



دانشگاه صنعتی شریف

دانشکده مهندسی برق

پردازش سیگنال‌های EEG

تمرین کامپیوترا سری سوم

دانشجو

سید ابوالفضل مرتضوی

۴۰۲۲۰۰۱۹۱

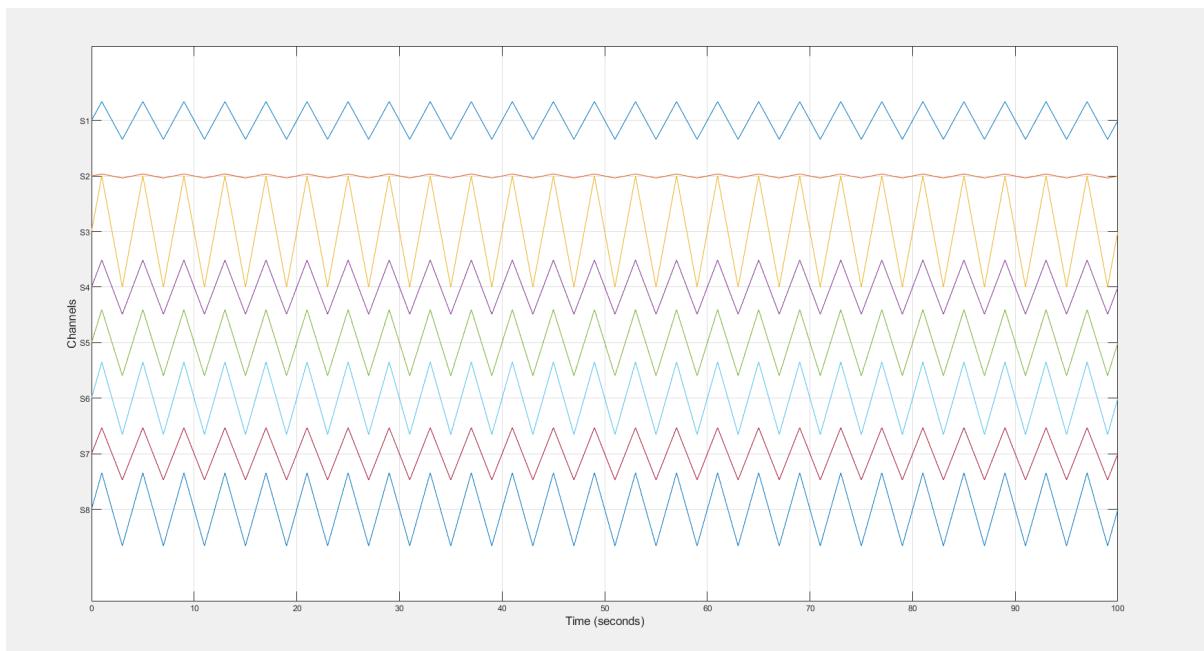
آذر ۱۴۰۲

تمرین اول

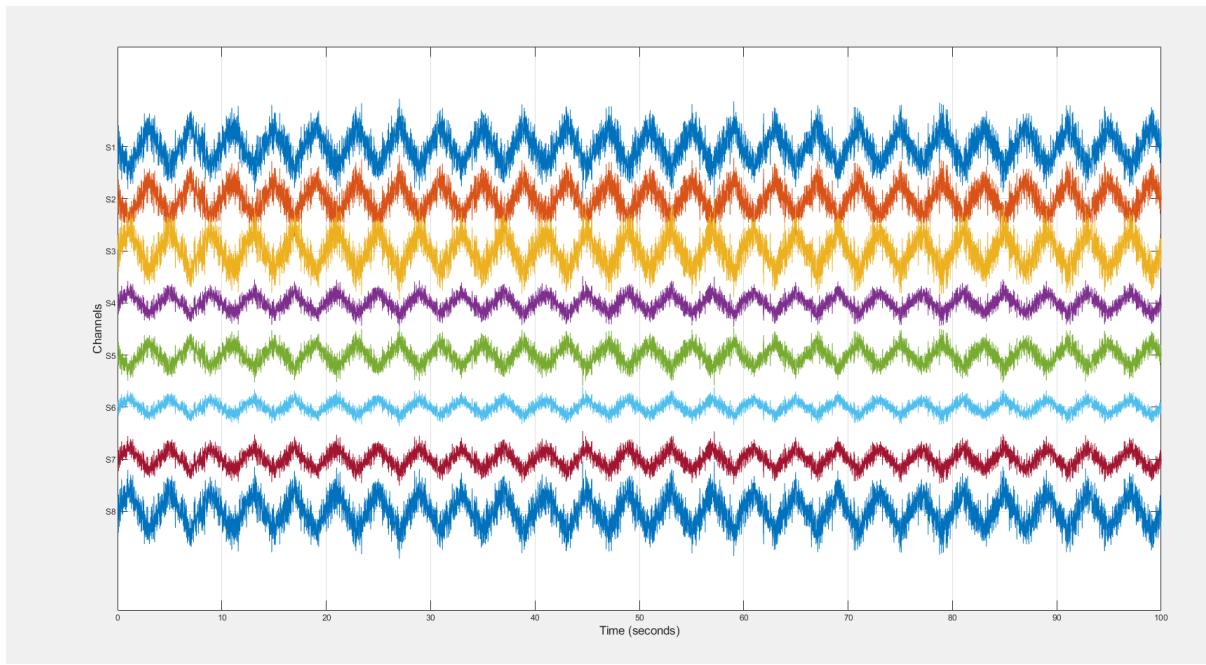
GEVD - ۱-۱

۱-۱-۱- الف

بعد از وارد کردن دادگان به مطلب و شیفت دادن آنها و در نهایت پیاده‌سازی الگوریتم GEVD خروجی زیر حاصل شد.



شکل ۱-۱ اثر اصلی منبع در مشاهدات

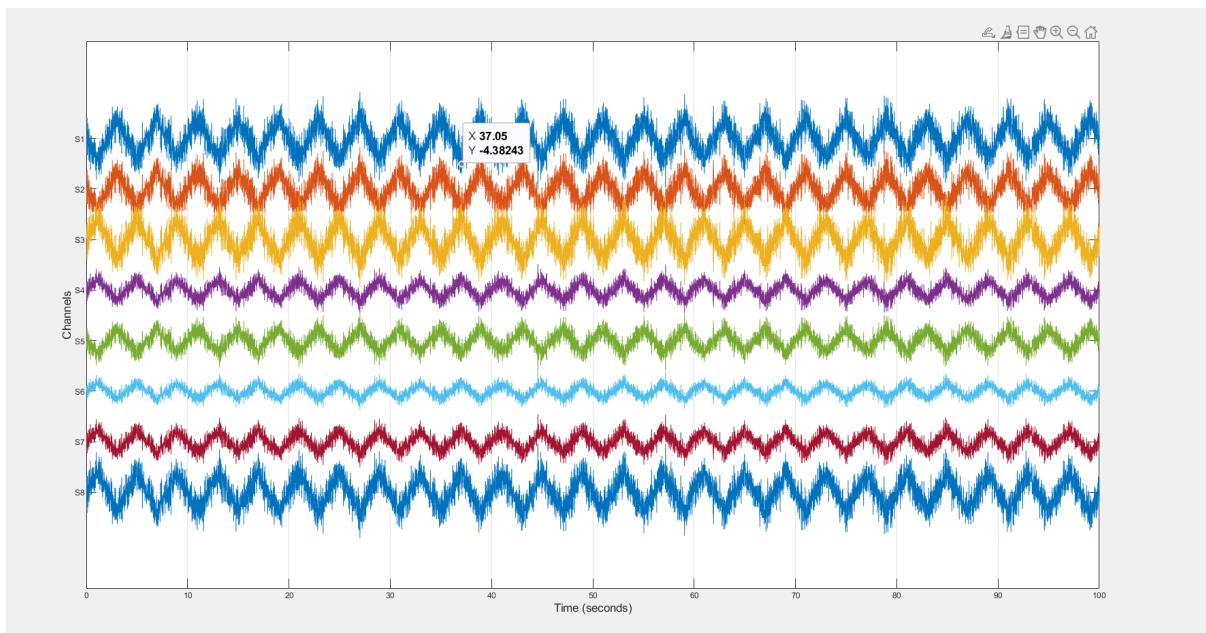


شکل ۱-۲ اثر به دست آمده با الگوریتم GEVD

خروجی به دست آمده، شبیه به داده‌ی اصلی است اما برروی این دادگان، مقداری نویز وجود دارد. خطای تخمین برای این منبع برابر با 547.899418679921 است. خطای تخمین به صورت مجموع توان دو های اختلاف داده‌ها در نظر گرفته شده است.

۱-۲-۱-ب

برای این بخش ابتدا باید مقدار T را به دست می‌آورديم. برای اين منظور، با استفاده از يك حلقه‌ی while مقادير مختلف T ها امتحان شدند، و در T های مختلف الگوریتم GEVD پياده‌سازی شد، تا اينکه T اى که كمترین مقدار خطای تخمین را داشت، به عنوان T اصلی در نظر گرفته شد. در نهايit خروجی اين الگوریتم به صورت زير است.

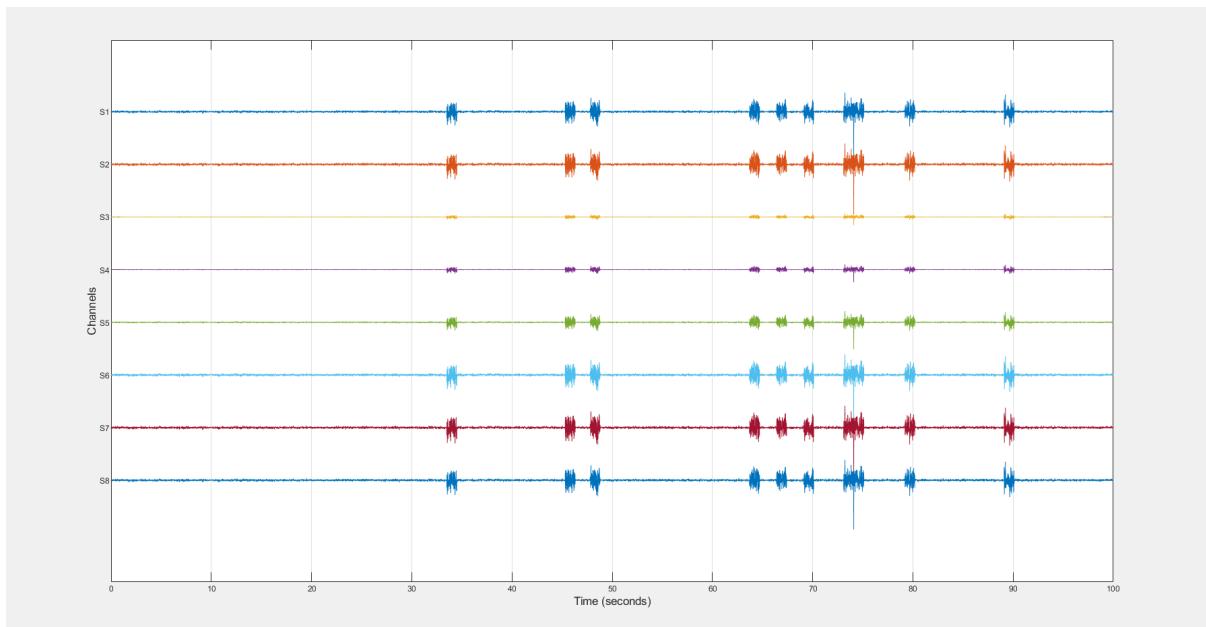


شکل ۱-۳ اثر به دست آمده

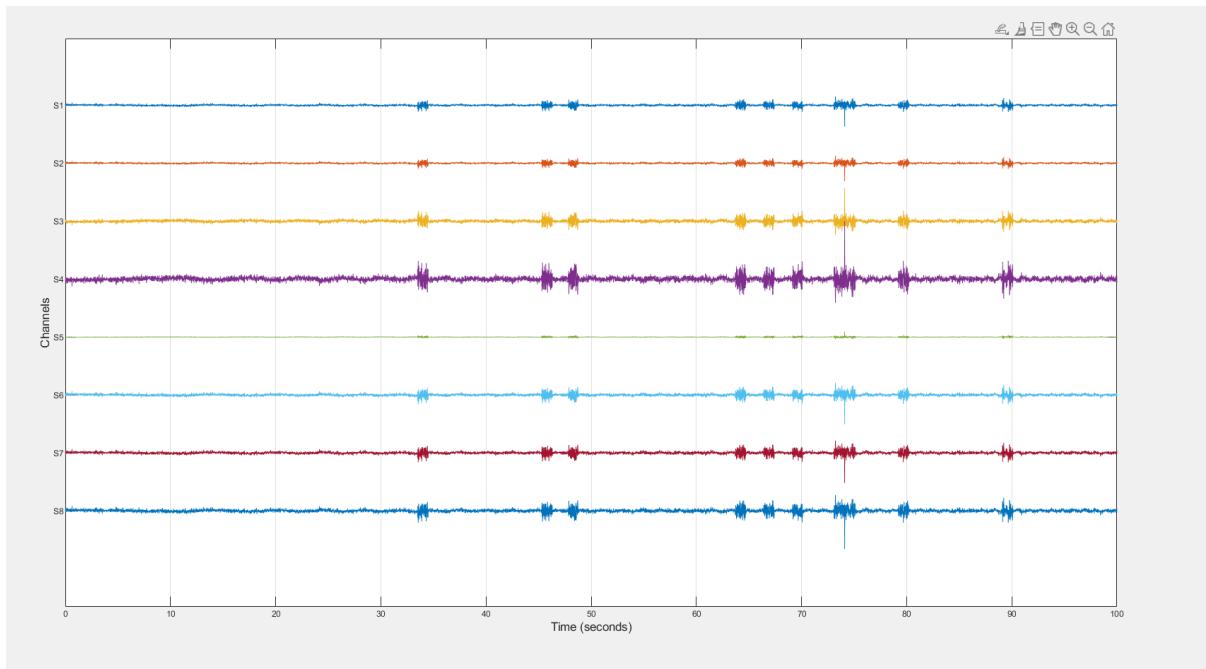
در نهایت خطای تخمین مینیمم برابر با 547.899418679921 شد که برابر با حالت قبل است. مقدار T نیز مانند حالت قبل، 400 به دست آمد.

۱-۳-۱-ج

در این بخش با در نظر گرفتن بردار T_{On} برای تخمین زمان خاموش و روشن و بودن منبع، الگوریتم GEVD بر روی دادگان پیاده‌سازی شد و خروجی به صورت زیر حاصل شد.



شکل ۱-۴ اثر اصلی منبع در مشاهدات

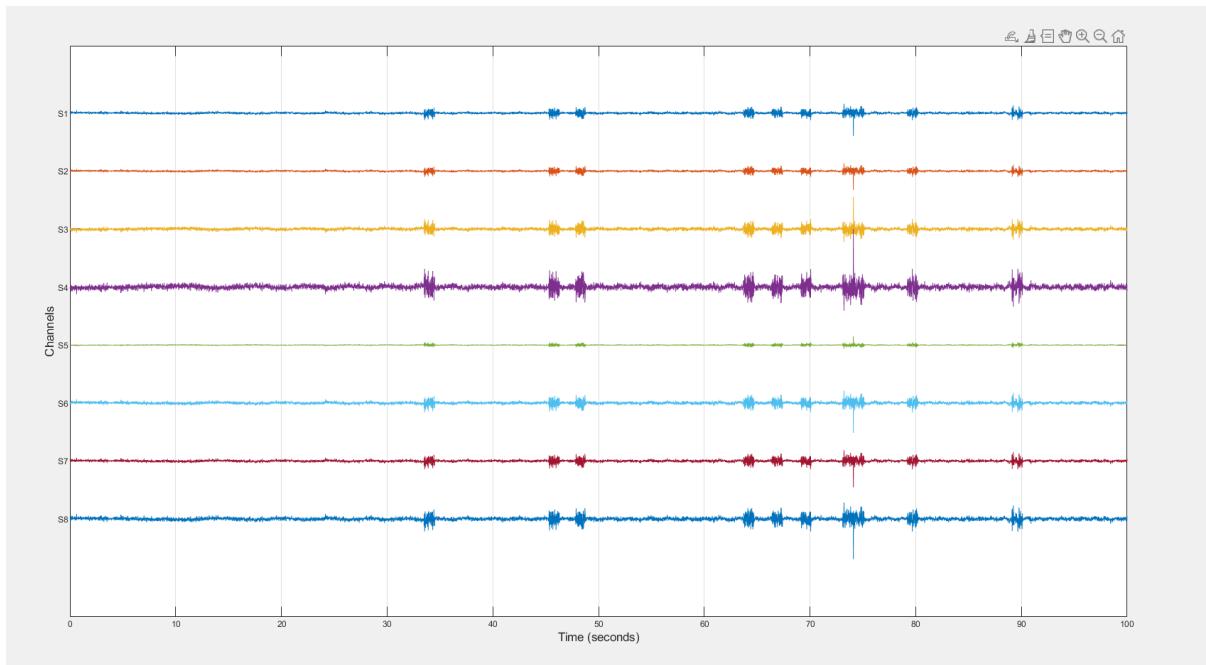


شکل ۱-۵ اثر تخمین زده شده با الگوریتم GEVD

مقدار خطا در این روش برابر با 496.502352833551 است. همانطور که قابل مشاهده است، اثر منبع با استفاده از روش GEVD با تقریب خوبی برابر با اثر اصلی است.

۱-۴-۱-۵

همانند بخش قبل، ولی این بار با بردار T_{On2} این الگوریتم پیاده‌سازی شد و خروجی به صورت زیر به دست آمد.

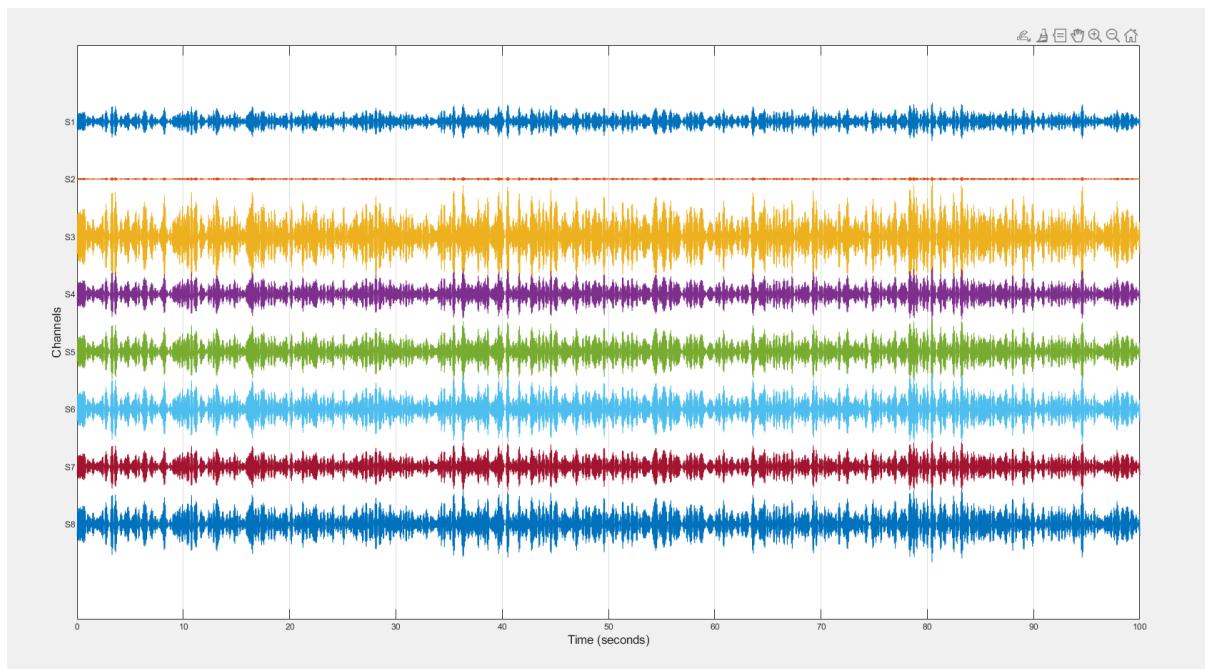


شکل ۶-۱ اثر منبع در مشاهدات با الگوریتم GEVD

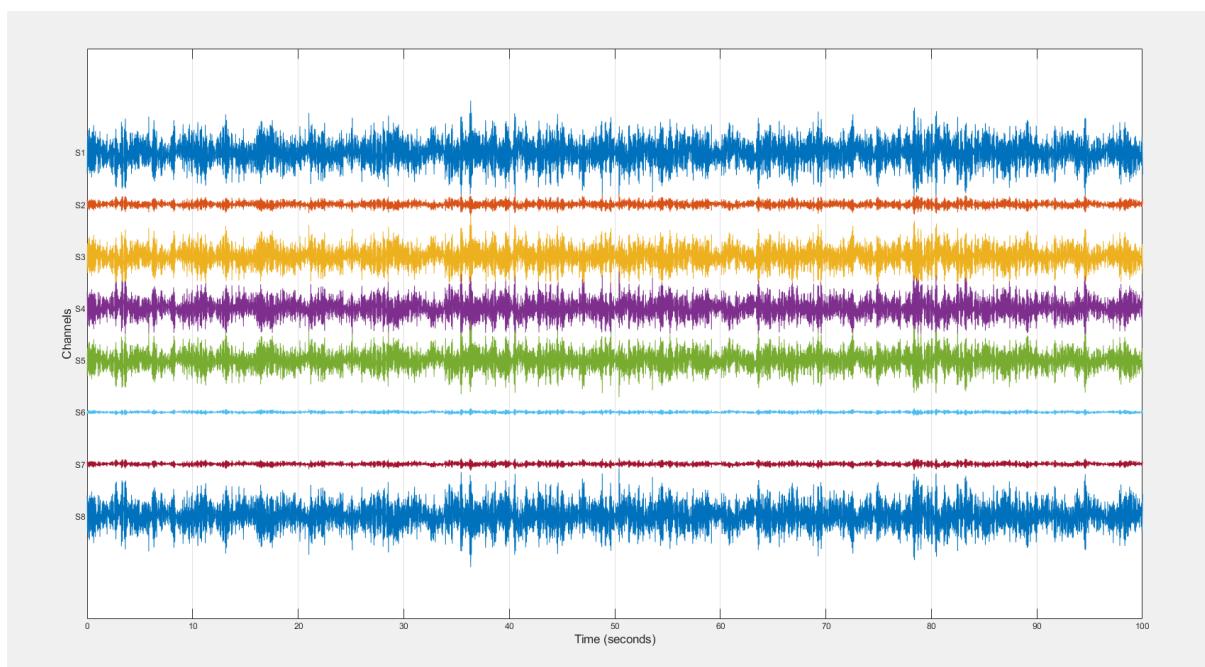
مقدار خطأ برابر با $503.650 \cdot 562244393$ شد. در این روش دامنهای سیگنال، نسبت به بخش قبلی کاهش داشته است و همچنان مقدار خطأ افزایش یافته است.

۵-۱-۱

در این بخش، ابتدا برروی دادگان اصلی، فیلتری میانگذر با درجه 100_1 ، با فرکانس قطع 10 تا 15 هرتز، استفاده شد. درجه فیلتر با تغییر و پیداکردن کمترین خطأ به دست آمد. بعد از انجام این مرحله، الگوریتم GEVD برروی دادگان پیاده شد و خروجی به دست آمد.



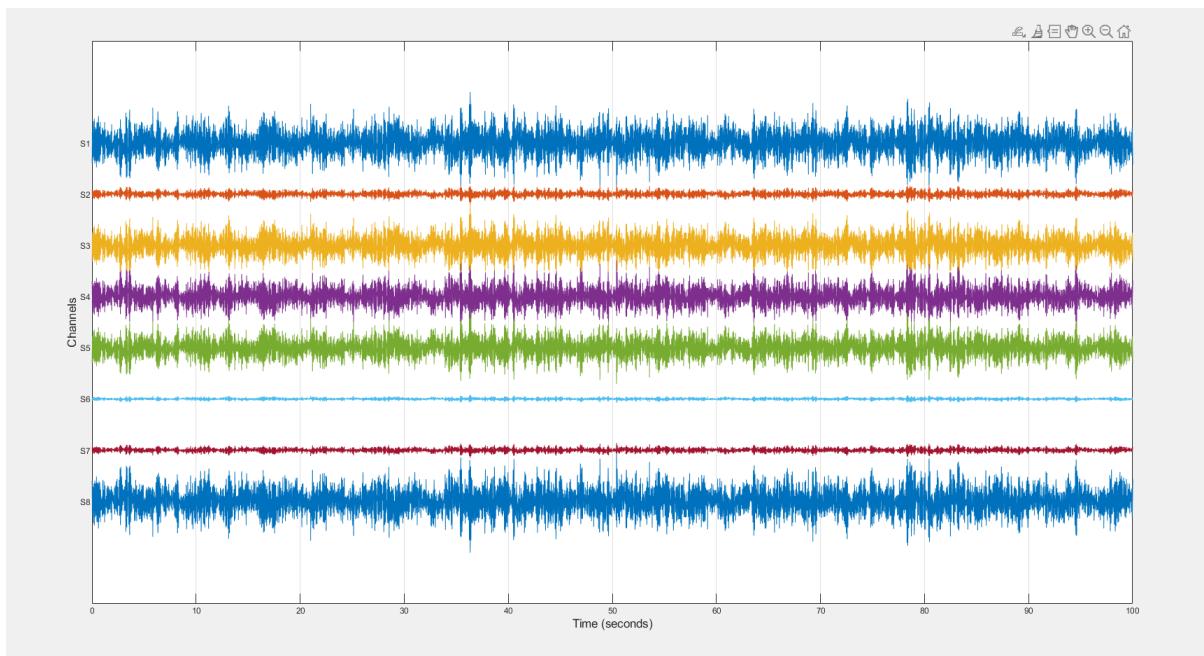
شکل ۱-۷ اثر اصلی منبع در مشاهدات



شکل ۱-۸ اثر به دست آمده با روش *GEVD*

مقدار خطا در این روش برابر با 377.322865307553 شد که نسبت به بخش‌های قبلی مقداری کمتر است. اما شکل خروجی شباهت کمتری به شکل اصلی دارد، این مورد می‌تواند به علت مناسب نبودن فیلتر استفاده شده باشد.

در این بخش نیز مانند بخش ۱-۱-۲ از یک حلقه‌ی while برای بهینه‌سازی استفاده شد. در این حلقه، پهنانی باندهای مختلف برای بازه‌ی فرکانسی امتحان شد. در داخل این حلقه، از یک حلقه‌ی for برای تست کردن، پهنانی باند، بازه‌های مختلف استفاده شد. برای مثال فرض کنید که پهنانی باند در ابتدا ۵ باشد، با استفاده از حلقه‌ی for این پهنا در بازه‌های ۵ تا ۱۰، ۶ تا ۱۱ و... امتحان می‌شود. سپس پهنانی باند به ۶ افزایش یافته و همین روند تکرار می‌شود. هرجا که مقدار خطا کمترین مقدار باشد، به عنوان بهترین تخمین بازه‌ی فرکانسی سیو می‌شود. با انجام این روند و پیاده‌سازی الگوریتم GEVD شکل‌های زیر حاصل شدند.



شکل ۱-۹ اثر به دست آمده برای منبع

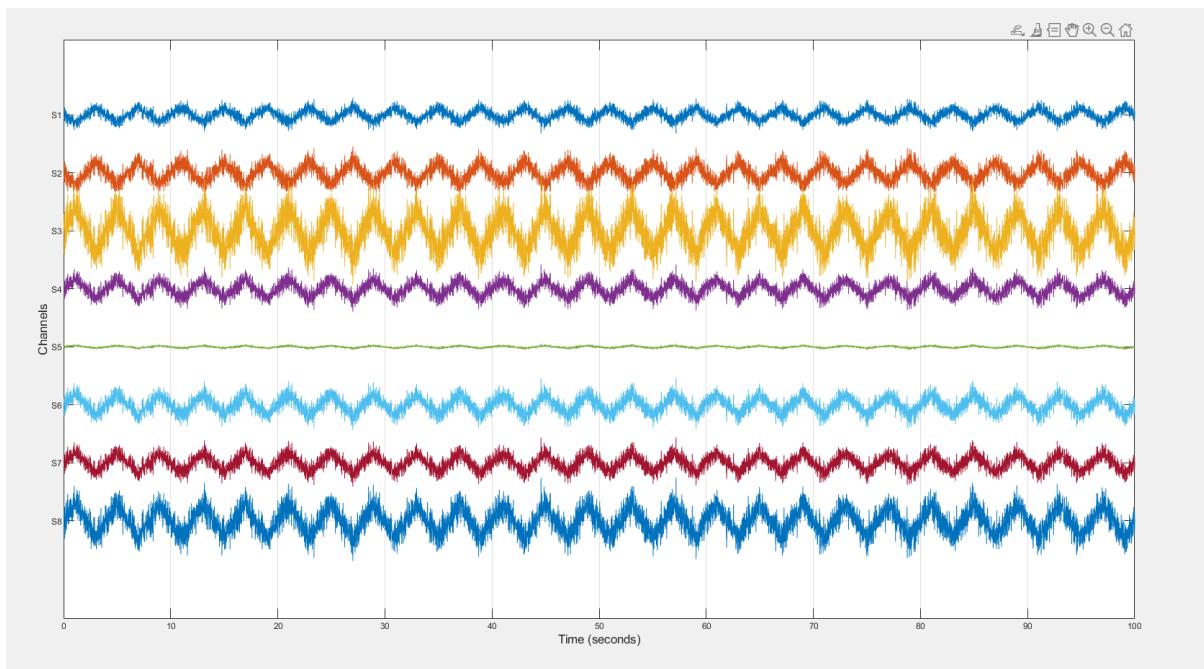
در این روش مقدار خطا نسبت به قبل کاهش یافت و این مقدار برابر با 371.696125187094 شد. بازه‌ی فرکانسی بهینه نیز برابر با 9 تا 14 هرتز به دست آمد.

در این دو بخش، خروجی الگوریتم، شباهت کمتری به دادگان اصلی دارد که این مورد به علت است که دادگان اصلی، به خودی خود شکل خاصی ندارند، و وقتی مقداری نویز نیز در خروجی الگوریتم بر روی دادگان سوار می‌شود، شکل خروجی شباهت کمی به دادگان اصلی خواهد داشت.

DSS - ۲-۱

۱-۲-۱- الف

در این بخش الگوریتم DSS برروی دادگان پیاده شد. مقدار محدودیت iteration برابر با ۱۰۰۰۰ در نظر گرفته شد تا از بینهایت بار تکرار شدن الگوریتم جلوگیری شود. خروجی این الگوریتم به صورت زیر به دست آمد.



شکل ۱-۱۰ اثر منبع در مشاهدات

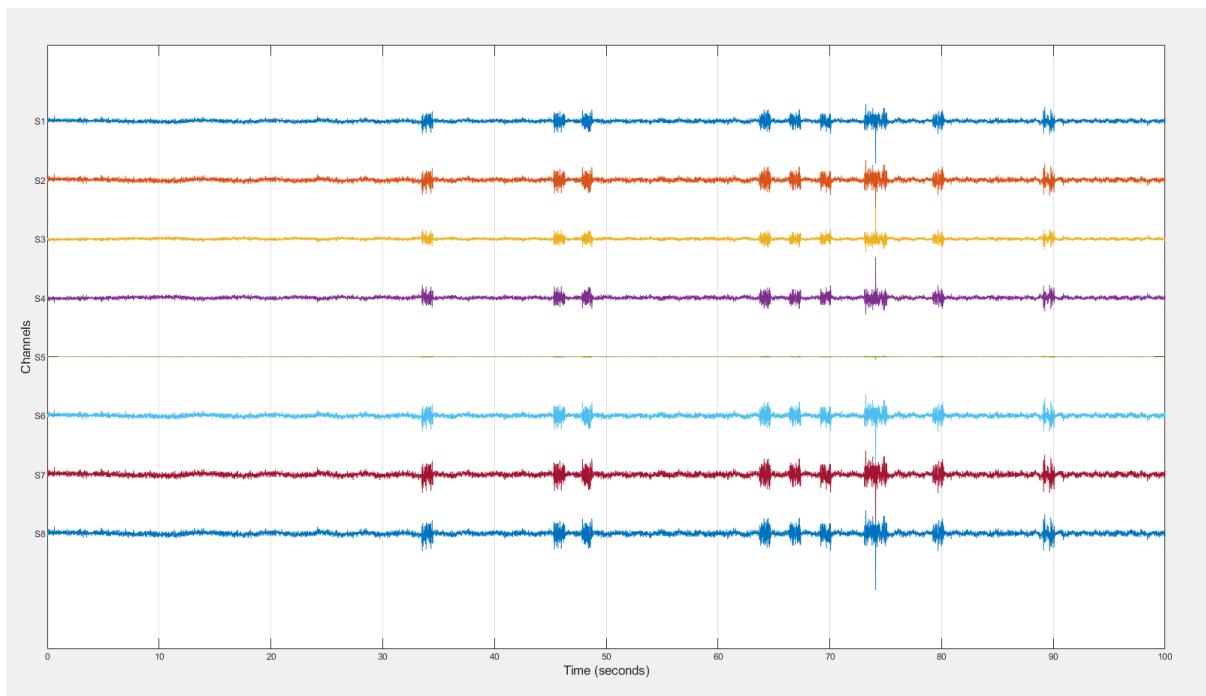
در این روش خطابرابر با 83.2797138557360 به دست آمد که نسبت به روش GEVD بسیار کمتر است، همچنین شکل خروجی دادگان، مقدار نویز کمتری نسبت به روش GEVD دارد.

۱-۲-۲- ب

در این روش نیز برای تخمین T الگوریتم گفته شده در بخش ۱-۱-۲- پیاده شد. خروجی این بخش مشابه بخش قبل است و T نیز برابر با 400 به دست آمد. خطابرابر با 400 مانند بخش قبل است.

۱-۳-۲- ج

در این بخش الگوریتم DSS برروی دادگان پیاده شد و خروجی به صورت زیر به دست آمد.

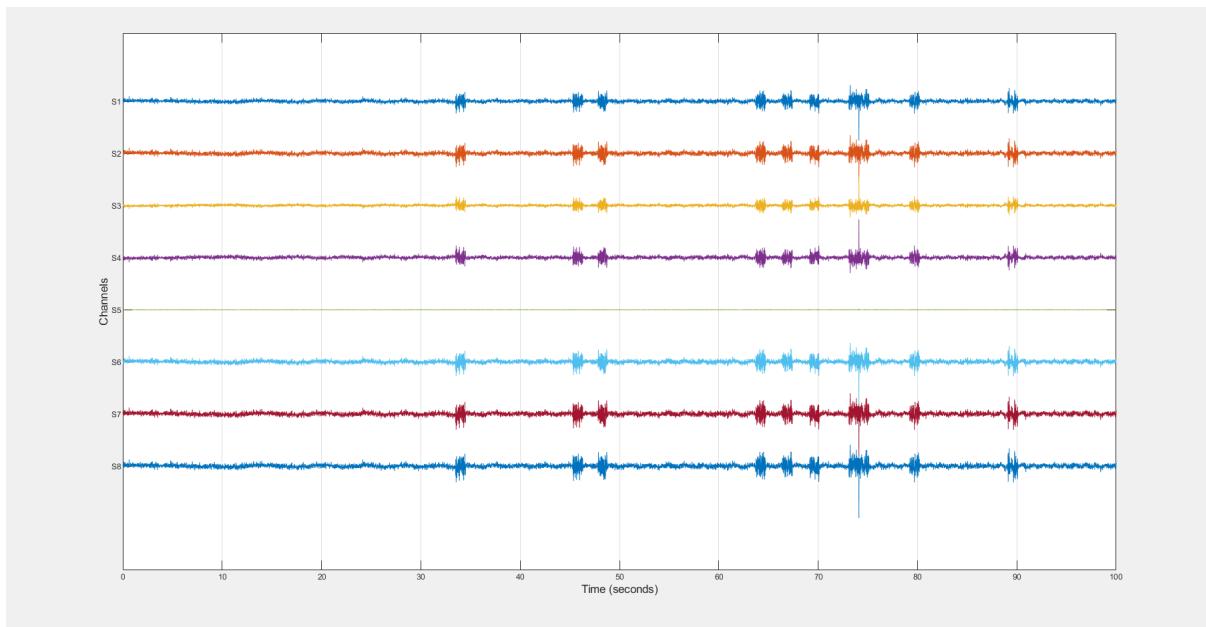


شکل ۱-۱۱ اثر به دست آمده با روش DSS

خروجی این روش نسبت به روش GEVD خطای کمتری دارد و این خطا برابر با 81.2168750348182 است. همچنین شکل به دست آمده نیز شباهت بیشتری به شکل اصلی دارد.

۴-۲-۱ د

خروجی به دست آمده برای این بخش به صورت زیر است.

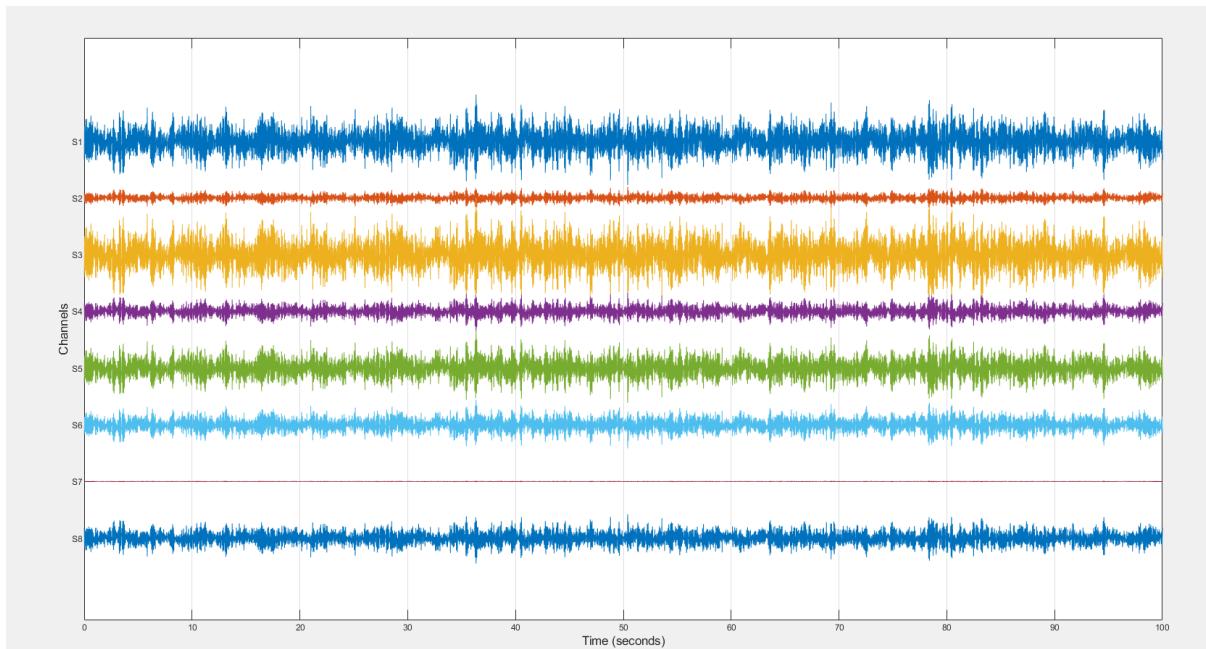


شکل ۱-۱۲ اثر منبع در مشاهدات

مقدار خطا در این بخش نسبت به بخش قبلی افزایش داشته و برابر با 81.5563317959227 است.

۵-۲-۱

در این روش نیز کارهای انجام شده در بخش ۱-۱-۵- تکرار شد و سپس الگوریتم DSS بر روی دادگان فیلتر شده پیاده شد.

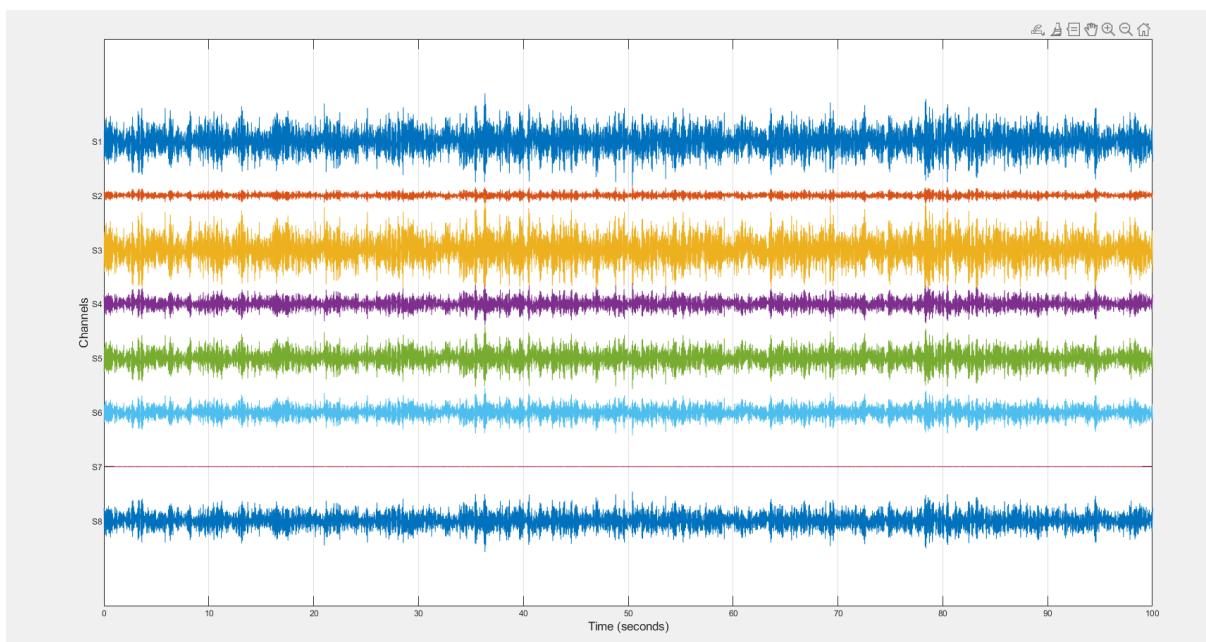


شکل ۱-۱۳ اثر منبع در مشاهدات با روش DSS

در این روش نیز خروجی شباهت زیادی به خروجی اصلی ندارد اما خطای این روش نسبت به GEVD کمتر است و برابر با 85.9032994452535 است.

۱-۶-۲-۱ و

الگوریتم گفته شده در ۱-۱-۶- برای پیدا کردن بهینه‌ترین بازه پیاده‌سازی شد و خروجی به صورت زیر به دست آمد.



شکل ۱-۱۴ اثر تخمین زده شده منبع در مشاهدات

مقدار خطای برابر با 85.7561132386168 است. همچنین رنج فرکانسی بهینه در این الگوریتم برابر با 9 تا 13 هرتز است.

تمرین دوم

برای افزودن نویز به دادگان، ابتدا باید ضریبی که نیاز بود تا نویز را با آن به سیگنال اضافه کنیم را به دست می‌آوریم.

$$x = S + \sigma N$$

برای این کار از رابطه‌ی موجود در اسلایدها استفاده شد. این رابطه در شکل زیر نشان داده شده است.

○ اضافه کردن یک نویز با SNR مشخص به یک سیگنال:

$$P_s = \|s\|_2^2$$

$$P_N = \|N\|_2^2$$

$$\text{داده شده } SNR_{dB} \rightarrow x[m] = s[m] + \sigma N[m] = s[m] + N'[m]$$

$$SNR_{dB} = 10 \log_{10} \frac{\|s\|^2}{\|\sigma N\|^2} = 10 \log_{10} \frac{P_s}{\sigma^2 P_N} \rightarrow \frac{P_s}{\sigma^2 P_N} = 10^{\left(\frac{SNR_{dB}}{10}\right)}$$

$$\rightarrow \sigma^2 = \frac{P_s}{P_N} \times 10^{-\left(\frac{SNR_{dB}}{10}\right)}$$

بعد از محاسبه‌ی این مورد نویز با سیگنال‌ها افزوده شد. در اینجا از نویز 3 و 4 برای افزودن به سیگنال استفاده شد. پس از افزودن نویزها شکل‌های زیر حاصل شدند.

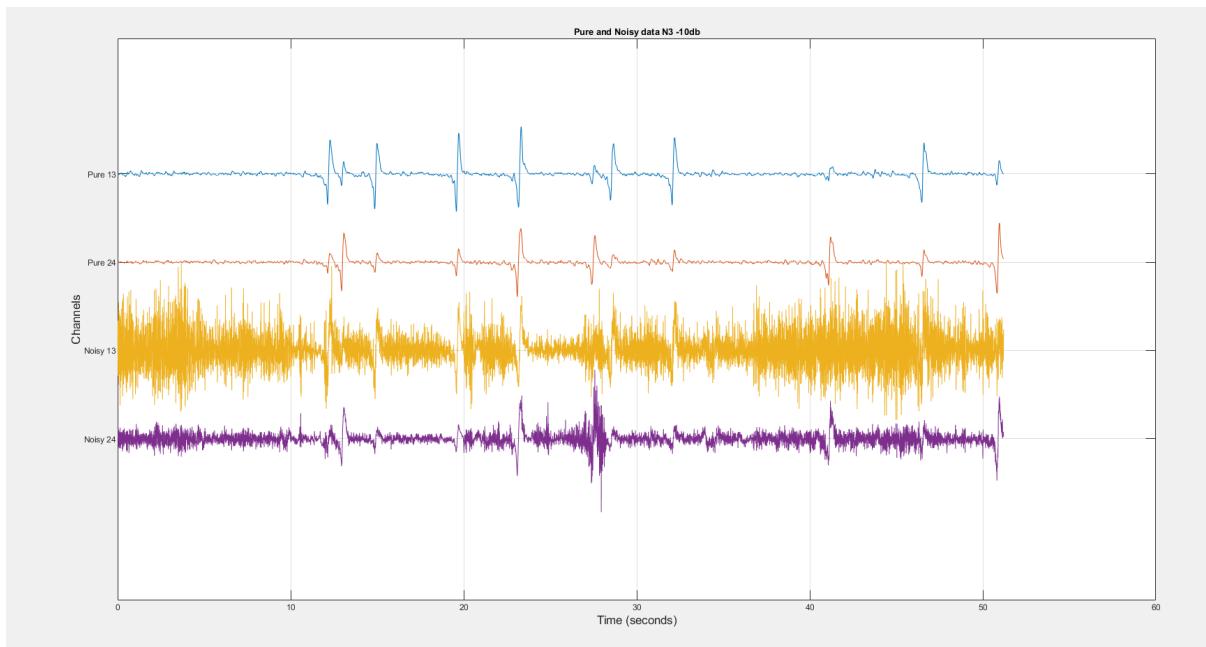
GEVD - ۱-۲

۱-۱-۲- الف

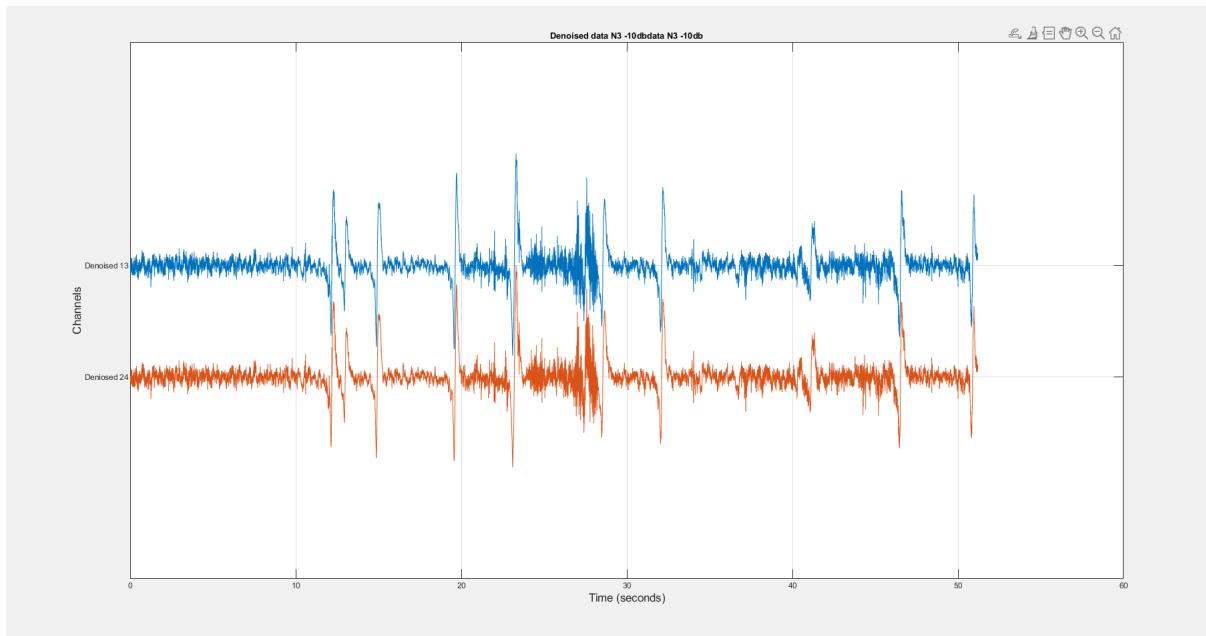
برای تخمین بردار on-off ابتدا داده‌ها رسم شدند تا مقدار آستانه برای پیدا کردن اسپایک‌ها به دست بیاید. این مقدار برابر با 0.59 به دست آمد. سپس برای هر سطر داده‌ها، هرجا که مقدار بیشتر از آستانه است، در ماتریس T ، در سطر و ستون مربوطه 1 و در غیر اینصورت 0 در نظر گرفته شد. سپس محل‌هایی که در آن‌ها بیشتر از نصف کانال‌ها، دارای مقدار یک هستن در بردار T_{rebuilt} ، با 1 در نظر گرفته شد و بقیه‌ی جاها صفر در نظر گرفته شدند. با این کار برداری با یک سطر داریم که عناصر آن نشان‌دهنده‌ی محل وقوع اسپایک‌ها هستند.

۱-۲-۲- ب و ج

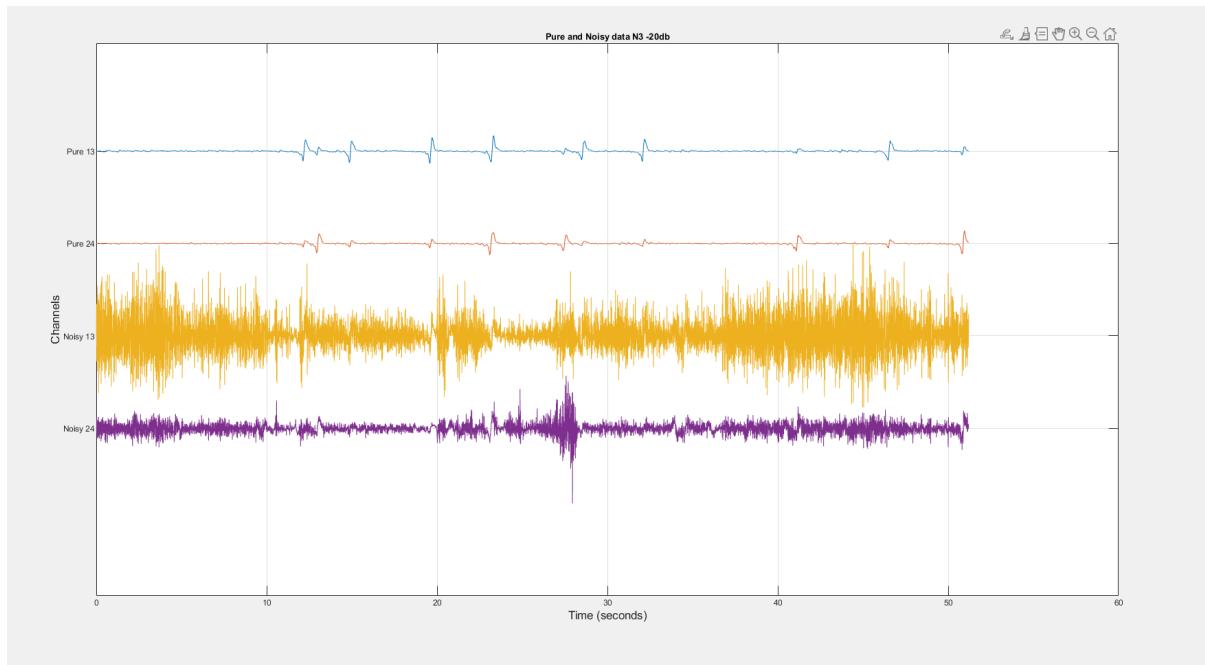
بعد از به دست آوردن بردار T به پیاده‌سازی الگوریتم GEVD بروی دادگان پرداخته شد. خروجی این روش برای نویزها و SNR های مختلف به صورت زیر هستند.



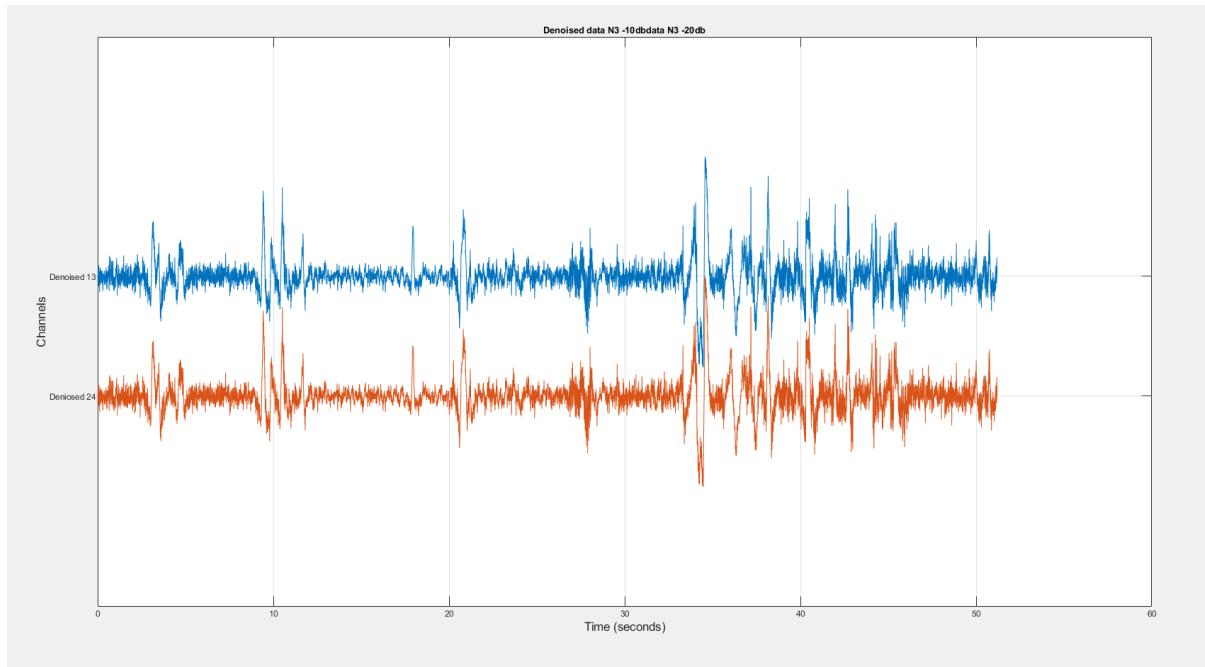
شکل ۱-۲ داده‌ی اصلی و نویز برای نویز ۳ با $SNR=-10db$



شکل ۲-۲ خروجی الگوریتم GEVD برای نویز ۳ با $SNR=-10db$

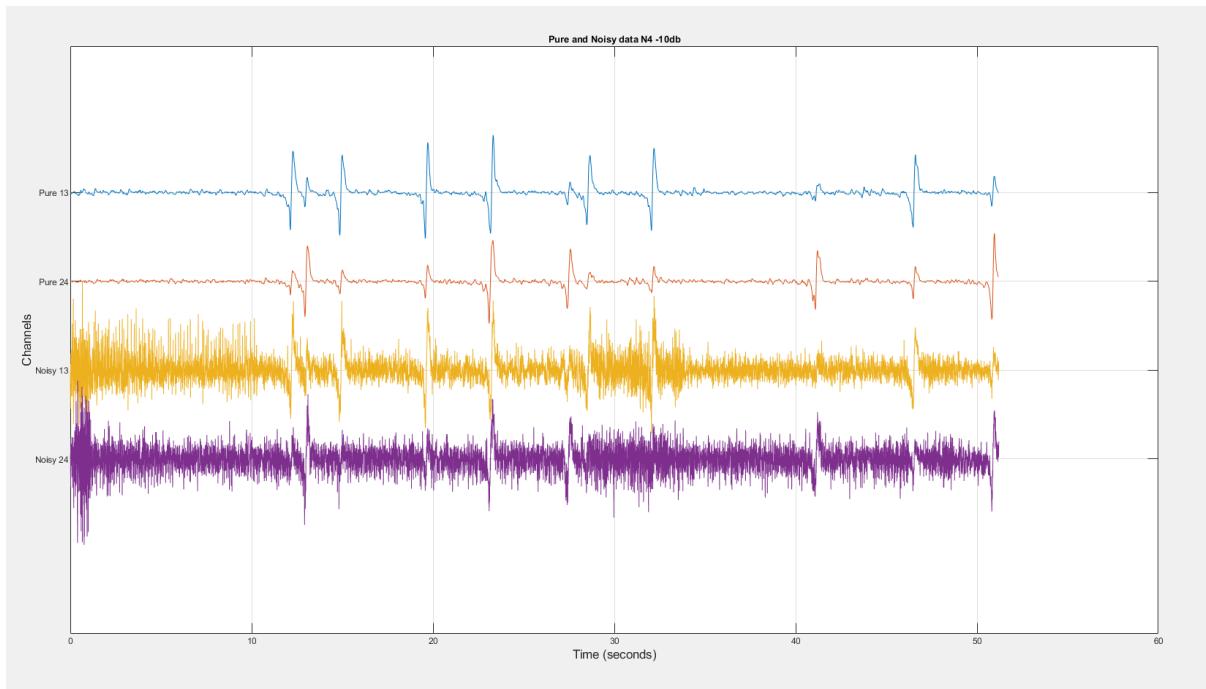


شکل ۳-داده‌ی اصلی و نویز برای نویز ۳ با $SNR=-20db$

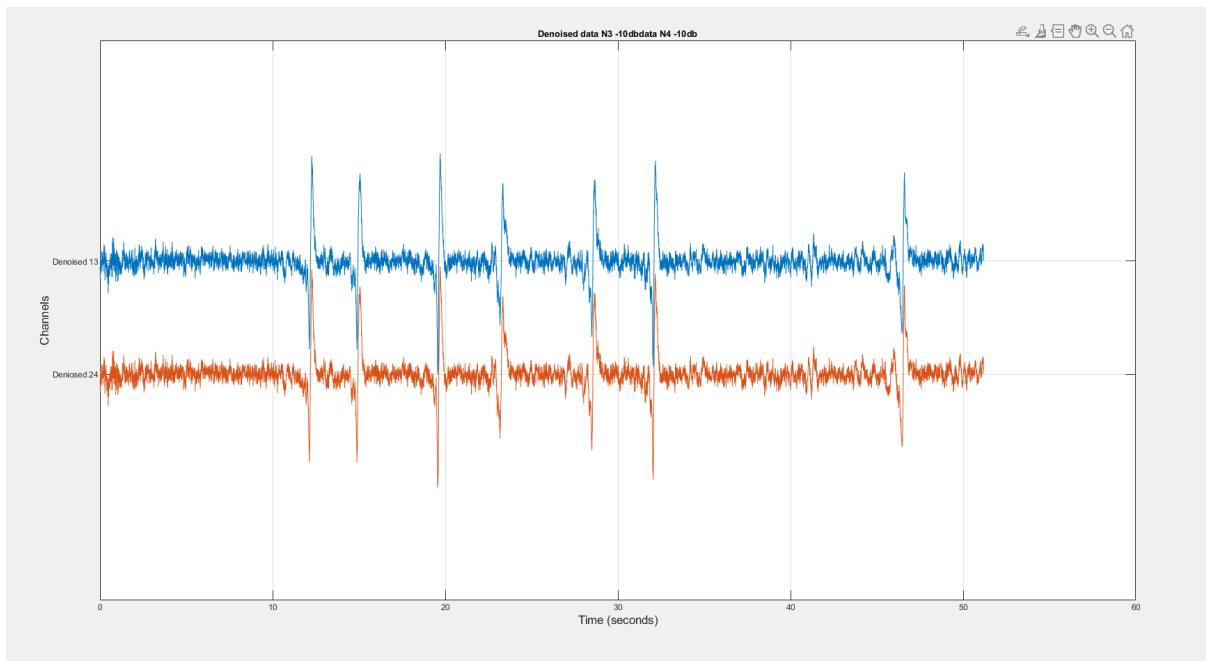


شکل ۴-خروجی الگوریتم GEVD برای نویز ۳ با $SNR=-10db$

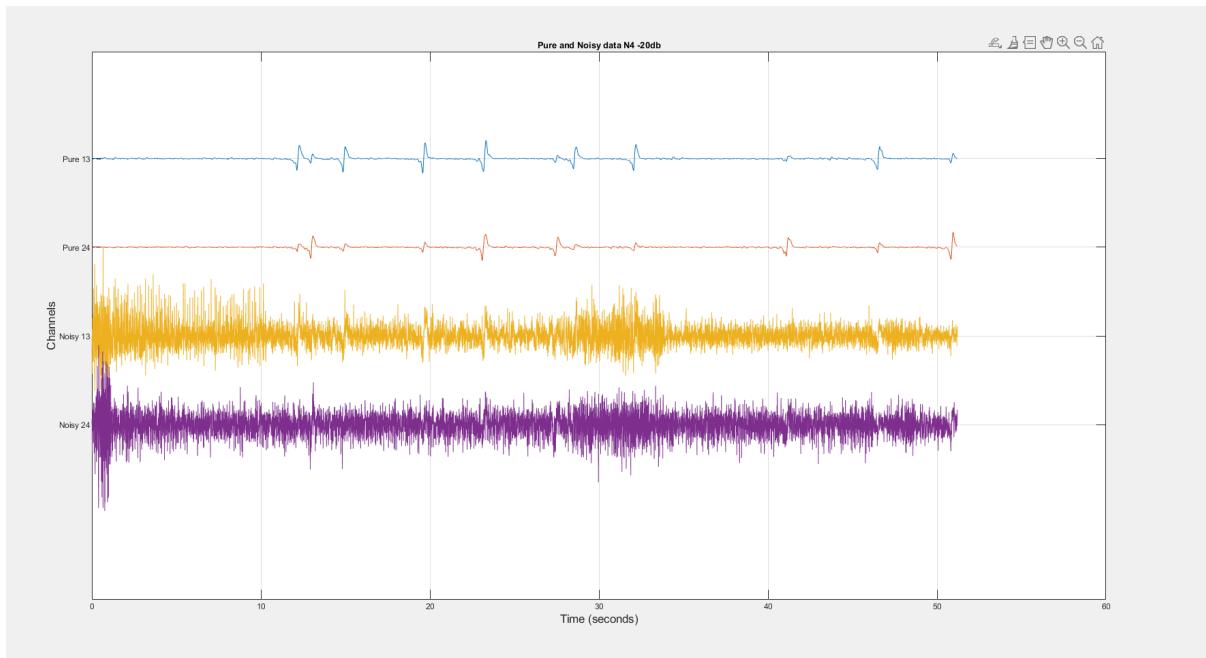
در نویز ۳، خروجی برای $SNR=-10db$ شباهت زیادی به داده‌ی اصلی دارد، اما در $SNR=-20db$ با توجه به توان بیشتر نویز نسبت به حالت قبل، خروجی شباهت کمتری به داده‌های اصلی دارد. اما در هر دو مورد می‌توان اسپایک‌ها را تشخیص داد.



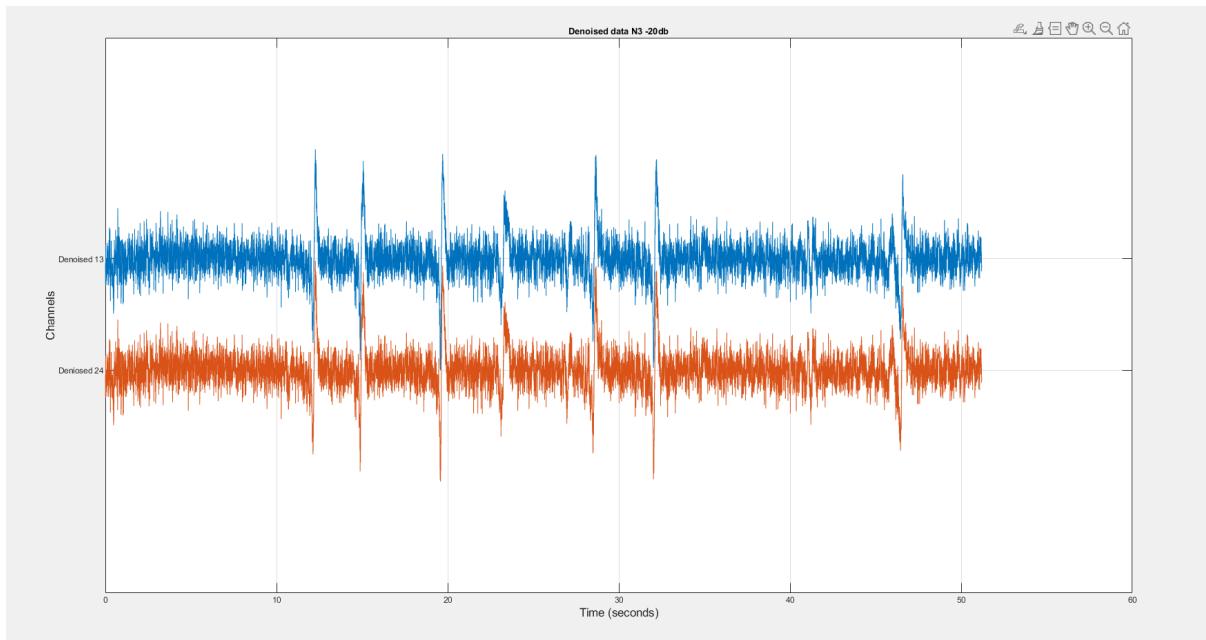
شکل ۲-۵ داده‌ی اصلی و نویز برای نویز ۴ با $SNR=-10db$



شکل ۲-۶ خروجی الگوریتم GEVD برای نویز ۴ با $SNR=-10db$



شکل ۷-داده‌ی اصلی و نویز برای نویز ۴ با $SNR=-20db$



شکل ۸-خروجی الگوریتم GEVD برای نویز ۴ با $SNR=-10db$

در نویز ۴ نیز شرایط همانند نویز، ۳ است و در اینجا نیز در خروجی‌ها اسپایک‌ها به راحتی قابل تشخیص هستند هرچند که در $SNR=-10db$ ، شکل خروجی شباهت بیشتری به شکل اصلی دارد.

نویز	RRMSE
Noise 3 SNR=-10db	۰.۹۶۵۵۵۸۶۳۴۱۴۳۶۰۹
Noise 3 SNR=-20db	۱.۱۳۵۲۵۱۰۶۹۲۶۹۴۲
Noise 4 SNR=-10db	۰.۹۹۵۷۴۵۳۸۷۶۷۱۳۳۵
Noise 4 SNR=-20db	۱.۰۵۹۴۲۹۱۰۸۳۱۴۸۸۸

با توجه به مقادیر به دست آمده برای RRMSE، مقدار خطای نسبی کل، برای نویزهایی با SNR=-20db بیشتر است و این مورد طبیعی است چرا که توان نویز در این موارد بیشتر است.

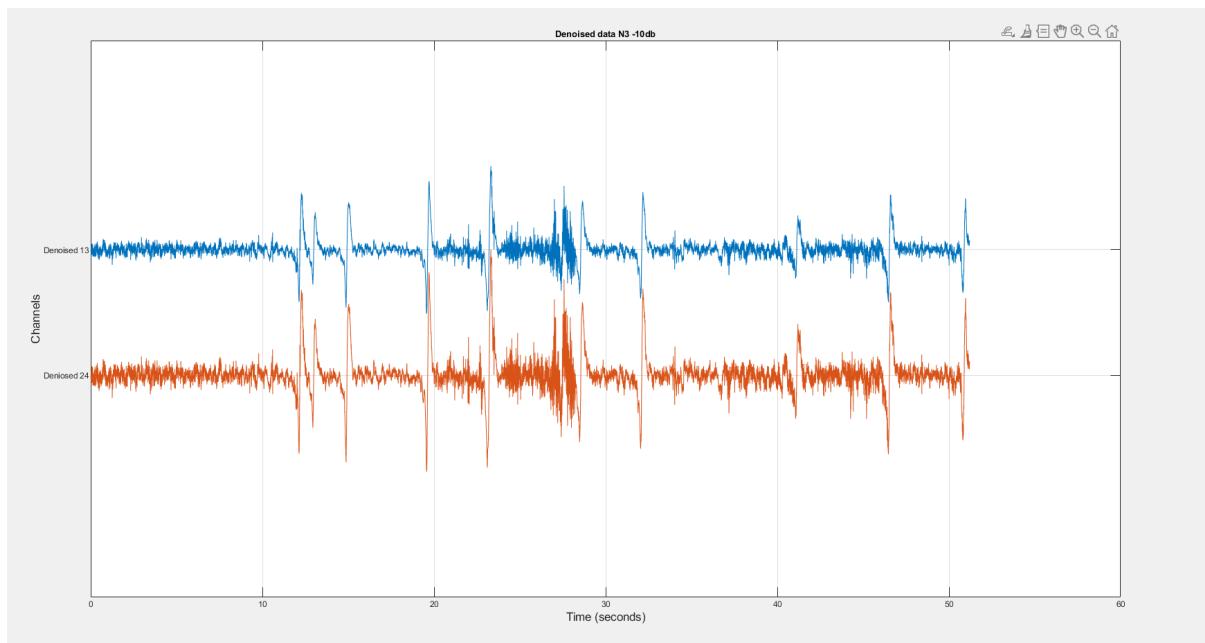
*** بخش ه بعد از خروجی‌های الگوریتم DSS می‌آید.

DSS - ۲-۲

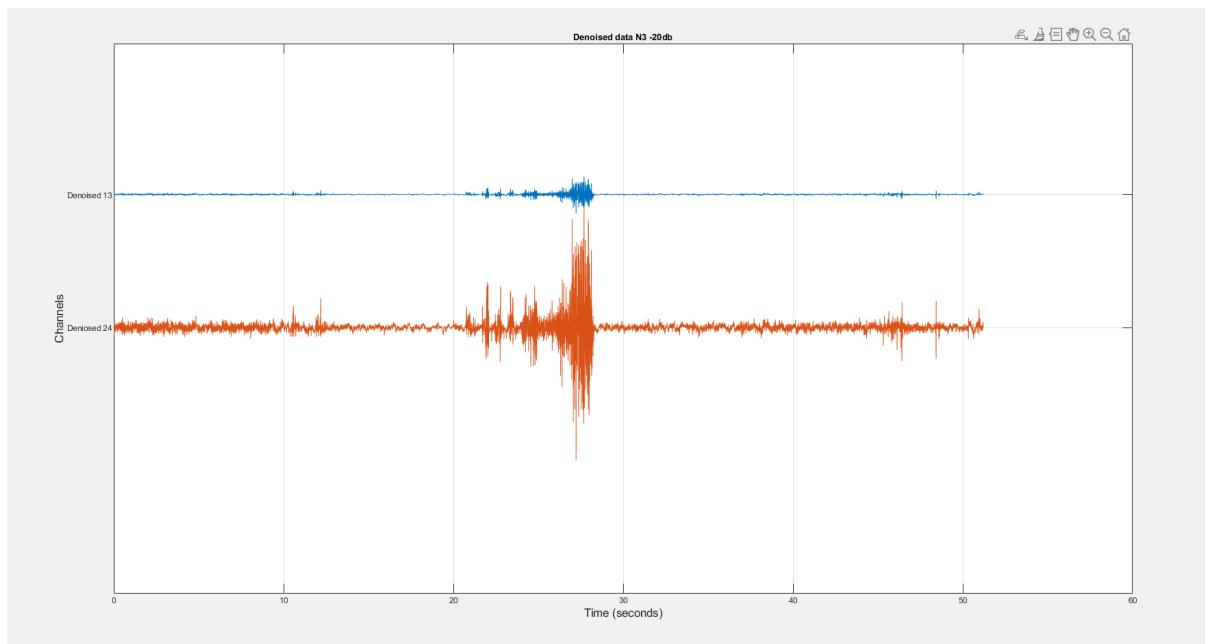
۱-۲-۲- الف

توضیحات این بخش در ۱-۱-۲- آمده است.

۲-۲-۲- ب و ج

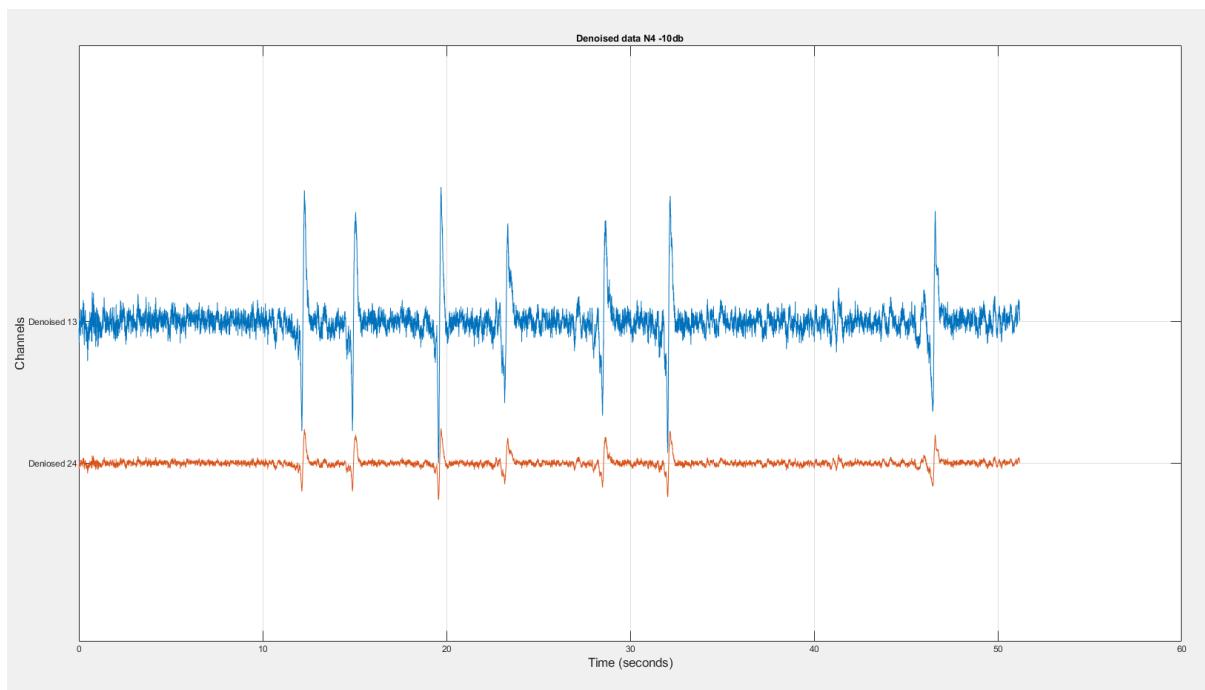


شکل ۲-۹ خروجی الگوریتم DSS برای نویز ۳ با SNR=-10db

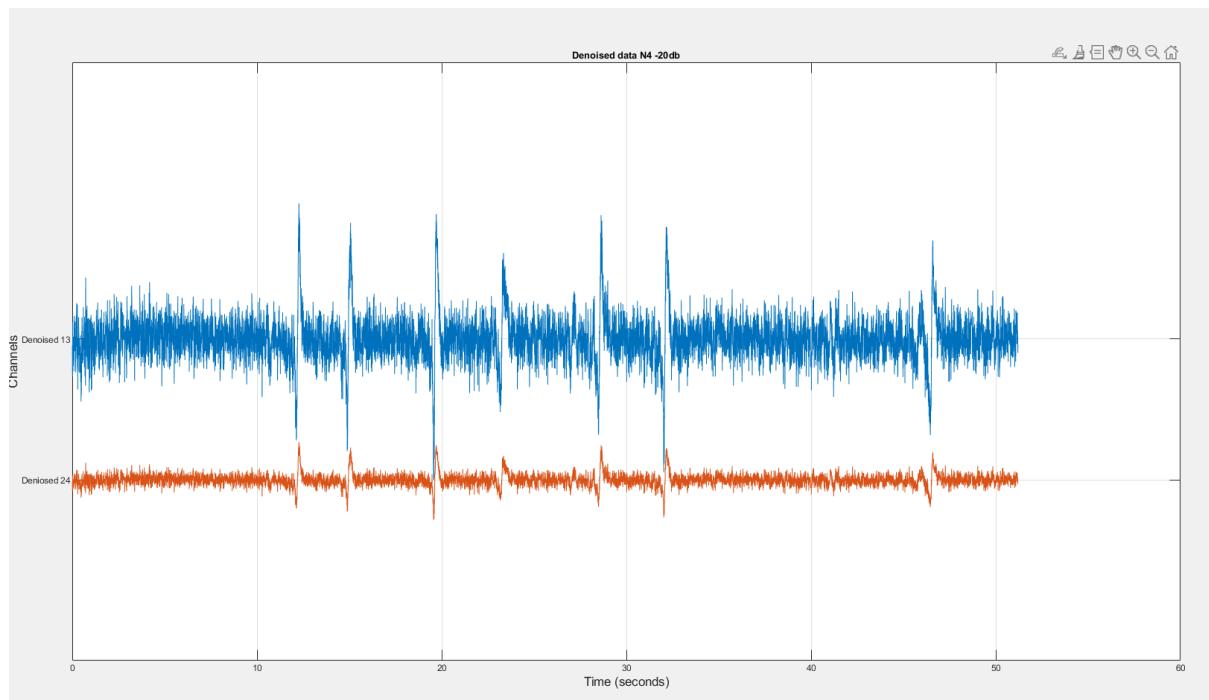


شکل ۲-۱۰ خروجی الگوریتم DSS برای نویز ۳ با $SNR=-20db$

در این روش، خروجی برای نویز ۳ با $SNR=-10db$ بهتر از روش GEVD است اما در مورد نویز با $SNR=-20db$ خروجی اصلاً شباهتی به خروجی اصلی ندارد، این مورد به علت توان بالای نویز است و الگوریتم DSS توانایی مقابله با این نویز را نداشته است.



شکل ۲-۱۱ خروجی الگوریتم DSS برای نویز ۴ با $SNR=-10db$



شکل ۲-۱۲ خروجی الگوریتم DSS برای نویز ۴ با $SNR=-20db$

در نویز ۴ بر خلاف نویز ۳، خروجی برای هر دو SNR قابل قبول است و می‌توان اسپایک‌هارا به راحتی تشخیص داد.

۳-۴-۲ د

نویز	RRMSE
Noise 3 SNR=-10db	۰.۸۴۵۵۵۸۶۳۴۱۴۳۶۰۹
Noise 3 SNR=-20db	۱.۰۳۵۲۵۱۰۶۹۲۶۹۴۲
Noise 4 SNR=-10db	۰.۸۹۵۷۴۵۳۸۷۶۷۱۳۳۵
Noise 4 SNR=-20db	۰.۹۶۹۴۲۹۱۰۸۳۱۴۸۸۸

در این روش نیز خطای نویزهایی با $SNR=-20db$ با توجه به بیشتر از دیگری است. چراکه در این SNR توان نویز بیشتر است و خطای را بالاتر می‌برد.

با مشاهدهی خروجی‌های به دست آمده و RRMSE در بین ۴ روش پیاده‌سازی شده، بهترین روش روش DSS است چرا که به خوبی توانسته که اسپایک‌هارا متمایز کند و همچنین پ RRMSE کمتری نسبت به دیگر روش‌ها دارد. به همین ترتیب دومین روش، روش GEVD روش سوم ICA و روش چهارم PCA است. روش PCA پایین‌ترین دقت و بیشترین خطای داشت و روش بهینه‌ای برای جداسازی منابع نمی‌باشد.