

平成27年度

デジタル信号処理課題4

—適応フィルタ—

レポート提出日：12月21日

学籍番号：1315193T

氏 名：薛 宇航

1 原理

1.1 音響システム

空気中の音の伝搬は、大気圧の変化が縦波として伝搬するものであり、この伝搬を線形であるとみなす。またスピーカーからマイクロホンまで n 点間の音の伝達特性は、物体の移動や気温変化などに伴い変化する量であるので、厳密には線形時変システムと考えられる。しかし、短い時間範囲であれば、その変化は十分小さいと考え、時不変システムとしてみなす。このとき、スピーカーからの信号を $x[n]$ 、スピーカーとマイクロホンの 2 点間の音の伝達特性を $h[k]$ とすると、マイクロホンにて観測される信号 $d[n]$ は、二つの信号 $x[n]$ と $h[k]$ の畳み込みによって、以下の式 (1) ように得られる。

$$d[n] \approx \sum_k h[k]x[n-k] \quad (1)$$

ここで上式では、人の声は存在せず、マイクロホンにはスピーカーの音のみ観測されるとしている。もし人の声も存在する場合は、マイクロホンにて観測される信号 $d[n]$ は、スピーカー音と、音声の加算によって表現される。これは下の式 (2) のように示す。

$$d[n] \approx \sum_k h[k]x[n-k] + s'[n] \quad (2)$$

ここで、 $s'[n]$ は、音声 $s[n]$ に伝達特性を畳み込んだ信号とする、式 (2) のフィルタ係数 $h[k]$ がわかれば、人の声に重畳したスピーカーからの音楽などの「音」を消して、人の声だけを聞くことができる。実際には手間のかかる作業隣る。そこで、ここでは LMS (Least Mean Square) アルゴリズムを用いてフィルタ係数 $h[k]$ を求める。

1.2 正規化 LMS アルゴリズムによる音楽消去

LMS アルゴリズムでは、逐次的にフィルタ係数を求める。時刻 n でのフィルタ係数を $h_n[k]$ とすると、式 (1) の左辺と右辺の誤差は

$$e[n] = d[n] - \sum_{k=0}^{K-1} h_n[k]x[n-k] \quad (3)$$

フィルタ係数の更新する式は以下になる。

$$h_{n+1}[k] = h_n[k] + \frac{\alpha}{\|x[n]\|^2} \times e[n] \times x[n-k] \quad (4)$$

2 課題 1

2.1 課題内容

web page にサンプルデータを置いてある。サンプルデータを使い (学番の下一桁が奇数のため、サンプル 1 となる)、サンプルデータの最初の 20 秒間、NLMS を用いてフィルタ係数を推定せよ。適応フィルタの初期値は 0、 $\alpha=0.02$ とせよ

適応フィルタの次数 K を 10, 20, \dots 50, \dots とした場合、誤差値がどのように変わっているかを述べよ。

2.2 解答内容

適正フィルタの次数 K が大きくなるにつれて、誤差 e の振動が小さくなっていく。それぞれ次数 K を 10、20、50 にした時、誤差 e はそれぞれ以下の図 1 に示すようになる。

なお、今の課題で使用するプログラムは以下のようになる。

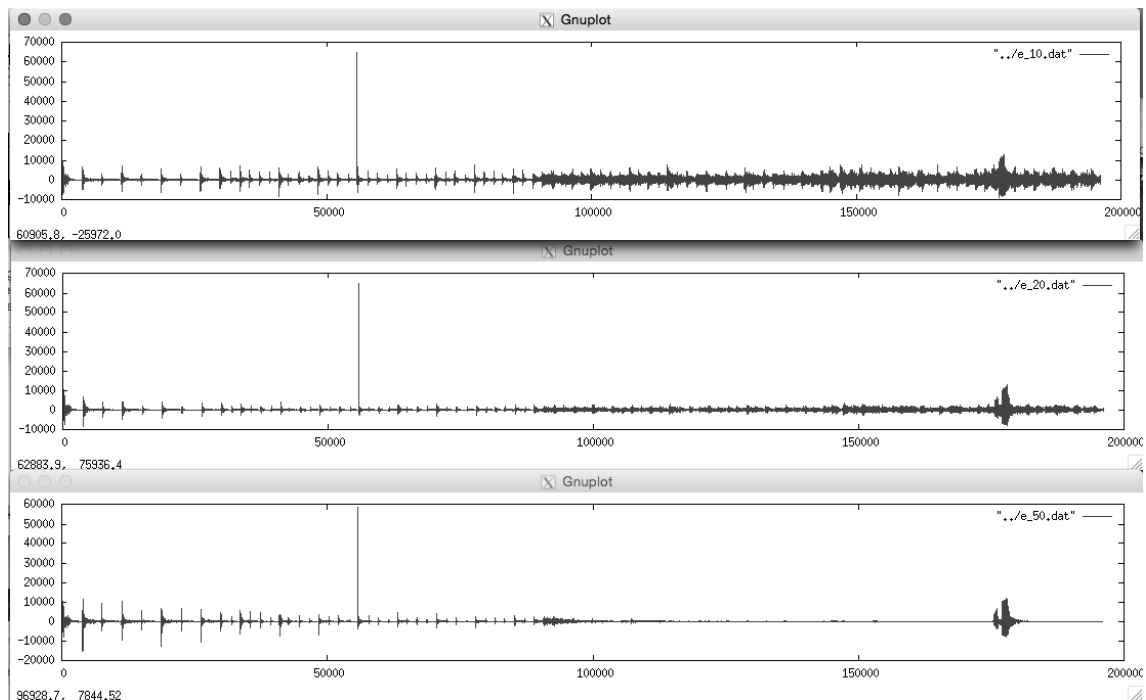


図 1 次数の変化によって誤差値の変わり方

ソースコード 1 課題 1 で使用するプログラム

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

#define Pi 3.14159265358979
#define alpha 0.01
#define K 50 //changed everytime to ensure the output
#define MAX 300000

int main(){
    char d_filename[128]="observed_song1.raw";//short
    char x_filename[128]="song1.raw";//short
    char e_filename[128]="e.raw";

    int k,n,size_d ,size_x;

    short *d,*x,s[MAX],tmp;
    FILE *file_input;
    FILE *file_output;

    double e[160000];
    double h_n[K]={0},h_n1[K]={0},norm;
    //double h;

    //operation of d
    if((file_input=fopen(d_filename,"rb"))==NULL){
        fprintf(stderr,"Cannot read %s.\n",d_filename);exit(-1);
    }
    size_d=fread(s,sizeof(short),MAX,file_input);
    d=(short *)calloc(size_d,sizeof(float));
    fclose(file_input);
    if((file_input=fopen(d_filename,"rb"))==NULL){
        fprintf(stderr,"Cannot read %s.\n",d_filename);exit(-1);
    }
    fread(d,sizeof(short),size_d,file_input);
    fclose(file_input);
```

```

//operation of x
if((file_input=fopen(x_filename,"rb"))==NULL){
    fprintf(stderr,"Cannot read %s.\n",x_filename);exit(-1);
}
size_x=fread(s,sizeof(short),MAX,file_input);
x=(short *)calloc(size_x,sizeof(short));
fclose(file_input);
if((file_input=fopen(x_filename,"rb"))==NULL){
    fprintf(stderr,"Cannot read %s.\n",x_filename);exit(-1);
}
fread(x,sizeof(short),size_x,file_input);
fclose(file_input);

for(n=0;n<size_d;n++){

    e[n]=(double)d[n];
    // at this time,n=n+1    update the h_n1 to h_n
    for(k=0;k<K;k++){
        h_n[k]=h_n1[k];
        e[n]-=h_n[k]*x[n-k];
    }
    norm=0.0;//initial norm
    for(k=0;k<K;k++){
        if(n>k){
            norm+=x[n-k]*x[n-k];
        }
    }

    for(k=0;k<K;k++){
        h_n1[k]=h_n[k];
        if(norm==0){
            norm = 1;
        }
        h_n1[k]=h_n[k]+alpha*e[n]*x[n-k]/norm;
    }
}

if((file_output = fopen(e_filename,"wb")) == NULL){
    fprintf(stderr, "Cannot write %s\n", e_filename);    exit(-1);
}
for(n=0;n<size_d;n++){
    //printf("%lf\n",e[n]);    //use gnuplot to plot error
    tmp = (short)e[n];
    fwrite(&tmp, sizeof(short), 1, file_output);
}

fclose(file_output);

free(d);
free(x);
return 0;
}

```

3 課題 2

3.1 課題内容

2 秒、10 秒、20 秒の時に推定されたフィルタ係数、及び正解の $h[k]$ を表示して、時刻とともに正解の $h[k]$ に近づいていることを確認せよ。

3.2 解答内容

$K=50$ の時、2 秒、10 秒、20 秒の時に推定されたフィルタ係数及び正解の $h[k]$ を同じ図表にプロットして、以下の図 2 のようになります。本レポートは `tex` によって作成されたため、引用する画像を全て `eps`

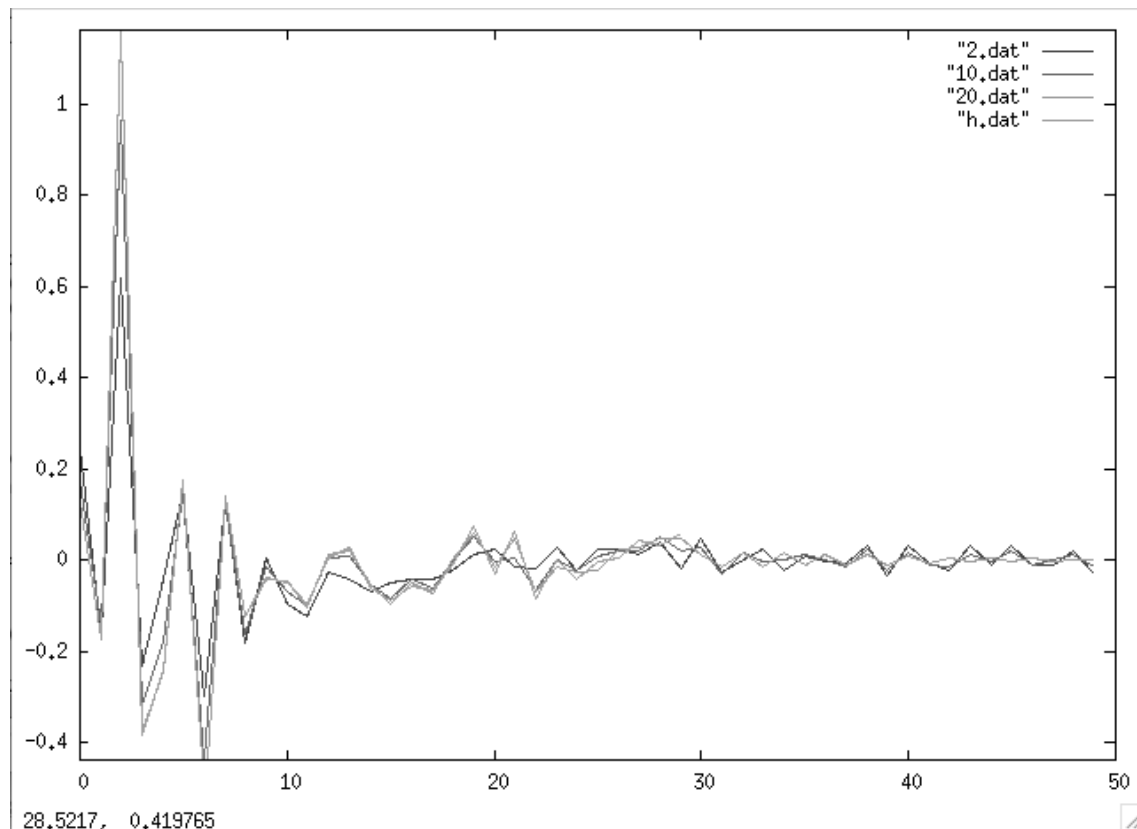


図 2 $K=50$ の推定フィルタ係数と正解の比較

ファイルの白黒になります。図からわかりにくいと思いますが、時間とともに正解の $h[k]$ に近づくことが確かである。なお、本レポートと共に、カラー画像も一緒に添付し、指定メールアドレスに送る。

ソースコード 2 課題 2 で使用するプログラム

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

#define Pi 3.14159265358979
#define alpha 0.01
#define K 50 //changed everytime to ensure the output
#define MAX 300000

int main(){
    char d_filename[128]="observed_song1.raw";//short
    char x_filename[128]="song1.raw";//short
    char h_filename[128]="imp.dat";//float
    char e_filename[128]="e.raw";
```

```

int k,n,size_h , size_d , size_x;

short *d,*x,s [MAX] ,tmp;
float *h,f [MAX];
FILE *file_input ;
FILE *file_output;

double e[160000];
double h_n [K]={0},h_n1 [K]={0},norm;
//double h;

//operation of d
if(( file_input=fopen(d_filename , "rb")==NULL){
    fprintf(stderr , "Cannot_read_%s.\n" , d_filename ); exit (-1);
}
size_d=fread(s , sizeof(short) ,MAX, file_input );
d=(short *) calloc (size_d , sizeof(float ));
fclose (file_input );
if(( file_input=fopen(d_filename , "rb")==NULL){
    fprintf(stderr , "Cannot_read_%s.\n" , d_filename ); exit (-1);
}
fread(d , sizeof(short) , size_d , file_input );
fclose (file_input );

//operation of x
if(( file_input=fopen(x_filename , "rb")==NULL){
    fprintf(stderr , "Cannot_read_%s.\n" , x_filename ); exit (-1);
}
size_x=fread(s , sizeof(short) ,MAX, file_input );
x=(short *) calloc (size_x , sizeof(short ));
fclose (file_input );
if(( file_input=fopen(x_filename , "rb")==NULL){
    fprintf(stderr , "Cannot_read_%s.\n" , x_filename ); exit (-1);
}
fread(x , sizeof(short) , size_x , file_input );
fclose (file_input );

//operation of h
if(( file_input=fopen(h_filename , "rb")==NULL){
    fprintf(stderr , "Cannot_read_%s.\n" , h_filename ); exit (-1);
}
size_h=fread(f , sizeof(float) ,MAX, file_input );
h=(float *) calloc (size_h , sizeof(float ));
fclose (file_input );
if(( file_input=fopen(h_filename , "rb")==NULL){
    fprintf(stderr , "Cannot_read_%s.\n" , h_filename ); exit (-1);
}
fread(h , sizeof(float) , size_h , file_input );
fclose (file_input );

// for (k=0;k<size_h ;k++){
//   printf("%lf\n" ,h[k] );
// }

for (n=0;n<size_d ;n++){
    if (n==8000*10){
        for (k=0;k<K;k++){
            printf ("%lf\n" ,h_n [k] );
        }
    }
}

```

```

e[n]=(double)d[n];
// at this time, n=n+1    update the h_n1 to h_n
for (k=0;k<K;k++){
    h_n[k]=h_n1[k];
    e[n]=h_n[k]*x[n-k];

}
norm=0.0;// initial norm
for (k=0;k<K;k++){
    if (n>k){
        norm+=x[n-k]*x[n-k];
    }
}

for (k=0;k<K;k++){
    h_n1[k]=h_n[k];
    if (norm==0){
        norm = 1;
    }
    h_n1[k]=h_n[k]+alpha*e[n]*x[n-k]/norm;
}
}

if ((file_output = fopen(e_filename, "wb")) == NULL){
    fprintf(stderr, "Cannot write %s\n", e_filename);    exit(-1);
}
for (n=0;n<size_d;n++){
    //printf("%lf\n", e[n]);    //use gnuplot to plot error
    tmp = (short)e[n];
    fwrite(&tmp, sizeof(short), 1, file_output);
}

fclose(file_output);

free(d);
free(x);
free(h);
return 0;
}

```

4 課題3

4.1 課題内容

21 秒から 23 秒の間に人が発話を行っている。この時間区間に置けるマイクロホンの観測信号 $d[n]$ 、スピーカーからの音楽を除去した信号 $s'[n]$ 、クリーン音声信号 $s[n]$ 、これら 3 つの信号の波形及びスペクトrogramを表示せよ。

4.2 解答内容

web page に置かれてる音声信号ファイルを読み込み、21 秒から 23 秒の間だけ操作して、 $d[n]$ 、 $s[n]$ 、及び $s'[n]$ の波形及びスペクトrogramをそれぞれ図 3、図 4、図 5 に示す。

なお、本課題に使用するプログラムは以下ようになります。

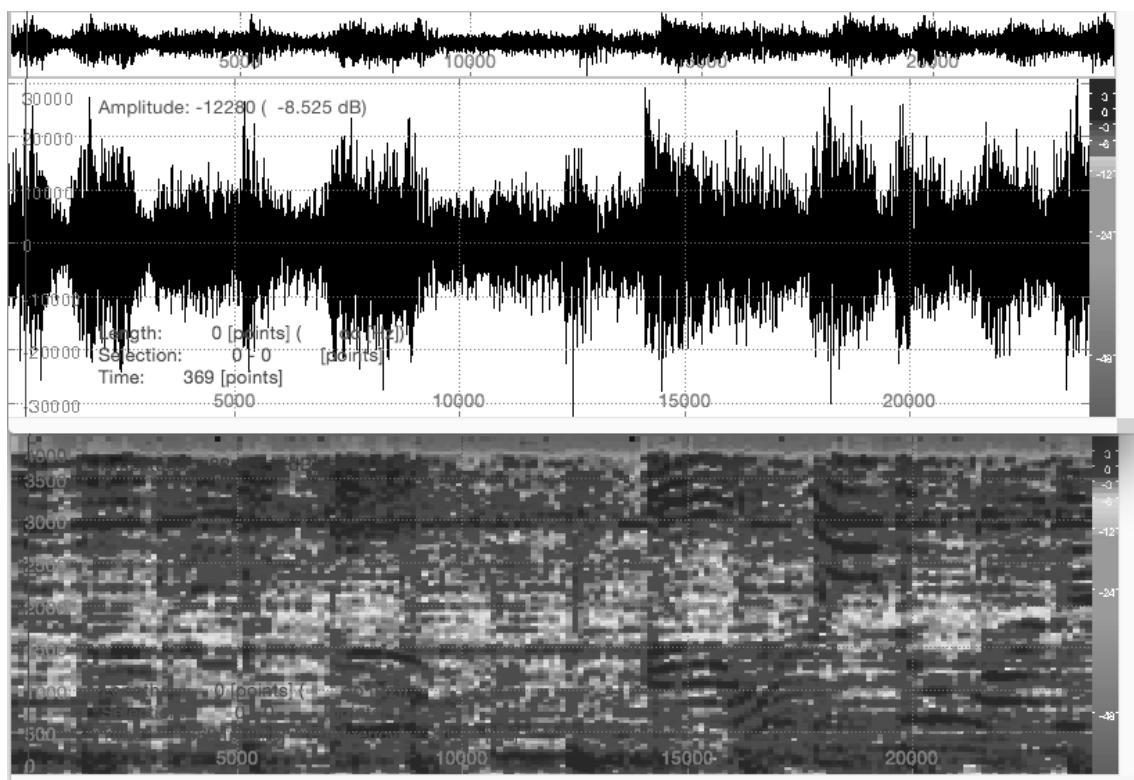


図 3 $d[n]$ の波形及びスペクトル (20s 23s)

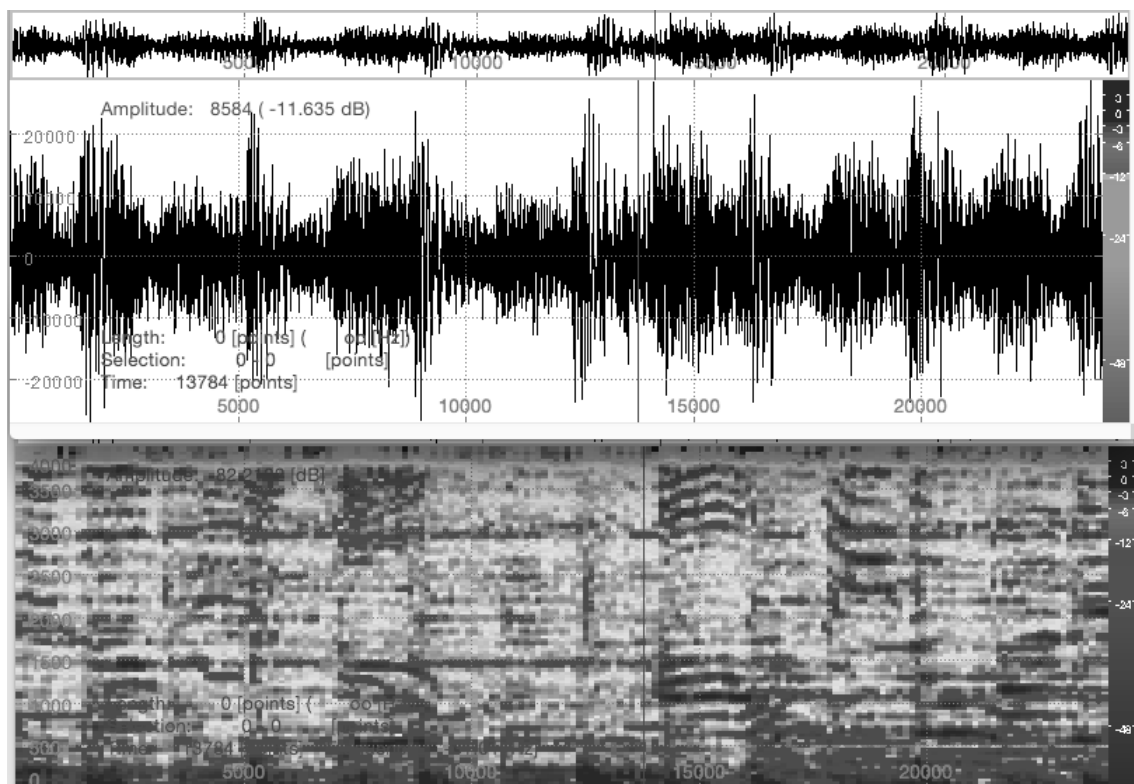


図 4 $s[n]$ の波形及びスペクトル (20s 23s)

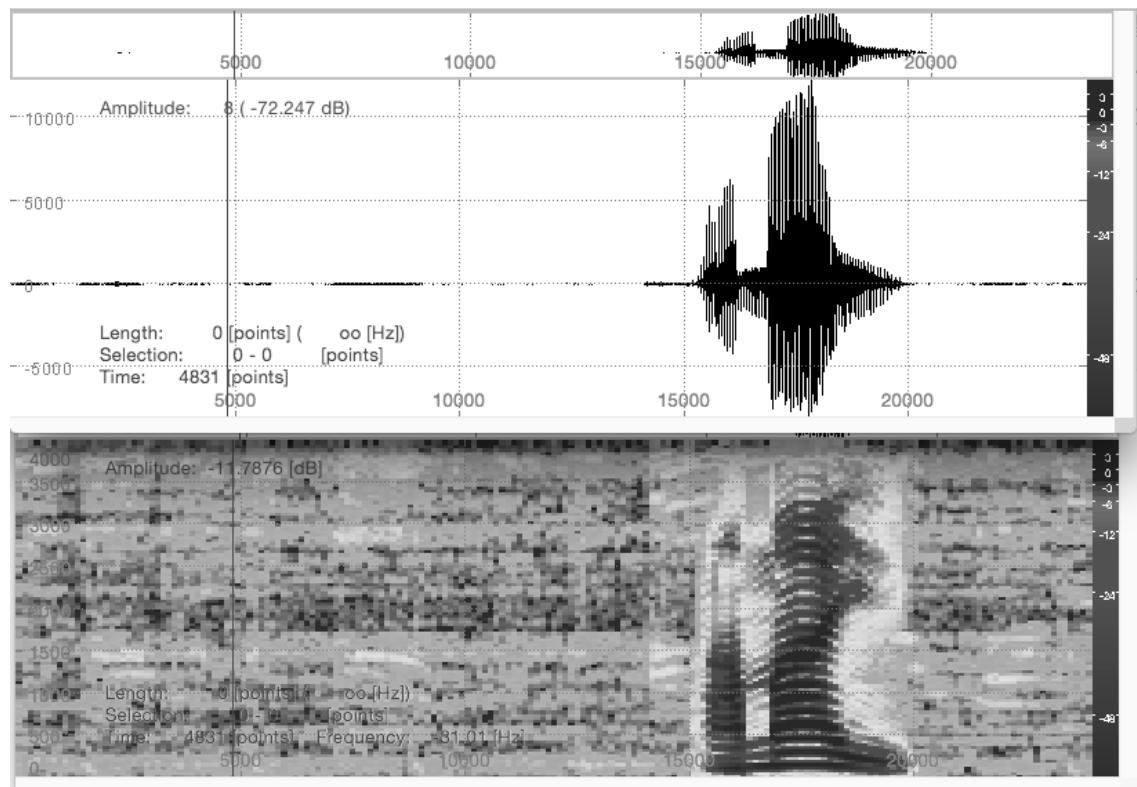


図 5 $s'[n]$ の波形及びスペクトル (20s 23s)

ソースコード 3 課題 3 で使用するプログラム

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

#define Pi 3.14159265358979
#define alpha 0.01
#define K 50 //changed everytime to ensure the output
#define MAX 300000

int main(){
    char d_filename[128]="observed_song1.raw";//short
    char x_filename[128]="song1.raw";//short
    char e_filename[128]="e.raw";
    char d_3[128]="d.raw";
    char s_3[128]="s.raw";
    char s_s_3[128]="s_s.raw";

    int k,n,size_d,size_x;

    short *d,*x,s[MAX],tmp;
    FILE *file_input;
    FILE *file_outputd;
    FILE *file_outputs;
    FILE *file_outputs_s;

    double e[160000];
    double h_n[K]={0},h_n1[K]={0},norm;
    //double h;

    //operation of d
    if((file_input=fopen(d_filename,"rb"))==NULL){
        fprintf(stderr,"Cannot read %s.\n",d_filename);exit(-1);
    }
    size_d=fread(s,sizeof(short),MAX,file_input);
```

```

d=(short *)calloc(size_d ,sizeof(float));
fclose(file_input);
if((file_input=fopen(d_filename , "rb"))==NULL){
    fprintf(stderr , "Cannot read %s\n" , d_filename); exit(-1);
}
fread(d, sizeof(short), size_d , file_input);
fclose(file_input);

//operation of x
if((file_input=fopen(x_filename , "rb"))==NULL){
    fprintf(stderr , "Cannot read %s\n" , x_filename); exit(-1);
}
size_x=fread(s, sizeof(short), MAX, file_input);
x=(short *)calloc(size_x , sizeof(short));
fclose(file_input);
if((file_input=fopen(x_filename , "rb"))==NULL){
    fprintf(stderr , "Cannot read %s\n" , x_filename); exit(-1);
}
fread(x, sizeof(short), size_x , file_input);
fclose(file_input);

for (n=0;n<size_d ;n++){

    e[n]=(double)d[n];
    // at this time, n=n+1    update the h_n1 to h_n
    for (k=0;k<K;k++){
        h_n[k]=h_n1[k];
        e[n]-=h_n[k]*x[n-k];
    }
    norm=0.0; // initial norm
    for (k=0;k<K;k++){
        if (n>k){
            norm+=x[n-k]*x[n-k];
        }
    }

    for (k=0;k<K;k++){
        h_n1[k]=h_n[k];
        if (norm==0){
            norm = 1;
        }
        h_n1[k]=h_n[k]+alpha*e[n]*x[n-k]/norm;
    }
}

if((file_outputd = fopen(d_3, "wb")) == NULL){
    fprintf(stderr , "Cannot write %s\n" , e_filename);    exit(-1);
}
if((file_outputs = fopen(s_3, "wb")) == NULL){
    fprintf(stderr , "Cannot write %s\n" , e_filename);    exit(-1);
}
if((file_outputs_s = fopen(s_s_3, "wb")) == NULL){
    fprintf(stderr , "Cannot write %s\n" , e_filename);    exit(-1);
}

for (n=20*8000;n<23*8000;n++){
    tmp = (short)d[n];
    fwrite(&tmp, sizeof(short), 1, file_outputd);
    tmp = (short)s[n];
    fwrite(&tmp, sizeof(short), 1, file_outputs);
    for (k=0;k<K;k++){
        d[n]-=h_n[k]*x[n-k];
    }
    tmp = (short)d[n];

```

```
        fwrite(&tmp, sizeof(short), 1, file_outputs_s);
    }

    fclose(file_outputd);
    fclose(file_outputs);
    fclose(file_outputs_s);

    free(d);
    free(x);
    return 0;
}
```

参考文献

- [1] 実験内容について: 情報知能工学実験指導書 (実験 II)