



به نام خدا



دانشگاه تهران
دانشکده مهندسی برق و کامپیوتر
BSS

گزارش تمرین ۹

سالار صفردوست

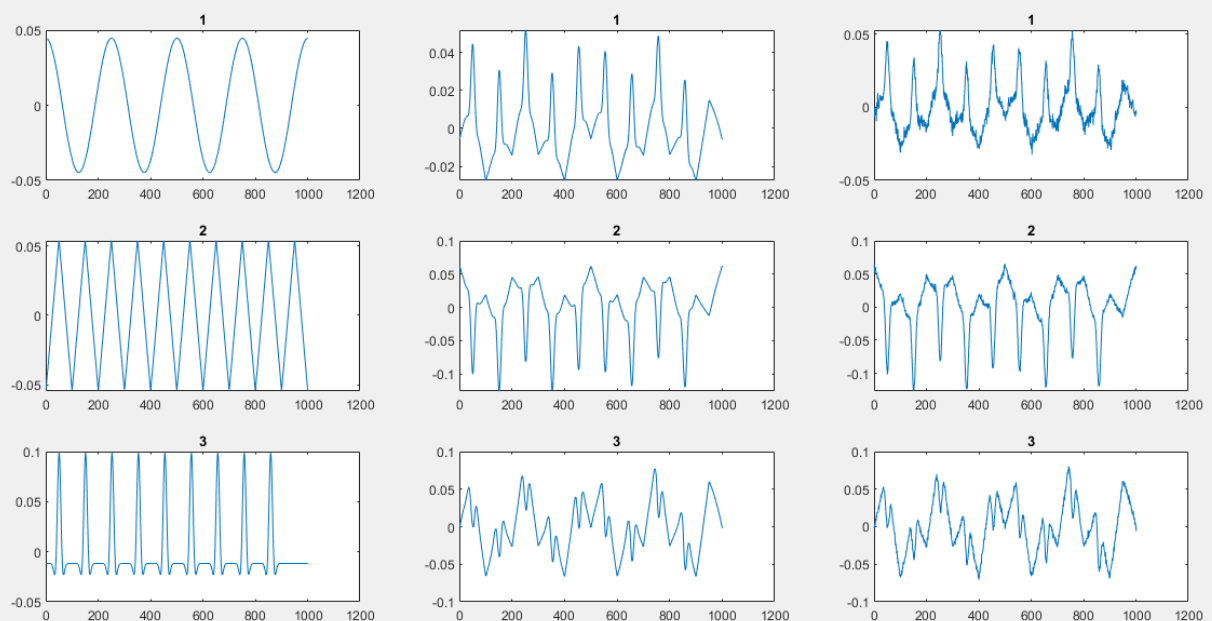
۸۱۰۱۹۹۴۵۰

۱۴۰۲/۰۳/۳۰

```

3 %% Preprocessing
4
5 - load hw9.mat
6
7 - X_raw = A*S;
8 - X = A*S+Noise;
9
10 - figure
11 - for i = 1:size(X,1)
12 -     subplot(3,3,3*i-2)
13 -     plot(S(i,:))
14 -     title(i)
15 -     subplot(3,3,3*i-1)
16 -     plot(X_raw(i,:))
17 -     title(i)
18 -     subplot(3,3,3*i)
19 -     plot(X(i,:))
20 -     title(i)
21 - end
22

```



```

1  function [B,S_hat,Similarity_Cell] = ICA_Deflation(Z,miu,Threshold)
2
3  [M,T] = size(Z);
4  B = normr(rand(M,M)*2-1);
5  df_dB = B.*0;
6  Psi = zeros(1,T);
7  Similarity_Cell = cell(M,1);
8
9
10 for m = 1:M
11     Similarity = [];
12     Similarity(1) = Threshold;
13     while Similarity(end) <= Threshold
14         y = B(m,:)*Z;
15         k = [y.*0+1 ; y ; y.^2 ; y.^3 ; y.^4 ; y.^5];
16         Psi = Psi_Extractor(k);
17
18         df_db = Psi*(Z.)/T ;
19
20         b = B(m,:) - miu*df_db;
21         if m~=1
22             b = ((eye(M)-B(1:m-1,:).'*B(1:m-1,:))*b.').';
23         end
24         b = normr(b);
25         Similarity = [Similarity,B(m,:)*b.'];
26         B(m,:) = b;
27     end
28     Similarity_Cell{m,1} = Similarity;
29 end
30
31 S_hat = B*Z;
32
33 end

```

```

23 %% Question 1_1
24
25 miu = 0.1;
26 Threshold = 1-1e-10;
27
28 [U,L] = eig(X*X. ');
29 W = L^-(1/2) * U. ';
30 Z = W*X;
31
32 [B,S_hat,Similarity_Cell] = ICA_Deflation(Z,miu,Threshold);
33
34 Final_Matrix = B*W*A;
35

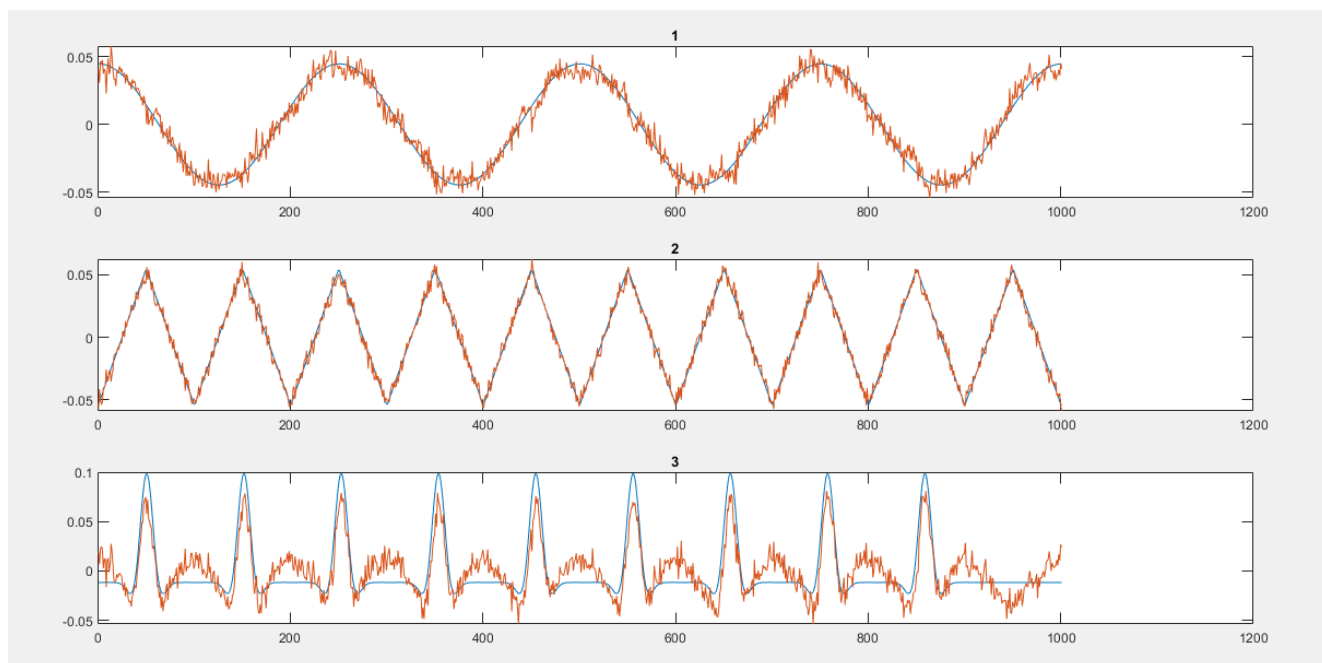
```

Variables - Final_Matrix			
3x3 double			
	1	2	3
1	-0.9880	-0.0964	0.1042
2	-0.0237	0.9962	-0.0091
3	0.0693	-0.5771	1.1367

```

36 %% Question 1_2
37
38 S_hat = Scale_Permutation_Recovery(S,S_hat);
39
40 figure
41 for n = 1:size(S,1)
42     subplot(3,1,n)
43     plot(S(n,:),hold on,plot(S_hat(n,:));
44     title(n)
45 end
46
47 Error1 = (norm(S-S_hat,'fro')/norm(S,'fro'))^2;
48

```



Error1

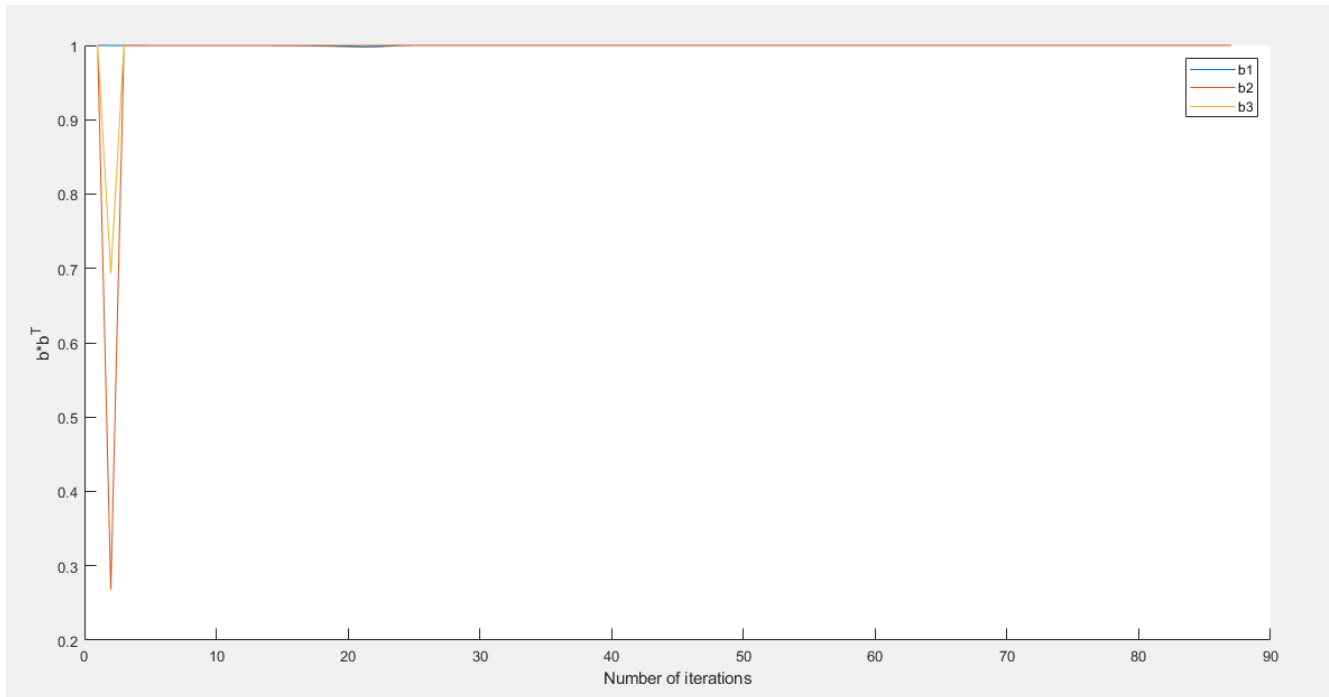
0.1269

*در این روش با توجه به هر بار ران(اینیشال متفاوت) ممکن بود به ارورهای دیگری مانند ۰.۱۶ نیز برسیم که بهترین خروجی به عنوان نتیجه قرار داده شد.

```

49 %% Question 1_3
50
51 figure
52 hold on
53 for m = 1:numel(Similarity_Cell)
54     plot(Similarity_Cell{m,1})
55 end
56 legend('b1','b2','b3')
57 xlabel('Number of iterations')
58 ylabel('b*b^T')
59

```



```

1 function [B,S_hat, Norm_Grad_list] = ICA_Equivariant(X,miu,Threshold)
2
3     [M,T] = size(X);
4     B = normr(rand(M,M)*2-1);
5     Psi = zeros(M,T);
6
7     Norm_Grad_list(1) = Threshold;
8
9     while Threshold <= Norm_Grad_list(end)
10         Y = B*X;
11         for m = 1:M
12             y = Y(m,:);
13             k = [y.*0+1 ; y ; y.^2 ; y.^3 ; y.^4 ; y.^5];
14             Psi(m,:) = Psi_Extractor(k);
15         end
16         D = Psi*(Y.')./T;
17         D( 1 == reshape((eye(size(D))),1,[]) ) =...
18             [0 0 0];
19         % ((1-diag(Y*Y.')./T)).');
20         Norm_Grad_list = [Norm_Grad_list,norm(D,'fro')];
21         B = normr(B - miu.*D*B);
22     end
23
24     S_hat = B*X;
25
26 end

```

```

60 %% Question 2_1
61
62 miu = 0.01;
63 Threshold = 1e-6;
64
65 [B,S_hat, Norm_Grad_list] = ICA_Equivariant(X*100,miu,Threshold);
66
67 Final_Matrix = B*A;
68

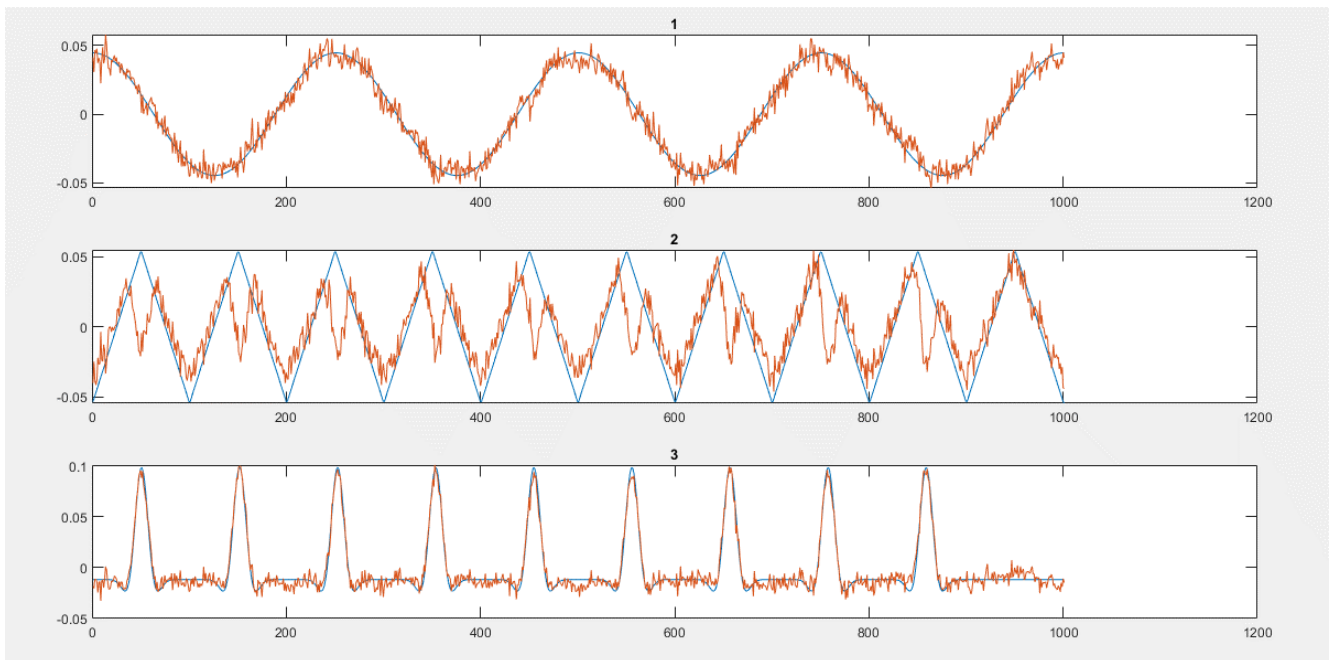
```

Variables - Final_Matrix			
3x3 double			
	1	2	3
1	0.4374	0.0420	-0.0426
2	-0.0134	0.0667	0.4679
3	0.0127	-0.3192	0.2220

```

69 %% Question 2_2
70
71 S_hat = Scale_Permutation_Recovery(S,S_hat);
72
73 figure
74 for n = 1:size(S,1)
75     subplot(3,1,n)
76     plot(S(n,:),hold on,plot(S_hat(n,:));
77     title(n)
78 end
79
80 Error2 = (norm(S-S_hat,'fro')/norm(S,'fro'))^2;
81

```

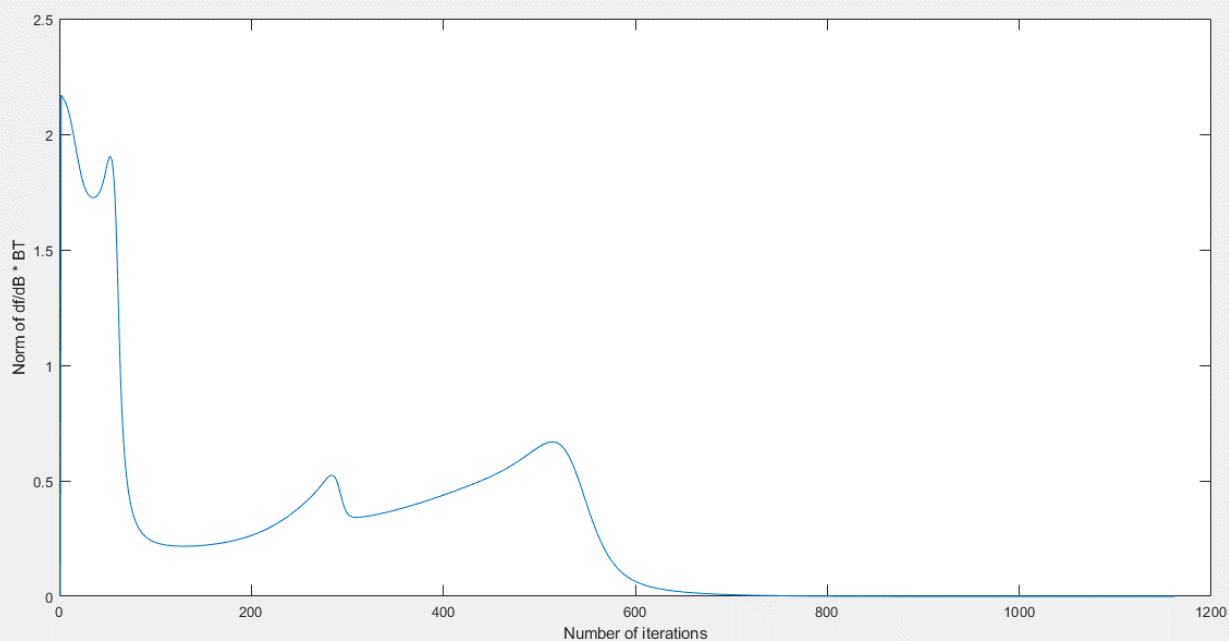


 Error2 0.1967

```

82 %% Question 2_3
83
84 - figure
85 - plot(Norm_Grad_list)
86 - xlabel('Number of iterations')
87 - ylabel('Norm of df/dB * BT')

```



بخش سوم

با توجه به اینکه ارور در تمرین ۸ برابر ۰.۱۹۶۷ و تعداد ایتريشن لازم برای همگرایی تقریباً برابر ۱۴۰۰ بوده است، می‌توان نتیجه گرفت که:

۱- روش Deflation از دو روش دیگر دقت بسیار بالاتری دارد و همچنین بسیار سریعتر همگرا می‌شود، پس میان سه روش Deflation از همه بهتر است.

۲- میان دو روش دیگر دیده می‌شود هر دو به یک ماتریس جداکننده و یک ارور منتهی شده‌اند و تنها تفاوت آن‌ها در سرعت همگرایی بیشتر روش Equivariant می‌باشد.