به نام خدا

دانشگاه تهران

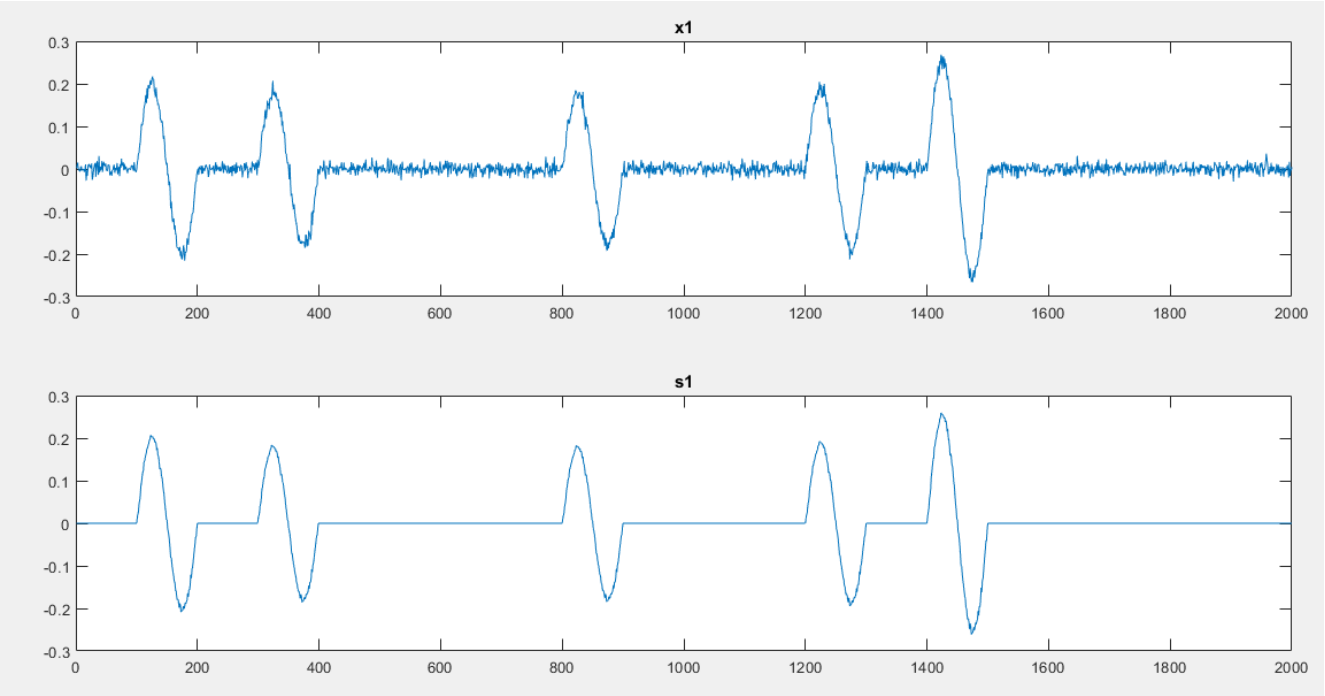
دانشکده مهندسی برق و کامپیوتر

BSS

گزارش تمرین ۷

سالار صفردوست
۸۱۰۱۹۹۴۵۰
۱۴۰۲/۰۳/۰۳

بخش اوّل



در $x1$ دیده می‌شود که در ۵ نقطه سیگنال تک تُن سینوسی آغاز می‌شود و همچنین طول این سیگنال‌ها به طور حدودی برابر ۱۰۰ می‌باشد.

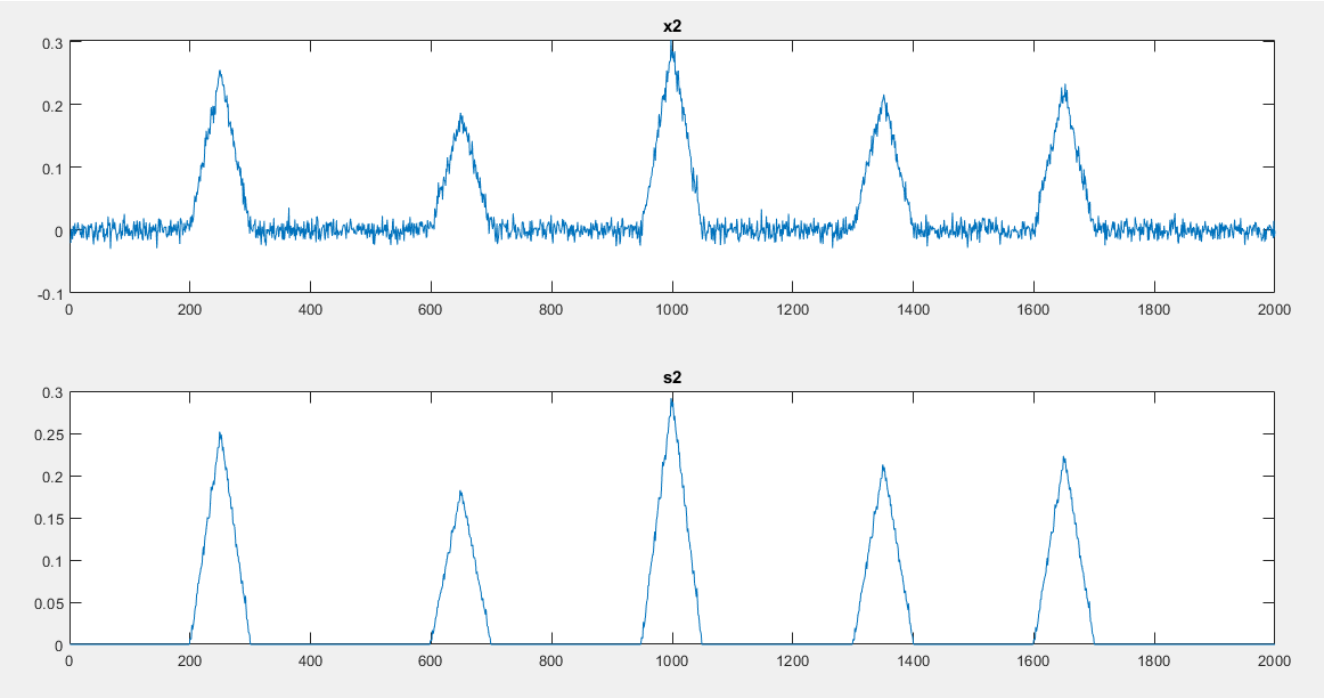
دیده می‌شود که با قطعه کد زیر $s1$ به خوبی بازیابی شده است.

```
%% Question 1

L = 100;
K = 5;
iterations = 20;

[s,alpha,tau] = SingleChannelSBD(x1,L,K,iterations);

s1 = x1*0;
for k = 1:K
    s1(tau(k):tau(k)+L-1) = s*alpha(k);
end

figure
subplot(2,1,1)
plot(x1)
title('x1')
subplot(2,1,2)
plot(s1)
title('s1')
```

```
function [s,alpha,tau] = SingleChannelSBD(x1,L,K,iterations)

    T = size(x1,2);
    alpha = rand(K,1);
    tau = floor(linspace(1,T,K+1));
    tau(end) = [];
    Y = zeros(K,L);

    for i = 1:iterations

        for k = 1:K

            Y(k,:) = x1(tau(k):tau(k)+L-1);
%             subplot(K,1,k)
%             plot(Y(k,:))

        end

        s = alpha.'*Y / (alpha.'*alpha);
        s = s/norm(s);

        [alpha,tau] = AlphaTauUpdate(x1,s,K,L,T);

    end

end
```

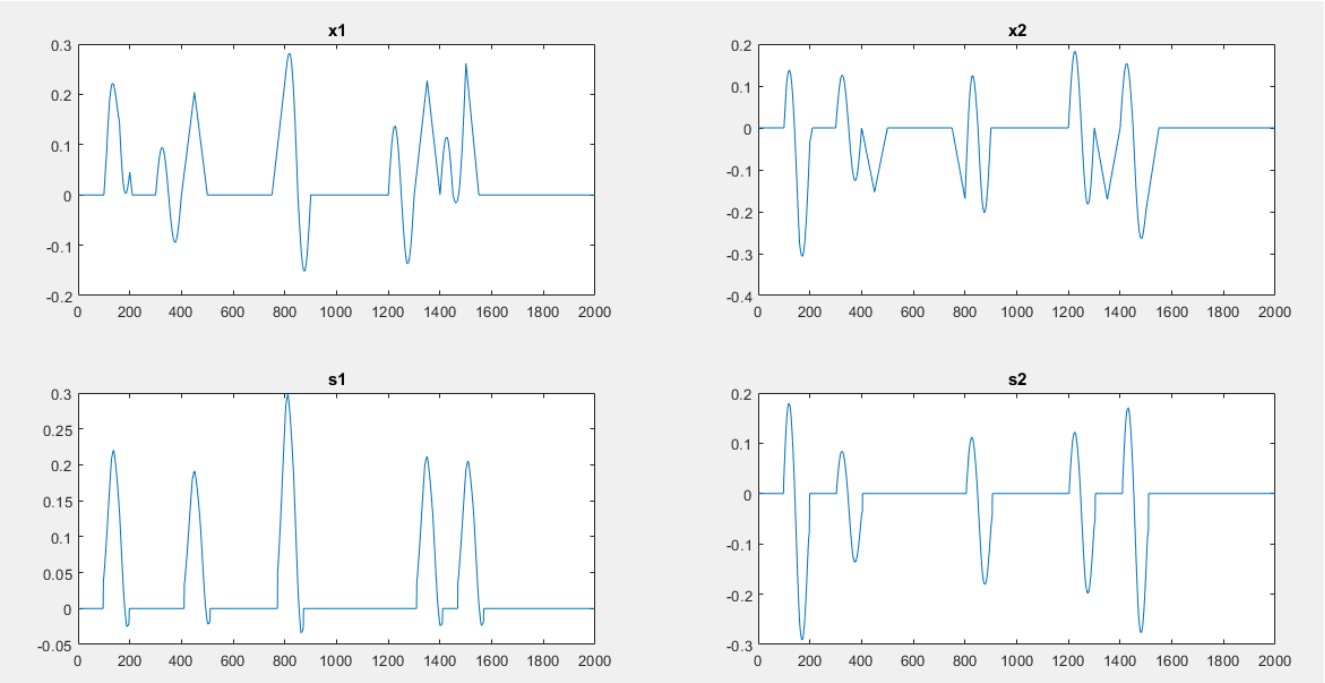
```
function [alpha,tau] = AlphaTauUpdate(x1,s,K,L,T)

    alpha = zeros(K,1);
    tau = alpha;

    [corr,lags] = xcorr(x1,s);
    corr = corr(lags>=0 & lags<=T-L);
%     plot(lags(lags>=0 & lags<=T-L),corr)

    for k = 1:K
        [alpha(k),tau(k)] = max(corr);
        corr(max(1,tau(k)-L+1):min(T-L,tau(k)+L-1)) = 0;
    end

end
```

بخش دوم




```
29      %% Question 2
30
31 -    L = 100;
32 -    K = 5;
33 -    iterations = 20;
34
35 -    [s,alpha,tau] = SingleChannelSBD(x2,L,K,iterations);
36
37 -    s2 = x2*0;
38 -    for k = 1:K
39 -        s2(tau(k):tau(k)+L-1) = s*alpha(k);
40 -    end
41
42 -    figure
43 -    subplot(2,1,1)
44 -    plot(x2)
45 -    title('x2')
46 -    subplot(2,1,2)
47 -    plot(s2)
48 -    title('s2')
49
```

بخش سوم



در X به نظر می‌رسد که دو نوع سیگنال سینوسی تک تن و سیگنال مثلثی شرکت داشته‌اند که حضور آن‌ها ۵ بار اتفاق افتاده است.

```
%% Question 3

L = 100;
K = 5;
N = 2;
iterations = 50;

[A,S] = MultiChannelSBD(X,L,K,N,iterations);

figure
subplot(2,2,1)
plot(X(1,:))
title('x1')
subplot(2,2,2)
plot(X(2,:))
title('x2')
subplot(2,2,3)
plot(S(1,:))
title('s1')
subplot(2,2,4)
plot(S(2,:))
title('s2')

```

```
function [A,S] = MultiChannelSBD(X,L,K,N,iterations)

    [M,T] = size(X);
    A = normc(2*rand(M,N)-1);

    for i = 1:iterations
        S = pinv(A)*X;
        for  n = 1:N
            [s,alpha,tau] = SingleChannelSBD(X(n,:),L,K,iterations);
            S(n,:) = zeros(1,T);
            for k = 1:K
                S(n,tau(k):tau(k)+L-1) = s*alpha(k);
            end
        end
        A = normc(X*pinv(S));
    end
end
```

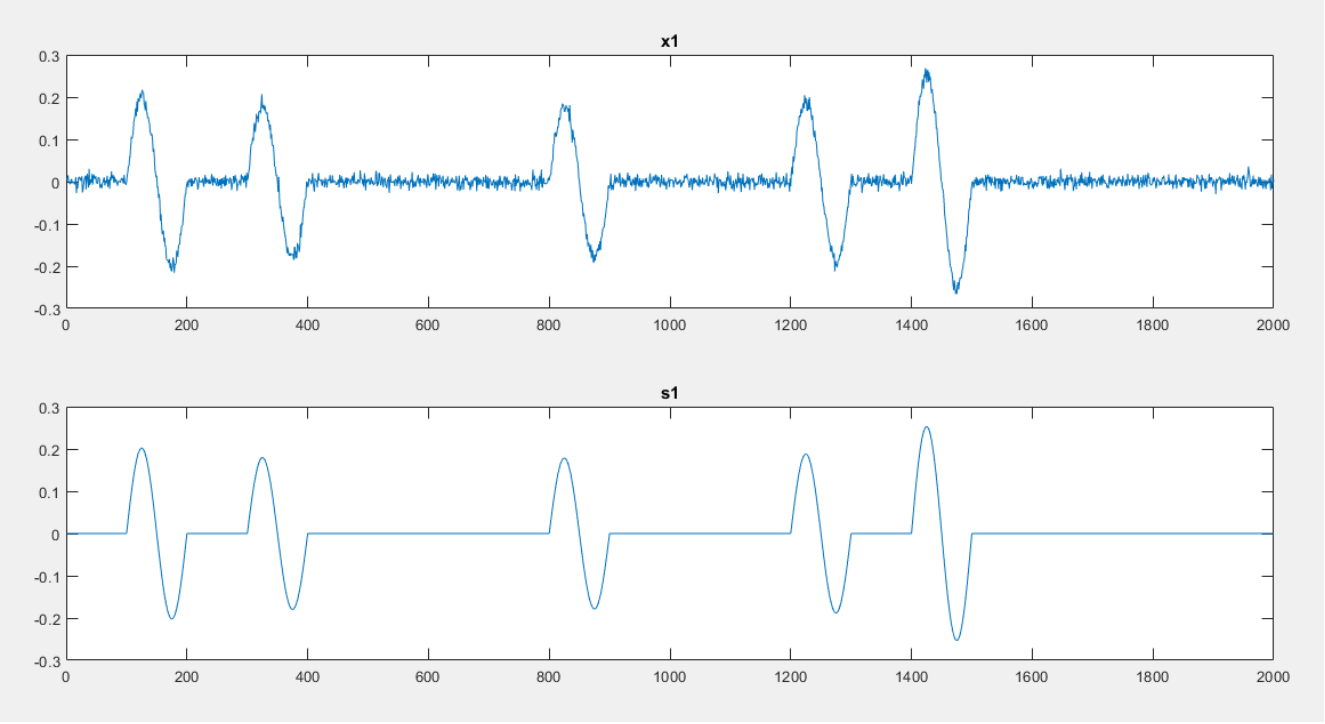
بخش چهارم

روش مورد استفاده برای به دس آوردن $s1$ در این بخش اینگونه است که فرض می‌شود می‌دانیم که سیگنال تکرار شونده از نوع سینوسی می‌باشد و تنها فرکانس، دامنه و محل‌های وقوع آن مجهول است.

بنابراین ابتدا با پیدا کردن ماکسیمم اندازه‌ی تبدیل فوریه‌ی سیگنال، فرکانس مربوط به سینوسی را می‌یابیم(مثبت و منفی سینوس با مثبت و منفی بودن مقدار تبدیل فوریه تصحیح می‌شود.)، سپس تابع سینوسی در یک پنجره‌ی L ساخته می‌شود و حالا که سیگنال تکرار شونده مشخص شده است، آن را برای تشخیص ضرایب و تأخیرها به تابع $AlphaTauUpdate$ می‌فرستیم.

```
%% Question 4

[s,alpha,tau] = SingleTuneExtractor(x1,L,K);

s1 = x1*0;
for k = 1:K
    s1(tau(k):tau(k)+L-1) = s*alpha(k);
end

figure
subplot(2,1,1)
plot(x1)
title('x1')
subplot(2,1,2)
plot(s1)
title('s1')
```

```
function [s,alpha,tau]=SingleTuneExtractor(x1,L,K)

    len = length(x1);
    x1_hat = fftshift(fft(x1))/len;
    f = (-len/2 : len/2-1)/len;

%     figure
%     plot(f,abs(x1_hat));

    i = find(abs(x1_hat) == max(abs(x1_hat)));
    t = 1:L;

    A = imag(x1_hat(i(1)));
    s = A*sin(2*pi*f(max(i)).*t);
    s = s/norm(s);

    [alpha,tau] = AlphaTauUpdate(x1,s,K,L,length(x1));
end
```


x1
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0
200
400
600
800
1000
1200
1400
1600
1800
2000
s1
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0
200
400
600
800
1000
1200
1400
1600
1800
2000