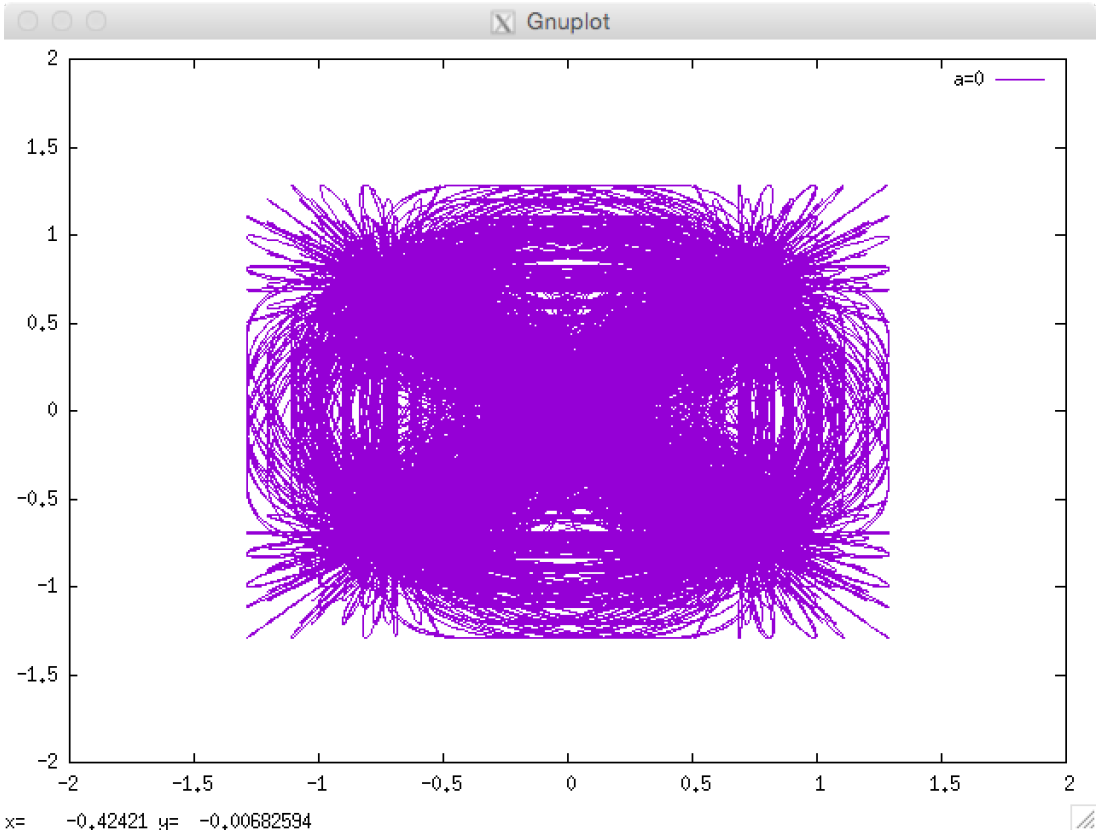

図 2.4 信号空間ダイアグラム $\alpha = 0$ (tushin8.c)

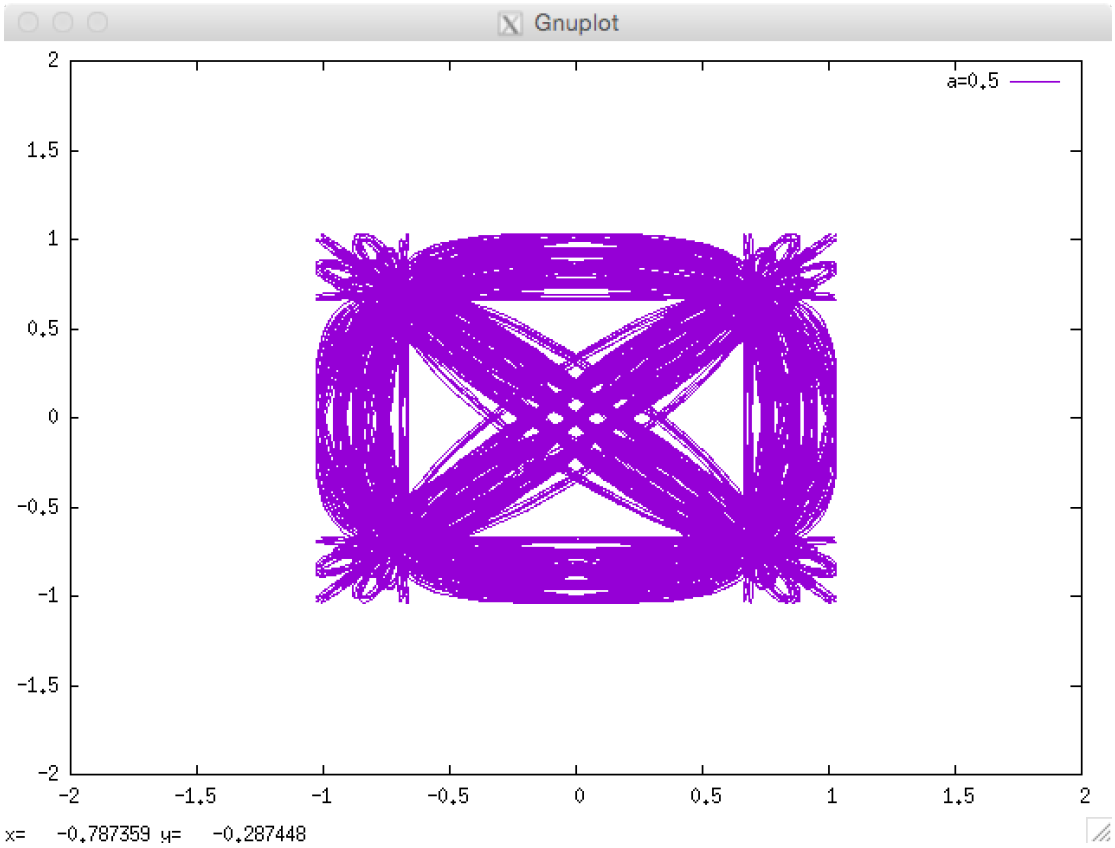

図 2.5 信号空間ダイアグラム $\alpha = 0.5$ (tushin7.c)

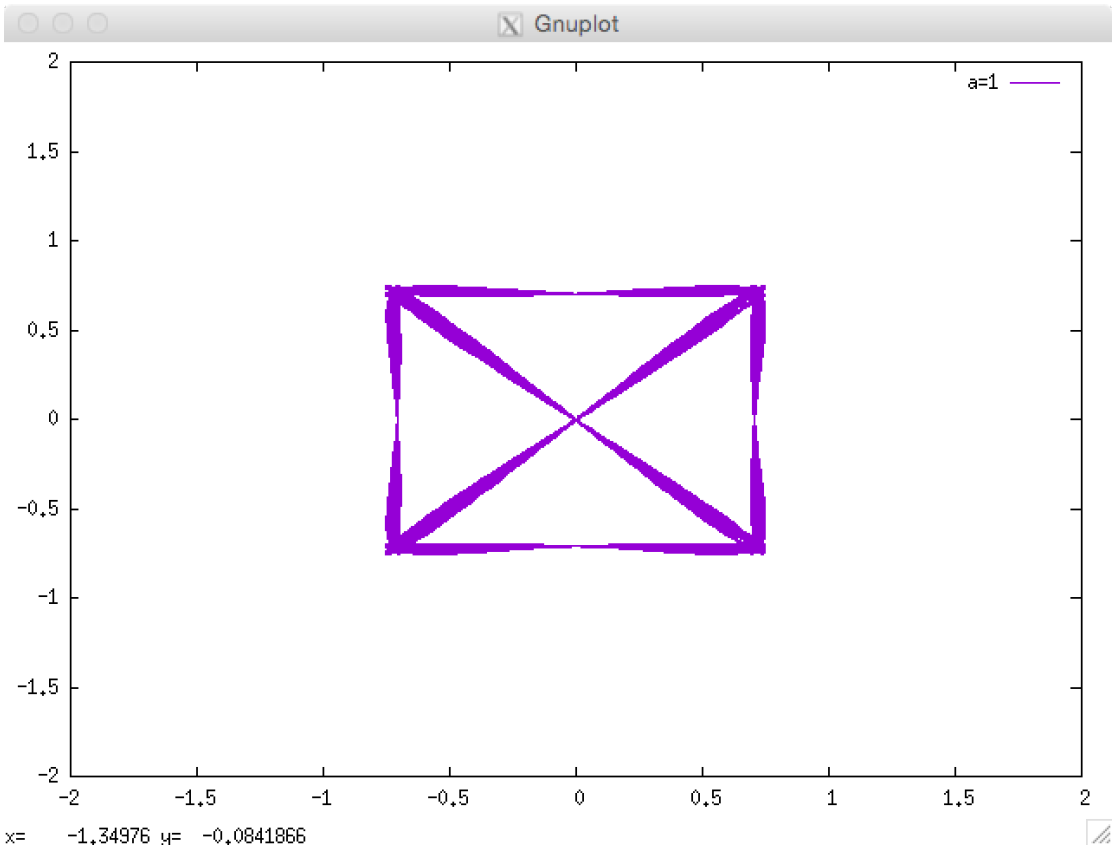

図 2.6 信号空間ダイアグラム $\alpha = 1$ (tushin6.c)

信号空間ダイアグラムは α が 1 に近づくほど、重なる部分が多く、信号空間ダイアグラムの観点から見ても、余分な振動の少ない綺麗なインパルス応答となることがわかる。

3. 16QAM の信号空間ダイアグラムとアイパターン

アイパターンを QPSK と同様に示す。図 3.1 図 3.2 図 3.3 に α=0, α=0.5, α=1 のときのアイパターンを示す。比較のため、T_0=1, A=1 の条件は同じである。

1 シンボル 4 桁のうち、下 2 桁を基準にすると、振幅は 00 のとき 3A/√10,01 のとき A/√10,11 のとき-A/√10,10 のとき-3A/√10 の 4 パターンある。上 2 桁がどの数値を取る場合でもこの法則は一緒であるため、下 2 桁の数にのみに依存する Q 軸方向について考えた。上 2 桁も上記と同様の組み合わせであるから、I 軸もアイパターンは同様の結果となる。

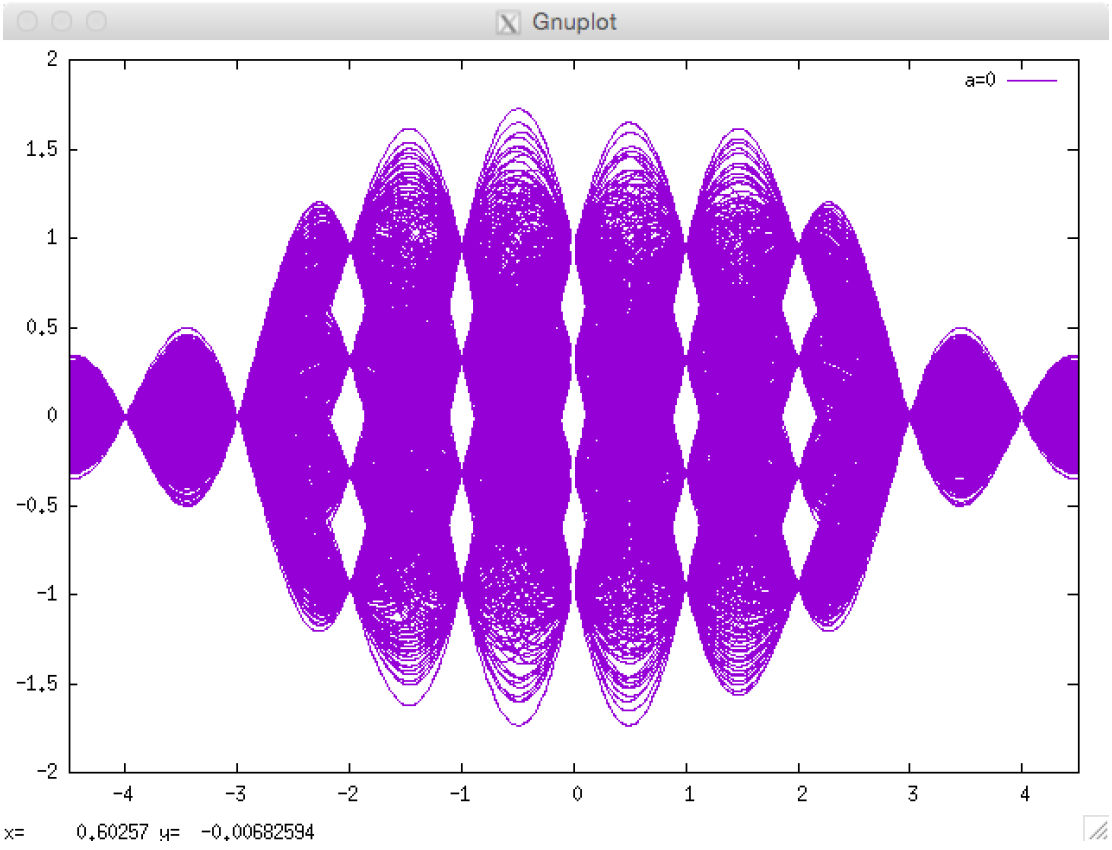

図 3.1 アイパターン　α =0(tusin11.c)

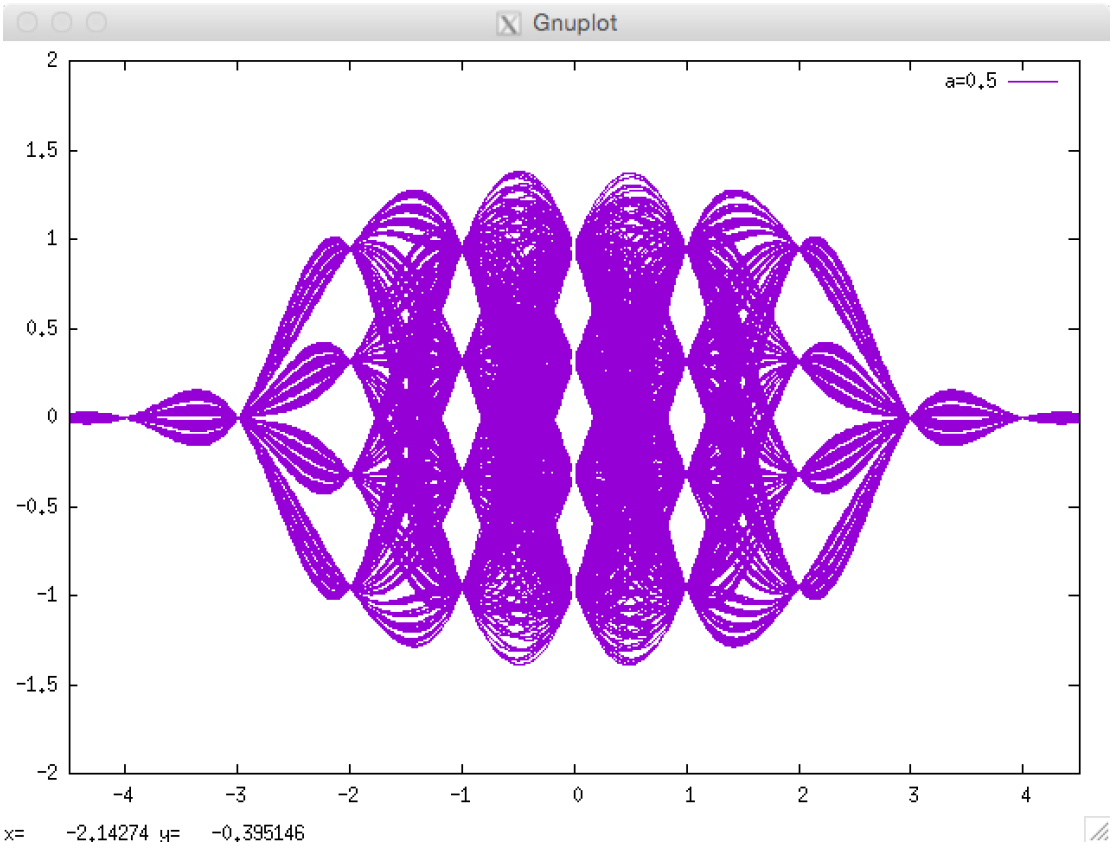

図 3.2 アイパターン　α =0.5(tushin10.c)

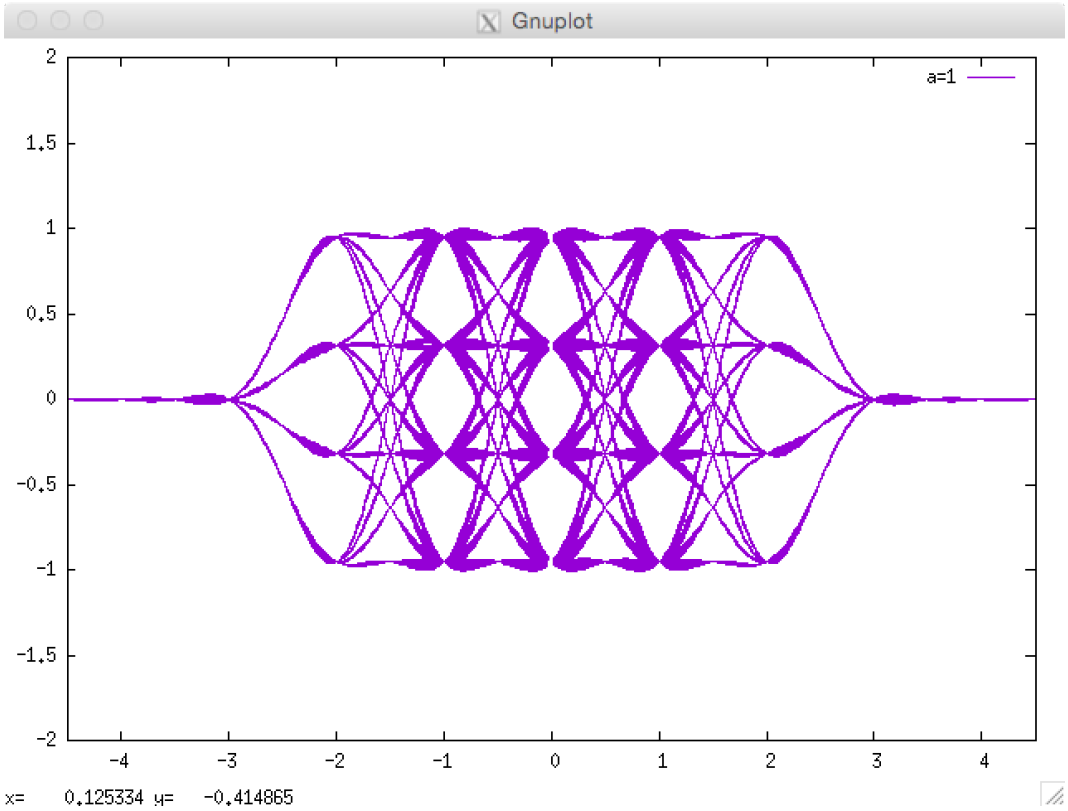

図 3.3 アイパターン　α=1(tushin.9)

次に、図 3.4 図 3.5 図 3.6 に α=0, α=0.5, α=1 のときの信号空間ダイアグラムを示す。信号空間ダイアグラムの波形数は、計算に時間がかかるため、10000 個とした。

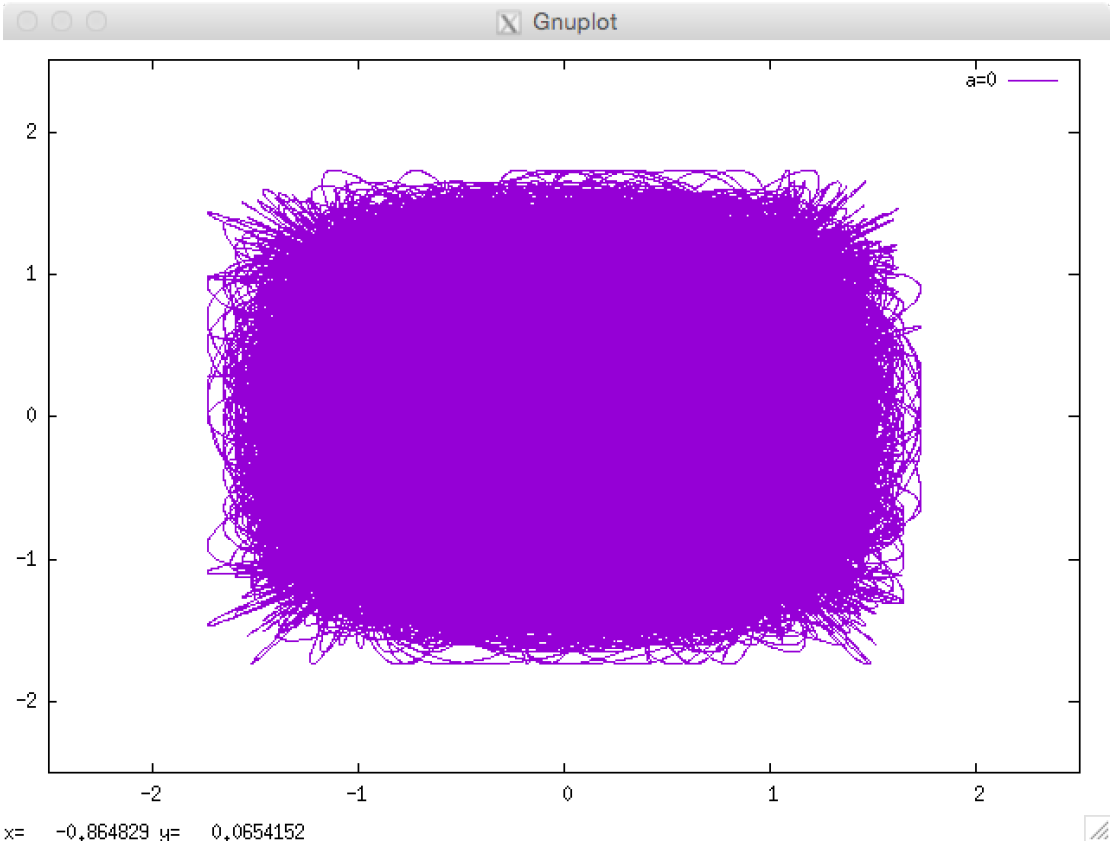

図 3.4 信号空間ダイアグラム　α =0 (tusin12.c)

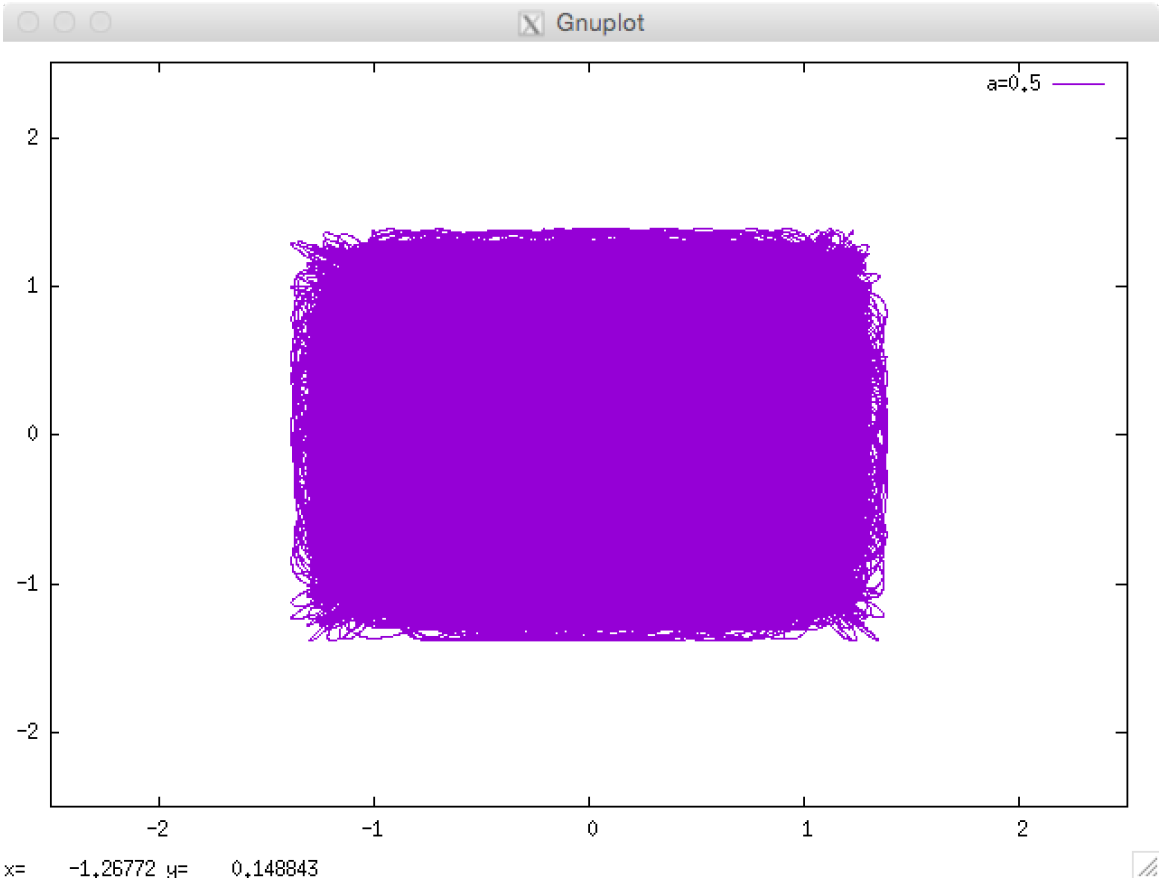

図 3.4 信号空間ダイアグラム　α =0.5(tushin13.c)

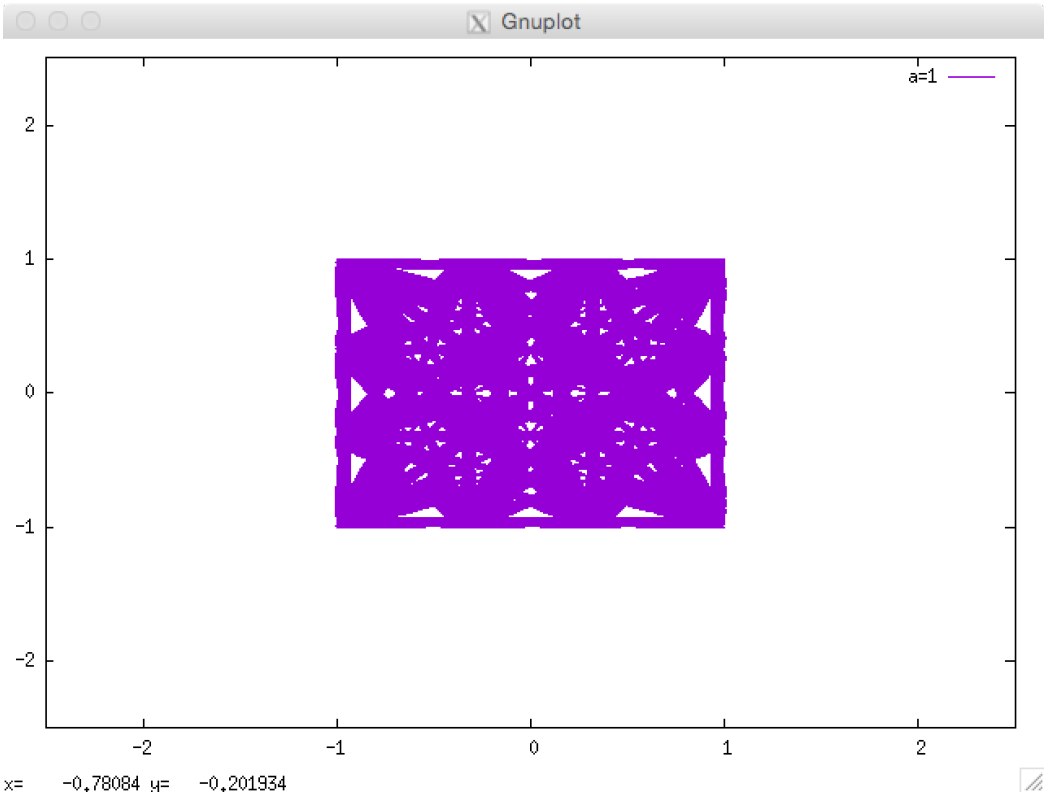

図 3.4 信号空間ダイアグラム　α =1(tushin14.c)

QPSK と 16QAM の比較

QPSK と 16QAM2 つの信号空間ダイアグラム、アイパターンを比較すると、α が 0 に近づくほど波形が混み、振動の大きいインパルス応答となることがわかる。さらに、同じロールオフ係数の QPSK と 16QAM で比較すると、16QAM の方が QPSK よりも振幅のパターンが I,Q 軸共に 2 つ多いので、その分サンプリングされる波形のパターンが多く(QPSK は 2^2 通り 16QAM は 4^4 通り)、その分、信号空間ダイアグラム、アイパターンは混雑した図となり、空白部分は少なかった。16QAM は 1 シンボルあたり 4 つ、QPSK は 1 シンボルあたり 2 つの bit を乗せることができ、16QAM の方が 1 シンボルあたりに乗せられる情報が多いが、その分割り当てる信号間の距離が近いため、アイパターンや信号ダイアグラムは QPSK よりも混み、QPSK に比べて質の悪い波形が出力されてしまう(謝り率を増大させる要因になりうる)というトレードオフの関係を、2 種類の変調方法で作成した信号空間ダイアグラムとアイパターンから確認することができた。

4. まとめ・感想

4.1 まとめとレポートの感想

信号空間ダイアグラム、アイパターンの観点からロールオフ係数を変えながら QPSK 信号を表示させることで、QPSK 信号の特徴、並びにコサインロールロールオフ係数の役割を、波形を確認しながら理解することができた(16QAM の方は、調べても出てこなかったのであっているかは自信がない)。また、16QAM と QPSK の信号空間ダイアグラムとアイパターンを比較し、シンボル

あたりの bit 数と波形の質とのトレードオフの関係を確認することができた。

逆フーリエ変換の導出も複雑で難しく、さらに c 言語を用いて複雑なインパルス応答を表示するというのもどのようにすれば良いか、アイパターンとはそもそもなにか、信号空間ダイアグラムもどう書けば良いかわからない等、次から次へと壁が現れ続け、テストも近いので諦めてしまおうかとも思ったが、ほとんど形ができ、また 1 つ 1 つの波形表示が、テキストと同じなるたびに達成感がとても強く、やってよかったと思う。理論を理解したあとも、C 言語の扱いに慣れておらず、gnuplot による波形表示なども今回初めて行ったため、波形表示方法の学習にも時間がかかってしまった。効率の良いプログラムを書くのに慣れていないため妥協した点もあり(信号を 5 桁の 2 進数で表したかった)、未熟なプログラムとなってしまったが、結果を確認できてよかったと思う。他の変調方法についても、時間に余裕ができたら確認してみたい。

4.2 講義の感想

前期の通信理論のように、スライドが配られていたので、スライドを読めば理解出来るのではないかと甘く見ていたが、いざ見てみると、授業の内容を聞いて、教科書を読んで、スライドを読んで、わからないことを調べてようやく大まかに意味がわかってくるような感じでとても難しく感じた。中間テストもあまり手応えがなかったため、このレポートは全部完成させようと頑張った。基礎の知識が曖昧で中間がうまくいかなかったのだと思うので、今からでも時間をかけて基礎を身につけ、知識を使えるようにしたい。

5. 参考文献

・高畑文雄、前原文明、笹森文仁共著、ディジタル無線通信入門 –電子書籍版- 、培風館(紙書籍書板)/2002、一般財団法人電波技術協会(電子書籍版)/2015

・前原文明教授配布、通信理論後半講義資料　第 1 回_2015.pdf 第 2 回_2015.pdf

・山本昌志、gnuplot によるグラフ作成、2007、http://www.yamamo10.jp/yamamoto/lecture/2007/5E_comp_app/gnuplot/mkgraph.pdf#search='c 言語+グラフ'

・アンリツ計測器カストマサービス株式会社計測サポートセンター、測定のイロハ、Eye パターンとは? ~高速デジタル信号の波形品質評価ツール、2011、http://www.anritsu-customersupport.com/Data/Sites/1/media/pdf/melmaga/201106-02.pdf#search

・Masahiro Takagi 、計測器基礎 A-09. Gnuplot によるグラフ作成(1)、2013、http://www.cc.kyoto-su.ac.jp/~mtkg/lecture/comp_A/2012/09.html

6. プログラムのソースコード

5.1 図 1 インパルス応答のロールオフ係数ごとの波形表示のために作成したプログラムのソースコード

```
#include <stdio.h>
#include <math.h>

int main(void) {
	FILE *gp;

	gp = popen("/Applications/gnuplot.app/bin/gnuplot -persist","w");
	fprintf(gp, "set xrange[-5:5]¥n");
	fprintf(gp,"plot pi**2*sin(pi*x)/(pi*x)*cos(pi*x)/(pi**2-4*pi**2*x**2) lt rgb "black" ¥n" );
	fprintf(gp,"replot pi**2*sin(pi*x)/(pi*x)*cos(0.75*pi*x)/(pi**2-4*0.75**2*pi**2*x**2)
¥n");
	fprintf(gp,"replot pi**2*sin(pi*x)/(pi*x)*cos(0.5*pi*x)/(pi**2-4*0.5**2*pi**2*x**2) ¥n");

	fprintf(gp,"replot pi**2*sin(pi*x)/(pi*x)*cos(0.25*pi*x)/(pi**2-4*0.25**2*pi**2*x**2)
¥n");
	fprintf(gp,"replot pi**2*sin(pi*x)/(pi*x)*1/(pi**2) ¥n");

	getchar();
	pclose(gp);

	return 0;

}
```

5.2　図 2.1,図 2.2,図 2.3 のアイパターンを示すために作成したプログラムのソースコード

まず、以下は α=1 の例である。

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define X 500
```

```
double randA() {
        rand(); rand();
        int j = rand()%2;
        if(j == 0) {
                return +1/sqrt(2);
         }else {
                return -1/sqrt(2);
         }
}

int main(void) {
        FILE *data,*gp;
        char *data_file;
        int i,k,m;
        double A[5],dx , x, y=0.0;

        data_file = "out.dat" ;
        data = fopen(data_file , "w");



for(int l=0;l<80;l++){
        for(k= 0;k<5;k++){
                A[k] = randA();
        }

        dx = 4*M_PI/X;
        for(i=0;i<=X;i++) {
                y=0;
                x = -2*M_PI+i*dx;
                for(m = -2; m<3 ; m++){

                        y += A[m+2]* pow(M_PI,2) * sin(M_PI*(x-m)) /(M_PI*(x-
m))*cos(M_PI*(x-m))/(pow(M_PI,2)-4*pow(M_PI,2)*pow((x-m),2));    /*...#1*/              }
        fprintf(data,"%f %f¥n",x,y);
```

```
        }
}
        fclose(data);


        gp = popen("/Applications/gnuplot.app/bin/gnuplot -persist","w");
        fprintf(gp,"set xrange [-4.5:4.5]¥n");
        fprintf(gp,"set yrange [-1.5:1.5]¥n");

                fprintf(gp,"plot ¥"%s¥" with lines linetype 1 title ¥"a=1¥"¥n",data_file);    /*...#2*/


                pclose(gp);


                return 0;
}
```

注

/*...#1*/は、

α=0.5 のとき　y += A[m+2]* pow(M_PI,2) * sin(M_PI*(x-m)) /(M_PI*(x-m))*cos(0.5*M_PI*(x-m))/(pow(M_PI,2)-4*pow(0.5,2)*pow(M_PI,2)*pow((x-m),2));

α=0 のとき y += A[m+2]* pow(M_PI,2) * sin(M_PI*(x-m)) /(M_PI*(x-m))/ (pow(M_PI,2));

/*...#2*/は、わかりやすくするため α=0.5 のとき title ¥"a=0.5¥" , α=0 のとき title ¥"a=0¥"とした。

5.3 図 2.4、図 2.5、図 2.6 の信号空間ダイアグラム表示のために作成したプログラムのソースコード

以下に示したのは、α=1 の例である。

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define X 500
```

```
double randA() {
        int j = rand()%2;
        if(j == 0) {
                return +1/sqrt(2);
         }else {
                return -1/sqrt(2);
         }
}

double randA2() {
        rand(); rand();
        int j = rand()%2;
        if(j == 0) {
                return +1/sqrt(2);
         }else {
                return -1/sqrt(2);
         }
}

int main(void) {
         FILE *data,*gp;
         char *data_file;
         int i,k,m;
         double A[5],A2[5],dx , x, y=0.0,y2=0.0;
         /*I*/
         data_file = "outIQ.dat" ;
         data = fopen(data_file , "w");


for(int l=0;l<900;l++){

         for(k= 0;k<5;k++){
                A[k] = randA();
                A2[k] = randA2();
```

```
		}

		dx = 4*M_PI/X;
		for(i=0;i<=X;i++) {
			y=0;
			y2=0;
			x = -2*M_PI+i*dx;
			for(m = -2; m<3 ; m++){

				y += A[m+2]* pow(M_PI,2) * sin(M_PI*(x-m)) /(M_PI*(x-
m))*cos(M_PI*(x-m))/(pow(M_PI,2)-4*pow(M_PI,2)*pow((x-m),2));	/*...#1*/
				y2 += A2[m+2]* pow(M_PI,2) * sin(M_PI*(x-m)) /(M_PI*(x-
m))*cos(M_PI*(x-m))/(pow(M_PI,2)-4*pow(M_PI,2)*pow((x-m),2));	/*...#1*/

		}
		fprintf(data,"%f %f¥n",y,y2);

	}
}
	fclose(data);

	gp = popen("/Applications/gnuplot.app/bin/gnuplot -persist","w");
	fprintf(gp,"set xrange [-2:2]¥n");
	fprintf(gp,"set yrange [-2:2]¥n");

		fprintf(gp,"plot ¥"%s¥" with lines linetype 1 title ¥"a=1¥"¥n",data_file);



		pclose(gp);

		return 0;
}
```

注

/*...#1*/ の pow(M_PI,2) * sin(M_PI*(x-m)) /(M_PI*(x-m))*cos(M_PI*(x-m))/(pow(M_PI,2)-

4*pow(M_PI,2)*pow((x-m),2));　の部分について、
α=0.5 のとき　pow(M_PI,2) * sin(M_PI*(x-m)) /(M_PI*(x-m))*cos(0.5*M_PI*(x-m))/(pow(M_PI,2)-4*pow(0.5,2)*pow(M_PI,2)*pow((x-m),2));

α=0 のとき pow(M_PI,2) * sin(M_PI*(x-m)) /(M_PI*(x-m))/(pow(M_PI,2));
となる。

/*...#2*/ 4.2 節　節末注の/*...#2*/と同様

5.4 16QAM のアイパターン、信号空間ダイアグラムのプログラムのソースコード
基本は 5.2 のアイパターン 5.3 信号空間ダイアグラムと同じであるため、異なる部分だけを示す。
異なる部分は、
randA()関数

```
randA() {
        rand(); rand();
        int j = rand()%4;
        if(j == 0) {
                return +3/sqrt(10);
         }else if(j == 1){
                return 1/sqrt(10);
         }else if(j == 2){
                return -3/sqrt(10);
         }else{
                return -1/sqrt(10);
         }
}
```

randA2() 関数

```
double randA2() {

        int j = rand()%4;
        if(j == 0) {
                return +3/sqrt(10);
         }else if(j == 1){
```

```
		return 1/sqrt(10);
	}else if(j == 2){
		return -3/sqrt(10);
	}else{
		return -1/sqrt(10);
	}
}
```

であり、その他の変更部分は 5.2,.5.3 の注と同様である(コサインロールオフフィルタのインパルス応答は変わらないため)。波形の表示個数は、アイパターンのとき 1000 個、信号空間ダイアグラムの時 10000 個である。

例として $\alpha = 0$ のときのアイパターン、信号空間ダイアグラムのプログラムを載せる。
アイパターン

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define X 500

double randA() {
	rand(); rand();
	int j = rand()%4;
	if(j == 0) {
		return +3/sqrt(10);
	 }else if(j == 1){
		return 1/sqrt(10);
	 }else if(j == 2){
		return -3/sqrt(10);
	}else{
		return -1/sqrt(10);
	}
}

int main(void) {
	FILE *data,*gp;
```

```
        char *data_file;
        int i,k,m;
        double A[5],dx , x, y=0.0;

        data_file = "out.dat" ;
        data = fopen(data_file , "w");



for(int l=0;l<1000;l++){
        for(k= 0;k<5;k++){
                A[k] = randA();
        }

        dx = 4*M_PI/X;
        for(i=0;i<=X;i++) {
                y=0;
                x = -2*M_PI+i*dx;
                for(m = -2; m<3 ; m++){

                        y += A[m+2]* pow(M_PI,2) * sin(M_PI*(x-m)) /(M_PI*(x-
m))/(pow(M_PI,2)); /*...#1*/
        }
        fprintf(data,"%f %f¥n",x,y);

    }
}
    fclose(data);



    gp = popen("/Applications/gnuplot.app/bin/gnuplot -persist","w");
    fprintf(gp,"set xrange [-4.5:4.5]¥n");
    fprintf(gp,"set yrange [-2:2]¥n");

        fprintf(gp,"plot ¥"%s¥" with lines linetype 1 title ¥"a=0¥"¥n",data_file);    /*...#2*/
```

```
	pclose(gp);


	return 0;
}
```

信号空間ダイアグラム

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define X 500

double randA() {
	rand(); rand();
	int j = rand()%4;
	if(j == 0) {
		return +3/sqrt(10);
	 }else if(j == 1){
		return 1/sqrt(10);
	 }else if(j == 2){
		return -3/sqrt(10);
	 }else{
		return -1/sqrt(10);
	 }
}

double randA2() {

	int j = rand()%4;
	if(j == 0) {
		return +3/sqrt(10);
	 }else if(j == 1){
		return 1/sqrt(10);
```

```
	}else if(j == 2){
		return -3/sqrt(10);
	}else{
		return -1/sqrt(10);
	}
}

int main(void) {
	FILE *data,*gp;
	char *data_file;
	int i,k,m;
	double A[5],A2[5],dx , x, y=0.0,y2=0.0;

	data_file = "out.dat" ;
	data = fopen(data_file , "w");



for(int l=0;l<10000;l++){
	for(k= 0;k<5;k++){
		A[k] = randA();
		A2[k] =randA2();
	}

	dx = 4*M_PI/X;
	for(i=0;i<=X;i++) {
		y=0;
		y2=0;
		x = -2*M_PI+i*dx;
		for(m = -2; m<3 ; m++){

			y += A[m+2]* pow(M_PI,2) * sin(M_PI*(x-m)) /(M_PI*(x-
m))/(pow(M_PI,2)); /*...#1*/
			y2 += A2[m+2]* pow(M_PI,2) * sin(M_PI*(x-m)) /(M_PI*(x-
m))/(pow(M_PI,2)); /*...#1*/
```

```
            }
            fprintf(data,"%f %f¥n",y,y2);

        }
}
        fclose(data);



        gp = popen("/Applications/gnuplot.app/bin/gnuplot -persist","w");
        fprintf(gp,"set xrange [-2.5:2.5]¥n");
        fprintf(gp,"set yrange [-2.5:2.5]¥n");

            fprintf(gp,"plot ¥"%s¥" with lines linetype 1 title ¥"a=0¥"¥n",data_file);    /*...#2*/



            pclose(gp);



            return 0;
}
```