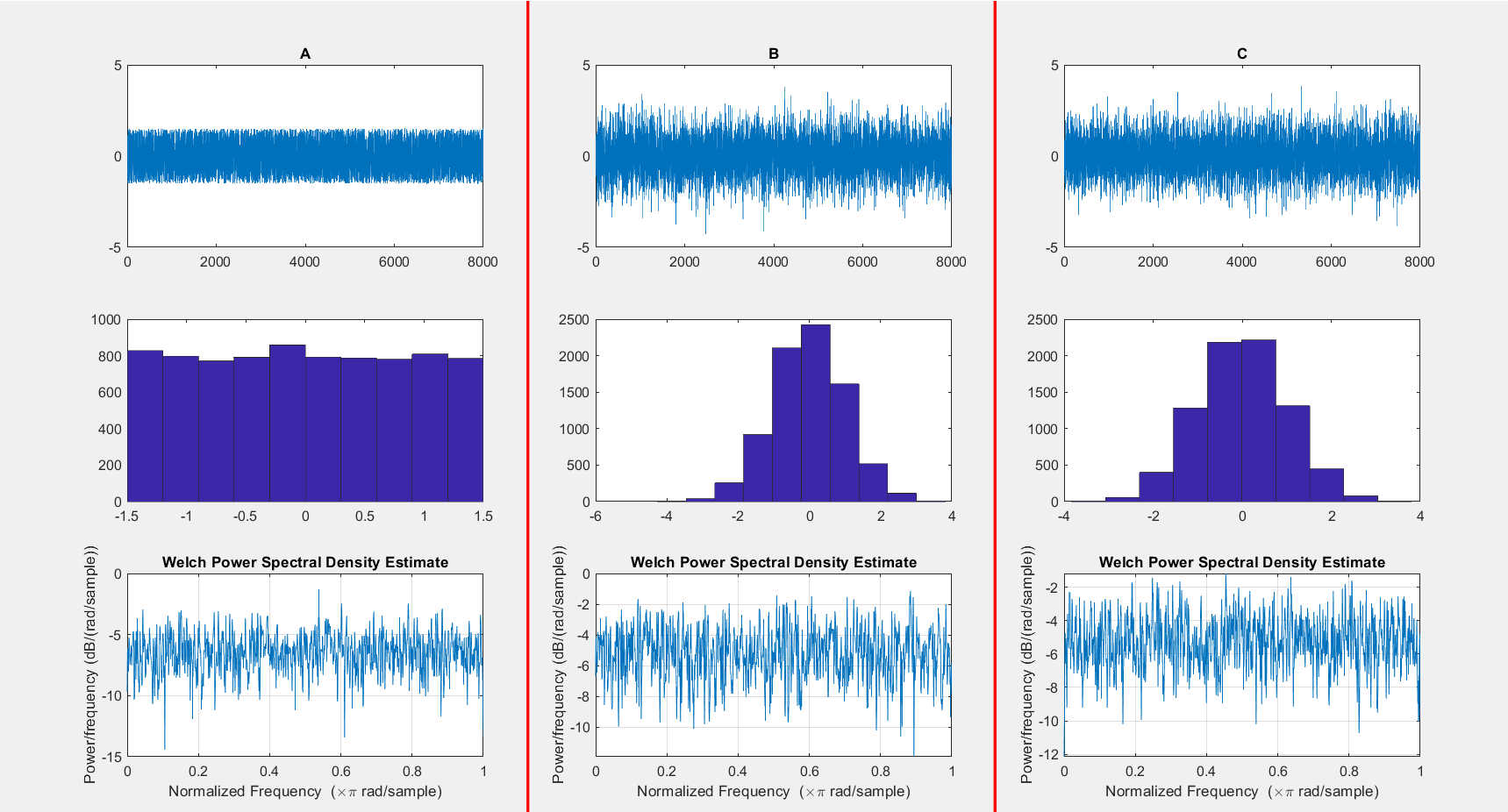
پاسخ سوال 1

تمامی بخش های خواسته شده در این سوال در شکل زیر است.



ستون اول تا سوم به ترتیب به قسمت های اول تا سوم اشاره دارند.

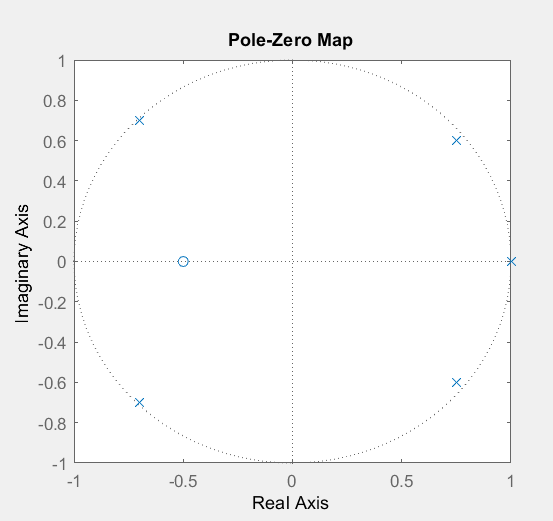
ردیف اول نویز را در حوزه زمان و ردیف دوم تابع توزیع (هیستوگرام) نویز و ردیف سوم نویز در حوزه فرکانس را نشان می‌دهد.

همانطور که مشخص است نویز قسمت سوم تقریب بسیار خوبی از نویز گوسی است و طیف آن تقریبا خط صاف است.(محور قائم لگاریتمی است).

پاسخ سوال 2

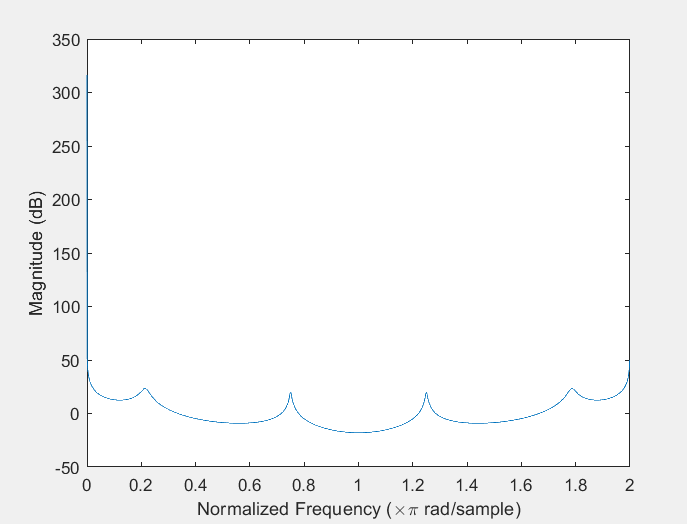
مدل ARMA داده شده دارای 5 قطب و یک صفر است که در شکل زیر نمایش داده شده اند. قطب های آن بسیار نزدیک به دایره واحد می‌باشند و یک قطب آن دقیقا روی دایره واحد است. فرم قطبی مکان دقیق قطب‌های آن در جدول زیر آمده است.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 0.9605,-0.2148pi | 0.9605,2.148pi | 0.9899,-0.75pi | 0.9899,0.75pi | roots(A) |



قسمت الف

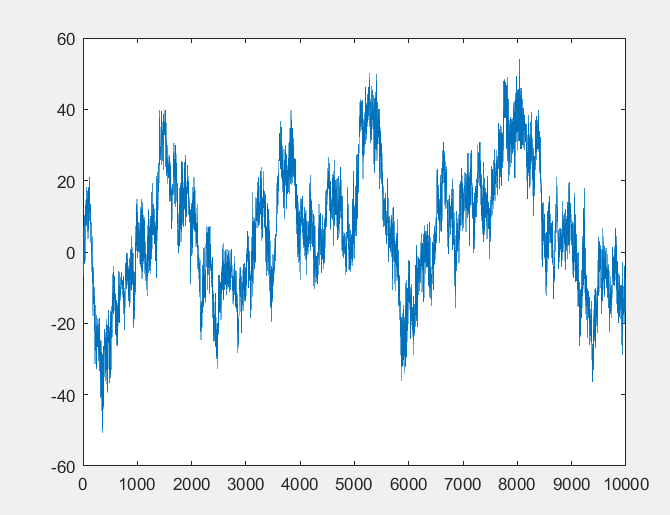
با استفاده از دستور freqz به راحتی می‌توان به طیف آن رسید. که در شکل زیر طیف آن برحسب db نمایش داده شده است.



بخاطر قطب در زوایه صفر روی دایره واحد، طیف در نزدیکی صفر به بینهایت میل ‌می‌کند. همچنین در زوایه گفته شده در جدول بالا مرتبط با مکان قطب ها، قله های نوک تیزی در آن زوایا در شکل بالا دیده می‌شود.

قسمت ب

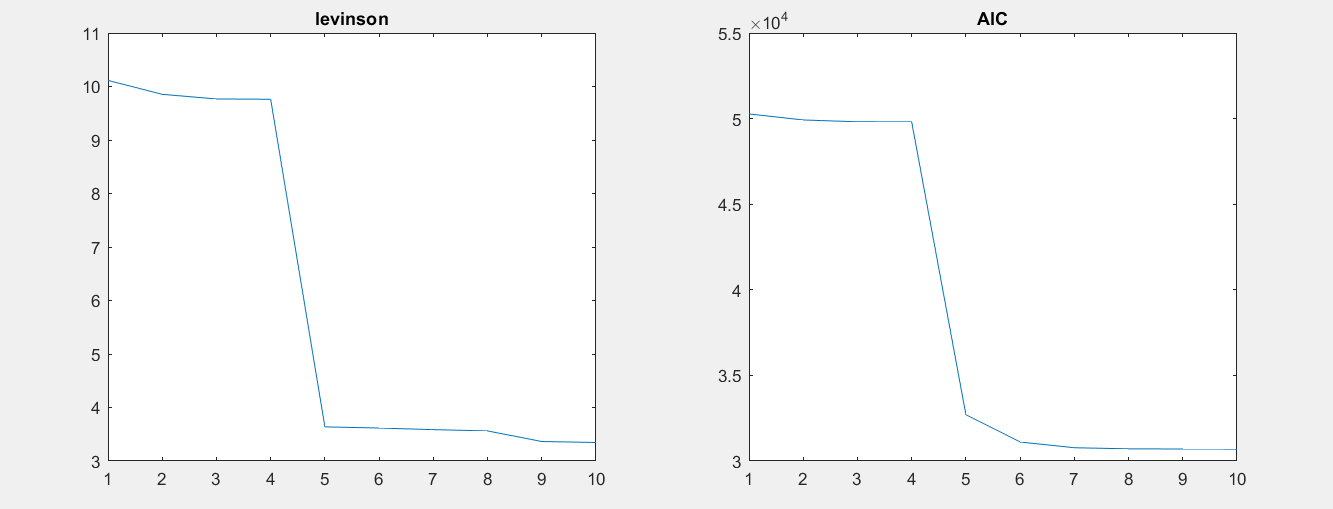
با استفاده از دستور filter و ورودی نویز سفید گوسی با واریانس واحد، سیگنال Y به شکل زیر ایجاد می‌شود.



قسمت پ

ابتدا به ازای مرتبه های مختلف p از مقدار 1 تا 10 پارامتر های AR با استفاده از دستور ar برای سیگنال Y معرفی شده در قسمت ب بدست آمد. سپس به ازای هر مقدار p مقادیر خطای مدل levinson-durbin و معیار AIC محاسبه شده است.

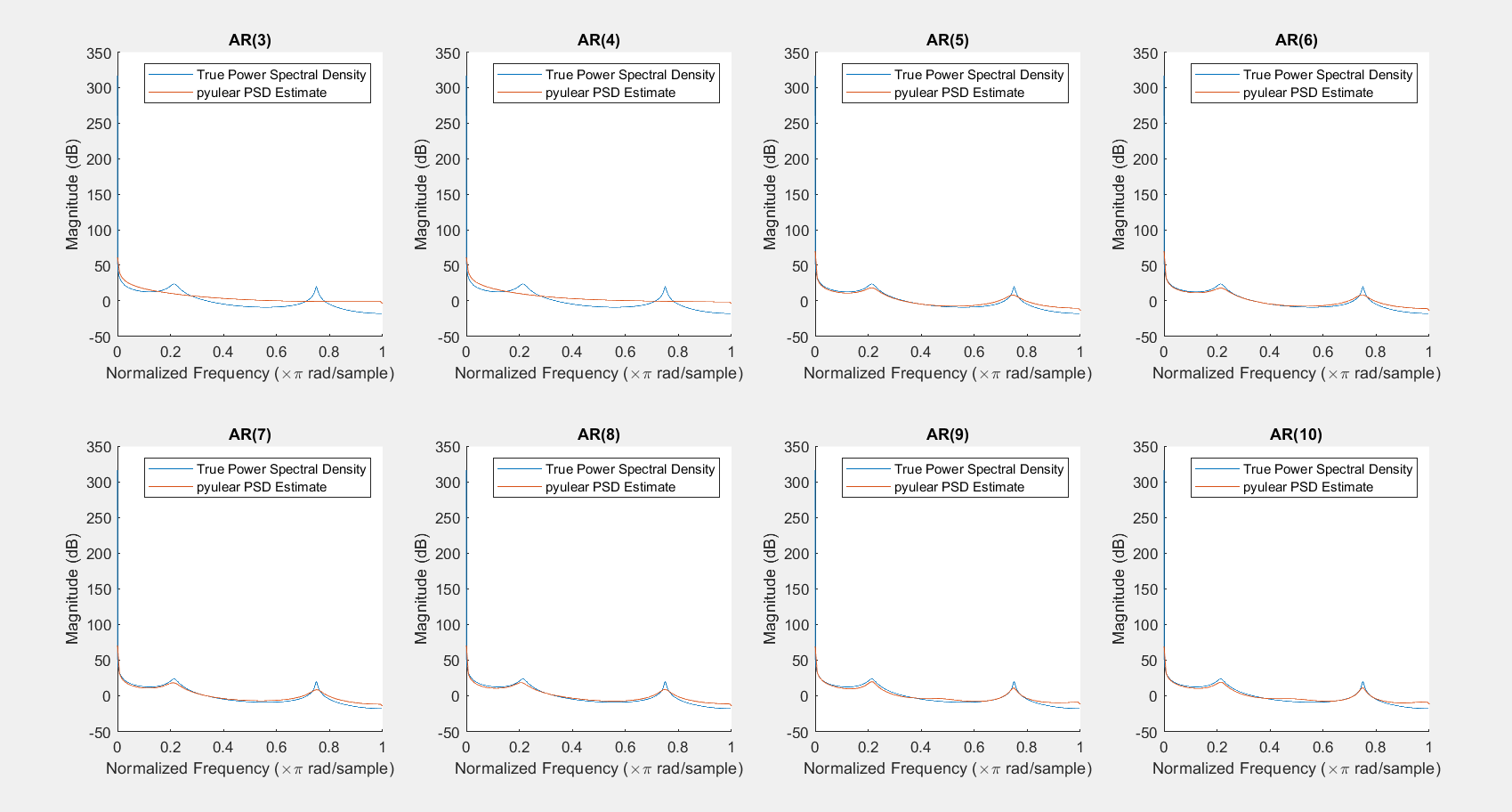
این مقادیر برحسب p در شکل زیر آمده است.



در این حالت شکل بالا به سادگی نشان می‌دهد که یک مقدار گپ بزرگ به ازای p = 5 اتفاق می‌افتد. این افت نشان می دهد که مقدار p = 5 مناسب است. همچنین معیار خطای levinson به صورت نسبی افت بیشتری را نشان می‌دهد این درحالیست که مقدار AIC به لحاظ مقداری افت قابل توجه ای را در مقدار مذکور داشته است.

قسمت ت

با استفاده از دستور pyulear به روش AR طیف مورد نظر برای مرتبه های 3 تا 10 رسم شده است.



؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟؟

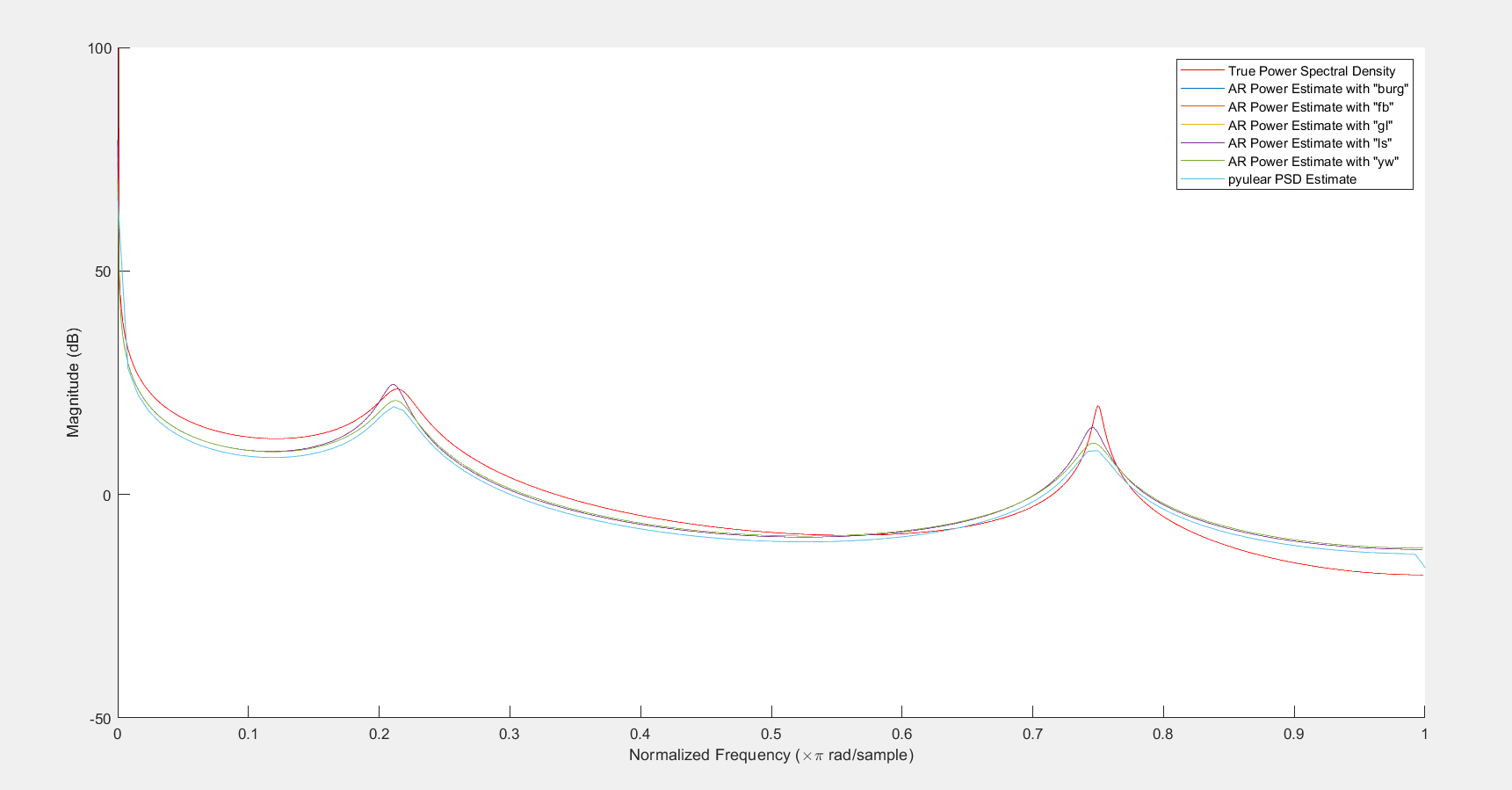
قسمت ث

ضرایب AR با استفاده از الگوریتم های مختلف در جدول زیر نشان داده است.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Algorithm | 1 | A1 | A2 | A3 | A4 | A5 |
| BURG | 1 | -1.16770 | -0.02374 | 0.01896 | 1.08697 | -0.91385 |
| FB | 1 | -1.16770 | -0.02370 | 0.01891 | 1.08697 | -0.91385 |
| GL | 1 | -1.16770 | -0.02374 | 0.01896 | 1.08697 | -0.91385 |
| LS | 1 | -1.16777 | -0.02369 | 0.01893 | 1.08708 | -0.91399 |
| YW | 1 | -1.16430 | -0.02254 | 0.01716 | 1.07153 | -0.90110 |

مشاهده می‌شود مقادیر BURG با GL یکی است. همچنین مقادیر آن ها تا 4 رقم اعشار با FB برابر است.

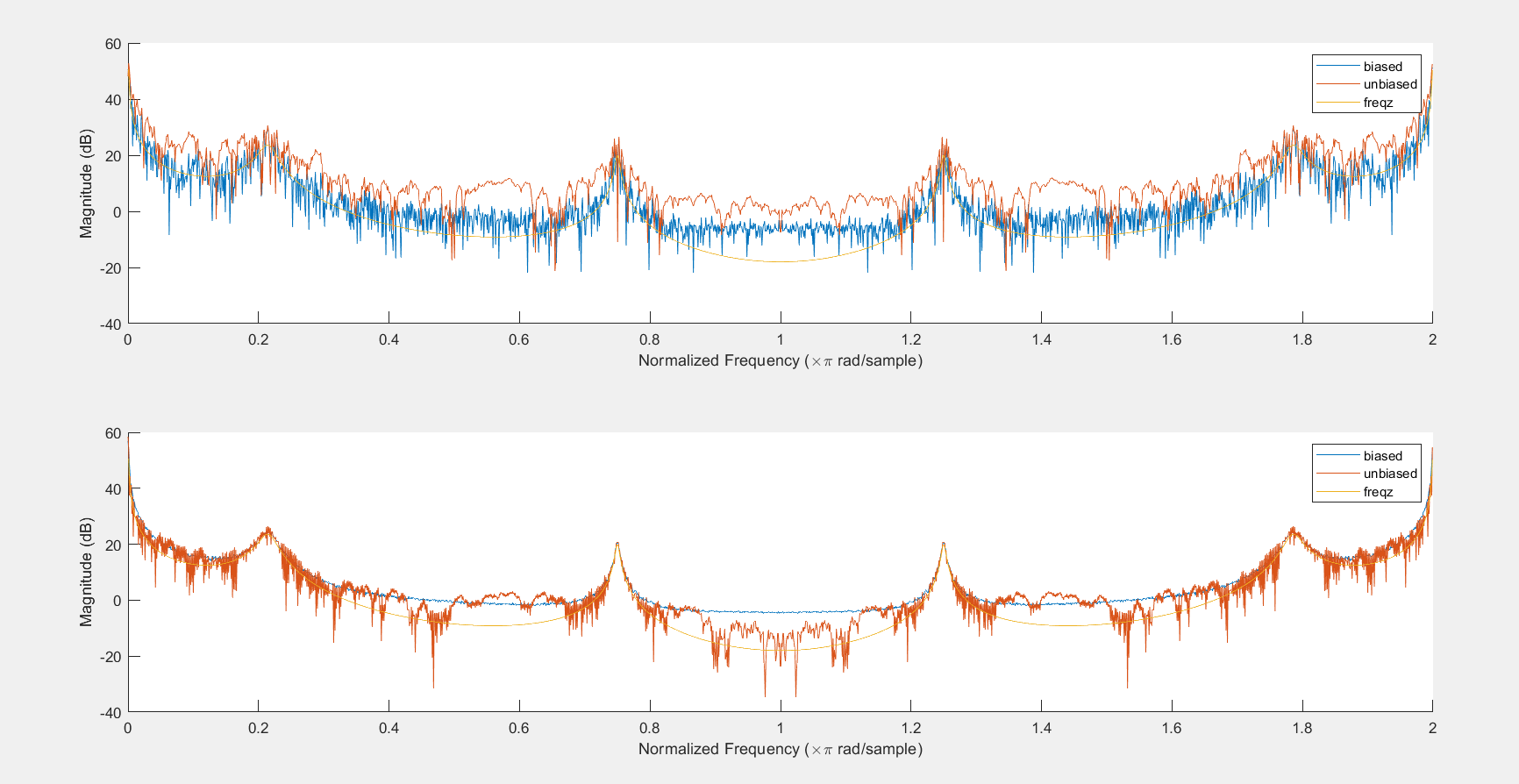
طیف سیگنال به ازای روش های مختلف بالا به همراه مقایسه با طیف واقعی و همچنین دستور pyulear به صورت زیر نمایش داده شده است.



پاسخ سوال 3

قسمت الف

روش BT اعمال شده است. موارد خواسته شده در زیر آمده است.

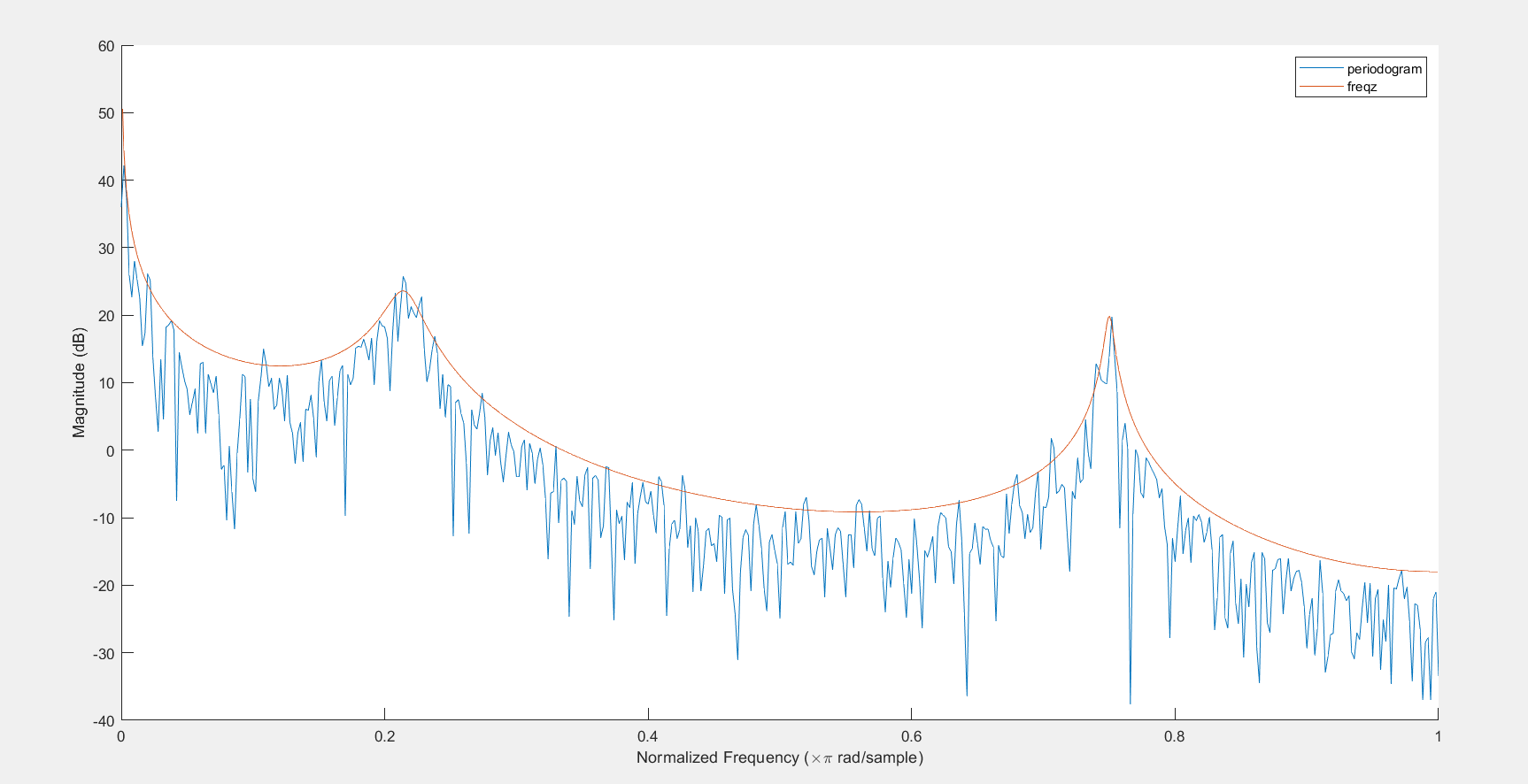


قسمت ب

پاسخ این قسمت در قسمت الف آمده است.

قسمت پ

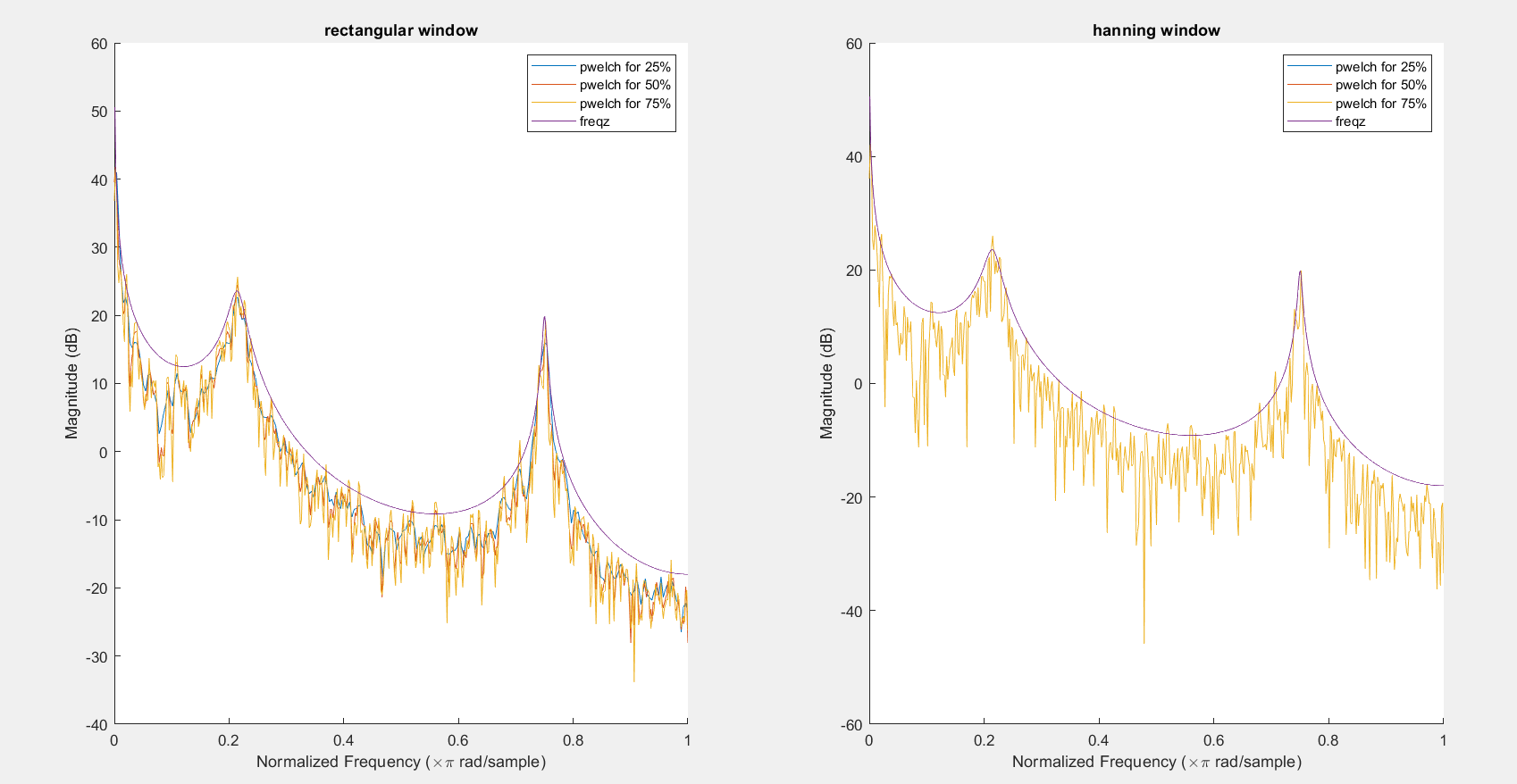
روش پریودوگرام با استفاده از دستور peridogram و اعمال پنجره هنینگ به صورت زیر خواهد بود.



قسمت ت

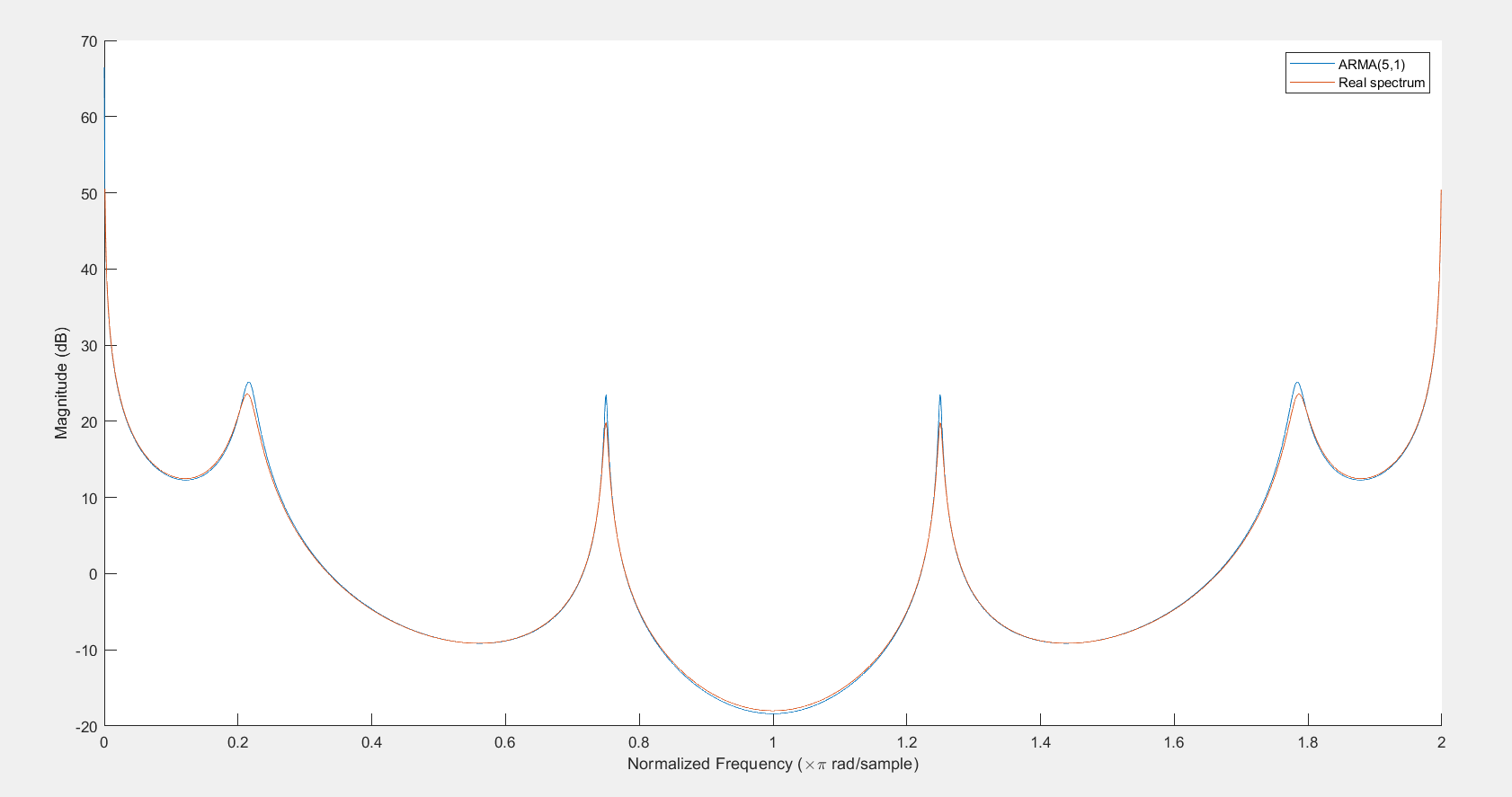
در این سوال یک بار با استفاده از پنجره مستطیلی و یک بار با استفاده از پنجره هنینگ با روش WOSA (pwelch) رسم شده است.

تفاوت میانگین و واریانس آن ها در شکل زیر مشخص است. مورد جالب آن این است که به ازای مقادیر مختلف هم‌پوشانی با استفاده از پنجره هنینگ تمامی آن مقادیر روی هم قرار گرفته اند.



قسمت ث

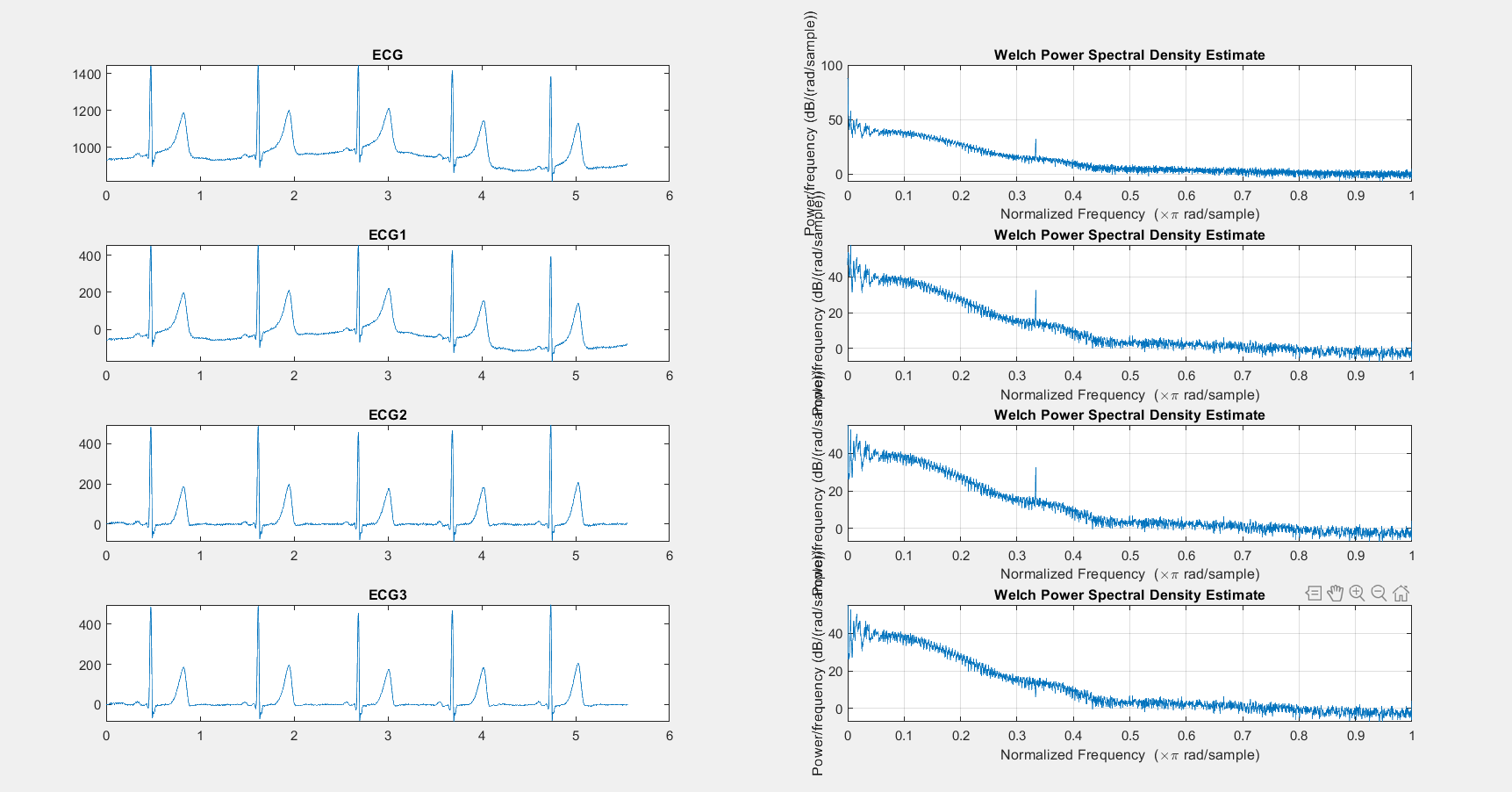
همانطور که در شکل زیر مشخص است تخمین پارامترهای ARMA(5,1) به خوبی با واقعیت جور شده است.



پاسخ سوال 4

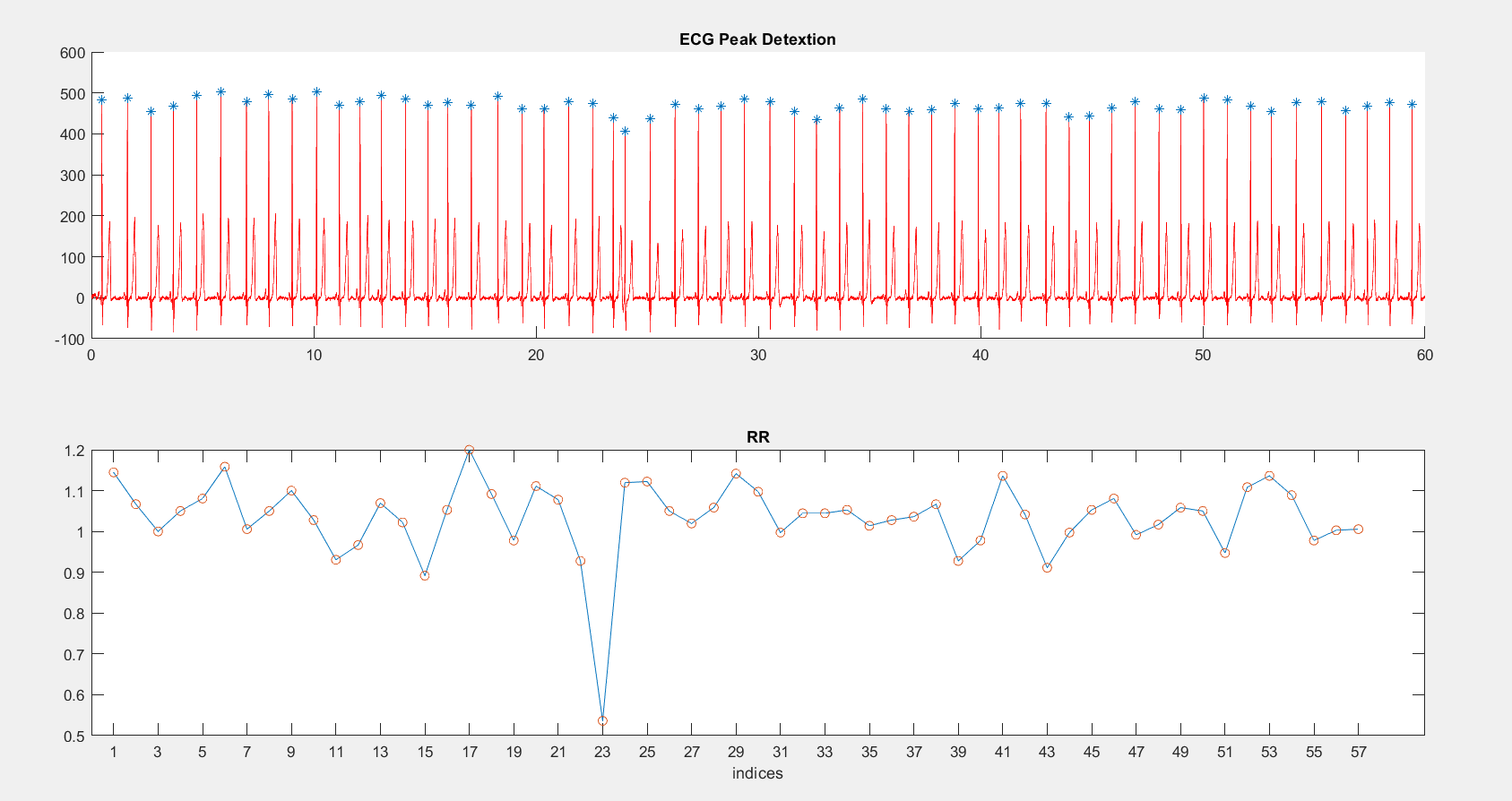
قسمت الف

سیگنال های ECG و ECG1 و ECG2 و ECG3 به همراه طیف آن‌ها در شکل زیر آمده اند.



قسمت ب

سیگنال ECG3 به همراه مقادیر یافته شده R در شکل زیر (قسمت بالایی) و سیگنال تاکوگرام (قسمت پایینی) در زیر رسم شده است.



روش بدست آوردن قله های ECG (همان R ها) در زیر تشریح شده است.

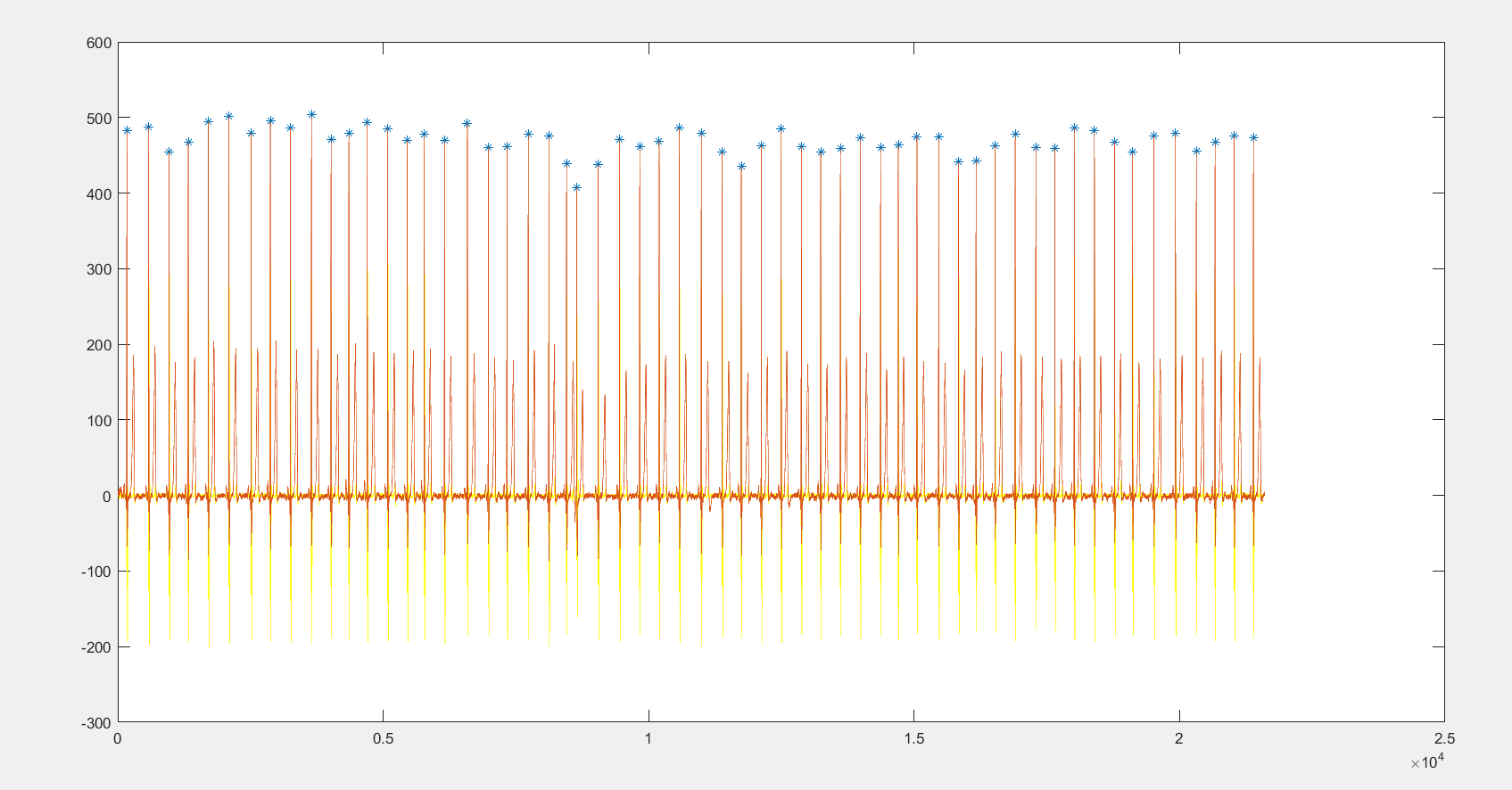
با استفاده از تابع myFindPeaks3

مهم ترین خاصیت استفاده شد در این تابع استفاده از یک فیلتر FIR است. این فیلتر از دو moving average مستطیلی درست شده است. اولی با طول mainLobe و دامنه 1/mainLobe و دومی به طول 2\*gaurd و دامنه -1/(2\*guard) ساخته شده است به طوری که فیلتر دوم را وسط بریده و جدا میکنیم و فیلتر اول را در وسط آن قرار می‌دهیم.

واضح است که میانگین این فیلتر صفر است پس حتما یک صفر در z=1 دارد.

به ازای mainLobe = 3 و guard = 5 خروجی به رنگ زرد در شکل زیر رسم شده است. تصویر قرمز رنگ خود سیگنال ECG است. همانطور که مشخص است صرفا در زمان هایی که یک پیک ناگهانی رخ می‌دهد مقدار بسیار قابل توجهی را دارد و در خارج آن بازه مقدار آن ناچیز خواهد بود بنابراین به راحتی می‌توان آن را با مقدار آستانه مقایسه کرد. در این روش قله های T از بین می‌روند و عملکرد آستانه را بهبود می‌بخشد.

توجه کنید که بخاطر طول فیلتر اندکی اندیس قله ها جابجا شده است که آن ها را به سادگی می‌توان کنترل نمود.



اگر بخواهیم این کار را با پکیج OSET امتحان کنیم دو تابع زیر را به شکل زیر فراخوانی می‌کنیم. مشاهده می‌شود idx و idx2 با peaksIdx یکسان خواهد بود.

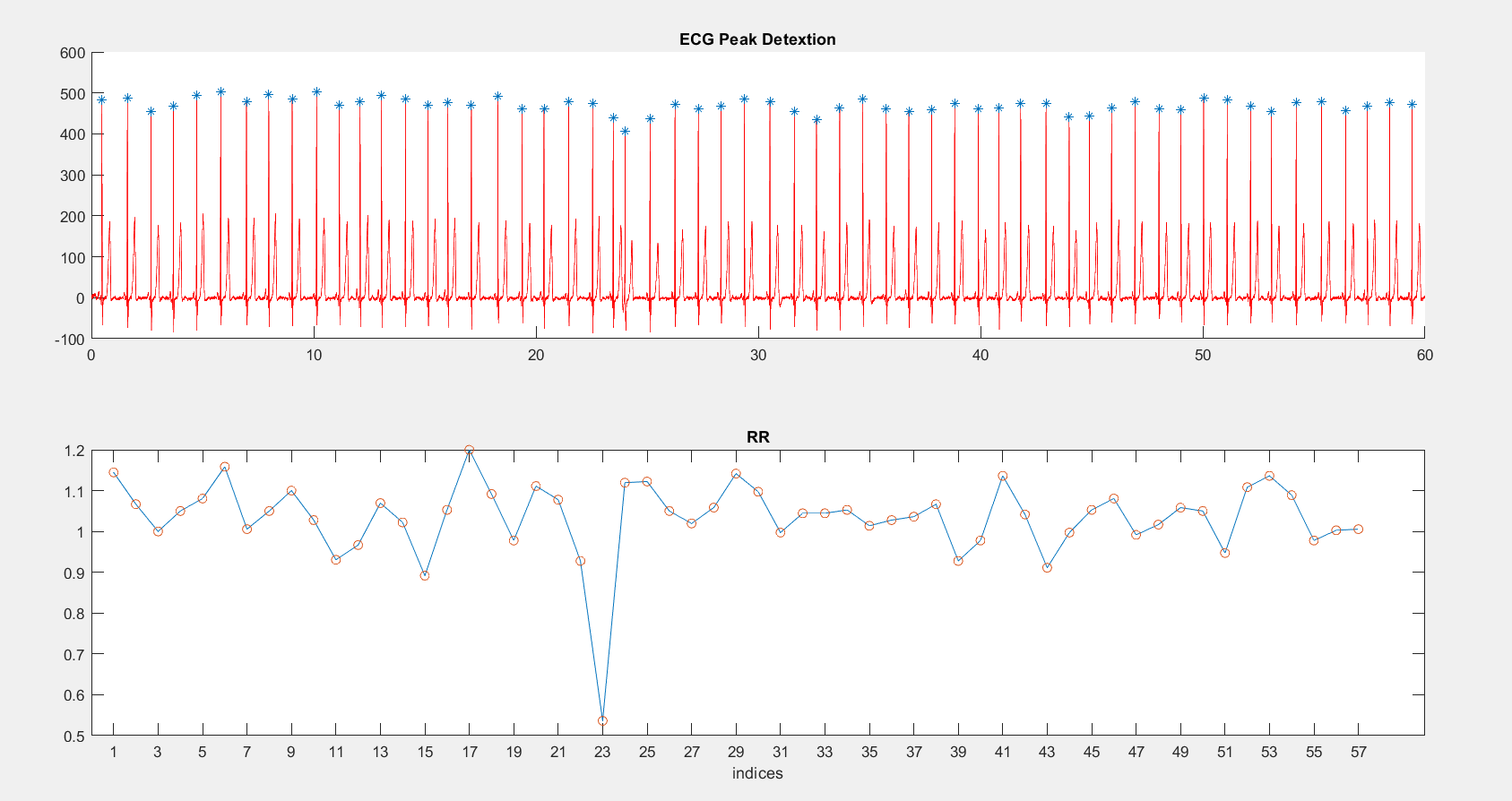
idx = find(PeakDetection(ECG3,1/fs));

idx2 = find(PeakDetection2(ECG3,180,1));

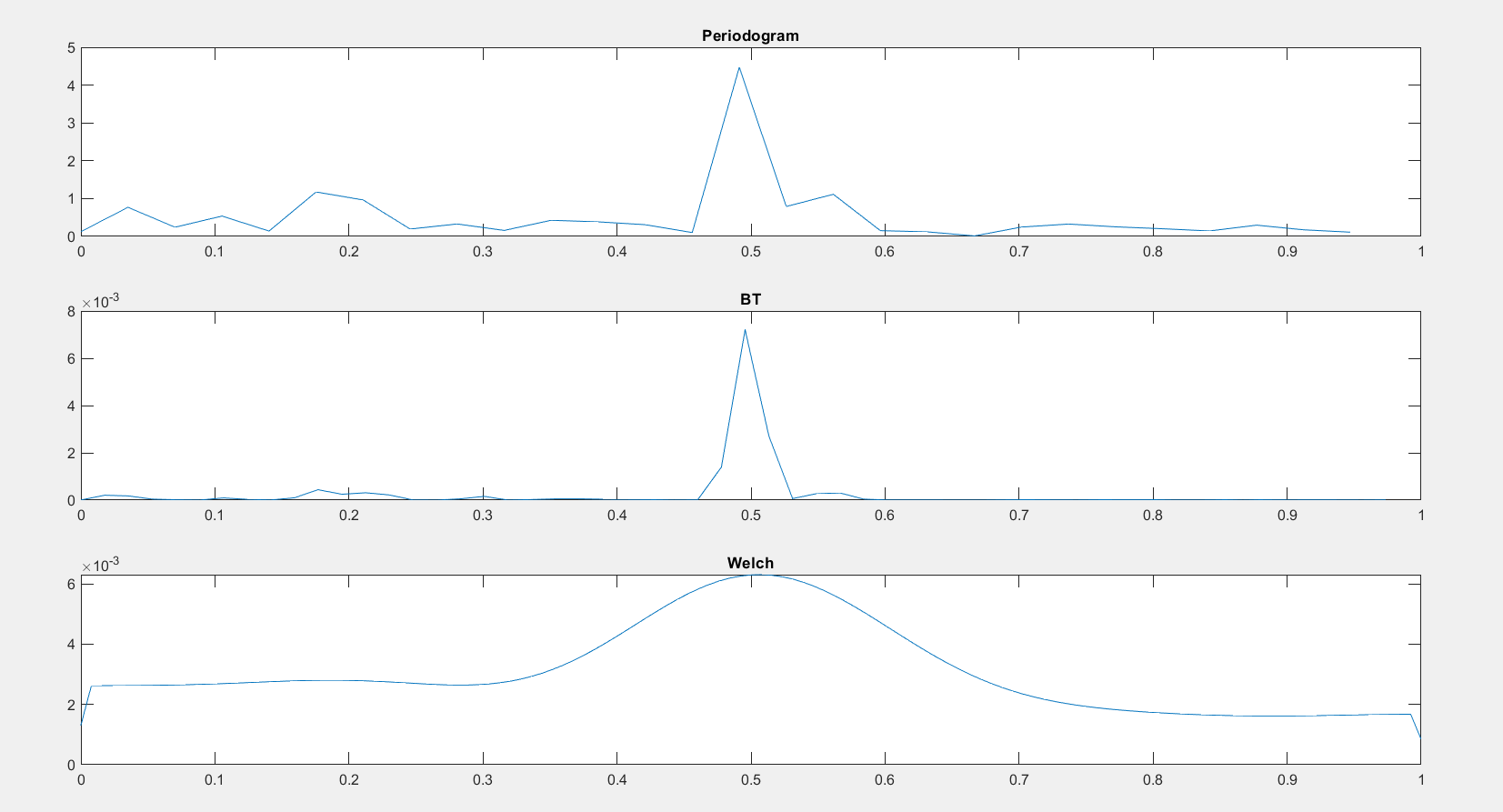
نکته قابل توجه این است که خروجی این توابع یک آرایه صفر و یکی به طول سیگنال ورودی آن است که در مکان های قله دار مقدار آن یک و در خارج آن صفر است از این رو تابع find را باید روی آن اجرا کرد تا به اندیس مکان های قله برسیم.

قسمت پ

در شکل اول قسمت ب سیگنال تاکوگرام محاسبه و رسم شده است. که در زیر نیز آمده است.

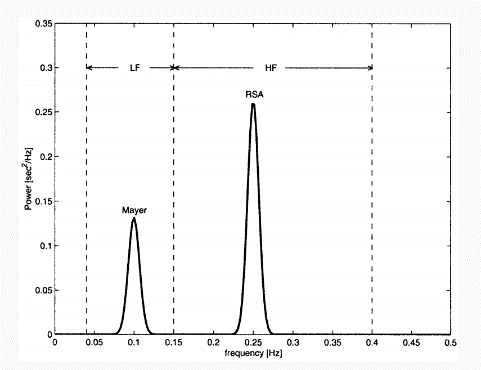


طیف این سیگنال به سه روش peridogram و BT و Welch محاسبه و در شکل زیر رسم شده است.



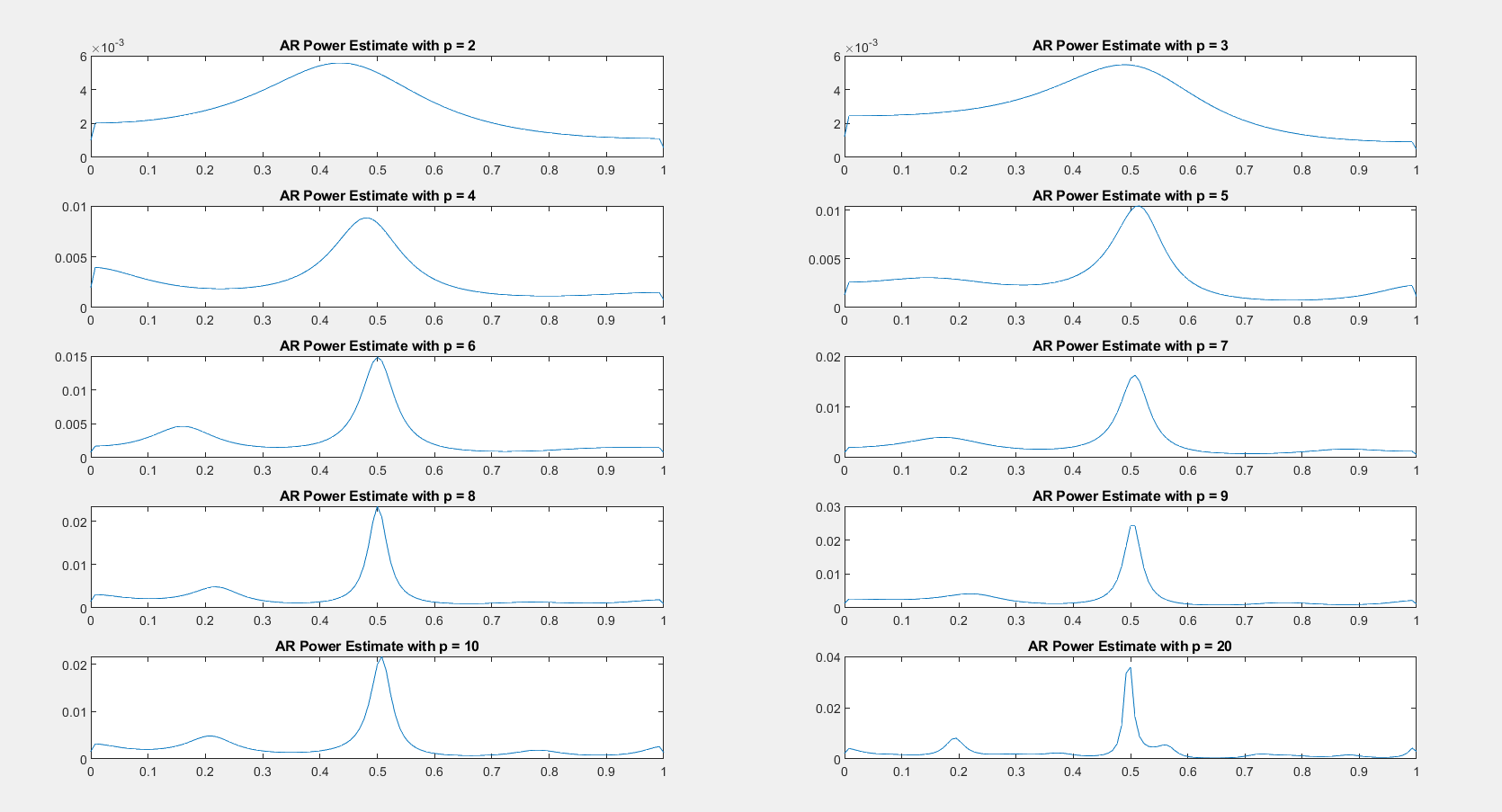
RSA

Mayer

در شکل بالا مکان تقریبی دو بخش RSA و Mayer قابل مشاهده است. البته روش welch نتیجه مناسبی را نشان نمی‌دهد.

این دو پیک در شکل روبرو قابل مشاهده است.

همچنین با دستور pyulear نیز طیف سیگنال محاسبه شده است. این دستور به روش AR طیف را تخمین می‌زند و سرعت بسیار بالایی را دارد.



RSA

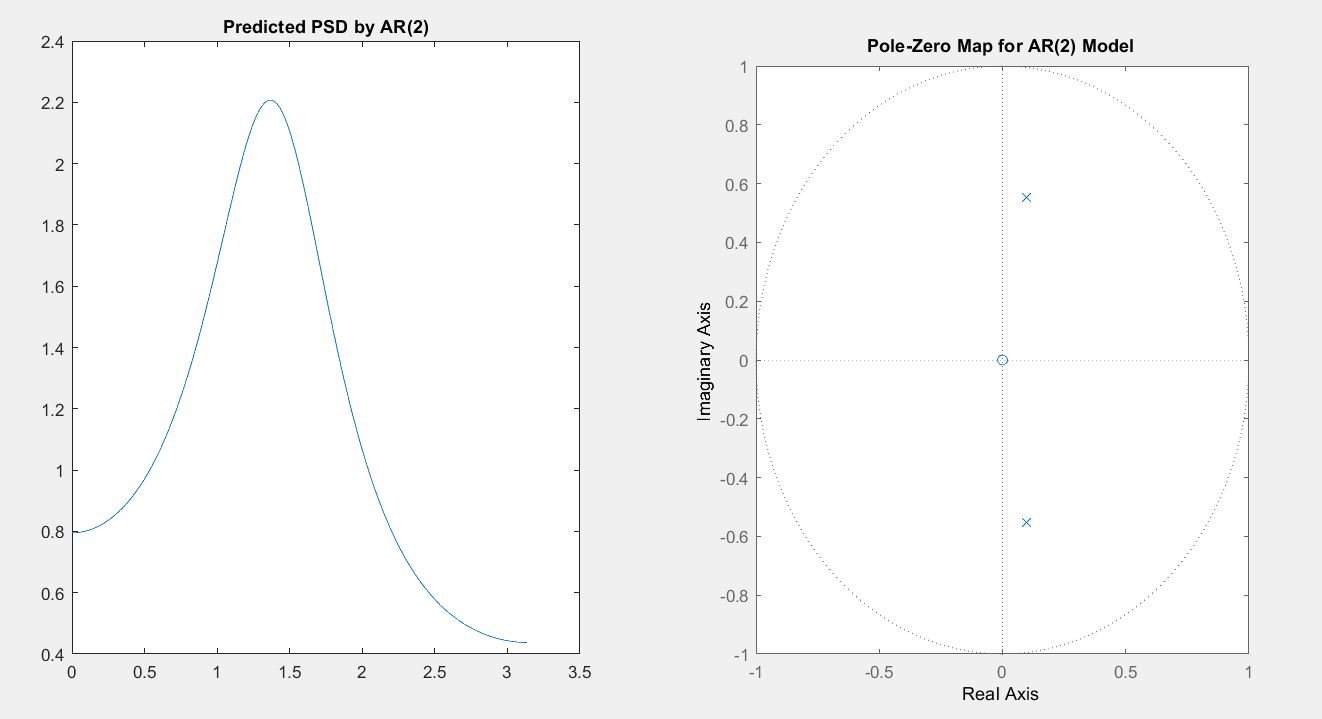
RSA

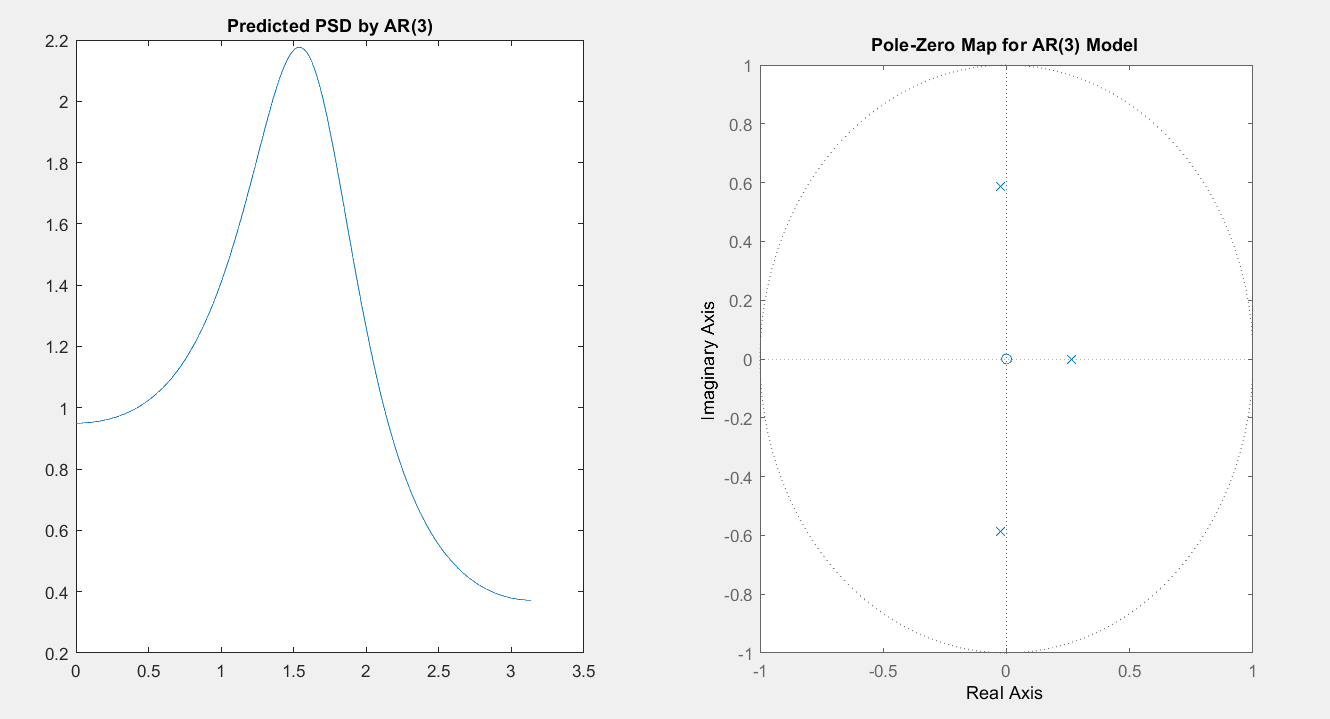
Mayer

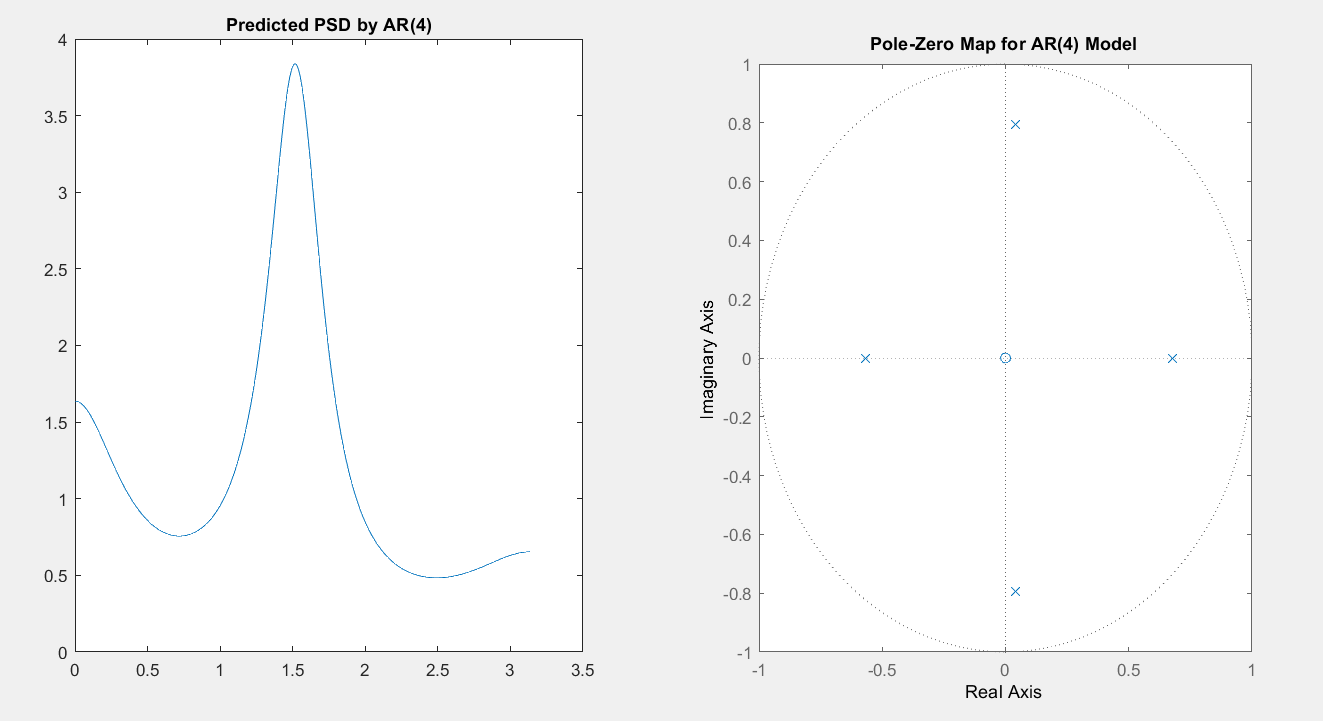
Mayer

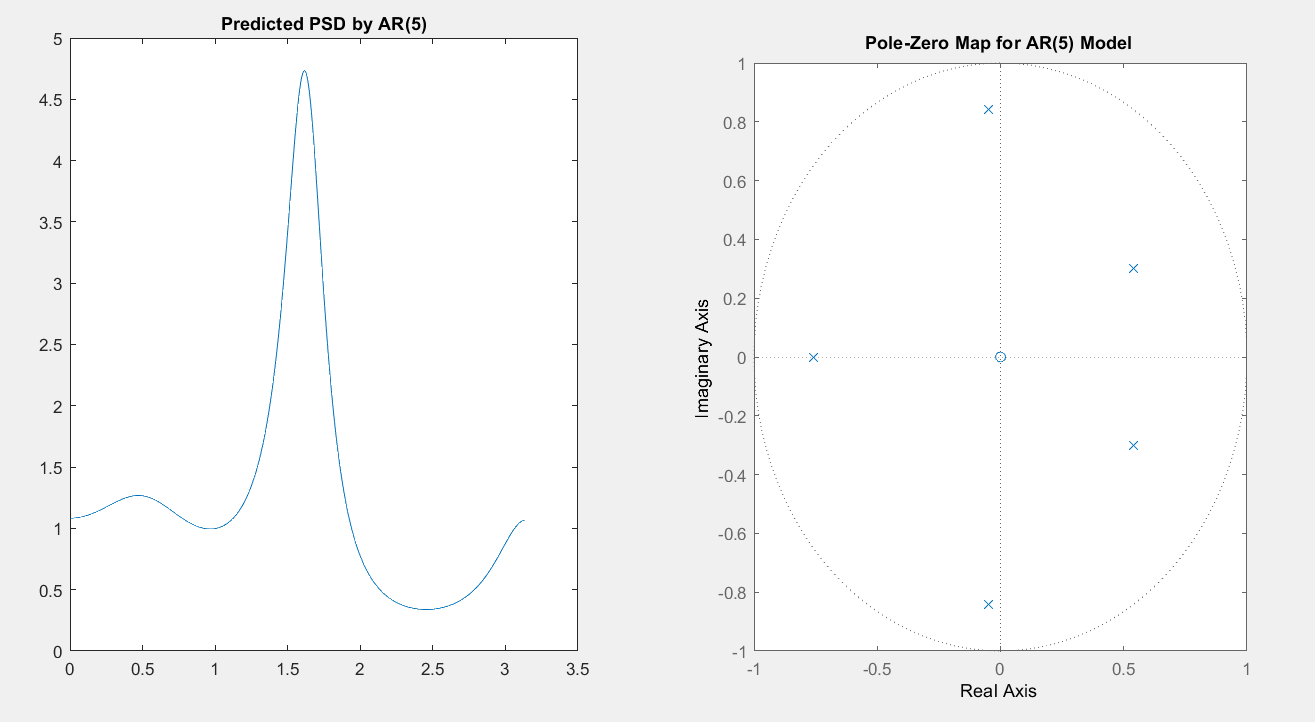
مشاهده می‌شود که اگر مرتبه AR کمتر از پنج باشد پیک mayer دیده نمی‌شود و برای مرتبه 5 این مقدار بسیار ضعیف است از مرتبه 6 به بالا این سیگنال بیشتر خود را نشان می‌دهد و نتایج دقیق تری را می‌سازد.

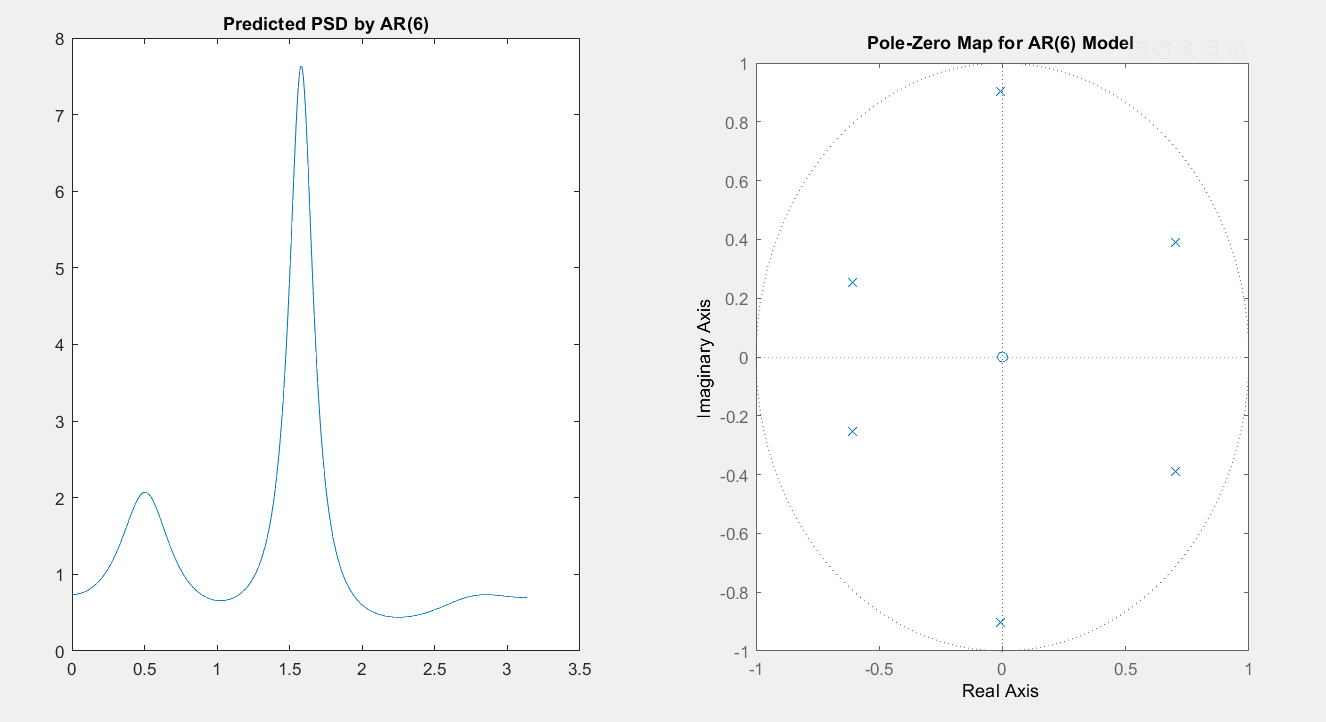
در شکل های زیر علاوه بر طیف سیگنال آرایش قطب‌ها نیز ثبت شده است.

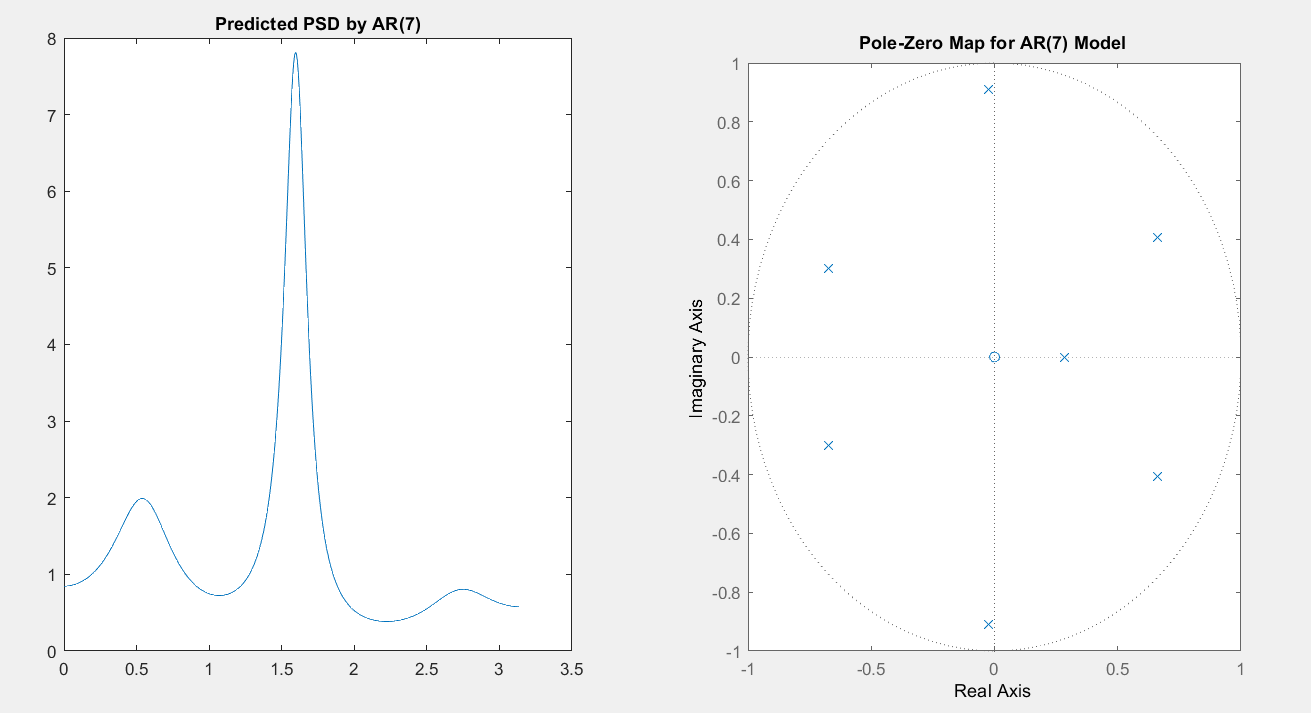


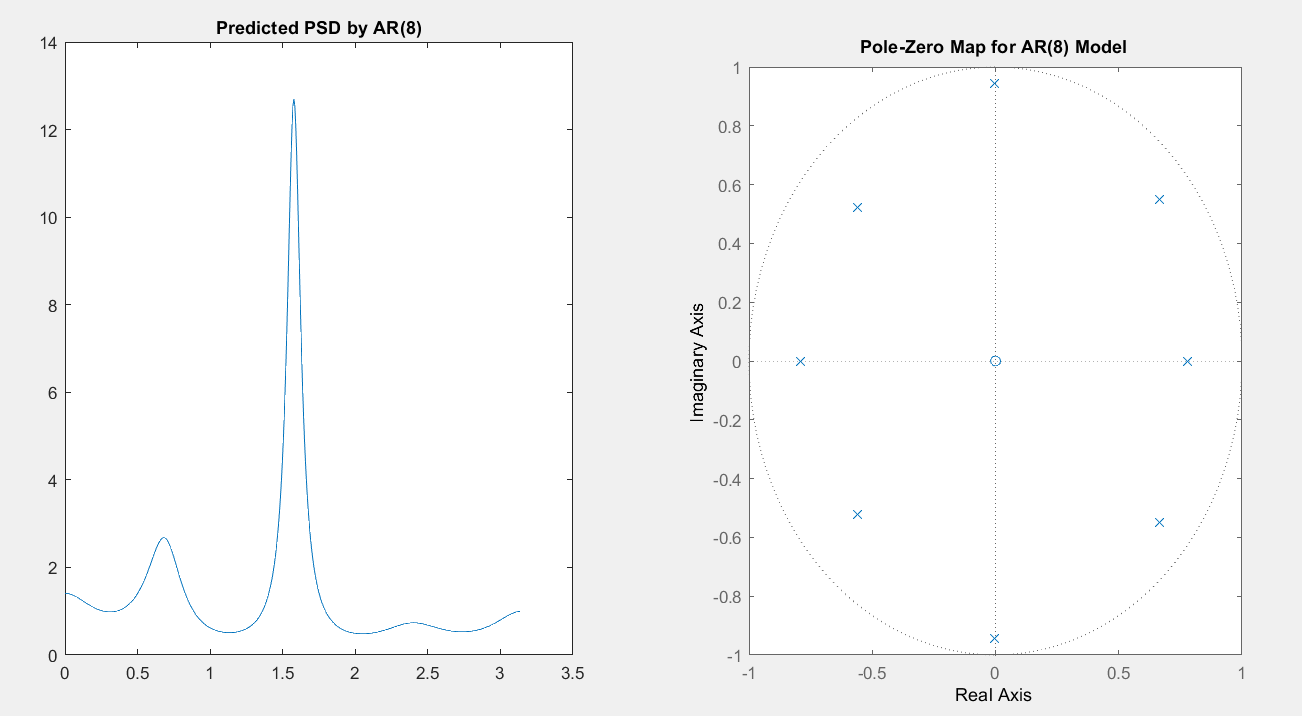


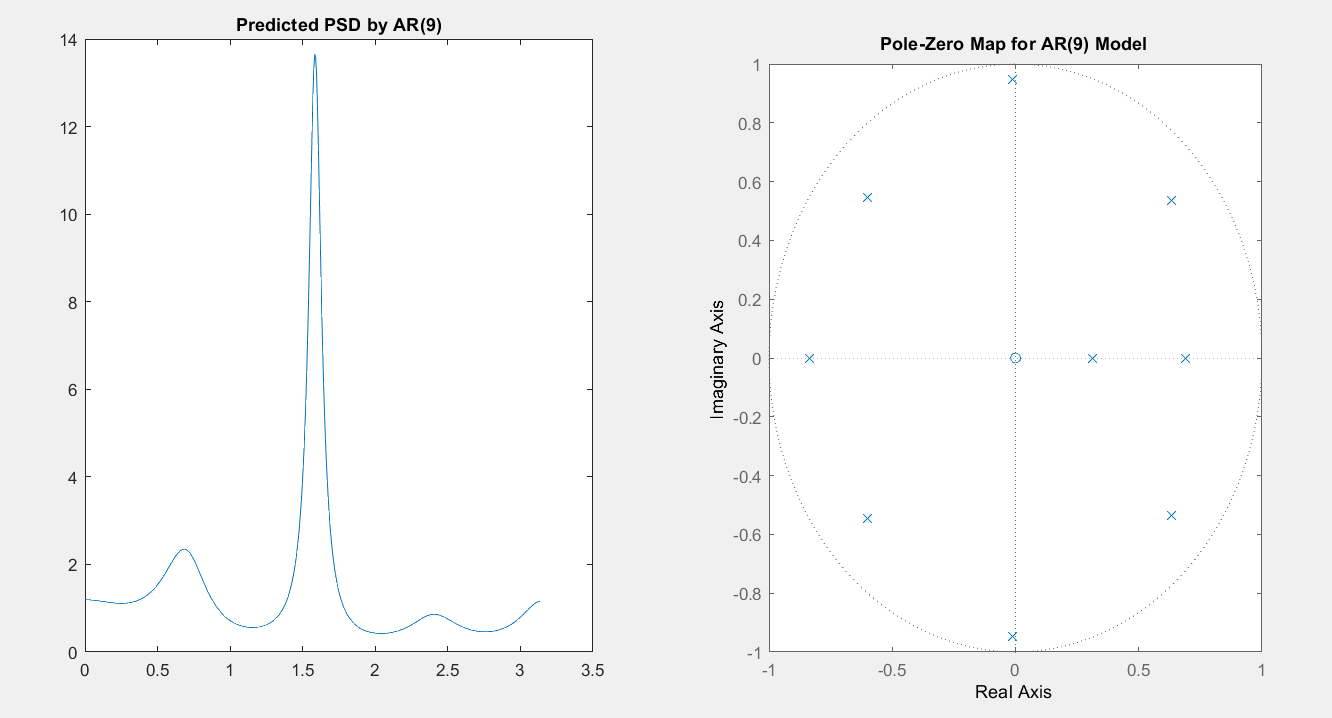


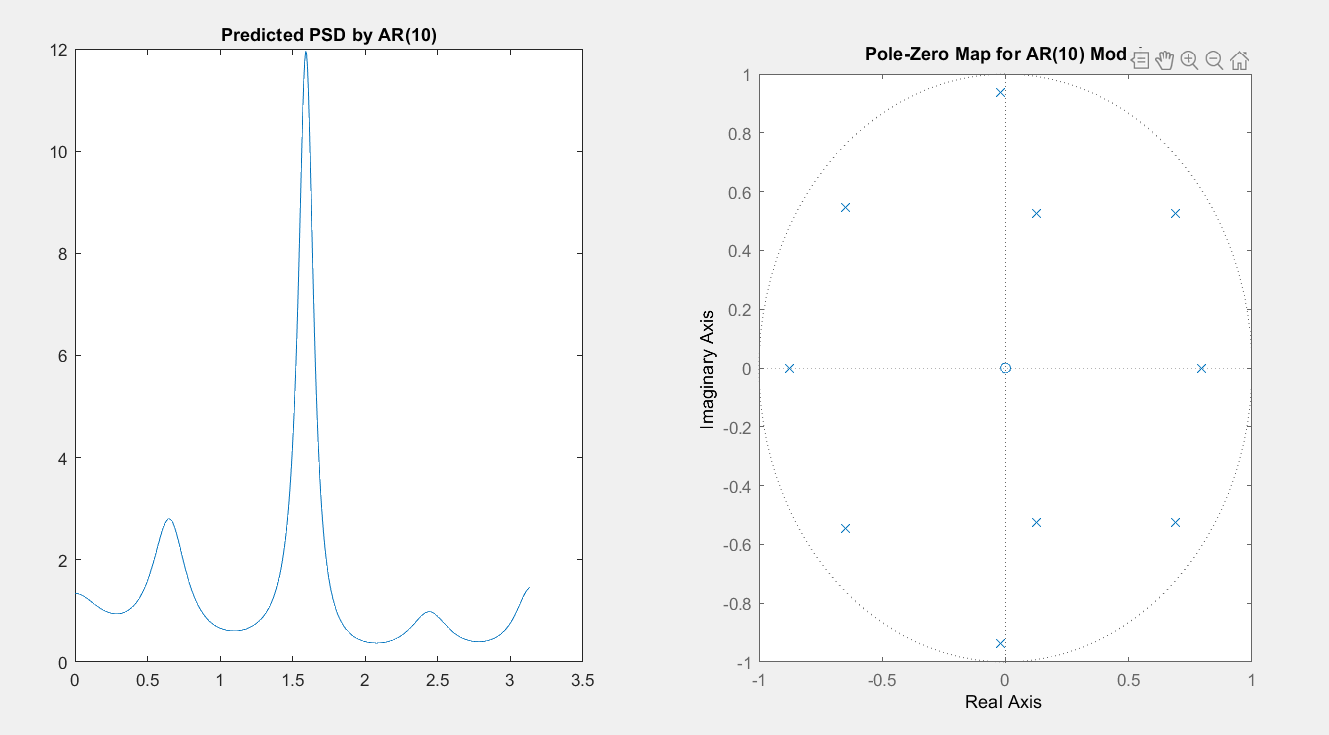


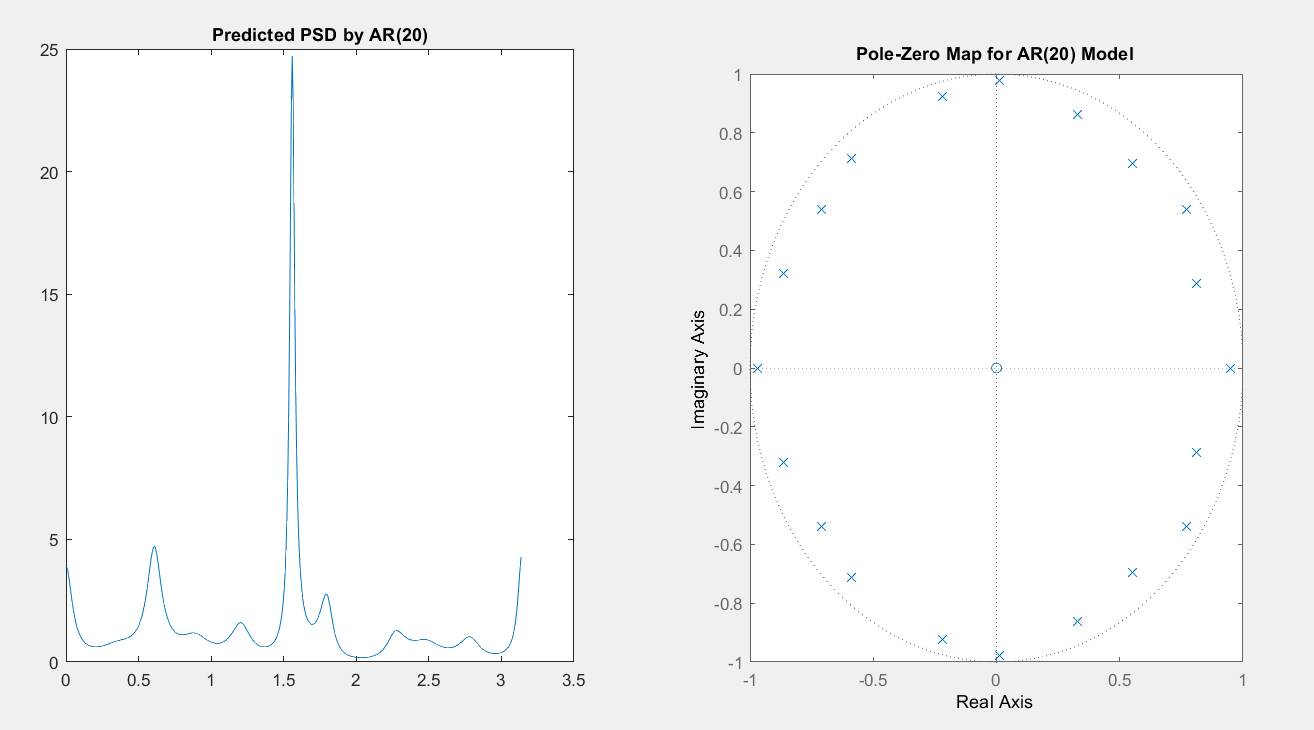












پاسخ سوال 5

دیتاست این قسمت از لینک زیر دانلود شده است.

<https://archive.physionet.org/cgi-bin/atm/ATM>

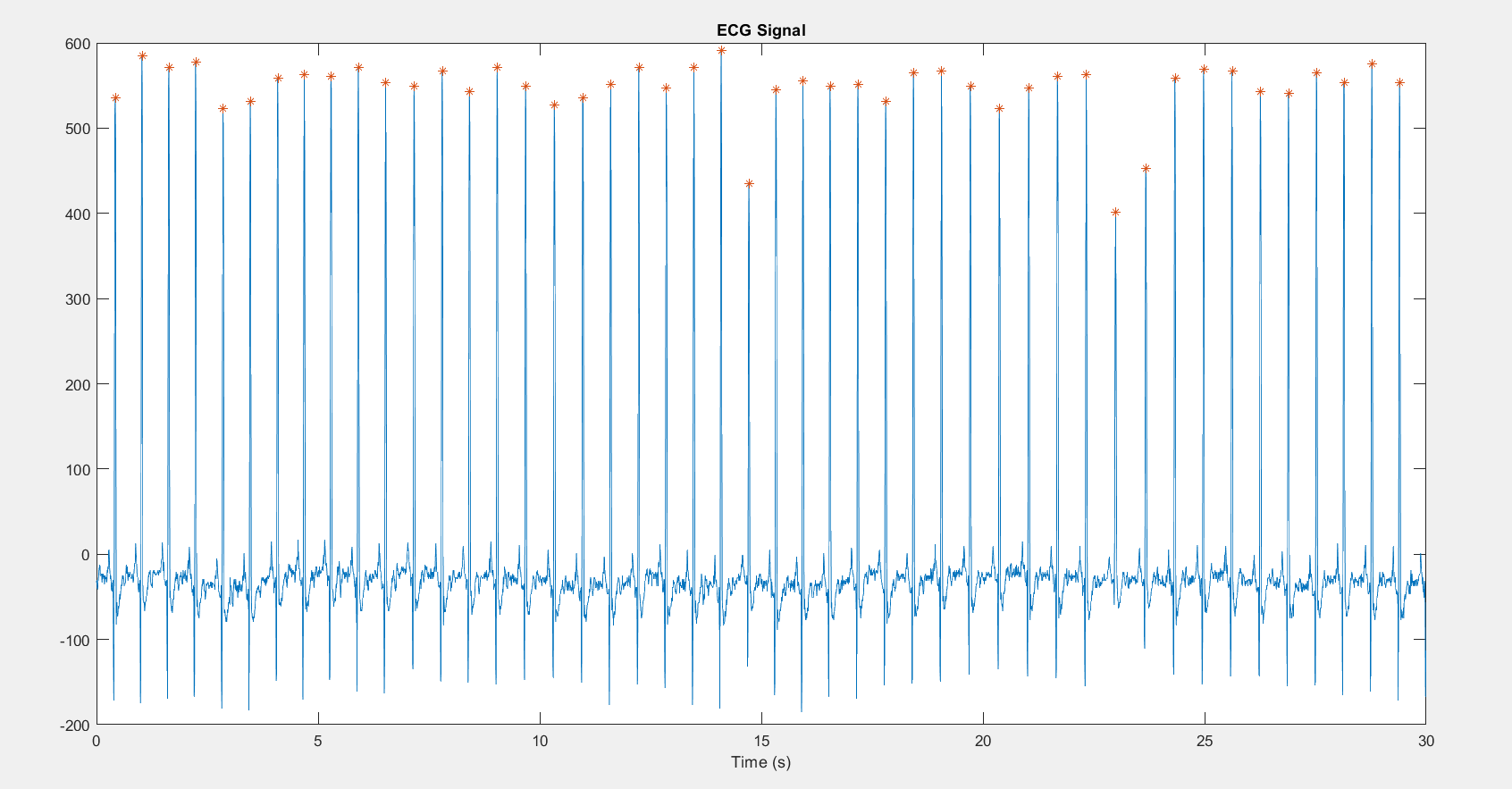
قسمت الف

برای این قسمت از دو دیتای 16265m.mat و 18184m.mat استفاده شده است.

ابتدا دیتای 16265m.mat را مورد بررسی قرار می‌دهیم .

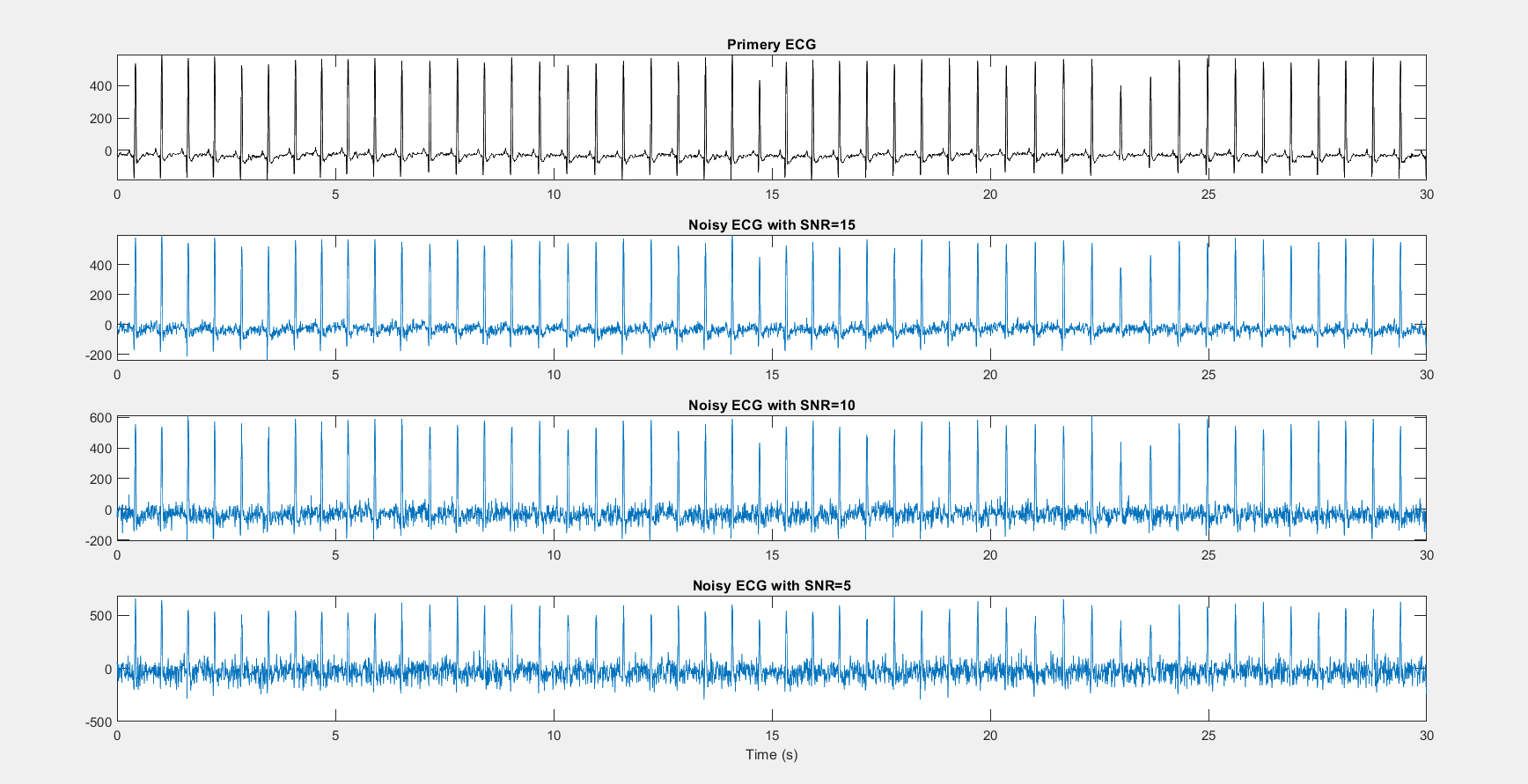
قسمت ب

با استفاده از تابع خودنوشته ی myFindPeaks3 نقاط پیک مشخص می‌شود.



قسمت پ

با استفاده از دستور awgn نویز با SNR مورد نظر اضافه می‌شود و سپس در متغیر ECG\_N به صورت ردیفی ذخیره می‌شود.



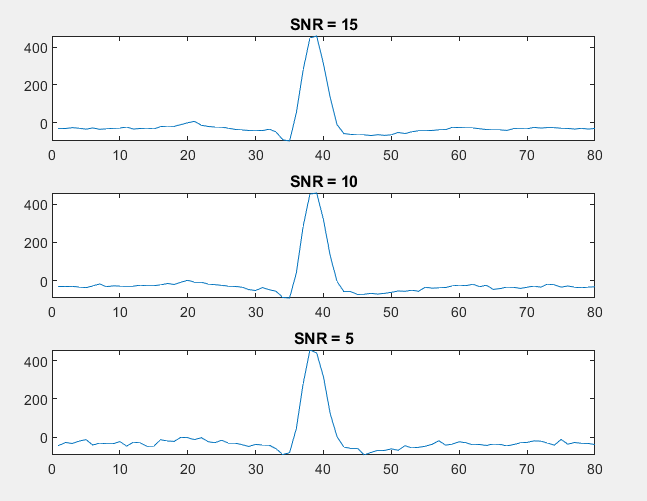
قسمت ت

ابتدا نقاط R را می‌یابیم و سپس زیر هم قرار می‌دهیم و با یکدیگر میانگین میگیریم. پس از این کار به مقدار نصف کمترین فاصله فواصل R را شیفت می‌دهیم تا در تقریبا وسط قرار بگیرد تا درست این عملیات انجام شود.

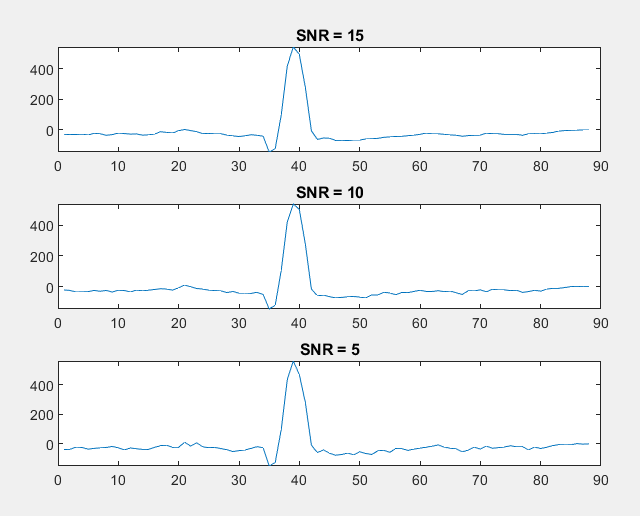
اما چالشی که وجود دارد این است که همه آن طول ها با هم برابر نیستند و این باعث مختل شدن عملیات جمع می‌شود. برای این کار دو راه حل امتحان شده اند.

راه حل اول: طول میانگین را برای خروجی امتحان میکنیم و با استفاده از دستور resample می‌توان طول مورد نظر را ایجاد کرد با این کار طول همه سیگنال ها یکی می‌شود این دستور در حقیقت یک sinc را با سیگنال conv می‌کند و برای همین نیز خروجی ممکن است smoothتر شود.

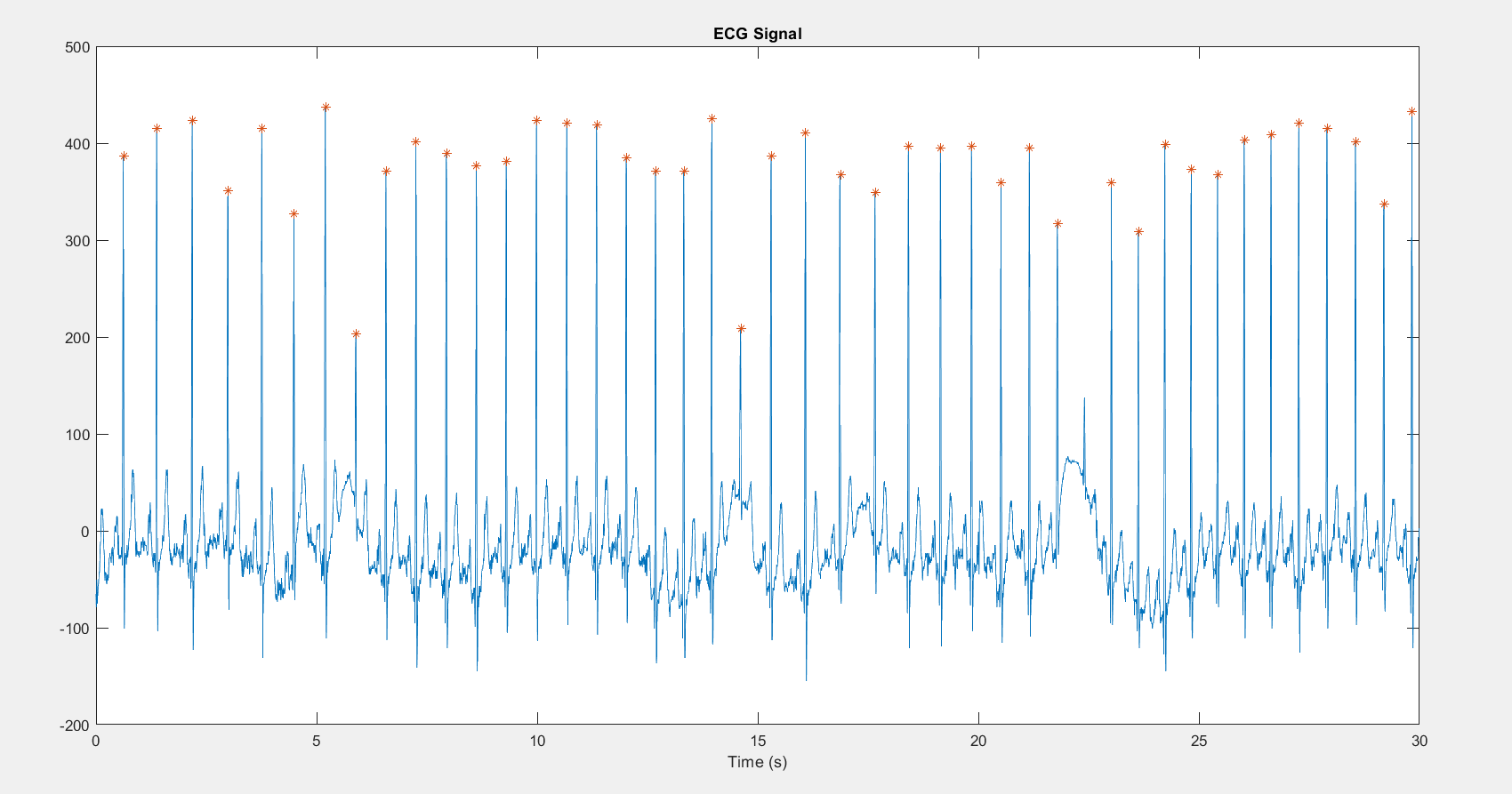
شکل زیر این روش را نشان می‌دهد.

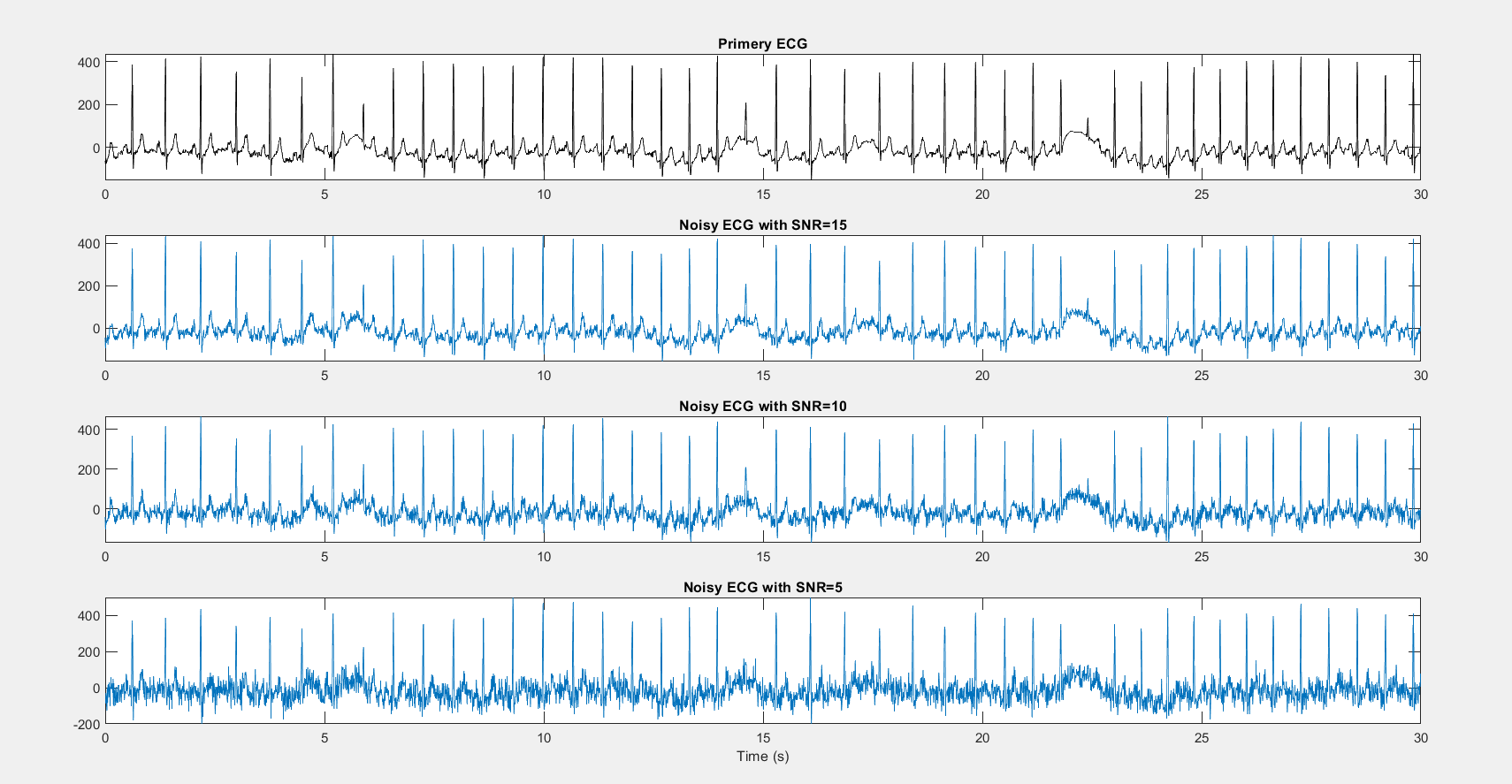


راه حل دوم: طول بیشینه را برای خروجی در نظر بگیریم و سپس باقی سیگنال ها را در انتهایشان صفر بگذاریم. و سپس میانگین بگیریم با این روش جواب زیر حاصل می‌شود.

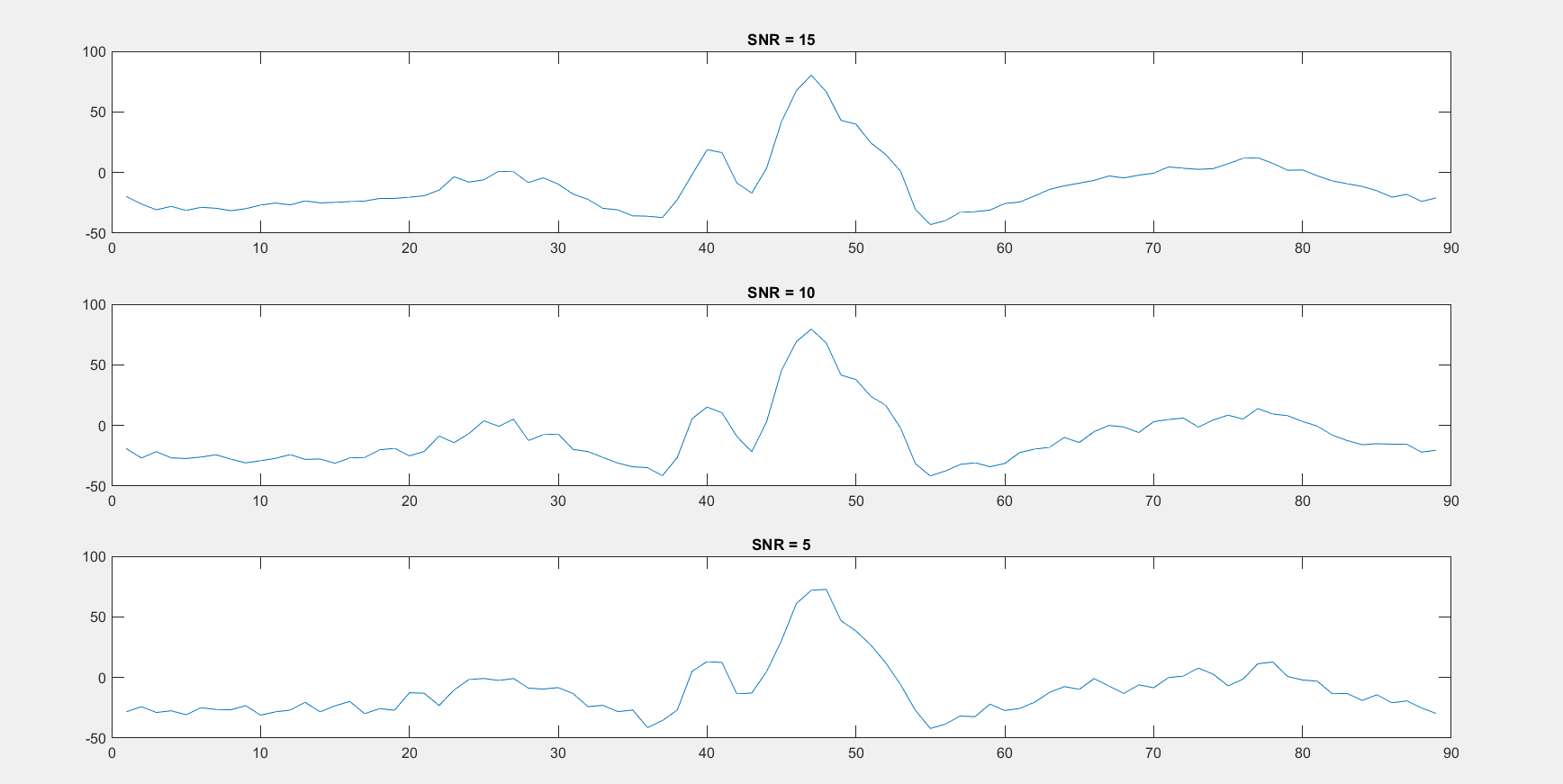


برخی قسمت های الگوی ECG در این سیگنال قابل تشخیص نیست بنابراین یک با دیگر قسمت های بالا را برای دیتای 18184m.mat تکرار می‌کنیم. خروجی قسمت های بالا در زیر قابل مشاهده است.

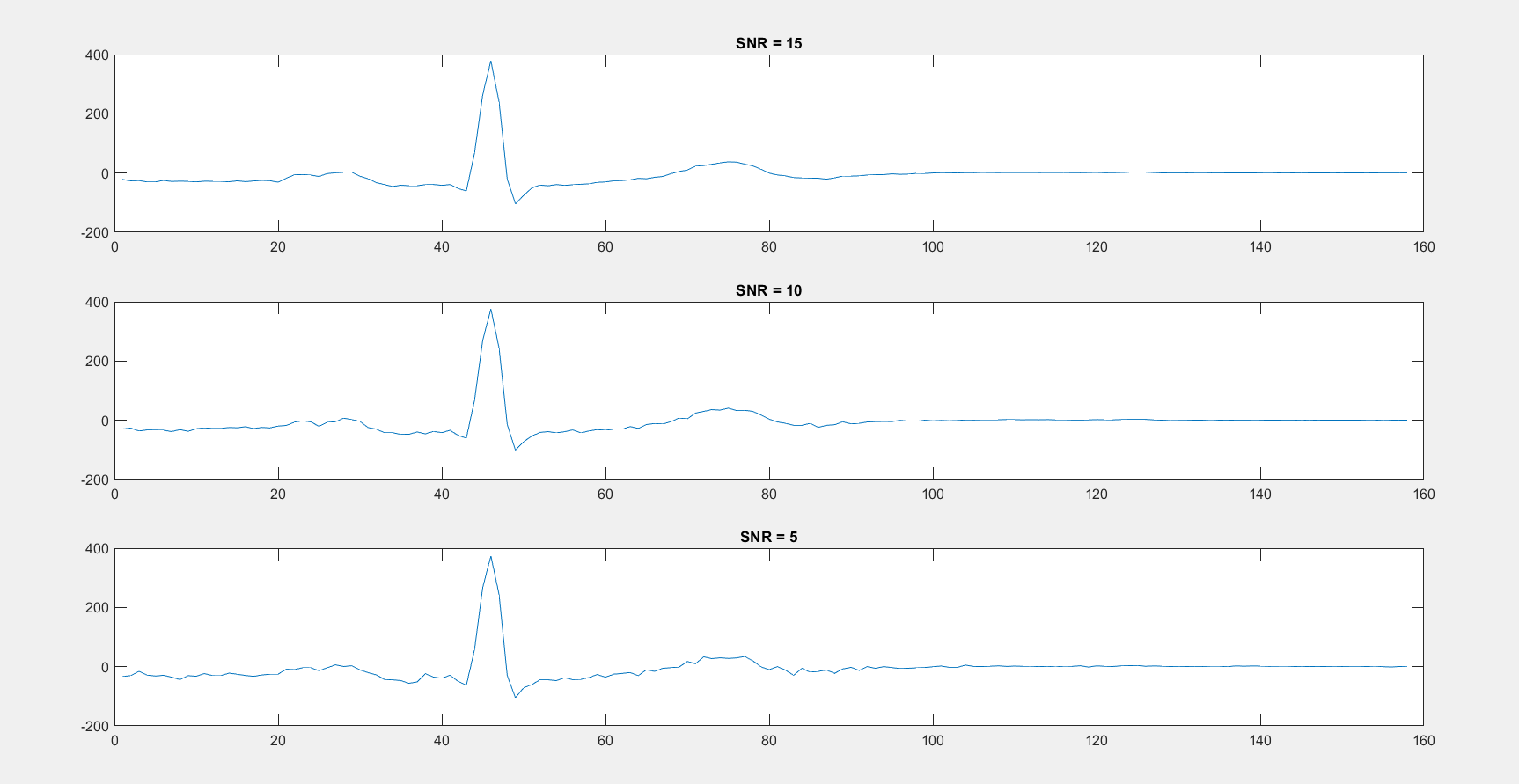




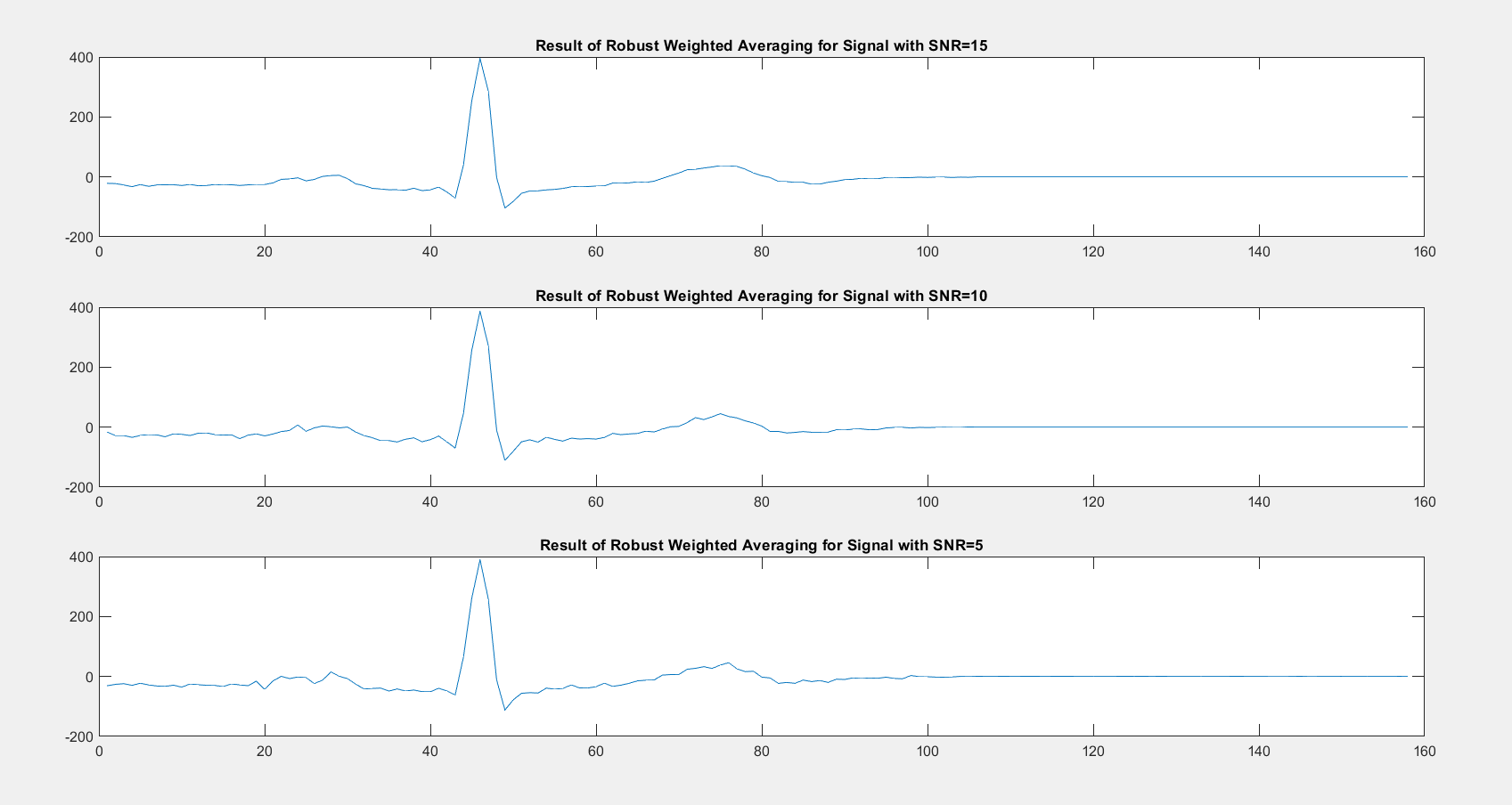
خروجی روش اول (با استفاده از دستور resample )



خروجی با استفاده از روش دوم ( صفر گذاری )

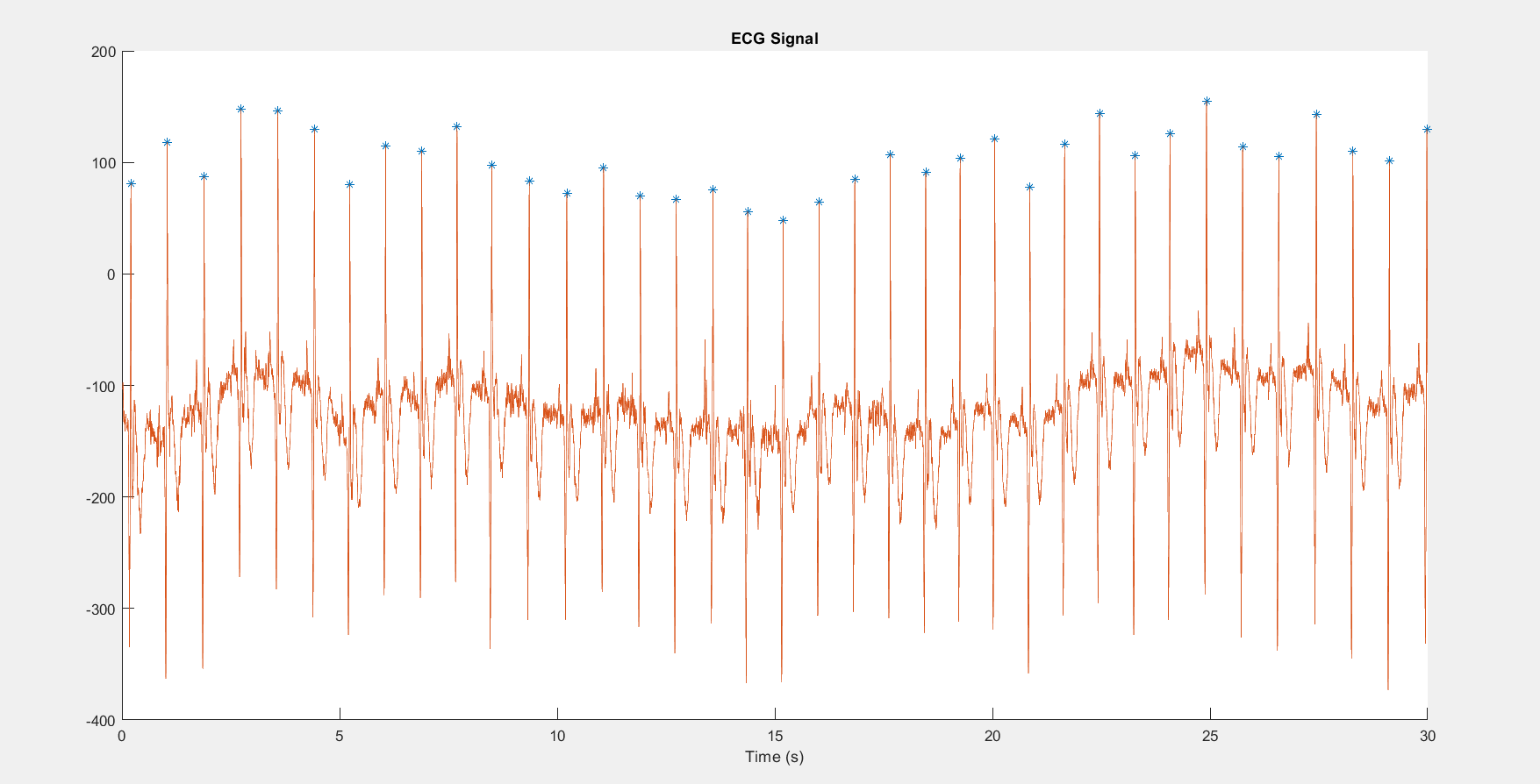


همچنین روش پیشنهادی مقاله نیز در زیر قابل مشاهده است.



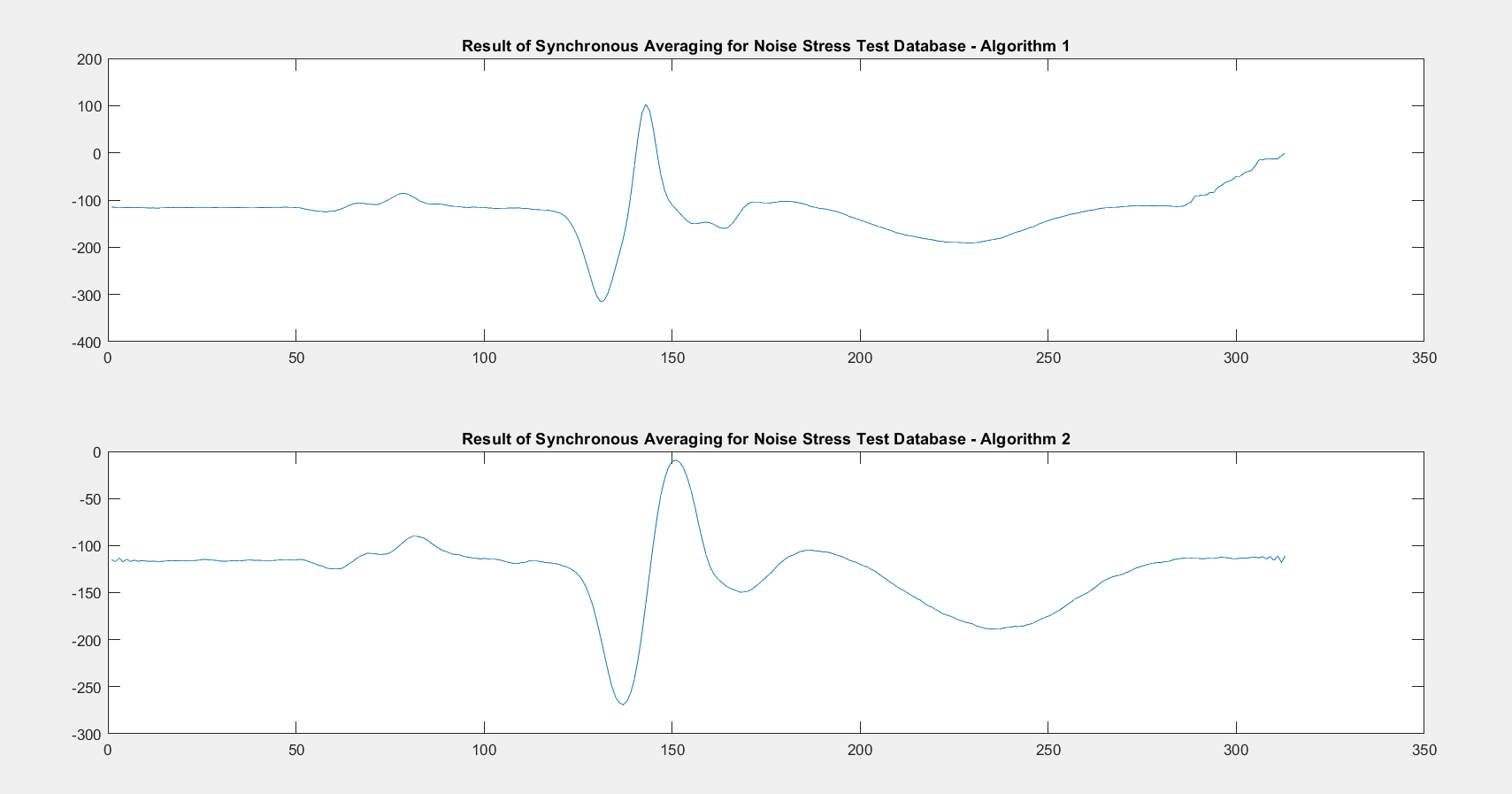
قسمت ت

داده های این قسمت نیز از لینک مذکور دانلود شده اند.



گرچه سیگنال نوسان دارد ولی myFindpeaks3 به خوبی بدون تغییر مقدایر آستانه پیک های R را مشخص می‌کند.

روش اول و دوم گفته شده در قسمت ت به صورت زیر خواهد بود.



همچنین روش ارایه شده در مقاله نیز در تصویر زیر آمده است.

