

به نام خدا



Blind Source Separation (BSS)

تکلیف شماره

4

محمد رضا آرانی

810100511

دانشگاه تهران

1402/01/24

جدول محتویات

3	بخش اول:
3	قسمت-1:
5	قسمت-2:
9	قسمت-3:
12	قسمت-4:
13	قسمت-5:
15	قسمت-6:
17	بخش دوم:
17	قسمت-1:
22	قسمت-2:
24	قسمت-3:
25	قسمت-4:

بخش اول:

بخش اول)

در این بخش می خواهیم جداسازی کور منابع در حالتی که منابع غیر ایستار و ناهمبسته هستند را بررسی کنیم. سیگنال دو منبع در پنجره ی زمانی k ($k = 1, 2, \dots, 5$) به صورت زیر است:

$$s_1^{(k)}(t) = c_k \sin(2\pi t) \quad k-1 \leq t < k$$

$$s_2^{(k)}(t) = d_k \sin(4\pi t) \quad k-1 \leq t < k$$

مقادیر c_k و d_k از درایه ی k بردارهای زیر انتخاب شده است:

$$c = [0.2 \quad 0.4 \quad 0.6 \quad -0.1 \quad -0.3]$$

$$d = [0.1 \quad 0.3 \quad -0.2 \quad 0.5 \quad -0.3]$$

با فرض $f_s = 20 \text{ Hz}$ و ماتریس مخلوط کننده زیر، ماتریس مشاهدات $X = A S$ را که ابعاد آن 2×100 است تولید کنید.

$$A = \begin{bmatrix} 0.8 & -0.6 \\ 0.6 & 0.8 \end{bmatrix}$$

ماتریس X را تا انتهای بخش اول ثابت در نظر بگیرید.

پس برای این مسئله، هر دو شرط Non-Stationary بودن و Uncorrelated بودن رو داریم! این یعنی مسئله جواب یکتا خواهد داشت.

قسمت-1:

الف) سیگنال منابع و سیگنال مشاهدات را رسم کنید.

برای تولید داده ها داریم:

```
fs = 20; %Hz
ts = 1/fs;

T_rec = 5-ts;
t = 0:ts:T_rec;

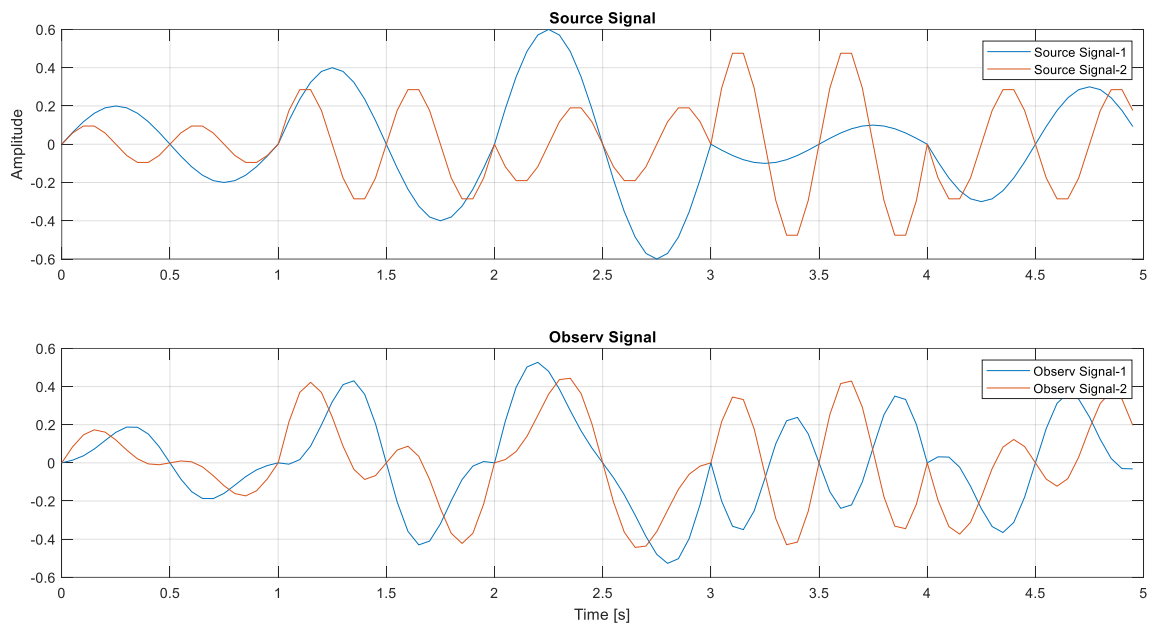
c = [0.2, 0.4, 0.6, -0.1, -0.3];
d = [0.1 0.3 -0.2 0.5 -0.3];

C = repmat(c, fs, 1);
D = repmat(d, fs, 1);
```

```
s1 = (C(:))'.*sin(2*pi*t);
s2 = (D(:))'.*sin(2*pi*t);

S = [s1;s2];

% Generate X:
A = [0.8,-0.6; 0.6 , 0.8];
X = A*S;
```



شکل 1

در شکل بالا، سیگنال منبع به همراه سیگنال مشاهدات برای هر دو کانال آورده شده است

قسمت-2:

ب) حال با دید BSS به مساله نگاه کنید. فقط با استفاده از داده های دو پنجره ی اول، سیگنال منابع را تخمین بزنید. کیفیت منابع تخمین شده را با معیار زیر گزارش کنید. قبل از استفاده از این رابطه حتماً ابهام دامنه و جایگشت را برطرف کنید.

$$E = \frac{\|\hat{S} - S\|_F^2}{\|S\|_F^2}$$

در صورت استفاده از داده های دو پنجره ی اول، باید از روابط زیر پیروی کرد:

$$\begin{aligned} \underline{X} &= \underline{A} \underline{S} \\ \underline{X}^{(1)} &= \underline{A} \underline{S}^{(1)} \rightarrow \underline{X}^{(1)} \underline{X}^{(1)T} = \underline{R}_x^{(1)} = \underline{A} \underline{S}^{(1)} \underline{S}^{(1)T} \underline{A}^T \\ \underline{X}^{(2)} &= \underline{A} \underline{S}^{(2)} \rightarrow \underline{X}^{(2)} \underline{X}^{(2)T} = \underline{R}_x^{(2)} = \underline{A} \underline{S}^{(2)} \underline{S}^{(2)T} \underline{A}^T \\ &\Rightarrow \begin{cases} \underline{R}_x^{(1)} = \underline{A} \underline{R}_s^{(1)} \underline{A}^T \\ \underline{R}_x^{(2)} = \underline{A} \underline{R}_s^{(2)} \underline{A}^T \end{cases} \xrightarrow{\underline{B} = \underline{A}^{-1}} \begin{cases} \underline{B} \underline{R}_x^{(1)} \underline{B}^T = \underline{R}_s^{(1)} \\ \underline{B} \underline{R}_x^{(2)} \underline{B}^T = \underline{R}_s^{(2)} \end{cases} \xrightarrow{\text{مقرّر}} \begin{cases} \underline{R}_s^{(1)} \\ \underline{R}_s^{(2)} \end{cases} \xrightarrow{\text{مقرّر}} \underline{R}_s \\ \underline{B} \underline{R}_x^{(1)} \underline{B}^T = \underline{B} \underline{R}_x^{(2)} \underline{B}^T \underline{\Gamma} \xrightarrow{\underline{B}^{-1}} \underline{R}_x^{(1)} \underline{B}^T = \underline{R}_x^{(2)} \underline{B}^T \underline{\Gamma} \\ \left[\underline{R}_x^{(1)T} \underline{R}_x^{(1)} \right] \underline{B}^T = \underline{B}^T \underline{\Gamma} \\ \underline{Q} \underline{Q}^T = \underline{I} \end{aligned}$$

شکل 2

روابط بیان شده برای این مسئله در جلسه-12

برای اینکه ماتریس مقادیر ویژه ما Orthonormal باشد ابتدا داده ها را سفید می کنیم:

% Data Whitening:

RX = X*X';

```
W_whit = chol(inv(RX));
```

```
Z = W_whit'*X;
```

```
RZ = Z*Z';
```

```
[U_Z , Lambda_Z] = eig(RZ);
```

داده‌های سفید شده دارای ماتریس کواریانس معادل ماتریس Identity بوده و همچنین مقادیر ویژه برابر 1 دارند. در زیر از سفید شدن داده‌ها می‌توان اطمینان حاصل کرد:

RZ =

```
1.0000 -0.0000
```

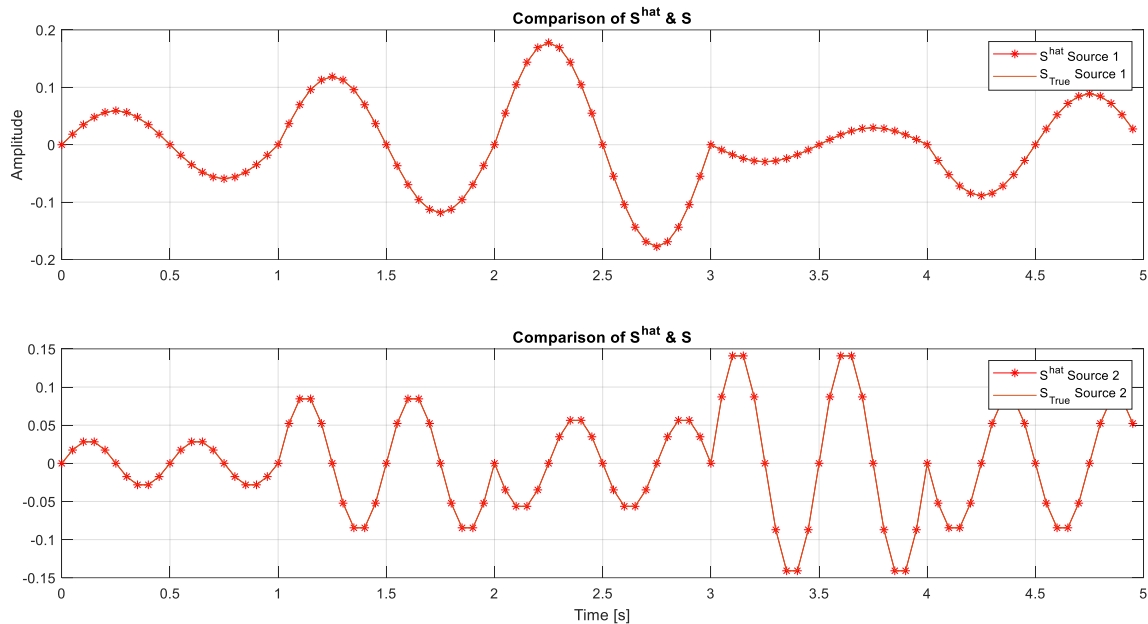
```
-0.0000 1.0000
```

Lambda_Z =

```
1.0000 0
```

```
0 1.0000
```

در نهایت پس از تخمین زدن ماتریس منابع، در شکل زیر می‌توان آنها را با مقادیر واقعی مقایسه کرد:



شکل 3

مشاهده می‌شود شکل کلی سیگنال منابع را به درستی یافته و در نقاطی با آن مماس نیز شده است. میزان خطا برای این حالت برابر:

$$\text{Error} = 1.8876\text{e-}31$$

می‌باشد.

```
S_Amp = S/norm(S,"fro");
B_Hat_Chosen = Perm_AMP_Disamb(B,Z,S_Amp);
S_hat = B_Hat_Chosen*Z;

S_hat_perm = S_hat;
% permutation disambiguation:
% S_hat_perm(1,:) = S_hat(2,:);
% S_hat_perm(2,:) = S_hat(1,:);

% Amplitude disambiguation:
% S_hat_perm = [1/max(S_hat_perm(1,:)) , 0; 0 , 1/max(S_hat_perm(2,:))]*S_hat_perm;
% S_hat_perm_AMP = [max(S(1,:)) , 0; 0 , max(S(2,:))]*S_hat_perm ;
S_hat_perm_AMP = S_hat_perm/norm(S_hat_perm,'fro');
Error = (norm(S_hat_perm_AMP-S_Amp,'fro'))^2/(norm(S_Amp,'fro'))^2 ;

disp("Error = "+Error);
```

برای رفع ابهام دامنه، فاز و جایگشت از تابع زیر استفاده می کنیم:

```

334 %% Perm_AMP_Disamb
335 function B_Hat_Chosen = Perm_AMP_Disamb(B_Hat,Z,S_Amp)
336 Perm_1 = B_Hat;
337 Perm_2 = -B_Hat;
338 Perm_3 = B_Hat([end, 1:end-1],:);
339 Perm_4 = -B_Hat([end, 1:end-1],:);
340 Perms = { Perm_1,Perm_2,Perm_3,Perm_4 };
341 % Estimate S:
342 S_hat_1 = Perm_1*Z;
343 S_hat_1_Normalised = -S_hat_1/norm(S_hat_1,'fro');
344 S_hat_2 = Perm_2*Z;
345 S_hat_2_Normalised = -S_hat_2/norm(S_hat_2,'fro');
346 S_hat_3 = Perm_3*Z;
347 S_hat_3_Normalised = -S_hat_3/norm(S_hat_3,'fro');
348 S_hat_4 = Perm_4*Z;
349 S_hat_4_Normalised = -S_hat_4/norm(S_hat_4,'fro');
350 % Calc Error:
351 Error_1 = (norm(abs(S_hat_1_Normalised)-abs(S_Amp),'fro'))^2 ;
352 Error_2 = (norm(abs(S_hat_2_Normalised)-abs(S_Amp),'fro'))^2 ;
353 Error_3 = (norm(abs(S_hat_3_Normalised)-abs(S_Amp),'fro'))^2 ;
354 Error_4 = (norm(abs(S_hat_4_Normalised)-abs(S_Amp),'fro'))^2 ;
355 [~, index] = min([Error_1,Error_2,Error_3,Error_4]);
356 B_Hat_Chosen = Perms{index};
357 end
358

```

شکل 4

```

%% Perm_AMP_Disamb
function B_Hat_Chosen = Perm_AMP_Disamb(B_Hat,Z,S_Amp)
Perm_1 = B_Hat;
Perm_2 = -B_Hat;
Perm_3 = B_Hat([end, 1:end-1],:);
Perm_4 = -B_Hat([end, 1:end-1],:);
Perms = { Perm_1,Perm_2,Perm_3,Perm_4 };
% Estimate S:
S_hat_1 = Perm_1*Z;
S_hat_1_Normalised = -S_hat_1/norm(S_hat_1,'fro');
S_hat_2 = Perm_2*Z;
S_hat_2_Normalised = -S_hat_2/norm(S_hat_2,'fro');
S_hat_3 = Perm_3*Z;
S_hat_3_Normalised = -S_hat_3/norm(S_hat_3,'fro');
S_hat_4 = Perm_4*Z;
S_hat_4_Normalised = -S_hat_4/norm(S_hat_4,'fro');
% Calc Error:
Error_1 = (norm(abs(S_hat_1_Normalised)-abs(S_Amp),'fro'))^2 ;
Error_2 = (norm(abs(S_hat_2_Normalised)-abs(S_Amp),'fro'))^2 ;
Error_3 = (norm(abs(S_hat_3_Normalised)-abs(S_Amp),'fro'))^2 ;
Error_4 = (norm(abs(S_hat_4_Normalised)-abs(S_Amp),'fro'))^2 ;
[~, index] = min([Error_1,Error_2,Error_3,Error_4]);
B_Hat_Chosen = Perms{index};
end

```


قسمت-3

در قدم بعدی، با استفاده از روش حل قطری سازی همزمان، تمامی پنجره های زمانی را در نظر گرفته و با استفاده از روش مطرح شده در کلاس، ماتریس B جدید را به دست می آوریم:

```
%% JOINT Diagonalization:

B_Hat = orth(rand(size(B)));
R_1 = zeros(size(RZ));
R_2 = R_1;

for iter = 1:10
    R_2 = 0;
    R_1 = 0;
    for k=1:K
        RZ_k = Z(:, (k-1)*fs+1:(k)*fs) * Z(:, (k-1)*fs+1:(k)*fs)';

        for j=1:length(B_Hat)
            if (j==1)
                b_j = B_Hat(j,:)' ;
                R_2 = R_2 + (RZ_k*b_j)*(RZ_k*b_j)';

            else
                b_j = B_Hat(j,:)' ; % Doubt Here!!!! Choose Columns or Rows of B???
                R_1 = R_1 + (RZ_k*b_j)*(RZ_k*b_j)';

            end
        end
    end

    [V_R1,Lambda_R1] = eig(R_1);
    [~, ind] = sort(diag(Lambda_R1),'descend');
    V_R1 = V_R1(:, ind);
    b_1 = V_R1(:,end);

    % For b_2:
    [V_R2,Lambda_R2] = eig(R_2);
    [~, ind] = sort(diag(Lambda_R2),'descend');
    V_R2 = V_R2(:, ind);
    b_2 = V_R2(:,end); % According to lowest eigen Value
```

```

% b_2 = ( eye(size(b_2)) - b_1*b_1' )*b_2 ;
b_2 = b_2 - (b_1'*b_2)*b_2 ;
b_2 = b_2/norm(b_2,2);

B_Hat = [b_1' ; b_2'];

end

S_hat_2 = B_Hat*Z;

S_hat_perm_2 = S_hat_2;
% permutation disambiguation:
% S_hat_perm_2(1,:) = S_hat_2(2,:);
% S_hat_perm_2(2,:) = -S_hat_2(1,:);

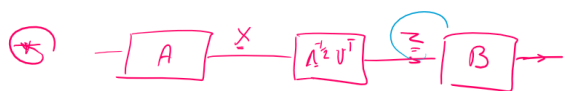
% Amplitude disambiguation:
% S_hat_perm_2 = [1/max(S_hat_perm_2(1,:)) , 0; 0 , 1/max(S_hat_perm_2(2,:))]*S_hat_perm_2;
% S_hat_perm_Amp_2 = [max(S(1,:)) , 0; 0 , max(S(2,:))]*S_hat_perm_2 ;
%
% S_hat_perm_Amp_2(2,:) = -S_hat_perm_Amp_2(2,:);
%
% Error = (norm(S_hat_perm_Amp_2-S,2))^2/(norm(S,2))^2 ;

S_hat_perm_Amp_2 = -S_hat_perm_2/norm(S_hat_perm_2, 'fro');
S_Amp = S/norm(S, "fro");
Error = (norm(S_hat_perm_Amp_2-S_Amp, 'fro'))^2/(norm(S_Amp, 'fro'))^2 ;

disp("Error = "+Error);

```

روش پیاده‌سازی شده همان روشی است که در جلسه-13 توضیح داده شده است.



$$f(B) = \sum_k \sum_i \sum_{j \neq i} (b_i^T R_k^{(i)} b_j)^2$$

$$B = \arg \min f(B)$$

$$\text{s.t. } b_i^T b_j = 1 \quad b_i^T b_j = 0 \quad i \neq j$$

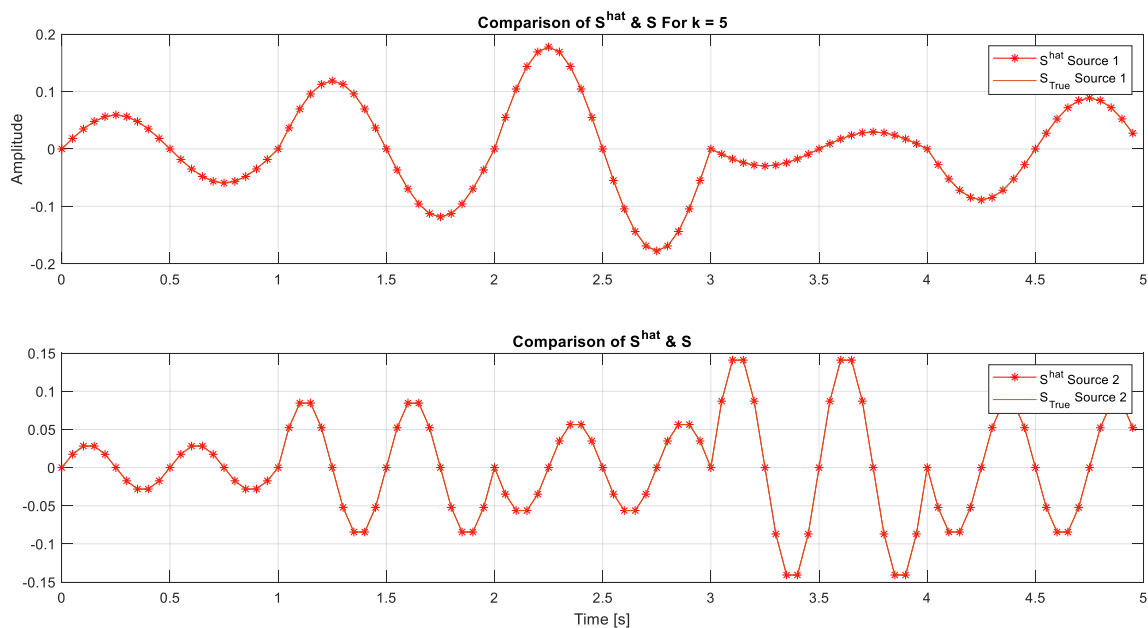
$$B = \begin{bmatrix} b_1^T \\ \vdots \\ b_n^T \end{bmatrix} \quad \downarrow \quad BB^T = I$$

$$f(b_i) = b_i^T \left(\sum_k \sum_{j \neq i} (R_k^{(i)} b_j) (R_k^{(i)} b_j)^T \right) b_i$$

$$\text{s.t. } b_i^T b_i = 1$$

$$f(b_i) = b_i^T R_i b_i, \quad b_i = \arg \min f(b_i)$$

شکل 5



شکل 6

مشاهده می شود که مقدار خطا بسیار کاهش یافته و تقریباً در هر دو کانال به همگرایی رسیده ایم!

مقدار خطای مورد نظر برابر است با:

$$\text{Error} = 3.9999\text{e-}12$$

قسمت-4

(د) حال فرض کنید نویز نیز وارد مساله می شود:

$$Y = X + \sigma W$$

در رابطه ی بالا اسکالر σ انحراف معیار نویز و ماتریس W ماتریس نویز می باشد که هر درایه ی آن از یک متغیر گوسی با میانگین صفر و واریانس یک انتخاب شده است. ماتریس W را پس از تولید نرمالیزه کنید به گونه ای که نرْم فروبینیوس آن یک شود. اسکالر σ را به گونه ای انتخاب کنید که قدرت سیگنال مطلوب به قدرت سیگنال نامطلوب برابر 20 db یا $SNR = \frac{\|X\|_F^2}{\sigma^2} = 100$ باشد. با استفاده از ماتریس Y که ابعاد آن 2×100 است، منابع را تخمین بزنید. کیفیت منابع تخمین شده را با معیار E گزارش کنید.

با اعمال شرایط جدید، سیگنال منابع تخمین زده شده در شکل زیر نمایش داده شده اند. همچنین خطای حاصل برابر:

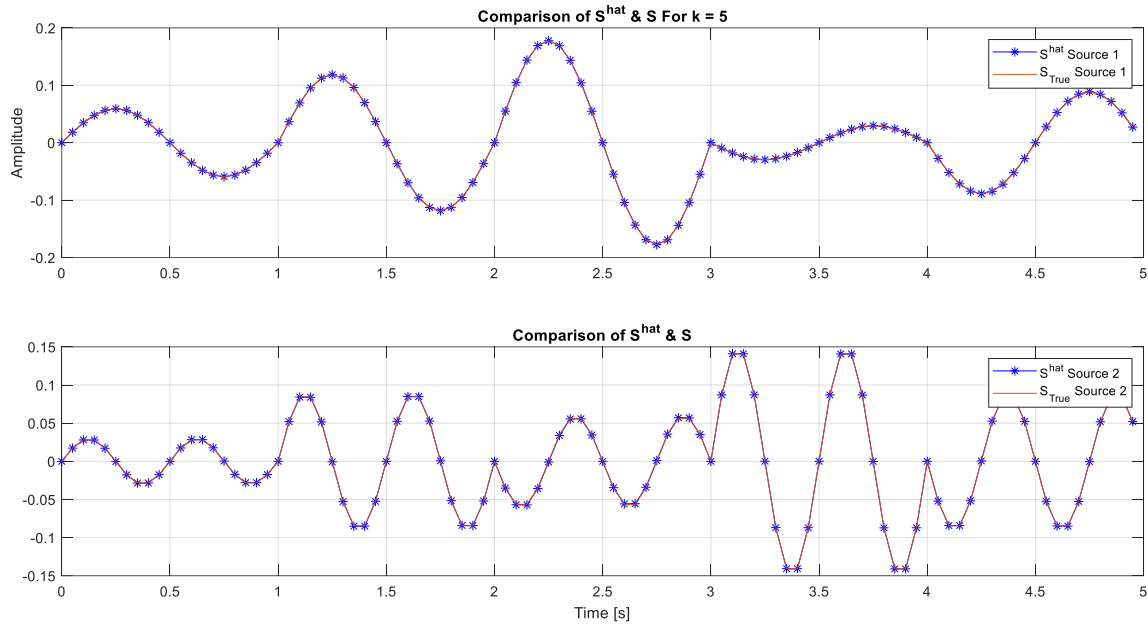
$$\text{Error} = 1.9735\text{e-}05$$

می باشد. ضرایب تقریب زده شده برابراند با:

$$B_Hat_K_Y =$$

$$0.5975 - 0.8018 -$$

$$0.7969 - 0.6042$$



شکل 7

قسمت-5:

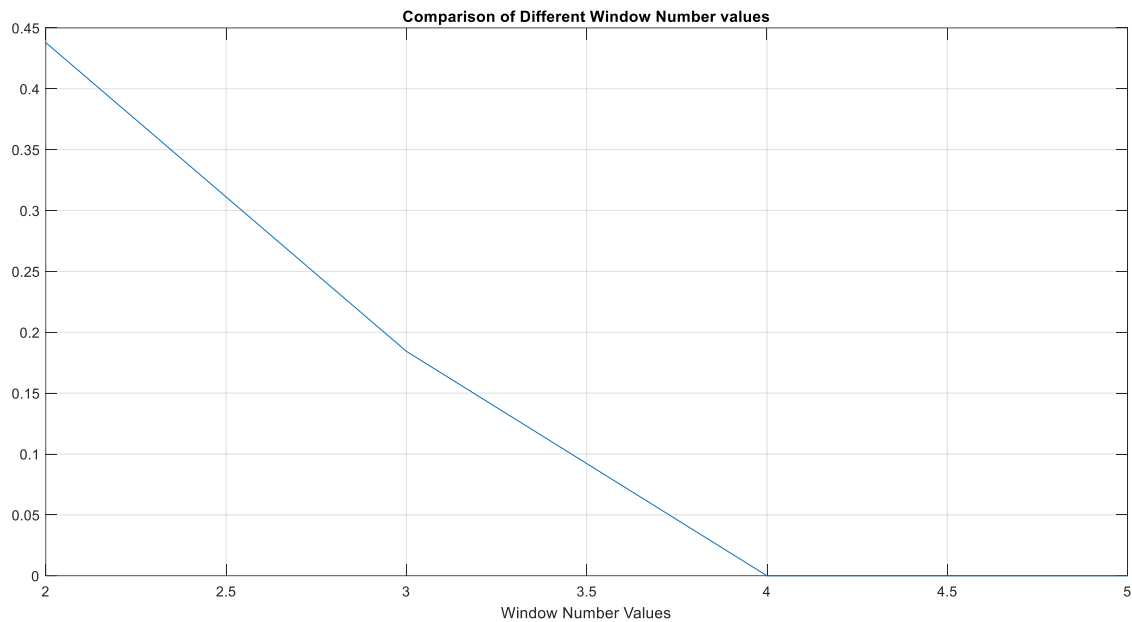
ه) ماتریس Y تولید شده در قسمت د که ابعاد آن 2×100 بود را در نظر بگیرید.

به ازای تعداد پنجره های مختلف ($K = 2, 3, 4, 5$) مساله ی BSS را حل کنید و نمودار میانگین E بر حسب K را رسم کنید. در واقع قسمت د را با فرض $K = 5$ حل کرده بودید.

* برای این که نتایج مستدل تری داشته باشید نویز را ۱۰۰ بار مختلف تولید کنید و هر بار Y را بسازید و سپس روش را به ازای K های مختلف پیاده کنید. در انتها میانگین E را بر حسب K رسم کنید.

مقدار خطای به ازای مقادیر مختلف پنجره ها به صورت زیر در آمده است:

(از 2 پنجره به ترتیب تا 5 پنجره)



شکل ۸

نمودار مقایسه میزان خطای مدل با 100 تکرار به ازای هر مقدار پنجره

% Calculate Error For Different Window Numbers:

```

cntr=1;
NUm_Iters =100;
BSS_Opt_K = cell(NUm_Iters,length(2:K));
Error_Wind_k = zeros(1,length(2:K));

for k=2:K
    for i=1:NUm_Iters
        BSS_Opt_K{cntr,i} = BSS_Solver(X,k,fs,S_Amp);
        Error_Wind_k(cntr)= Error_Wind_k(cntr)+BSS_Opt_K{cntr,i}.Error;
    end
    Error_Wind_k(cntr) = Error_Wind_k(cntr)/NUm_Iters;
    cntr=cntr+1;
end

```

Error_Wind_k =

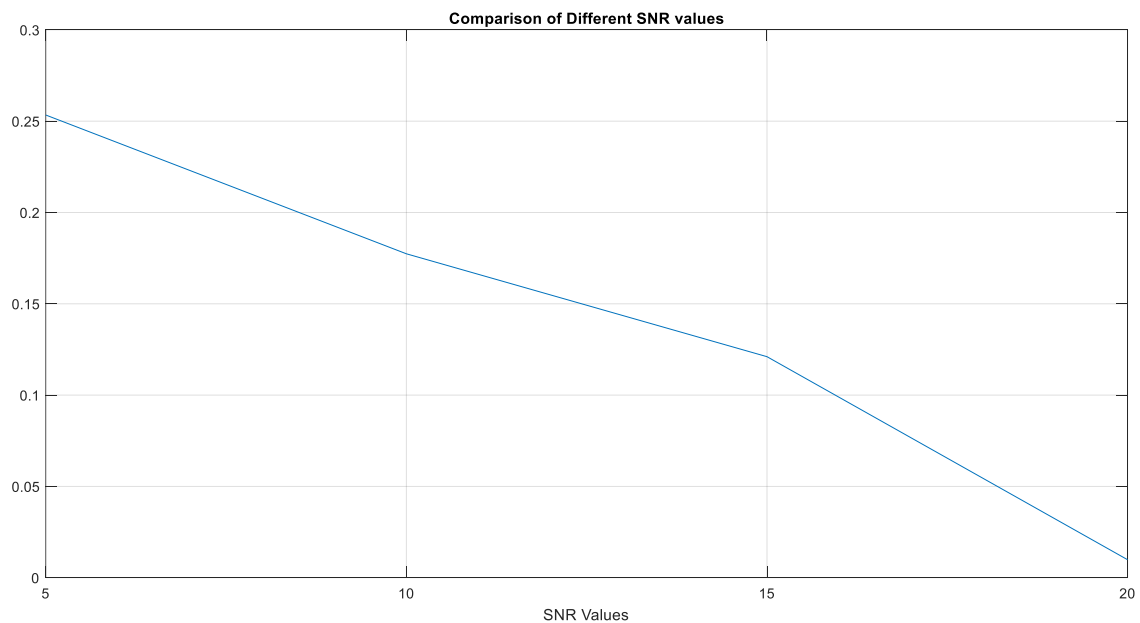
0.4380 0.1844 0.0000 0.0000

قسمت-6

در زیر نمودار خطا بر حسب میزان SNR آورده شده است:

```
% Solve for Different SNR values:
SNR_vec = [5,10,15,20];
NUm_Iters = 100;
BSS_Sols = cell(length(SNR_vec),NUm_Iters);
Window_Num = K;
Error_SNR = zeros(1,length(SNR_vec));

for i=1:length(SNR_vec)
    SNR = SNR_vec(i);
    for iter =1:NUm_Iters
        BSS_Opt_Noisy =BSS_Noisy(X,SNR,Window_Num,fs,S_Amp);
        BSS_Sols{i,iter} = BSS_Opt_Noisy;
        Error_SNR(i) = BSS_Opt_Noisy.Error+Error_SNR(i);
    end
    Error_SNR(i) = Error_SNR(i)/NUm_Iters;
end
```



شکل 9

منطقی است که نمودار خطا بر حسب SNR یک نمودار کاهشی باشد. با افزایش SNR قدرت نویز کاهش یافته و در نتیجه خطای کمتری در Reconstruction خواهیم داشت.

Error_SNR =

0.2534 0.1774 0.1210 0.0099

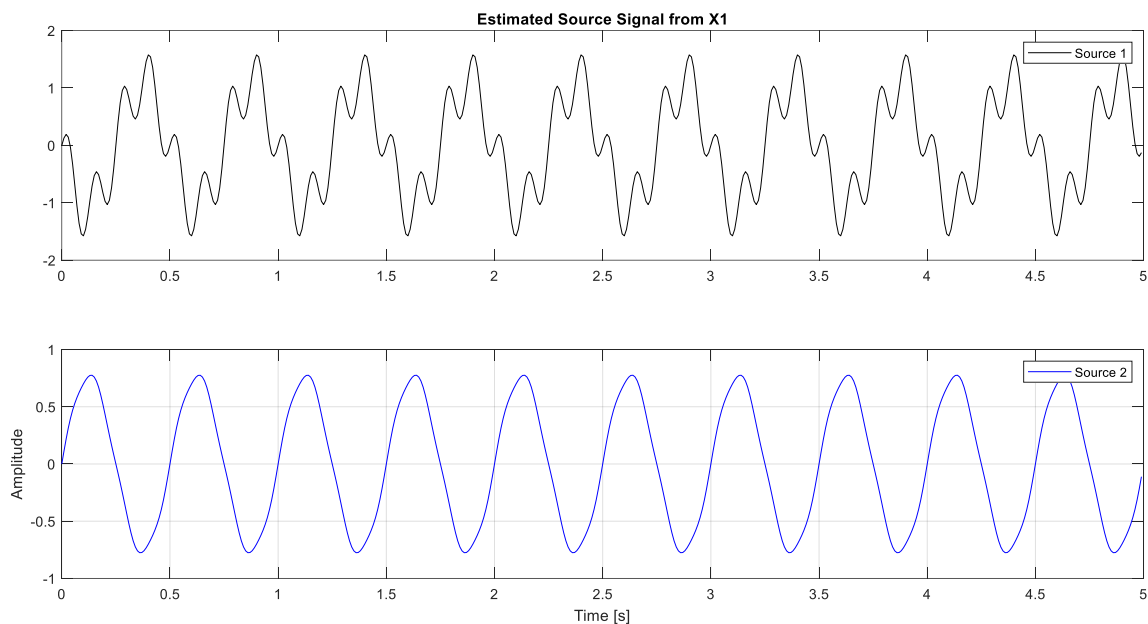
بخش دوم:

ماتریس مشاهدات $X1$ که در اختیار شما قرار گرفته است از ترکیب خطی دو منبع توأم ناهمبسته بدون حضور هر گونه نویزی به وجود آمده است ($f_s = 100 \text{ Hz}$).

قسمت-1:

الف) سیگنال منابع را با استفاده از مشاهدات $X1$ تخمین زده و رسم کنید.

با استفاده از توابع نوشته شده ی قبلی، سیگنال تخمین زده شده برابر است با:



شکل 10

واضح است که شماره ی منبع و اسکیل آن ممکن است دچار ابهام باشد و به دلیل نداشتن مرجع برای مقایسه، نمی توان آن را رفع کرد.

```
X1_Data = load("hw4-X1.mat");
```

```

fs_X1 = 100; %100Hz
ts_X1 = 1/fs_X1;

X1 = X1_Data.X1;
Window_Num = length(X1)/fs_X1;
T_rec_X1 = Window_Num;
t_x1 = 0:ts_X1:T_rec_X1-ts_X1;
S_Amp = randn(size(X1));
BSS_S0l_X1 = BSS_S0lver(X1,Window_Num,fs_X1,S_Amp);
S_hat_X1 = BSS_S0l_X1.B_Hat_Chosen*X1;

```

البته با توجه به اینکه می‌توان این مسئله را با داشتن 2 پنجره به صورت یکتا حل کرد داریم:

در واقع در اینجا برای انتخاب سائز پنجره و محل شروع آن، باید در نظر داشت که لزوماً دو سیگنال سینوسی با 2 فرکانس متفاوت در تمام زمان‌ها ناهمبسته نبوده و بسته به شروع و پایان پنجره می‌توانند همبستگی داشته باشند.

$$\int \sin(2\pi f_1 t) * \sin(2\pi f_2 (t - \tau)) d\tau \text{ from } t1 \text{ to } t2$$

این انتگرال را باید کمینه کنیم، برای این کار داریم:

$$\cos(2a) - \cos(2b) = \sin(a + b) \sin(a - b)$$

$$\frac{1}{2} \int \left(\cos\left(\frac{2\pi(f_1 t + f_2 t - f_2 \tau)}{2}\right) - \cos\left(\frac{2\pi(f_1 t - f_2 t + f_2 \tau)}{2}\right) \right) d\tau$$

$$\frac{1}{2f_2\pi} \left[\sin\left(\frac{2\pi(f_1t + f_2t - f_2\tau)}{2}\right) + \sin\left(\frac{2\pi(f_1t - f_2t + f_2\tau)}{2}\right) \right]_{t_1}^{t_2} =$$

از طرفی داریم:

$$\sin(A) + \sin(B) = 2 \sin\left(\frac{A+B}{2}\right) \cos\left(\frac{A-B}{2}\right)$$

پس عبارت فوق معادل زیر خواهد شد:

$$\frac{2}{2f_2\pi} \left[\sin\left(\frac{2\pi f_1}{2}t\right) \cos\left(\frac{2\pi}{2}(f_2t - f_2\tau)\right) \right]_{t_1}^{t_2}$$

$$\frac{1}{f_2\pi} [\sin(\pi f_1 t) \cos(\pi f_2(t - t_2)) - \sin(\pi f_1 t) \cos(\pi f_2(t - t_1))]$$

$$\cos a - \cos b$$

$$-2 \sin\left(\frac{a+b}{2}\right) \sin\left(\frac{a-b}{2}\right)$$

شکل 11

$$= -2 \frac{\sin(\pi f_1 t)}{f_2\pi} \left[\sin\left(\pi f_2\left(t - \frac{t_2 + t_1}{2}\right)\right) \sin\left(\left(\frac{-t_2 + t_1}{2}\right)\right) \right]$$

پس لزوماً حاصل این انتگرال روی هر بازه‌ی دلخواه برابر 0 نخواهد شد.

شرایط 0 شدن انتگرال فوق:

- اول اینکه آخرین عبارت سینوسی برابر 0 باشد که یعنی همان طول بازه‌ی انتگرال برابر 0 باشد.
- دوم اینکه:

$$\sin(\pi f_1 t) = 0 \rightarrow \pi f_1 t = k\pi \rightarrow t = \frac{k}{f_1} = kT_1;$$

- سوم اینکه:

$$\sin\left(\pi f_2 \left(t - \frac{t_2 + t_1}{2}\right)\right) = 0 \rightarrow \pi f_2 \left(t - \frac{t_2 + t_1}{2}\right) = m\pi \rightarrow$$

$$t = \frac{m}{f_2} + \frac{(t_1 + t_2)}{2} = mT_2 + \frac{(t_1 + t_2)}{2};$$

از طرفی میدانیم که برای 0 شدن انتگرال روی جمع دو سیگنال با فرکانس‌های مختلف، باید طول بازه را ک.م.م. پریودهای دو سیگنال بگیریم. در واقع داریم:

پریود سیگنال اول: T_1

پریود سیگنال دوم: T_2

طول پنجره: $\text{lcm}(T_1, T_2)$

با توجه به دستور فوق، می‌توان طول پنجره را پیدا کرد.
با اینکار، میزان پنجره برابر 0.5 ثانیه به دست خواهد آمد.

```
function BSS_Opt = BSS_Unique_Solver(X,fs,Window_Size)

RX = X*X';
[U_X , Lambda_X] = eig(RX);
% Whietning:
Whitener =(Lambda_X^(-1/2))*U_X.';
% Z=X;
Whitener =1;
Z = Whitener*X;

% RZ = Z*Z';
% [U_Z , Lambda_Z] = eig(RZ);
% Window_Size = length(Z);

RZ_k1 = Z( :, 1:floor(Window_Size) ) *Z(:, 1:floor(Window_Size) )';
RZ_k2 = Z( :, floor(Window_Size)+1:end ) *Z(:, floor(Window_Size)+1:end )';

[Q,Lambda_Z_K] = eig(inv(RZ_k2)*RZ_k1);
B = Q.*Whitener;

BSS_Opt.B = B;
BSS_Opt.Z = Z;
BSS_Opt.X = X;
BSS_Opt.fs = fs;

end
```

و برای حوزه‌ی طیف آن داریم:

```
BSS_Opt_Sol_Unique = BSS_Unique_Solver(X2,fs_X2,length(X2)/2);
```

```

S_hat_X2_Unique = BSS_Opt_S01_Unique.B*X2;

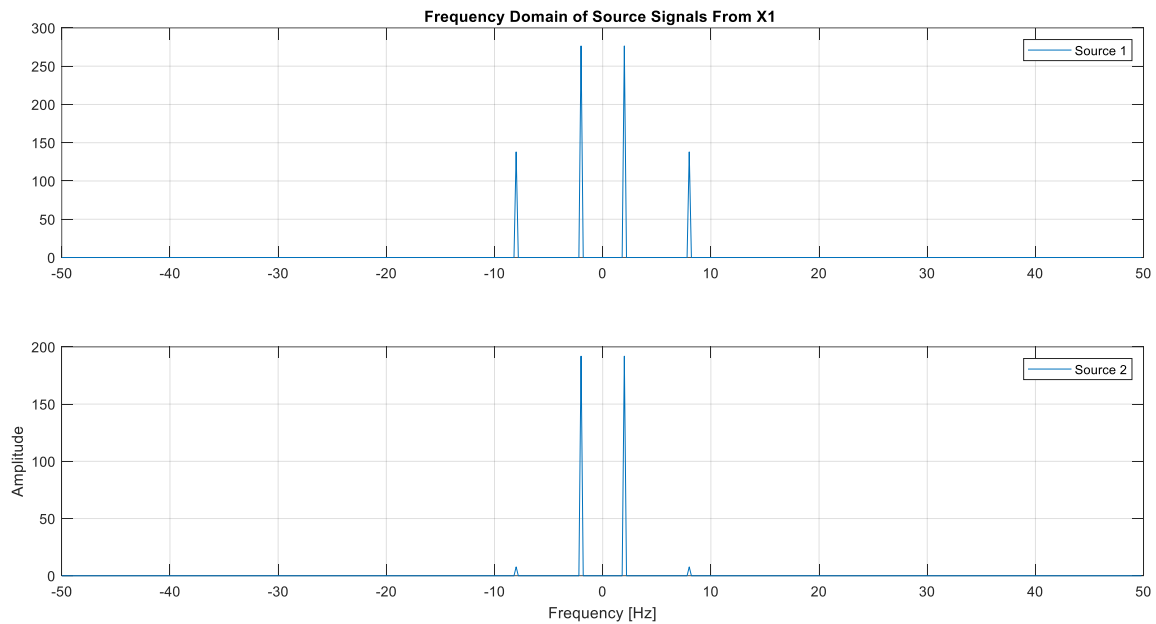
delta_f_X2 = 1/T_rec_X2;
freq_X2 = -fs_X2/2:delta_f_X2:fs_X2/2-delta_f_X2; % Frequency Axis
% Source 1 FFT:
S_hat_X2_F_1 = fftshift(fft(S_hat_X2_Unique(1,:)));
% Source 2 FFT:
S_hat_X2_F_2 = fftshift(fft(S_hat_X2_Unique(2,:)));

figure()
subplot(2,1,1)
plot(freq_X2,abs(S_hat_X2_F_1))
grid on
title("Frequency Domain of Source Signals From X2 ")
legend("Source 1")
subplot(2,1,2)
plot(freq_X2,abs(S_hat_X2_F_2))
grid on
legend("Source 2")
xlabel("Frequency [Hz]")
ylabel("Amplitude")

```

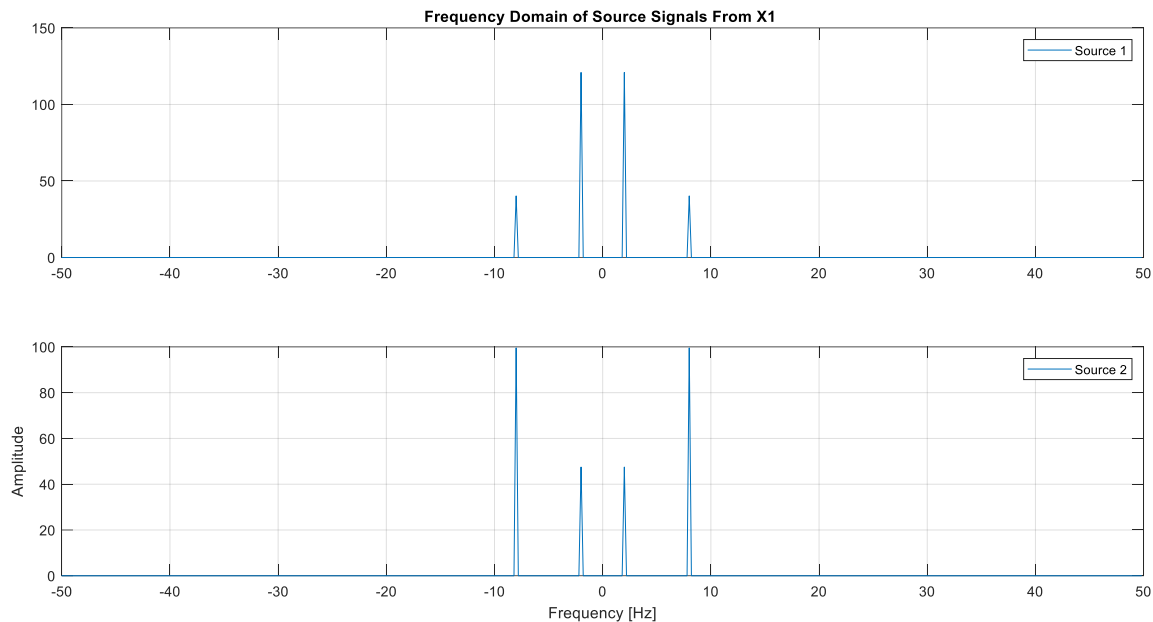
قسمت-2:

ب) تبدیل فوریه ی سیگنال منابع تخمین زده شده را رسم کنید. مشاهده ی خود را گزارش کنید. آیا می توانستیم با اعمال یک فیلترینگ ساده روی مشاهدات، منابع را تخمین بزنیم؟



شکل 12

با داشتن 5 پنجره‌ی زمانی سیگنال منابع در حوزه‌ی فرکانس به صورت بالا در می‌آیند. با این فرمت نمی‌توان دو منبع را از یکدیگر تفکیک کرد چرا که فرکانس مشترک دارند. با در نظر گرفتن سفید کردن و طول پنجره‌ی مناسب، خروجی به فرمت زیر درخواهد آمد:



شکل 13

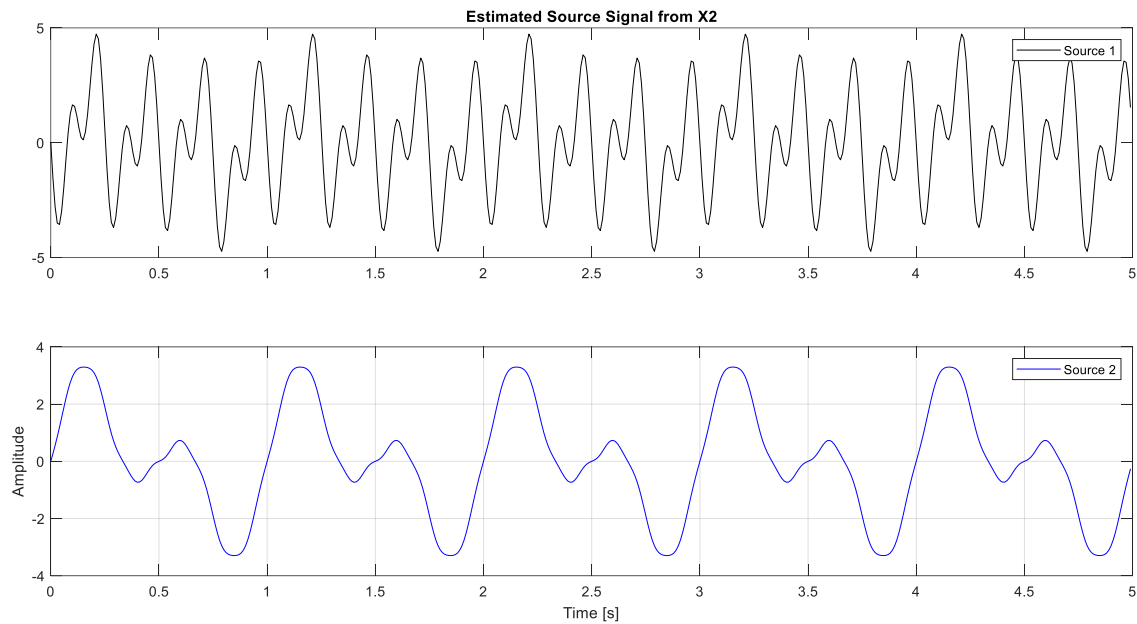
در شکل فوق، دو منبع به صورت نسبی قابل تفکیک در حوزه‌ی فرکانس هستند. چرا که قدرت سیگنال با فرکانس دوم در هر دو طیف، از نصف قدرت دیگری نیز کمتر بوده و می‌توان هر منبع را یک تون سینوسی فرض کرد.

با این فرض می‌توان این منابع را با یک فیلتر بالاگذر یا پایین‌گذر تیز، جداسازی کرد.

قسمت-3:

ماتریس مشاهدات $X2$ که در اختیار شما قرار گرفته است نیز از ترکیب خطی دو منبع توأم ناهمبسته بدون حضور هر گونه نویزی به وجود آمده است ($f_s = 100 \text{ Hz}$).

(ج) سیگنال منابع را با استفاده از مشاهدات $X2$ تخمین زده و رسم کنید.

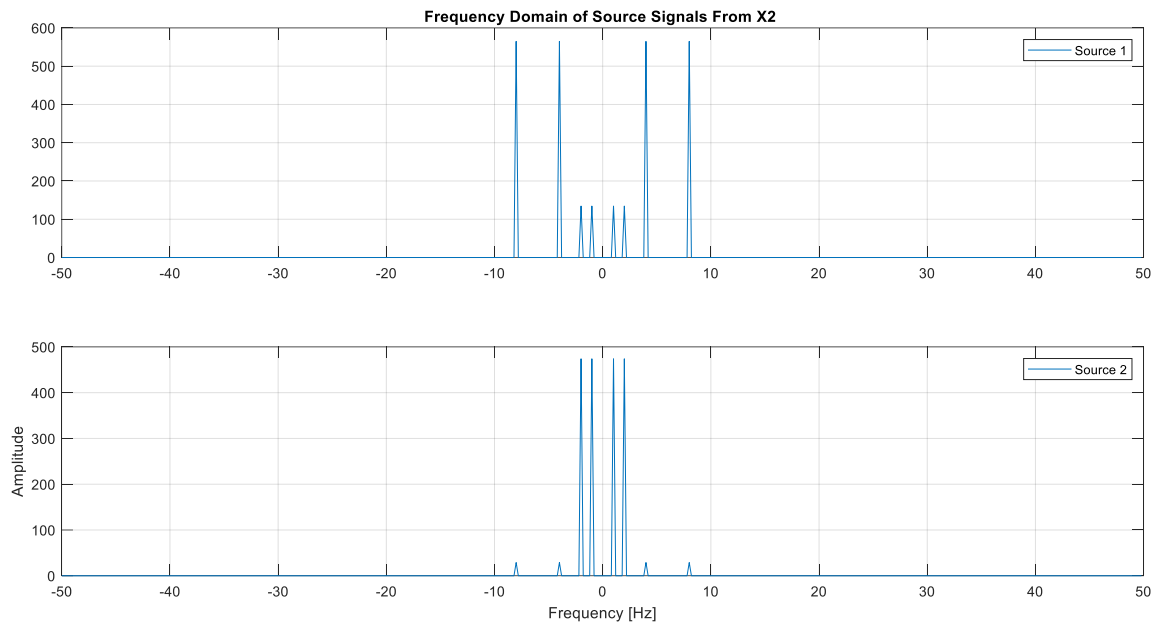


شکل 14

قسمت-4:

د) تبدیل فوریه ی سیگنال منابع تخمین زده شده را رسم کنید. مشاهده ی خود را گزارش کنید. آیا می توانستیم با اعمال یک فیلترینگ ساده روی مشاهدات، منابع را تخمین بزنیم؟ چرا؟

برای سیگنال داده شده ی X_2 نیز همین اتفاق می افتد:



شکل 15

مشاهده می‌شود که دو سیگنا سینوسی هر کدام با 2 فرکانس متفاوت، در این طیف حضور دارند که تقریباً در این بازه‌ی گرفته شده، از یکدیگر ناهم بسته‌اند! پس می‌توان با استفاده از فیلتر فرکانسی تیز، از یکدیگر تفکیک کرد.

پایان