

سوال ۱: با توجه به مطالب کلاس، الگوریتم ALS را به صورت زیر پیاده‌سازی می‌کنیم.

```
function U = CP_ALS(T, U1_0, U2_0, U3_0)
    U1 = U1_0;
    U2 = U2_0;
    U3 = U3_0;
    number_of_iterations = 100;
    [I1, I2, I3] = size(T);
    T1 = zeros(I1, I2*I3);
    for i3 = 1:I3
        for i2 = 1:I2
            T1(:, (i3-1)*I2+i2) = T(:, i2, i3);
        end
    end
    T2 = zeros(I2, I1*I3);
    for i3 = 1:I3
        for i1 = 1:I1
            T2(:, (i3-1)*I1+i1) = T(i1, :, i3);
        end
    end
    T3 = zeros(I3, I1*I2);
    for i2 = 1:I2
        for i1 = 1:I1
            T3(:, (i2-1)*I1+i1) = T(i1, i2, :);
        end
    end
    for i = 1:number_of_iterations
        U1 = T1*pinv(kr(U3, U2)');
        U2 = T2*pinv(kr(U3, U1)');
        U3 = T3*pinv(kr(U2, U1)');
    end
    U = {U1, U2, U3};
end
```

برای بررسی درستی الگوریتم پیاده‌سازی شده، ابتدا عملکرد آن را به ازای یک تانسور تصادفی با ابعاد  $I_1 = 4$ ,  $I_2 = 3$  و  $I_3 = 2$ ، که با استفاده از دو تانسور رتبه-۱ ساخته شده است، ارزیابی می‌کنیم. ورودی به صورت زیر بدست می‌آید:

$T(:, :, 1) =$

-0.3121	-0.0149	0.3083
-0.4903	-0.0068	0.3324
2.1409	-0.0058	-1.1275
0.2672	0.0969	-1.0331

$T(:, :, 2) =$

-0.0393	-0.1331	1.2389
-0.0089	-0.0777	0.7150
-0.0739	0.0586	-0.4950
0.3011	0.7799	-7.2974

حال با استفاده از الگوریتم مذکور مولفه‌های تانسور ورودی را بدست می‌آوریم.

مولفه‌های سازنده تانسور به صورت زیر بدست می‌آیند:

```
U1 =
    0.4237    0.4417
    0.2469    0.7035
   -0.1847   -3.0921
   -2.4836   -0.3297

U2 =
    0.1159   -0.5311
    0.2901    0.0032
   -2.7170    0.2631

U3 =
   -0.1364    1.3020
   -1.0822   -0.0591
```

سپس بر اساس مولفه‌های خروجی الگوریتم، تانسور ورودی را بازسازی می‌کنیم. خروجی به صورت زیر بدست می‌آید:

```
recovered_T(:, :, 1) =
   -0.3121   -0.0149    0.3083
   -0.4903   -0.0068    0.3324
    2.1409   -0.0058   -1.1275
    0.2672    0.0969   -1.0331
```

```
recovered_T(:, :, 2) =
   -0.0393   -0.1331    1.2389
   -0.0089   -0.0777    0.7150
   -0.0739    0.0586   -0.4950
    0.3011    0.7799   -7.2974
```

$$\|T - \text{recovered\_T}\|_F = 4.6883 * 10^{-15}$$

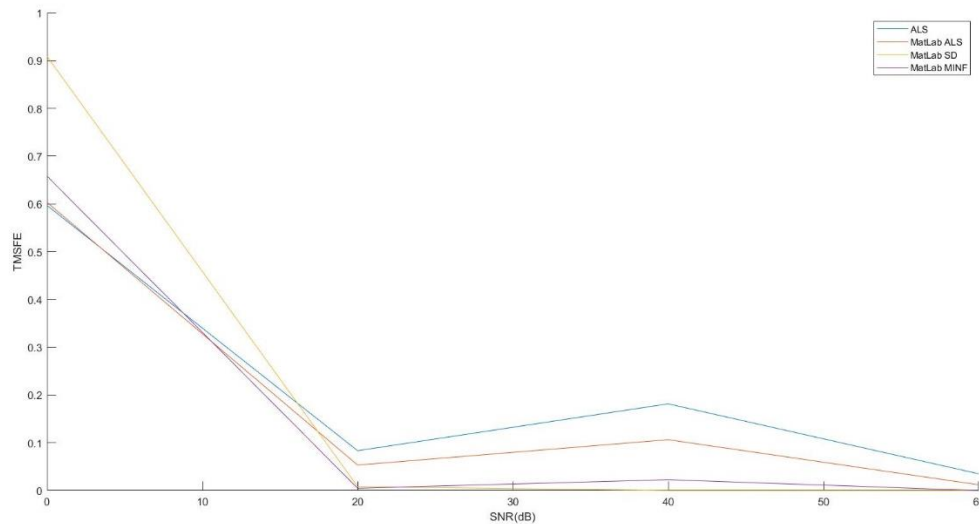
به وضوح الگوریتم پیاده‌سازی شده، عملکرد مناسبی داشته است و توانسته است، مولفه‌های سازنده تانسور را به درستی بدست آورد.

برای اطمینان از درستی الگوریتم، فرآیند فوق را به ازای چند تانسور تصادفی مختلف، تکرار کردیم؛ نتایج مشابهی حاصل شد و الگوریتم توانست در همه موارد، عملکرد دقیق داشته باشد.

سوال ۲: الف) موارد خواسته شده را به ترتیب انجام می‌دهیم. یعنی، ۵۰ بار آزمایش را تکرار می‌کنیم و در هر تکرار، ابتدا به ازای SNR های مختلف تانسور  $\text{noisy\_T}$  را با اضافه کردن تانسور  $N$  به تانسور  $T$  با ضریب مناسب  $\alpha$  تشکیل می‌دهیم و سپس با اجرای چهار الگوریتم مذکور در صورت سوال و محاسبه خطا برای هر یک، از طریق معیار TMSFE، عملکرد آن‌ها را مقایسه می‌کنیم.

خروجی به صورت زیر بدست می‌آید:

```
ave =
    0.5968    0.6031    0.9090    0.6580
    0.0834    0.0533    0.0083    0.0047
    0.1814    0.1064    0.0001    0.0226
    0.0350    0.0120    0.0000    0.0000
```



به طور واضح به ازای SNR های پایین عملکرد روش های مبتنی بر ALS بهتر است؛ اما با بالا رفتن مقدار SNR روش های SD و MINF نسبت به روش های مبتنی بر ALS، به مقادیر کوچک تری از خطا دست می یابند. همچنین در SNR های بالا، روش SD نسبت به روش MINF، نسبت به تکرار آزمایش مقاوم تر است و در همه موارد، عملکرد خوبی دارد؛ اما روش MINF در درصد کمی از موارد با مشکل مواجه می شود و به همین خاطر میانگین خطای بیشتری دارد.

(ب) با توجه به مطالب کلاس، الگوریتم HOSVD را به صورت زیر پیاده سازی می کنیم.

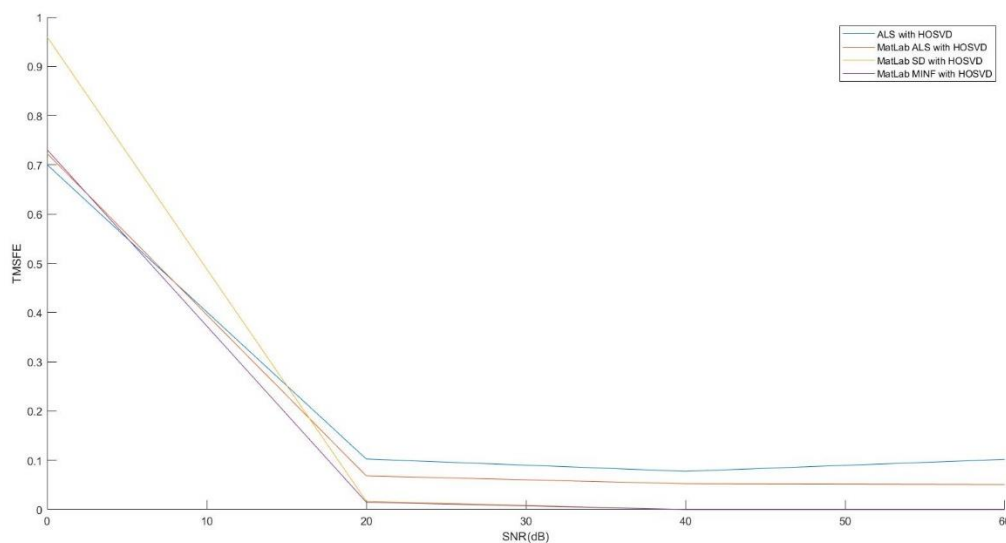
```
function U = HOSVD(T, R)
    [I1, I2, I3] = size(T);
    T1 = zeros(I1, I2*I3);
    for i3 = 1:I3
        for i2 = 1:I2
            T1(:, (i3-1)*I3+i2) = T(:, i2, i3);
        end
    end
    [U1, ~, ~] = svd(T1);
    U1 = U1(:, 1:R);
    T2 = zeros(I2, I1*I3);
    for i3 = 1:I3
        for i1 = 1:I1
            T2(:, (i3-1)*I3+i1) = T(i1, :, i3);
        end
    end
    [U2, ~, ~] = svd(T2);
    U2 = U2(:, 1:R);
    T3 = zeros(I3, I1*I2);
    for i2 = 1:I2
        for i1 = 1:I1
            T3(:, (i2-1)*I2+i1) = T(i1, i2, :);
        end
    end
    [U3, ~, ~] = svd(T3);
    U3 = U3(:, 1:R);
    U = {U1, U2, U3};
end
```

تغییرات خواسته شده را انجام می‌دهیم. یعنی، برای تعیین ماتریس‌های عامل اولیه، از تجزیه HOSVD استفاده می‌کنیم.

خروجی به صورت زیر بدست می‌آید:

ave =

0.7003	0.7231	0.9600	0.7307
0.1026	0.0683	0.0169	0.0149
0.0776	0.0524	0.0001	0.0001
0.1019	0.0509	0.0000	0.0000



در اثر استفاده از تجزیه HOSVD برای تعیین ماتریس‌های اولیه، دو تفاوت مشهود است.

(۱) نوسان عملکرد روش‌های مبتنی بر ALS به ازای SNR های مختلف، کاهش یافته است و رفتاری تقریباً یکنوا کاهشی دیده می‌شود.

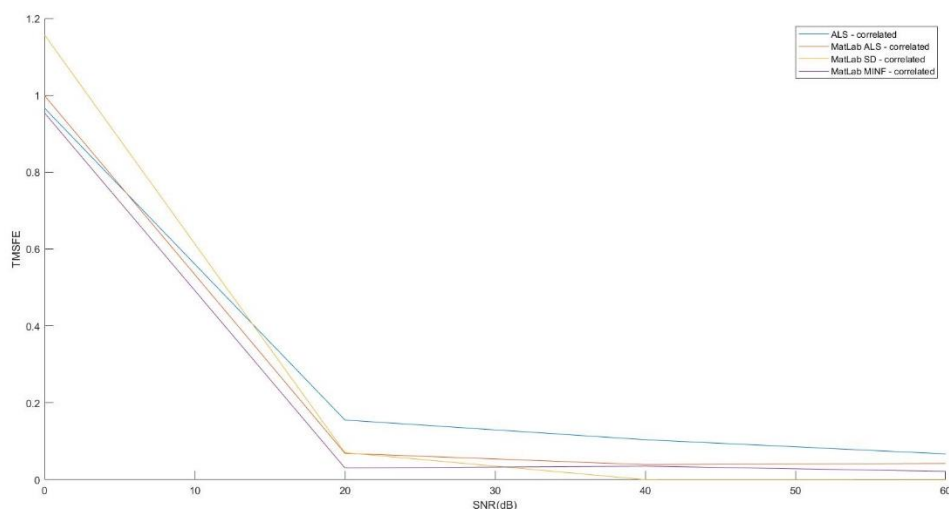
(۲) در SNR های بالا، مقاومت روش MINF نسبت به تکرار آزمایش، افزایش یافته است. در نتیجه توانسته است، در همه موارد، عملکرد خوبی داشته باشد و به میانگین خطای کمتری نسبت به روش SD، دست یابد.

(ج) موارد خواسته شده را به ترتیب انجام می‌دهیم. یعنی، در استخراج ماتریس‌های عامل  $U_{org}^{(1)}$  و  $U_{org}^{(2)}$ ، بین ستون‌های اول و دوم هر یک، مطابق روش مذکور در صورت سوال، همبستگی ایجاد می‌کنیم. بقیه موارد را مشابه با قسمت‌های قبلی پیاده‌سازی می‌کنیم.

خروجی به صورت زیر بدست می‌آید:

ave =

0.9662	0.9986	1.1567	0.9536
0.1553	0.0687	0.0704	0.0306
0.1041	0.0392	0.0003	0.0351
0.0671	0.0424	0.0000	0.0219



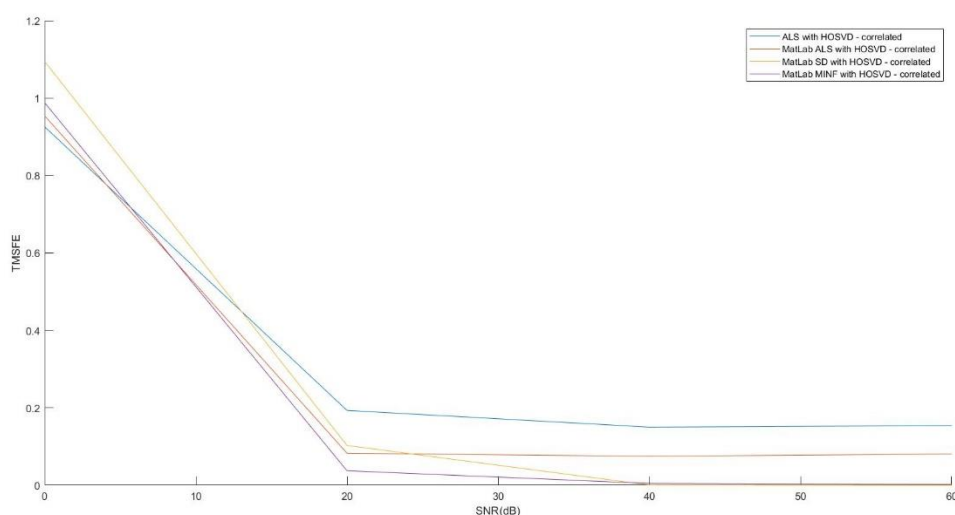
به وضوح میانگین خطا، تقریباً در همه موارد افزایش یافته است. همچنین مشکل عدم مقاومت روش MINF نسبت به تکرار آزمایش، در مقایسه با قسمت‌های قبل، افزایش یافته است و در درصد بیشتری از موارد با مشکل مواجه شده است، اما روش SD همچنان عملکرد خوب خود را حفظ کرده است.

(د) تغییرات خواسته شده را انجام می‌دهیم. یعنی، برای تعیین ماتریس‌های عامل اولیه، از تجزیه HOSVD استفاده می‌کنیم.

خروجی به صورت زیر بدست می‌آید:

ave =

0.9258	0.9539	1.0945	0.9878
0.1933	0.0828	0.1028	0.0375
0.1502	0.0749	0.0007	0.0048
0.1542	0.0809	0.0001	0.0023



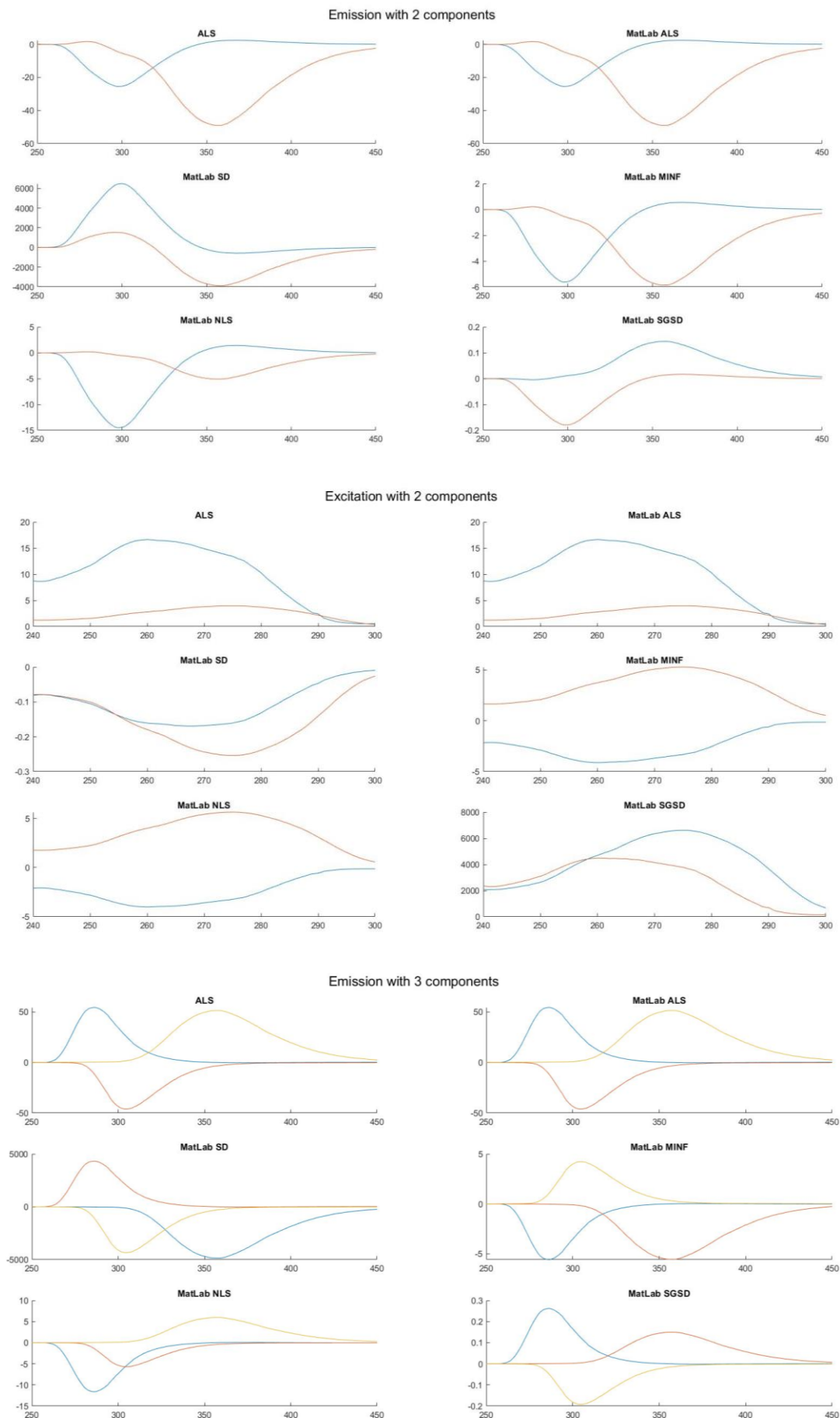
در اثر استفاده از تجزیه HOSVD برای تعیین ماتریس‌های اولیه، دو تفاوت مشهود است.

(۱) میانگین خطای روش‌های مبتنی بر ALS تقریباً به ازای همه SNR های ممکن، افزایش یافته است.

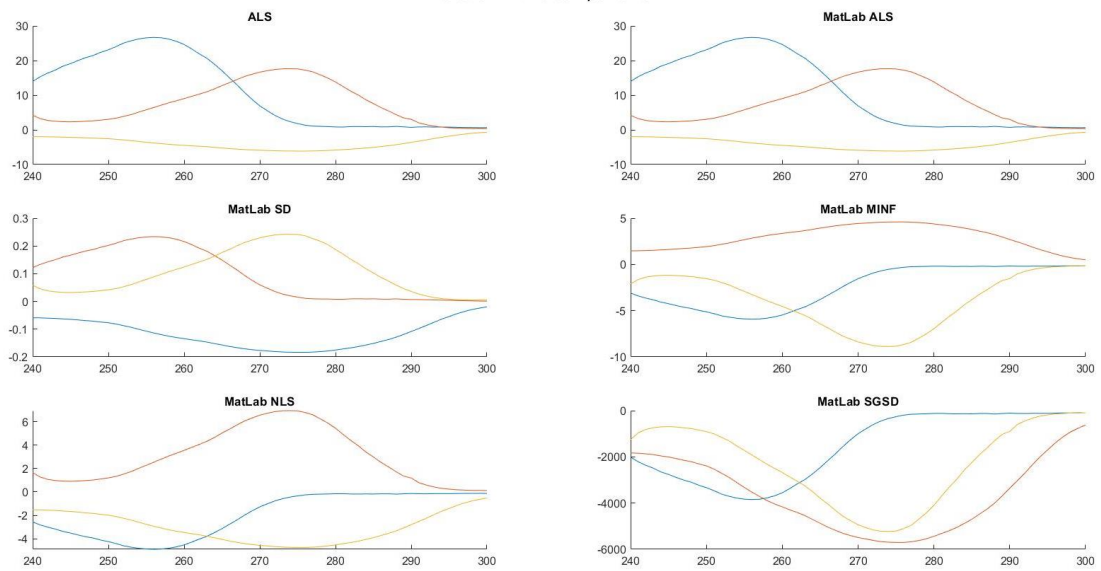
(۲) در SNR های بالا، مقاومت روش MINF نسبت به تکرار آزمایش، افزایش یافته است.

سوال ۳: علاوه بر روش‌های مذکور در سوال ۲، به دلخواه، روش‌های NLS و SGSD را نیز بررسی می‌کنیم.

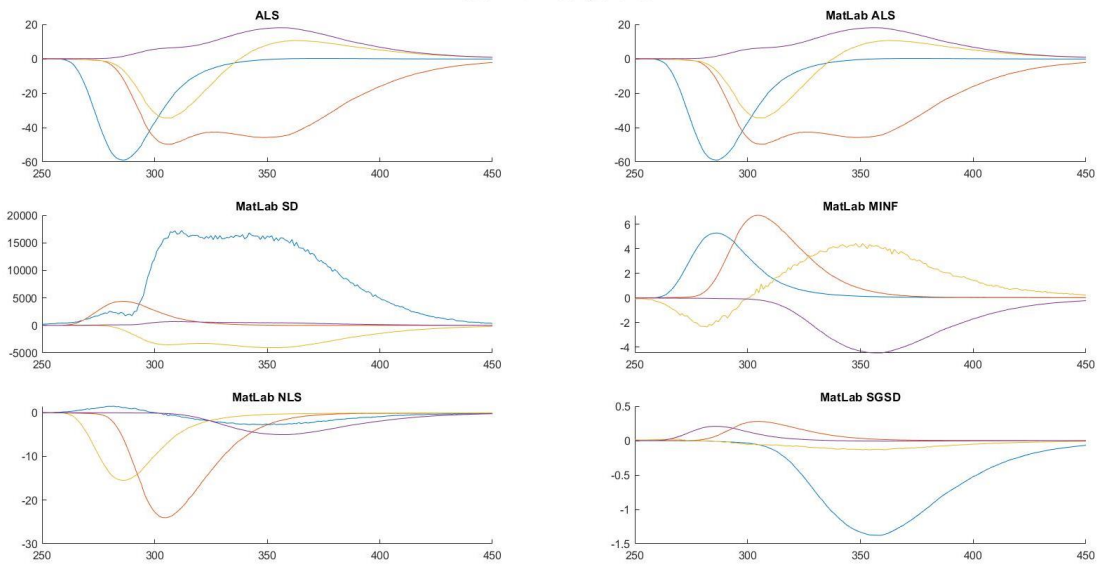
خروجی به صورت زیر بدست می‌آید:



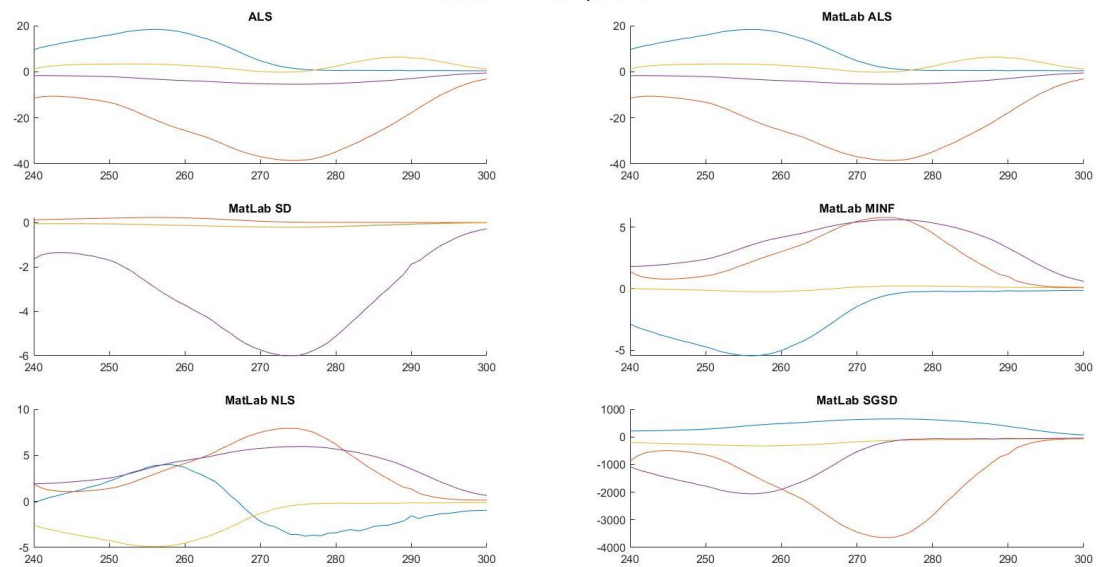
### Excitation with 3 components



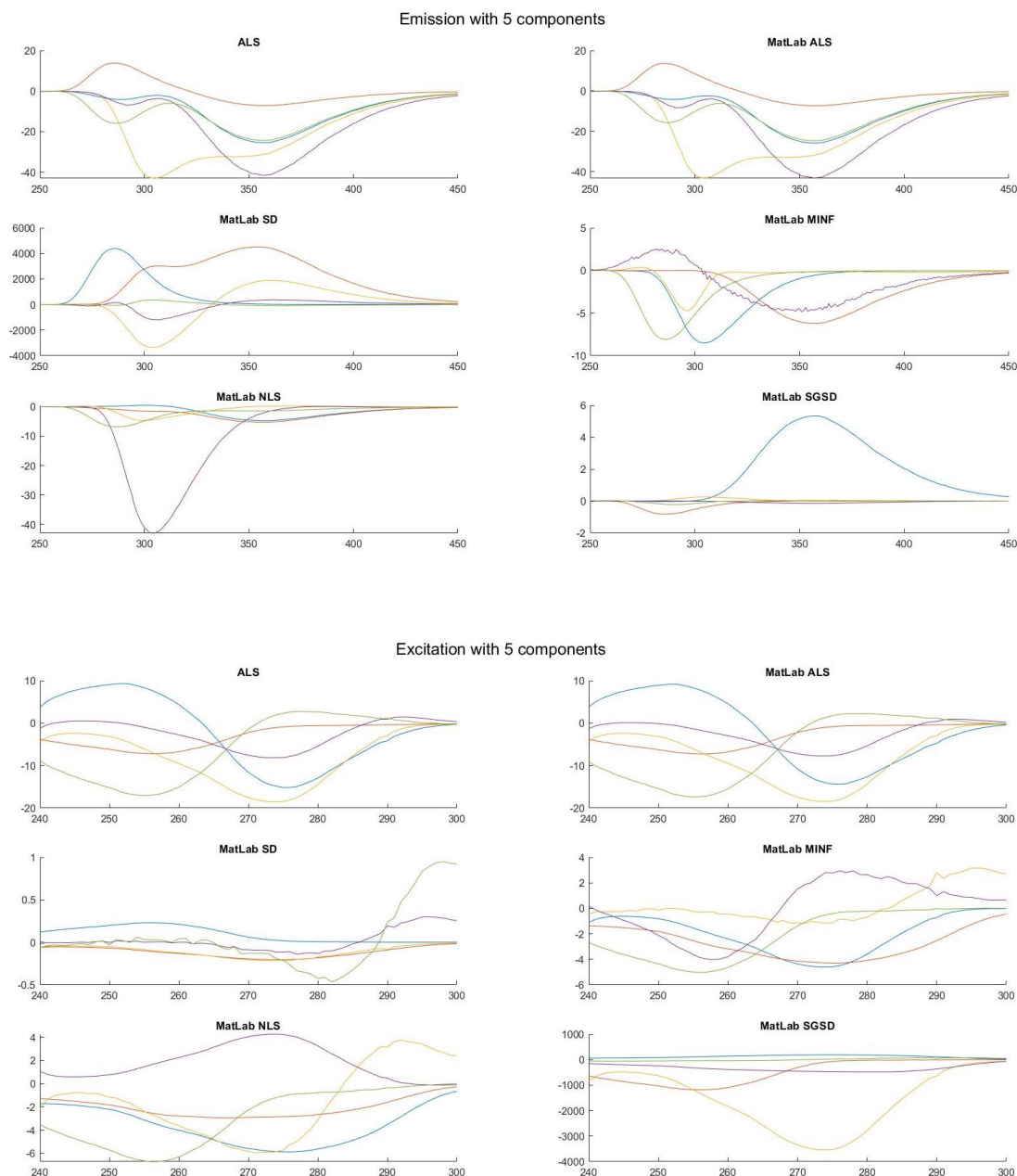
### Emission with 4 components



### Excitation with 4 components







به ازای  $R = 2$ ، همه روش‌ها توانسته‌اند دو مولفه معتبر شناسایی کنند. شایان ذکر است هر دو مولفه شناسایی شده، در اثر وجود مولفه سوم، مقداری با واقعیت خود اختلاف دارد، که البته قابل صرف نظر کردن است.

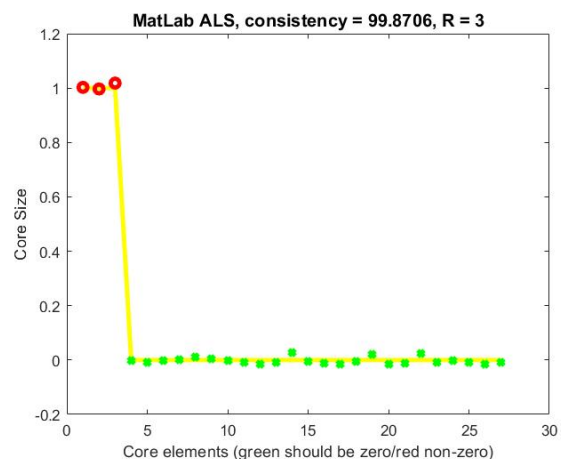
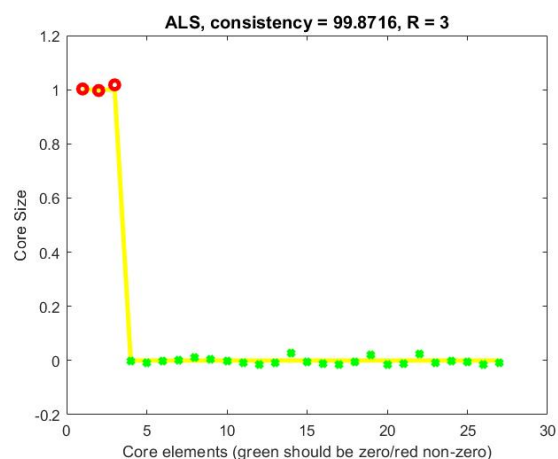
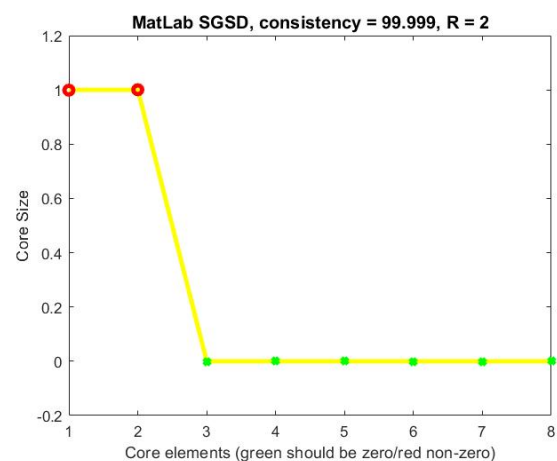
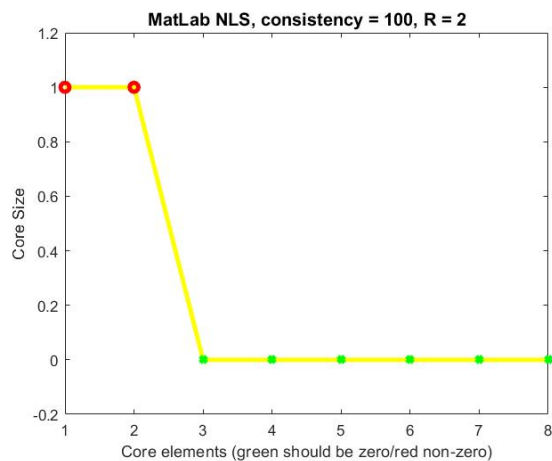
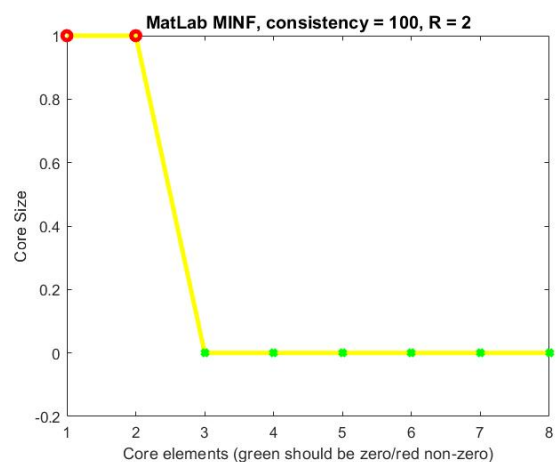
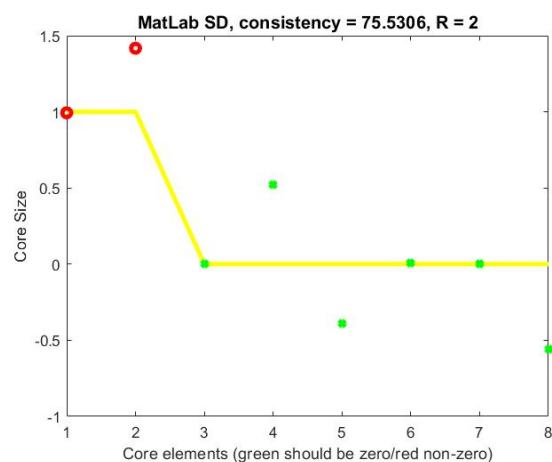
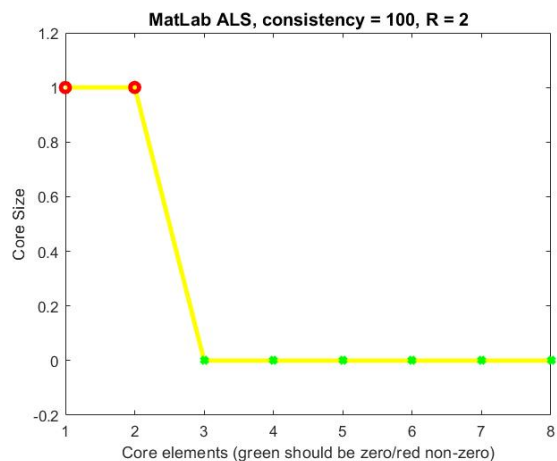
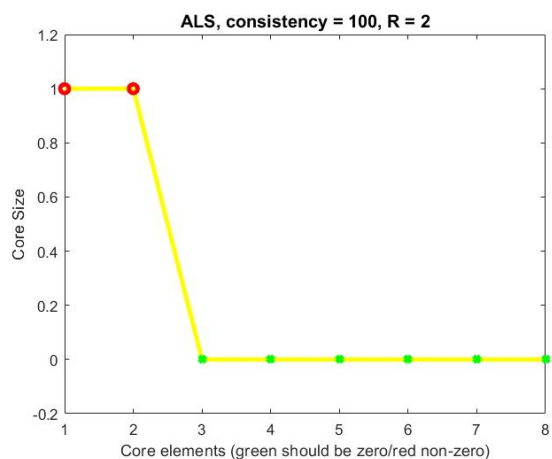
به ازای  $R = 3$ ، همه روش‌ها توانسته‌اند سه مولفه مورد نظر را شناسایی کنند و به جز در یک جایگشت و یک مقیاس، که عدم قطعیت‌های ذاتی مسئله CP هستند، تقریباً عملکرد یکسانی دارند.

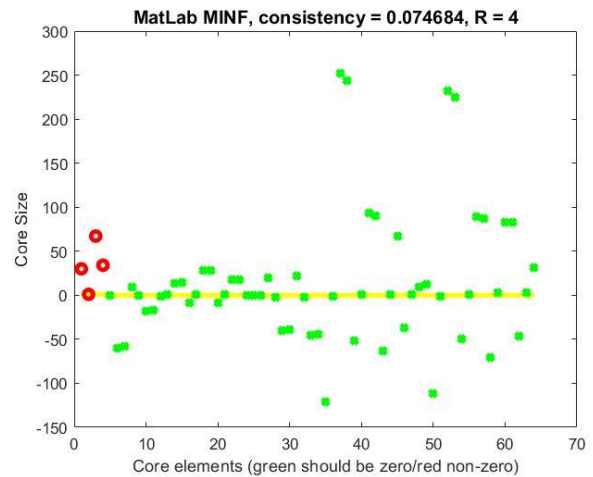
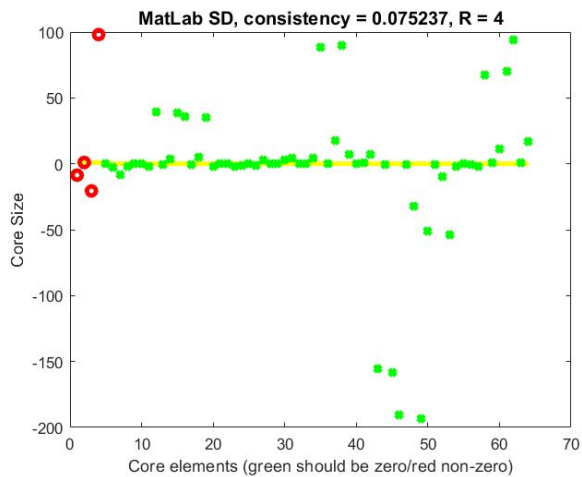
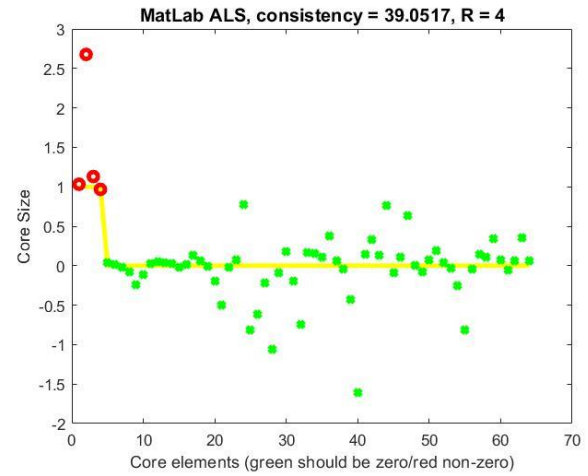
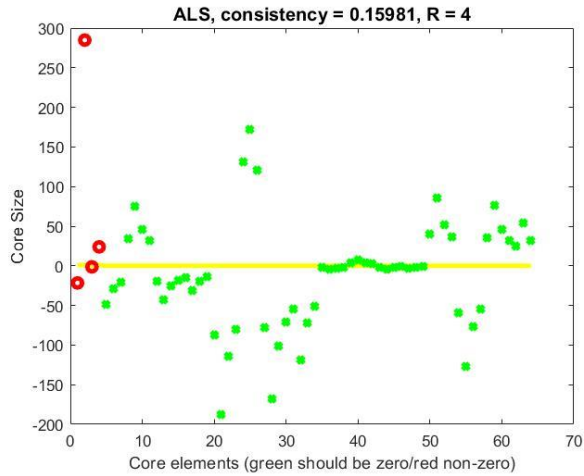
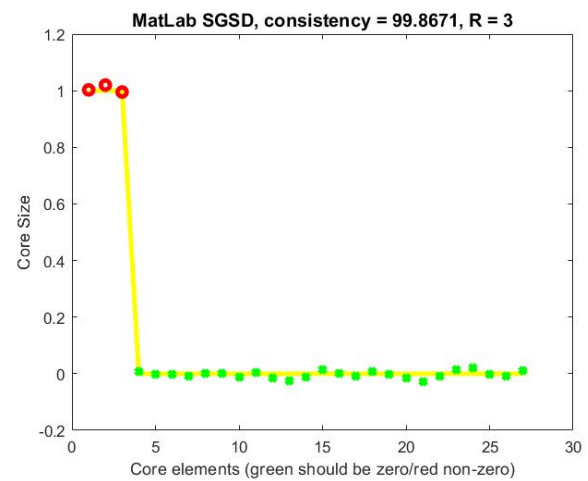
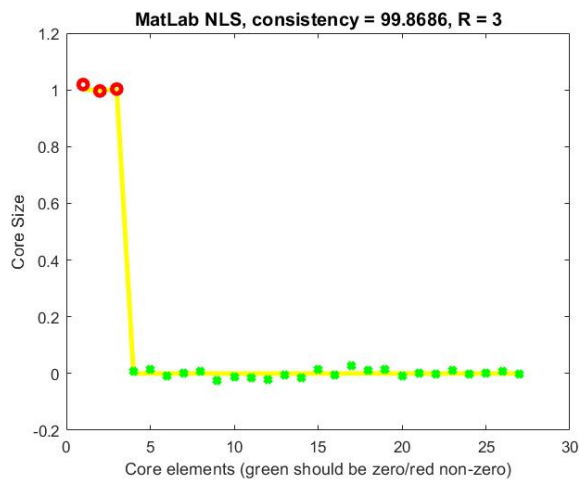
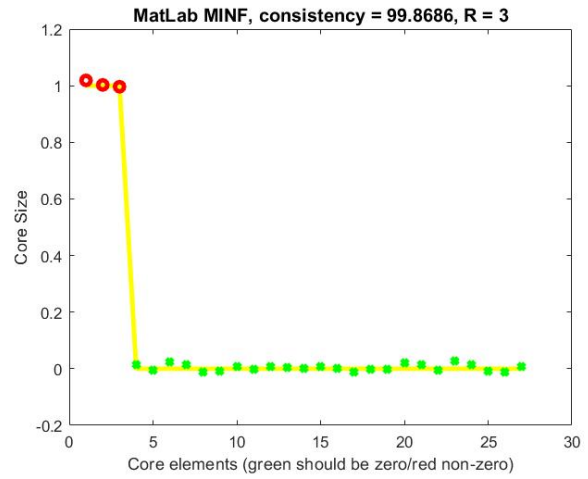
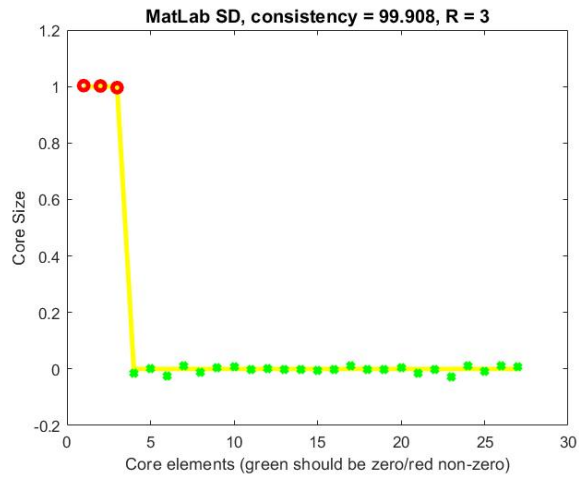
به ازای  $R = 4$ ، روش‌های مبتنی بر ALS، چهار مولفه تشخیص داده‌اند و عملکرد مناسبی نداشته‌اند. روش SD، با وجود اینکه سه مولفه را شناسایی کرده است، نتوانسته است مولفه‌های درستی بدست آورد. روش MINF چهار مولفه تشخیص داده است؛ اما نتوانسته است مولفه‌های اصلی را نیز شناسایی کند. روش‌های NLS و SGSD، سه مولفه بدست آورده‌اند که تقریباً برابر مولفه‌های مورد نظر هستند.

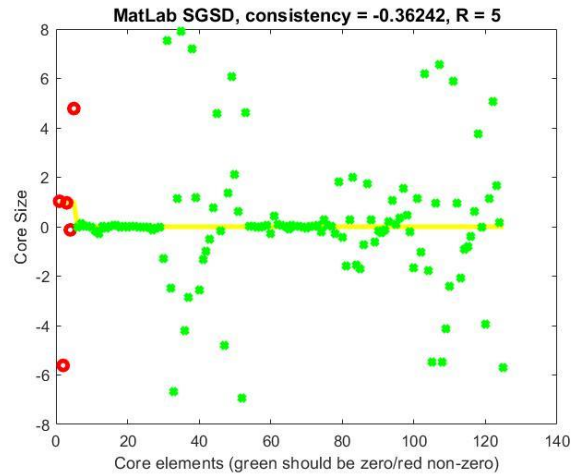
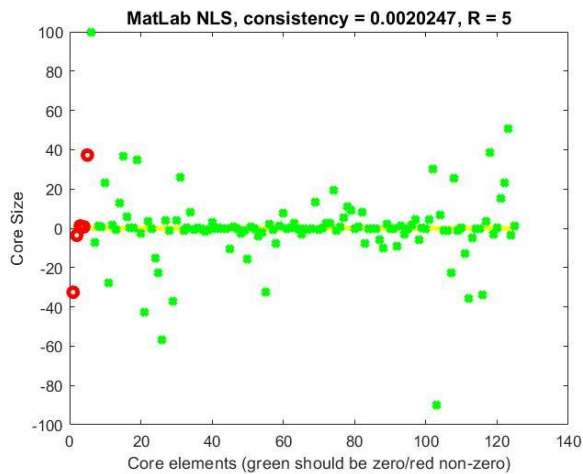
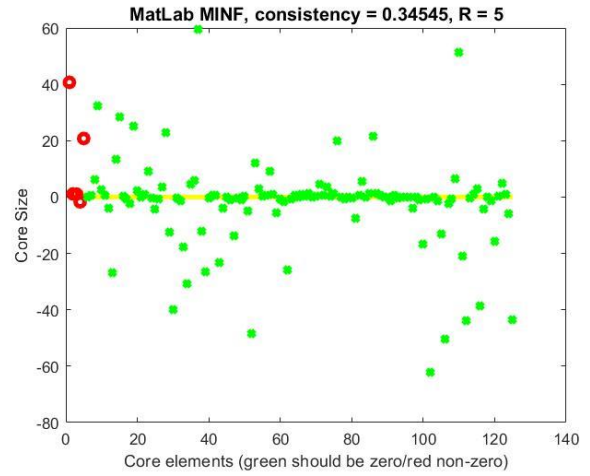
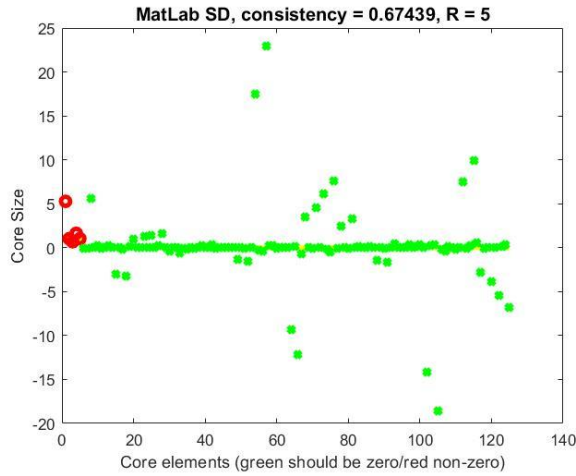
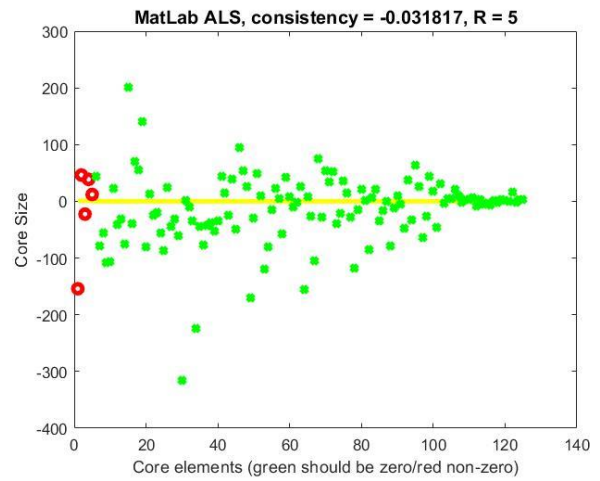
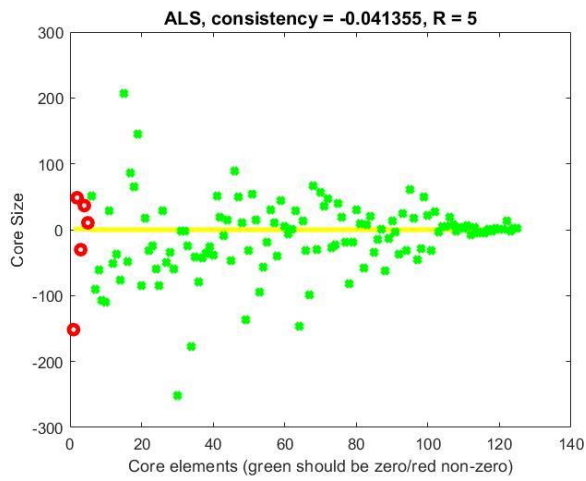
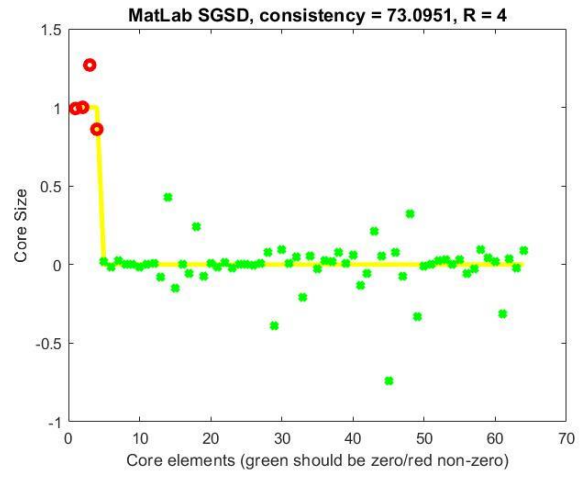
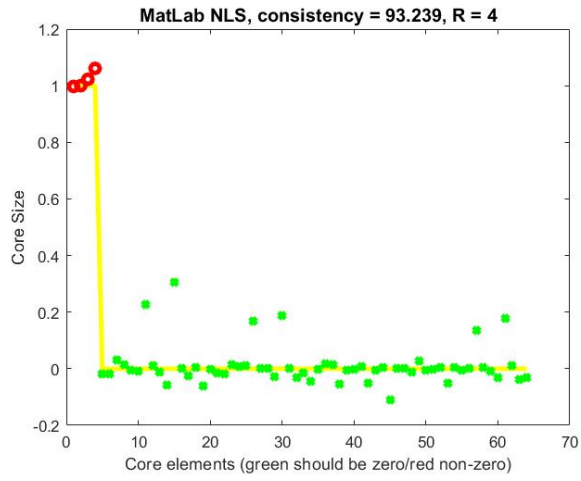
به ازای  $R = 5$ ، روش‌های مبتنی بر ALS و روش SD، پنج مولفه تشخیص داده‌اند و عملکرد مناسبی نداشته‌اند. روش MINF نیز پنج مولفه تشخیص داده است؛ اما نتوانسته است مولفه‌های اصلی را نیز شناسایی کند. روش‌های NLS و SGSD، نیز با مشکل مواجه شده‌اند و یک مولفه قوی و چندین مولفه ضعیف شناسایی کرده‌اند.



(ب) خروجی به صورت زیر بدست می آید:







با توجه به نمودارهای بالا، core consistency، به ازای  $R = 2$  و  $R = 3$  مقادیر بالایی دارد. در حالی که به ازای  $R = 4$  و  $R = 5$  چنین نیست. لذا می‌توان نتیجه گرفت که تجزیه CP، تنها به ازای  $R = 2$  و  $R = 3$ ، تجزیه معتبری برای داده مورد بررسی است و به ازای مقادیر بزرگ‌تر عملکرد مطلوب خود را از دست می‌دهد.