

# ヒューマンインターフェース

-report2-

055702B

池野谷克俊

2007年6月30日 土曜日

## 1 課題

- (1) プログラムを解読する。(どこで何を行っているかを数式とともに示す。)
- (2) 提供プログラムでは Levinson-Durbin 法を SCILAB の関数 `lev()` で実現しています。関数を使わず SCILAB 言語でプログラミングして実現しましょう。
- (3) LPS を算出し、スペクトル、単位円上に PLOT しましょう (LSP の算出は LSP 解説.pdf を参考にしてください)。
- (4) プログラムの最後に伝達特性=0 の根を `roots()` により算出し、極周波数と帯域幅を極周波数の小さい順に表示するプログラムを追加してください。0 から 5 KHz の極だけを抽出し、極周波数で昇順にソーティングしましょう。実軸に根が存在する場合もあるので、0 から 5 KHz の極は次数の半分とは限りません。結果があっているか、スペクトルと極配置図により確認しましょう。
- (5) 男性音と女性音それぞれで、分析長を 100 から 300、分析次数を 8 から 20 程度に変化させ、スペクトルがどう変化するかを調べ、その結果について考察しましょう。

## 2 解答及び考察

### 2.1 課題 1

#### Part.1

```
fft_len=512;           // FFT のサンプル数
start=5000;           // 分析位置
len=300;              // 分析長
order=14;             // 分析次数
pre_emp=0.0;          // プリエンファシス
print(%io(2), 'start= '); start = read(%io(1),1,1);
print(%io(2), start, len, fft_len, order);

x=loadwave('1.wav'); // 1.wav というファイルの読み込み
```

この部分では、変数の初期値の設定、ファイルの読み込みなどが行われている。

## Part.2

```
// Hamming 窓掛け&プリエンファシス
for i=1:len
    win(i) = 0.54 - 0.46 * cos(2 * %pi * i / len);
    x1(i) = (x(i+start)-pre_emp*x(i-1+start)) * win(i);
end
for i=len+1:fft_len // ゼロ詰め
    x1(i) = 0;
    win(i) = 0;
end
```

最初の for 分で、

$$win(i) = 0.56 - 0.46 \times \cos\left(\frac{2\pi i}{len}\right)$$

という演算を行うことで、ハミング窓を決定している。  
窓掛けは次式で行われている。

$$x(i) = \{x(i + start) - pre\_emp \times x(i - 1 + start)\} \times win(i)$$

ただし、変数 *pre\_emp* の値は 0 なので

$$x(i) = x(i + start) \times win(i)$$

となる。

次の for 文では *win(i)*, *x(i)* 余った配列部分に 0 を代入している。

## Part.3

```
// DFT 対数スペクトルの算出
fft_spc=20*log10(abs(fft(x1,-1)));

// 自己相関関数の算出
j=1;
for i=0:order
    r(j)=0;
    for n=1:len-i
        r(j)=r(j)+x1(n)*x1(n+i);
    end
    j=j+1;
end
```

DFT 対数スペクトルの算出部分ではフーリエ変換を行い、abs 関数で絶対値に変換した後、 $20 * \log_{10}()$  によって dB の値を算出している。式を以下に示す。

$$fft\_spc = 20 \times \log_{10} \left| \sum_{n=1}^N x1(n) \times e^{j2\pi(k-1)\left(\frac{n-1}{N}\right)} \right|$$

自己相関関数の算出部分では、以下の式によって関数を算出している。

$$r(i) = \sum_{t=0}^{N-1-i} x1(t) \times x1(t+i)$$

#### Part.4

```
// Levinson 算法による LPC 分析
[ar,sigma2,rc]=lev(r); // sigma2=r(0)+_sum_{i=1}^order(r(i)*a(i))

// LPC 対数スペクトルの算出
a(1)=1;
for i=1:order
    a(i+1)=ar(i);
end
for i=order+1:fft_len-1 // ゼロ詰め
    a(i+1)=0;
end
ar_spc=-20*log10(abs(fft(a,-1)))+10*log10(sigma2);
```

Levinson 算法による LPC 分析の部分では、lev 関数によって、LPC 分析を行っている。ar は LPC 係数、sigma2 は予測誤差、rc は PARCOR 係数を示している。

LPC 対数スペクトルの算出の部分では、a(1) に 1 を代入し、a(2) からは LPC 係数が代入されている。それから配列の余った部分に 0 を代入している。最後に DFT 対数スペクトルと同様に dB 単位へ変換を行い、予測誤差との和を算出している。

#### Part.5

```
// 残差のスペクトルの算出
for n=1:len
    res(n)=x(start+n);
    for i=1:order
        res(n)=res(n)+ar(i)*x(start+n-i);
    end
end
for n=len+1:fft_len
    res(n)=0;
end
res_spc=20*log10(abs(fft(res,-1)));
```

残差のスペクトルの算出は以下の式によって求められる。

$$res(n) = x(start + n) + \sum_{i=1}^{order} ar(i) \times x(start + n - i)$$

上の演算の後、DFT 対数スペクトルと同様に dB 単位へ変換を行っている。

## Part.6

```
// スペクトルの描画
xset('window',1); xbas(1);
rect=[1,min(fft_spc),fft_len/2,max(fft_spc)];
tics=[2,4,2,4];
plotframe(rect,tics, [%f,%f], ['LPC', 'Freq.', 'Amp. [dB]'], [0,0,1,0,0.5]);
n=1:fft_len/2;
plot2d(n,fft_spc(n),1,"000");
plot2d(n, ar_spc(n),2,"000");
plot2d(n,res_spc(n),3,"000");

// 極を算出し描画
HAR=poly(a(1:order+1), 'z', 'coeff');
pp=roots(HAR);
for i=1:order
    pp(i)=1/pp(i);
end
x=0:0.1:2*pi;
// xbas();
rect=[-1,-1,1,1];
tics=[2,5,2,5];
plotframe(rect,tics, [%f,%f], ["Unit Circle", "Re.", "Im."], [0.25,0.5,0.5,0.5]);
plot2d(cos(x),sin(x),1,"000");
ra=real(pp); ia=imag(pp);
plot2d(ra,ia,-3);
xgrid();
```

この部分では、グラフの表示など出力に関する処理が行われている。

## 2.2 課題 2

以下に追加したソースを示す。

## ソースコード

```
// Levinson 算法による LPC 分析
[ar, sigma2, rc]=lev(r); // sigma2=r(0)+_sum_{i=1}^order(r(i)*a(i))

//lev 関数を使わない場合
w=r(2:order+1);
u=r(1:order);
a2=zeros(order,order);

for m=1:order
    k(m)=w(m-1+1)/u(m-1+1);
    u(m+1)=u(m-1+1)*(1-k(m)^2);

    for i=1:m-1
        a2(m,i)=a2(m-1,i)-k(m)*a2(m-1,m-i);
    end

    a2(m,m)=-k(m);

    if(m==order) then break,end;

    w(m+1)=r(m+1+1);

    for i=1:m
        w(m+1)=w(m+1)+a2(m,i)*r(m+1-i+1);
    end
end

//デバック用
ar2=a2(order,:)' ;
sigma2N=u(order+1);

//ar=a2(order,:)' ;
//sigma2=u(order+1);
rc=k;
```

初めに lev 関数を使わない場合で、うまく計算が行われているか確かめるために、 $[ar, \sigma2, rc] = lev(r)$ ; をコメントアウトせずに実行し、lev 関数を使った場合の値との差を求めてみた。以下にその実行結果を示す。

## 実行結果

```
-->exec('lpc2.sci')
-->fft_len=512;          // FFT$B$N%5%$W%k?t(B
-->start=5000;          // $BJ,@00LCV(B
-->len=300;             // $BJ,@0D9(B
-->order=14;           // $BJ,@0<!?t(B
-->pre_emp=0.0;        // $B%W%j%(%s%U!%7%9(B
-->print(%io(2),'start= '); start = read(%io(1),1,1);

start=
-->3000

          ~省略~

-->ar2 - ar
ans =

    1.0E-13 *
    0.0932587
   - 0.2087219
    0.2009504
   - 0.1265654
    0.1576517
   - 0.1454392
    0.0310862
   - 0.0693889
    0.2003953
   - 0.1887379
    0.1709743
   - 0.2275957
    0.1948441
   - 0.0591194
-->sigma2 - sigma2N
ans =

   - 1.518E-17
```

実行結果から分かるように、誤差が小さいので、lev 関数を使わない場合でも上手く演算できていることがわかる。

次に、 $[ar, \sigma2, rc] = lev(r)$ ; とデバック用の場所の2行をコメントアウトし  $ar = a2(order, :)$ ;  $\sigma2 = u(order + 1)$ ; として実行してみたときのグラフと lev 関数を使った場合のグラフを示す。

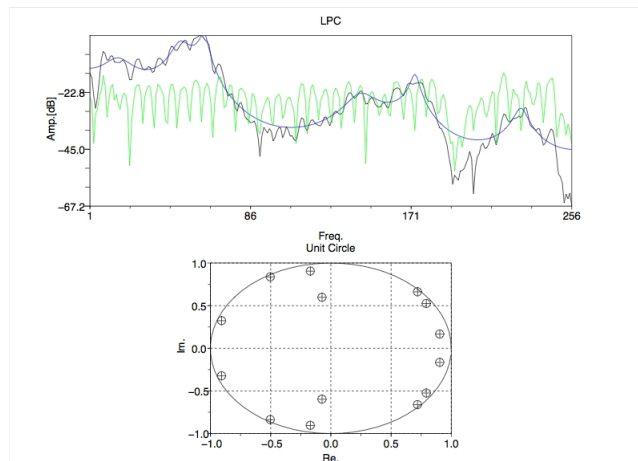


図 1: 実行結果 (lev()) を使ったグラフ

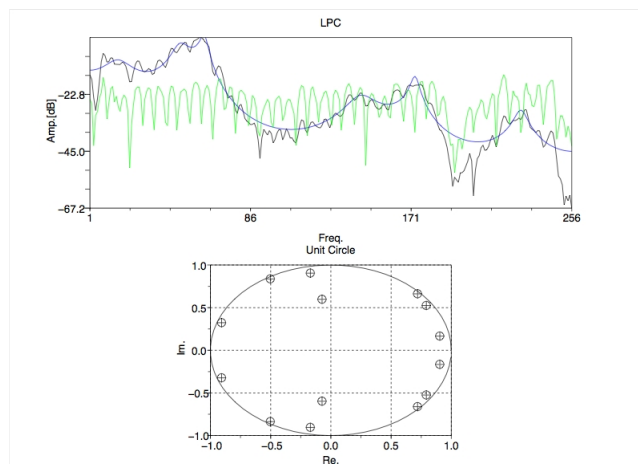


図 2: 実行結果 (lev()) を使っていないグラフ

グラフもほぼ同じ結果となっているので、うまく演算できている事が分かる。

### 2.3 課題 3

追加したソースを以下に示す。



## ソースコード

```
//lpc->lsp
lpc(1)=1;
for n=1:order
    lpc(n+1)=ar(n);
end
lpc_order=order;
fs=FS;
dft_size=4096;
lsp=zeros(1,lpc_order);
omega=zeros(1,lpc_order/2);
theta=zeros(1,lpc_order/2);
p=zeros(1,dft_size);
q=zeros(1,dft_size);
p(1)=lpc(1);
q(1)=lpc(1);
for n=2:lpc_order+1,
    p(n)=lpc(n)+lpc(lpc_order+3-n);
    q(n)=lpc(n)-lpc(lpc_order+3-n);
end
p(lpc_order+2)=lpc(1);
q(lpc_order+2)=-lpc(1);
P=abs(fft(p,-1));
m=1;
for k=2:dft_size/2,
    if P(k-1)>P(k) & P(k)<P(k+1)
        omega(m)=k-1;
        m=m+1;
    end
end
Q=abs(fft(q,-1));
m=1;
for k=2:dft_size/2,
    if Q(k-1)>Q(k) & Q(k)<Q(k+1)
        theta(m)=k-1;
        m=m+1;
    end
end
for m=1:lpc_order/2,
    lsp(2*m-1)=omega(m)*fs/dft_size;
    lsp(2*m)=theta(m)*fs/dft_size;
end
```

以下にその実行結果を示す。

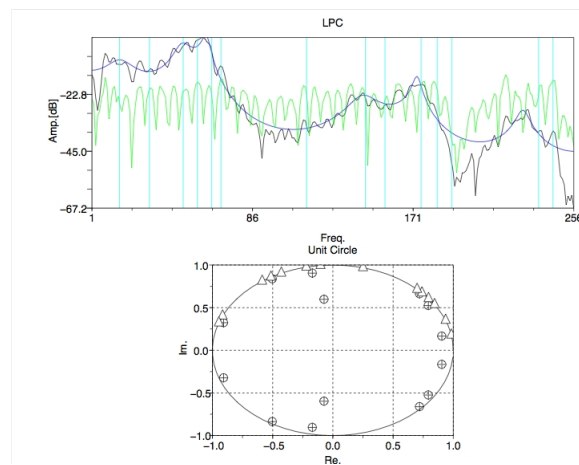


図 3: LSP

## 2.4 課題 4

ソースに追加した部分を以下に示す。

### ソースコード

```
//F と B のソートिंग
////まずは F から負の値を消す
j=0;
for i=1:length(F)
    if(F(i)>=0) then
        j=j+1;
        F2(j)=F(i);
        B2(j)=B(i);
        Fnum(j)=i;
    end;
end

////正の値になった F を小さい順に並べ替える
n=length(F2);
for i=1:n
    for i=1:n-1
        if(F2(i)>F2(i+1)) then
            W=F2(i);
            F2(i)=F2(i+1);
            F2(i+1)=W;
            W=B2(i);
            B2(i)=B2(i+1);
            B2(i+1)=W;
        end;
    end;
end;

for i=1:n
    printf("holmant %d F=%f B=%f \n",i,F2(i),B2(i));
end;
```

実行結果を以下に示す。

## 実行結果

```
-->exec('lpc3.sci')
-->//$B=i4|CM$N@_Dj(B
-->clear;
-->FS=10000;
-->fft_len=512;          // FFT$B$N%5%s%W%k?t(B
-->start=5000;          // $BJ,@00LCV(B
-->len=300;             // $BJ,@0D9(B
-->order=14;           // $BJ,@0<!?t(B
-->pre_emp=0.0;        // $B%W%j%(%s%U!%7%9(B
-->print(%io(2),'start= '); start = read(%io(1),1,1);

start=
-->3000

~省略~

holmant 1 F=289.061601 B=265.500449
holmant 2 F=929.838857 B=159.777313
holmant 3 F=1183.031928 B=71.447946
holmant 4 F=2693.409238 B=1617.560788
holmant 5 F=2795.169382 B=263.449821
holmant 6 F=3361.283275 B=78.653612
holmant 7 F=4455.512984 B=122.415711
```

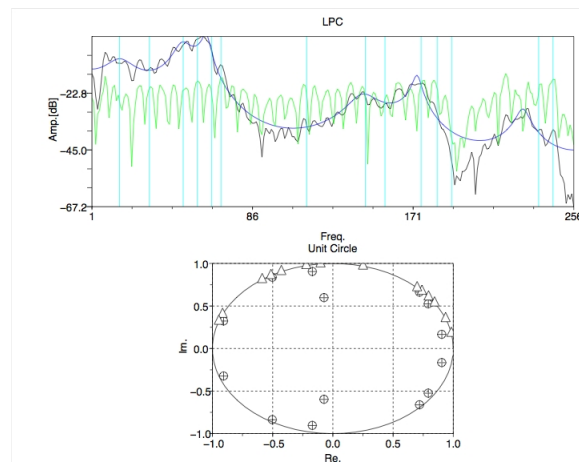


図 4: 出力図

## 2.5 課題 5

## 2.6 分析長の変更

最初に以下のパラメータで実行してみた。

## パラメータ

```
len=100;           // 分析長  
order=14;         // 分析次数
```

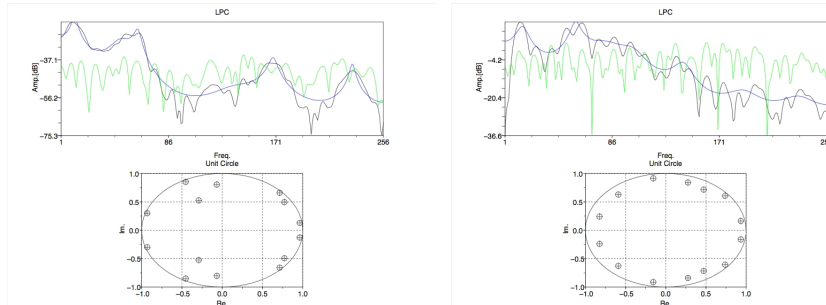


図 5: 男性音 : 分析長=100 分析次数=14 図 6: 女性音 : 分析長=100 分析次数=14

次に以下のパラメータで実行した。

## パラメータ

```
len=300;           // 分析長  
order=14;         // 分析次数
```

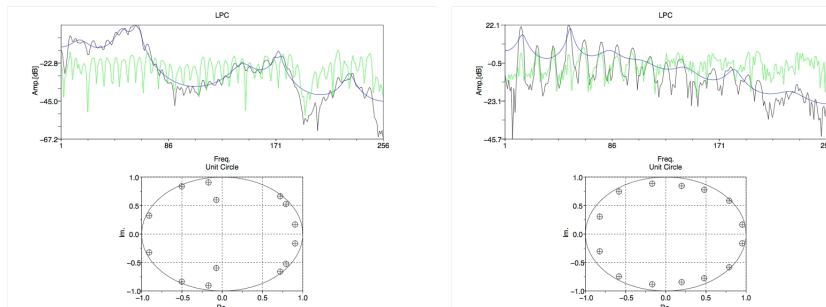


図 7: 男性音 : 分析長=300 分析次数=14 図 8: 女性音 : 分析長=300 分析次数=14

結果から分かるように、分析長の値を大きくすると、DFT スペクトルと残差予測スペクトルの振動が細かく表示されている。これは周波数分解能が高くなったからである。逆に分析長の値を小さくすると、時間分解能が高くなる。

## 2.7 分析次数の変更

最初に以下のパラメータで実行してみた。

### パラメータ

```
len=200;           // 分析長  
order=8;          // 分析次数
```

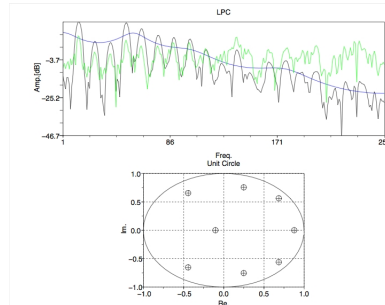
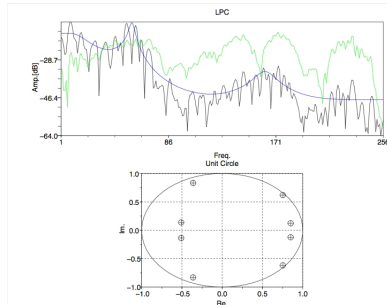


図 9: 男性音 : 分析長=200 分析次数=8 図 10: 女性音 : 分析長=200 分析次数=8

次に以下のパラメータで実行した。

### パラメータ

```
len=200;           // 分析長  
order=20;         // 分析次数
```

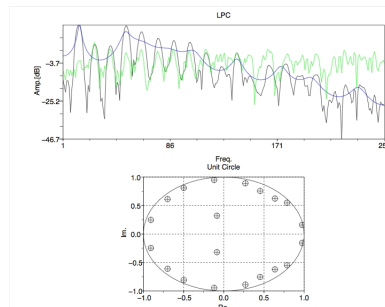
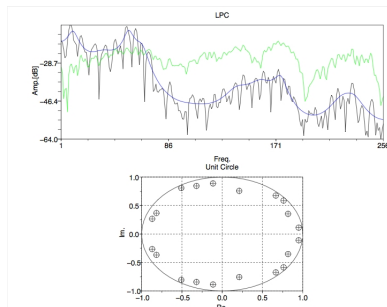


図 11: 男性音 : 分析長=200 分析次数=20 図 12: 女性音 : 分析長=200 分析次数=20

結果から分かるように、分析次数を大きくすると極が増加している。また、男性音と女性音を比べると女性音の方が DFT スペクトル、LPC 予測残差スペクトルの値が大きいことが分かる。