

باسمہ تعالیٰ



دانشگاه صنعتی شریف

دانشکده مهندسی برق

سیگنال‌ها و سیستم‌ها

گزارش کار تمرین سری دوم متلب

امیرحسین افشار راد

۹۵۱۰۱۰۷۷

بهزاد منیری

۹۵۱۰۹۵۶۴

استاد

دکتر حمید کربلایی آقاجان

زمستان ۱۳۹۶

بخش اول - طراحی فیلتر گسسته**۱-۱ و ۲**

ماتریس داده شده را به کمک دستور `load` در متلب باز کرده و ستون اول از سطوح یک تا پنج این ماتریس را در یک بردار، پشت سر هم قرار می‌دهیم.

۱-۳

اندازه تبدیل فوریه این سیگنال را بر حسب فرکانس رسم می‌کنیم.تابع `fft` متلب فرکانس صفر را در درایه اول و باقی فرکانس‌های مثبت و منفی را به ترتیب از کوچترین فرکانس منفی تا بزرگترین فرکانس مثبت در بعد از آن قرار می‌دهد.

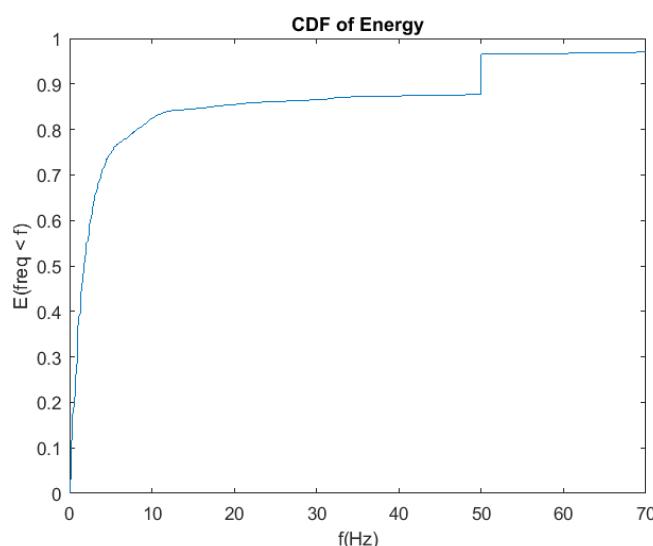
برای نگاشت این فرکانس‌ها بر روی محور واقعی فرکانس و رسم نمودار اندازه تبدیل فوریه، تابع `plotFFT` را نوشتیم. این تابع سیگنال و فرکانس نمونه برداری را در ورودی دریافت می‌کند و بعد از انجام نگاشت مذکور، نمودار تبدیل فوریه این سیگنال را با مشخصات داده شده را رسم می‌کند.

`plotFFT(y, Fs, inf, sup, titlestr, ylabelstr, ysup)`

در کل این تمرین، تمام نمودار‌های حوزه فرکانس را به کمک این تابع رسم می‌کنیم.

دیتا مقدار بسیار زیادی انرژی در DC دارد و به دلیل اینکه در DC هیچ اطلاعاتی وجود ندارد و مهم‌ترین عوامل به وجود آمدن این نویز، انقباض عضلات بدن است و اینکه هر فرمان انقباضی از مغز به یکی از عضلات، می‌تواند مقداری نویز DC به داده‌های ما اضافه کند با کسب اجازه از دکتر کربلایی مقدار DC هر کانال به ازای هر فعالیت را با کم کردن میانگین سیگنال از خود سیگنال حذف کردیم و تمام پردازش‌های آینده در این تمرین را با سیگنال بدون DC انجام می‌دهیم.

حال فرکانسی را پیدا می‌کنیم که نود درصد انرژی سیگنال در آن فرکانس و فرکانس‌های کوچکتر از آن است. بنا بر قضیه پارسوال، انرژی سیگنال و انتگرال توان دوم تبدیل فوریه تنها در یک ضریب تفاوت دارند پس می‌توانیم این محاسبه را مستقیماً در حوزه فرکانس انجام دهیم.

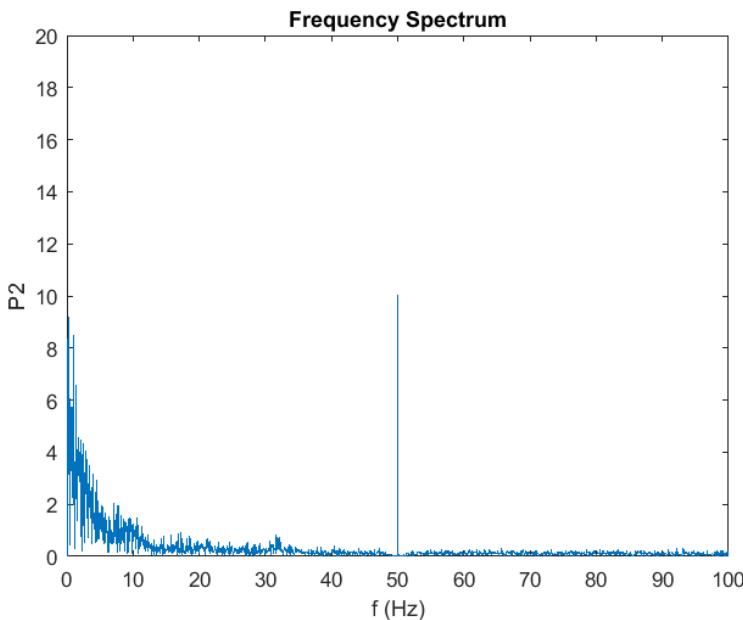


شکل ۱

تابع چگالی تجمعی انرژی $\sum_{i=-\infty}^N Y_i^2 / \sum_{i=-\infty}^{\infty} Y_i^2$ بر حسب فرکانس را رسم می‌کنیم. شکل ۱.

دیده می‌شود که در فرکانس پنجاه هرتز به نود درصد انرژی خواهیم رسید.

در صورتی که مقدار DC سیگنال را حذف نمی‌کردیم، بیش از نود درصد انرژی سیگنال در خود فرکانس صفر قرار می‌گرفت که نتیجه‌ای نامناسبی می‌بود



شکل ۲

۴ - ۱

سیستمی علیٰ با تابع تبدیل گویا در نظر بگیرید. به دلیل علیٰ بودن ناحیه همگرایی تابع تبدیل باید خارج دورترین قطب باشد پس سیستم نمی‌تواند در نمی‌تواند در بی‌نهایت قطب داشته باشد یعنی درجهٔ صورت از درجهٔ مخرج کوچکتر است و $b_0 \neq 0$.

$$H(z) = \frac{Y}{X} = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_p z^{-p}}{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_q z^{-q}} \quad q < p, b_0 \neq 0$$

$$X(z)[a_0 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_p z^{-p}] = Y(z)[b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_q z^{-q}]$$

$$X(z) \left[\frac{a_0}{z^0} + \frac{a_1}{z^1} + \cdots + \frac{a_p}{z^p} \right] = Y(z) \left[\frac{b_0}{z^0} + \frac{b_1}{z^1} + \cdots + \frac{b_q}{z^q} \right]$$

تبدیل Z وارون در ناحیه همگرایی بیرونی

$$a_0 x[n] + a_1 x[n-1] + \cdots + a_p x[n-p] = b_0 y[n] + b_1 y[n-1] + \cdots + b_q y[n-q]$$

که همان نتیجهٔ مد نظر ماست.

ادعا می‌کنیم نمی‌توان یک فیلتر FIR ایده‌آل ساخت. پاسخ ضربهٔ یک فیلتر FIR ایده‌آل در حوزهٔ فرکانس پالس است از طرفی می‌دانیم که تبدیل فوریه وارون پالس یک Sinc است که در تمام زمان‌ها مقدار دارد و این تابع یکتاست پس نمی‌توان فیلتر FIR ایده‌آل ساخت.

فیلتر ما FIR و علی است یعنی پاسخ ضربه آن به فرم $[h[n] = h_0\delta[n] + h_1\delta[n-1] + \dots + h_p\delta[n-p]]$ است پستابع تبدیل آن با تبدیل زد گرفتن از پاسخ ضربه به صورت زیر به دست می‌آید.

$$H(z) = \frac{Y(z)}{X(z)} = h_0 + \frac{h_1}{z} + \dots + \frac{h_p}{z^p} \rightarrow H(z) = h_0 X(z) + \frac{h_1}{z} X(z) + \frac{h_2}{z^2} X(z) + \dots \rightarrow$$

$$h[n] = h_0 x[n] + h_1 x[n-1] + \dots = \sum_{k=0}^N h_k x[n-k]$$

به طور کلی یک فیلتر FIR می‌تواند به تعداد صفرهایش، قطب در مبدا داشته باشد و هیچ قطب دیگری ندارد. صفرهای یک فیلتر FIR هر جایی به جز مبدا می‌توانند باشند.

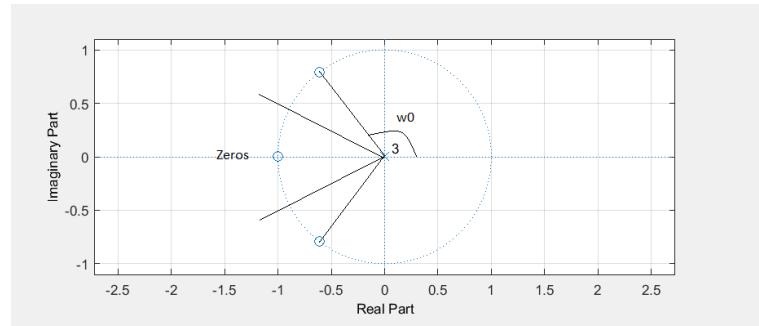
با استفاده از اطلاعاتی که تاکنون به دست آورده‌ایم می‌توانیک یک فیلتر FIR پایین گذر طراحی کنیم. برای مشخص کردن این فیلتر کافی است که صفرهای آن را مشخص کنیم.

فرض کنیم قصد داریم فیلتری پایین گذر با فرکانس قطع ω_0 بسازیم. تابع تبدیل این فیلتر به فرم زیر است.

$$H(z) = \frac{K(z - z_1)(z - z_2) \dots (z - z_p)}{z^p}$$

برای مشخص کردن پاسخ فرکانسی باید اندازه و فاز تابع تبدیل را در $\exp(j\omega)$ به دست بیاوریم.

$$H(e^{j\omega}) = \frac{K(e^{j\omega} - z_1)(e^{j\omega} - z_2) \dots (e^{j\omega} - z_p)}{e^{jp\omega}} \rightarrow |H(e^{j\omega})| = |K| |e^{j\omega} - z_1| |e^{j\omega} - z_2| \dots |e^{j\omega} - z_p|$$



این بدین معناست که هر چه به صفرها نزدیک تر باشیم اندازه پاسخ فرکانسی کوچکتر خواهد شد و سیستم ورودی را تضعیف خواهد کرد پس باید در نزدیکی فرکانس‌هایی که قصد تضعیف آنها را داریم صفر بگذاریم.

ما روش زیر را برای قرار دادن صفرها پیشنهاد می‌کنیم.

فرض کنید فرکانس قطع ω_0 باشد. از فرکانس ω_0 با فاصله‌های برابر به تعداد مرتبه فیلتر،

صفر قرار می‌دهیم. ضرب χ عددی بین صفر و یک است. اگر از این ضریب استفاده نمی‌کردیم در خود فرکانس قطع، فیلتر اولین

Ripple خود را انجام می‌داد و کاملاً صفر می‌شد در حالی که انتظار داریم انتظار داریم فیلتر به مرور و در فرکانس‌های کمی بزرگ‌تر از فرکانس قطع به این مرحله برسد.

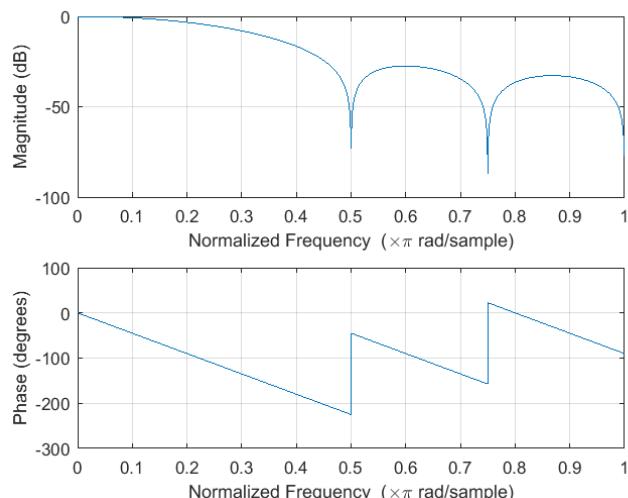
برای مشخص کردن مرتبه فیلتر از محک زیر استفاده می‌کنیم. با استفاده از روش بالا با فرکانس قطع خواسته شده می‌سازیم. محاسبه می‌کنیم چه مقدار که چه کسری از انرژی فیلتر در قبل از فرکانس قطع قرار گرفته است. اگر این کسر از حد آستانه‌ای ای که خودمان انتخاب کردیم بیشتر بود، فیلتر را می‌پذیریم و در غیر این صورت یک مرتبه فیلتر می‌افزاییم و فرایند بالا را تکرار می‌کنیم.

پس در نهایت فیلتر ما با چهار عدد زیر به طور کامل توصیف می‌شود.

۱. فرکانس نرمالیزه قطع فیلتر (ω_0)
۲. عدد χ که در بالا معرفی شد و نمایشگر محل شروع صفر گذاری است.
۳. شعاع دایره‌ای که صفرها را در آن قرار می‌دهیم. (r_0)
۴. حداقل درصد انرژی فیلتر در قبل از فرکانس قطع (percent)

تابع LowPassFilterDesigner دقیقاً مراحل ذکر شده را انجام می‌دهد. چهار عدد بالا را به عنوان ورودی گرفته و فیلتر را طراحی می‌کند.

```
function [b, a] = LowPassFilterDesigner (r0, w0, percent, chi)
```



شکل ۴

شکل رویرو مثالی از خروجی این تابع برای مشخصات زیر است:

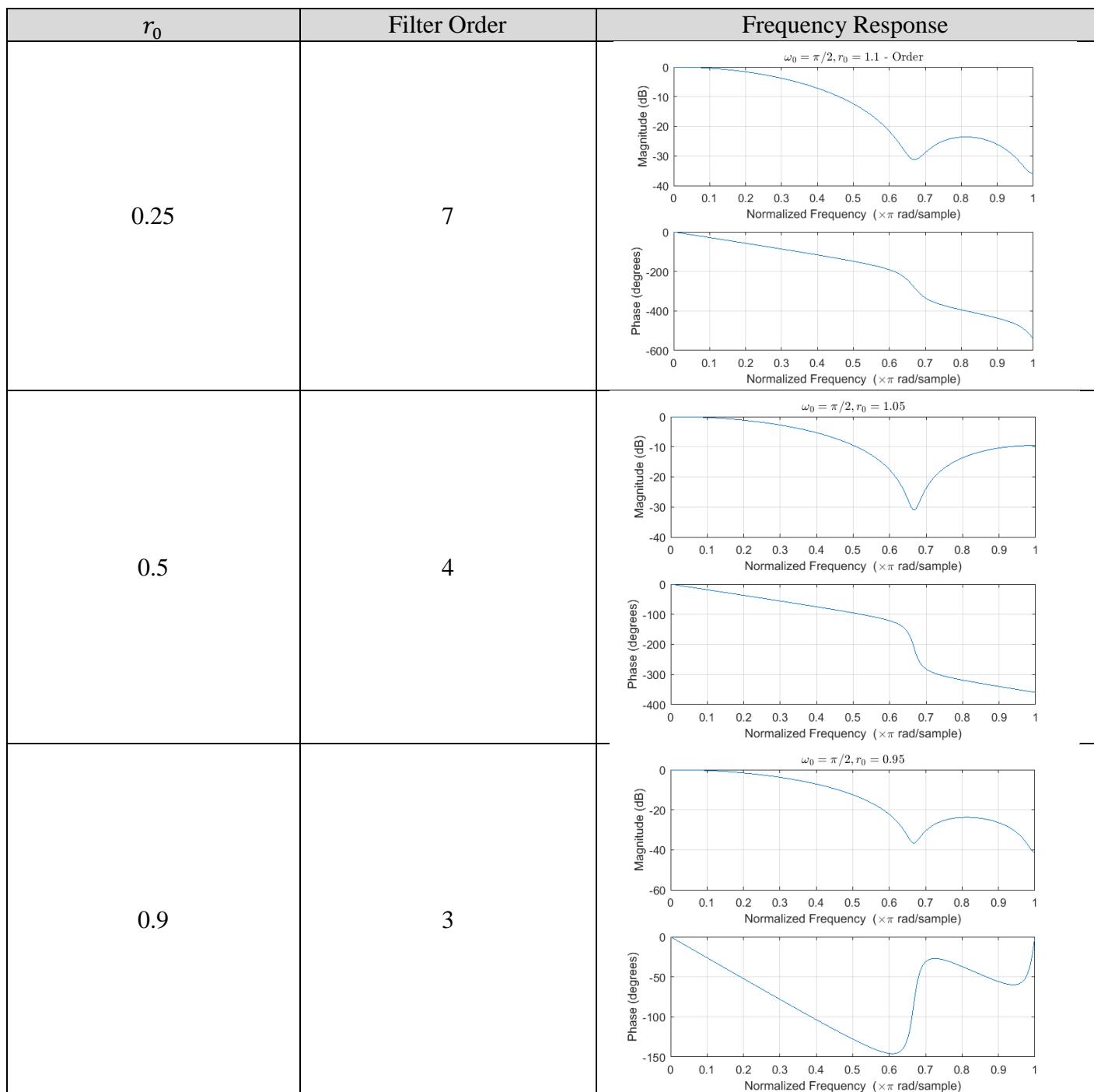
$$\omega_0 = 0.25 \quad \chi = \frac{1}{3} \quad r_0 = 1 \quad percent = 0.9$$

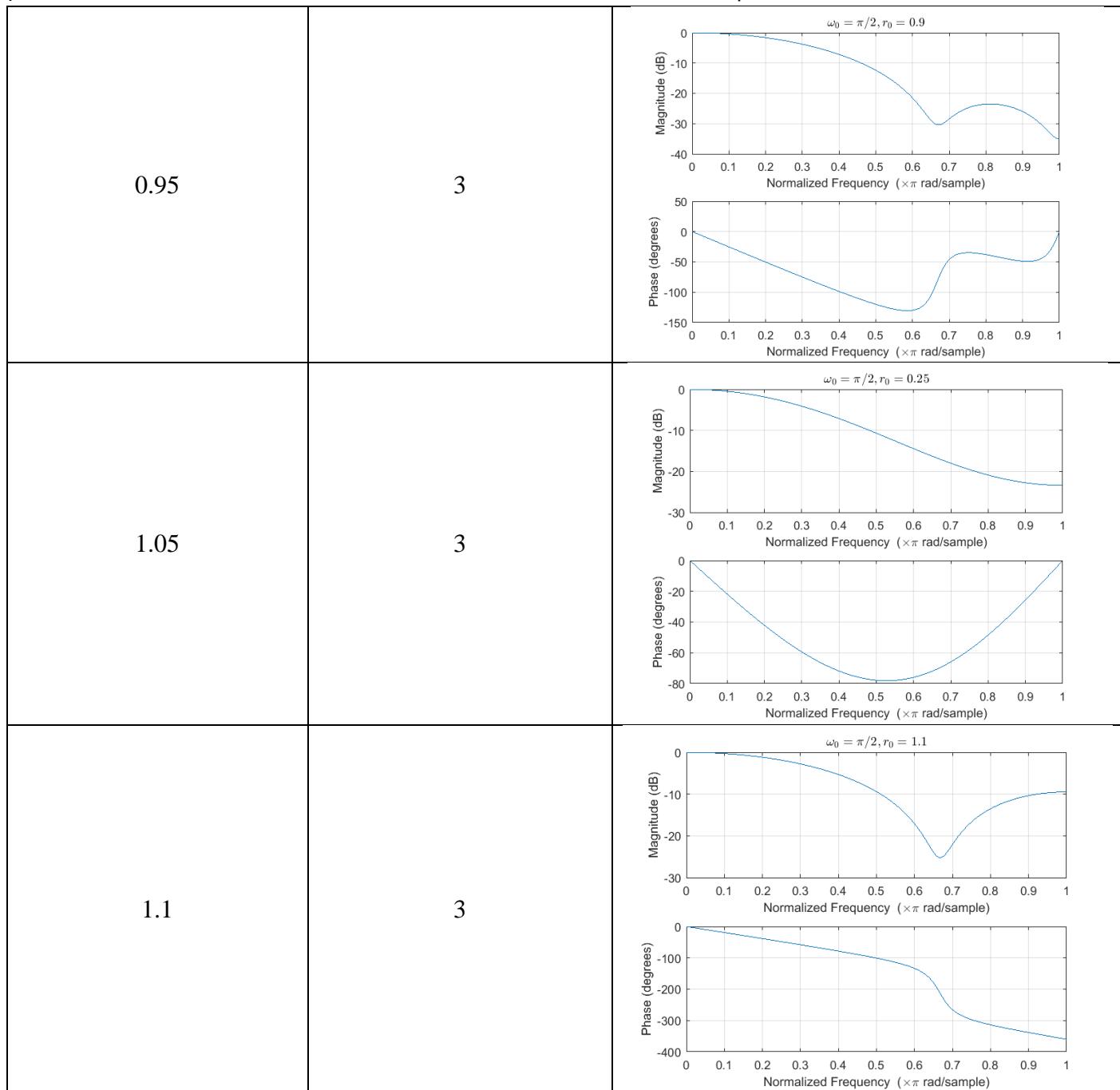
با این تنظیمات، فیلتر ما فرکانس قطع را ۸ دسیبل تضعیف می‌کند که تقریباً در حدود حد معروف ۳ دسیبل است.

این فیلتر، مرتبه هشت است و ۹۰ درصد انرژی آن قبل از فرکانس ۰/۲۴۵۳۸ است.

در این بخش به بررسی اثر تغییر شعاعی که صفرها روی آن قرار گرفته‌اند می‌پردازیم. فیلتری با فرکانس قطع $\frac{\pi}{2}$ و درصد انرژی ۹۵ درصد و $\chi = \frac{1}{3}$ می‌سازیم. این کار را برای چند شعاع مختلف تکرار می‌کنیم.

$$r_0 = \{0.25, 0.5, 0.9, 0.95, 1.05, 1.1\}$$





جدول ۱

هر چه صفر‌ها به دایرۀ واحد نزدیک‌تر باشند، اثر تضعیفی آنها افزایش می‌یابد. این موضوع با رابطه

$$H(e^{j\omega}) = \frac{K(e^{j\omega} - z_1)(e^{j\omega} - z_2) \dots (e^{j\omega} - z_p)}{e^{jp\omega}} \rightarrow |H(e^{j\omega})| = |K| |e^{j\omega} - z_1| |e^{j\omega} - z_2| \dots |e^{j\omega} - z_p|$$

نیز مطابقت دارد. اگر صفرها را بر روی دایرۀ واحد قرار بدهیم، فیلتر آن فرکانس‌ها را به طور کامل حذف می‌کند. این ویژگی از نظر ما بسیار مثبت است پس از این به بعد در صورت لزوم از این نوع فیلتر استفاده می‌کنیم.

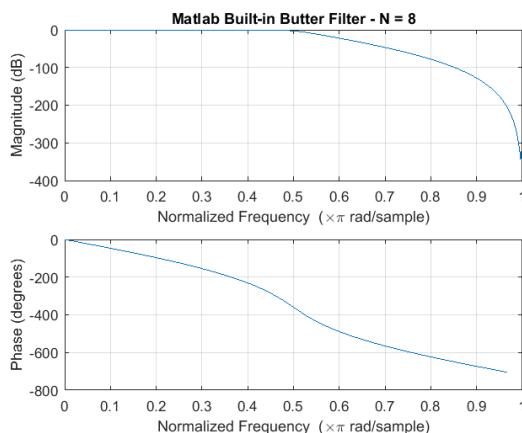
دور کردن صفرها از دایرۀ واحد باعث کوچک شدن مشتق دامنه و فاز پاسخ فرکانس نیز خواهد شد که باز هم مورد پسند ما نیست.

برای این بخش، تابع Sinc که در تمام زمان‌ها مقدار دارد را بریده و تنها N جمله از آن نگه می‌داریم. تابع Sinc یک فیلتر پایین گذر ایده‌آل است و تابع Sinc بریده شده، تقریبی از این فیلتر ایده‌آل خواهد بود.

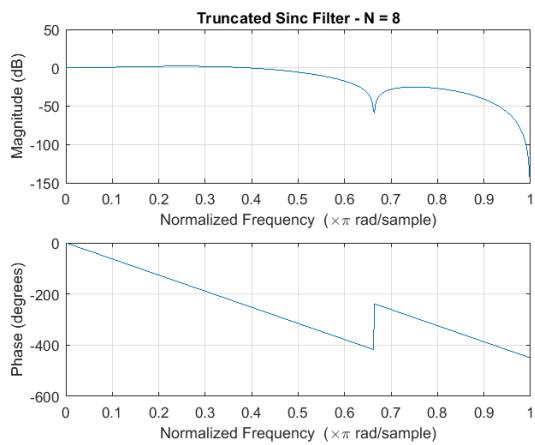
پاسخ ضربه این سیستم محدود در زمان است پس فیلتر FIR است.

در مثال زیر، با نگه داشتن هشت جمله از Sinc ، یک فیلتر پایین گذر مرتبه هشت می‌سازیم.

شکل ۶ پاسخ فرکانسی فیلتر Truncated Sinc با مرتبه ۸ ماست و شکل ۵ فیلتر Butter متلب با مرتبه ۸ است.



شکل ۵

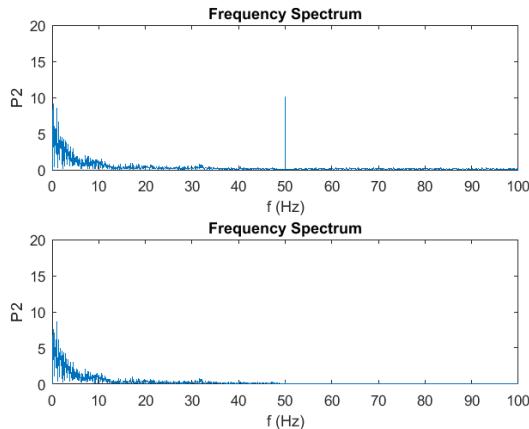


شکل ۶

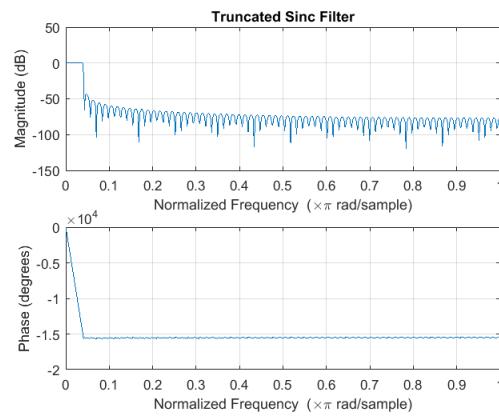
با مطالعه ما فیلتر Butterworth هیچ Ripple ای ندارد و همواره Sinc فیلتر Flat است اما فیلتر Sinc بریده شده، همان‌طور که در شکل‌های ۶ و ۷ مشخص است Ripple دارد.

شکل ۷ در صفحه بعد یک نمونه از فیلتر Sinc بریده ما با مرتبه ۴۲۳۹ است. همان‌طور که مشاهده می‌کنید آن فیلتر تا حد زیادی به فیلتر ایده‌آل نزدیک شده است.

برداری که درسوال ۲ به دست آورده‌ایم را تا 50Hz با استفاده از فیلتری که درقسمت قبل ساخته‌ایم فیلتر می‌کنیم. مرتبه فیلتر را آنقدر بزرگ انتخاب می‌کنیم که 99.9% از انرژی کل فیلتر در قبل از فرکانس قطع آن باشد.



شکل ۱



شکل ۷

به دلیل شرط بسیار سخت بر روی انرژی فیلتر، مرتبه فیلتر ما بسیار بالا رفت. فیلتر بالا فیلتری از مرتبه ۴۲۳۹ است و 99.9% درصد از انرژی آن در قبل از فرکانس قطع آن است. شکل ۷ پاسخ فرکانس فیلتر و شکل ۸، تبدیل فوریه سیگنال، قبل و بعد از اعمال فیلتر است.

در حالت کلی برای بهبود کیفیت فیلتر می‌توان آن را در یک **hamming** نیز ضرب کرد که با توجه به نظر دستیارهای آموزشی از انجام این کار در قسمت اول تمرین خودداری شده ولی در قسمت دوم از آن استفاده شده است.

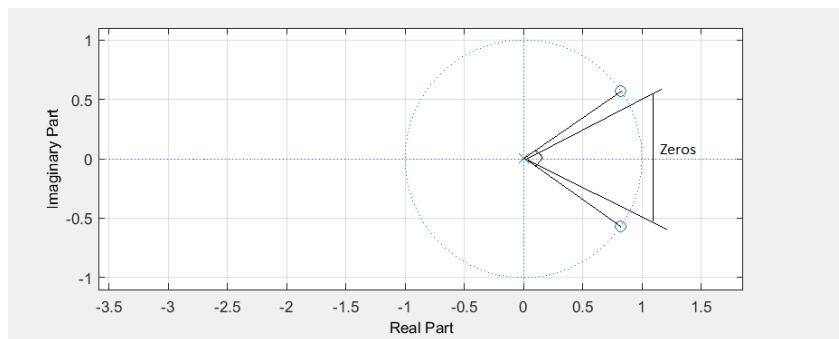
حال یک فیلتر butter نیز انجام می‌دهیم. متاسفانه فیلتر باطر در متلب نمی‌تواند مرتبه بزرگتر از ۵۰۰ داشته باشد و تا این مرتبه هنوز به حد 99.9% درصد از انرژی نرسیده‌ایم.

Error using butter (line 62)

Filter order too large.

۱ - ۱۲

برای ساخت یک فیلتر بالا گذرنیز همانند سوال ۸ عمل می‌کنیم با این تفاوت که صفرها را در سمت چپ قرار می‌دهیم. به شکل زیر توجه کنید



شکل ۹

همانند سوال ۸، صفرها را با فاصلهٔ یکسان از نقطهٔ صفر تا ω_0 قرار می‌دهیم. فیلتر ما حقیقی است بنابراین به ازای هر صفر که قرار می‌دهیم باید مزدوج آن را نیز قرار دهیم. برای پیدا کردن مرتبهٔ مناسب برای فیلتر هم از شرط سوال ۸ استفاده می‌کنیم با این تفاوت که شرط می‌کنیم درصدی خاص از انرژی فیلتر، در بعد از فرکانس قطع باشد.

پس در نهایت فیلتر ما با چهار عدد زیر به طور کامل توصیف می‌شود.

۱. فرکانس نرمالیزهٔ قطع فیلتر (ω_0)
۲. عدد χ که در بالا معرفی شد و نمایشگر محل شروع صفرگذاری است.
۳. شعاع دایره‌ای که صفرها را در آن قرار می‌دهیم. (r_0)
۴. حداقل درصد انرژی فیلتر در بعد از فرکانس قطع (percent)

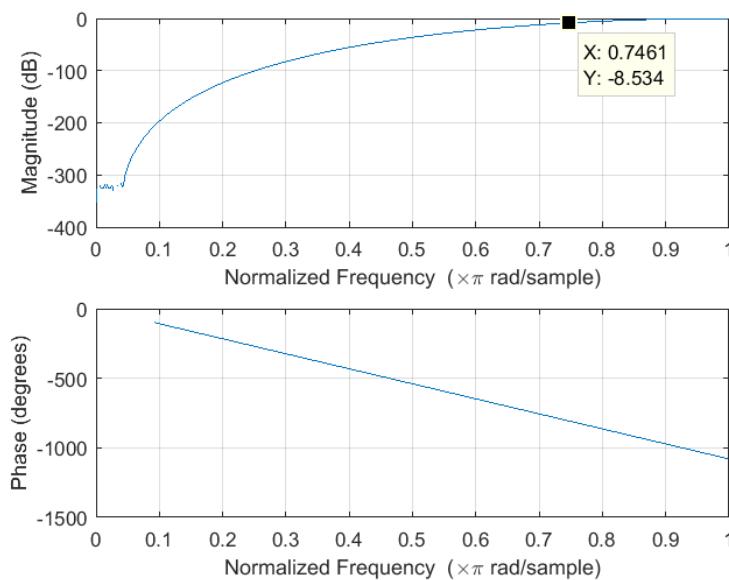
تابع HighPassFilterDesigner دقیقاً مراحل ذکر شده را انجام می‌دهد. چهار عدد بالا را به عنوان ورودی گرفته و فیلتر را طراحی می‌کند.

```
function [b, a] = HighPassFilterDesigner (r0, w0, percent, chi)
```

شکل ۱۰ مثالی از خروجی اینتابع برای مشخصات زیر است:

$$\omega_0 = 0.75 \quad \chi = \frac{1}{3} \quad r_0 = 1 \quad percent = 0.95$$

با این تنظیمات، فیلتر ما فرکانس قطع را ۷.۵ دسیبل تضعیف می‌کند که تقریباً در حدود حد معروف ۳ دسیبل است.
این فیلتر، مرتبه ۱۲ است و ۹۵ درصد انرژی آن بعد از فرکانس قطع است.



شکل ۱۰

در ادامه این بخش به بررسی اثر تغییر شعاعی که صفرها روی آن قرار گرفته‌اند می‌پردازیم. فیلتری با فرکانس قطع $\frac{\pi}{2}$ و درصد انرژی ۹۵ درصد و $\chi = \frac{1}{3}$ می‌سازیم. این کار را برای چند شعاع مختلف تکرار می‌کنیم.

$$r_0 = \{0.25, 0.5, 0.9, 0.95, 1.05, 1.1\}$$

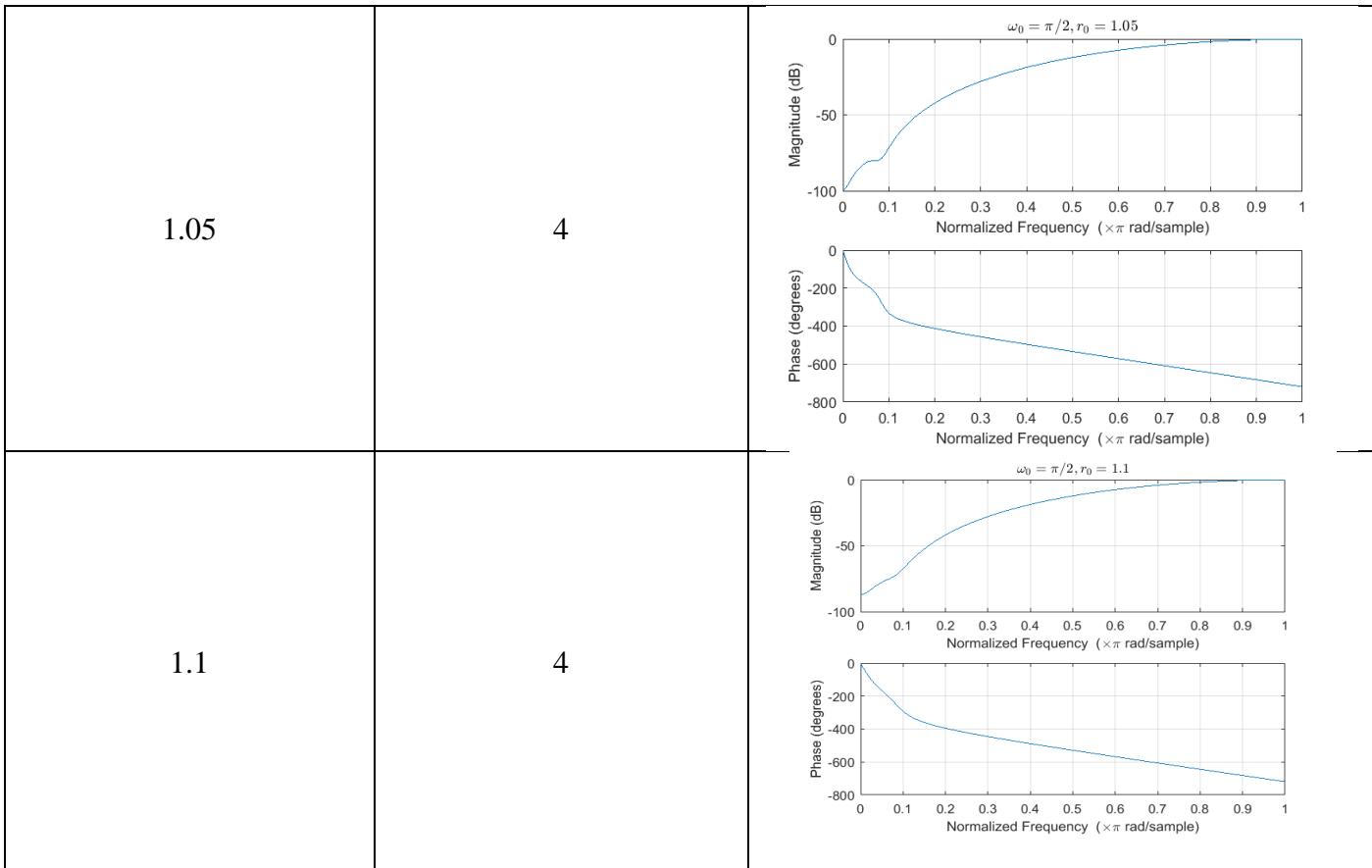
هر چه صفرها به دایره واحد نزدیک‌تر باشند، اثر تضعیفی آنها افزایش می‌یابد. این موضوع با رابطه

$$H(e^{j\omega}) = \frac{K(e^{j\omega} - z_1)(e^{j\omega} - z_2) \dots (e^{j\omega} - z_p)}{e^{jp\omega}} \rightarrow |H(e^{j\omega})| = |K| |e^{j\omega} - z_1| |e^{j\omega} - z_2| \dots |e^{j\omega} - z_p|$$

نیز مطابقت دارد. اگر صفرها را بر روی دایره واحد قرار بدھیم، فیلتر آن فرکانس‌ها را به طور کامل حذف می‌کند. این ویژگی از نظر ما بسیار مثبت است پس از این به بعد در صورت لزوم از این نوع فیلتر استفاده می‌کنیم.

دور کردن صفرها از دایره واحد باعث کوچک شدن مشتق دامنه و فاز پاسخ فرکانس نیز خواهد شد که باز هم مورد پسند ما نیست.

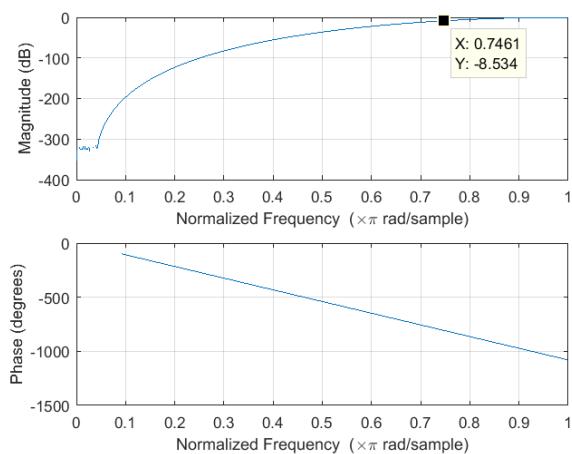
r_0	Filter Order	Frequency Response
0.25	6	<p>Top plot: Magnitude (dB) vs Normalized Frequency ($\times \pi$ rad/sample). The magnitude increases from -30 dB at 0.1 rad/sample to 0 dB at 1 rad/sample. The corner frequency is approximately 0.4 rad/sample.</p> <p>Bottom plot: Phase (degrees) vs Normalized Frequency ($\times \pi$ rad/sample). The phase starts at 0 degrees at 0 rad/sample, reaches a peak of about 80 degrees at 0.4 rad/sample, and then decreases back to 0 degrees at 1 rad/sample.</p>
0.5	4	<p>Top plot: Magnitude (dB) vs Normalized Frequency ($\times \pi$ rad/sample). The magnitude increases from -40 dB at 0.1 rad/sample to 0 dB at 1 rad/sample. The corner frequency is approximately 0.4 rad/sample.</p> <p>Bottom plot: Phase (degrees) vs Normalized Frequency ($\times \pi$ rad/sample). The phase starts at 0 degrees at 0 rad/sample, reaches a peak of about 120 degrees at 0.4 rad/sample, and then decreases back to 0 degrees at 1 rad/sample.</p>
0.9	4	<p>Top plot: Magnitude (dB) vs Normalized Frequency ($\times \pi$ rad/sample). The magnitude increases from -100 dB at 0.1 rad/sample to 0 dB at 1 rad/sample. The corner frequency is approximately 0.4 rad/sample.</p> <p>Bottom plot: Phase (degrees) vs Normalized Frequency ($\times \pi$ rad/sample). The phase starts at 0 degrees at 0 rad/sample, reaches a peak of about 250 degrees at 0.1 rad/sample, and then decreases back to 0 degrees at 1 rad/sample.</p>
0.95	4	<p>Top plot: Magnitude (dB) vs Normalized Frequency ($\times \pi$ rad/sample). The magnitude increases from -100 dB at 0.1 rad/sample to 0 dB at 1 rad/sample. The corner frequency is approximately 0.4 rad/sample.</p> <p>Bottom plot: Phase (degrees) vs Normalized Frequency ($\times \pi$ rad/sample). The phase starts at 0 degrees at 0 rad/sample, reaches a peak of about 250 degrees at 0.1 rad/sample, and then decreases back to 0 degrees at 1 rad/sample.</p>



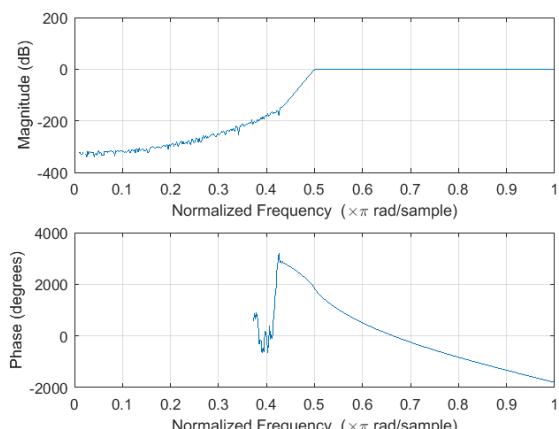
جدول ۲

می‌دانیم که Sinc یک فیلتر ایده‌آل پایین‌گذر است. برای ساخت یک فیلتر بالاگذر ایده‌آل می‌توانیم فیلتر بخش ۱۰ را در یک منفی ضرب کرده و با $[n]\delta$ جمع کنیم. بدیهی است که مرتبه این فیلتر باید همواره فرد باشد تا فیلتر مناسبی به دست بیاید.

با روش بالا یک فیلتر با فرکانس قطع ۰.۷۵ و مرتبه ۹۱ می‌سازیم. شکل ۱۱. شکل ۱۴ نمونه‌ای از این فیلتر با مرتبه بسیار بالا و بسیار نزدیک به ایده‌آل است.



شکل ۱۱

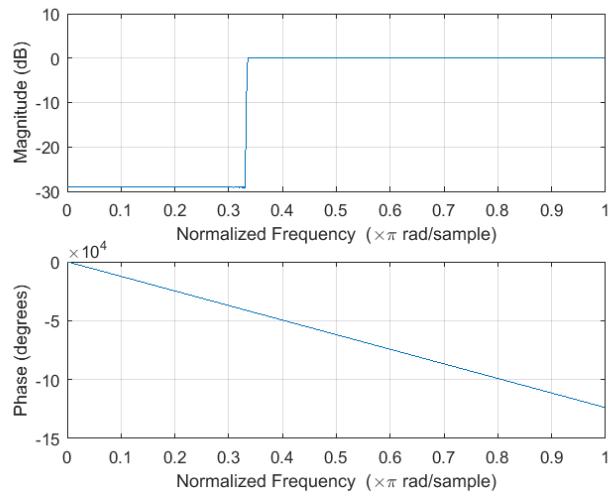


شکل ۱۲

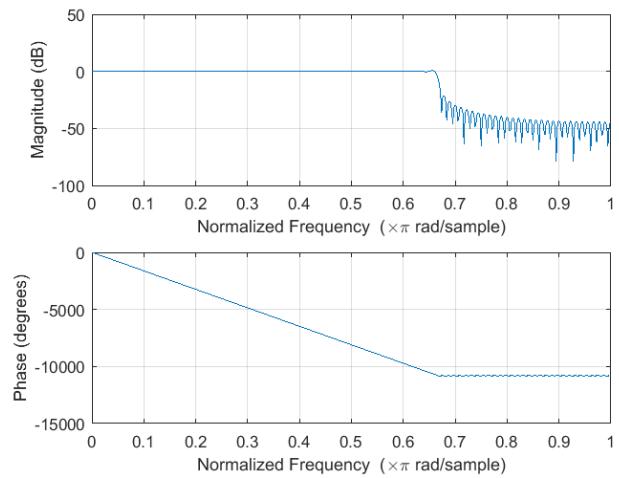
شکل ۱۲ یک فیلتر Butterworth مرتبه ۹۱ بالاگذر است.

با پشت سر هم قرار دادن دو فیلتر پایین‌گذر و بالاگذر می‌توان یک فیلتر میان‌گذر ساخت. کانولوشن پاسخ ضربه این فیلترها پاسخ ضربه سیستم معادل است.

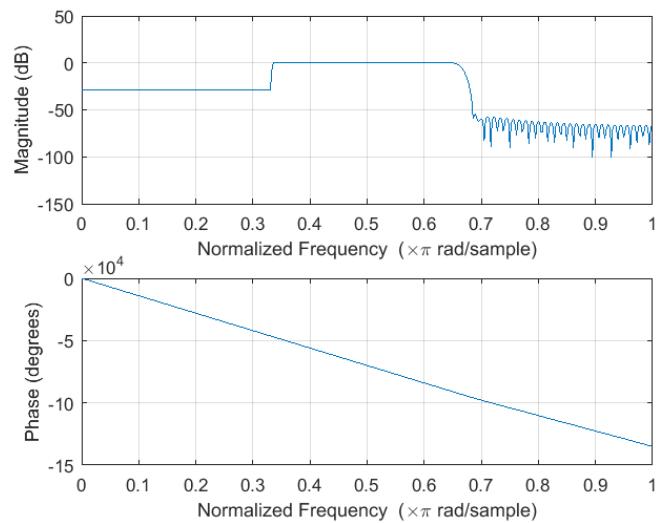
با استفاده از فیلتر سینک بریده شده بالاگذر و پایین‌گذر با مرتبه بالا که در سوال‌های قبل معرفی شد و کانولوشن پاسخ ضربه آنها یک فیلتر میان‌گذر می‌سازیم.



شکل ۱۴



شکل ۱۳



شکل ۱۵

فیلتر بالاگذر از مرتبه ۱۳۷۷ و فیلتر پایین‌گذر از مرتبه ۱۸۱ است.

سؤال دوم)

(۱-۲)

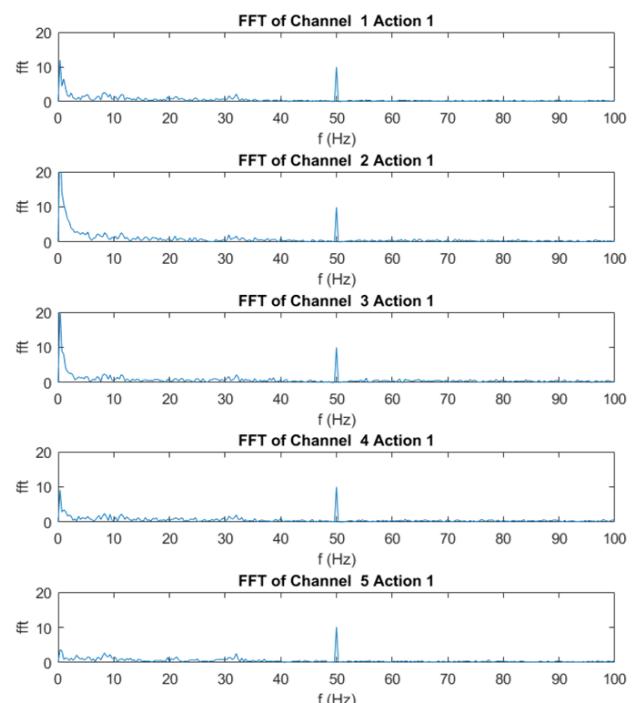
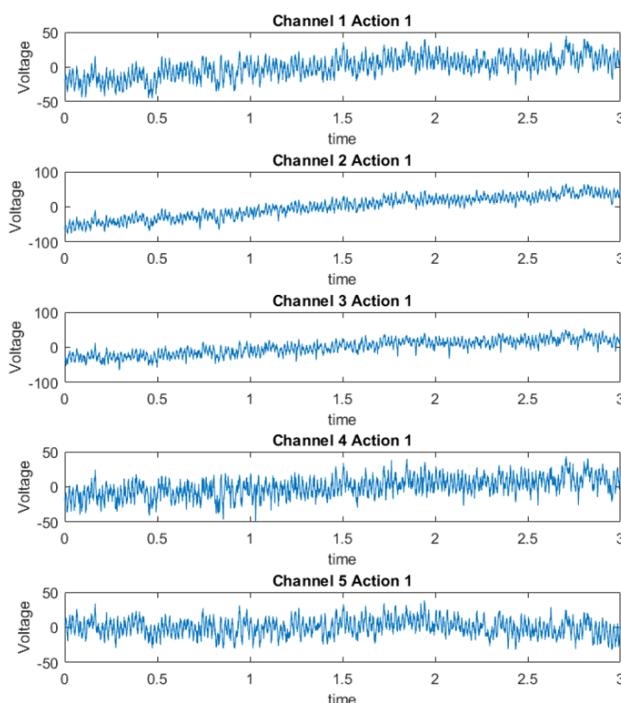
میانگین، انحراف معیار، مینیمم و ماکسیمم برای تمامی کانال‌های سه فعالیت اول محاسبه شدند و در استراکت **stats** ذخیره شده است. این استراکت دارای چهار فیلد **mean**, **min**, **std**، و **max** می‌باشد. به دلیل حجم زیاد خروجی این بخش، از گزارش داده‌های عددی در این گزارش صرف نظر می‌کنیم.

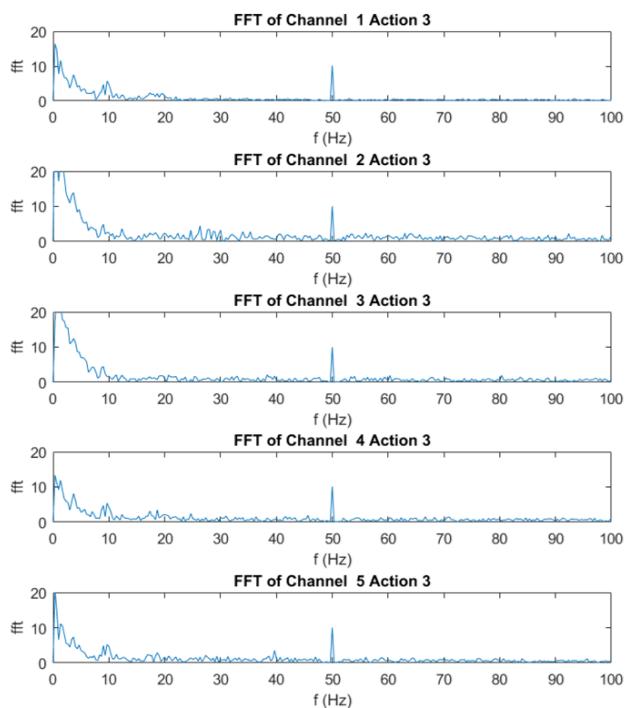
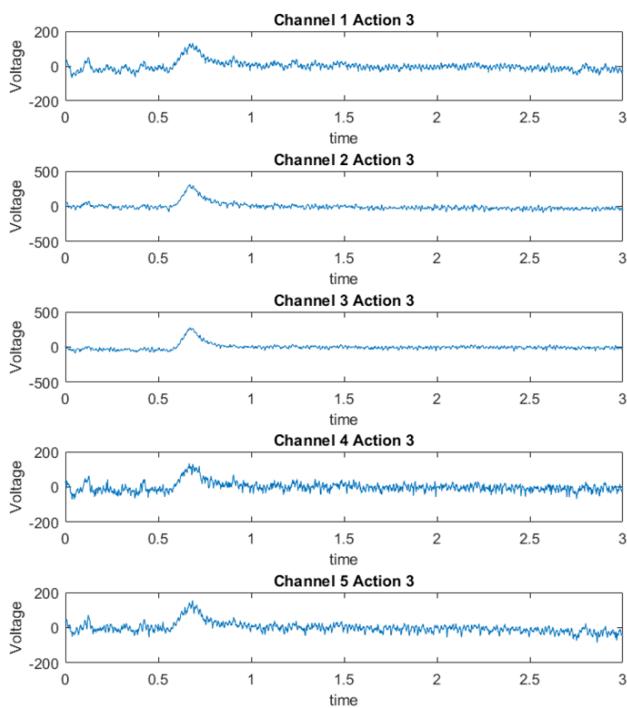
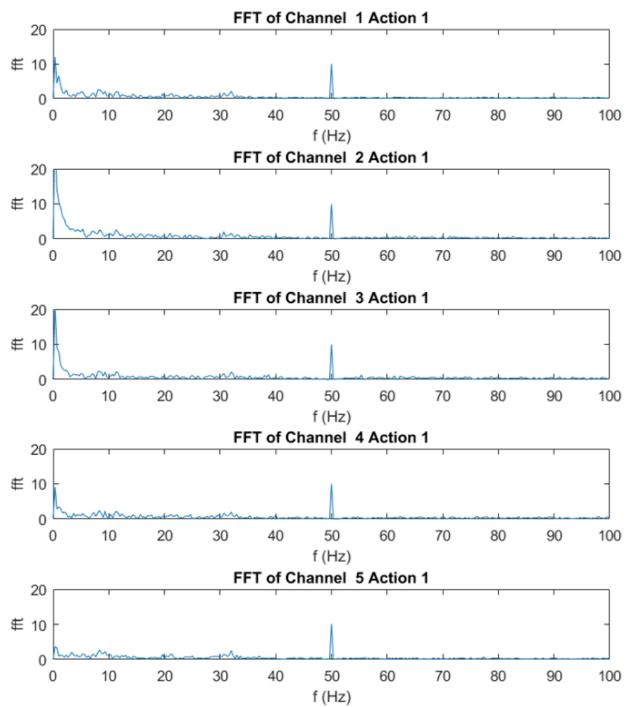
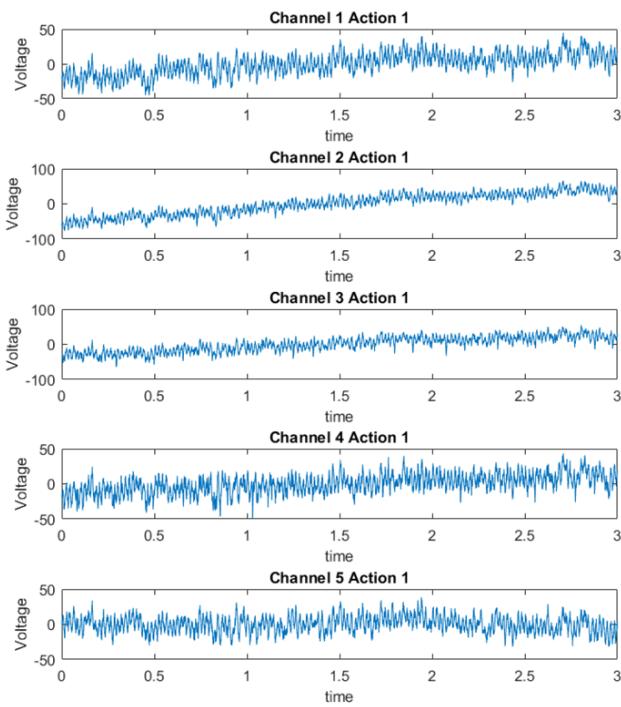
 1x3 [struct](#) with 4 fields

Fields	std	mean	max	min
1	1x64 double	1x64 double	1x64 double	1x64 double
2	1x64 double	1x64 double	1x64 double	1x64 double
3	1x64 double	1x64 double	1x64 double	1x64 double
4				

(۲-۲)

ابتدا شکل سیگنال‌های ۵ کانال اول برای ۳ فعالیت اول را در حوزه زمان و فرکانس رسم می‌کنیم:





همان طور که در سه نمودار فوق مشاهده می‌شود، آنچه در مورد محتوای فرکانسی سیگنال‌های معزی انتظار داشتیم، به شکل دقیق در نمودارهای فوق دیده نمی‌شود، چراکه در فرکانس‌های بالای ۳۰ هرتز نیز مقدار قابل توجهی از محتوای فرکانسی سیگنال‌های معزی مشاهده می‌شود.

در قسمت‌های بعدی، در مورد منشاً این مسئله بحث می‌کنیم.

(۳-۲)

به استناد به <http://www.bem.fi/book/13/13.htm>، در مورد سیگنال‌های EEG می‌توان گفت که محدوده فرکانسی آن‌ها حداً کثر تا حدود 50Hz و بزرگی دامنه نوسان آن در حدود $100\mu V$ هنگام اندازه‌گیری از روح پوست سر و حدود 1mV هنگام اندازه‌گیری از روح سطح مغز است.

بنابراین با توجه به مقدار DC موجود در سیگنال‌های داده شده (که در بخش اول، آن‌ها را حذف کردیم) و نیز محتوای فرکانسی مشاهده شده، توان نویز موجود قابل توجه است و باید در ادامه فیلتر شود.

در مورد نویز DC، یکی از مهم‌ترین عوامل، انقباض عضلات بدن است که هر فرمان انقباضی از مغز به یکی از عضلات، می‌تواند مقداری نویز DC به داده‌های ما اضافه کند.

همچنین در مورد نویزهای غیر DC، مهم‌ترین عامل موجود، نویز برق شهر در فرکانس 50Hz است که با توجه به توان بالای آن و حضور در محل آزمایش، (و با تبدیل بدن انسان به یک آتن) به شکل واضحی روی سیگنال‌های دریافتی قرار می‌گیرد.

(۴-۲)

اولین اقدام ما (که در قسمت اول انجام شد) حذف مقدار DC (با توجه به توضیحات بخش قبل) و ذخیره کردن کل اطلاعات داده شده در ماتریس EEG بود. لازم به ذکر است این کار، ممکن است داده‌های ما را در فرکانس‌های بسیار پایین (کمتر از 0.5Hz) نامعتبر کند که ما این اشکال را می‌پذیریم، چرا که فرکانس‌های بسیار پایین حاوی اطلاعات چندانی نیستند.

در ادامه، کار مهمی که باید انجام شود، حذف نویز فرکانس‌های بالا است. برای این کار، آستانه‌ی 60Hz را انتخاب کردیم و با استفاده از فیلتر پایین‌گذر طراحی شده در قسمت قبل (از یک فیلتر مرتبه ۲۰۰ استفاده کردیم) این فرکانس‌ها را حذف کردیم.

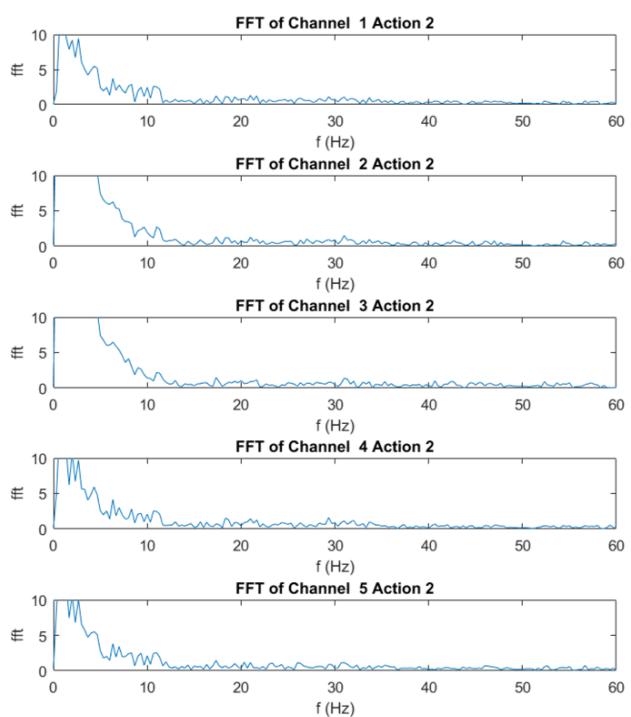
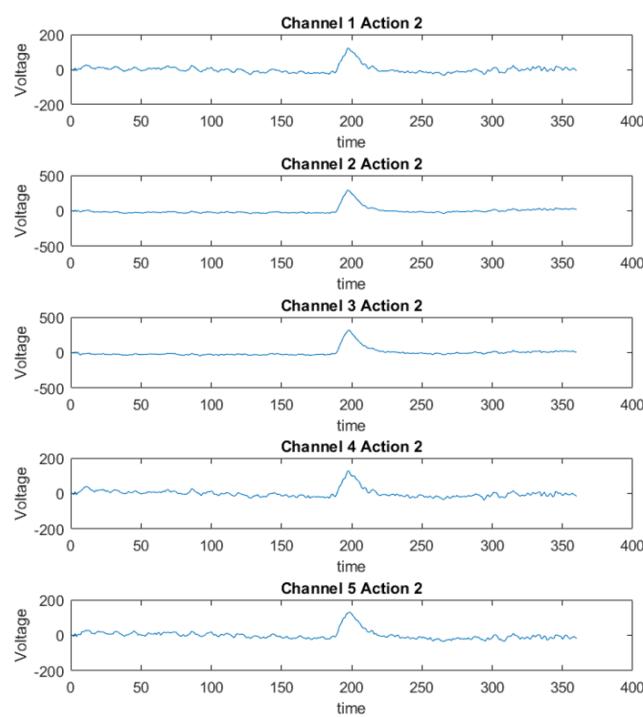
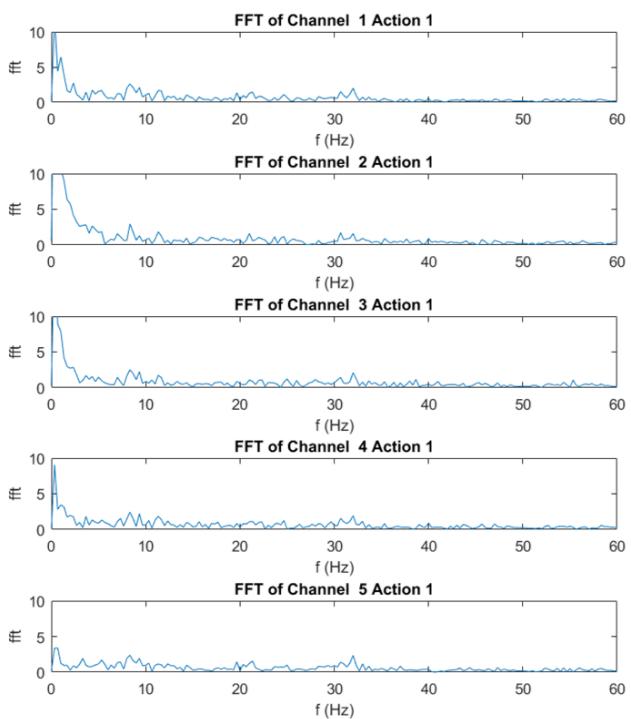
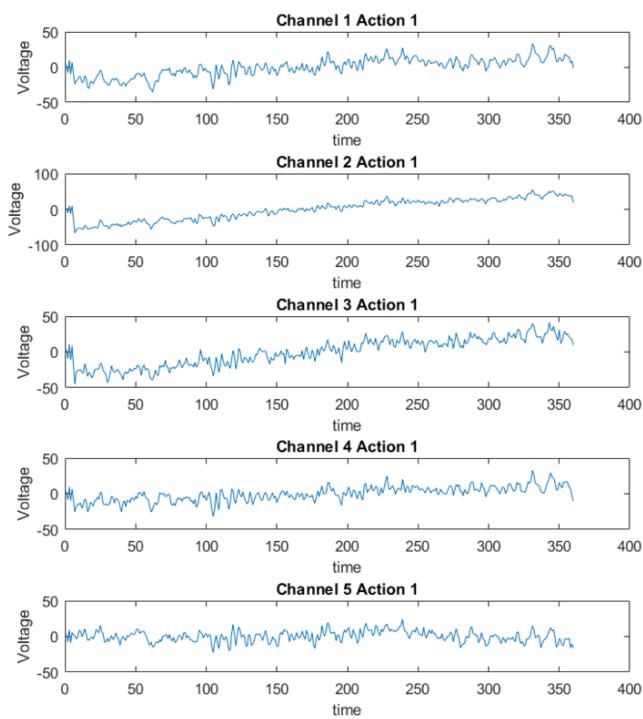
لازم به ذکر است، برای استفاده بهتر از فیلتر پایین‌گذر قسمت قبل، آن را داخل تابعی به نام $LPF(x, N, Fc, Fs)$ قرار دادیم که x سیگنال ورودی، N مرتبه فیلتر، Fc فرکانس قطع، و Fs فرکانس نمونه‌برداری می‌باشد.

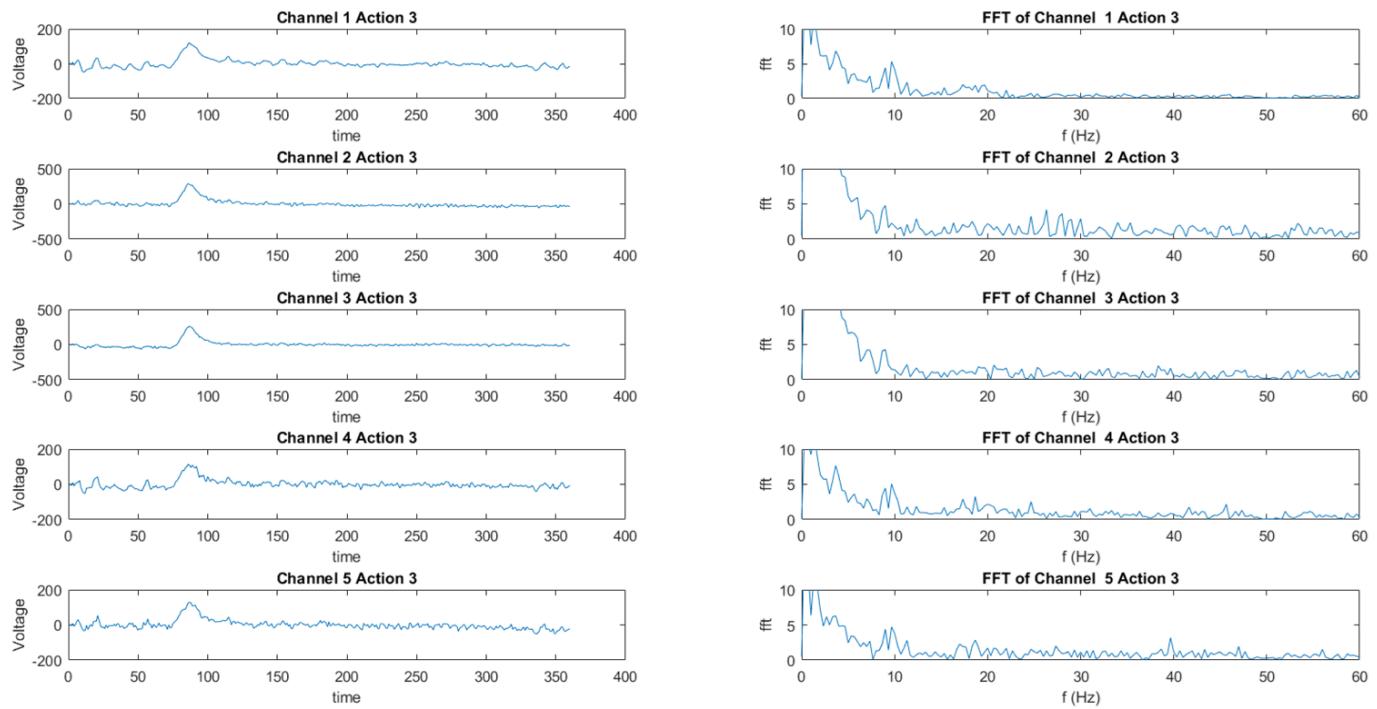
اطلاعات به دست آمده پس از اعمال این فیلتر را در ماتریس filtered ذخیره کردیم.

در ادامه، برای آن که کار با سیگنال ساده‌تر باشد، با توجه به این که اطلاعات بالای 60Hz را حذف کرده‌ایم، عمل down-sampling را انجام دادیم و فرکانس نمونه‌برداری را تا دو برابر بزرگ‌ترین فرکانس موجود کاهش دادیم. به این ترتیب، فرکانس نمونه‌برداری از 2400Hz تبدیل به 120Hz شد. این دیتا را در ماتریس downFiltered ذخیره کردیم. از این پس، با این ماتریس به عنوان دیتای اصلی کار خواهیم کرد.

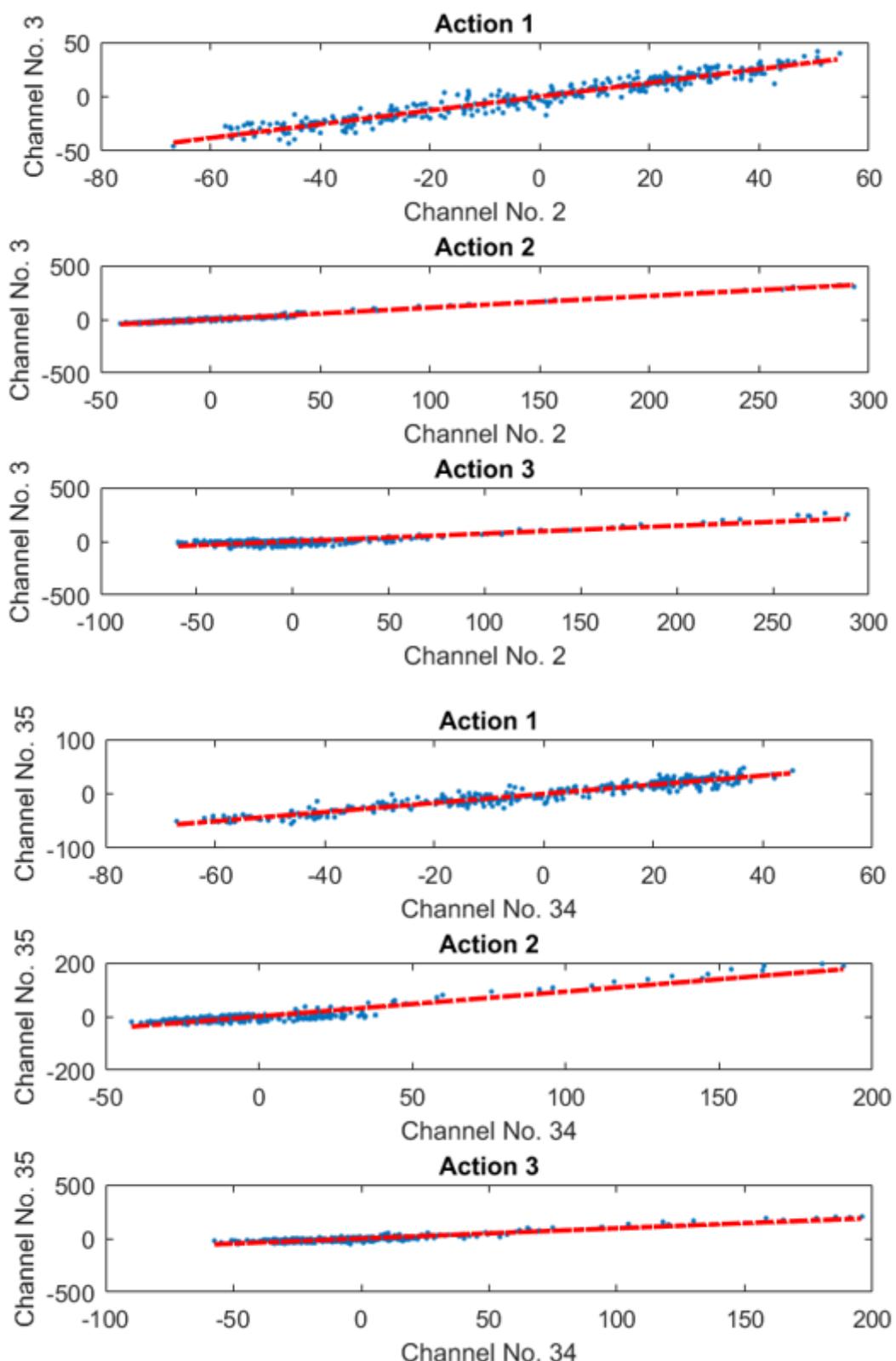
نهایتاً، تنها کار مهمی که باقی می‌ماند، حذف نویز برق شهر است. برای این کار، وارد حوزه فرکانس شدیم و برای تمامی کانال‌ها و فعالیت‌ها، محتوای فرکانسی مربوط به فرکانس 50Hz (می‌دانیم که در حوزه فرکانس، هر فرکانس معادل دو نقطه از محور ω است و باید فرکانس‌های منفی را نیز در نظر بگیریم) را پاک کرده و آن را با میانگین دو فرکانس مجاورش جایگزین کردیم.

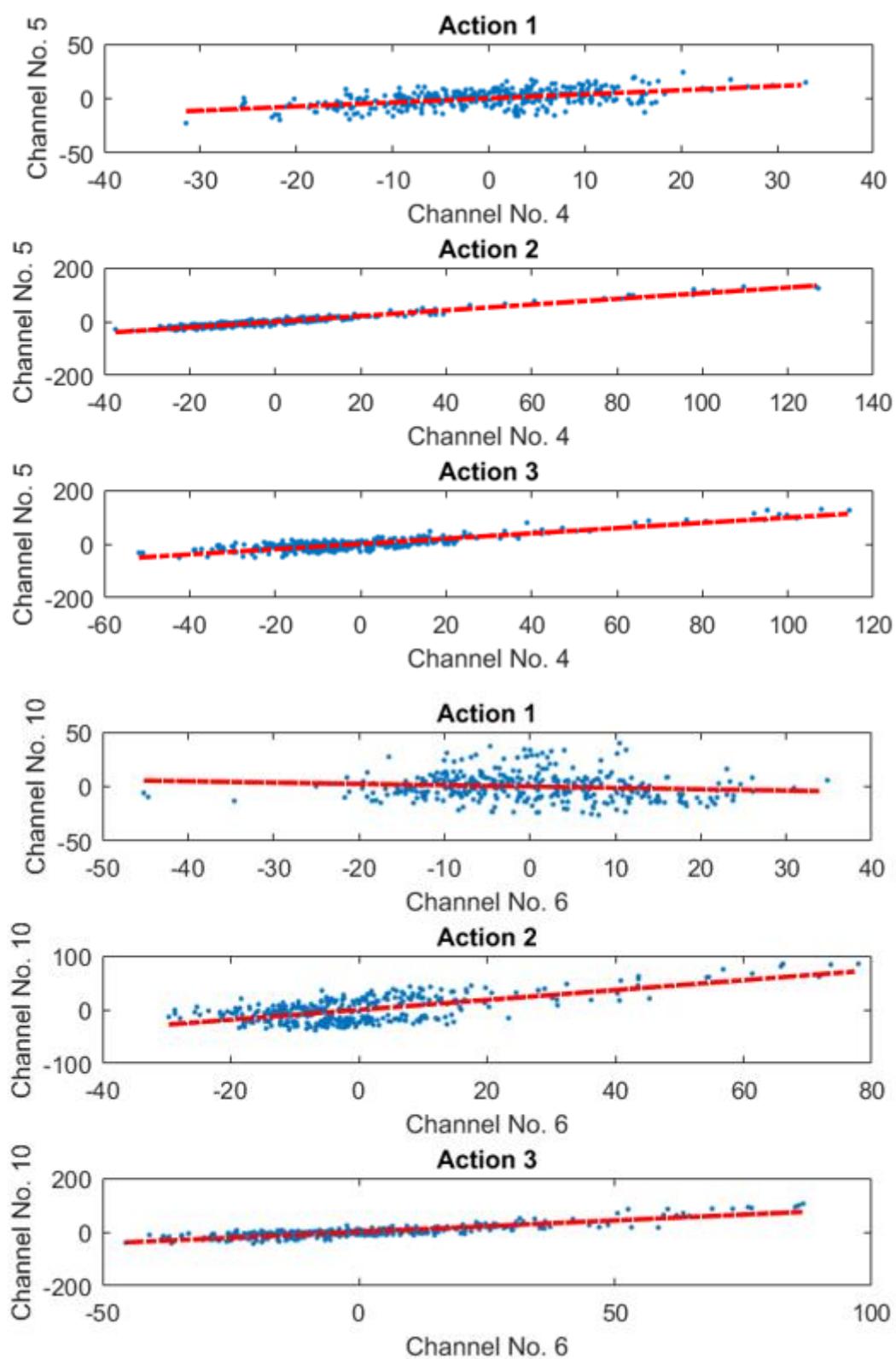
به این ترتیب، فیلترهای مورد نیاز همگی اعمال شدند و آماده‌ی رسم مجدد نمودارها هستیم. تنها موردی که هنوز از حل شدن قطعی آن اطمینان نداریم، وجود برخی داده‌های پرت است. البته بخش قابل توجهی از این داده‌ها احتمالاً در اثر فیلترهای فرکانسی ما از بین رفته‌اند، با این حال، برای حصول اطمینان از این مسئله، دامنه‌ی تغییرات سیگنال‌ها را بررسی کردیم تا ببینیم آیا سیگنال مربوط به یکی از الکترودها و فعالیت‌ها، دامنه‌ای بیش از اندازه بزرگ‌تر از سایرین دارد یا خیر. نتیجه این بررسی منفی بود و این یعنی احتمالاً داده‌ی پرتی نداریم.

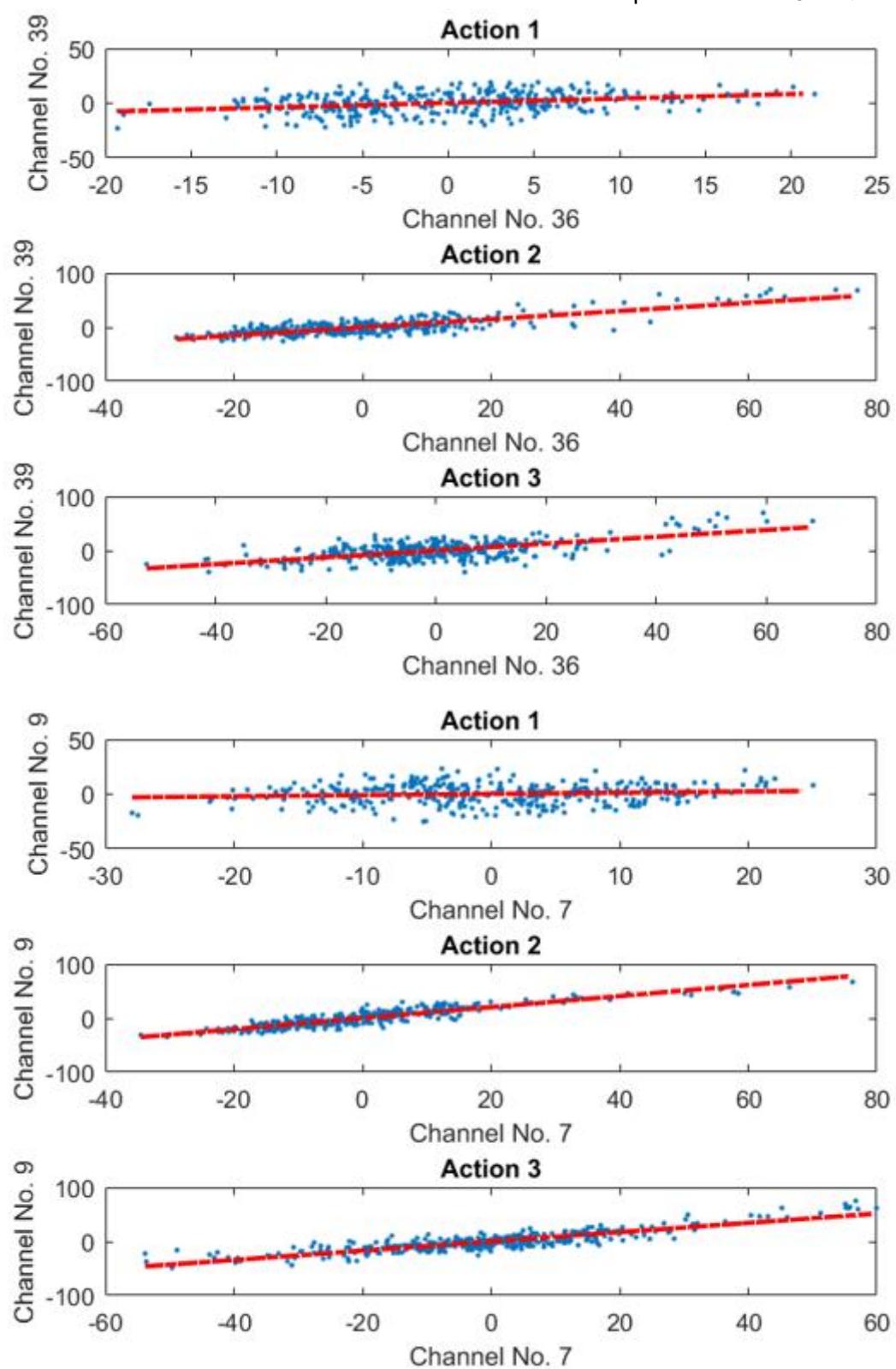


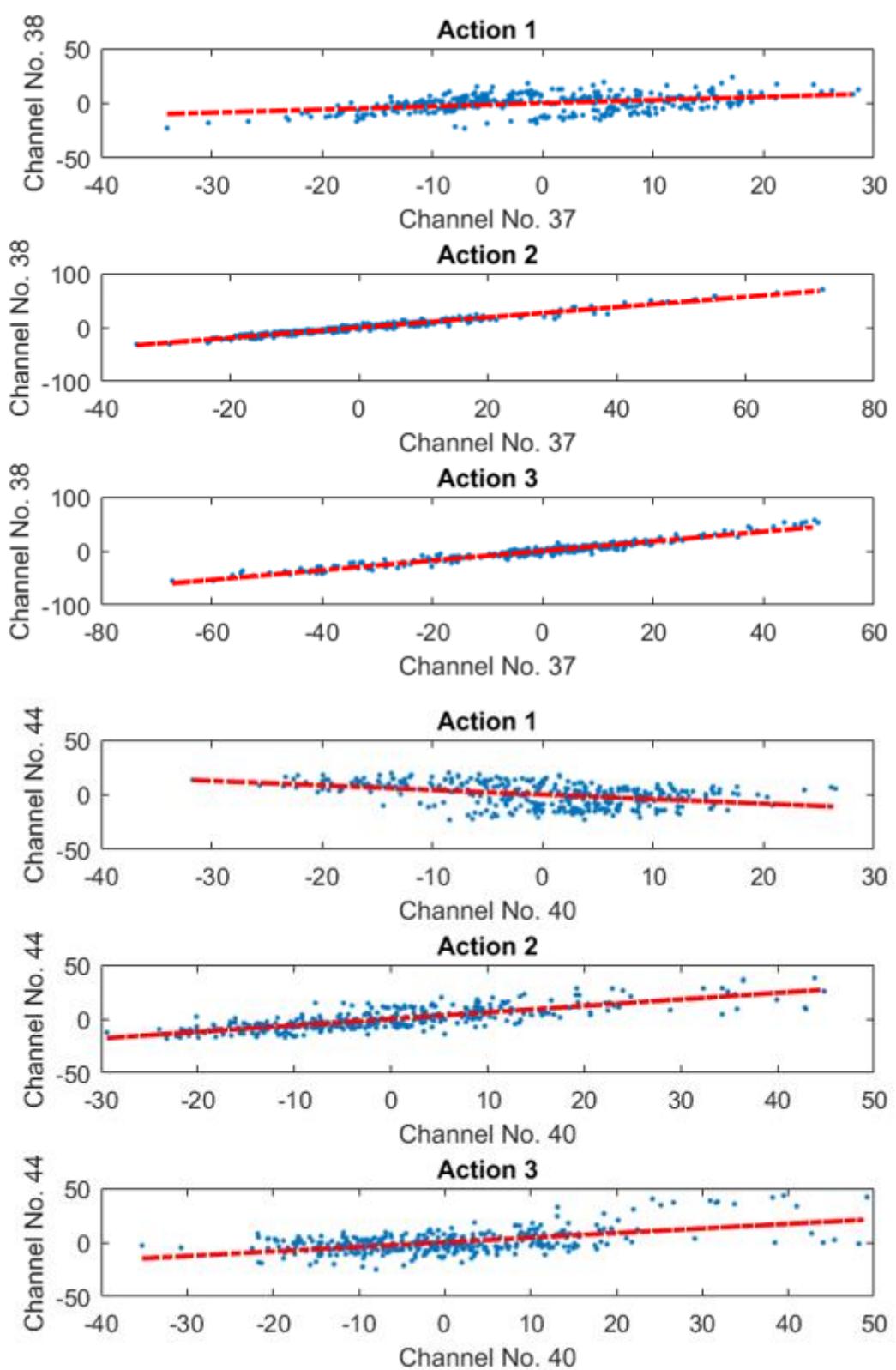


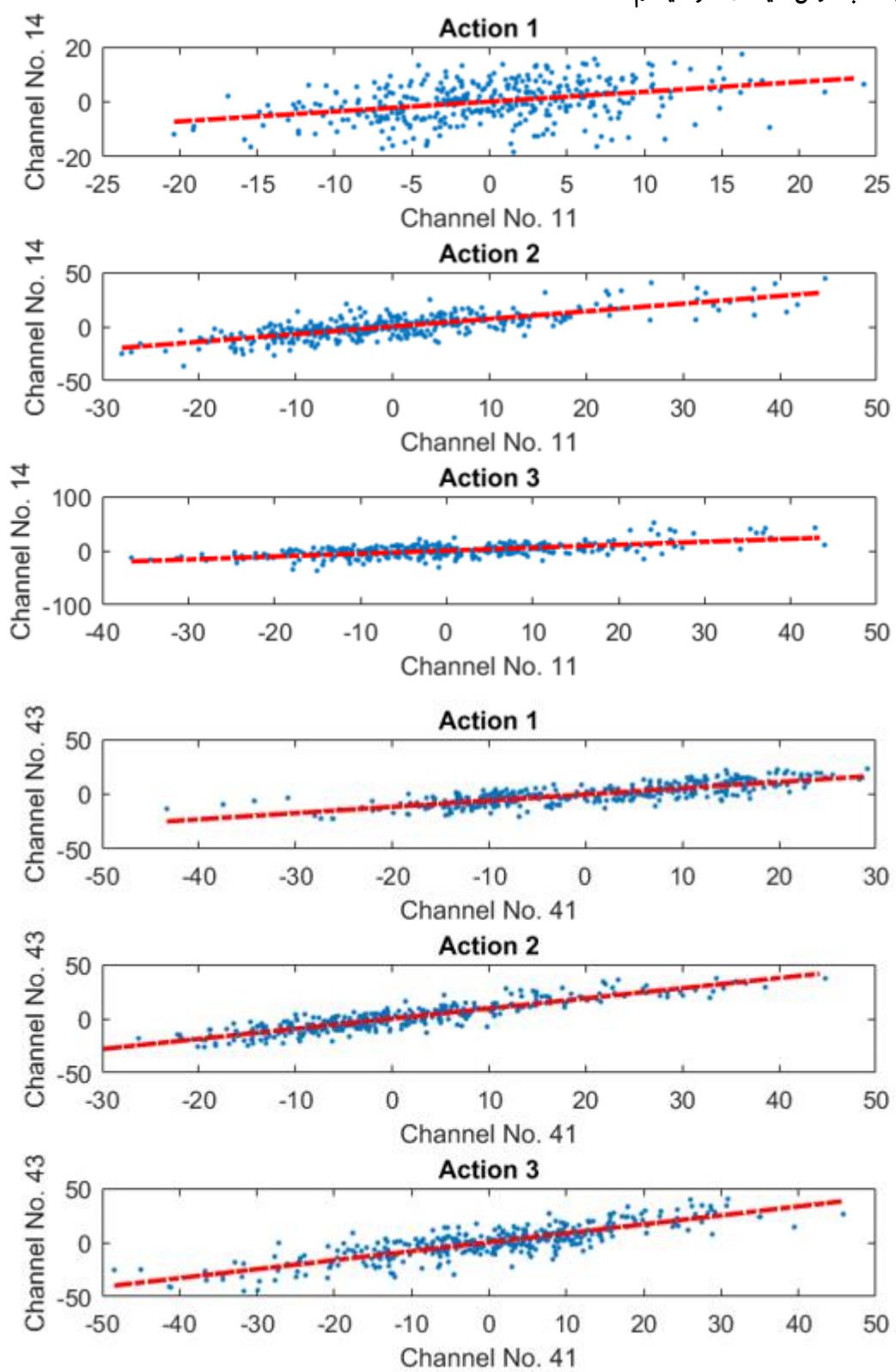
می‌بینیم که شکل سیگنال‌ها در حوزه زمان تمیزتر شده و همچنین با توجه به محدوده دامنه سیگنال‌ها، به نظر داده‌ی پرتوی وجود ندارد.

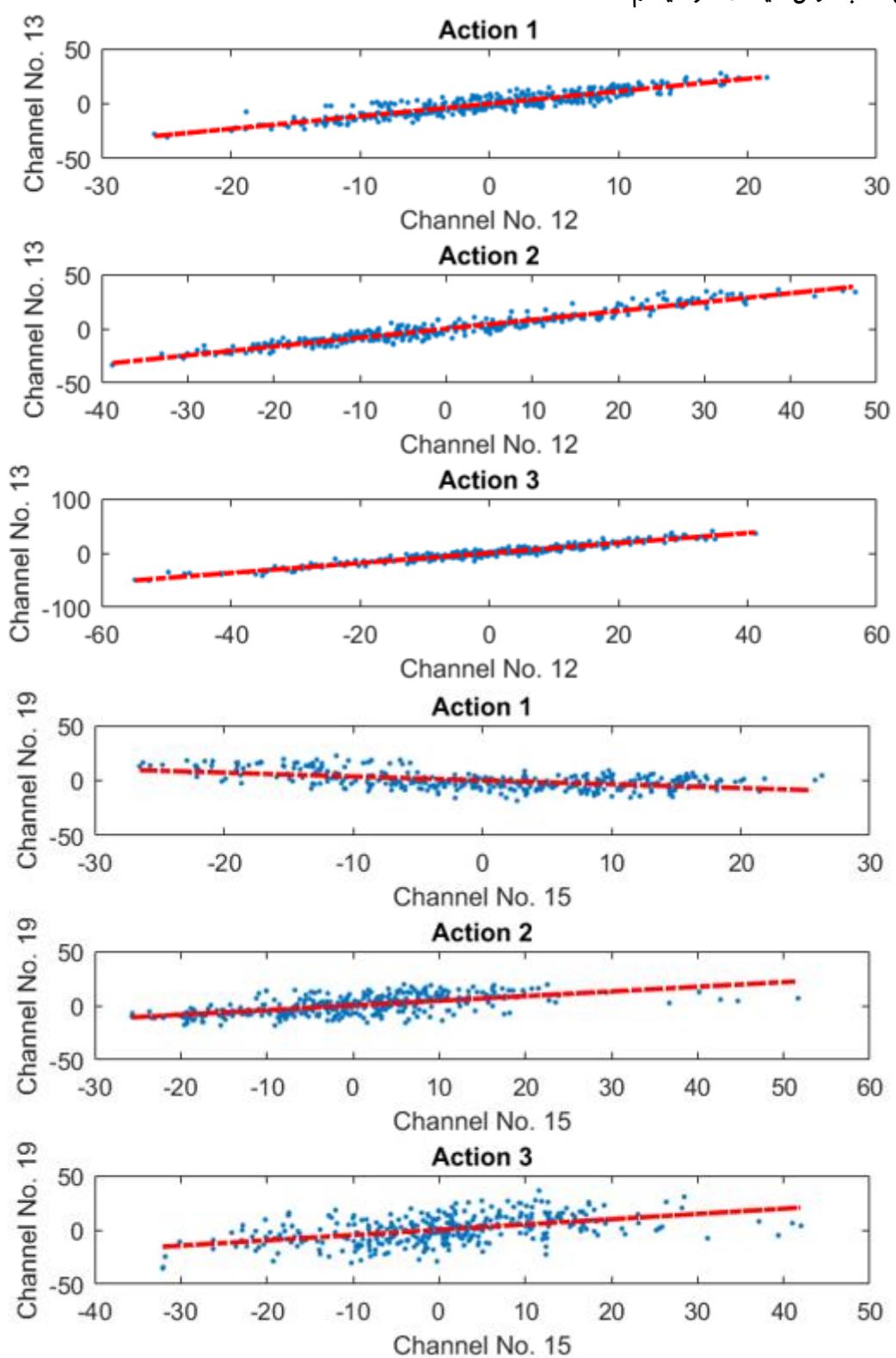


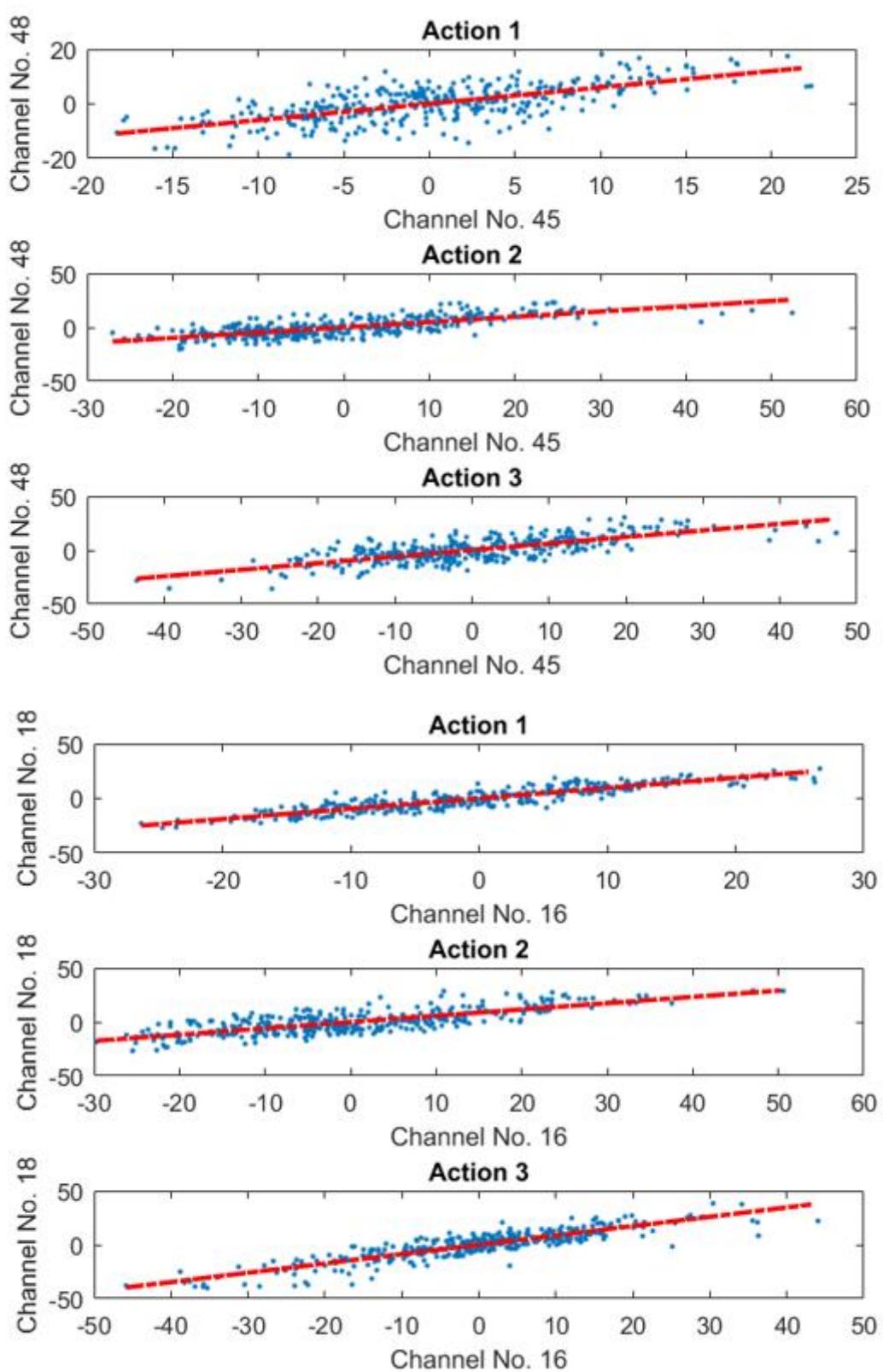


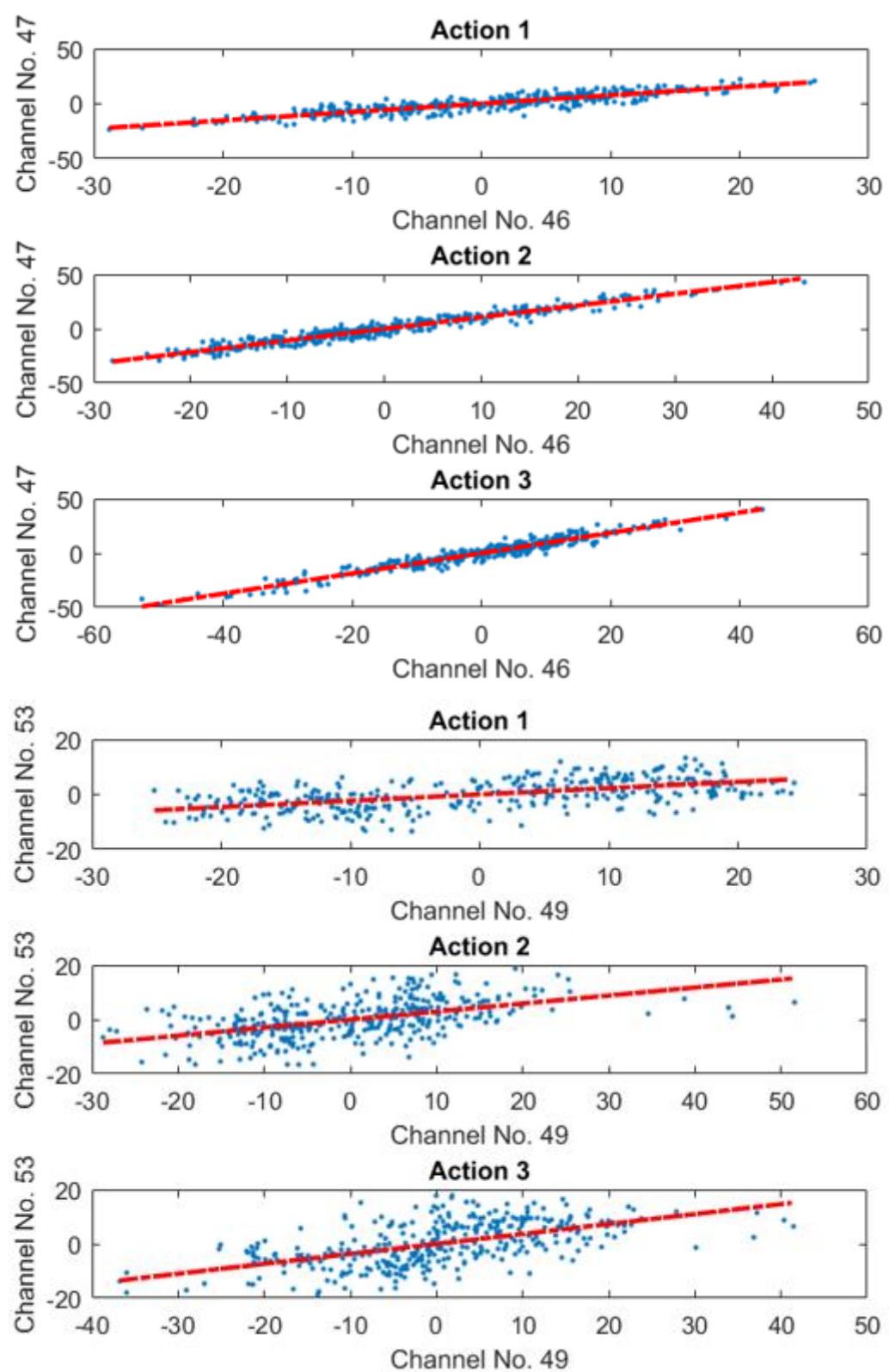


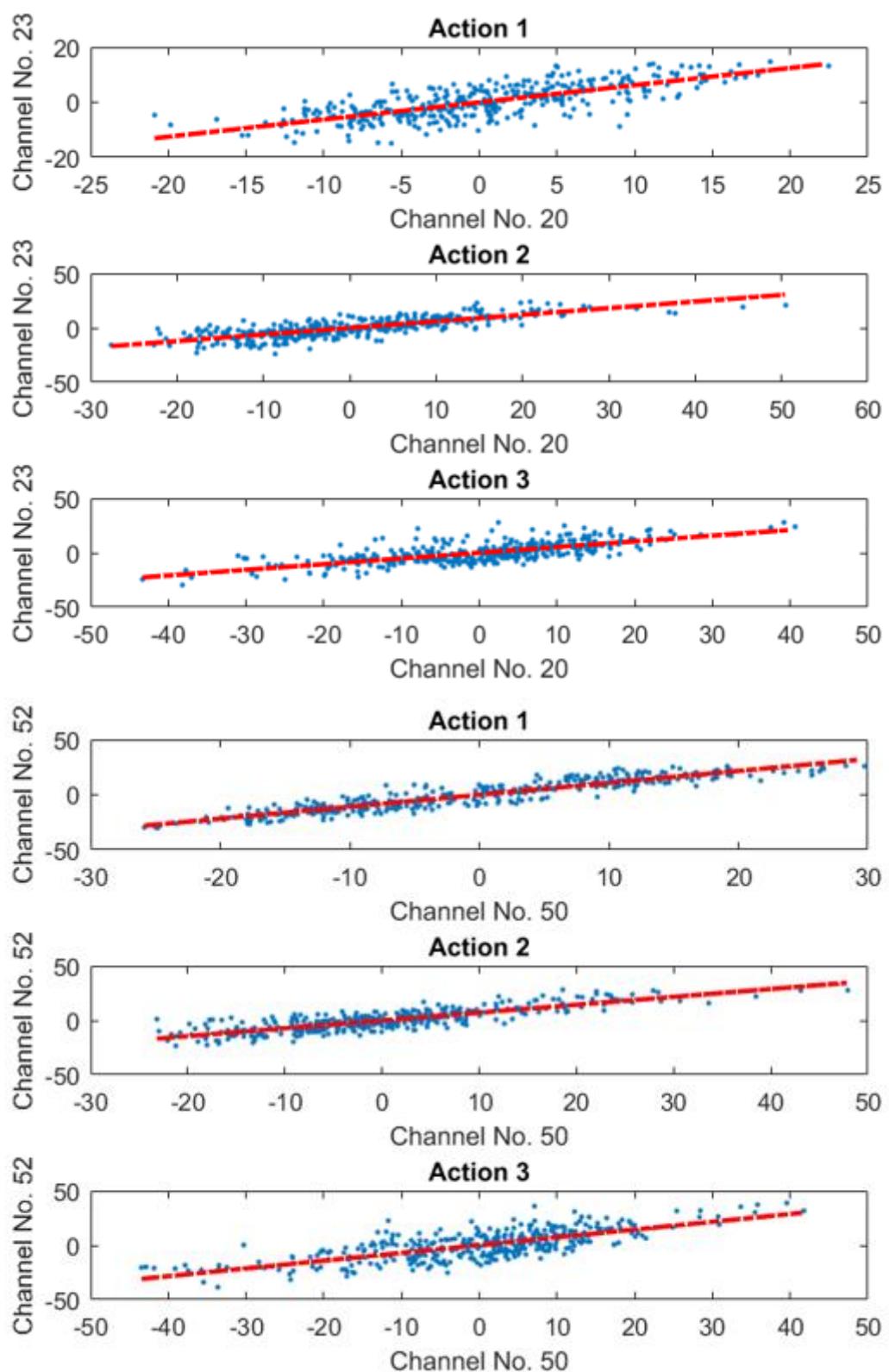


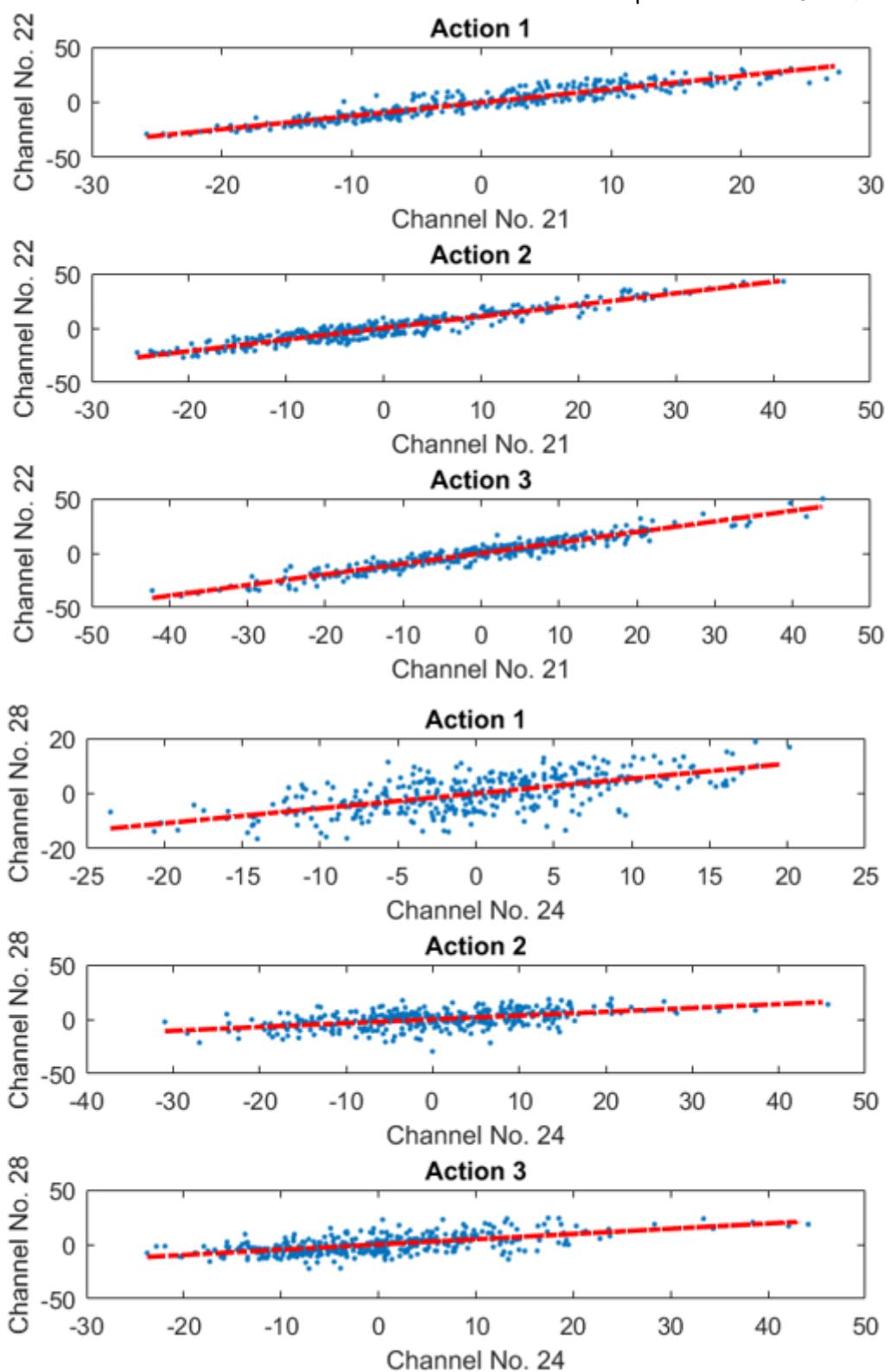


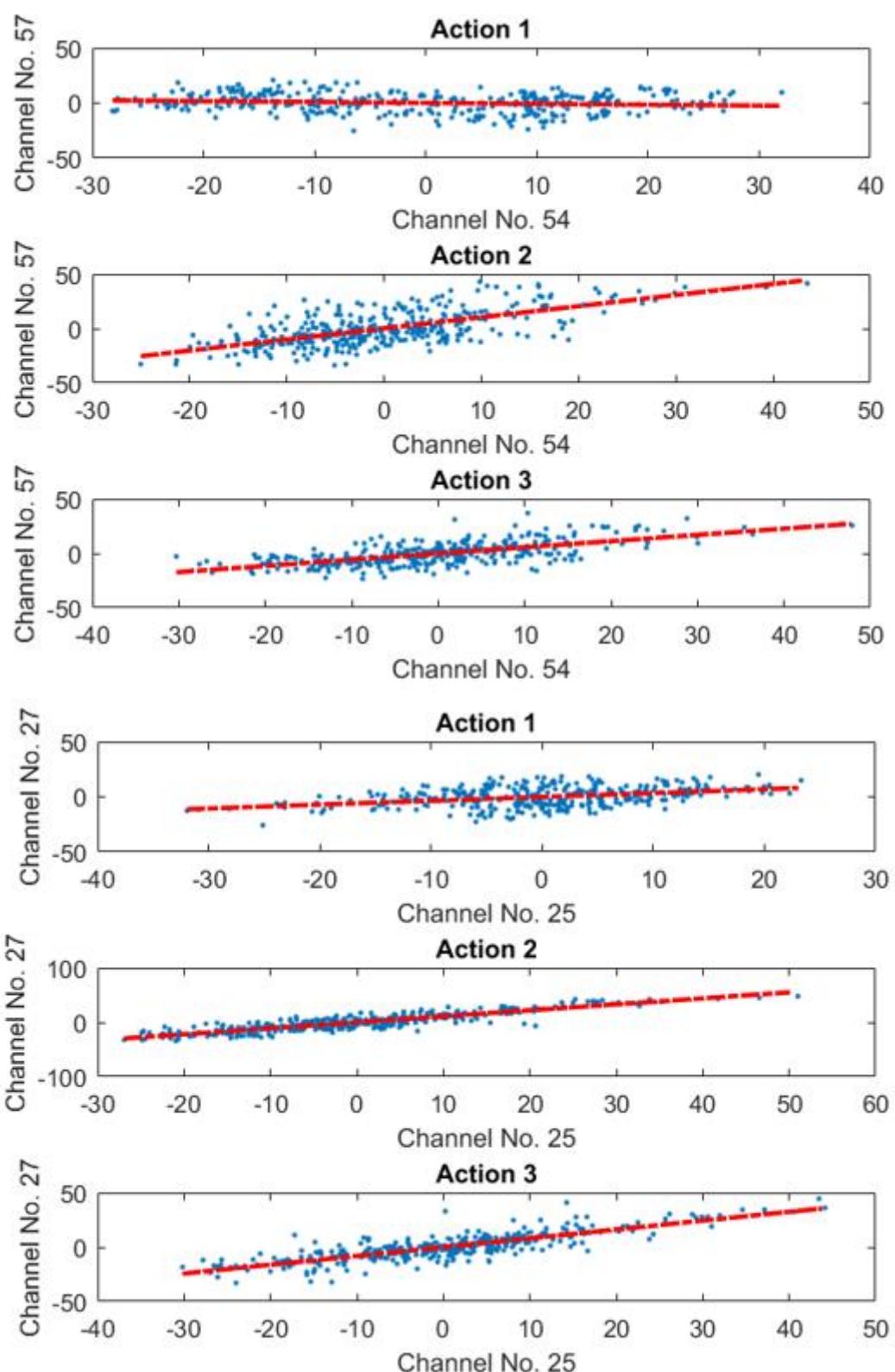


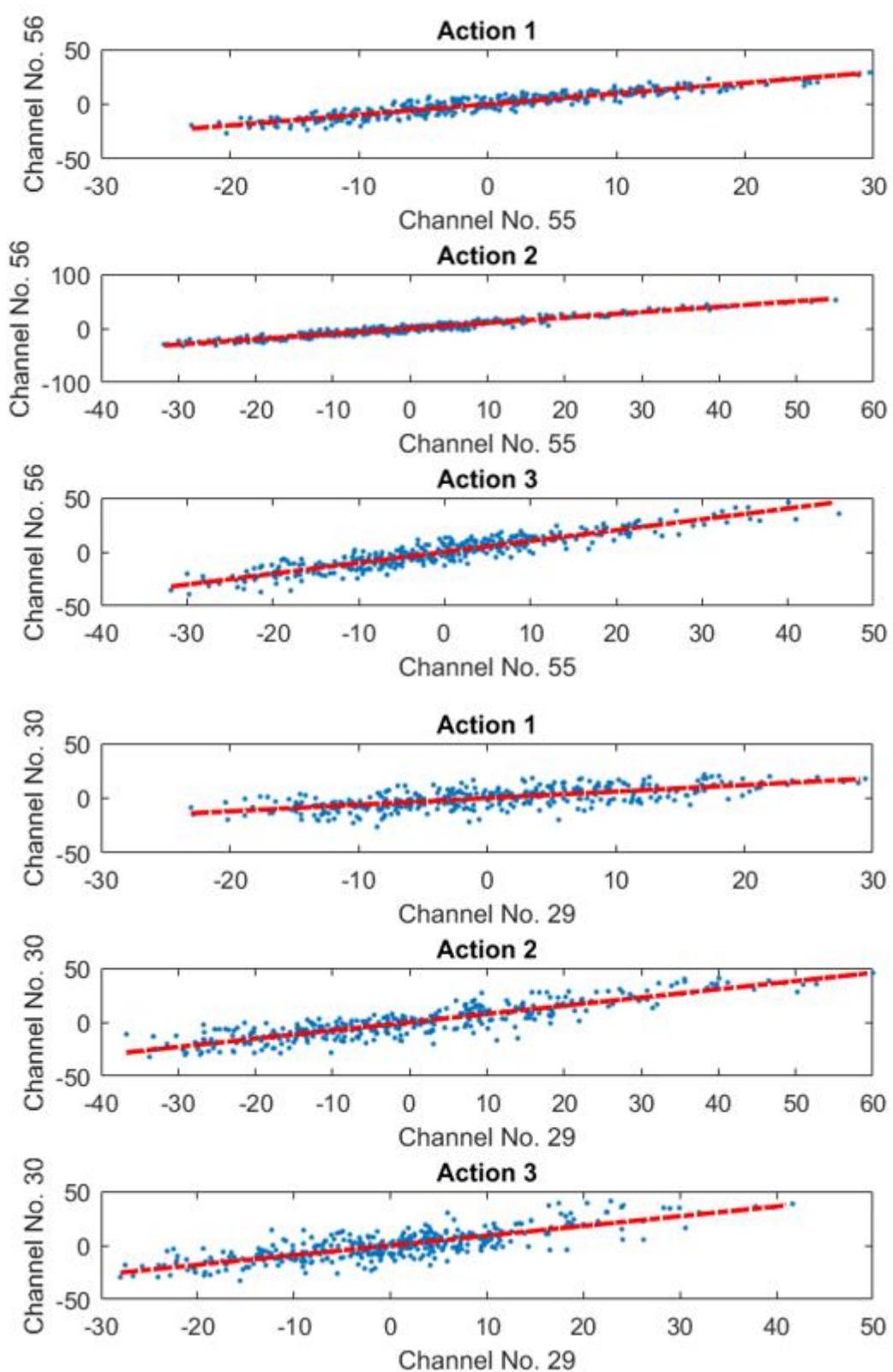


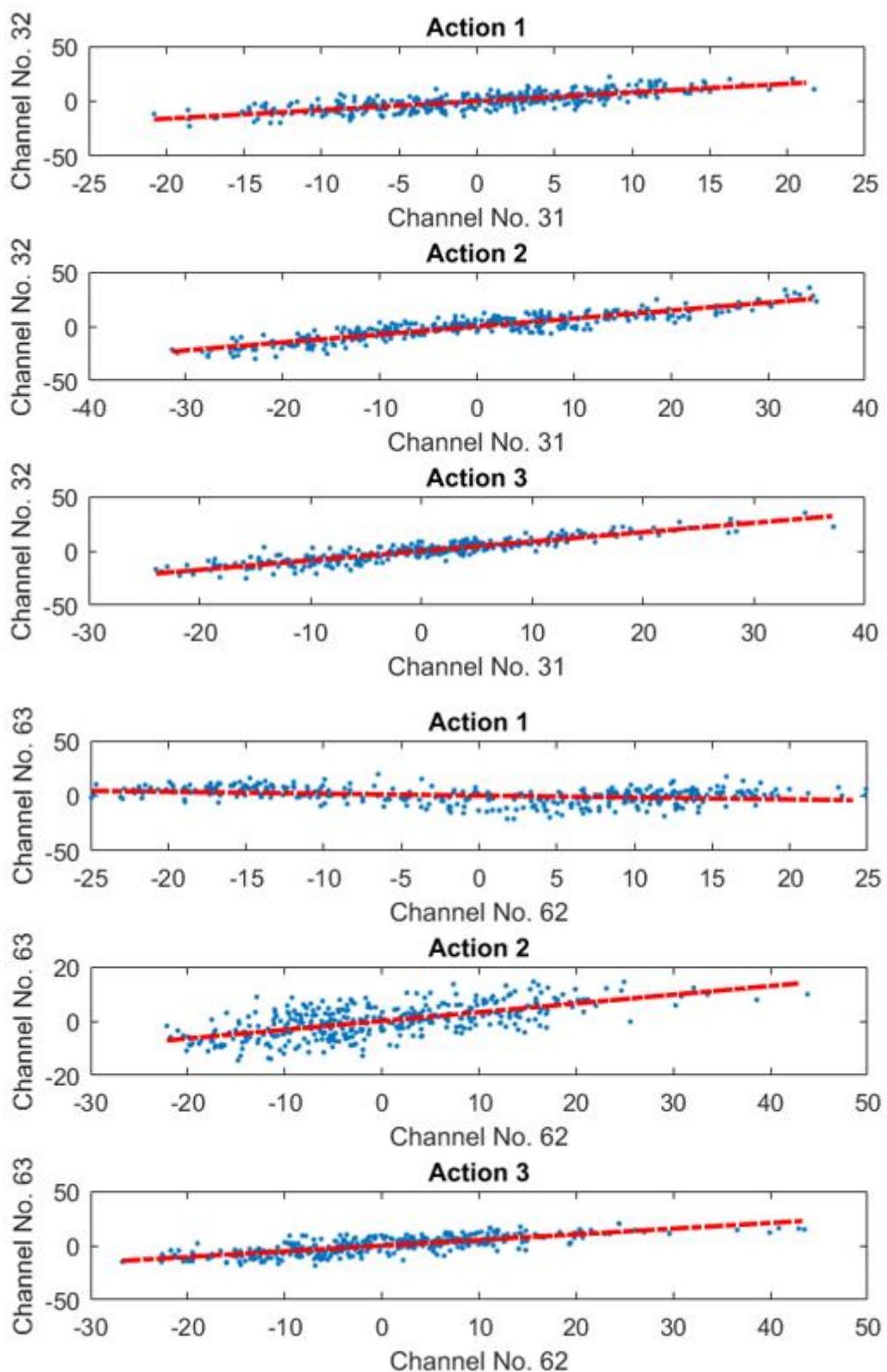






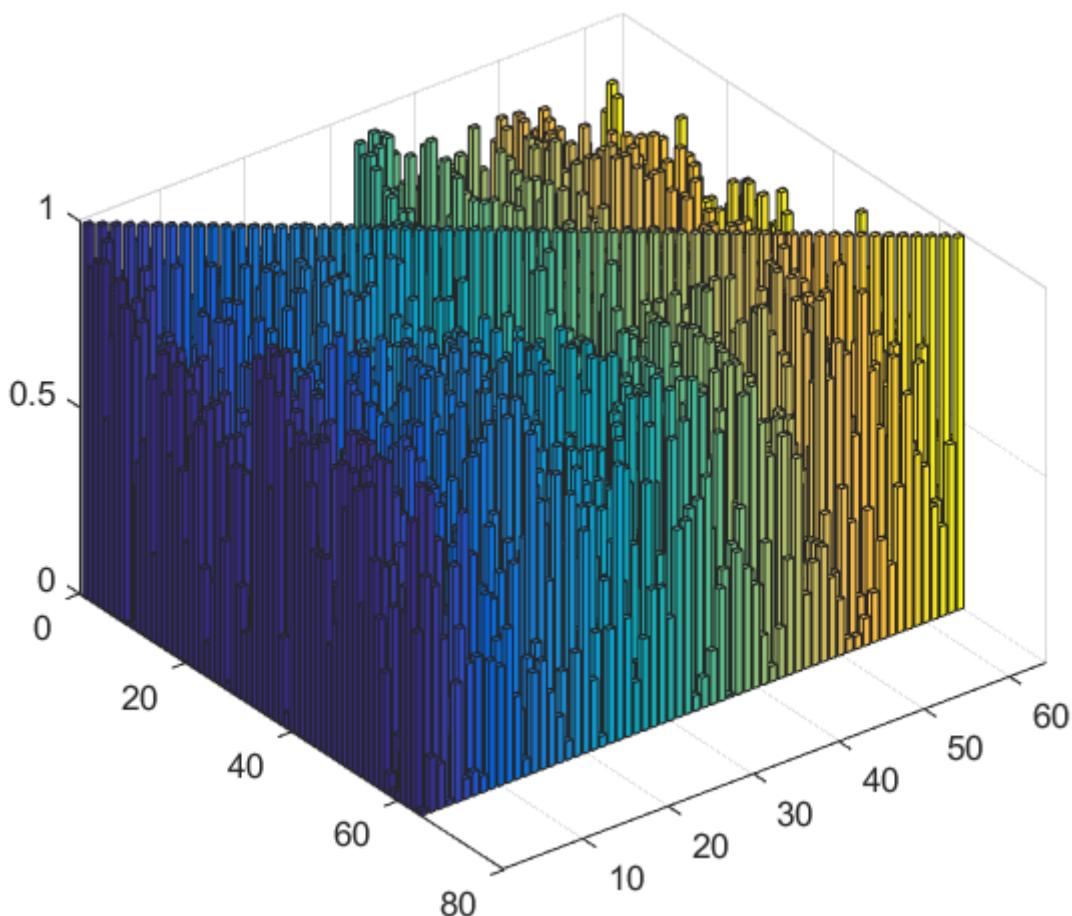




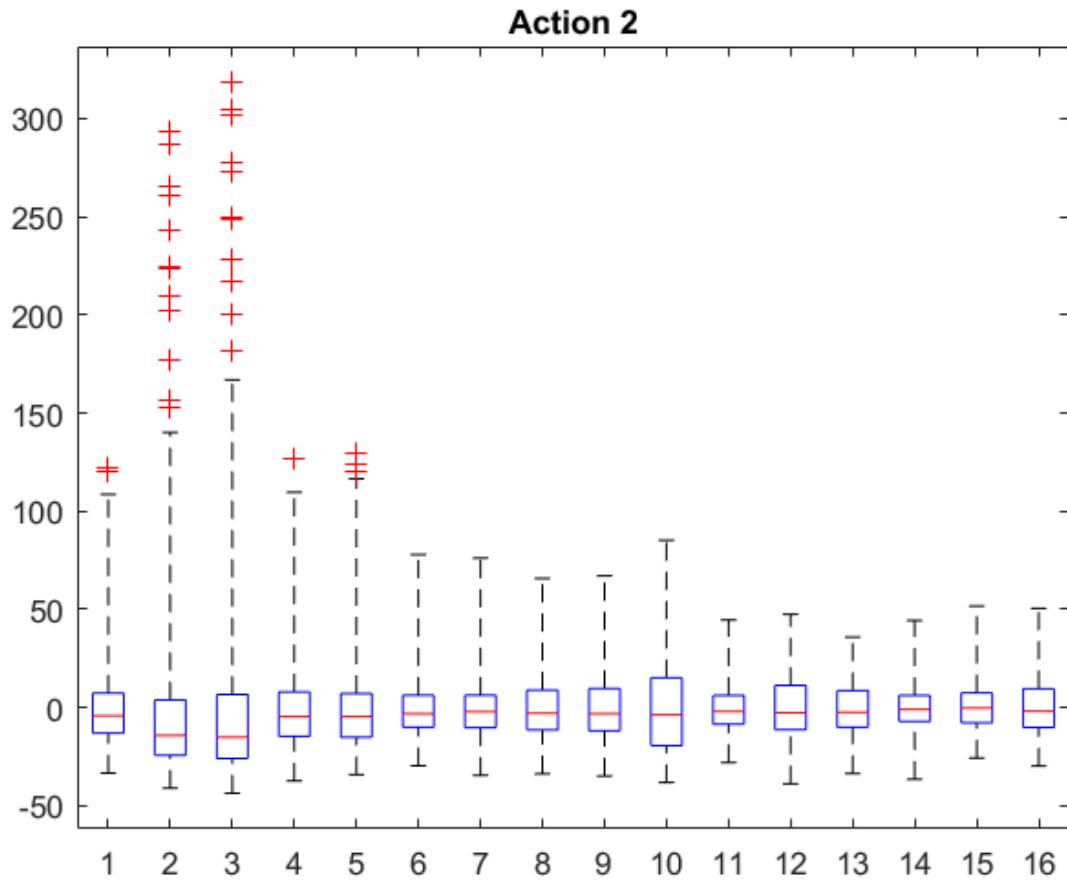
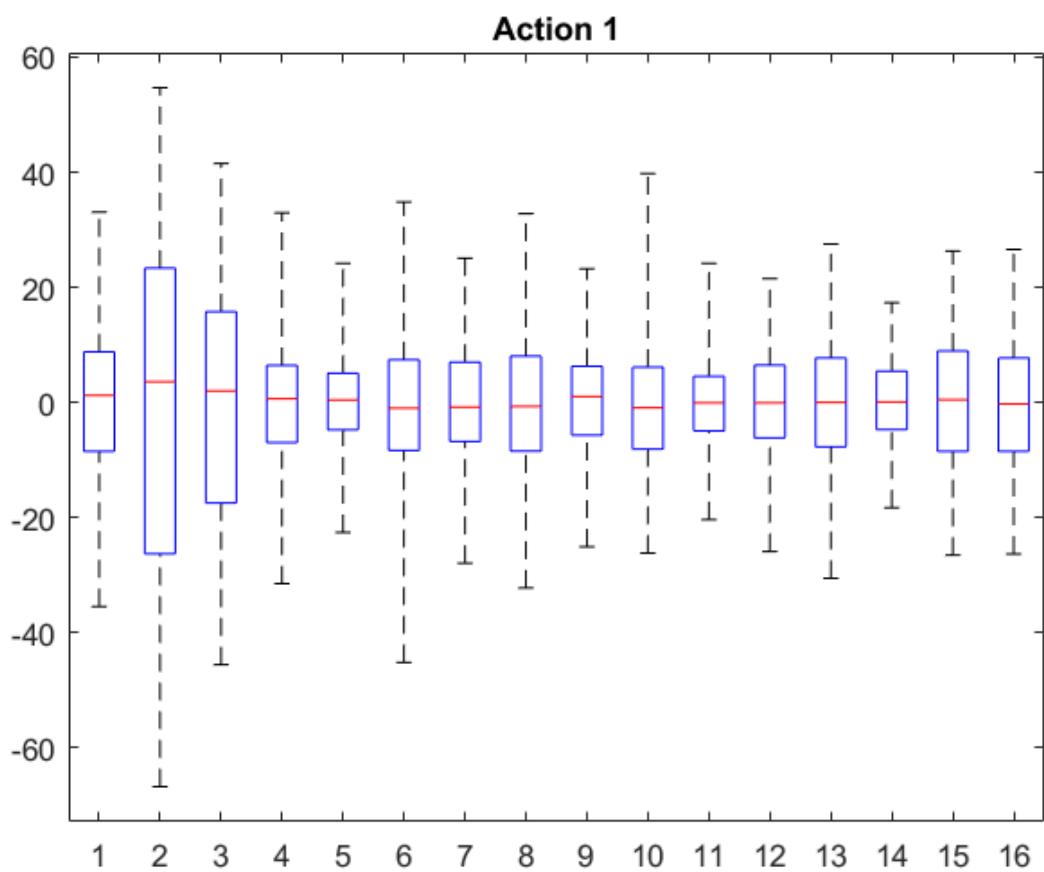


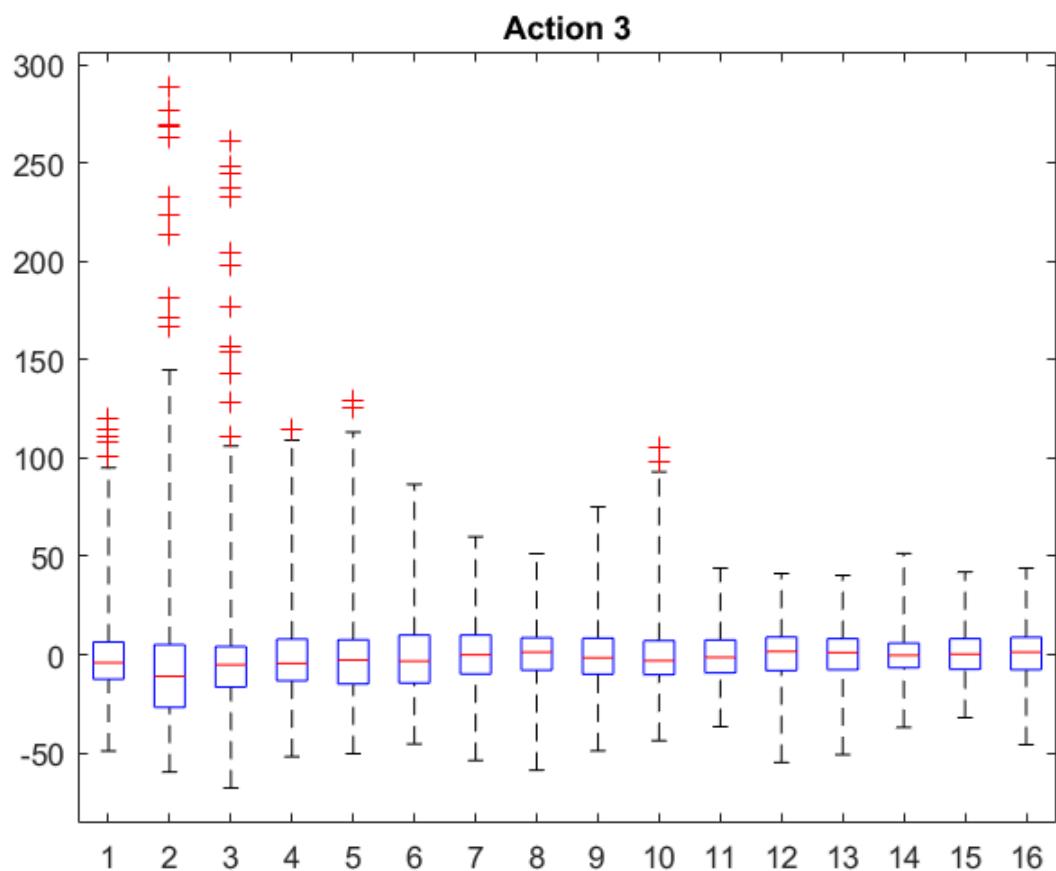
مشاهده می‌شود که رفتار این دو الکترود وابستگی بالایی به هم دارند، به گونه‌ای که در اکثر این نمودارها، شکل حاصل به تقریب خوبی حول یک خط توزیع شده است. اگر به شکل ایده‌آل، نمودار دقیقاً به یک خط تبدیل شود، دو الکترود کاملاً به هم وابسته هستند، به گونه‌ای که با داشتن اطلاعات یکی، می‌توانیم اطلاعات دیگری را به تمامی استخراج کنیم.

ماتریس خواسته شده را تشکیل دادیم و در ماتریس cor_mat ذخیره کردیم. نمودار زیر، مقادیر قدر مطلق ماتریس کوریلیشن را نشان می‌دهد:



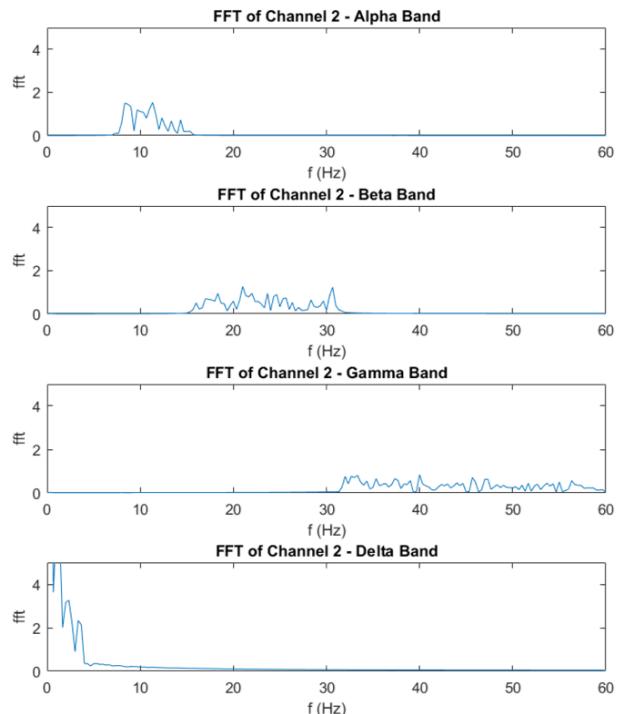
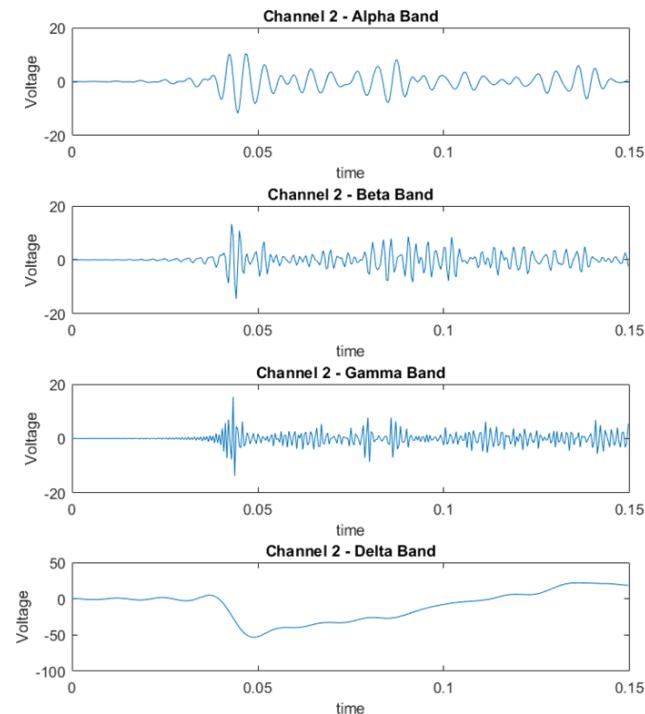
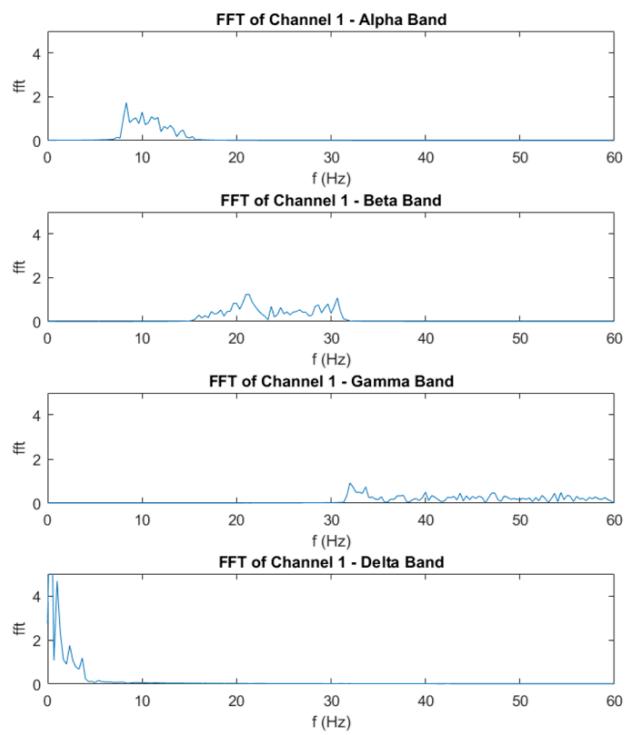
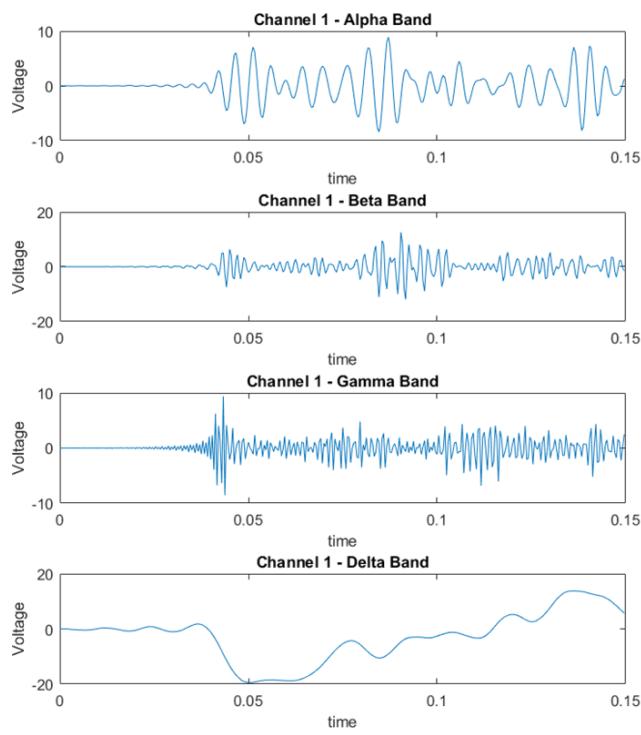
با توجه به اطلاعاتی که تا به اینجا به دست آورده‌ایم، پاسخ دقیق به این سؤال که «کدام الکترودها همبستگی بیشتری دارند؟» را نمی‌توانیم به شکل دقیق و حساب شده بیان کنیم، اما با در نظر گرفتن نتایج قسمت قبلی، و بررسی درایه‌ی ماتریس فوق، می‌توان ادعا کرد الکترودهایی که محل آن‌ها روی سرنزدیک به هم، یا متقارن نسبت به محور تقارن عمودی وسط سر هستند، در مقایسه با سایرین همبستگی بیشتری دارند.

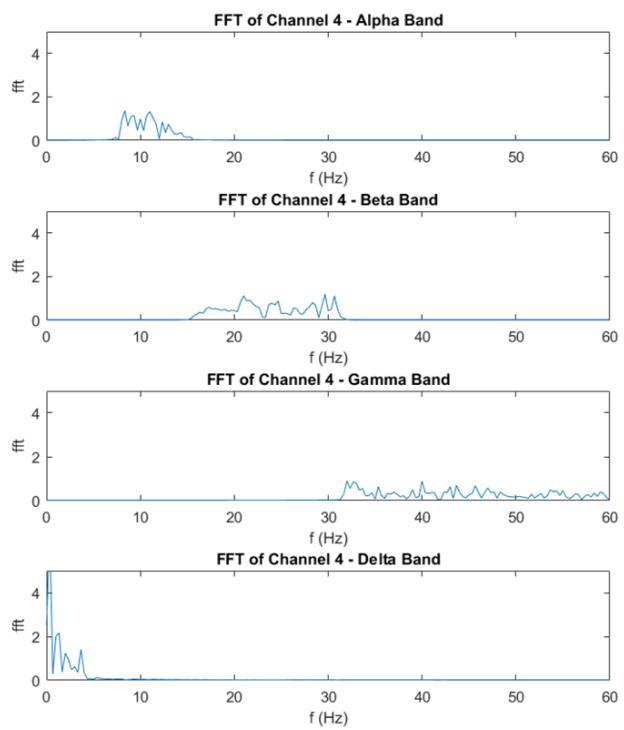
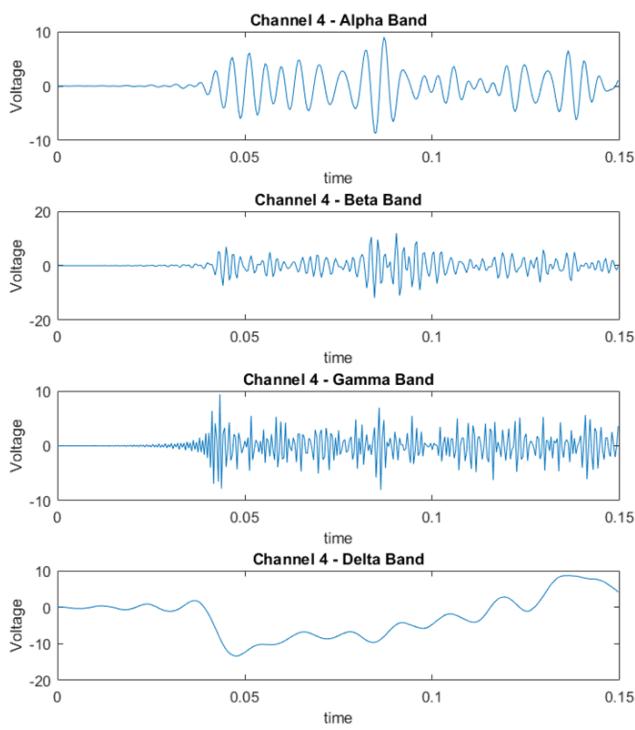
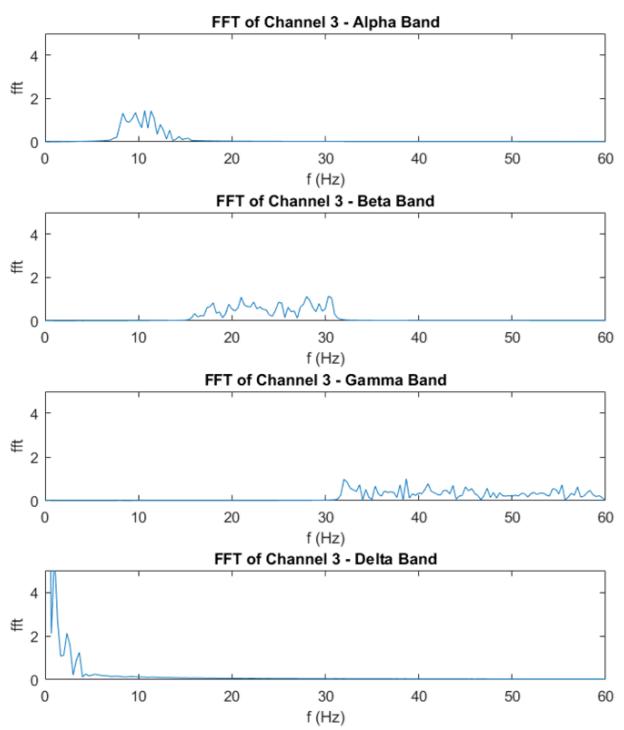
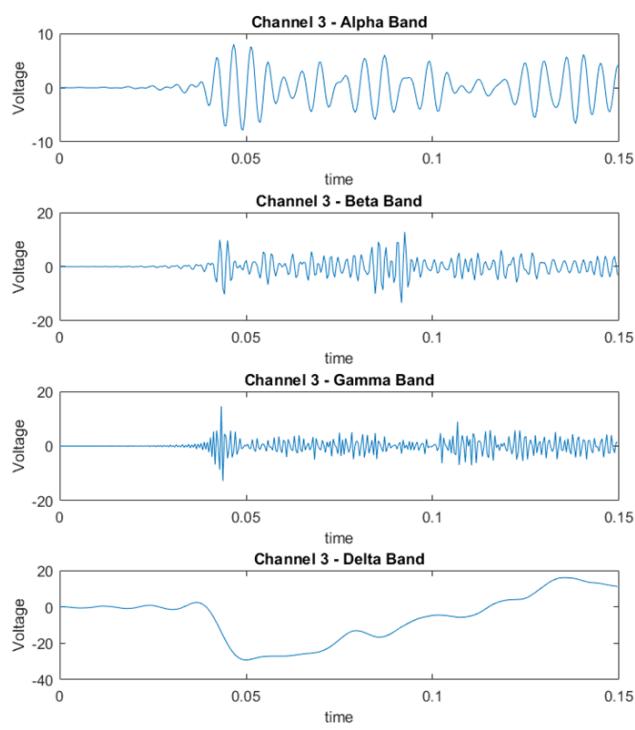




در سه نمودار فوق، پراکندگی دیتای مربوط به ۱۶ الکترود اول و ۳ فعالیت اول را در قالب boxplot مشاهده می‌کنید.

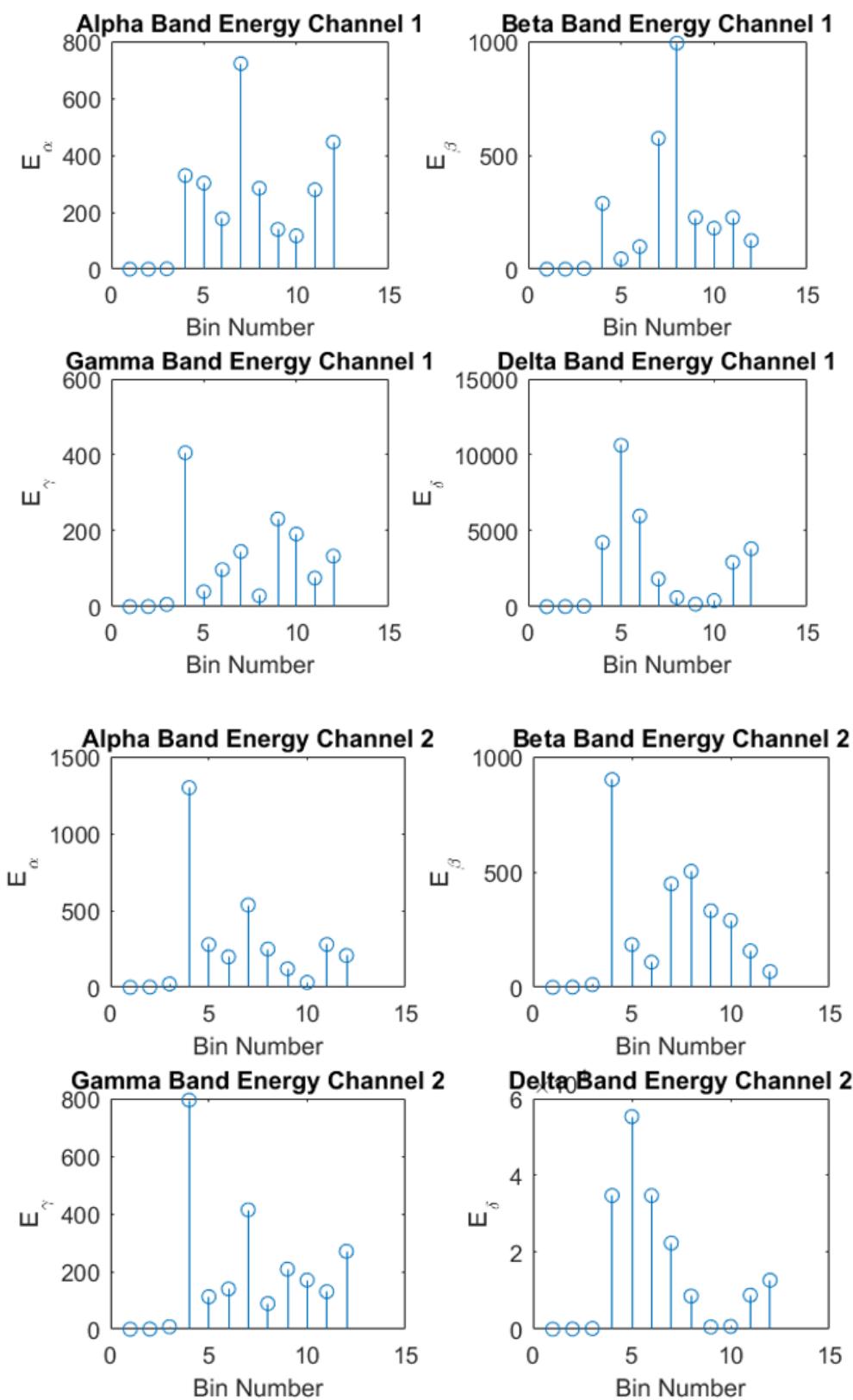
اطلاعات باندهای فرکانسی در فایل bandsData.mat به پیوست ارسال می‌گردد. همچنین نمودارهای باندهای فرکانسی (در حوزه زمان و فرکانس) مربوط به فعالیت ۱ از کانال‌های ۱ تا ۴ را در ادامه مشاهده می‌کنید: (هر واحد محور زمان، ۲۰ ثانیه است)

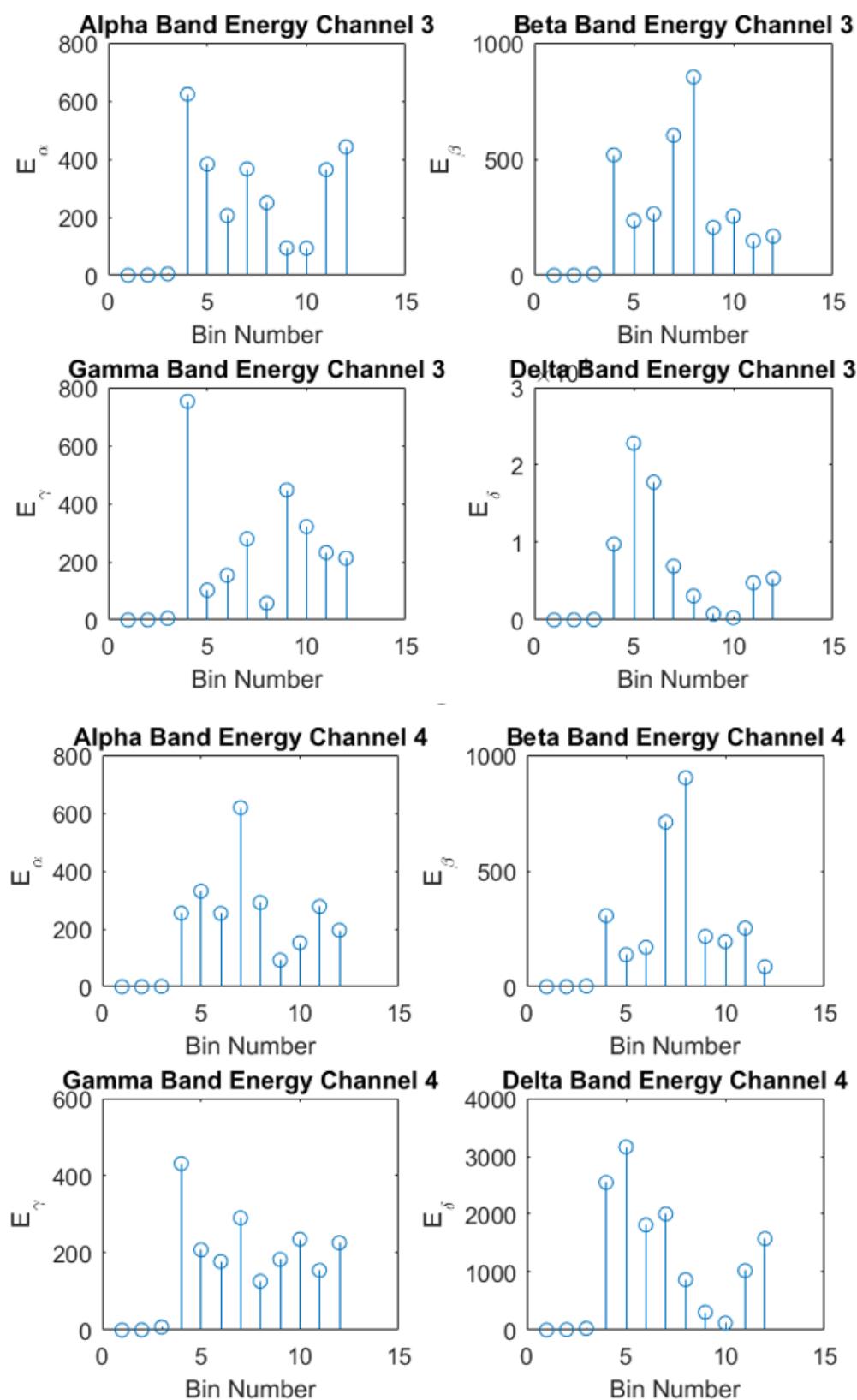




در این بخش، به دلیل این که منظور دقیق سؤال را متوجه نشدیم و دو استنباط مختلف داشتیم، هر دو را پیاده‌سازی کردیم: استنباط اول) پنجره‌های زمانی باید فیکس باشند، به این ترتیب کل سیگنال که ۳ ثانیه زمان دارد، به ۱۲ پنجره تقسیم می‌شود.

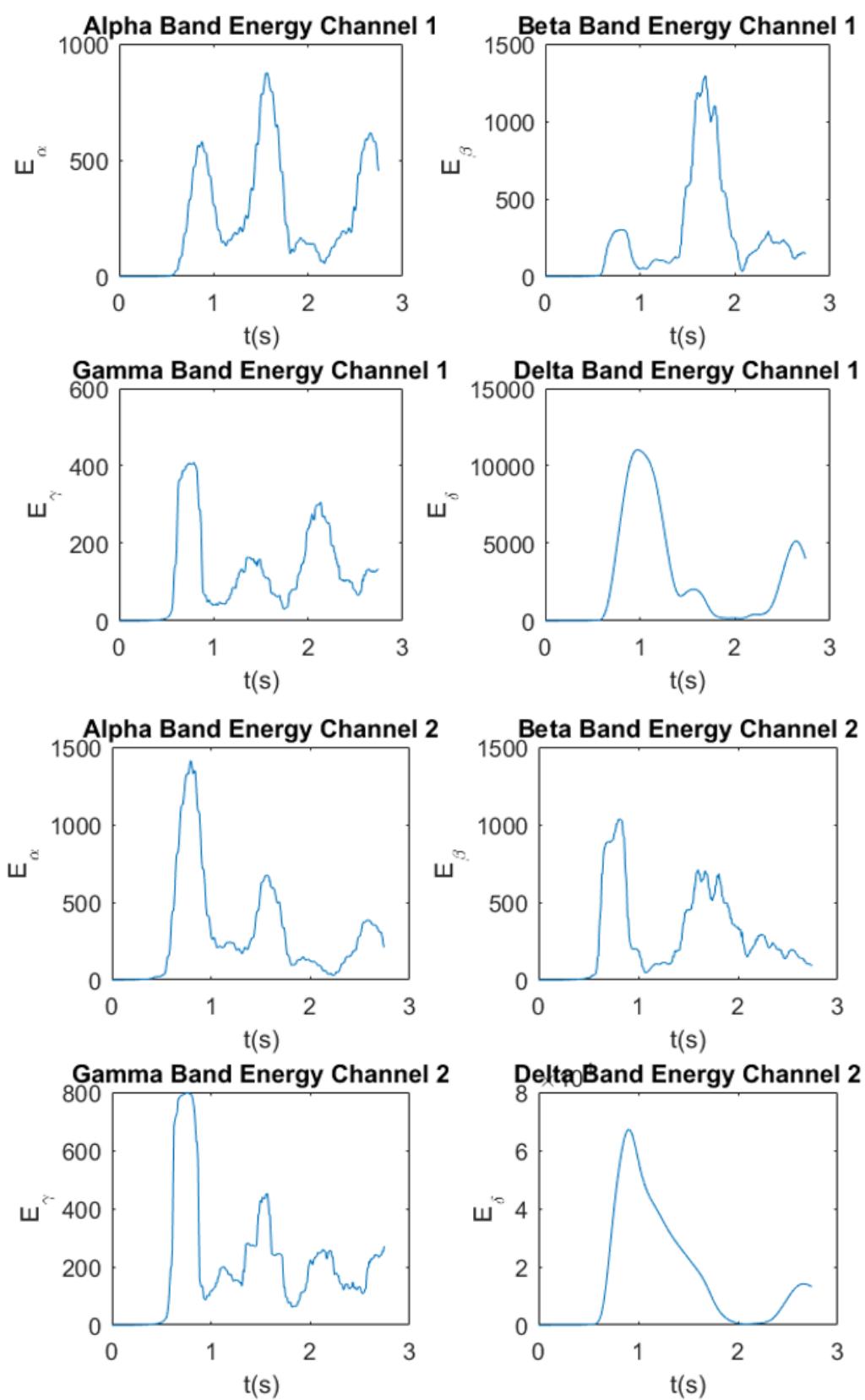
در ادامه، نمودارهای انرژی باندهای مختلف را در این پنجره‌های زمانی مشاهده می‌کنید:

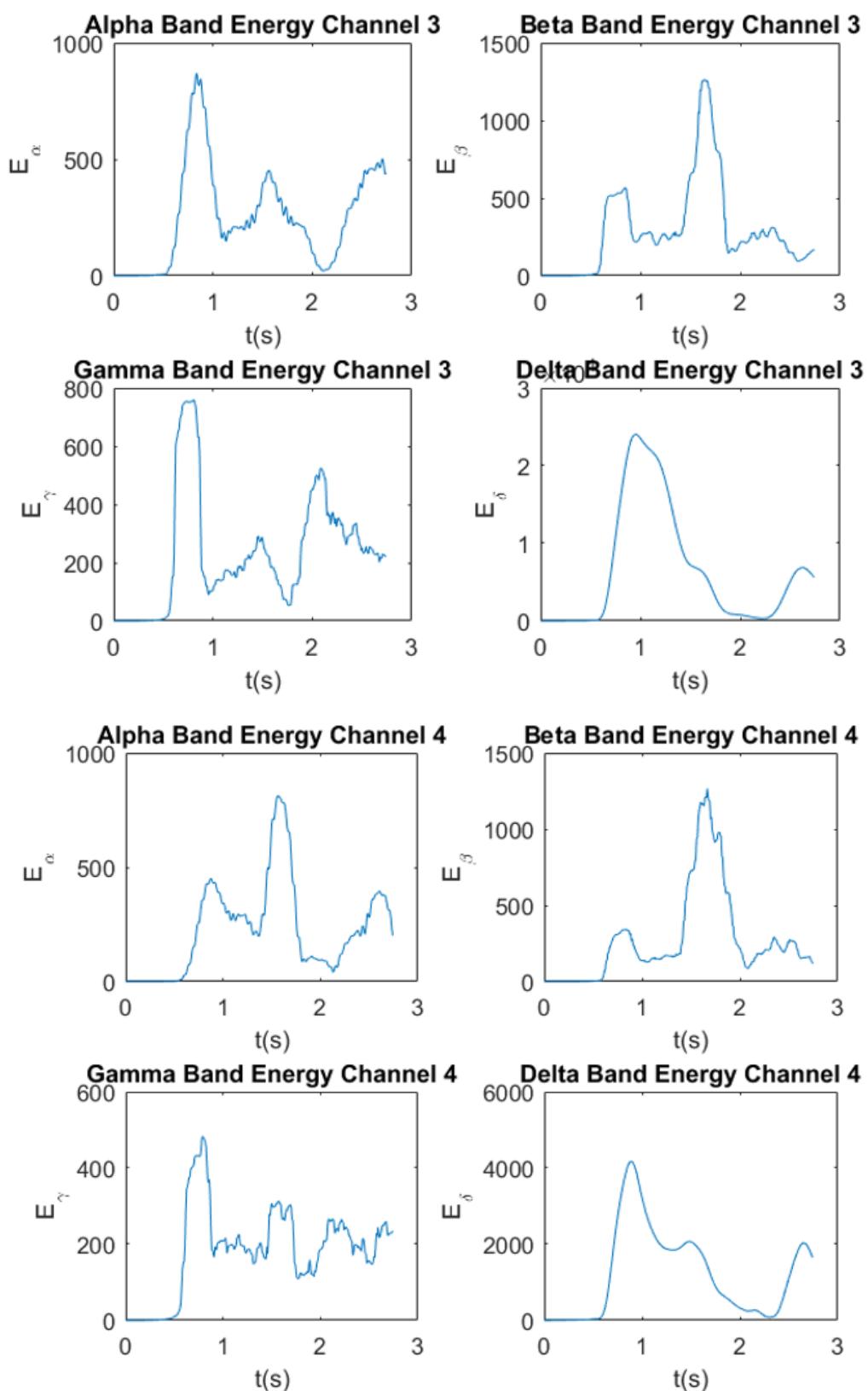




گزارش کار تمرین متلب درس سیگنال‌ها و سیستم‌ها
استنباط دوم) پنجره‌ی زمانی ۲۵۰ میلی‌ثانیه‌ای باید در طول زمان بلغزد و جلو برود و نموداری (تقریباً) پیوسته از انرژی باندهای فرکانسی پدید آید.

در ادامه، نمودارهای انرژی باندهای مختلف را با توجه به روش فوق مشاهده می‌کنید:





الگوریتم خوشبندی ما به شرح زیر است:

ابتدا از میان تمامی الکترودها، دو الکترودی را که کمترین همبستگی را دارند، انتخاب می‌کنیم و در دو دسته مختلف قرار می‌دهیم. در این مرحله، هدف ما این است که تمامی الکترودها را به دو دسته تقسیم کنیم. بنابراین، الکترود را که در نظر گرفته و همبستگی آن را با اعضای کنونی در دسته بررسی می‌کنیم و هر الکترود را در دسته‌ای قرار می‌دهیم که بیشترین همبستگی را با یکی از اعضای آن داشته باشد. (دقیق کنید که مبنای تصمیم‌گیری ما برای قرار دادن یک الکترود در یک دسته، ماکسیمم همبستگی است، نه متوسط یا مینیمم آن، یعنی همبستگی الکترود مدنظر را با تمامی اعضای هر دو دسته می‌سنجیم، سپس به یک الکترود می‌رسیم که ماکسیمم همبستگی را با الکترود مدنظر می‌دارد و آن را در این دسته قرار می‌دهیم). به این ترتیب، نهایتاً دو دسته خواهیم داشت که احتمالاً تعداد اعضای نابرابر دارند، ولی از نظر همبستگی، تا حدی به هم نزدیک شده‌اند. سپس برای هر دسته، بررسی می‌کنیم که آیا شرط خوشبندی برقرار است یا نه. یعنی بررسی می‌کنیم که آیا همبستگی دوبعدی تمامی الکترودهای یک دسته، بیشتر از حد آستانه‌ی مشخص شده است یا نه. چنان‌چه این شرط برقرار نبود، فرایند فوق را برای هر کدام از این دسته‌ها مجدداً تکرار می‌کنیم. به این ترتیب، تعداد دسته‌ها زیادتر شده و اعضای آن‌ها کمتر می‌شود. این فرایند را تا زمانی ادامه می‌دهیم که تمامی خوشبندی‌های ایجاد شده، شرط لازم را ارضاء کنند، یعنی همبستگی دوبعدی هر خوشبندی بزرگ‌تر یا مساوی آستانه‌ای باشد که تعیین کردہ‌ایم. بدیهی است بالا بردن این حد آستانه، فرایند را طولانی‌تر، تعداد خوشبندی‌ها را بیشتر، و تعداد اعضای هر خوشبندی را کمتر می‌کند.

شاید این نکته به نظرتان رسیده باشد که این الگوریتم، یک ایراد جالب توجه دارد و آن ایراد، این است که ترتیب انتخاب الکترودها برای تصمیم‌گیری، در تصمیم‌گیری برای الکترودهای بعدی مؤثر است. یعنی ممکن است به ازای یک ترتیب مشخص از انتخاب و تصمیم‌گیری برای الکترودها، نتیجه‌ای متفاوت از یک ترتیب انتخاب دیگر به دست آید. برای حل این موضوع، رو به روش تکرار تصادفی آورده‌ایم. یعنی به تعداد زیاد، فرایند خوشبندی را تکرار می‌کنیم و هر بار، ترتیب انتخاب الکترودها برای تصمیم‌گیری را تصادفی در نظر می‌گیریم. سپس، بررسی می‌کنیم که در چند درصد از دفعات تکرار، خوشبندی‌های مشابه به وجود آمده‌اند و اولویت خوشبندی نهایی را بر این مبنای تنظیم می‌کنیم که اولویت با الکترودهایی است که در دفعات بیشتری از تکرار تصادفی آزمایش، در یک خوشبندی قرار گرفته باشند.

برای پیاده‌سازی الگوریتم فوق، دوتابع نوشته‌ایم:

cluster (correlation, threshold, Index) – ۱

این تابع، اطلاعات ورودی را یک بار خوشبندی می‌کند. ورودی‌های تابع عبارتند از correlation ماتریس همبستگی است، threshold که حد آستانه‌ای است که به واسطه‌ی آن، شرط خوشبندی مشخص می‌شود و باید عددی بین صفر و یک باشد؛ و Index برداری است که اندیس الکترودهایی که قرار است خوشبندی شوند در آن قرار دارد. بدیهی است تعداد اعضای Index کوچک‌تر یا مساوی عدد n است که n، بُعد ماتریس correlation است. (یعنی ماتریس correlation یک ماتریس $n \times n$ است)

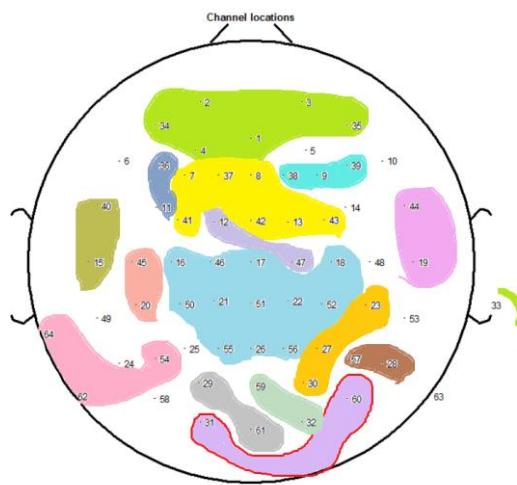
final_cluster (correlation, threshold, Index, n) – ۲

این تابع، وظیفه‌ی تکرار تصادفی عملیات خوشبندی و ارائه‌ی خوشبندی نهایی را به عهده دارد. سه ورودی اول این تابع، مشابه تابع cluster هستند و آخرین ورودی آن، n، تعداد دفعات تکرار تصادفی آزمایش خوشبندی است. بدیهی است هر چه n بیشتر شود، دقت عملکرد خوشبندی بیشتر شده و تصادفی بودن آن کمتر می‌شود.

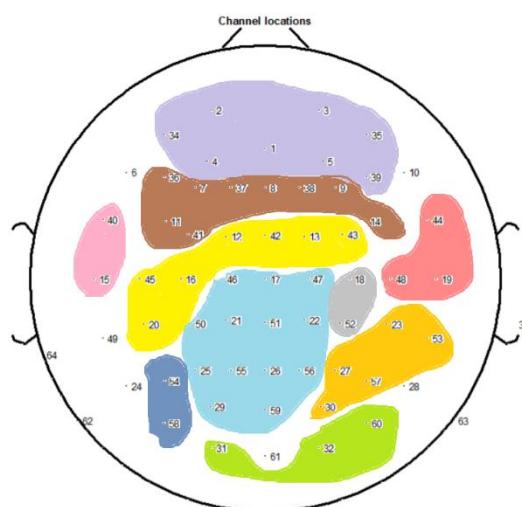
البته یک نکته‌ی مهم لازم به ذکر است و آن، این است که مسئله‌ی خوشبندی جواب یکتاً ندارد. بنابراین، هر چقدر هم که n را زیاد کنیم ممکن است باز هم به خوشبندی‌های متفاوتی بررسیم که این، یک ایراد برای کار ما تلقی نمی‌شود، چراکه همان طور که گفته شد، مسئله‌ی خوشبندی پاسخ یکتاً ندارد و هر خوشبندی‌ای که در شرط تعیین شده صدق کند، یک پاسخ صحیح است.

حال با الگوریتم فوق به خوشبندی الکتروودها می‌پردازیم. برای این کار threshold را برابر 0.8 می‌گیریم تا تعداد خوشبندی حدود 20 تا بشود. کمتر گرفتن آن باعث می‌شود خوشبندی‌های بسیار بزرگ پدید بیایند و زیاد گرفتن باعث به وجود آمدن تعداد زیادی دسته‌تکی خواهد شد که هر دو برای ما نامطلوب هستند.

ابتدا بر اساس داده‌های فعالیت اول، الکتروودها را خوشبندی می‌کنیم. برای افزایش دقت عملیات خوشبندی، در الگوریتم فوق، n را 10000 می‌گیریم تا به خوشبندی پایدار و مناسبی برسیم. الگوریتم ما الکتروودها را به صورت زیر خوشبندی می‌کند.



به نظر می‌رسد داده‌های الکتروودهای نزدیک به هم، شباهت بسیار زیادی به هم دارند. این موضوع با انتظار ما همخوانی دارد زیرا منشأ سیگنال EEG سیگنال‌های داخلی مغزی است که به دلیل عبور از جمجمه تا حد زیادی و میانگین گیری شده‌اند و هر الکتروود میانگین تضعیف شده سیگنال‌های داخلی نواحی اطراف را اندازه‌گیری می‌کند پس الکتروودهای مجاور باید کورریلیشن بالایی داشته باشند. خوشبندی را یکبار دیگر بر اساس داده‌های فعالیت دوم انجام می‌دهیم. تغییراتی در خوشبندی دیده می‌شود که طبیعی است زیرا در فعالیت‌های مختلف، ممکن است بخش‌های متفاوتی از مغز با هم به فعالیت بپردازند.



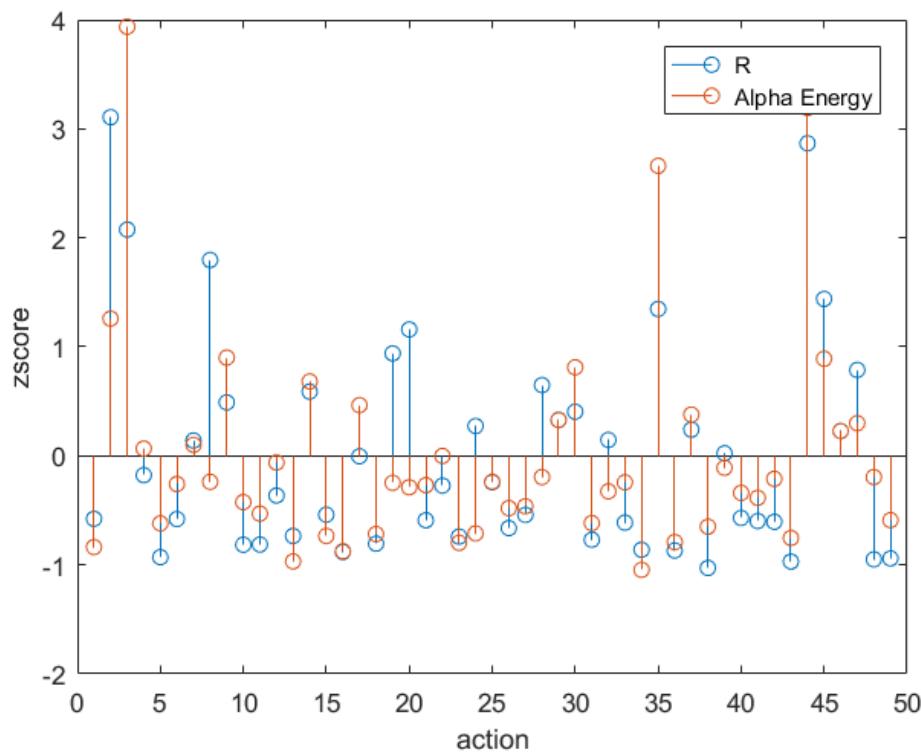
در این بخش، برای تشخیص حالات استراحت، از مقادیر انرژی باندهای فرکانسی در فعالیت‌های مختلف کمک می‌گیریم. برای این منظور، باید اطلاعاتی راجع به ارتباط هر یک از باندهای فرکانسی با عملکرد فرد داشته باشیم. به طور خلاصه، می‌توان بیان کرد که باند آلفا در حالت استراحت و عدم تمکز افزایش یافته و بر خلاف آن، باند بتا که بیشتر در هنگام تمکز و فعالیت ذهنی درگیر است، در وضعیت استراحت کاهش می‌یابد. البته باندهای فرکانسی دیگر نیز می‌توانند حاوی اطلاعاتی از این قبیل باشند، اما در مقایسه با باندهای آلفا و بتا، می‌توان از آن‌ها صرف نظر کرد، لذا ما نیز در این بخش فقط از اطلاعات همین دو باند استفاده خواهیم کرد.

برای سنجش وضعیت فرد و رسیدن به یک پاسخ نهایی، لازم است که با یک کمیت مشخص کار کنیم و نمی‌توانیم به صورت کیفی نظر بدھیم. با توجه به توضیحات فوق، مبتنی بر افزایش باند آلفا و کاهش باند بتا در هنگام استراحت، کمیت زیر را تعریف می‌کنیم که در آن، E_α و E_β انرژی باندهای آلفا و بتا هستند.

$$R = \frac{E_\alpha}{E_\beta}$$

البته می‌شد به شکل‌های دیگری نیز کمیت‌هایی مبتنی بر باند آلفا و بتا تعریف کرد، اما ما این تعریف را اولاً به دلیل بدون بُعد بودن متغیر R، و دوم به استناد از <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3812603/> انتخاب کردیم.

تا به این جا، به یک مقدار کمی (R) برای تصمیم‌گیری خود دست یافته‌ایم؛ اما هنوز تمام مسئله حل نشده است، چراکه یک معیار نیز برای تصمیم‌گیری می‌خواهیم، یعنی به دنبال یک حد آستانه در مقادیر R (یا هر روش دیگری) هستیم که بتوانیم با داشتن مقدار R، در مورد حضور یا عدم حضور فرد در حالت استراحت تصمیم بگیریم. ابتدا ادعا می‌کنیم که نمی‌توان بدون هیچ گونه اطلاعات خارجی، آستانه‌ای برای R انتخاب کرد، چراکه با توجه به عملکرد مغز، این حد آستانه ممکن است هر مقداری را به خود بگیرد. برای مقابله با این مشکل، داده‌ها را نه به تنها یی، که در کنار هم می‌سنجیم. بنابراین، R را متغیری تصادفی در نظر می‌گیریم که در این آزمایش، ۴۹ نمونه از آن را در اختیار داریم. حال، به دنبال مقادیری از R هستیم که به شکل قابل توجهی بزرگ باشند، برای این کار، نمودار z-score مربوط به R، یعنی نمودار متغیر $\frac{R - \mu_R}{\sigma_R}$ را رسم می‌کنیم و با توجه به مقادیر آن در فعالیت‌های مختلف، در مورد وضعیت هر کدام از این فعالیت‌ها تصمیم می‌گیریم.



در شکل مقابل، نمودار z-score مربوط به R را در کنار نمودار z-score مربوط به آلفا از نظر می‌گیریم. علت رسم نمودار دوم، آن است که همواره در مراجع علمی، باند آلفا را به عنوان مؤلفه‌ی اصلی حالت استراحت معرفی می‌کنند و در نظر گرفتن آن برای انجام تصمیم‌گیری نهایی خالی از لطف نیست. (اما دقت داریم که ملاک اصلی ما، کمیت R است)

حال، می‌خواهیم تصمیم‌گیری نهایی را انجام دهیم. با توجه به نمودار فوق، تعداد ۷ فعالیت از ۴۹ فعالیت، مقادیر R خارج از بازه $\mu_R + \sigma_R \leq R \leq \mu_R - \sigma_R$ که از میان این هفت فعالیت، دو فعالیت وجود دارند که مقدار انرژی باند آلفای آن‌ها دارای منفی است، یعنی کمتر از متوسط انرژی باند آلفای همه‌ی فعالیت‌ها. با توجه به این که انرژی باند آلفا، مهم‌ترین فاکتور (در میان باندهای فرکانسی) برای حضور در حالت استراحت است، این دو فعالیت را از لیست فعالیت‌های انتخابی برای حالت استراحت کنار می‌گذاریم و ۵ فعالیت باقیمانده را به عنوان پاسخ می‌پذیریم.

به این ترتیب، فعالیت‌های ۲، ۳، ۳۵، ۴۴، و ۴۵، مربوط به حالت استراحت هستند.

البته این نکته نیز شایان ذکر است که می‌توانستیم بازه‌ی سخت‌گیرانه‌تری را برای این تصمیم‌گیری در نظر بگیریم. با توجه به نمودار رسم شده، اگر این بازه را به شکل $k\sigma_R \leq R \leq \mu_R + k\sigma_R$ در نظر بگیریم، (دقت داریم که فعالیت‌های منتخب، فعالیت‌هایی هستند که مقدار R آن‌ها خارج از بازه‌ی مذکور قرار بگیرد، نه داخل آن) به ازای مقدار $k = 1.5$ ، دو فعالیت ۳۵ و ۴۵ از لیست فوق خارج می‌شوند و اضافه کردن تدریجی مقدار k ، می‌تواند تعداد جواب‌های ما را کمتر کند. البته مشاهده می‌شود که بعد از $k = 1.5$ ، مقادیر R به شکل معنی‌داری بزرگ‌تر هستند و بزرگ‌کردن بیش از اندازه k برای رد کردن فعالیت‌های باقیمانده معقول نیست.

همچنین می‌توانیم مقدار k را کوچک و کوچک‌تر کنیم تا تعداد فعالیت‌های پذیرفته شده به عنوان حالت استراحت زیاد شود، که هر چقدر k به صفر نزدیک‌تر شود، ارزش انتخاب‌های ما را پایین‌تر می‌آورد.

در حالت کلی اگر ما چند داده حالت استراحت برای این فرد خاص در اختیار داشتیم، می‌توانستیم از روی آن‌ها یک بازه‌ی اطمینان مناسب پیدا کنیم و با آن بررسی کنیم که آیا دیگر فعالیت‌ها که اطلاعاتی از آنها نداریم استراحت هستند یا خیر و این روش بسیار دقیق‌تر از روش استفاده شده در این تمرین جواب می‌داد اما به دلیل اینکه این داده‌ها را در اختیار نداریم به همین روش بسته کردیم.