```
%Problem 9.1
% encode text string as T?spaced 4?PAM sequence
str='01234 I wish I were an Oscar Meyer wiener 56789 ';
m=letters2pam(str);N=length(m); % 4?l e v e l s ignal of length N
% zero pad T?spaced symbol sequence to c r eat e upsampled
% T/M?spaced sequence of s cal ed T?spaced pul s e s (T=1)
M=100; % oversampling f ac tor
mup=zeros(1,N*M) ; % Hamming pul s e f i l t e r with
mup(1:M:N*M)=m; % T/M?spaced impulse response
p=hamming(M); % bl i p pulse of width M
x=filter(p , 1 ,mup) ; % convolve pul se shape with data
figure(1),plotspec(x,1/M) % baseband AM modulation
t=1/M:1/M:length(x)/M; % T/M?spaced time vector
fc=[50 30 3 1 0.5]; % c a r r i e r f requency
for i=1:5
c=cos(2*pi* fc(i) *t ) ; % carrier
r=c.*x; % modulate message with c a r r i e r
% AM demodulation of r e c e ived s i gnal sequence r
c2=cos(2* pi* fc(i) *t ) ; % synchronized cos ine f or mixing
x2=r .* c2 ; % demod r e c e ived s i gnal
fl =50; fbe=[0 0.1 0.2 1 ] ; % LPF parameters
damps=[1 1 0 0 ] ;
b=firpm( fl , fbe , damps ) ; % cr eate LPF impulse response
x3=2* filter (b , 1 , x2 ) ; % LPF and s c a l e s i gnal
% ext r ac t upsampled pul s e s us ing c o r r e l a t i o n
 implemented
% as a convolving f i l t e r ; f i l t e r with pulse and normal ize
y=filter (fliplr(p)/(pow(p)*M),1,x3 );
% set delay to first symbol?sample and increment by M
z=y(0.5*fl+M:M:N*M) ; % downsample to symbol r at e
figure(i+1),plot([1:length(z)],z,'.') % pl o t so f t de c i s i ons
% de c i s i on device and symbol matching performance assessment
mprime=quantalph(z,[-3,-1,1,3])'; % quantize alphabet
cvar=(mprime-z)*(mprime-z)'/length(mprime) , % cluster variance
lmp=length (mprime ) ;
pererr=100*sum(abs(sign(mprime-m(1:lmp))))/lmp, % symbol er r or
% decode de c i s i on device output to text s t r ing
reconstructedmessage=pam2letters(mprime)
end

%The carrier frequencies worked when they were greater than the
 greatest
%frequency component, which was 1. for this reason 0.5 did not work.


%Problem 9.2
% encode text string as T?spaced 4?PAM sequence
str='01234 I wish I were an Oscar Meyer wiener 56789 ';
m=letters2pam(str);N=length(m); % 4?l e v e l s ignal of length N
% zero pad T?spaced symbol sequence to c r eat e upsampled
% T/M?spaced sequence of s cal ed T?spaced pul s e s (T=1)
M=[1000 25 10]; % oversampling f ac tor
```

```
for i2=1:3
mup=zeros(1,N*M(i2)) ; % Hamming pulse filter with
mup(1:M(i2):N*M(i2))=m; % T/M?spaced impulse response
p=hamming(M(i2)); % blip pulse of width M
x=filter(p , 1 ,mup) ; % convolve pulse shape with data
figure(7),plotspec(x,1/M(i2)) % baseband AM modulation
t=1/M(i2):1/M(i2):length(x)/M(i2); % T/M?spaced time vector
fc=20; % carrier frequency
c=cos(2*pi* fc *t ) ; % carrier
r=c.*x; % modulate message with carrier
% AM demodulation of received signal sequence r
c2=cos(2* pi* fc *t ) ; % synchronized cosine for mixing
x2=r .* c2 ; % demod received signal
fl =50; fbe=[0 0.1 0.2 1 ] ; % LPF parameters
damps=[1 1 0 0 ] ;
b=firpm( fl , fbe , damps ) ; % create LPF impulse response
x3=2* filter (b , 1 , x2 ) ; % LPF and scale signal
% extract upsampled pulses using correlation
 implemented
% as a convolving filter; filter with pulse and normalize
y=filter (fliplr(p)/(pow(p)*M(i2)),1,x3 );
% set delay to first symbol?sample and increment by M
z=y(0.5*fl+M(i2):M(i2):N*M(i2)) ; % downsample to symbol rate
figure(7+i2),plot([1:length(z)],z,'.') % plot soft decisions
% decision device and symbol matching performance assessment
mprime=quantalph(z,[-3,-1,1,3])'; % quantize alphabet
cvar=(mprime-z)*(mprime-z)'/length(mprime) , % cluster variance
lmp=length (mprime ) ;
pererr=100*sum(abs(sign(mprime-m(1:lmp))))/lmp, % symbol error
% decode decision device output to text string
reconstructedmessage=pam2letters(mprime)
end

%When the oversampling frequency is too low (i.e. M = 10), the message
%fails, only capturing some of the message and random letters for the
 other
%parts


%Problem 9.3
%If we remove the LPF at the beginning of the receiver is removed, the
 high
%frequency noise will no longer be filtered out, making the signal
 harder
%to filter correctly. If there are other users, these will interfere
 with
%the signal and make correct transmission problematic.


% encode text string as T?spaced 4?PAM sequence
str='01234 I wish I were an Oscar Meyer wiener 56789 ';
m=letters2pam(str);N=length(m); % 4?level signal of length N
% zero pad T?spaced symbol sequence to create upsampled
% T/M?spaced sequence of scaled T?spaced pulses (T=1)
```

```
M=100; % oversampling factor
mup=zeros(1,N*M) ; % Hamming pulse filter with
mup(1:M:N*M)=m; % T/M?spaced impulse response
p=hamming(M); % blip pulse of width M
x=filter(p , 1 ,mup) ; % convolve pulse shape with data
figure(11),plotspec(x,1/M) % baseband AM modulation
t=1/M:1/M:length(x)/M; % T/M?spaced time vector
fc=20; % carrier frequency
fcother = 30;                               %other user
c=cos(2*pi* fc *t ) ; % carrier
cother = cos(2*pi*fcother*t);                   %other user
r=c.*x; % modulate message with carrier
rother = cother.*x;                              %other user
% AM demodulation of received signal sequence r
c2=cos(2* pi* fc *t ) ; % synchronized cosine for mixing
x2=r .* c2 ; % demod received signal
fl =50; fbe=[0 0.1 0.2 1 ] ; % LPF parameters
damps=[1 1 0 0 ] ;
%b=firpm( fl , fbe , damps ) ; % create LPF impulse response
%x3=2* filter (b , 1 , x2 ) ; % LPF and scale signal
% extract upsampled pulses using correlation
 implemented
% as a convolving filter ; filter with pulse and normalize
y=filter (fliplr(p)/(pow(p)*M),1,x2 );
% set delay to first symbol?sample and increment by M
z=y(0.5*fl+M:M:N*M) ; % downsample to symbol rate
figure(12),plot([1:length(z)],z,'.') % plot soft decisions
% decision device and symbol matching performance assessment
mprime=quantalph(z,[-3,-1,1,3])'; % quantize alphabet
cvar=(mprime-z)*(mprime-z)'/length(mprime) , % cluster variance
lmp=length (mprime ) ;
pererr=100*sum(abs(sign(mprime-m(1:lmp))))/lmp, % symbol error
% decode decision device output to text string
reconstructedmessage=pam2letters(mprime)

%Problem 9.4
%The lowest cutoff frequency is at 0.015, and the highest cutoff
 frequency
%is at 0.999998

% encode text string as T?spaced 4?PAM sequence
str='01234 I wish I were an Oscar Meyer wiener 56789 ';
m=letters2pam(str);N=length(m); % 4?level signal of length N
% zero pad T?spaced symbol sequence to create upsampled
% T/M?spaced sequence of scaled T?spaced pulses (T=1)
M=100; % oversampling factor
mup=zeros(1,N*M) ; % Hamming pulse filter with
mup(1:M:N*M)=m; % T/M?spaced impulse response
p=hamming(M); % blip pulse of width M
x=filter(p , 1 ,mup) ; % convolve pulse shape with data
figure(13),plotspec(x,1/M) % baseband AM modulation
t=1/M:1/M:length(x)/M; % T/M?spaced time vector
fc=20; % carrier frequency
c=cos(2*pi* fc *t ) ; % carrier
```

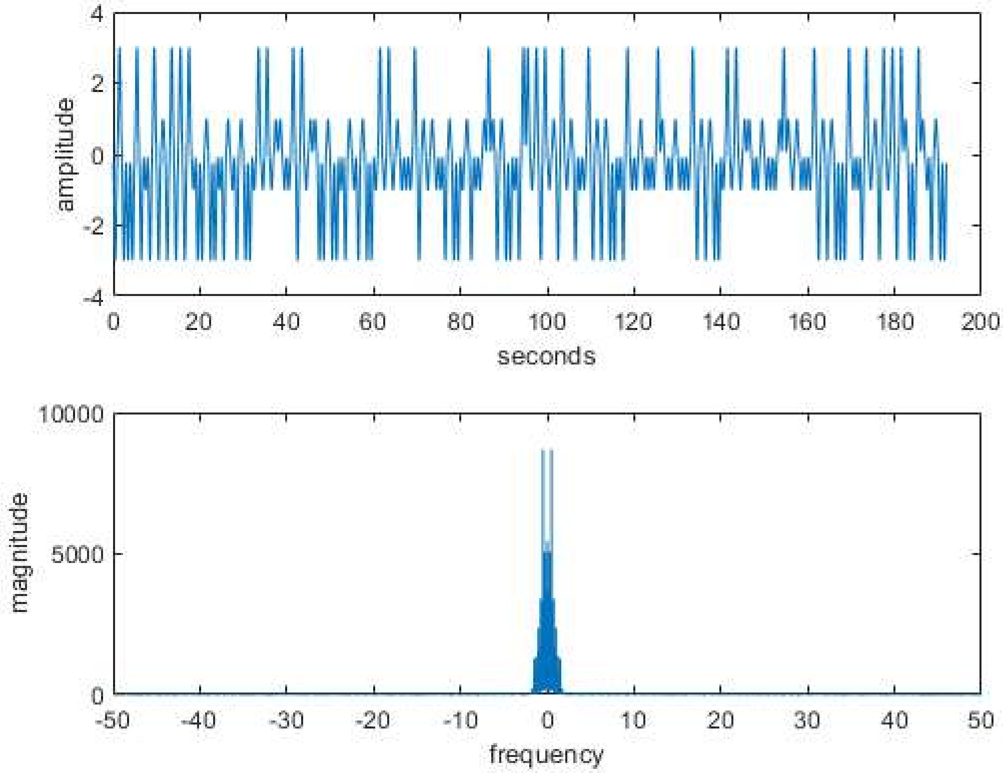
```
r=c.*x; % modulate message with carrier
% AM demodulation of received signal sequence r
c2=cos(2* pi* fc *t ) ; % synchronized cosine for mixing
x2=r .* c2 ; % demod received signal
fl =50; fbe=[0 0.014 0.016 1 ] ; % LPF parameters (at lowest cutoff)
damps=[1 1 0 0 ] ;
b=firpm( fl , fbe , damps ) ; % create LPF impulse response
x3=2* filter (b , 1 , x2 ) ; % LPF and scales signal
% extract upsampled pulses using correlation
 implemented
% as a convolving filter; filter with pulse and normalize
y=filter (fliplr(p)/(pow(p)*M),1,x3 );
% set delay to first symbol?sample and increment by M
z=y(0.5*fl+M:M:N*M) ; % downsample to symbol rate
figure(14),plot([1:length(z)],z,'.') % plot soft decisions
% decision device and symbol matching performance assessment
mprime=quantalph(z,[-3,-1,1,3])'; % quantize alphabet
cvar=(mprime-z)*(mprime-z)'/length(mprime) , % cluster variance
lmp=length (mprime ) ;
pererr=100*sum(abs(sign(mprime-m(1:lmp))))/lmp, % symbol error
% decode decision device output to text string
reconstructedmessage=pam2letters(mprime)

% encode text string as T?spaced 4?PAM sequence
str='01234 I wish I were an Oscar Meyer wiener 56789 ';
m=letters2pam(str);N=length(m); % 4?level signal of length N
% zero pad T?spaced symbol sequence to create upsampled
% T/M?spaced sequence of scaled T?spaced pulses (T=1)
M=100; % oversampling factor
mup=zeros(1,N*M) ; % Hamming pulse filter with
mup(1:M:N*M)=m; % T/M?spaced impulse response
p=hamming(M); % blip pulse of width M
x=filter(p , 1 ,mup) ; % convolve pulse shape with data
figure(15),plotspec(x,1/M) % baseband AM modulation
t=1/M:1/M:length(x)/M; % T/M?spaced time vector
fc=20; % carrier frequency
c=cos(2*pi* fc *t ) ; % carrier
r=c.*x; % modulate message with carrier
% AM demodulation of received signal sequence r
c2=cos(2* pi* fc *t ) ; % synchronized cosine for mixing
x2=r .* c2 ; % demod received signal
fl =50; fbe=[0 0.9998 0.9999 1 ] ; % LPF parameters (high cutoff)
damps=[1 1 0 0 ] ;
b=firpm( fl , fbe , damps ) ; % create LPF impulse response
x3=2* filter (b , 1 , x2 ) ; % LPF and scales signal
% extract upsampled pulses using correlation
 implemented
% as a convolving filter; filter with pulse and normalize
y=filter (fliplr(p)/(pow(p)*M),1,x3 );
% set delay to first symbol?sample and increment by M
z=y(0.5*fl+M:M:N*M) ; % downsample to symbol rate
figure(16),plot([1:length(z)],z,'.') % plot soft decisions
% decision device and symbol matching performance assessment
mprime=quantalph(z,[-3,-1,1,3])'; % quantize alphabet
```

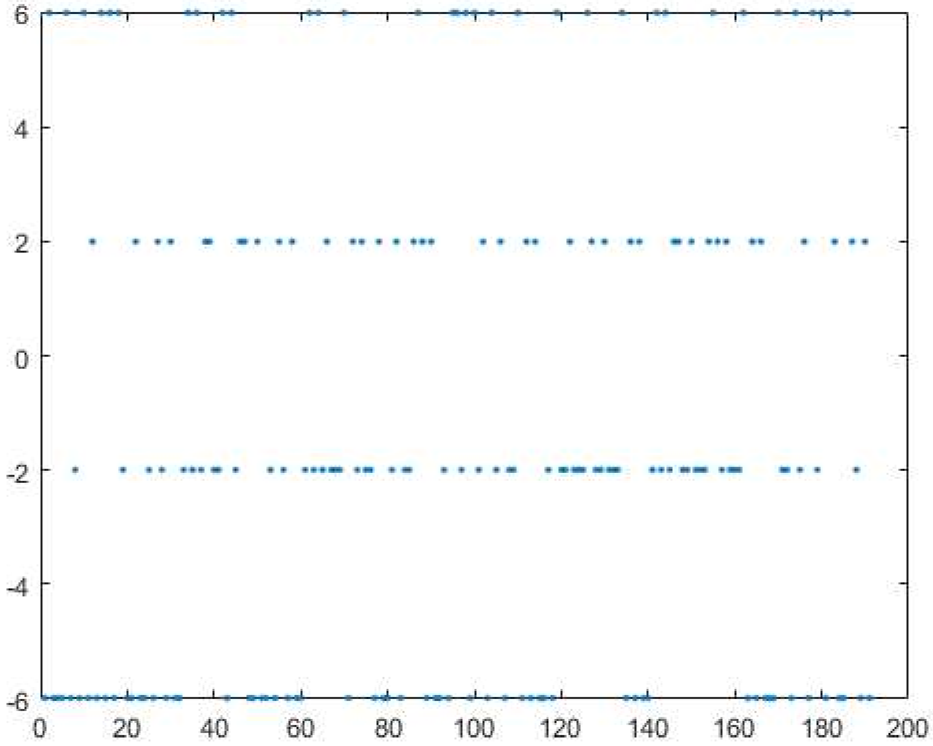
```
cvar=(mprime-z)*(mprime-z)'/length(mprime) , % cluster variance
lmp=length (mprime ) ;
pererr=100*sum(abs(sign(mprime-m(1:lmp))))/lmp, % symbol error
% decode decision device output to text string
reconstructedmessage=pam2letters(mprime)

%Problem 9.5
% encode text string as T?spaced 4?PAM sequence
str='01234 I wish I were an Oscar Meyer wiener 56789 ';
m=letters2pam(str);N=length(m); % 4?level signal of length N
% zero pad T?spaced symbol sequence to create upsampled
% T/M?spaced sequence of scaled T?spaced pulses (T=1)
M=100; % oversampling factor
mup=zeros(1,N*M) ; % Hamming pulse filter with
mup(1:M:N*M)=m; % T/M?spaced impulse response
p=hamming(M); % blip pulse of width M
x=filter(p , 1 ,mup) ; % convolve pulse shape with data
figure(17),plotspec(x,1/M) % baseband AM modulation
t=1/M:1/M:length(x)/M; % T/M?spaced time vector
fc=20; % carrier frequency
c=cos(2*pi* fc *t ) ; % carrier
r=c.*x; % modulate message with carrier
% AM demodulation of received signal sequence r
c2=cos(2* pi* fc *t ) ; % synchronized cosine for mixing
x2=r .* c2 ; % demod received signal
fl =4; fbe=[0 0.1 0.2 1 ] ; % LPF parameters
damps=[1 1 0 0 ] ;
b=firpm( fl , fbe , damps ) ; % create LPF impulse response
x3=2* filter (b , 1 , x2 ) ; % LPF and scale signal
% extract upsampled pulses using correlation
 implemented
% as a convolving filter; filter with pulse and normalize
y=filter (fliplr(p)/(pow(p)*M),1,x3 );
% set delay to first symbol?sample and increment by M
z=y(0.5*fl+M:M:N*M) ; % downsample to symbol rate
figure(18),plot([1:length(z)],z,'.') % plot soft decisions
% decision device and symbol matching performance assessment
mprime=quantalph(z,[-3,-1,1,3])'; % quantize alphabet
cvar=(mprime-z)*(mprime-z)'/length(mprime) , % cluster variance
lmp=length (mprime ) ;
pererr=100*sum(abs(sign(mprime-m(1:lmp))))/lmp, % symbol error
% decode decision device output to text string
reconstructedmessage=pam2letters(mprime)

%The shortest lowpass filter you can use is the 4th order, because
 this
%allows rapid attenuation


cvar =

    4.8476
```

```
pererr =

     0


ans =

    'dropping last 3 PAM symbols'


reconstructedmessage =

    '01234 I wish I were an Oscar Meyer wiener 56789'


cvar =

   2.9274e-05


pererr =

     0


ans =

    'dropping last 3 PAM symbols'


reconstructedmessage =

    '01234 I wish I were an Oscar Meyer wiener 56789'


cvar =

   4.1244e-05


pererr =

     0


ans =

    'dropping last 3 PAM symbols'


reconstructedmessage =

    '01234 I wish I were an Oscar Meyer wiener 56789'
```

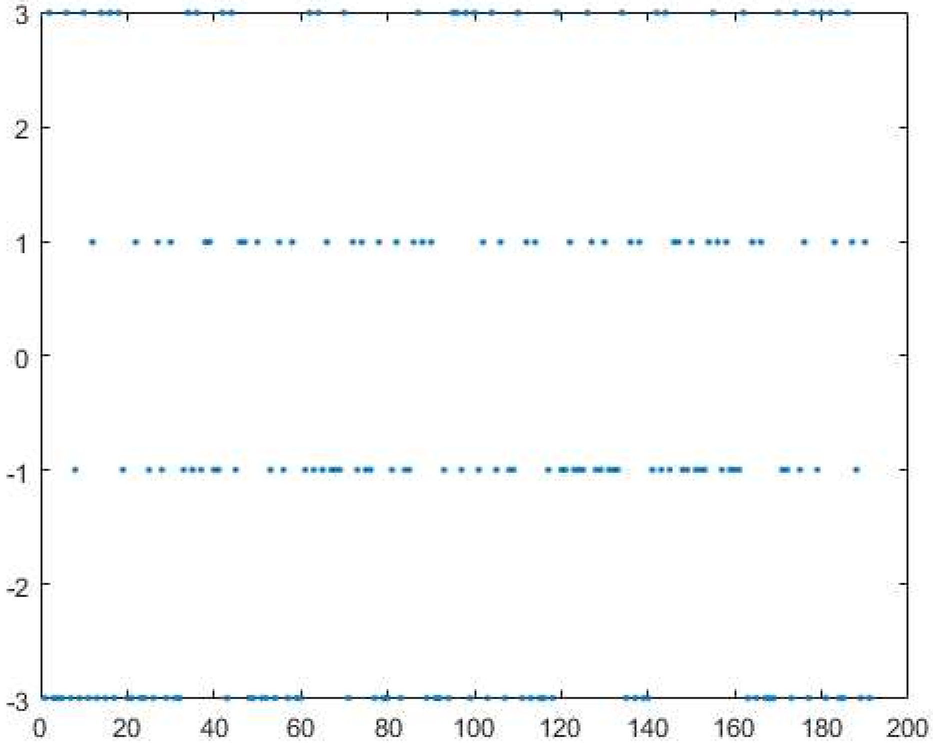
```
cvar =

    0.0912

pererr =

     0

ans =

    'dropping last 3 PAM symbols'

reconstructedmessage =

    '01234 I wish I were an Oscar Meyer wiener 56789'

cvar =

    0.2103

pererr =

   48.6911

ans =

    'dropping last 3 PAM symbols'

reconstructedmessage =

    'eeffeeYefifieYefefeeejeZffefeYeiefefiejefeeffii'

cvar =

   6.5608e-05

pererr =

     0

ans =

    'dropping last 3 PAM symbols'
```

```
reconstructedmessage =

    '01234 I wish I were an Oscar Meyer wiener 56789'


cvar =

   1.1759e-05


pererr =

     0


ans =

    'dropping last 3 PAM symbols'


reconstructedmessage =

    '01234 I wish I were an Oscar Meyer wiener 56789'


cvar =

    2.2068


pererr =

   16.9312


ans =

    'dropping last 1 PAM symbols'


reconstructedmessage =

    '013340M0s)s(0M0s%se0qn0O331s0Meyes0s)enes053389'


cvar =

    0.2241


pererr =
   48.6911


ans =

    'dropping last 3 PAM symbols'


reconstructedmessage =

    'eeffeeYefifieYefefeeejeZffefeYeiefefiejefeeffii'


cvar =

    0.4813


pererr =

     0


ans =

    'dropping last 3 PAM symbols'


reconstructedmessage =

    '01234 I wish I were an Oscar Meyer wiener 56789'


cvar =

    0.5938


pererr =

     0


ans =

    'dropping last 3 PAM symbols'


reconstructedmessage =

    '01234 I wish I were an Oscar Meyer wiener 56789'


cvar =
    0.3811


pererr =

     0


ans =

    'dropping last 3 PAM symbols'


reconstructedmessage =

    '01234 I wish I were an Oscar Meyer wiener 56789'
```

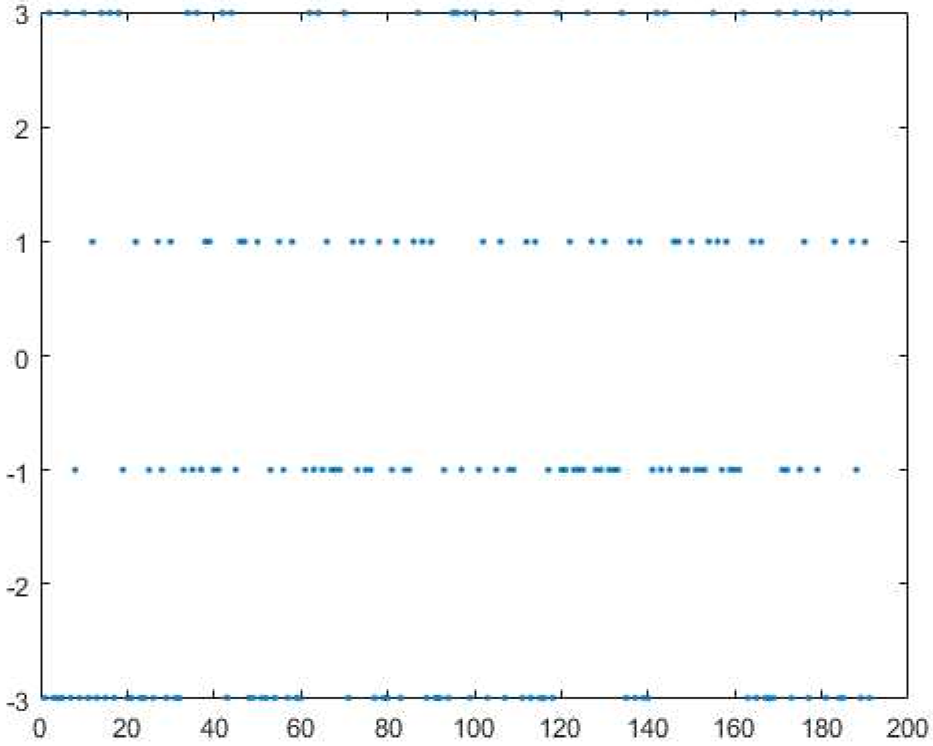


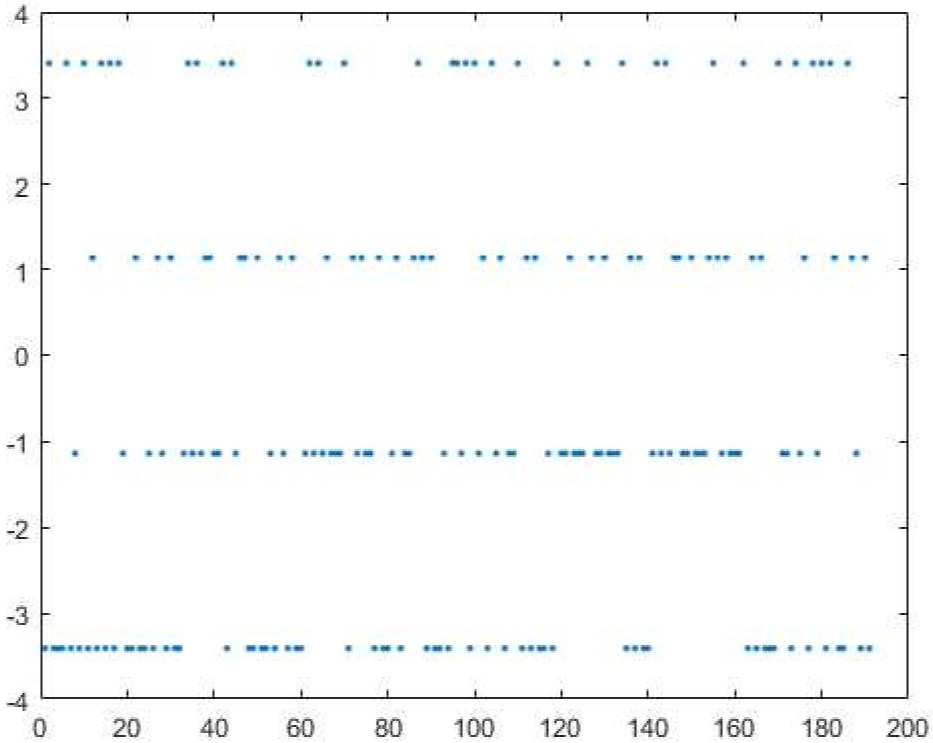


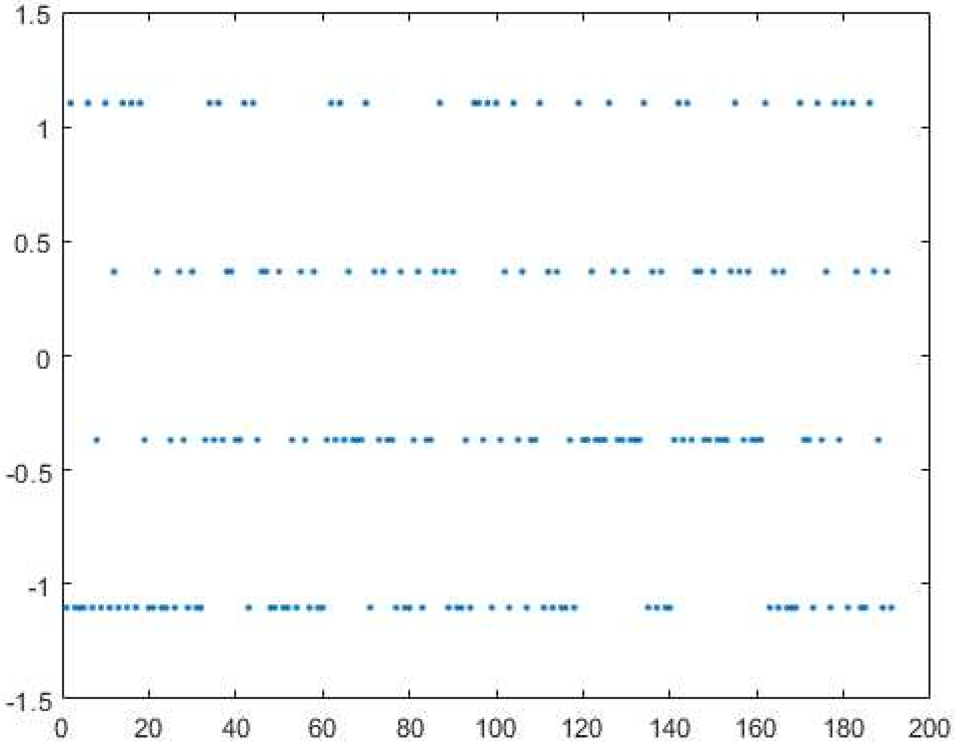
6
4
2
0
-2
-4
-6
0
20
40
60
80
100
120
140
160
180
200

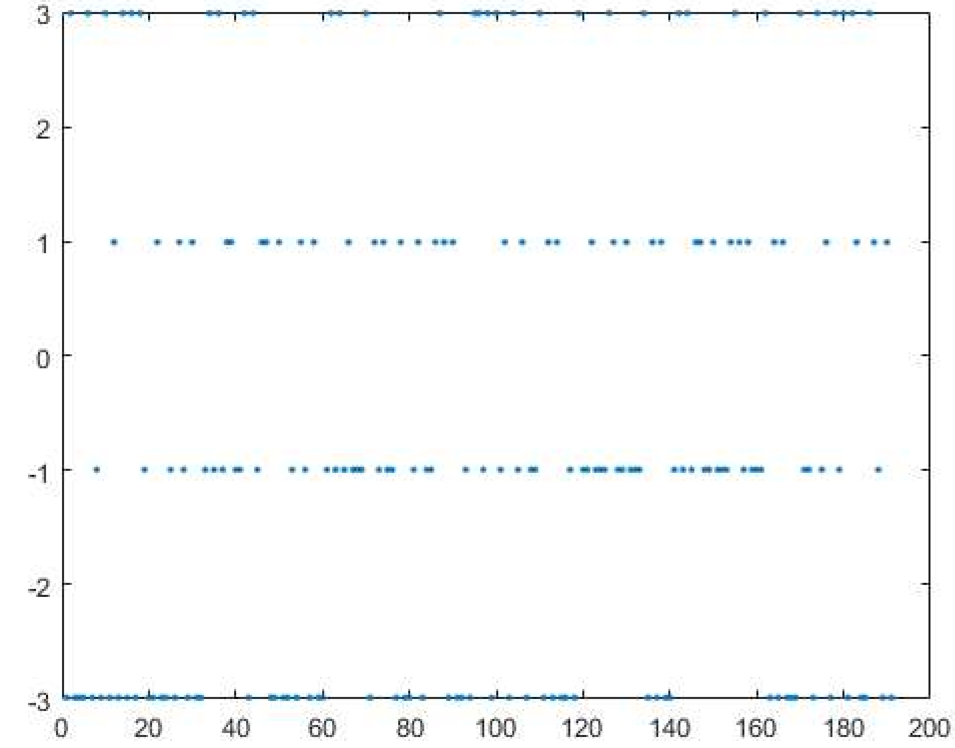


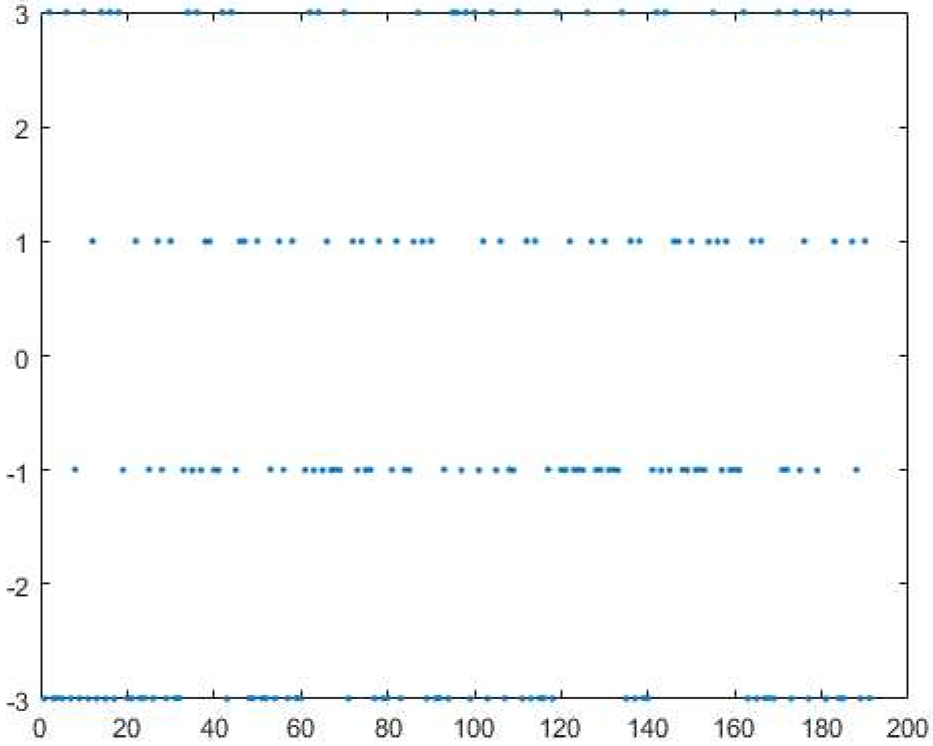
3
2
1
0
-1
-2
-3
0
20
40
60
80
100
120
140
160
180
200

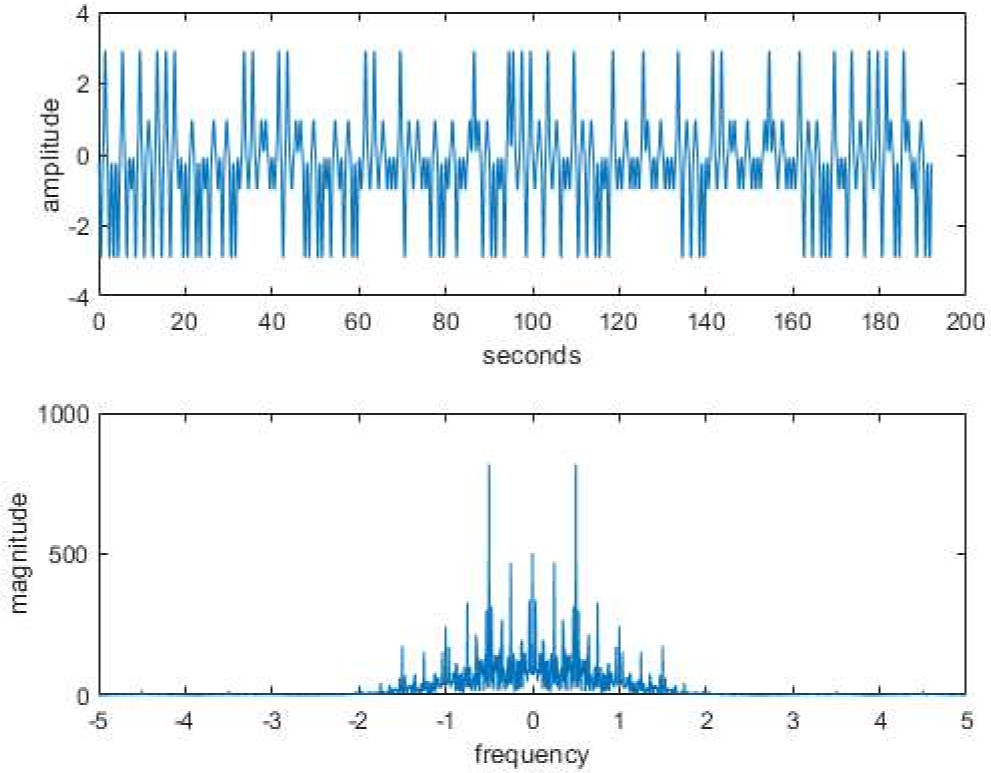

3
2
1
0
-1
-2
-3
0
20
40
60
80
100
120
140
160
180
200

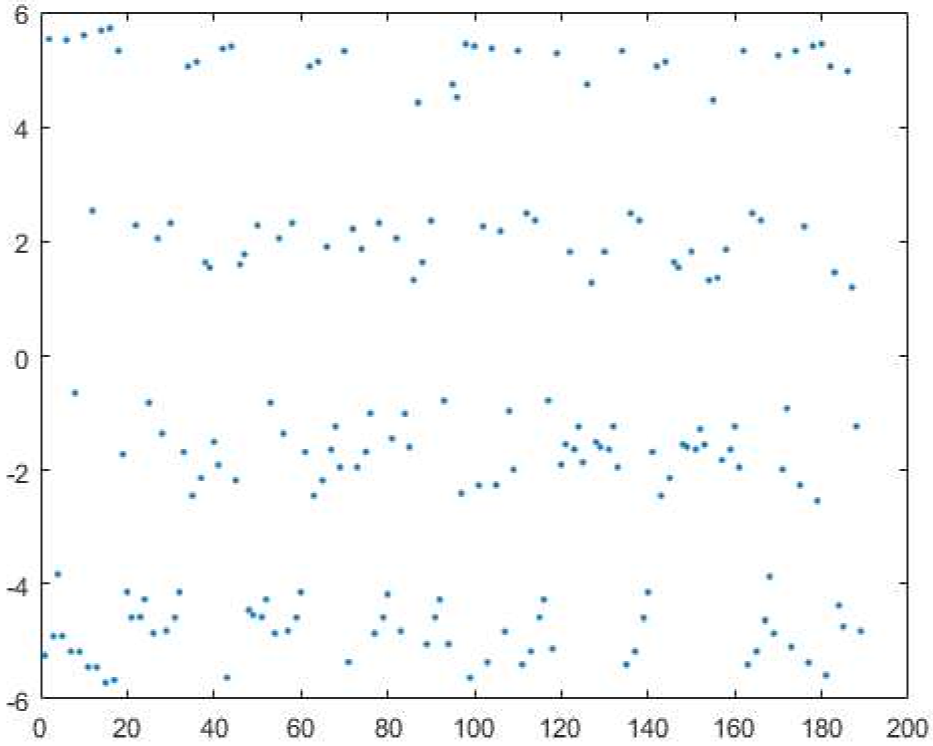

4
3
2
1
0
-1
-2
-3
-4
0
20
40
60
80
100
120
140
160
180
200

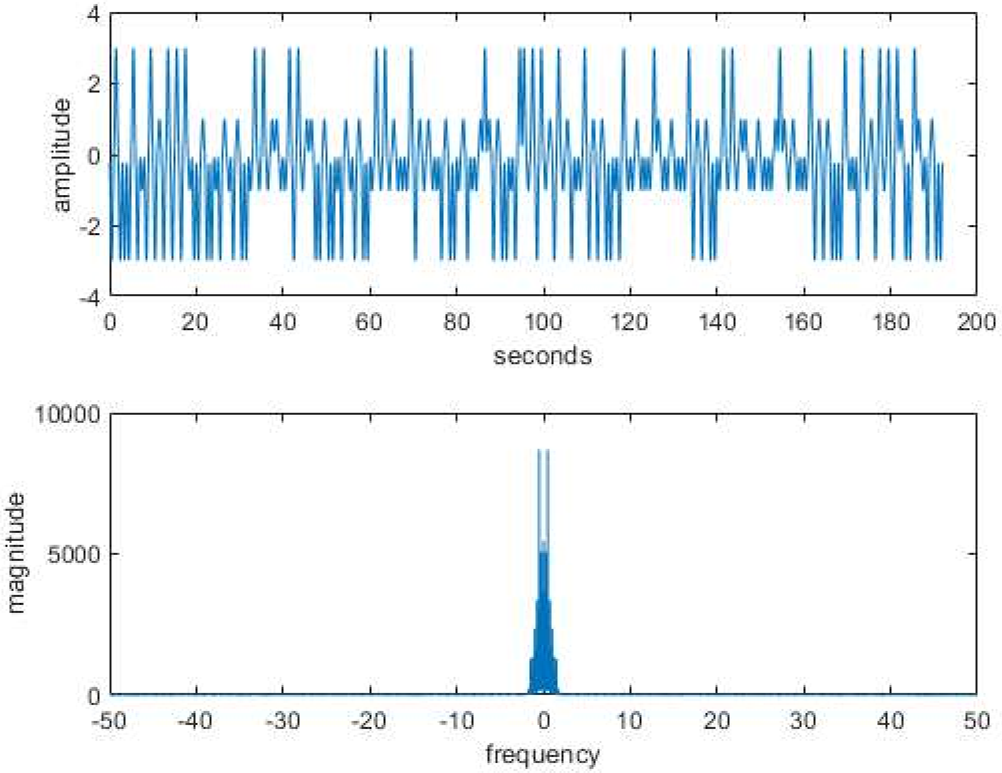

1.5
1
0.5
0
-0.5
-1
-1.5
0
20
40
60
80
100
120
140
160
180
200

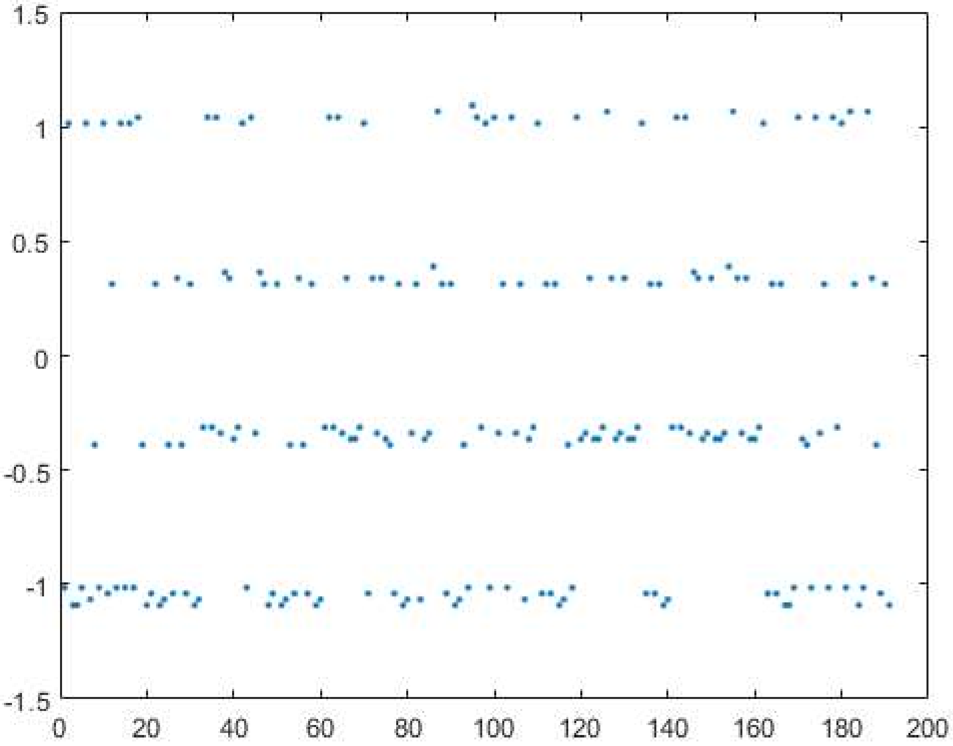

3
2
1
0
-1
-2
-3
0
20
40
60
80
100
120
140
160
180
200

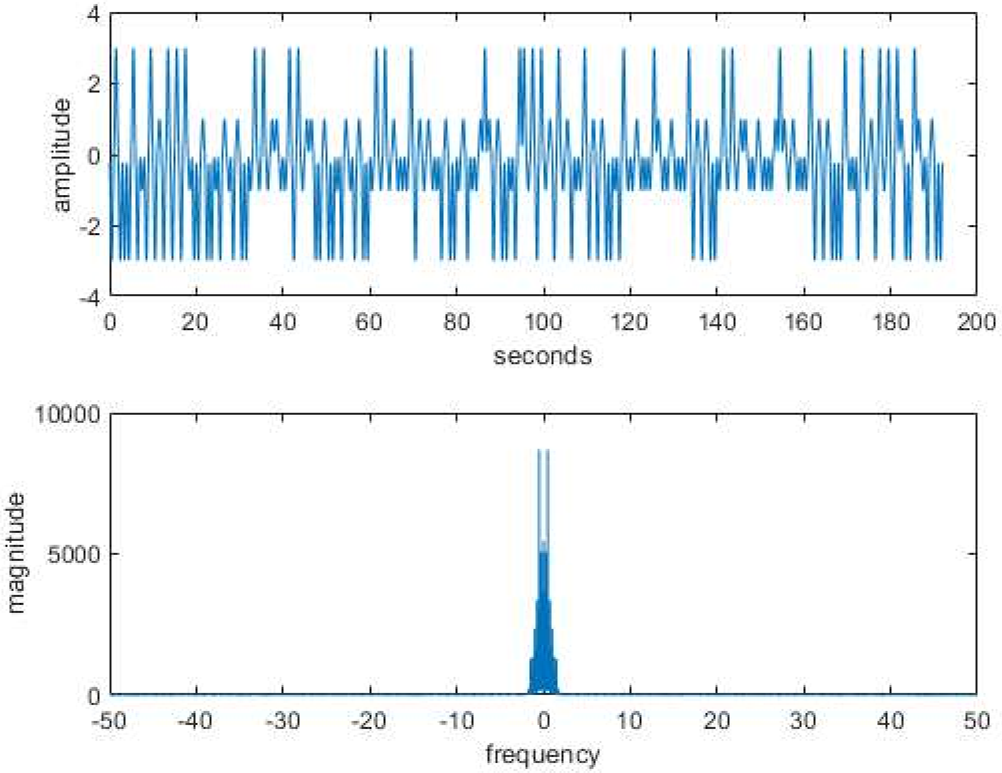

3
2
1
0
-1
-2
-3
0
20
40
60
80
100
120
140
160
180
200

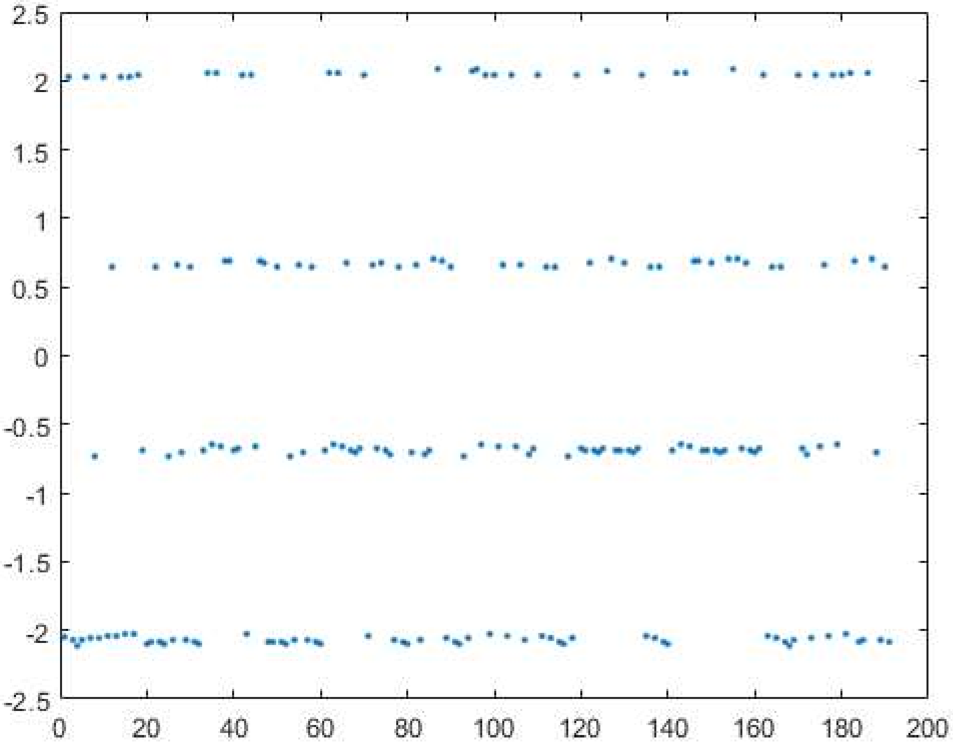

4
2
0
-2
-4
amplitude
0
20
40
60
80
100
120
140
160
180
200
seconds
1000
500
0
magnitude
-5
-4
-3
-2
-1
0
1
2
3
4
5
frequency

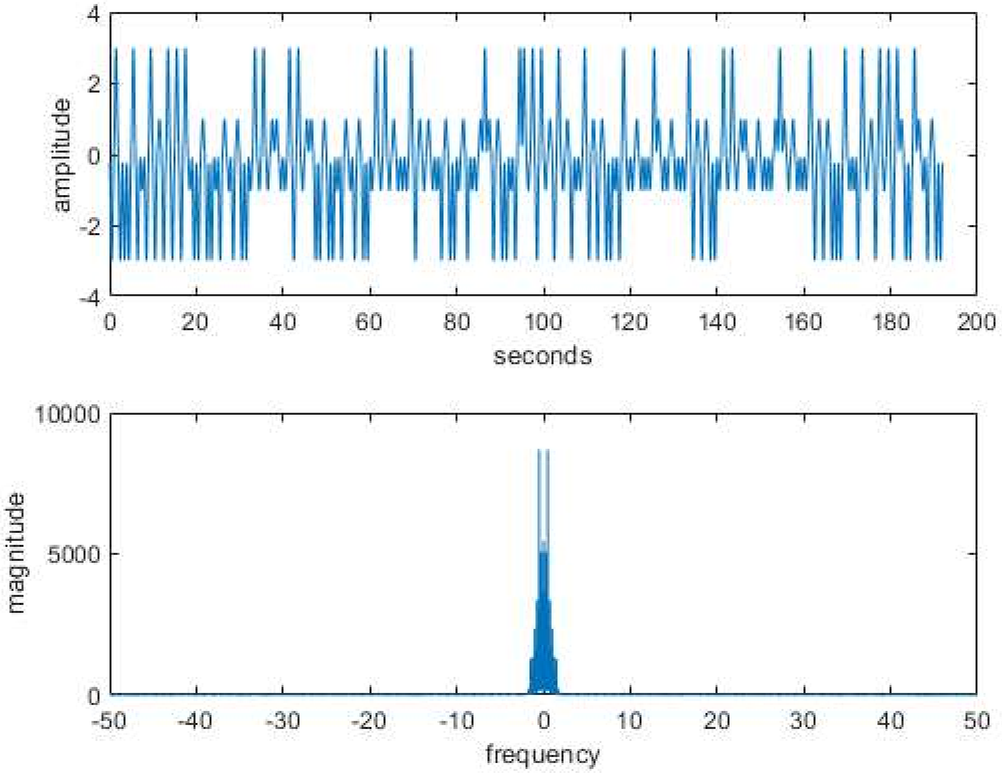

6
4
2
0
-2
-4
-6
0
20
40
60
80
100
120
140
160
180
200

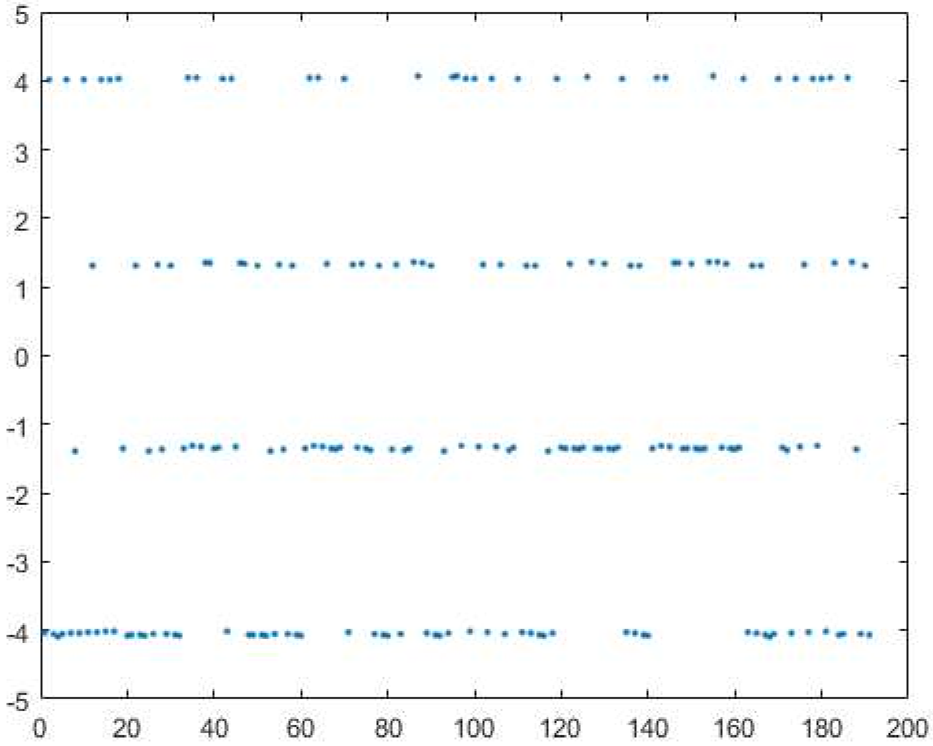

4
2
0
-2
-4
amplitude
0
20
40
60
80
100
120
140
160
180
200
seconds
10000
5000
0
magnitude
-50
-40
-30
-20
-10
0
10
20
30
40
50
frequency

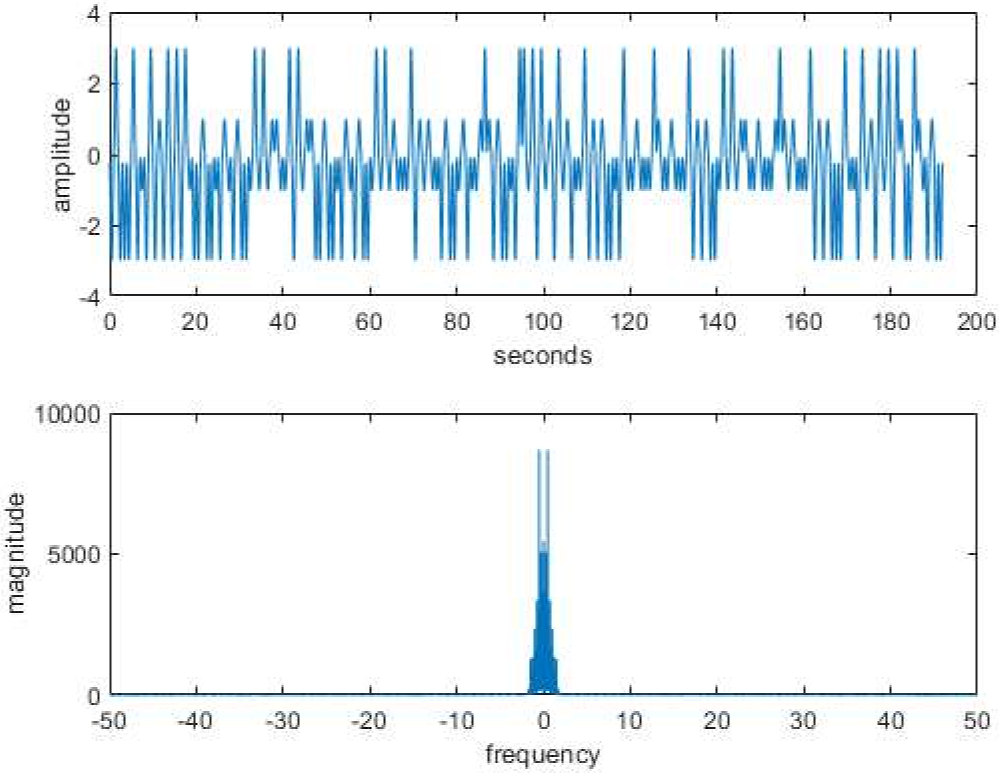

1.5
1
0.5
0
-0.5
-1
-1.5
0
20
40
60
80
100
120
140
160
180
200

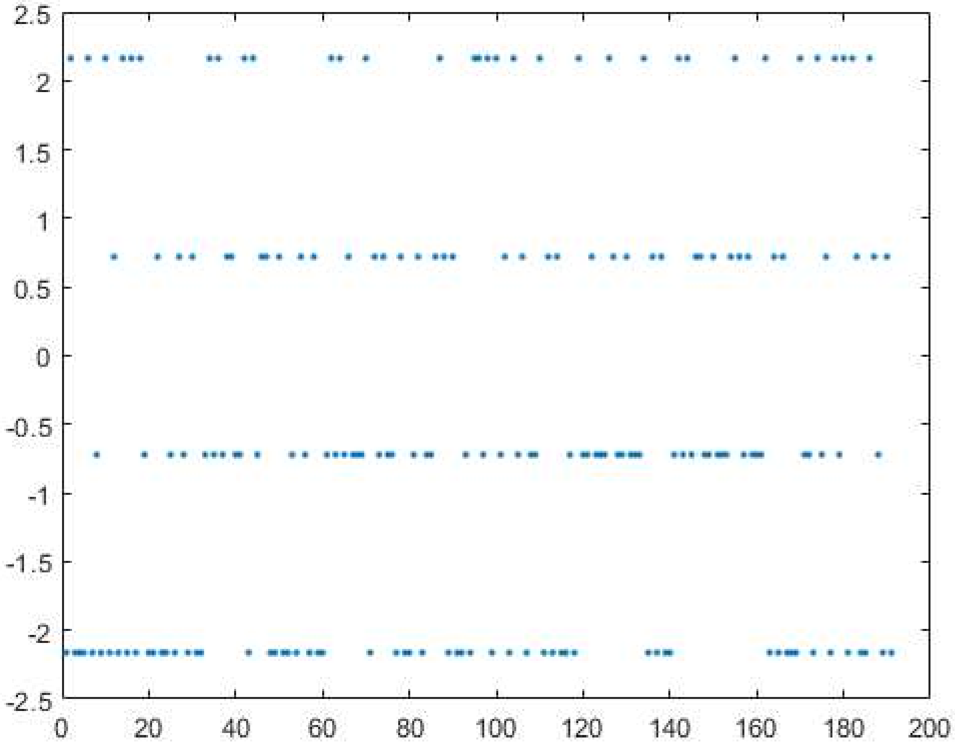

4
2
0
-2
-4
amplitude
0
20
40
60
80
100
120
140
160
180
200
seconds
10000
5000
0
magnitude
-50
-40
-30
-20
-10
0
10
20
30
40
50
frequency


2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
0
20
40
60
80
100
120
140
160
180
200


4
2
0
-2
-4
amplitude
0
20
40
60
80
100
120
140
160
180
200
seconds
10000
5000
0
magnitude
-50
-40
-30
-20
-10
0
10
20
30
40
50
frequency


5
4
3
2
1
0
-1
-2
-3
-4
-5
0
20
40
60
80
100
120
140
160
180
200


4
2
0
-2
-4
amplitude
0
20
40
60
80
100
120
140
160
180
200
seconds
10000
5000
0
magnitude
-50
-40
-30
-20
-10
0
10
20
30
40
50
frequency



Published with MATLAB® R2019b