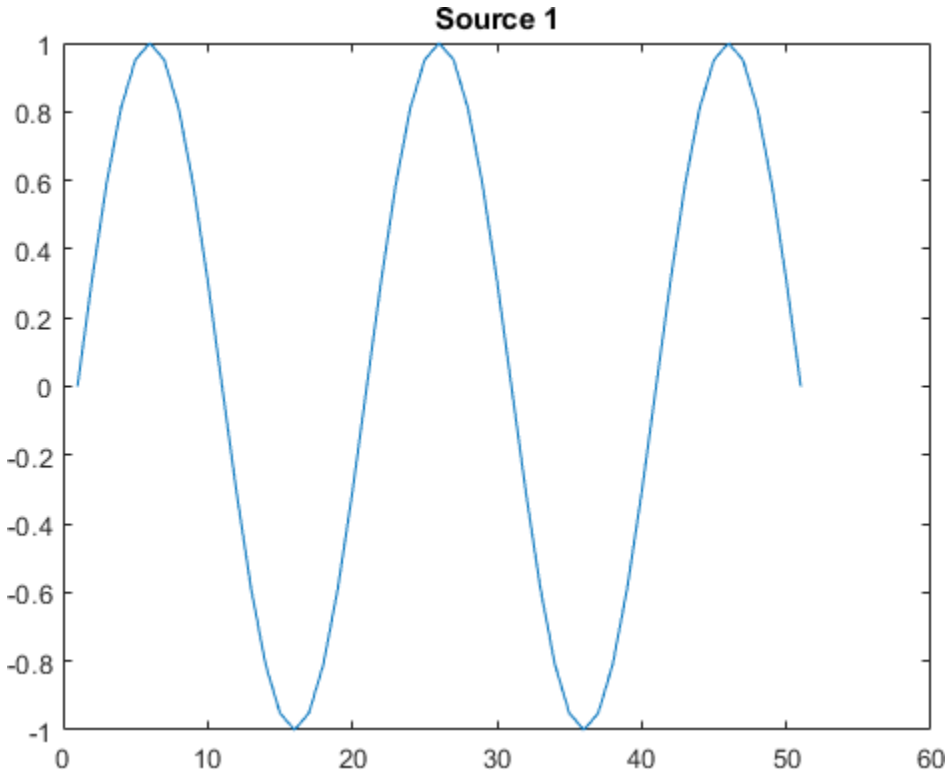
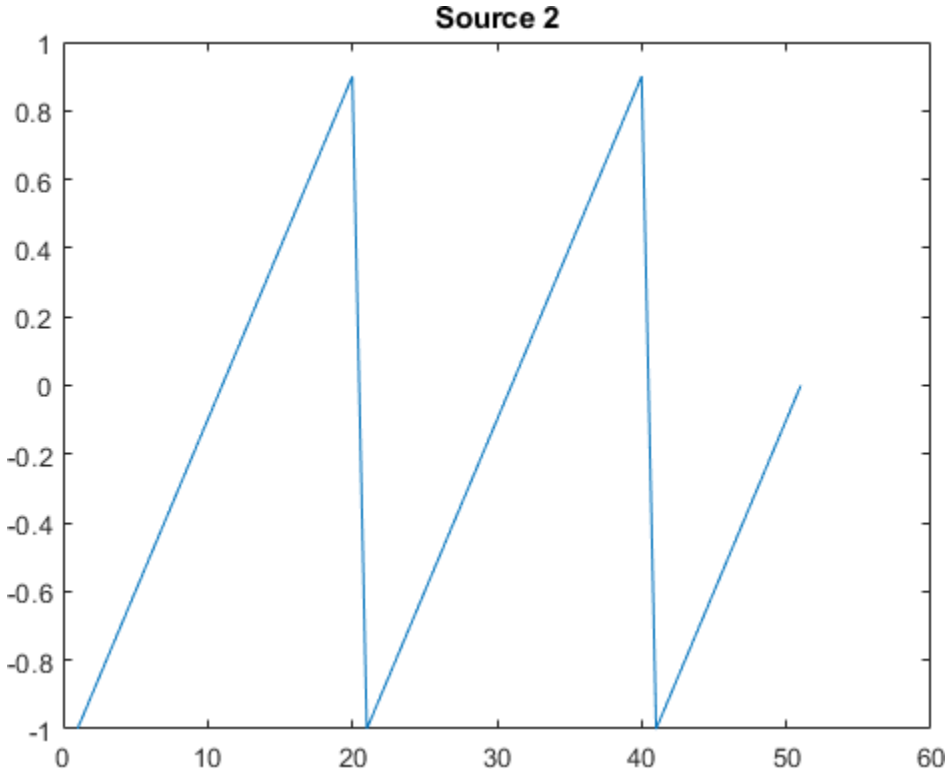
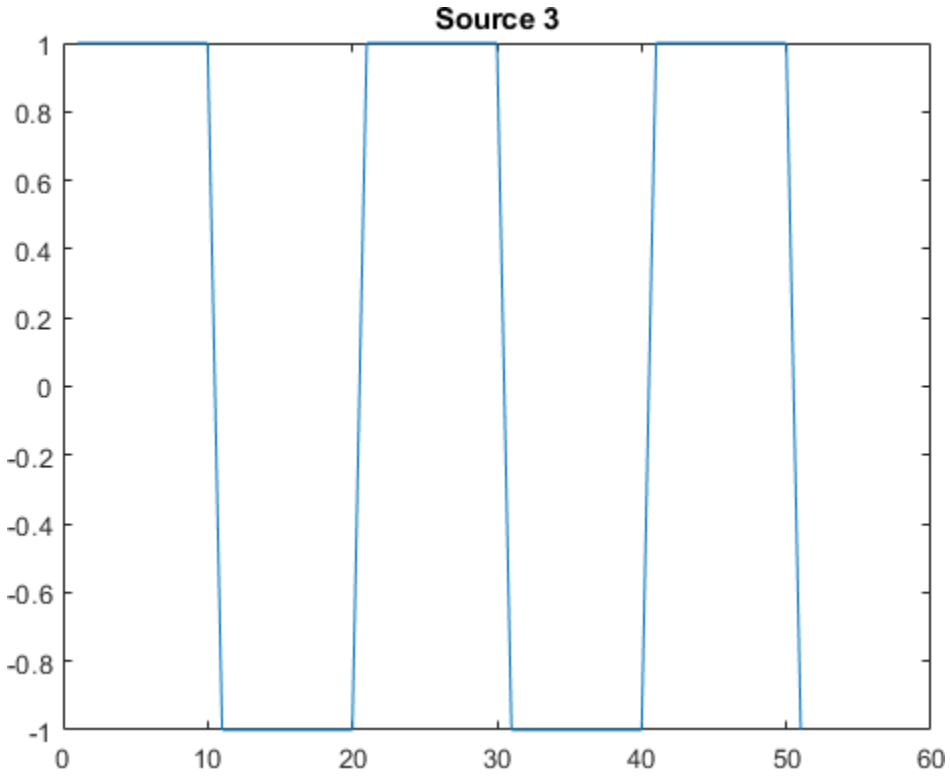
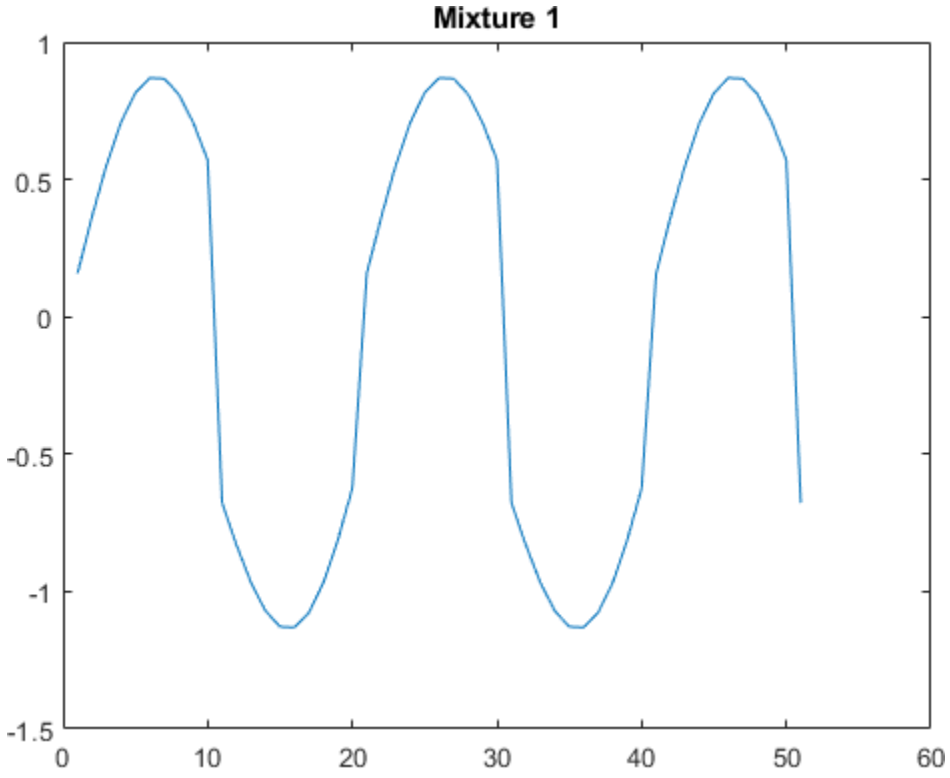
Generating Signals and Mitures

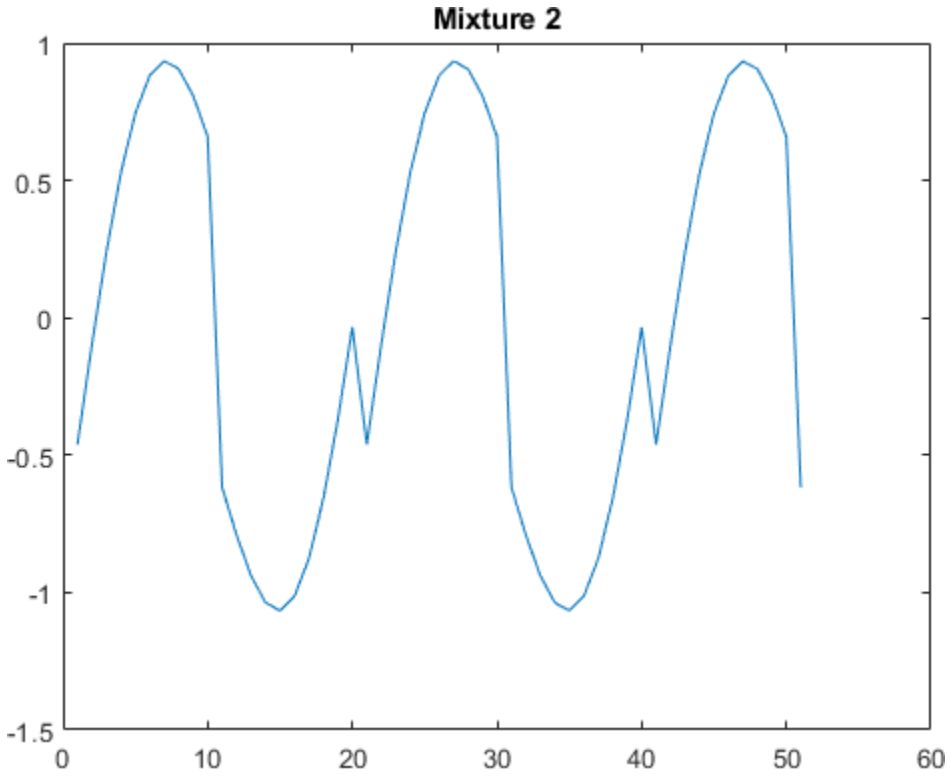
```
clc;
clear all;
close all;

% Defining the time period and frequency for the signals.
tp = 0:0.1:5;
freq = 0.5;

% Genrating 3 signals. Three types of waveforms have been used -
% Sine, Sawtooth and Square
signal1 = sin(2*pi*freq*tp);
signal2 = sawtooth(2*pi*freq*tp);
signal3 = square(2*pi*freq*tp);

% Plotting the 3 source sigals
figure, plot(signal1),title('Source 1');
figure, plot(signal2),title('Source 2');
figure, plot(signal3),title('Source 3');

% Generating two mixtures of signals by multiplying each signal with a
% random number and adding them up.
x1 = rand(1)*signal1 + rand(1)*signal2 + rand(1)*signal3;
x1 = x1/max(x1);
x2 = rand(1)*signal1 + rand(1)*signal2 + rand(1)*signal3;
x2 = x2/max(x2);
x1_bar = x1 - mean(x1);
x2_bar = x2 - mean(x2);

% Plot for generated mitures
figure, plot(x1_bar),title('Mixture 1');
figure, plot(x2_bar),title('Mixture 2');

% Combining the mixtures into a matrix
X = [x1 ; x2];

% Combining the sources into a matrix
S = [signal1 ; signal2 ; signal3];
```


Source 1
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
10
20
30
40
50
60


Source 2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
10
20
30
40
50
60


Source 3
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
10
20
30
40
50
60


Mixture 1
1
0.5
0
-0.5
-1
-1.5
0
10
20
30
40
50
60



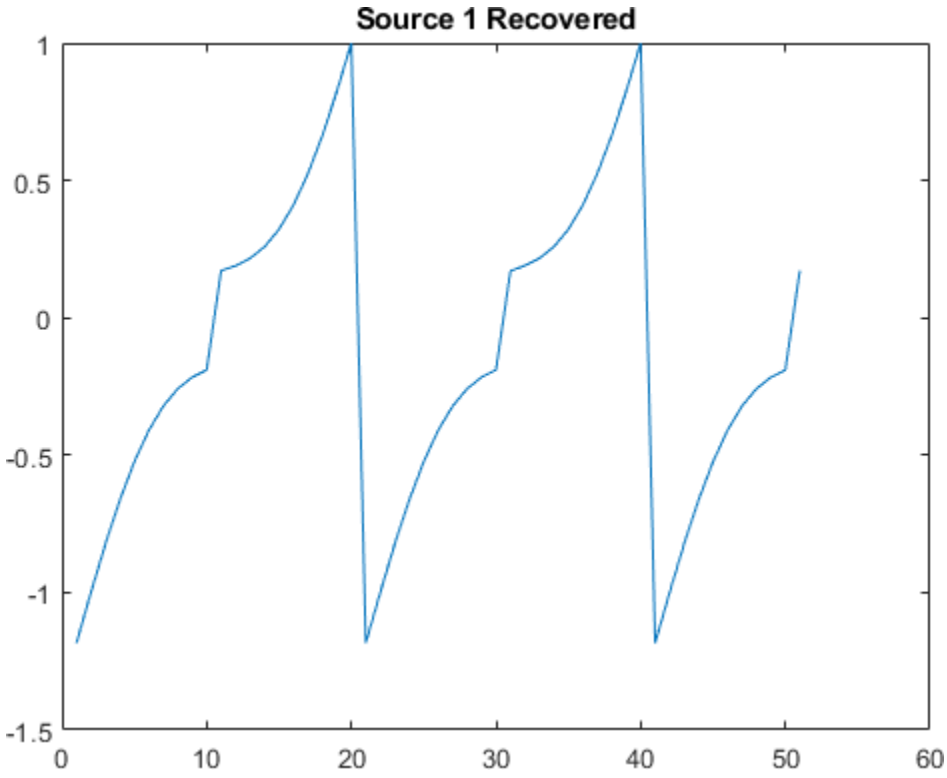
Obtaining Signals back from given Mixtures

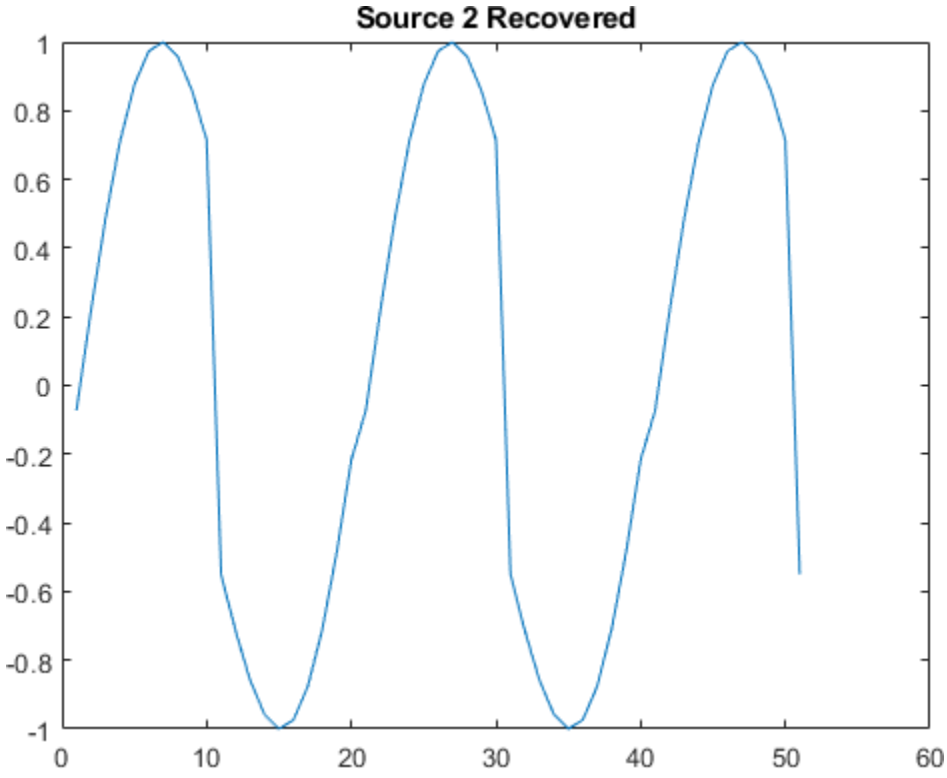
```
% A has been initialised with the values suggested in the paper
% Columns of A i.e ai = (cos(alphai), sin(alphai)) with alpha between
% 0 and pi.
alpha1 = pi/4;
alpha2 = pi/2;
alpha3 = 3*pi/4;
A = [cos(alpha1), cos(alpha2), cos(alpha3); -cos(alpha1), -
cos(alpha2), sin(alpha3)];

% Initializing Weights(Neurons) as suggested in the paper

w1 = [cos(alpha1)];
w_1 = [-cos(alpha1)];
w2 = [cos(alpha2)];
w_2 = [-cos(alpha2)];
w3 = [cos(alpha3)];
w_3 = [sin(alpha3)];
```

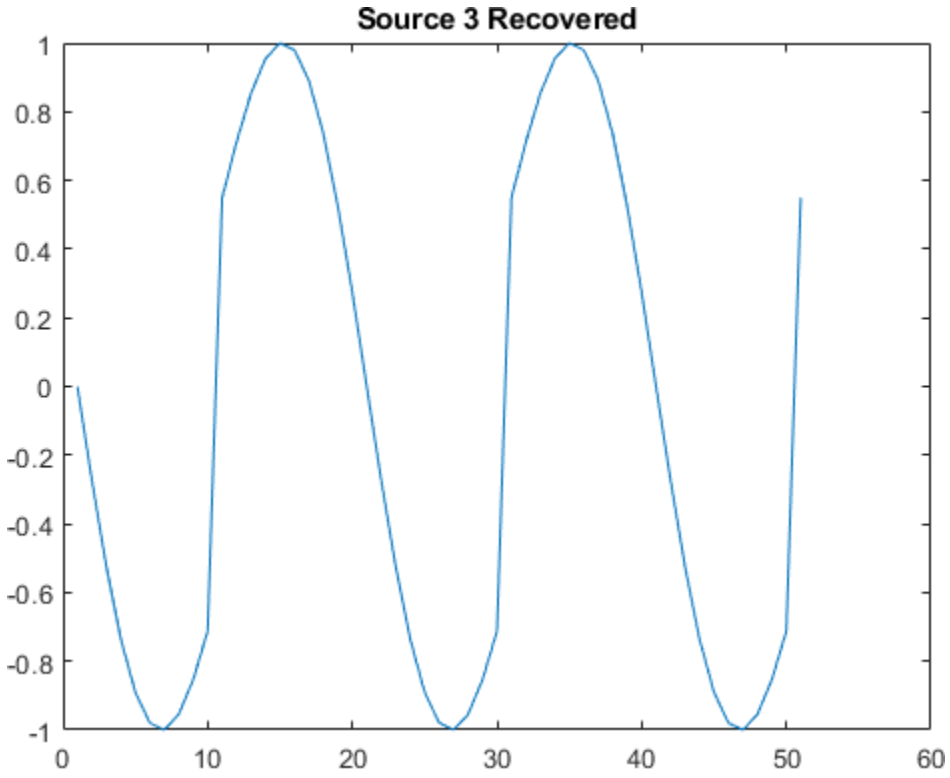
```
% Learning rate
n = 0.5;

% There are a total of 6 weights in the mixing matrix and each weight
% has been updated 50 times to obtain suitable values of weights for
 further
% calculations.
% Pie function as mentioned in the paper has been used.

for i= 2:51 % Updation of weight 1 for x1
    y = pie(x1_bar(1,1:i-1));
    a = w1 + n * pie(-1*y-w1);
    w1 = [1/3 pie(a)];
end
 w1_f = w1(1,51);

for i= 2:51 % Updation of weight 1 for x1
    y = pie(x1_bar(1,1:i-1));
    a = w2 + n * pie(y-w2);
    w2 = [2/3 pie(a)];
end
w2_f = w2(1,51);

for i= 2:51  % Updation of weight 3 for x1
    y = pie(x1_bar(1,1:i-1));
    a = w3 + n * pie(-1*y-w3);
    e1 = i-1;
    if(isnan(a))
        break;
    end
    w3 = [1 pie(a)];
end

if(e1 == 50)
  w3_f = w3(1,51);
else
  w3_f = w3(1,e1);
end

 for i= 2:51  % Updation of weight 1 for x2
    y = pie(x2_bar(1,1:i-1));
    a = w_1 + n * pie(y-w_1);
    w_1 = [-1/3 pie(a)];
 end
  w_1_f = w_1(1,51);

for i= 2:51  % Updation of weight 2 for x2
    y = pie(x2_bar(1,1:i-1));
    a = w_2 + n * pie(y-w_2);
    w_2 = [-2/3 pie(a)];
end
  w_2_f = w_2(1,51);

for i= 2:51  % Updation of weight 3 for x2
```

```
    y = pie(x2_bar(1,1:i-1));
    a = w_3 + n * pie(-1*y-w_3);
    e2 = i-1;
    if(isnan(a))
        break;
    end
    w_3 = [-1 pie(a)];
end

if(e2 == 50)
  w_3_f = w_3(1,51);
else
  w_3_f = w_3(1,e2);
end

%Final mixing matrix has been obtained after getting updated weights
A_obtained = [w1_f, w2_f, w3_f ; w_1_f, w_2_f, w_3_f];

%Scaling matrix L has been initialised
L = [0.5, 0, 0 ; 0, 0.5, 0 ; 0, 0, 1];

%Permuation matrix
P = eye(3);

%Approximation matrix of A has been found
B = A_obtained*P*L;
C = B-A;
error = norm(C);
disp(error);

%Psuedo inverse of B has been used to find the source
S_pred = pinv(B)*X;

%Error between obtained source and initial source is calculated
E = S-S_pred;

%Plotting the obtained source signals
s11 = S_pred(1,:);
s11 = s11/max(s11);
figure,plot(s11),title('Source 1 Recovered');

s22 = S_pred(2,:);
s22 = s22/max(s22);
figure,plot(s22), title('Source 2 Recovered');

s33 = S_pred(3,:);
s33 = s33/max(s33);
figure,plot(s33),title('Source 3 Recovered');
```

```
    1.4868
```


Source 1 Recovered
1
0.5
0
-0.5
-1
-1.5
0
10
20
30
40
50
60


Source 2 Recovered
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
10
20
30
40
50
60



Published with MATLAB® R2018a