

تمرین شماره 7

علیرضا حسینی

شماره دانشجویی : ۸۱۰۱۰۱۱۴۲

جداسازی کور منابع

دکتر اخوان

بهار 1402

فهرست مطالب

4: 1 سوال ۱-۱
10: 2 سوال ۱-۲
12: 3 سوال ۱-۳
13: 4 سوال ۱-۴

فهرست اشکال

4 شکل (۱-۱) کد متلب لود دیتا ها و ویژوال کردن x1
4 شکل (۲-۱) داده نویزی تک کاناله x1
5 شکل (۳-۱) نقاط زمانی شروع سیگنال
5 شکل (۴-۱) مقدار دهی اولیه سوال 1
6 شکل (۵-۱) مقدار دهی اولیه رخداد ها
7 شکل (۶-۱) کد متلب بازسازی سیگنال و محاسبه خطا
8 شکل (۷-۱) بررسی همگرایی و رسم شکل موج ها
9 Reconstructed signal – Q1 شکل (۸-۱)
9 Estimated Waveform – Q1 شکل (۹-۱)
10 شکل (۱۰-۱) داده نویزی تک کاناله x2
11 Reconstructed signl – x2 شکل (۱۱-۱)
11 Esrimated Waveform – x2 شکل (۱۲-۱)
13 شکل (۱۳-۱) تصاویر 2 کانال و سیگنال های Reconstructed شده
14 شکل (۱۴-۱) اجرای single channel sparse blind deconv در حوزه فرکانس
14 شکل (۱۵-۱) بازسازی سیگنال ساخته شده و تبدیل آن به حوزه زمان به ifft
15 شکل (۱۶-۱) سیگنال اصلی و بازسازی شده با deconv در حوزه فرکانس

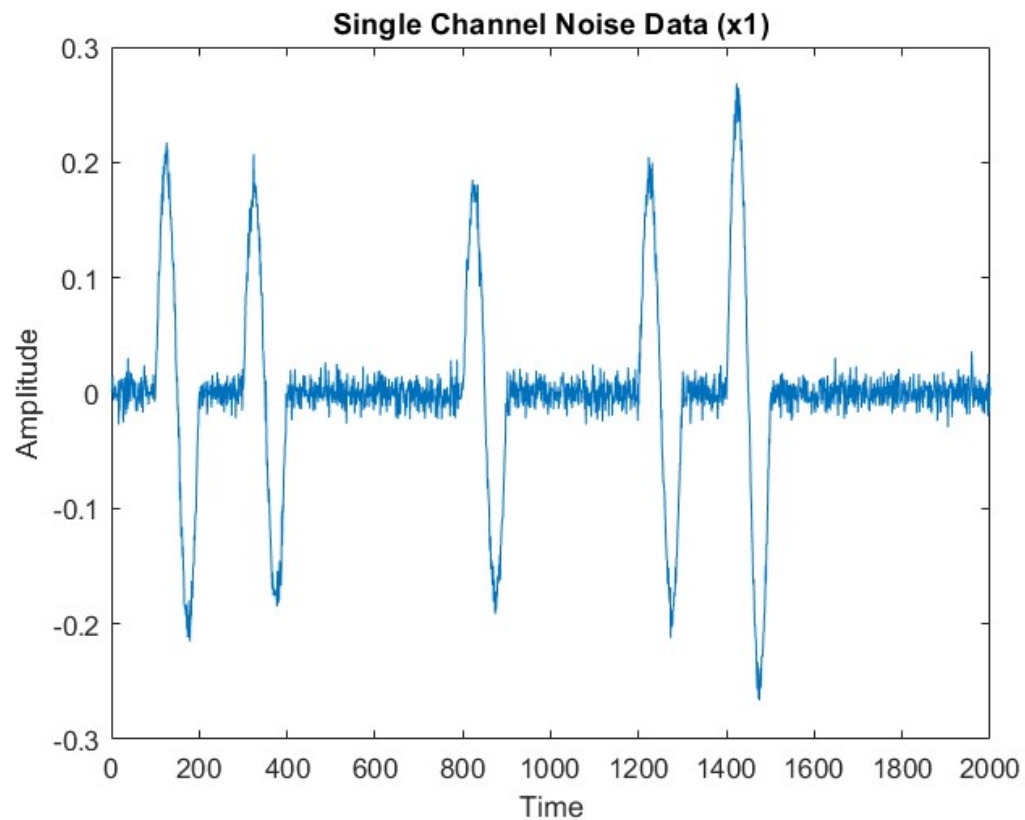
۱-۱- سوال ۱:

به کمک دستور زیر داده ها را لود کرده و ویژوال میکنیم.

```
% Load the data from hw7.mat
data = load('hw7.mat');
x1 = data.x1;

% Visualize the single channel noise data x1
figure;
plot(x1);
title('Single Channel Noise Data (x1)');
xlabel('Time');
ylabel('Amplitude');
```

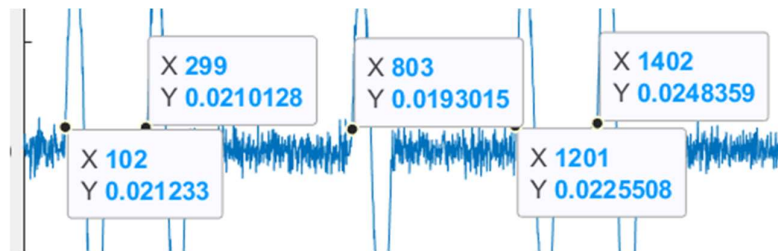
شکل (۱-۱) کد متلب لود دیتا ها و ویژوال کردن x1



شکل (۱-۲) داده نویزی تک کاناله x1

مشاهده میشود 5 تا اسپایک داشته و سیگنال ها از نقاط زیر شروع شده است.

102 و 299 و 803 و 1201 و 1402



شکل (۳-۱) نقاط زمانی شروع سیگنال

Single channel sparse deconvolution یک تکنیک پردازش سیگنال است که برای بازیابی سیگنال ورودی پراکنده و ناشناخته از شکل پیچیده آن با یک سیستم یا کانال ناشناخته استفاده می شود. این تکنیک فرض می کند که سیگنال ورودی اسپارس است، به این معنی که تنها دارای چند جزء مهم است، و هدف آن تخمین سیگنال اسپارس است.

در ادامه ما کد را به قسمت های مختلف تقسیم می کنیم و برای هر بخش توضیح می دهیم.

مقداردهی اولیه داده ها:

کد با مقداردهی اولیه متغیرها و پارامترهای لازم شروع می شود. K نشان دهنده تعداد رخداد ها یا اجزای

مهم در سیگنال، L نشان دهنده طول شکل موج، و T طول سیگنال ورودی است.

```
K = 5; % Number of events
L = 100; % Length of waveform
T = length(x1); % Length of input signal
```

شکل (۴-۱) مقداردهی اولیه سوال 1

مقدار دهی رخداد:

نقاط شروع رخداد ها در آرایه t_w تعریف شده است. این نقاط مکان هایی را در سیگنال نشان می دهد که رخداد ها از آنجا شروع می شوند. آرایه t_w_last با افزودن L به هر نقطه شروع که نشان دهنده آخرین نقاط رویدادها است مقداردهی اولیه می شود. آرایه آلفا مقادیر آلفای اولیه را برای هر رخداد نشان می دهد.

```
t_w = [102, 299, 803, 1201, 1402]; % Starting points of events
t_w_last = t_w + L; % Last points of events
alpha = [1, 1, 1, 1, 2]; % Initial alpha values
e = 10; % Initialization for error
iteration = 1;
```

شکل (۵-۱) مقدار دهی اولیه رخداد ها

دکانولوشن:

کد برای تخمین سیگنال پراکنده و شکل موج وارد یک حلقه می شود. حلقه تا رسیدن به همگرایی ادامه می یابد که با شرط $t_w == t_w_last$ تعیین می شود. در هر تکرار:

آ. تخمین شکل موج:

حلقه یک ماتریس y را با انتخاب نمونه های L از سیگنال ورودی بر اساس نقاط شروع رویداد فعلی (t_w) می سازد.

تخمین شکل موج \hat{s} با ضرب y در شبه معکوس ماتریس آلفا به دست می آید.

شکل موج \hat{s} برای داشتن نرم واحد نرمال شده است.

ب به روز رسانی موقعیت های رخداد ها:

یک ماتریس Z با انتخاب نمونه های L از سیگنال ورودی برای همه موقعیت های ممکن ایجاد می شود.

حلقه بر روی هر رخداد تکرار می شود و مراحل زیر را انجام می دهد:

پیش بینی های \hat{s} بر روی Z را با استفاده از ضرب ماتریس محاسبه میشود.

شاخص I را پیدا کرده که با حداکثر مقدار مطلق نمایش مطابقت دارد.

موقعیت رویداد t_w را با I به روز می‌کنیم.

مقدار آلفا را برای رخداد بر اساس مقدار پیش بینی شده در شاخص I به روز می‌کنیم.

بخش مربوط به Z_{reduced} را روی صفر تنظیم کنید تا مشارکت‌ها از رخداد شناسایی شده حذف شوند.

ج بازسازی محاسبه سیگنال و خطا:

یک سیگنال پراکنده s_i با اختصاص مقادیر آلفا به موقعیت‌های مربوطه در سیگنال ایجاد می‌شود.

خطای خطا به عنوان فاصله اقلیدسی بین سیگنال ورودی اصلی x_1 و s_i با s_{hat} محاسبه می‌شود.

```
while true
    % Update waveform estimation (s_hat) given fixed events
    for k = 1:K
        y(:, k) = x1(t_w(k):t_w(k) + L - 1);
    end
    s_hat = y * pinv(alpha);
    s_hat = s_hat / norm(s_hat);

    % Update events (t_w) given fixed waveform estimation
    for j = 1:T - L + 1
        Z(:, j) = x1(j:j + L - 1);
    end
    Z_reduced = Z;
    for k = 1:K
        projections = s_hat' * Z_reduced;
        [~, I] = max(abs(projections));
        alpha_pro = projections(I);
        t_w(k) = I;
        alpha(k) = alpha_pro;
        Z_size = Z_reduced;
        Z_reduced(:, max(1, I - L + 1):min(size(Z_size, 2), I + L - 1)) = 0;
    end

    % Calculate the reconstructed signal and error
    si = zeros(1, T);
    for q = 1:length(t_w)
        si(t_w(q) + L/2) = alpha(q);
    end
    error = norm(x1 - conv(si, s_hat, 'same'));
```

شکل (۶-۱) کد متلب بازسازی سیگنال و محاسبه خطا

د. بررسی همگرایی:

کد بررسی می کند که آیا موقعیت های رخداد فعلی t_w با موقعیت های رخداد قبلی t_{w_last} یکسان است یا خیر.

اگر موقعیت رویداد تغییر نکرده باشد، حلقه شکسته می شود که نشان دهنده همگرایی است.

ترسیم سیگنال بازسازی شده:

پس از فرآیند دکانولوشن، کد سیگنال ورودی اصلی را رسم می کند و سیگنال بازسازی شده به دست آمده

از انحراف سیگنال پراکنده si را با شکل موج تخمینی s_hat پوشش می دهد.

رسم شکل موج تخمینی:

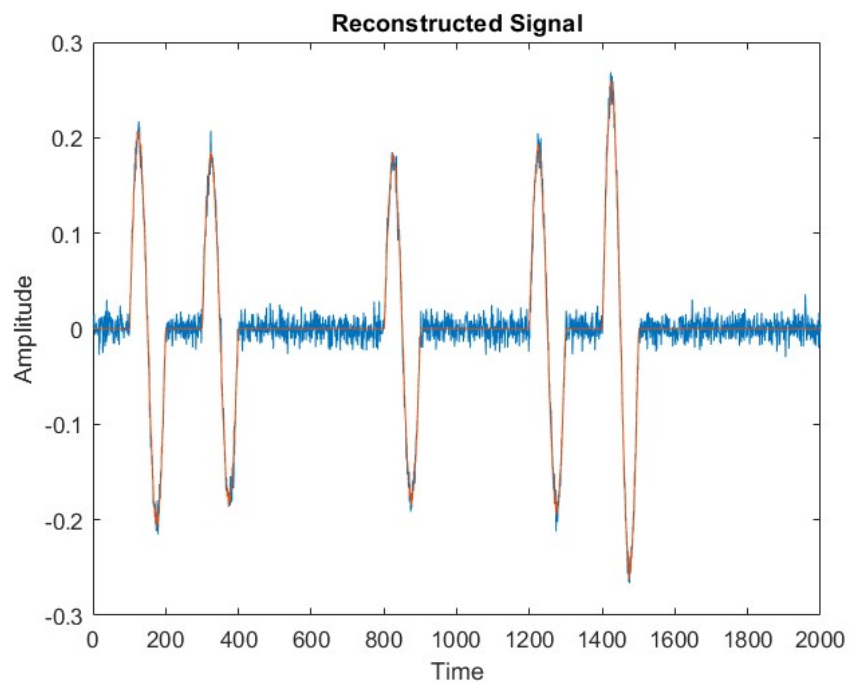
در نهایت، کد شکل موج تخمینی s_hat را ترسیم می کند.

```
% Check convergence
if t_w == t_w_last
    break;
end
t_w_last = t_w;
iteration = iteration + 1;
end

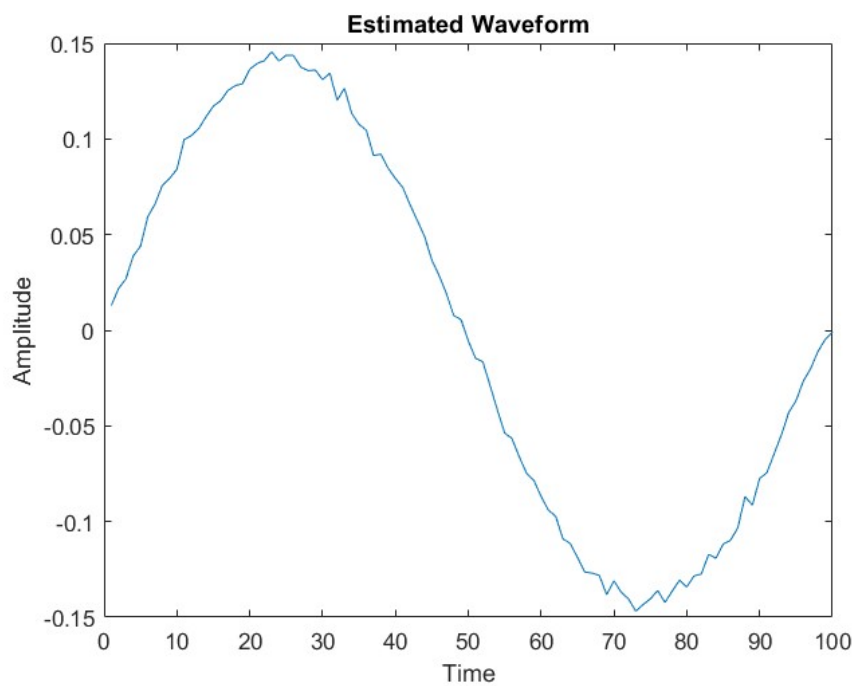
%% Plot the reconstructed signal
figure;
plot(x1);
hold on;
plot(conv(si, s_hat, 'same'));
title('Reconstructed Signal');
xlabel('Time');
ylabel('Amplitude');

%% Plot the estimated waveform
figure;
plot(s_hat);
title('Estimated Waveform');
xlabel('Time');
ylabel('Amplitude');
```

شکل (۷-۱) بررسی همگرایی و رسم شکل موج ها



Reconstructed signal – Q1 شکل (۱-۸)

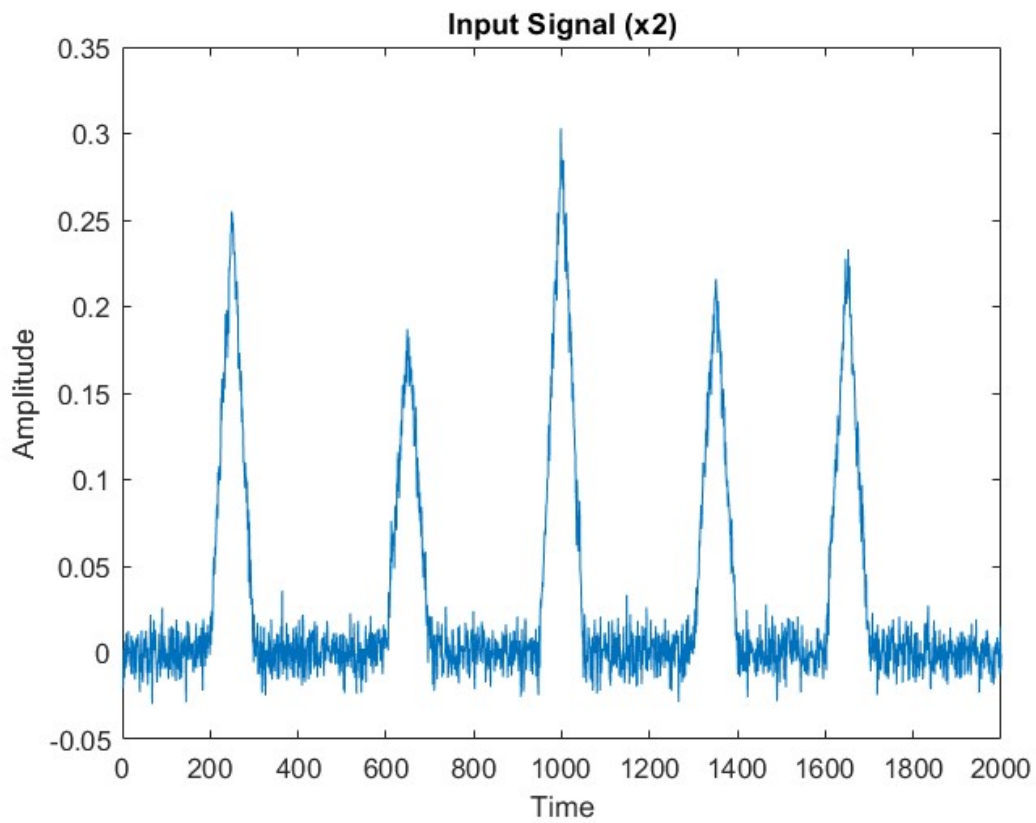


Estimated Waveform – Q1 شکل (۱-۹)

همگرایی در 3 تا iteration رخ داده است.

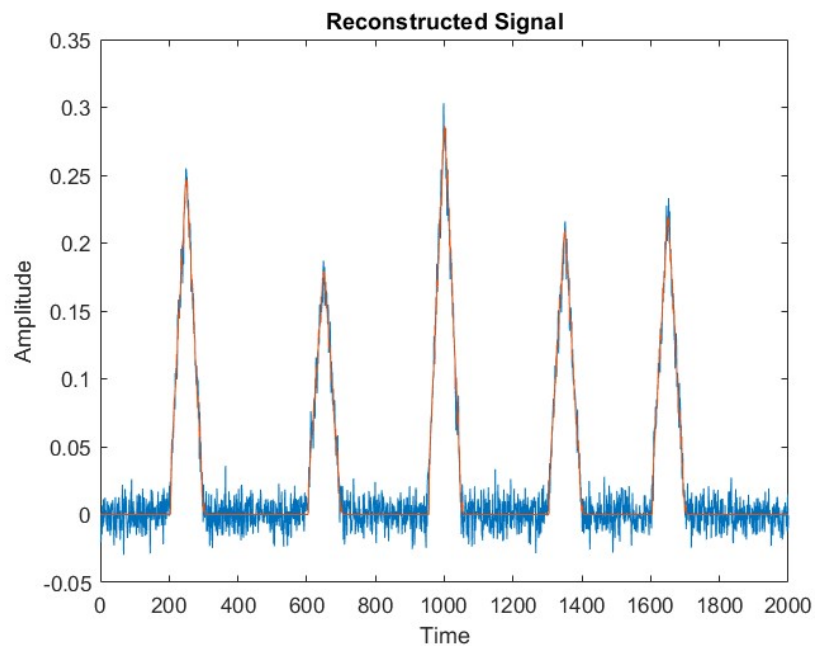
۱-۲- سوال 2:

داده نویزی تک کاناله x_2 به صورت زیر است:

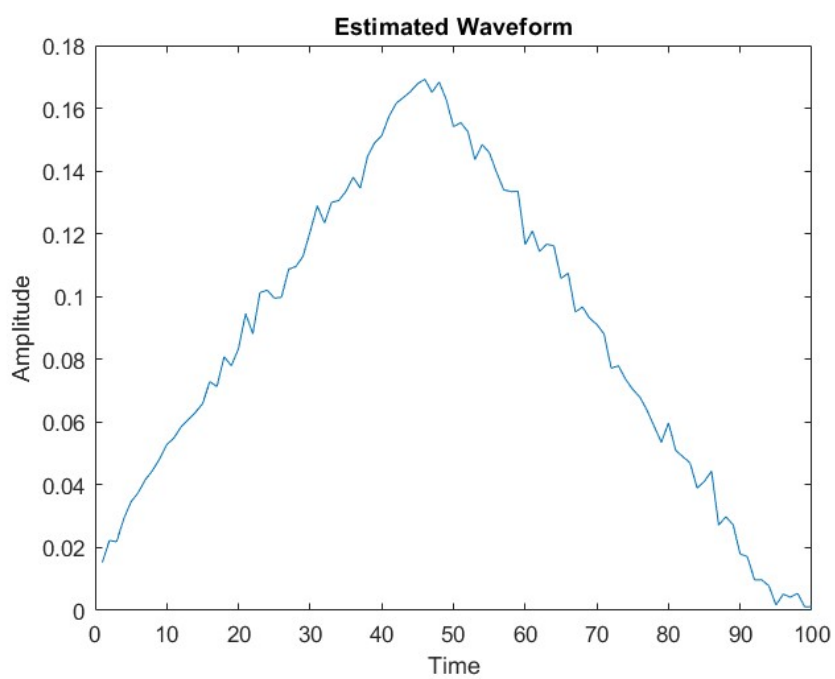


شکل (۱۰-۱) داده نویزی تک کاناله x_2

نقاط شروع سیگنال 201 و 608 و 951 و 1308 و 1608 میباشد.



Reconstructed signal – x2 شکل (۱-۱۱)



Esimated Waveform – x2 شکل (۱-۱۲)

۳-۱- سوال 3:

همانند قبل تقریباً عمل میکنیم. مراحل پیاده سازی به صورت زیر میباشد.

برآورد منبع برای کانال 1:

اولین کانال x_1 را از سیگنال ورودی X استخراج می کند.

تاخیرهای زمانی اولیه taw_1 ، تاخیرهای زمانی قبلی taw_1_last و ضرایب اختلاط α_1 را تعریف میکنیم.

یک حلقه داخلی برای تخمین منابع و به روز رسانی تأخیرهای زمانی و ضرایب راه اندازی می کند.

در حلقه داخلی، منابع s_1_hat با استفاده از وقفه ها و ضرایب زمانی فعلی برآورد می شوند.

تاخیرهای زمانی taw_1 و ضرایب آلفا 1 با استفاده از الگوریتم STCT به روز می شوند.

حلقه داخلی تا زمانی که معیارهای همگرایی برآورده شود ادامه می یابد.

برآورد منبع برای کانال 2:

مشابه مرحله قبل، این بخش تخمین منبع و STCT را برای کانال دوم x_2 انجام می دهد.

ساختن برآورد منبع ترکیبی S_hat :

منابع تخمینی S_1_hat و S_2_hat را با اطلاعات STCT متناظر Si_1 و Si_2 خود ترکیب می کند.

تابع $conv$ برای ادغام هر تخمین منبع با اطلاعات STCT متناظر آن استفاده می شود.

به روز رسانی $Mixing Matrix A$:

ماتریس A را با استفاده از منابع تخمینی S_hat و سیگنال ورودی X به روز می کند.

هر ستون A نرمال می شود تا نرم واحد داشته باشد.

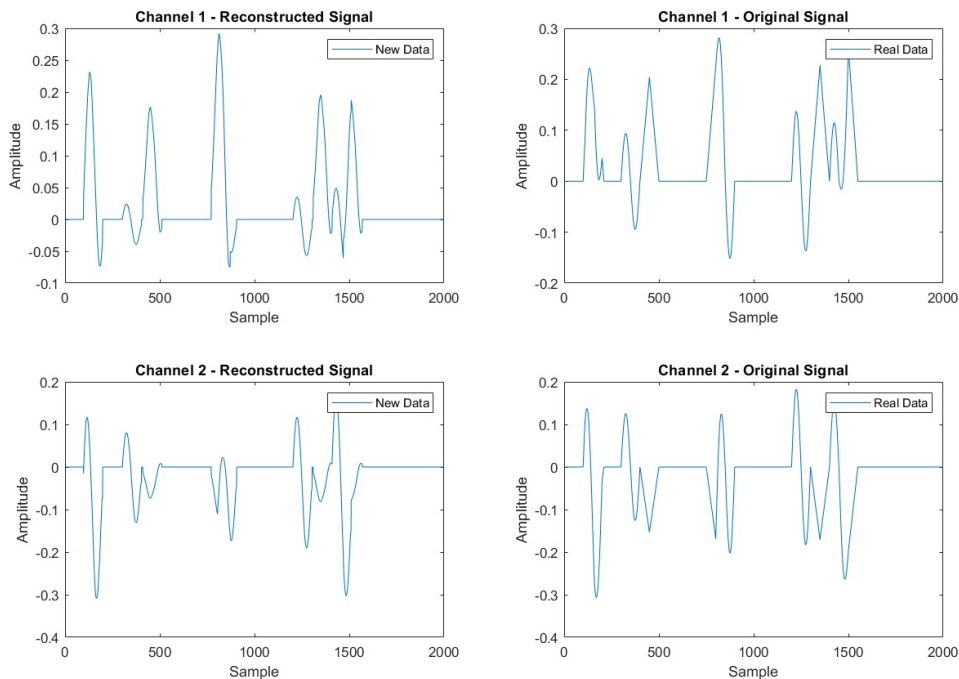
بررسی همگرایی و خاتمه:

خطای بازسازی e را با مقایسه سیگنال ورودی X و سیگنال بازسازی شده $\hat{S} * A$ محاسبه می کند.

بررسی می کند که آیا خطای e بر اساس تغییر از دو تکرار گذشته همگرا شده است یا خیر.

اگر معیارهای همگرایی برآورده شوند، حلقه تکراری خاتمه می یابد.

****** کد طولانی بود تصاویر در گزارش نیاورده است ولی در پیوست ارسال شده است.



شکل (۱-۱۳) تصاویر ۲ کانال و سیگنال های Reconstructed شده

در این حالت هم همگرا شد ولی تفاوت با حالت های قبلی این است که در تعداد iteration بیشتری همگرایی

صورت میپذیرد.

۴-۱- سوال 4:

در این بخش پس از تعریف مساله و لود دیتا به صورت زیر عمل میکنیم.

1- به کمک دستور fft سیگنال را به فضای فرکانس میریم.

2- در حوزه فرکانس همان مراحل قبل را تکرار میکنیم.

```
% Perform single-channel sparse blind deconvolution in the frequency domain
K = 5; % Number of sources/components
L = 100; % Length of the segments
T = length(x1); % Total number of samples

taw = starting_points;
taw_last = taw;
alpha = ones(1, K);

i = 1;
while i < 100
    % W is fixed
    Y = zeros(L, K);
    for k = 1:K
        Y(:, k) = X1(taw(k):taw(k) + L - 1);
    end

    S_hat = Y * pinv(alpha);
    S_hat = S_hat / norm(S_hat);

    % S is fixed
    Z = zeros(L, T - L + 1);
    for j = 1:T - L + 1
        Z(:, j) = X1(j:j + L - 1);
    end

    Z_reduced = Z;
    for k = 1:K
        projections = S_hat' * Z_reduced;
        [~, index] = max(abs(projections));
        alpha_pro = projections(index);
        taw(k) = index;
        alpha(k) = alpha_pro;
        Z_size = Z_reduced;
        Z_reduced(:, max(1, index - L + 1):min(size(Z_size, 2), index + L - 1)) = 0;
    end

    if isequal(taw, taw_last)
        break;
    end

    taw_last = taw;
    i = i + 1;
end
```

شکل (۱-۱۴) اجرای single channel sparse blind deconv در حوزه فرکانس

3- سیگنال بازسازی شده را در حوزه زمان بازسازی میکنیم.

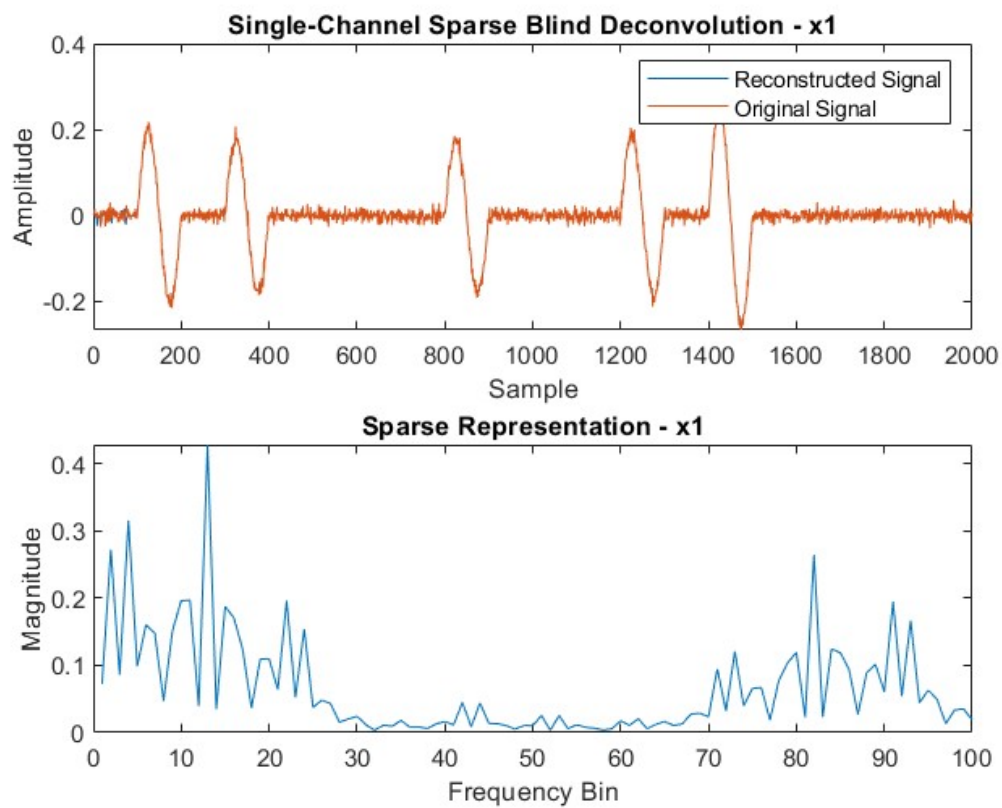
```
% Reconstruct the signal in the time domain
Si = zeros(1, T);
for q = 1:length(taw)
    Si(taw(q) + L / 2) = alpha(q);
end

S1_hat = S_hat;
Si1 = Si;

% Apply inverse Fourier Transform to obtain the reconstructed signal in the time domain
x1_reconstructed = ifft(Si1_hat);
```

شکل (۱-۱۵) بازسازی سیگنال ساخته شده و تبدیل آن به حوزه زمان به ifft

لازم به ذکر است با ترشولد های پایین همگرایی رخ نداد و خروجی در نهایت به صورت زیر می باشد که نشان از این دارد که به خوبی نتوانسته بازسازی صورت بگیرد.



شکل (۱۶-۱) سیگنال اصلی و بازسازی شده با deconv در حوزه فرکانس