

[6b] 音声分析（ケプストラム）プリエンファシス

[以下に示す青色のコマンドは Scilab のコンソール上で実行できます.]

■ グラフィック・ウィンドウの初期設定 ([1] 参照)

```
exec('DefaultWindow.sce');
```

を実行します。

■ 音声(母音)の周波数分析

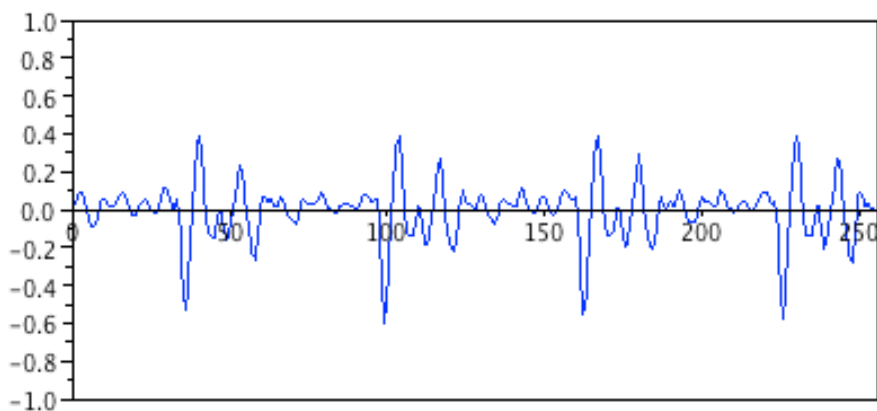
「あ——」という音声(母音 /あ/)の周波数分析をしましょう。データの取得法はこれまでと同様です。

```
x = wavread('a_8k.wav');  
y = resize_matrix(x,1,256); plot(y)
```

グラフィックウィンドウのサイズ等を変更します。(しなくてもかまいません)

```
n = 0:255;  
clf();  
plot2d(n,y,frameflag = 1,axesflag = 1, rect=[0,-1,256,1], style = 2);  
f = gcf(); a = gca();  
f.figure_position = [800,40];  
f.figure_size = [480,320];  
a.x_location = "origin";
```

波形を示します。

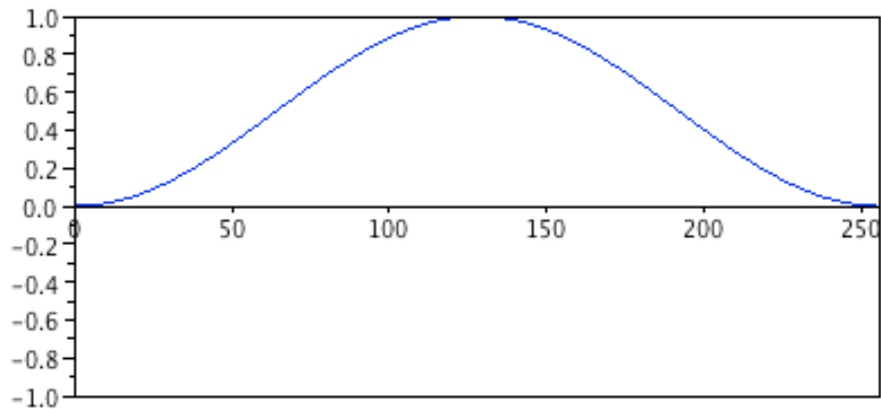


ハニング窓を作成します。

```

n = 0:255;
han = (1 - cos(2 * %pi * n / 256)) / 2; scf();
f = gcf(); a = gca();
f.figure_position = [800,40+f.figure_id*20];
f.figure_size = [480,320];
a.x_location = "origin";
plot2d(n, han, frameflag = 1, axesflag = 1, rect=[0,-1,256,1], style = 2);

```

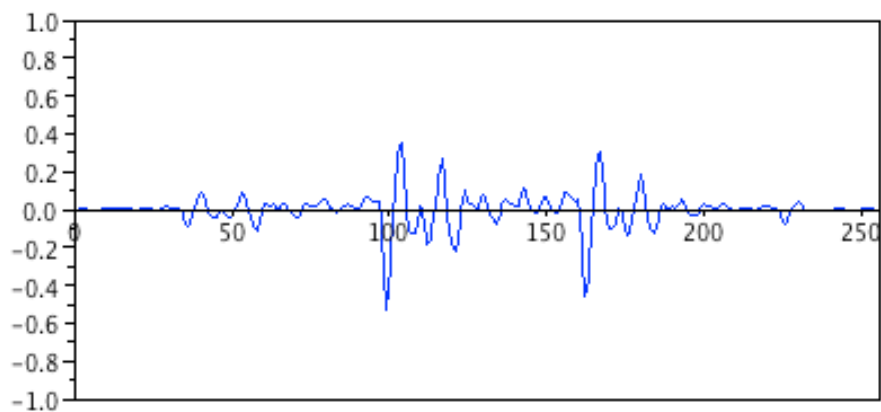


取得した波形にハニング窓を掛けます.

```

yh = y .* han;
//配列の要素ごとのかけ算は '.*' を使用します.
scf();
f = gcf(); a = gca();
f.figure_position = [800,40+f.figure_id*20];
f.figure_size = [480,320];
a.x_location = "middle";
plot2d(n, yh, frameflag = 1, axesflag = 1, rect=[0,-1,256,1], style = 2);

```



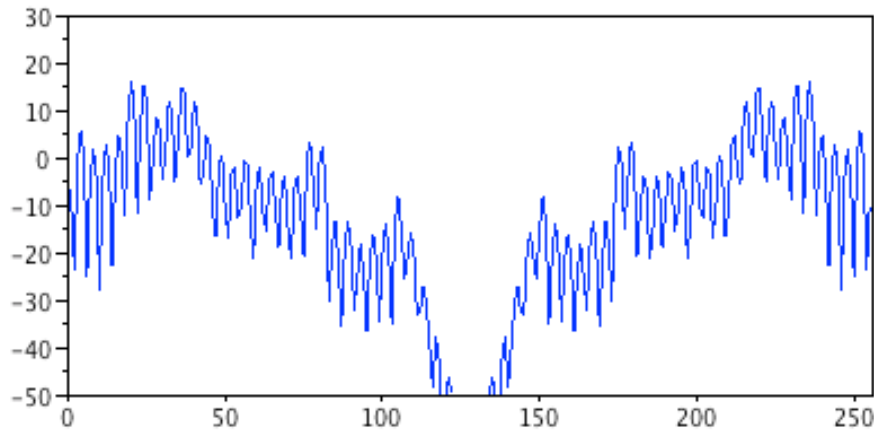
続いて、フーリエ変換を実行します.

```

Yh = fft(yh);

```

```
lgYh = 20 * log10(abs(Yh));
plot2d(n, lgYh, frameflag = 1, axesflag = 1, rect=[0,-50,256,30], style = 2);
```



スペクトルの絶対値の対数をとっていますので、「対数スペクトル」といいます。
 グラフに示されたスペクトルの右半分は負の周波数成分で左半分と対称になっていることはもう理解できますね。データのサンプリング周波数は8 kHzですから、周波数軸はグラフ中央(128の辺り)が約4 kHzとなります。

-----《新規追加処理 ここから》-----

■プリエンファシス

上のスペクトルを見て気がつくことは、周波数が高くなるに従い減少傾向が認められます。共振特性(フォルマント)を推定するには全体的に平坦な特性を有していた方が好ましいと考えられます。そのために行う処理が「プリエンファシス」です。この処理によって高域を強調します。

具体的には、信号 $x(n)$ に対して次に示す差分演算を行います。

$$z(n) = x(n+1) - 0.98 * x(n)$$

プログラムでは次のようになります。

```
z1 = x(1:256);    // x(n) に対応
z2 = x(2:257);    // x(n+1) に対応
z = z2 - 0.98*z1; // 差分演算
```

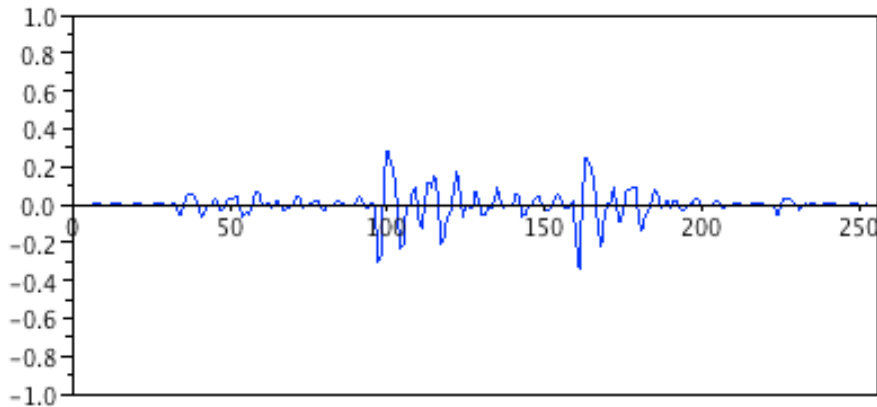
この信号 z について上で実行した y に対する処理を繰り返します。

```
y = z;
```

としてハニング窓を掛けるところから再び実行します。先に求めた信号のスペクトルを比較のために保存しておきます。

```
Yh0 = Yh; lgYh0 = lgYh;
yh = y .* han;
//配列の要素ごとのかけ算は '.*' を使用します。
scf();
f = gcf(); a = gca();
```

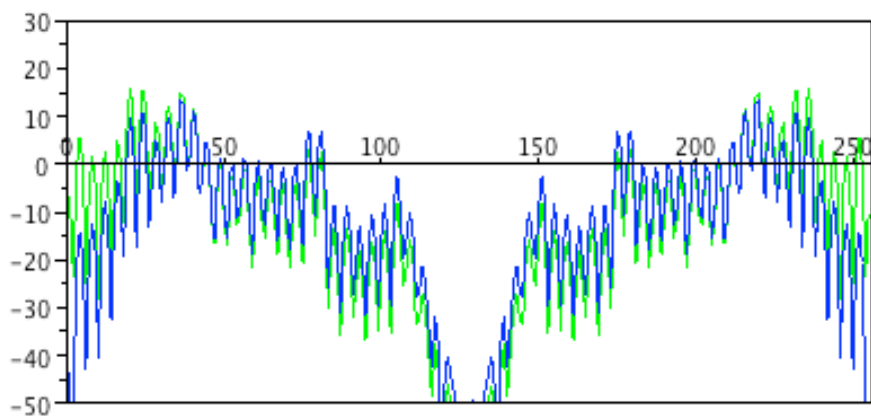
```
f.figure_position = [800,40+f.figure_id*20];
f.figure_size = [480,320];
a.x_location = "middle";
plot2d(n, yh, frameflag = 1, axesflag = 1, rect=[0,-1,256,1], style = 2);
```



```
Yh = fft(yh);
lgYh = 20 * log10(abs(Yh));

scf();
plot2d(n, lgYh0, frameflag = 1, axesflag = 1, rect=[0,-50,256,30], style = 3);

plot2d(n, lgYh, frameflag = 1, axesflag = 1, rect=[0,-50,256,30], style = 2);
```



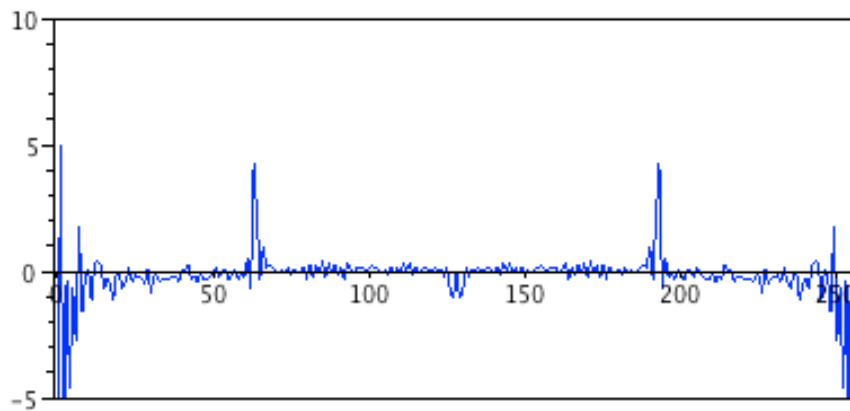
音声のスペクトルは上のようになります。緑が当初のスペクトル、青がプリエンファシス処理後のスペクトルです。高域が増大しているのが分かります。

-----《新規追加処理 ここまで》-----

◆ ケプストラム(Cepstrum)

```
ceps = ifft(lgYh);
scf(); f= gcf(); a=gca();
f.figure_position = [800,40+f.figure_id*20];
f.figure_size = [480,320];
```

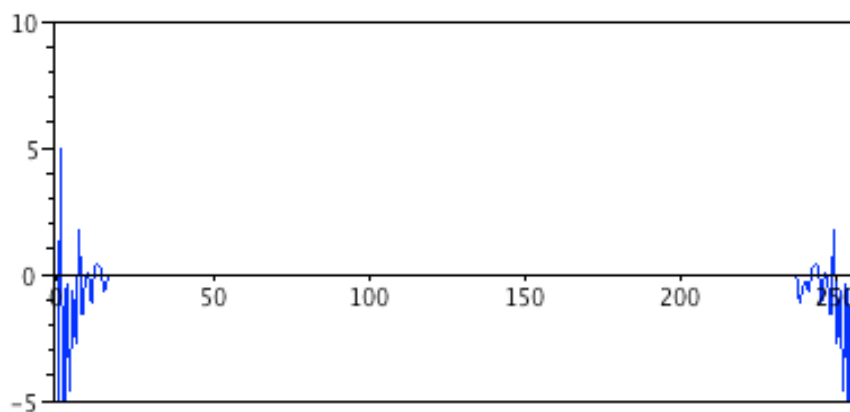
```
a.x_location = "origin";
plot2d(n, ceps, frameflag = 1, axesflag = 1, rect=[-1,-5,257,10], style = 2);
```



対数スペクトルの逆フーリエ変換を「ケプストラム (Cepstrum)」といいます。これによって、対数スペクトルを構成するなだらかな成分と、細かな変動成分を分離することができます。前者を「スペクトル包絡」、後者を「スペクトルの微細構造」ということがあります。

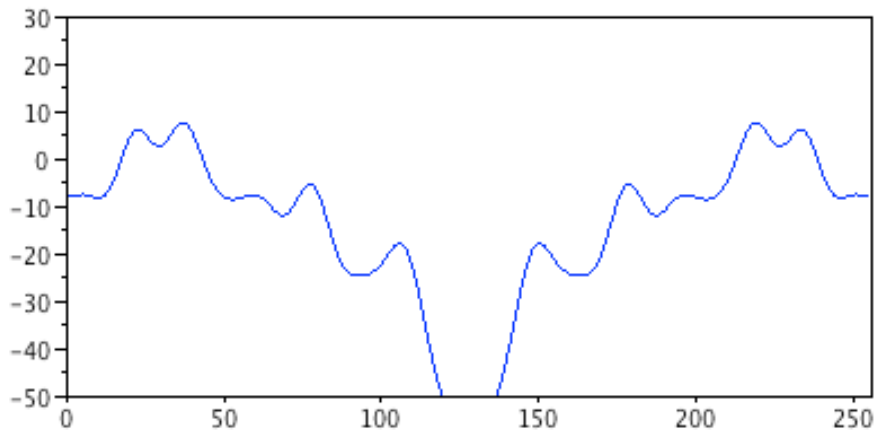
ケプストラムにローパスリフタ(ケプストラムにおけるフィルタ)をかけます。

```
ii=18;
for i = ii: 256 - ii,
    ceps(i) = 0;
end
scf();
f = gcf(); a = gca();
f.figure_position = [800,40+f.figure_id*20];
f.figure_size = [480,320];
a.x_location = "origin";
plot2d(n, ceps, frameflag = 1, axesflag = 1, rect=[-1,-5,257,10], style = 2);
```



このように、ケプストラムの高域成分を除去した後に、フーリエ変換によって元に戻すと

```
fceps = fft(ceps);  
scf(); f = gcf(); a = gca();  
f.figure_position = [800,40+f.figure_id*20];  
f.figure_size = [480,320];  
plot2d(n, fceps, frameflag = 1, axesflag = 1, rect=[0,-50,256,30], style = 2);  
a = gca(); a.x_location = "bottom";
```



スペクトル上の3つのピークの周波数を求めてみましょう。このピークは「声道」(声帯から口唇に至る空隙)の共振特性を示しています。これを「フォルマント」といい、低い周波数から順に、第1フォルマント、第2フォルマント、第3フォルマントといいます。このフォルマントの位置で音声(母音)は特徴づけられます。

ピーク値を推定しますと

```
sp = real(fceps);  
for i = 2:length(sp)/2,  
    if (sp(i+1) - sp(i))<0 & (sp(i) - sp(i-1))>0 then  
        printf("%d\t%f\t%f\n",i,(i-1)*8000/256,sp(i))  
    end  
end
```

25	750.000000	1.333543
39	1187.500000	6.402080
61	1875.000000	-5.270042
78	2406.250000	-1.201533
107	3312.500000	-12.041058

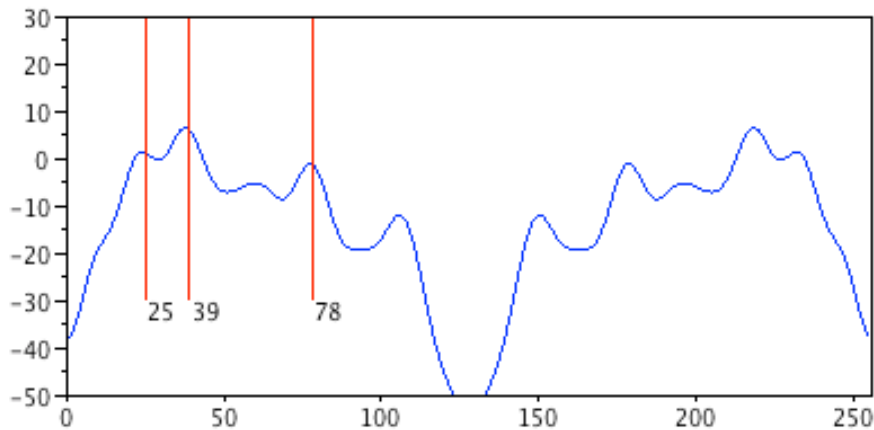
となります。小さなピークは無視することにしますと、

```
i1=25; plot([i1,i1],[-30,30],'r-');xstring(i1,-35,string(i1));  
i2=39; plot([i2,i2],[-30,30],'r-');xstring(i2,-35,string(i2));  
i3=78; plot([i3,i3],[-30,30],'r-');xstring(i3,-35,string(i3));  
printf('F1, F2, F3 = [%d, %d, %d]', [i1-1,i2-1,i3-1]*8000/256);
```

[F1, F2, F3] = [750, 1187, 2406]

単位は Hz です.母音「あ」のフォルマントが求まりました.

Scilab の配列は1から始まりますから,周波数を考える場合,1を減じています.



第1フォルマントを横軸に, 第2フォルマントを縦軸にとってフォルマントを図示してみましょう.

これをフォルマント分布図といいます. まず, 日本語の母音/a/, /i/, /u/, /e/, /o/ の分布と, 男声, 女声の平均値を示します.

次のプログラムを作成して実行します. プログラム名: FormantMap.sce

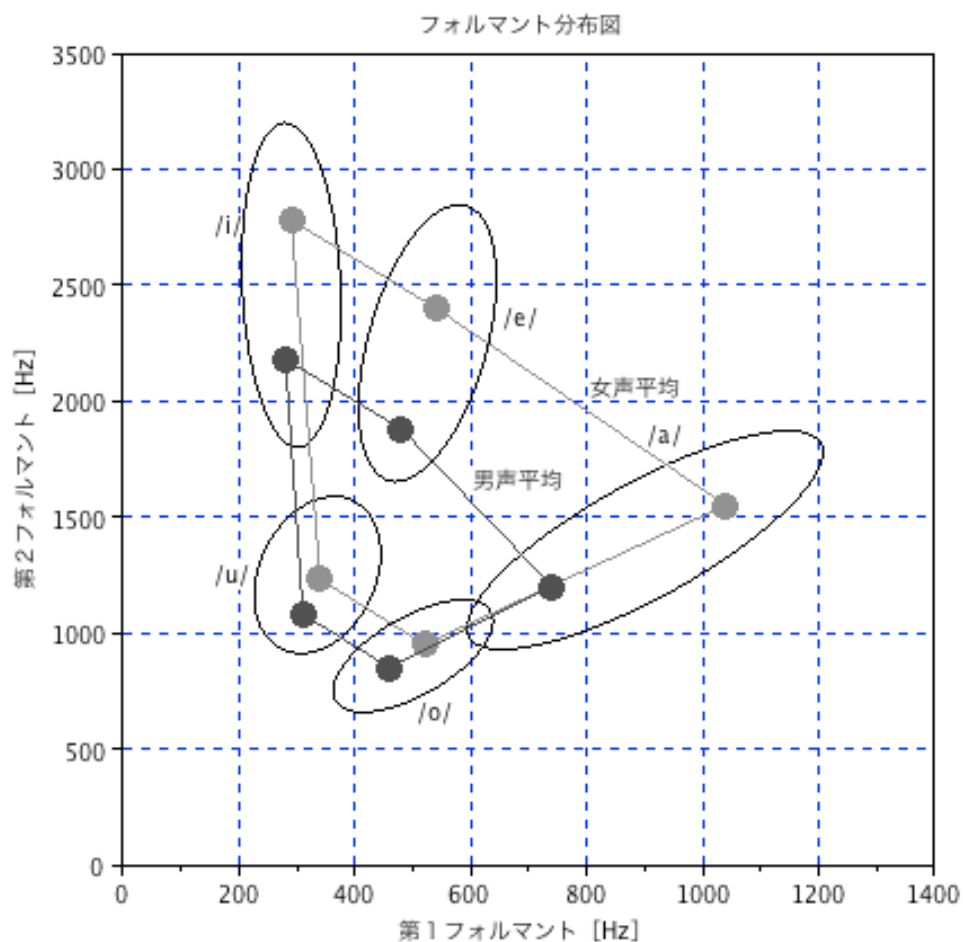
```
exec(' oval2. sci ');
[x, y]=oval(500, 900, 260, 100, %pi*0.37);plot2d(x, y, rect=[0, 0, 1400, 3500])
[x, y]=oval(335, 1250, 340, 105, %pi*0.47);plot2d(x, y)
[x, y]=oval(290, 2500, 700, 85, %pi*0.505);plot2d(x, y)
[x, y]=oval(525, 2250, 600, 105, %pi*0.47);plot2d(x, y)
[x, y]=oval(900, 1400, 540, 160, %pi*0.33);plot2d(x, y)
xstring(900, 1800, string("/a/"))
xstring(650, 2300, string("/e/"))
xstring(150, 2700, string("/i/"))
xstring(150, 1200, string("/u/"))
xstring(500, 600, string("/o/"))
xstring(800, 2000, string("女声平均"))
xstring(600, 1600, string("男声平均"))
p1 = [740, 1200]; p2 = [480, 1880]; p3 = [280, 2180]; p4 = [310, 1080]; p5 = [460, 850];
p6 = [1040, 1550]; p7 = [540, 2400]; p8 = [290, 2780]; p9 = [340, 1230]; p0 = [520, 950];
Line=[p6; p7; p8; p9; p0; p6]';
plot2d(Line(1, :), Line(2, :), style=color(0, 128, 0))
a=gce();c=a.children;
c.mark_style=0;c.mark_size=1;
c.mark_foreground=color(128, 128, 128);
c.foreground=color(128, 128, 128);
Line=[p1; p2; p3; p4; p5; p1]';
plot2d(Line(1, :), Line(2, :), style=color(0, 0, 128))
a=gce();c=a.children;
c.mark_style=0;c.mark_size=1;
c.mark_foreground=color(64, 64, 64);
c.foreground=color(64, 64, 64);
a=gca();a.box="on";
a.grid=[2, 2];
a.x_label.text="第1フォルマント [Hz] ";
a.y_label.text="第2フォルマント [Hz] ";
```

```
a.title.text="フォルマント分布図";
```

oval2.sci

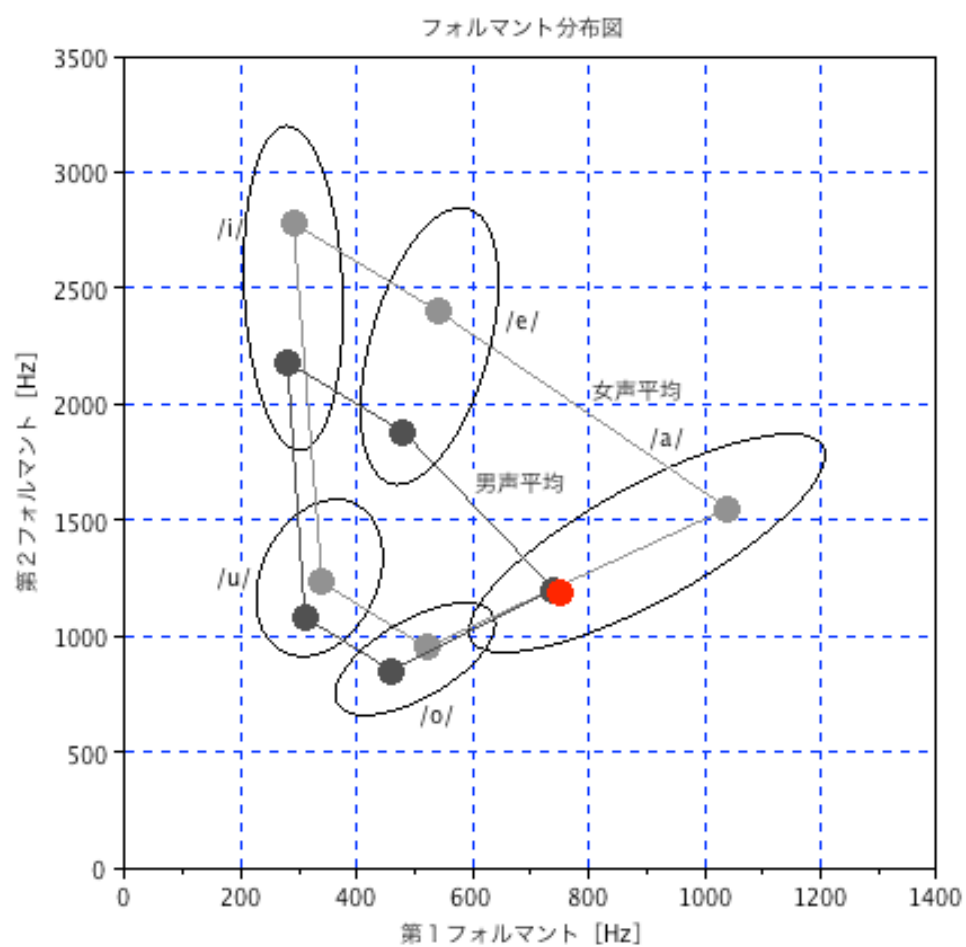
```
function [X, Y]=oval(x, y, w, h, theta)
// (x,y):center, w:width, h:height, theta: angle
i=0:0.02:1;
X=w*cos(i*2*pi);
Y=h*sin(i*2*pi);
XY=rotate([X;Y], theta);
X=XY(1, :)+x;
Y=XY(2, :)+y;
endfunction
```

これを実行すると



このような図ができ上がります。ここに、上で求めたフォルマントをプロットしてみましょう。

```
F1=750;F2=1187;
plot2d(F1,F2)
a=gce();c=a.children;
c.mark_style=0;c.mark_size=1;c.mark_foreground=color(255,0,0);
```



プリエンファシス処理前の結果と比較しても、特に、大きな違いは生じていない、と言っておきましょう。