به نام خدا

دانشگاه صنعتی خواجه نصیرالدین طوسی

دانشکده برق

شناسایی سیستم

گزارش تمرین شماره یک

[علیرضا قاسمی]

[40110214]

دکتر مهدی علیاری شوره دلی

آبان ماه 1402

فهرست مطالب

عنوان شماره صفحه

بخش ۱: سوالات تحلیلی

سوال اول

۱. الف) در روش Regularization بایاس و دقت تخمین پارامتر θ را محاسبه کنید. در مقایسه با حالت unregularized تغییرات را چگونه توجیه می‌کنید؟

ب) (امتیازی) – انتظار دارید کدام یک از پارامترها بعد از اعمال Regularization به صفر میل کند؟

در روش LS اگر تعداد مشاهدات کم باشد یا وابستگی بین ورودی ها وجود داشته باشد و به طور کامل از یک دیگر مستقل نباشند ماتریس $\underline{X}^T\underline{X}$ مقادیر ویژه نزدیک به صفر پیدا میکند و $\left(\underline{X}^T\underline{X}\right)^{-1}$ مقادیر بسیار بزرگی پیدا میکند و مشکل Near-Singular بودن به وجود میاید. به همین خاطر از روش Regularization استفاده میکنیم.

برای حل این مشکل تابع هزینه اولیه را تغییر میدهیم و یک ترم جدید به ان اضافه میکنیم با این کار مقادیر تتا های تخمین زده شده در تابع هزینه اورده میشوند به همین دلیل مقادیر تتا ها نمیتوانند بیش از اندازه بزرگ شوند

$$I(\underline{\theta}, \lambda) = \frac{1}{2}\left(\underline{e}^T\underline{e} + \lambda\,\underline{\theta}^T\underline{\theta}\right) = \frac{1}{2}\left(\underline{e}^T\underline{e} + \lambda\sum_{i=1}^{n}\theta_i^2\right)$$

حال با استفاده از مشتق گرفتن از رابطه بالا مقدار تتا بدست میاید که برابر است با

$$\underline{\hat{\theta}} = \left(U^TU + \lambda\underline{I}\right)^{-1}U^Ty$$

برای محاسبه بایاس این تخمین مقدار امید تتا تخمین زده شده را محاسبه میکنیم

$$E\{\underline{\hat{\theta}}\} = E\left\{\left(U^TU + \lambda\underline{I}\right)^{-1}U^Ty\right\} = E\left\{\left(U^TU + \lambda\underline{I}\right)^{-1}U^T(U\theta + n)\right\}$$

$$E\{\underline{\hat{\theta}}\} = E\left(\left(U^TU + \lambda\underline{I}\right)^{-1}U^TU\theta\right) + E\left(\left(U^TU + \lambda\underline{I}\right)^{-1}U^Tn\right)$$

با توجه به اینکه U,θ مقادیر ثابتی هستند از داخل امید خارج میشوند

$$E\{\underline{\hat{\theta}}\} = \left(U^TU + \lambda\underline{I}\right)^{-1}U^TU\theta + \left(U^TU + \lambda\underline{I}\right)^{-1}U^TE(n)$$

با توجه به فرض اولیه مسئله نویز دارای میانگین صفر میباشد پس امید ان برابر با صفر است

$$E\{\underline{\hat{\theta}}\} = \left(U^TU + \lambda\underline{I}\right)^{-1}U^TU\theta + n$$

در بخشی از فرمول بالا مقدار $\lambda\underline{I}$ را اضافه و کم میکنیم

$$E\{\underline{\widehat{\theta}}\} = \left(U^TU + \lambda\underline{I}\right)^{-1}(U^TU + \lambda\underline{I} - \lambda\underline{I})\theta$$

$$E\{\underline{\widehat{\theta}}\} = (\left(U^TU + \lambda\underline{I}\right)^{-1}(U^TU + \lambda\underline{I}) - \left(U^TU + \lambda\underline{I}\right)^{-1}\lambda\underline{I})\theta$$

$$E\{\underline{\widehat{\theta}}\} = (I - \lambda\left(U^TU + \lambda\underline{I}\right)^{-1})\theta$$

$$E\{\underline{\widehat{\theta}}\} = (\theta - \lambda\left(U^TU + \lambda\underline{I}\right)^{-1}\theta)$$

$$E\{\underline{\widehat{\theta}}\} \neq \theta$$

همانگونه که مشاهده میشود دیگر امید تتا تخمین زده شده برابر با تتا واقعی نیست و یک عبارت ثابتی با تتا واقعی اختلاف دارد به همین خاطر این تخمین دارای بایاس میباشد

مقدار کوواریانس تتا تخمین زده شده به صورت زیر بدست میاید

$$cov(\theta) = E((\theta - E(\theta))(\theta - E(\theta))^T)$$

برای ساده سازی محاسبات بالا ابتدا مقدار $\theta - E(\theta)$ را حساب میکنیم

$$\theta - E(\theta) = \left(U^TU + \lambda\underline{I}\right)^{-1}U^Ty - (I - \lambda\left(U^TU + \lambda\underline{I}\right)^{-1})\theta$$

$$= \left(U^TU + \lambda\underline{I}\right)^{-1}U^T(U\theta + n) - (I - \lambda\left(U^TU + \lambda\underline{I}\right)^{-1})\theta$$

$$= \left(U^TU + \lambda\underline{I}\right)^{-1}U^TU\theta + \left(U^TU + \lambda\underline{I}\right)^{-1}U^Tn - (I - \lambda\left(U^TU + \lambda\underline{I}\right)^{-1})\theta$$

$$= (\left(U^TU + \lambda\underline{I}\right)^{-1}U^TU - (I - \lambda\left(U^TU + \lambda\underline{I}\right)^{-1}))\theta + \left(U^TU + \lambda\underline{I}\right)^{-1}U^Tn$$

$$= (\left(U^TU + \lambda\underline{I}\right)^{-1}(U^TU + \lambda\underline{I} - \lambda\underline{I}) - (I - \lambda\left(U^TU + \lambda\underline{I}\right)^{-1}))\theta + \left(U^TU + \lambda\underline{I}\right)^{-1}U^Tn$$

$$= (I - \lambda\underline{I}(U^TU + \lambda\underline{I}) - (I - \lambda\left(U^TU + \lambda\underline{I}\right)^{-1}))\theta + \left(U^TU + \lambda\underline{I}\right)^{-1}U^Tn$$

$$= 0 + \left(U^TU + \lambda\underline{I}\right)^{-1}U^Tn$$

$$\theta - E(\theta) = \left(U^TU + \lambda\underline{I}\right)^{-1}U^Tn$$

$$cov(\theta) = E((\theta - E(\theta))(\theta - E(\theta))^T)$$

$$cov(\theta) = E(\left(U^TU + \lambda\underline{I}\right)^{-1}U^Tn)(\left(U^TU + \lambda\underline{I}\right)^{-1}U^Tn)^T)$$

$$cov(\theta) = E(\left(U^TU + \lambda\underline{I}\right)^{-1}U^Tn)(n^TU\left(U^TU + \lambda\underline{I}\right)^{-1}))$$

با توجه به اینکه U,θ مقادیر ثابتی هستند از داخل امید خارج میشوند

$$cov(\theta) = \left(U^TU + \lambda\underline{I}\right)^{-1}U^TE(nn^T)\,U\left(U^TU + \lambda\underline{I}\right)^{-1}$$

کوواریانس نویز را برابر با σ^2 در نظر میگیریم

$$cov(\theta) = \sigma^2\left(U^TU + \lambda \underline{I}\right)^{-1} U^TU\left(U^TU + \lambda \underline{I}\right)^{-1}$$

قسمت ب

در اینجا میتوانیم با تنظیم مقدار λ مقدار بایاس و دقت روش تخمین را تغییر دهیم اما بین مقدار بایاس و دقت یک trade off وجود دارد هر چه مقدارλ به سمت صفر میل کند تخمین بدست امده شبیه به تخمین LS میشود و بایاس به سمت صفر میل میکند ولی دقت خروجی بدتر میشود و مقادیر تتا های تخمین زده شده تغییرات بیشتری دارد

در مقابل هر چه مقدار λ بزرگ تر شود دقت خروجی بهتر میشود و مقدار $cov(\theta)$ به سمت صفر میل میکند ولی مقدار بایاس تخمین بدست امده مخالف صفر میشود و مقدار امید تتا تخمین زده شده مخالف با تتا های واقعی میباشد

سوال دوم

ماتریس هسیان برابر با مشتق دوم تابع هزینه نسبت به تتا های تخمین زده میباشد

تابع هزینه در روش LS به صورت زیر تعریف میشود

$$I(\underline{\theta}) = \frac{1}{2}\underline{\theta}^T \underline{H}\,\underline{\theta} + \underline{h}^T\underline{\theta} + \frac{1}{2}h_0$$

$$\underline{H} = \underline{X}^T\underline{X},$$

$$\underline{h} = -\underline{X}^T\underline{y},$$

$$h_0 = \underline{y}^T\underline{y}.$$

در نتیجه ماتریس هسیان برابر است با

$$\underline{H} = \frac{\partial^2 I(\underline{\theta})}{\partial \underline{\theta}^2} = \underline{X}^T \underline{X}.$$

در روش LS دقت و یا کوواریانس تخمین بدست امده برابر است با

$$\text{cov}\{\underline{\hat{\theta}}\} = \sigma^2 \left(\underline{X}^T \underline{X}\right)^{-1} = \sigma^2 \underline{H}^{-1}$$

اگر مقادیر روی قطر اصلی ماتریس هسیان کوچک شوند و یا به عبارت دیگه ماتریس هسیان نزدیک به ویژه شود معکوس ان بسیار بزرگ شده و مشکل Near-Singular بودن به وجود میاید. در این مسئله یکی از مقادیر روی قطر اصلی ماتریس هسیان برابر با صفر میشود پس ماتریس هسیان ویژه میباشد و معکوس پذیر نیست و با خطا Non-Singular بودن مواجه میشویم و برای درایه دوم ماتریس معکوس مقدار بی نهایت پیدا کرده و در نتیجه تتا متناظر با ان یکتا نیست و مقادیر ان به سمت بی نهایت میل میکند

ب

در این مسئله با توجه به ماتریس هسیان داده شده داریم.

$$cov\{\underline{\hat{\theta}}\} = \sigma^2 (X^T X)^{-1} = \sigma^2 (H)^{-1} = \sigma^2 \begin{bmatrix} \frac{1}{80} & 0 & 0 & 0 \\ 0 & \infty & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & \frac{1}{6} \end{bmatrix}$$

هر چه مقادیر روی قطر اصلی ماتریس معکوس هسیان بزرگتر شود مقدار کوواریانس تتا های تخمین زده شده بزرگ تر میشود دقت تخمین بدتر میشود و به طور بلعکس هر چه مقادیر روی قطر اصلی کوچک تر شود دقت تخمین بهتر میشود با توجه به مقدار ماتریس کوواریانس بالا به ترتیب تتا های اول،چهارم،سوم و دوم دقت بالایی دارند

ج

مقدار حساسیت تخمین بدست امده نسبت به هر پارامتر متناسب با مقادیر روی قطر ماتریس هسیان است هر چه این مقادیر کوچک تر باشد حساسیت کمتر است و تغییرات این متغیر تاثیر کمتری بر روی تغییرات خطا خواهد داشت و به طور بلعکس

با توجه به ماتریس هسیان داده شده در این مسئله پارامتر دوم در تغییرات خطا خروجی نقشی ندارد و بعد از ان پارامتر سوم حساسیت کمتری دارد

سوال 3

اگر در تخمین پارامتر های یک سیستم پارامتر های انتخاب شده از یک دیگر مستقل خطی نباشند و به یک دیگر به طور کامل وابسته باشند در این صورت یکی از پارامتر های انتخاب شده اضافی میباشد و این پارامتر اضافی انتخاب شده به پیشبینی نویز میپردازد و تخمین بدست امده دارای خطا میباشد و همچنین این مسئله میتواند باعث Near-Singular شدن مسئله شود.

برای حل این مشکل میتوان از روش های زیر استفاده کرد

یک استفاده از روش های انتخاب بهینه رگرسور ها به طور مثال با استفاده از روش های متعامد سازی رگرسور ها Orthogonal Regressors ، روش های F.S , B.E و یا روش های Ridge Regression در ابتدا به کاهش تعداد رگرسور ها بپردازیم و در ادامه با تعداد رگرسور کمتر مسئله LS را حل کنیم

راه حل دوم استفاده از روش لاگرانژ است در این روش با اضافه کردن پارامتر های لاگرانژ به مسئله میتوانیم قید داده شده در مسئله را حذف کنیم و مسئله را به یک مسئله بدون قید با یک متغیر بیشتر تبدیل کنیم.

سوال چهار

$$\ddot{x} - 3\dot{x} + 2x = 0$$

برای حل این معادله از ان تبدیل لاپلاس میگیریم و جمله عمومی معادله دیفرانسیل را بدست میاوریم

$$s^2 - 3s + 2 = 0 \quad s_1 = 1, s_2 = 2$$

$$x(t) = c_1 e^t + c_2 e^{2t}$$

$$x(t) = c_1 e^t + c_2 e^{2t} \xrightarrow{t=0} x(0) = c_1 + c_2$$

$$\dot{x}(t) = c_1 e^t + 2c_2 e^{2t} \xrightarrow{t=0} \dot{x}(0) = c_1 + 2c_2$$

$$c_1 = 2x(0) - \dot{x}(t)$$
$$c_2 = \dot{x}(t) - x(0)$$
$$x(t) = (2x(0) - \dot{x}(t))e^t + (\dot{x}(t) - x(0))e^{2t}$$

حال میتوان با در نظر گرفتن تبدیل های زیر معادله بالا را همانند یک معادله LS در نظر گرفت.

$$\theta_1 = (2x(0) - \dot{x}(t))$$
$$\theta_2 = (\dot{x}(t) - x(0))$$
$$u_1 = e^t$$
$$u_2 = e^{2t}$$

در این معادله مقدار $x(t)$ را به عنوان فرض مسئله داریم پس میتوانیم به راحتی مقدار تتا های فرض شده را بدست بیاوریم و در انتها با استفاده از تتا های بدست امده و حل دو معادله دو مجهول مقادیر خواسته شده مسئله را بدست اوریم

بخش۲: سوالات شبیه سازی

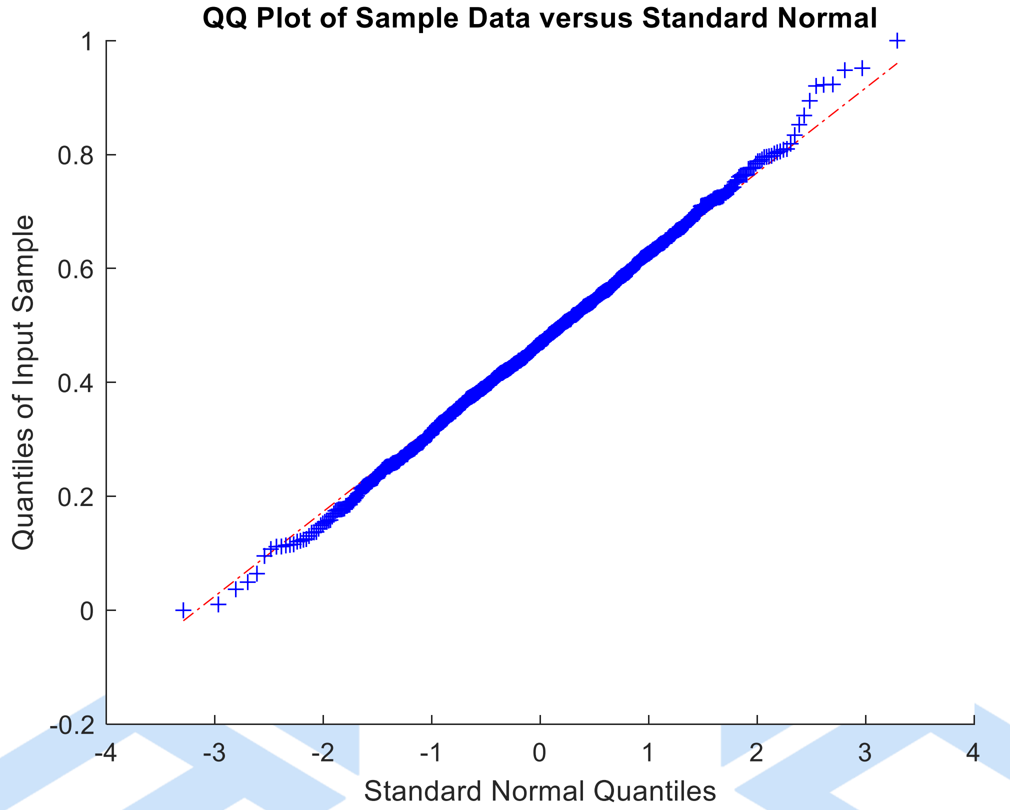
سوال اول

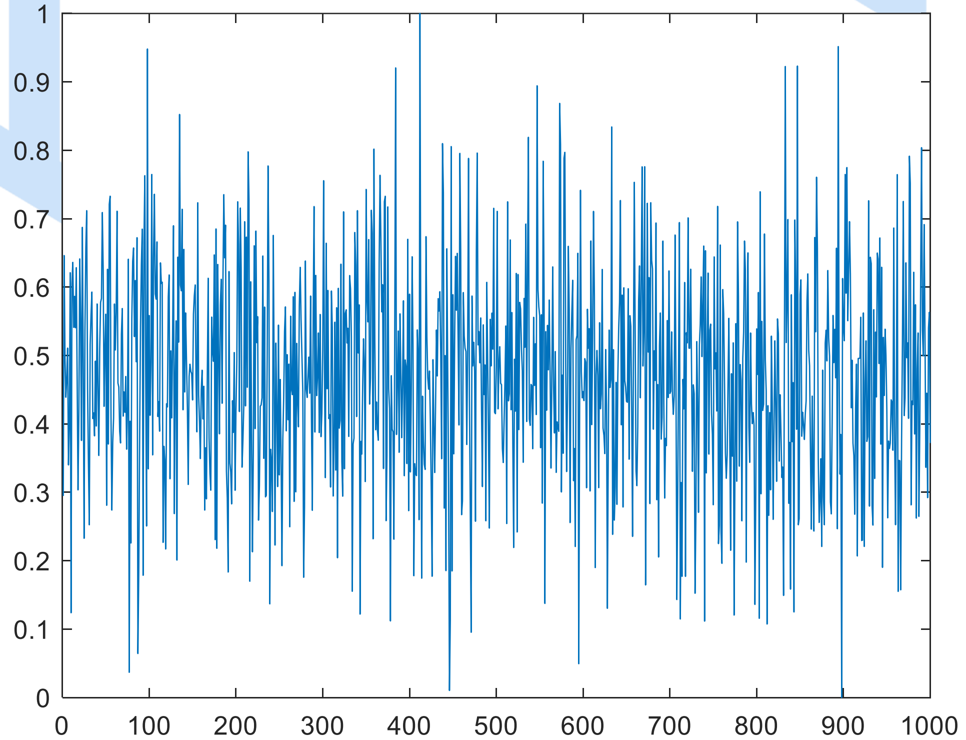
در ابتدا هزار داده با استفاده از دستور randn تولید میکنیم سپس داده های تولید شده را در بازه صفر تا یک نرمالیزه میکنیم.

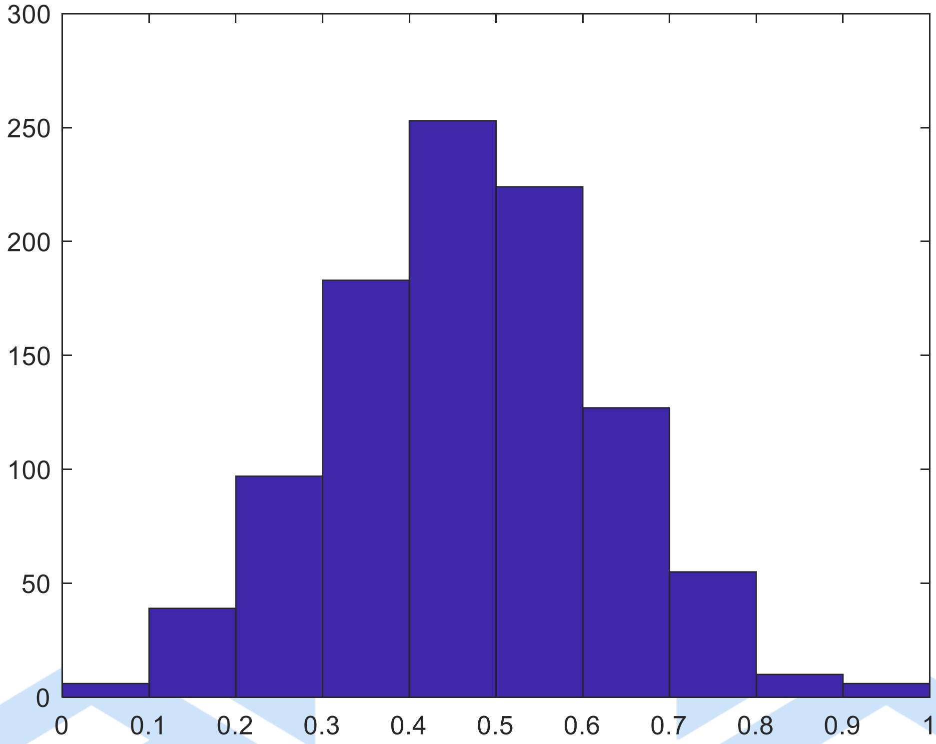
$$data = \frac{D - D_{min}}{D_{max} - D_{min}}$$

توزیع داده های ساخته شده به صورت زیر میباشد

همان گونه که نمودار زیر نشان میدهد داده های تصادفی تولید شده نزدیک به خط قرمز میباشند در نتیجه توزیع داده ها شبیه به توزیع نرمال است. با استفاده از نمودار های histogram و اندازه خود داده ها نیز میتوانیم این موضوع را متوجه شویم.

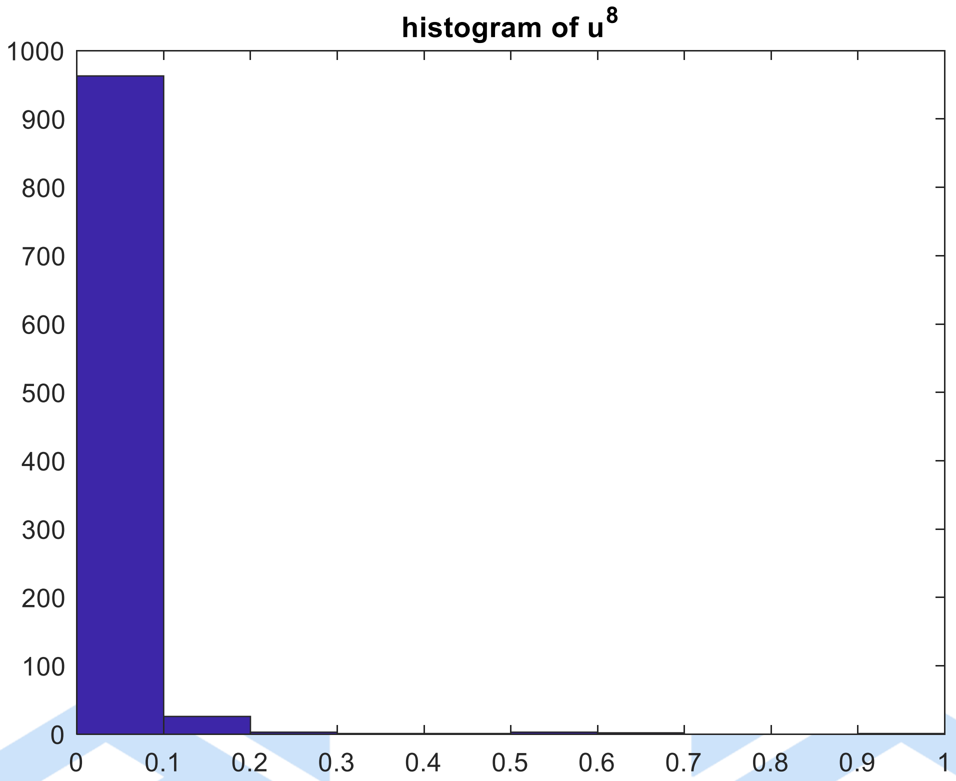

QQ Plot of Sample Data versus Standard Normal
Quantiles of Input Sample
Standard Normal Quantiles
1
0.8
0.6
0.4
0.2
0
-0.2
-4
-3
-2
-1
0
1
2
3
4


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
100
200
300
400
500
600
700
800
900
1000

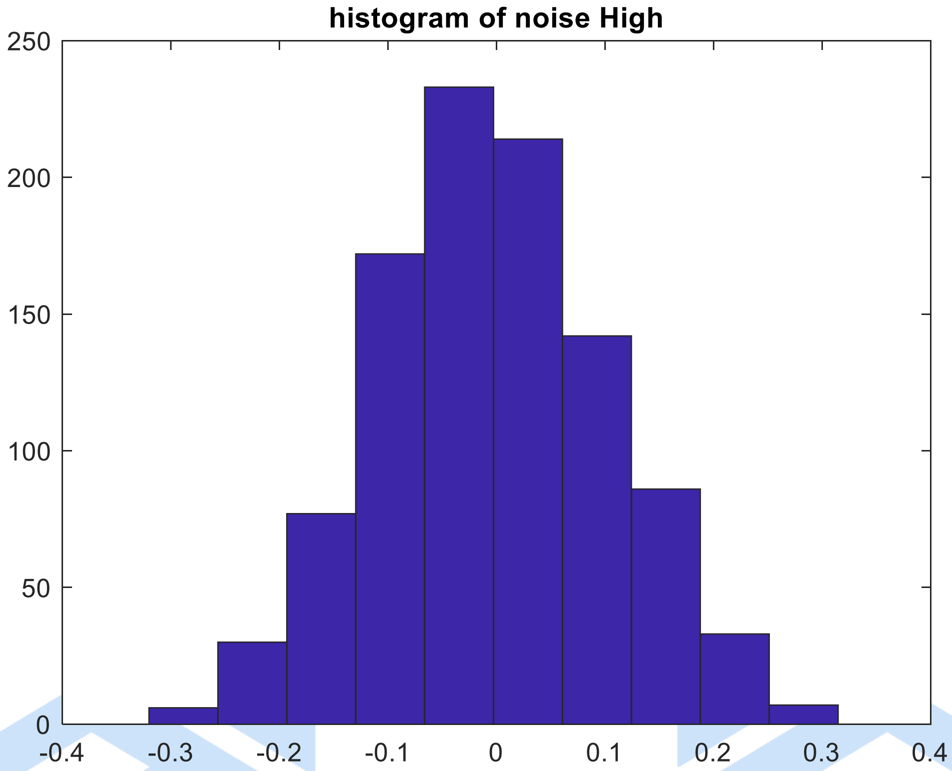


برای اینکه در همه مسئله ها از یک سری داده استفاده کنیم در ابتدا برای تولید داده ها با استفاده از دستور rng(40) از یک seed داده های رندوم را تولید میکنیم همچنین داده های تولید شده را در فایل data.mat ذخیره میکنیم و در بقیه سوالات از این دیتا ها استفاده میکنیم

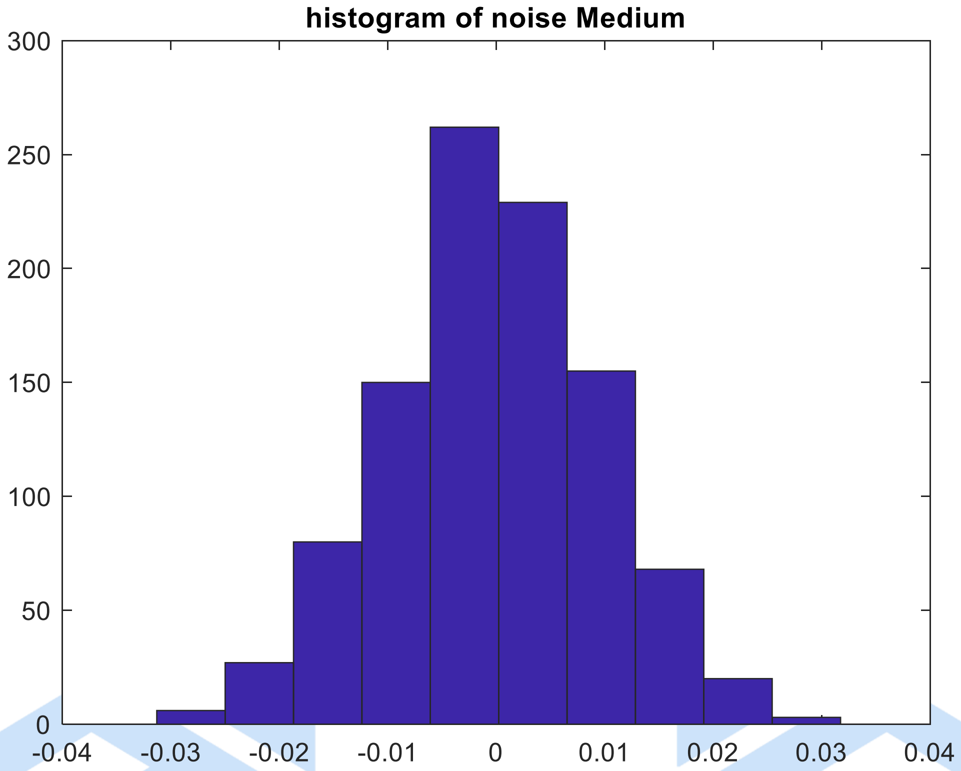
در بازه بین صفر تا یک هر چه عدد توان بالاتری داشته باشد مقدار ان کوچک تر میشود به همین خاطر رگرسور های داده شده در این سیستم مقادیر خیلی کمی دارند به همین خاطر نویز میتواند به راحتی بر روی این پارامتر ها تاثیر بگذارد و انها را خراب کند در شکل زیر هیستوگرام رگرسور اخر رسم شده است همانگونه که مشاهده میشود بیشتر رنج مقادیر این رگرسور در بازه [0,0.1] قرار دارد به همین خاطر اگر مقدار نویز در این بازه قرار بگیرد میتوان تاثیر این رگرسور را در فرایند شناسایی از بین ببرد و نویز جای این رگرسور را بگیرد



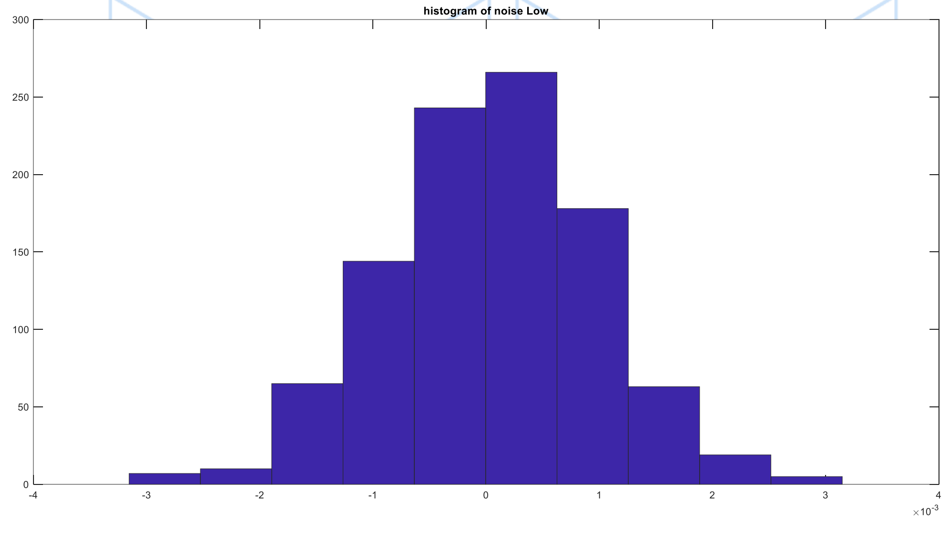
با انتخاب مقدار انحراف معیار برابر با 0.1 برای نویز مقدار نویز در بیشتر مواقع هم اندازه مقدار رگرسور است و میتوان این رگرسور را خراب کند. این واریانس را به عنوان واریانس زیاد انتخاب میکنیم در شکل زیر توزیع نویز تولید شده با این واریانس را مشاهده می شود



با انتخاب مقدار انحراف معیار برابر با 0.01 برای نویز متوسط مقدار نویز در بیشتر مواقع 0.1 مقدار رگرسور است و میتواند بر روی این رگرسور تاثیر کمی بگذارد . این واریانس را به عنوان واریانس متوسط انتخاب میکنیم در شکل زیر توزیع نویز تولید شده با این واریانس را مشاهده می شود



با انتخاب مقدار انحراف معیار برابر با 0.001 برای نویز متوسط مقدار نویز در بیشتر مواقع 0.01 مقدار رگرسور است و بر روی این رگرسور تاثیر بسیار کمی میگزارد . این واریانس را به عنوان واریانس کم انتخاب میکنیم در شکل زیر توزیع نویز تولید شده با این واریانس را مشاهده می شود



البته انتخاب این سه نوع واریانس به صورت دقیق صورت نگرفته است و اگر مقادیر واریانس برای نویز بیشتر از مقدار حداکثر انتخاب شده باشد میتواند تاثیر بسیار بیشتری بر روی رگرسور ها گذاشته و خروجی روش شناسایی را تخریب کند و هر چر واریانس انتخاب شده برای از مقادیر پیشنهاد شده برای واریانس های کم و متوسط کمتر باشد نتایج خروجی تخمین بهتر میشود.

Regularization

در محاسبات روش LS که در شکل زیر فرمول ان اورده شده است نیاز به محاسبه معکوس یک ماتریس داریم. در این جا با توجه به شرایط مسئله (کم بودن تعداد داده ها در هر دسته) مقادیر ویژه این ماتریس نزدیک به صفر میشوند در نتیجه مقدار معکوس ان بسیار بزرگ میشود

$$\underline{\hat{\theta}} = \left(\underline{X}^T \underline{X}\right)^{-1} \underline{X}^T \underline{y} .$$

بدون اضافه کردن ترم Regularization در متلب اخطار زیر را میگیریم

```
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate.
RCOND =  1.656114e-16.
```

با اضافه کردن یک مقدار بسیار کوچک به اندازه 0.000001 به مسئله در کلیت مسئله مشکلی ایجاد نمیشود اما این خطا برطرف میشود و همچنین مقادیر تتا تخمین زده شده دیگر مقادیر بسیار بزرگ نمیشوند و کوواریانس تتا ها بهبود پیدا میکند اما مقدار میانگین تتا ها اندکی خراب میشوند ولی با توجه به اینکه این مقدار بسیار کم است میتوان از تاثیر ان بر روی میانگین صرف نظر کرد

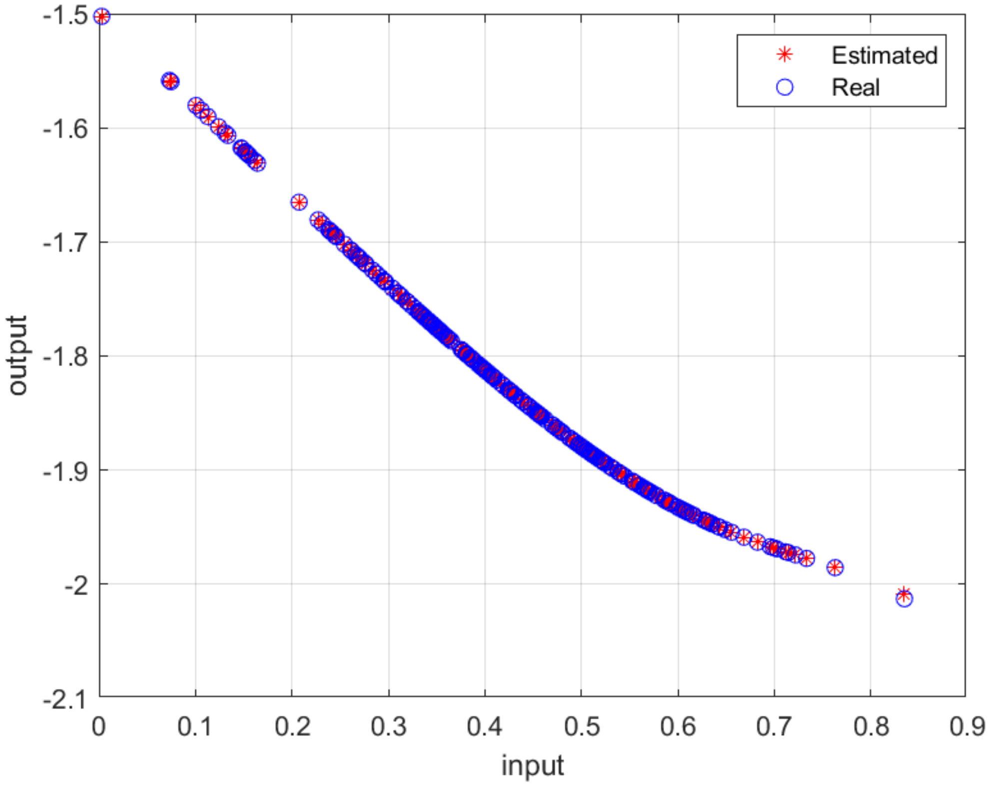
الف

حالت داده های بدون نویز

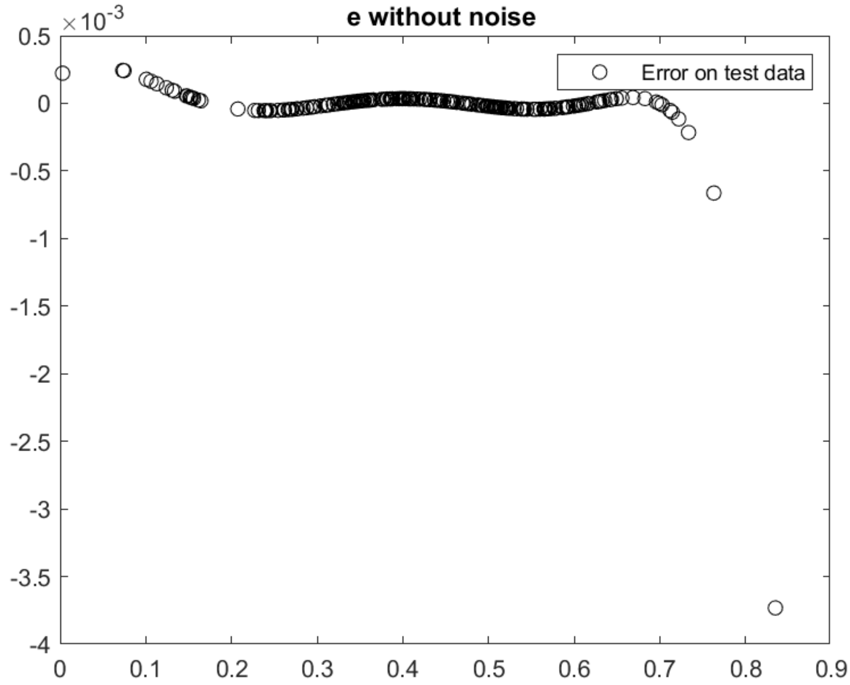
برای حل این سوال داده ها را به ده دسته صد تایی تقسیم میکنیم و از هر دسته 75 داده به عنوان داده های اموزش انتخاب میکنیم. و 25 داده باقی مانده را به دسته تست انتقال میدهیم. برای شناسایی ابتدا 10 بار با استفاده از 75 داده به تخمین پارامتر های میپردازیم و در انتها مقادیر میانگین تخمین های بدست امده را برای سنجش داده های تست استفاده میکنیم

همان گونه که مشاهده میشود خروجی مقدار تخمین زده شده و مقدار واقعی بر روی یک دیگر قرار گرفته اند و این نشان دهنده این است که تخمین LS بدون نویز به درستی عمل میکند

معیار درست بودن تخمین های بدست امده نزدیک تتا های تخمین زده شده به مقدار تتا های واقعی میباشد که در تخمین های با نویز کم این مقدار اختلاف کمتر است و در تخمین های با نویز زیاد این اختلاف بیشتر میشود



مقدار خطا این حالت در شکل زیر نشان داده شده است که بسیار کم میباشد.

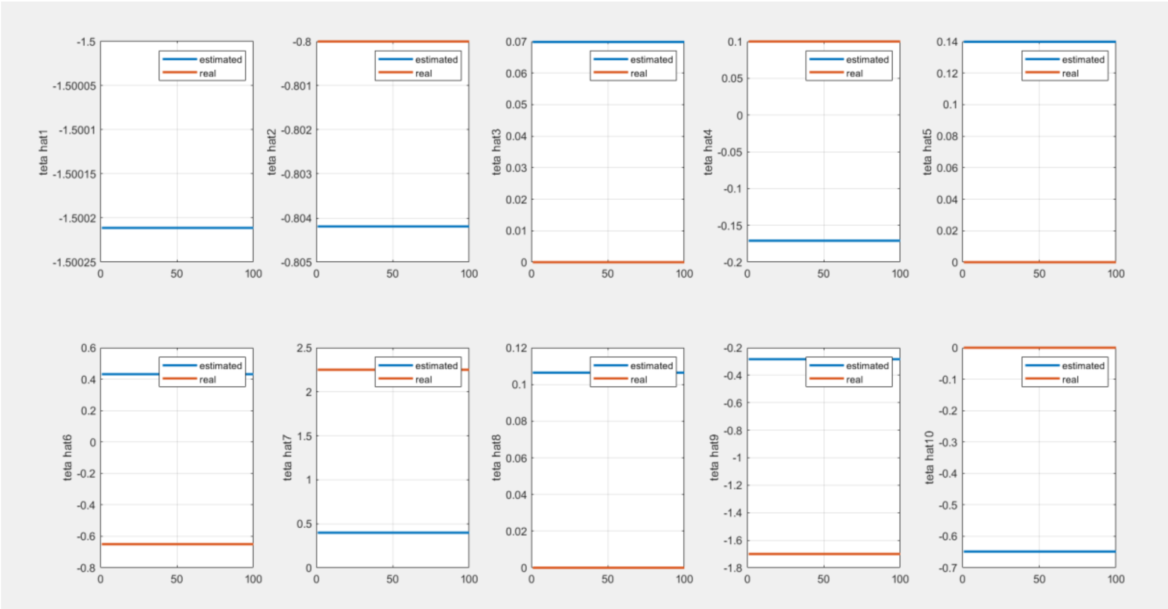


برای مقایسه روش های مختلف از معیار MSE بر روی خطا ها استفاده میکنیم

در این حالت مقدار MSE خطا برابر با

```
Mean squared normalized error  =

    5.9699e-08
```

و مقادیر تتا های تخمین زده شده با مقدار ناچیزی خطا برابر با مقادیر تتا های واقعی میباشند. این مقدار خطا به خاطر استفاده از روش Regularization میباشد

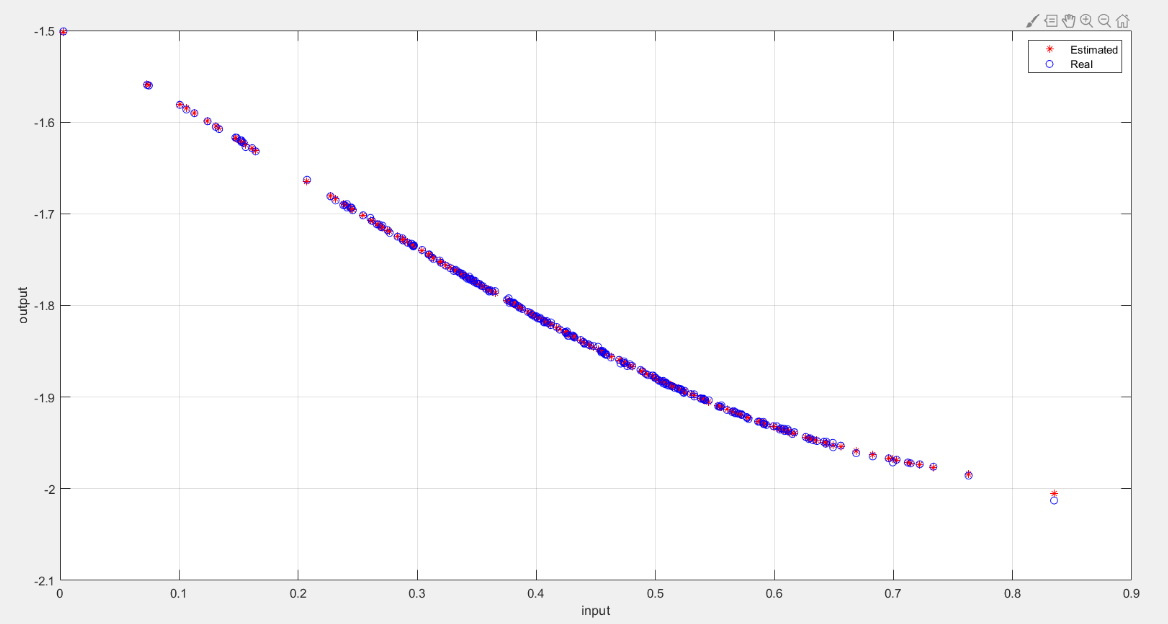


مقادیر تتا تخمین زده شده به صورت زیر میباشد

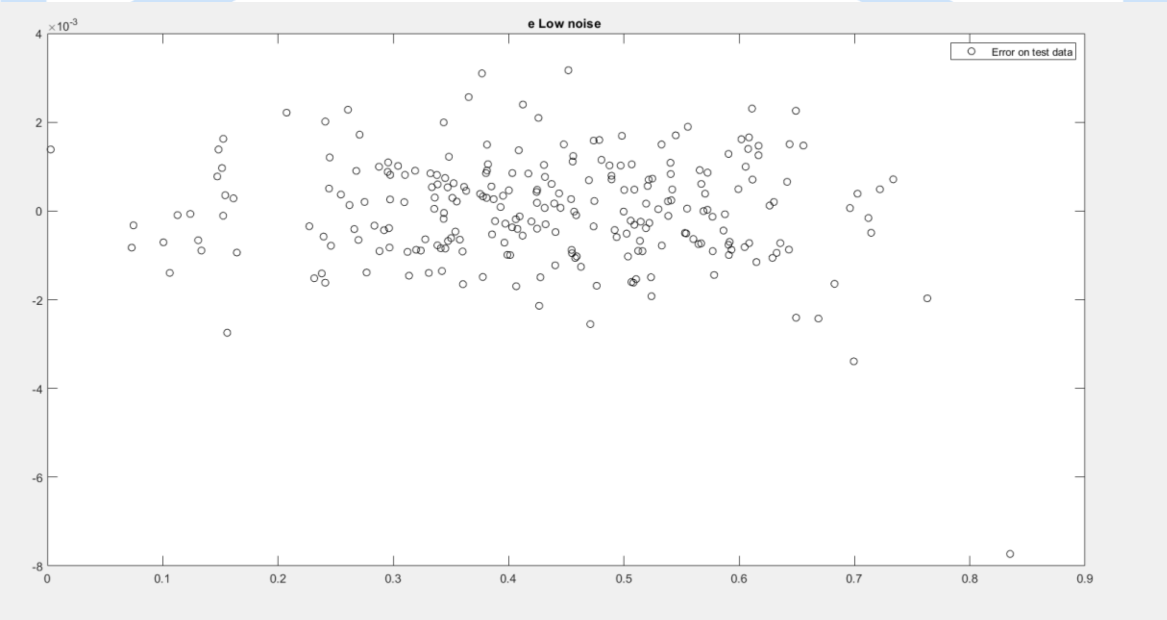
-1.5002
-0.8042
0.0699
-0.1707
0.1398
0.4316
0.3985
0.1064
-0.2834
-0.6485

حالت داده های با نویز کم

در شکل زیر خروجی مقدار تخمین زده شده و مقدار واقعی در بیشتر مواقع بر روی یک دیگر قرار گرفته اند و این نشان دهنده این است که تخمین LS با نویز بسیار کم نیز به درستی کار میکند. و تنها برای انتهای بازه مقداری خطا از خود نشان میدهد که این نیز به دلیل کمبود داده در ان بازه منطقی است



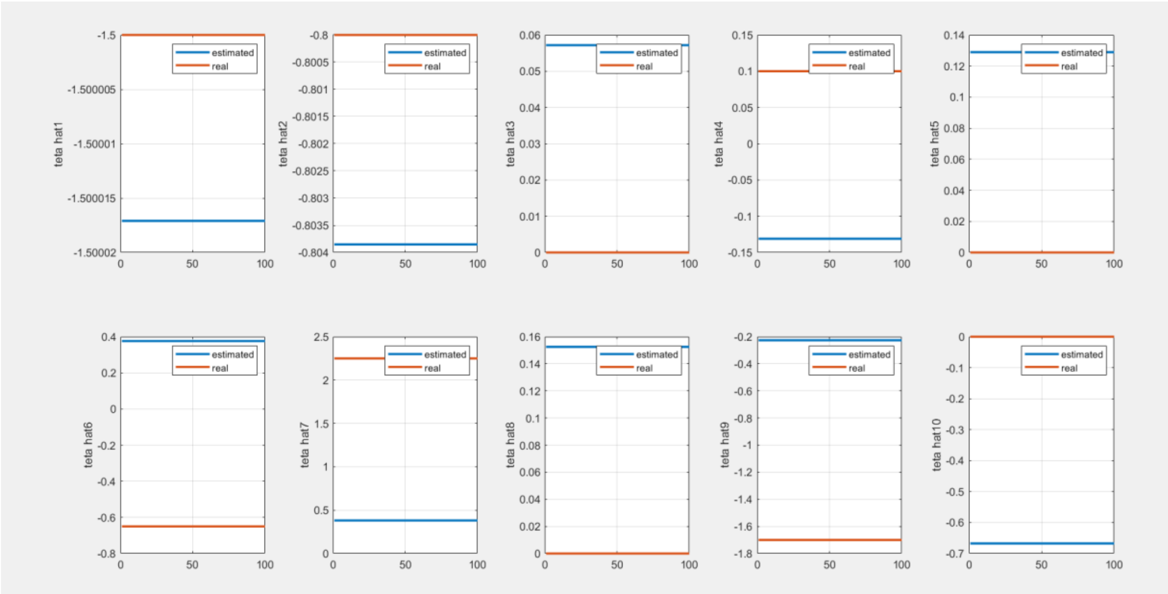
مقدار خطا این حالت در شکل زیر نشان داده شده است که نسبت به حالت قبلی بیشتر میباشد اما هنوز هم مقدار کمی است.



مقادیر mse خطا در این حالت برابر است با که نسبت به حالت قبلی 100 برابر بیشتر شده است

```
Mean squared normalized error  =

    1.4091e-06
```

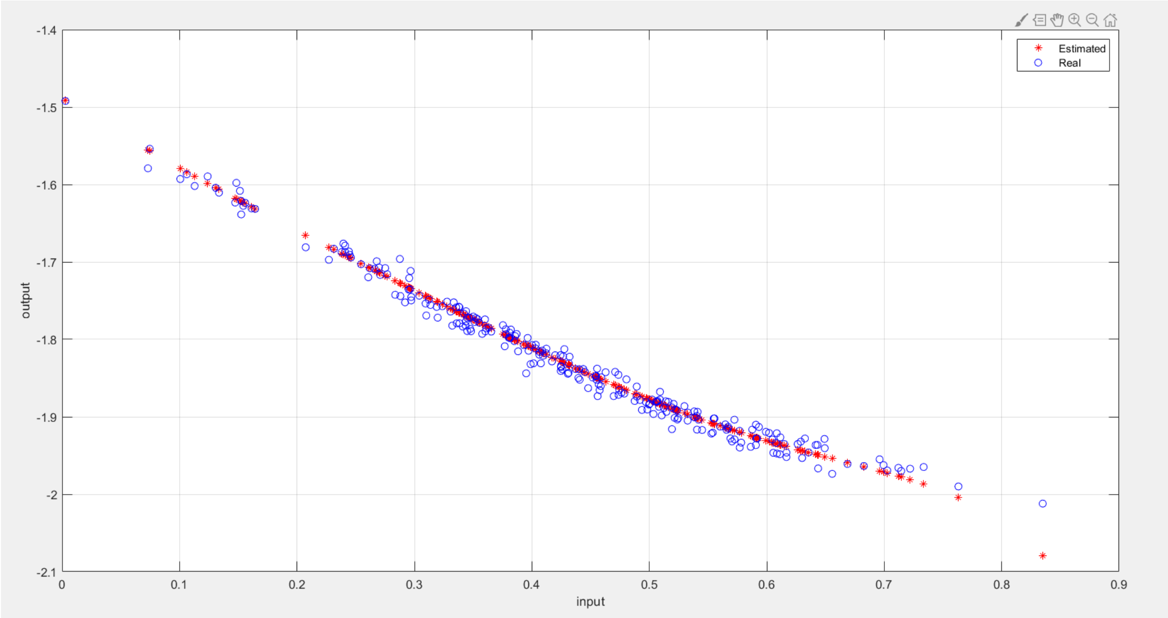
مانند حالت قبلی مقادیر تتا های تخمین زده شده با مقادیر تتا های واقعی دارای اختلاف کمی هستند



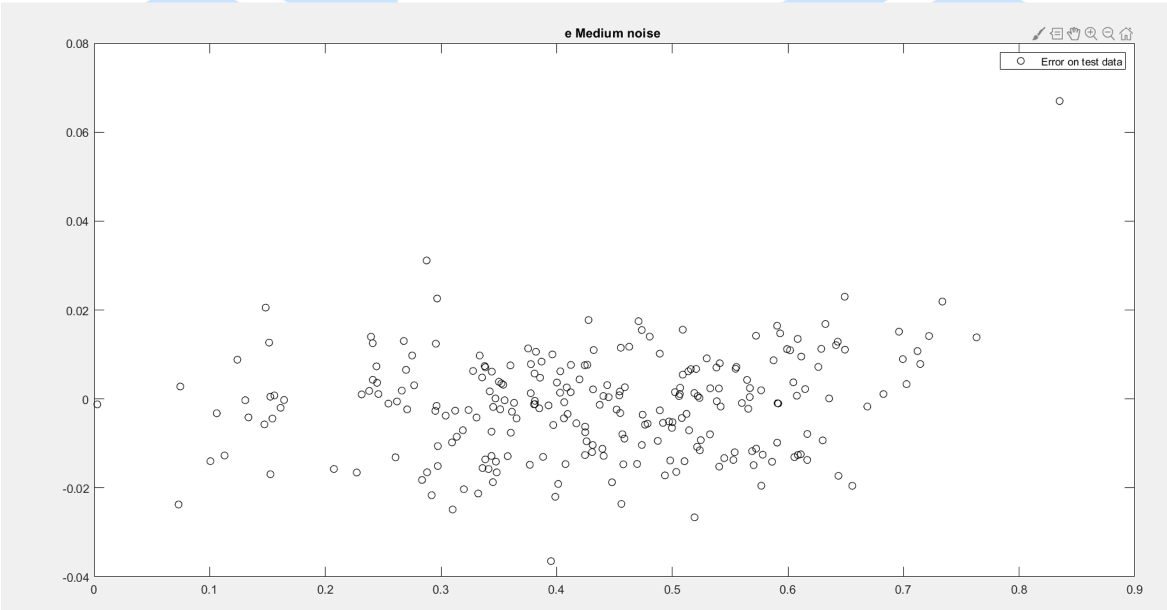
-1.5000
-0.8039
0.0572
-0.1311
0.1289
0.3760
0.3811
0.1525
-0.2259
-0.6675

حالت داده های با نویز متوسط

در شکل زیر خروجی مقدار تخمین زده شده و مقدار واقعی در وسط بازه بر روی یک دیگر قرار گرفته اند ولی در دو سر انتهایی بازه مقداری خطا بسیار بیشتر و قابل ملاحظه میشود که این نیز به دلیل کمبود داده در ان بازه و وجود نویز با واریانس متوسط منطقی است همچنین مقدار خطا در این بازه نسبت به حالت قبلی بسیار بیشتر شده است



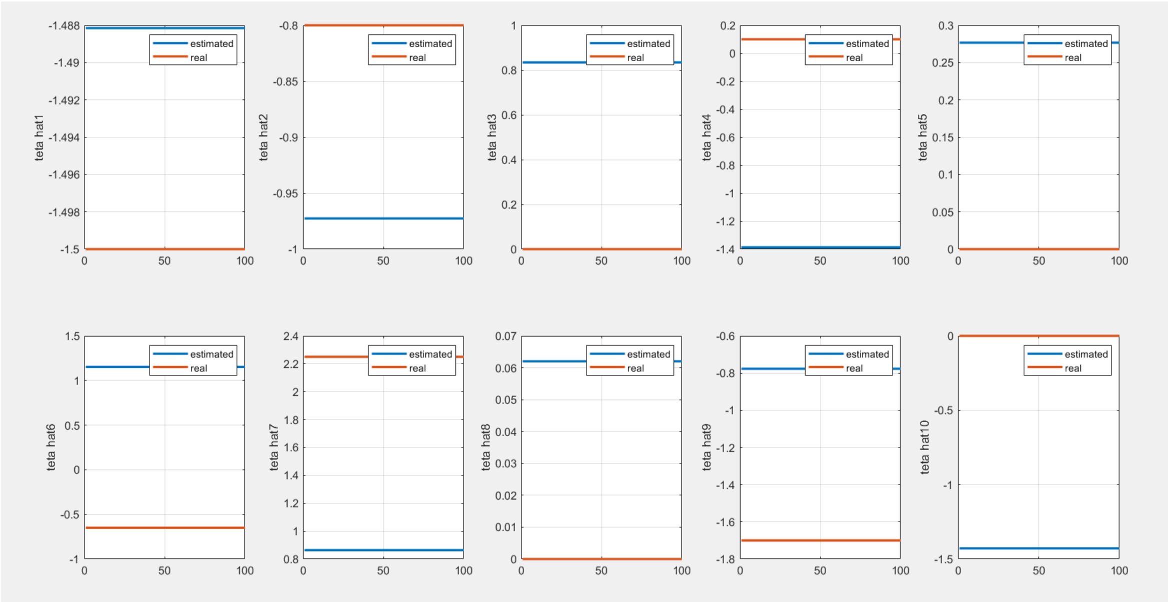
مقدار خطا این حالت در شکل زیر نشان داده شده است



مقادیر mse خطا در این حالت برابر است با که نسبت به حالت قبلی 1000 برابر بیشتر شده است

```
Mean squared normalized error  =

1.3466e-04
```

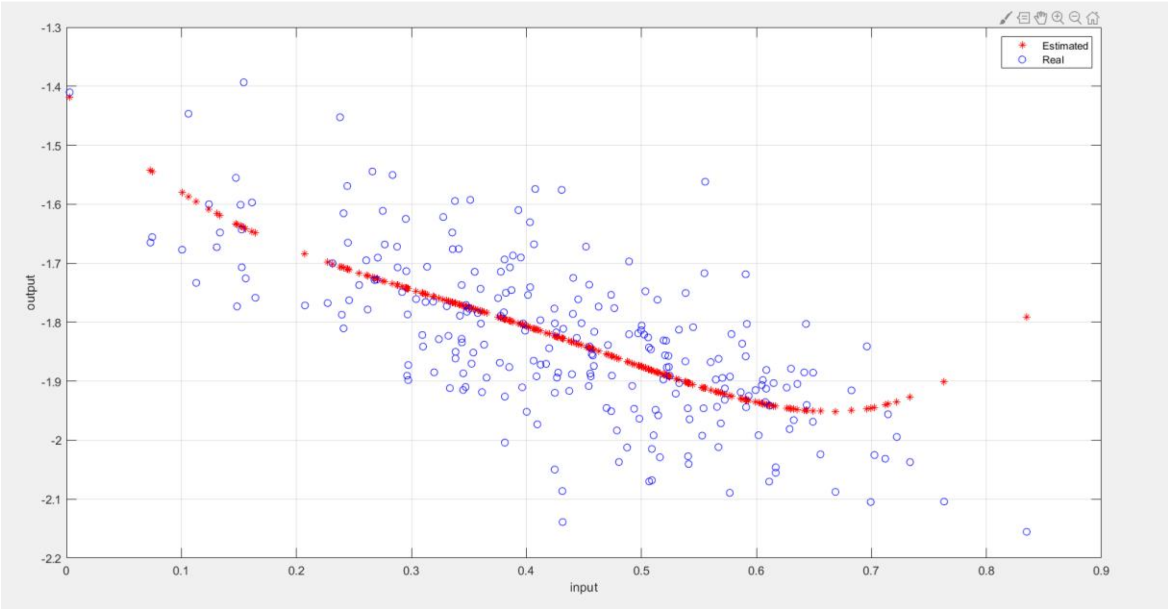
مقادیر تتا های تخمین زده شده با مقادیر تتا های واقعی دارای اختلاف بسیار کمی هستند اما این اختلاف نسبت به حالت قبلی بیشتر شده است



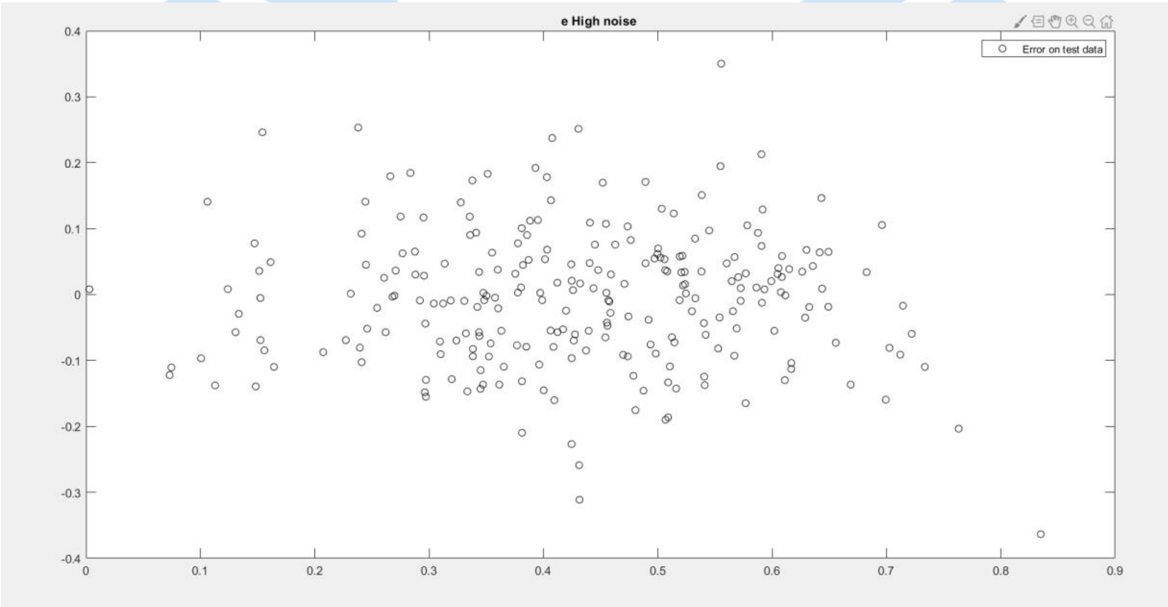
-1.4881
-0.9726
0.8341
-1.3870
0.2767
1.1522
0.8635
0.0620
-0.7769
-1.4283

حالت داده های با نویز زیاد

در شکل زیر خروجی مقدار تخمین زده شده و مقدار واقعی در بازه وسطی بر روی یک دیگر قرار گرفته اند ولی در دو سر انتهایی بازه مقداری خطا بسیار بیشتر و قابل ملاحظه میشود که این نیز به دلیل کمبود داده در ان بازه و وجود نویز با واریانس زیاد منطقی است همچنین مقدار خطا در این بازه نسبت به حالت قبلی بسیار بیشتر شده است و میتوان گفت که تخمین انجام شده خطا بسیار زیادی دارد



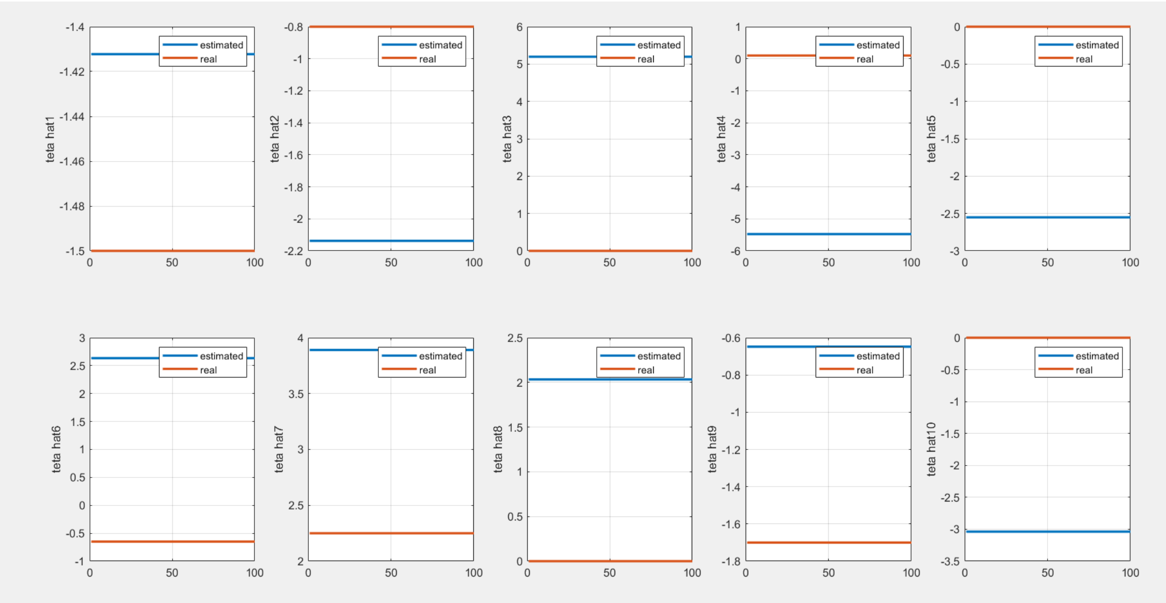
مقدار خطا این حالت در شکل زیر نشان داده شده است



مقادیر mse خطا در این حالت برابر است با که نسبت به حالت قبلی 100 برابر بیشتر شده است و مقدار خطا در این حالت بسیار زیاد است و تخمین صورت گرفته دقت بسیار کمی دارد

```
Mean squared normalized error  =

   0.0105
```

مقادیر تتا های تخمین زده شده با مقادیر تتا های واقعی دارای اختلاف بیشتری نسبت به حالت قبلی هستند



همان طور که در بخش های بالا مشاهده شده پارامتر های تخمین زده شده در حالت بدون نویز برابر با پارامتر های واقعی اند و هر چه مقدار نویز بیشتر شود مقدار فاصله پارامتر های تخمین زده شده از مقدار پارامتر های واقعی بیشتر شده است.

ب

نمودار Error Bar میزان اطمینان ما به تخمین صورت گرفته را مشخص میکند. این نمودار از یک باند تشکیل شده است که با اطمینان میتوان گفت که مقدار خروجی بدون خطا حتما در این باند قرار میگیرد با استفاده از این نمودار میزان اطمینان به تخمین در بازه های مختلف را میتوان بررسی کرد در بازه هایی از نمودار که این باند باز میشود نشان دهنده این موضوع است که در این ناحیه تعداد داده ها کم بوده است یا واریانس نویز در این نواحی بیشتر بوده است در نتیجه باند بزرگتری برای مقدار خطا در نظر گرفته میشود که نشان دهنده دقت کمتر تخمین در این بازه میباشد و به طور برعکس در نواحی که باند نمودار جمع تر میشوند نشان دهنده تراکم داده ها در این ناحیه میباشد و در نتیجه باند کوچکتری برای خطا بدست میاید و تخمین صورت گرفته از دقت بیشتری برخوردار است.

فرمول محاسبه Error Bar به صورت زیر است

$$y \pm \sqrt{diag\ (cov(y))}$$

$$cov(y) = E((y - E(y))(y - E(y))^T)$$
$$cov(y) = E(X(\theta - E(\theta))(\theta - E(\theta))^T X^T)$$
$$cov(y) = Xcov(\theta)X^T$$
$$cov(\theta) = \delta^2(X^T X)^{-1}$$
$$cov(n) = \delta^2 I$$

در فرمول بالا به مقدار کوواریانس نویز نیاز داریم از انجایی که در واقعیت به کوواریانس نویز دسترسی مستقیم نداریم باید یک تقریبی از ان را قرار بدهیم با فرض درست بودن نتیجه LS و مقدار خطا باید شبیه به مقدار نویز شود در نتیجه در اینجا به جای کوواریانس نویز از کوواریانس خطا استفاده میکنیم

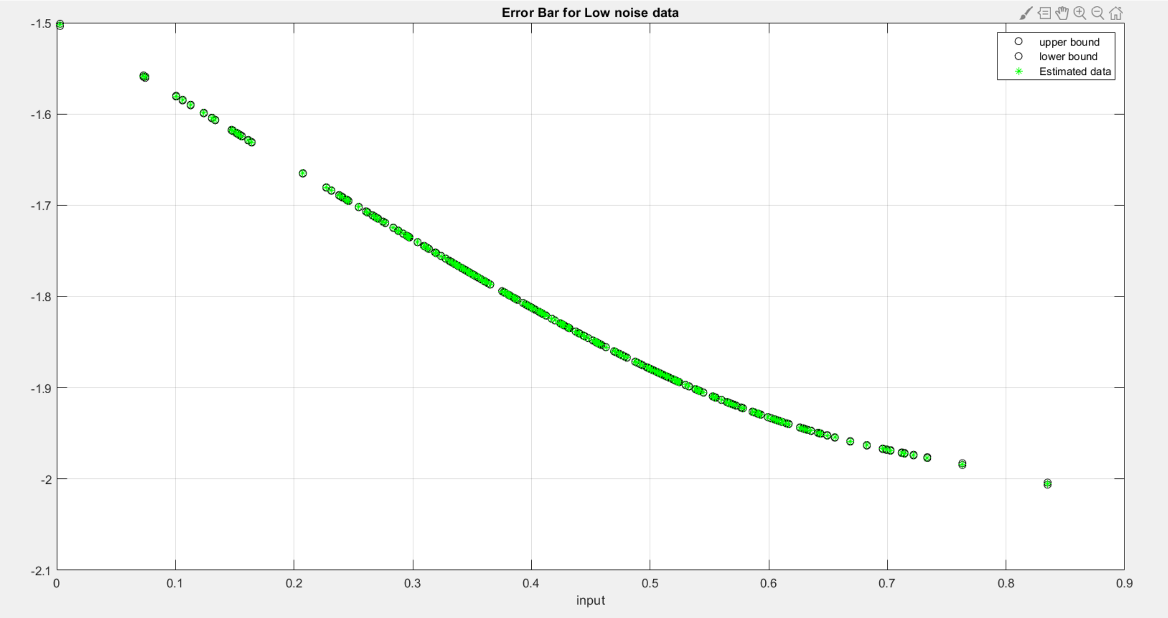
با توجه به اینکه میدانیم میانگین خطا برابر با صفر است داریم

$$cov(e) = E(e^T e) = E(trace(ee^T)) = trace\ E(ee^T) = trace\ (\delta^2(I - X(X^T X)^{-1} X^T))$$
$$= \delta^2(N - trace(X(X^T X)^{-1} X^T))$$
$$= \delta^2(N - trace(I_n))$$
$$E(e^T e) = \delta^2(N - n)$$
$$\delta^2 = \frac{E(e^T e)}{(N - n)}$$

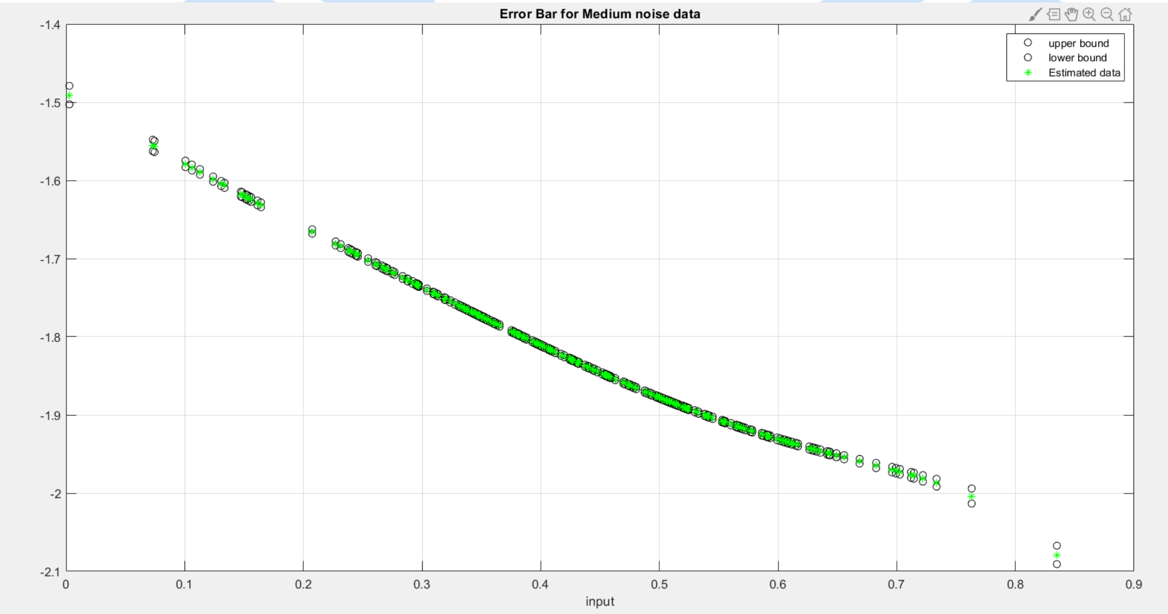
با استفاده از اثبات بالا میتوانیم نمودار Error Bar را رسم کنیم.

میدانیم که توزیع داده به صورت نرمال است و در ابتدا و انتها بازه مقدار داده ها کمتر است وقتی که نویز کم میباشد دقت تخمین LS در ان بازه ها قابل قبول است اما وقتی نویز زیاد میشود در ابتدا و انتها بازه به دلیل وجود داده های کمتر و همچنین مقدار نویز زیادتر تخمین بدتر به دست میاید و تخمین بدست امده اطمینان کمتری داریم و مقدار بازه Error Bar بزرگتر میشود

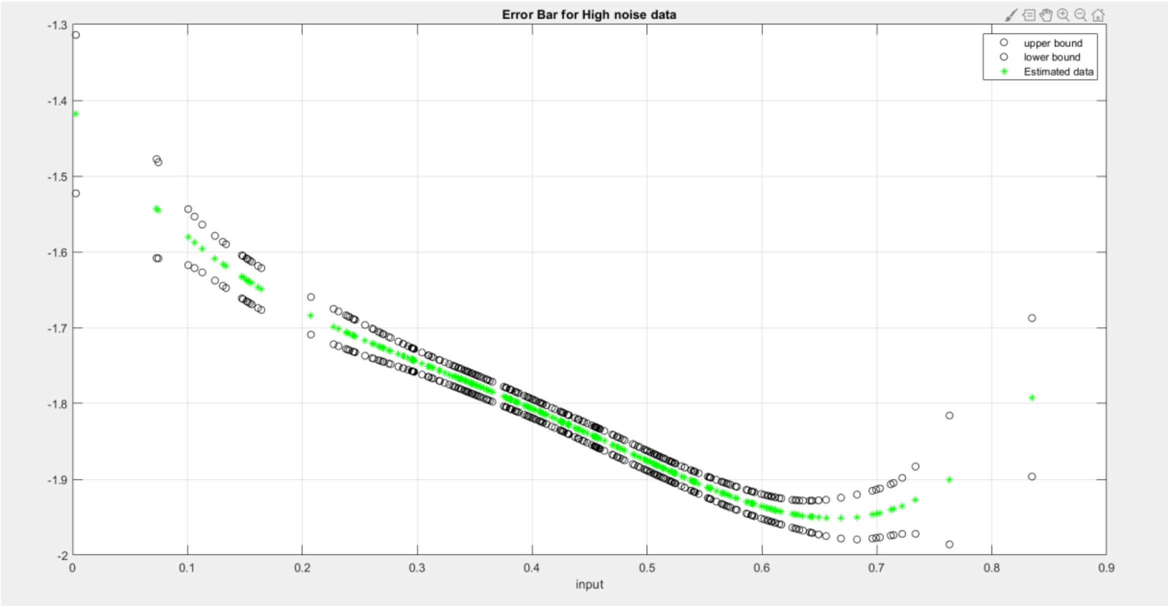
حالت داده ها با نویز کم



حالت داده ها با نویز متوسط



حالت داده ها با نویز زیاد

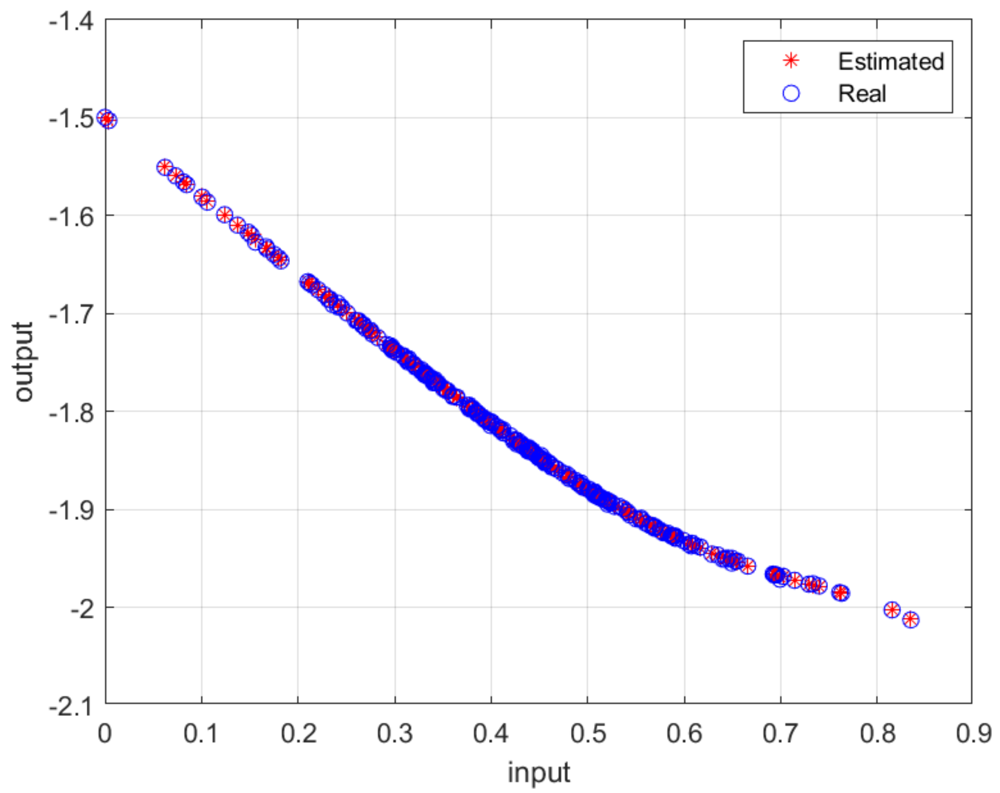


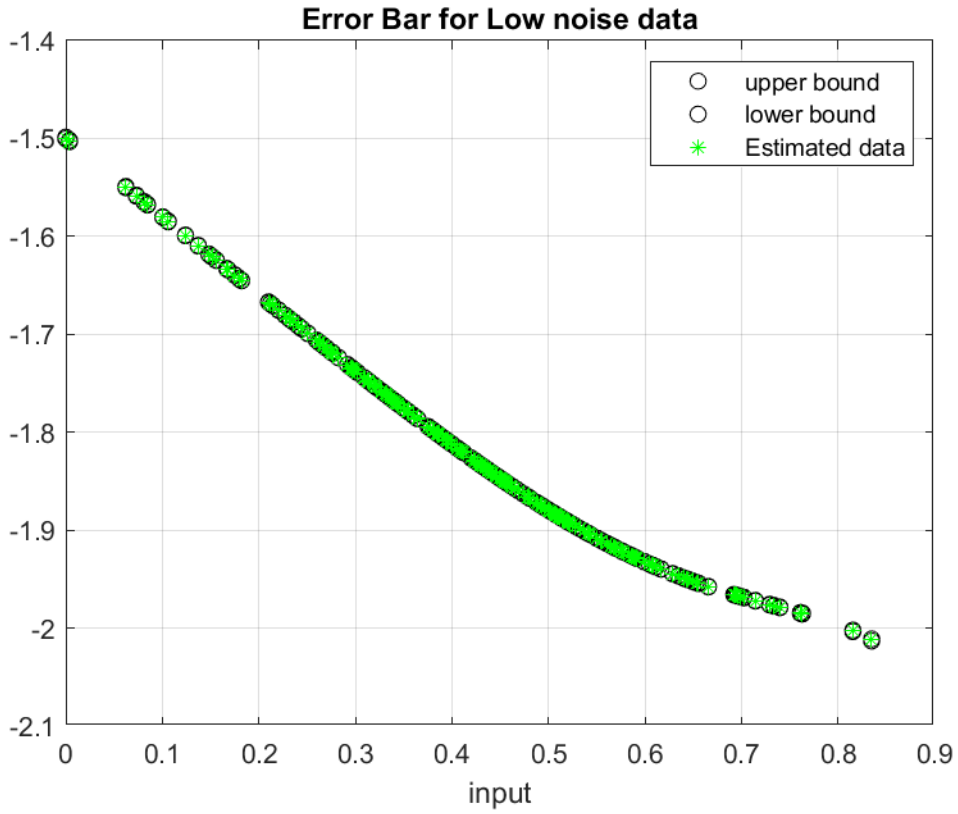
قسمت ج

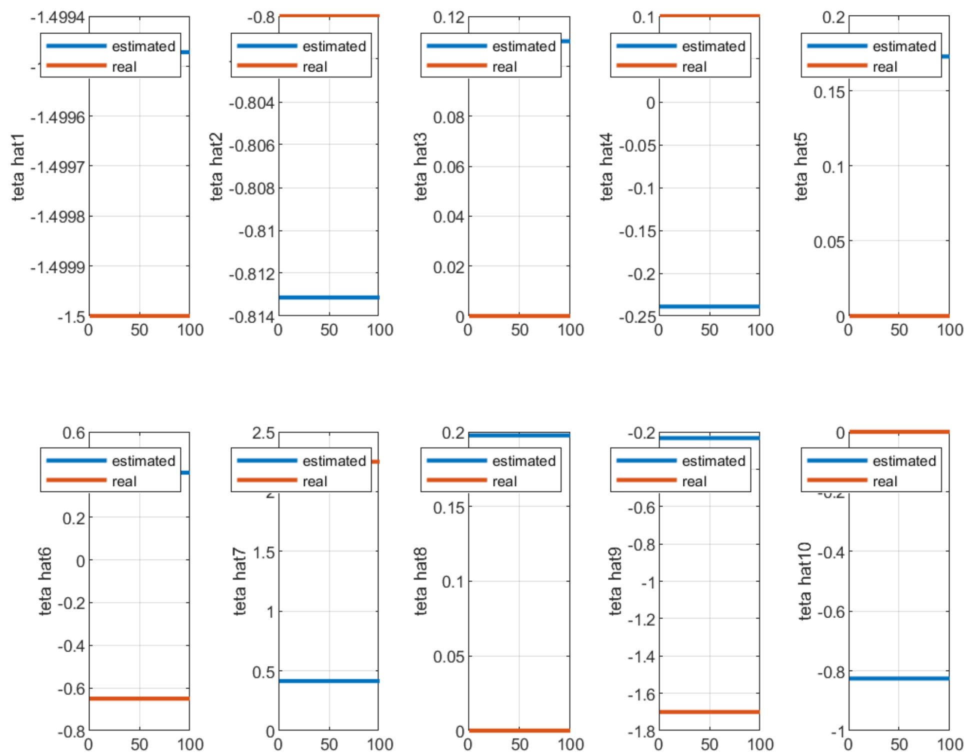
با دوبرابر کردن تعداد مشاهدات یعنی 150 داده اموزش و 50 داده تست نتایج زیر را بدست میاوریم

در انتها بعد از شکل ها نتیجه گیری انجام گرفته است

حالت با نویز کم

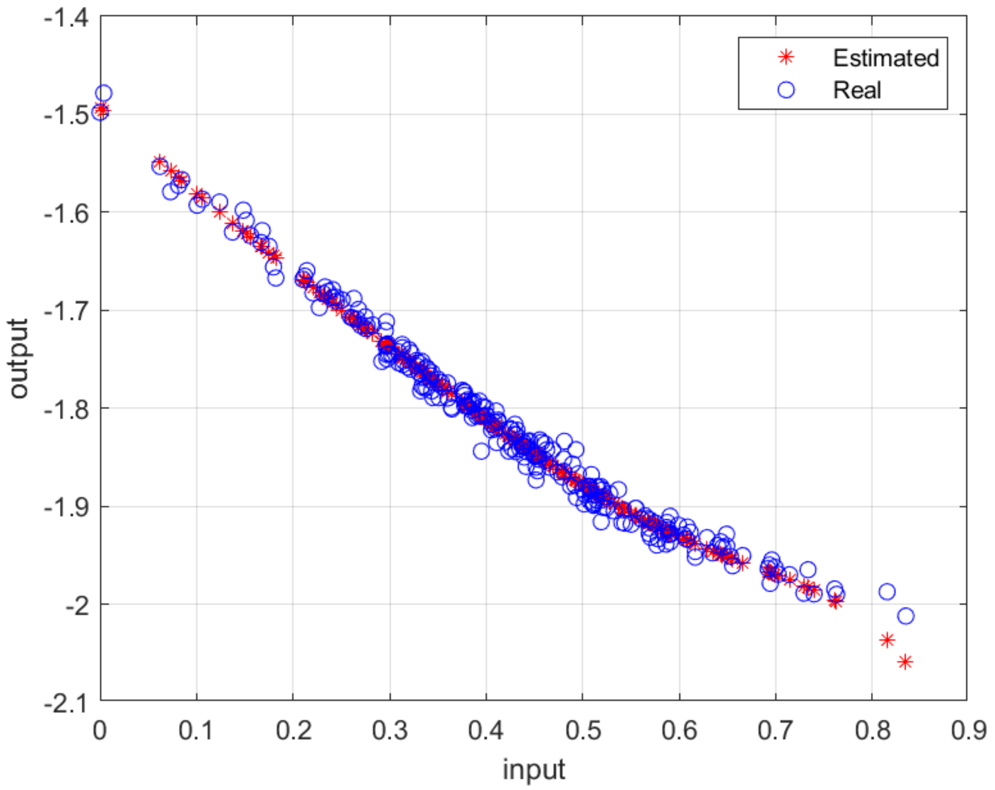



Error Bar for Low noise data
upper bound
lower bound
Estimated data
input
-1.4
-1.5
-1.6
-1.7
-1.8
-1.9
-2
-2.1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

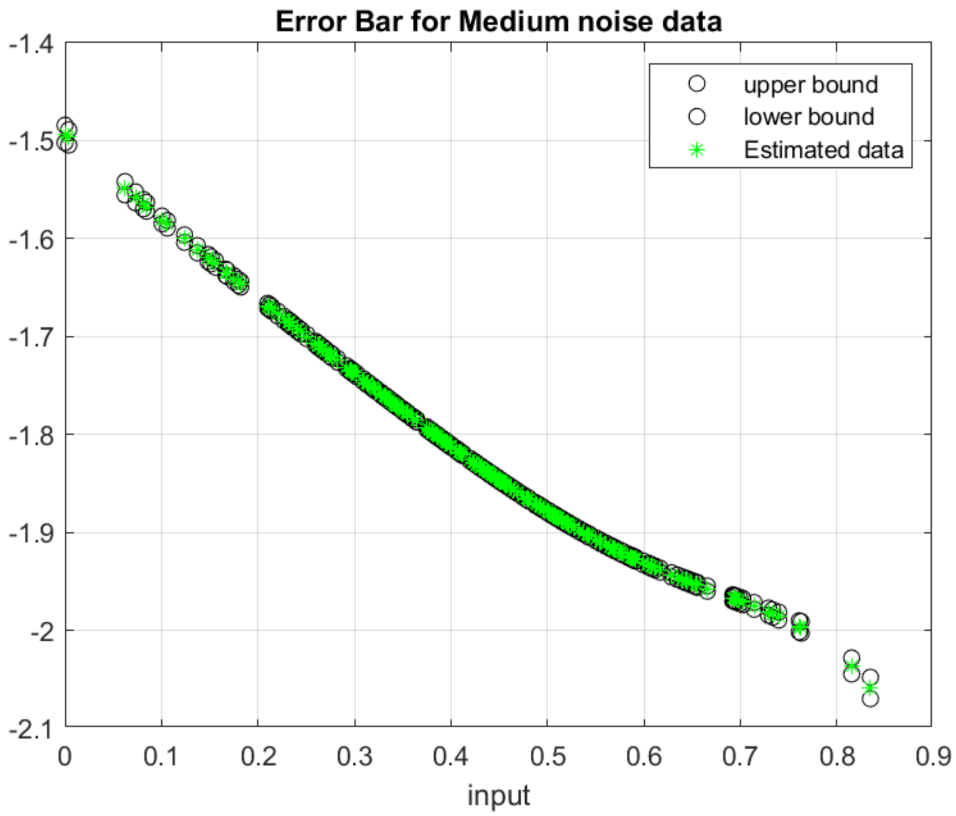

estimated
real
teta hat1
teta hat2
teta hat3
teta hat4
teta hat5
teta hat6
teta hat7
teta hat8
teta hat9
teta hat10
0
50
100

```
Mean squared normalized error  =

   1.1854e-06
```

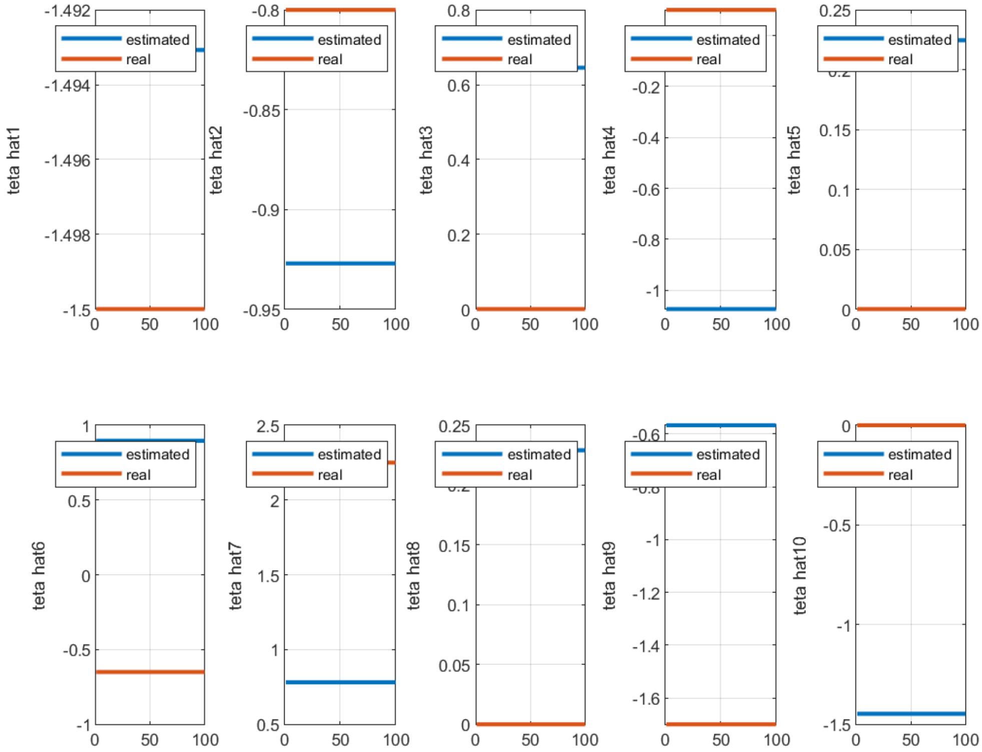
حالت دوم داده های با نویز متوسط



Estimated
Real
output
input

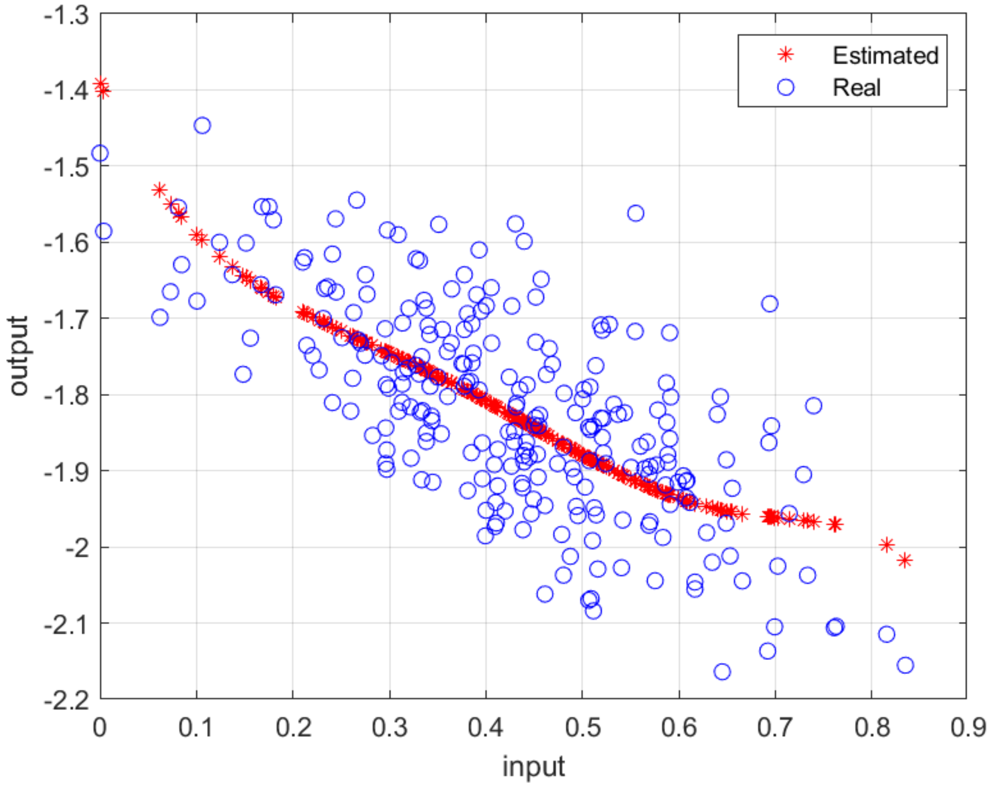


Error Bar for Medium noise data
upper bound
lower bound
Estimated data
input

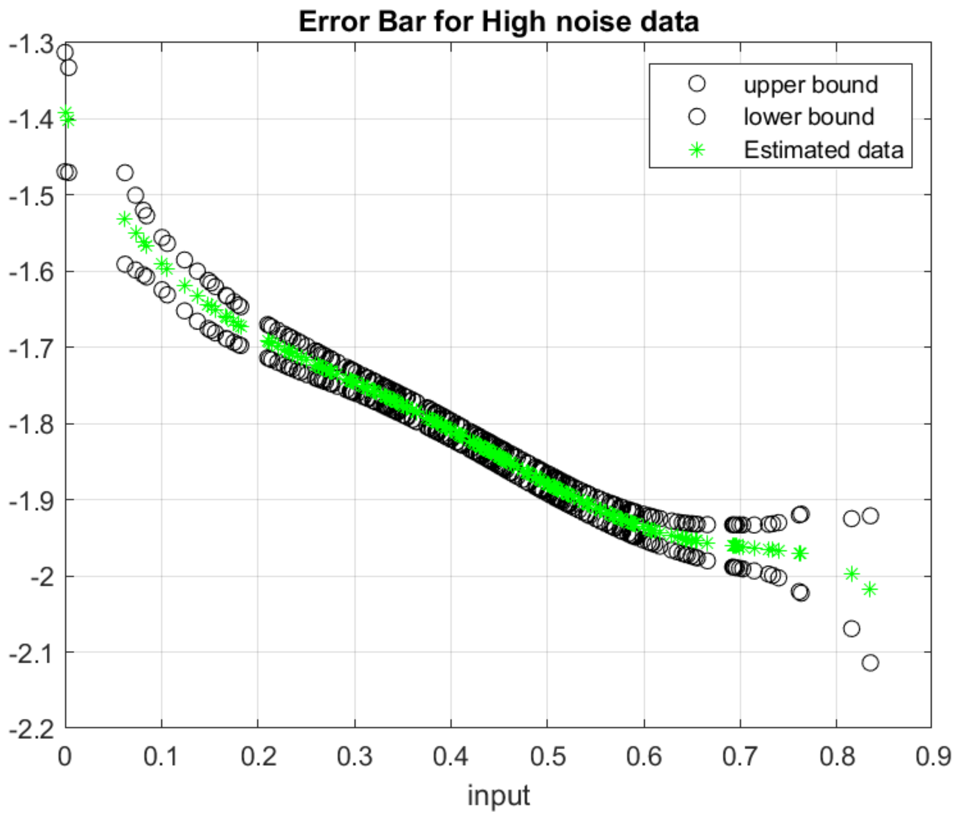



```
Mean squared normalized error  =

   1.3145e-04
```

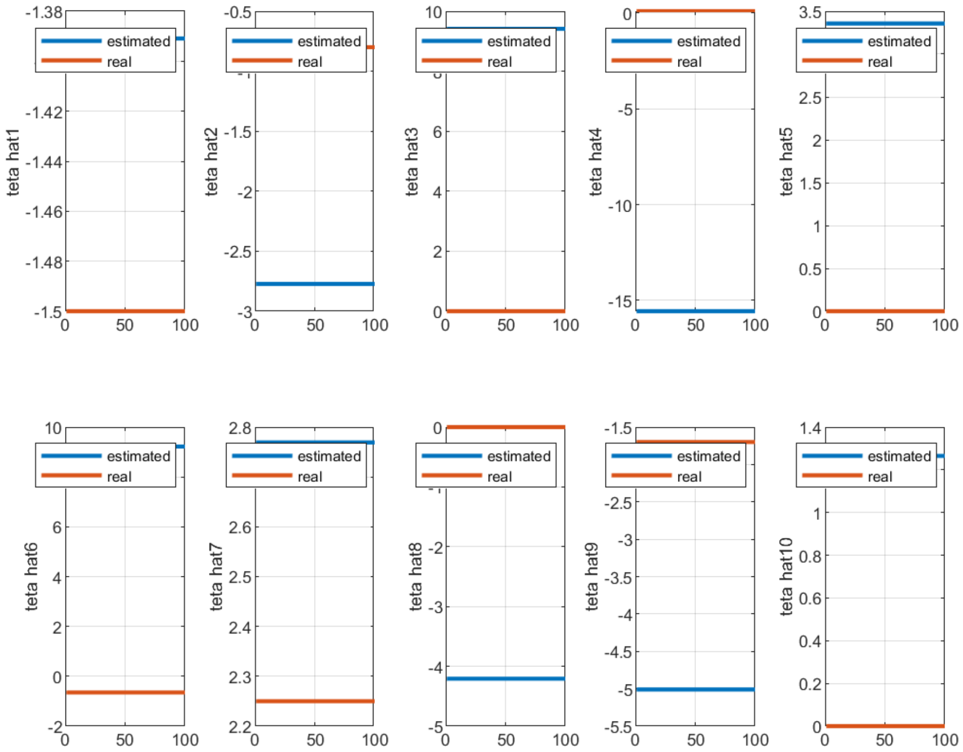
حالت سوم داده های با نویز زیاد



Estimated
Real
output
input
-1.3
-1.4
-1.5
-1.6
-1.7
-1.8
-1.9
-2
-2.1
-2.2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9



Error Bar for High noise data
upper bound
lower bound
Estimated data
input
-1.3
-1.4
-1.5
-1.6
-1.7
-1.8
-1.9
-2
-2.1
-2.2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9




```
Mean squared normalized error  =

      0.0102
```

نتیجه گیری

	حالت اول 75 داده اموزش 25 داده تست	حالت دوم 150 داده اموزش 50 داده تست
نویز کم	1.4 e-6	1.18 e-6
نویز متوسط	1.34 e-4	1.31 e-4
نویز زیاد	0.0105	0.0102

شکل 1

همانگونه که در جدول بالا مشاهده میشود با افزایش تعداد مشاهدات الگوریتم نتیجه بهتری در هر سه حالت مختلف نویز از خود نشان داده است و علت این امر این است که در فرمول های اثبات شده برای

حالتی است که N به سمت بی نهایت میرود خاصیت های اماری تخمین مثل بدون بایاس بودن و دقت بالا داشتن را دارا میباشد و در صورتی که N محدود باشد این خاصیت ها به طور کامل برقرار نیستند

البته قبل از اضافه کردن ترم رگولازیسون این اختلاف بسیار بیشتر بود اما با اضافه کردن این ترم این اختلاف کمتر شد اما هنوز نتایج حالت دوم بسیار بهتر از نتایج حالت اول میباشد

قسمت د

در ابتدا داده ها را به دو دسته اموزش و تست تقسیم میکنیم برای هر دسته با مقادیر مختلف n به شناسایی سیستم میپردازیم و مقادیر خطا خروجی را محاسبه میکنیم و در یک جدول نمایش میدهیم مشاهده میشود که در برای داده های اموزش با افزایش مقدار n مقدار خطا کمتر میشود نمودار داده های تست نیز تا حدی مشابه با داده های اموزش میباشد اما بعد از یک مقدار n خطا نمودار تست شروع به افزایش پیدا کردن میکند و زیاد میشود این نقطه قبل از افزایش مقدار خطا در داده های تست به عنوان n بهینه انتخاب میشود

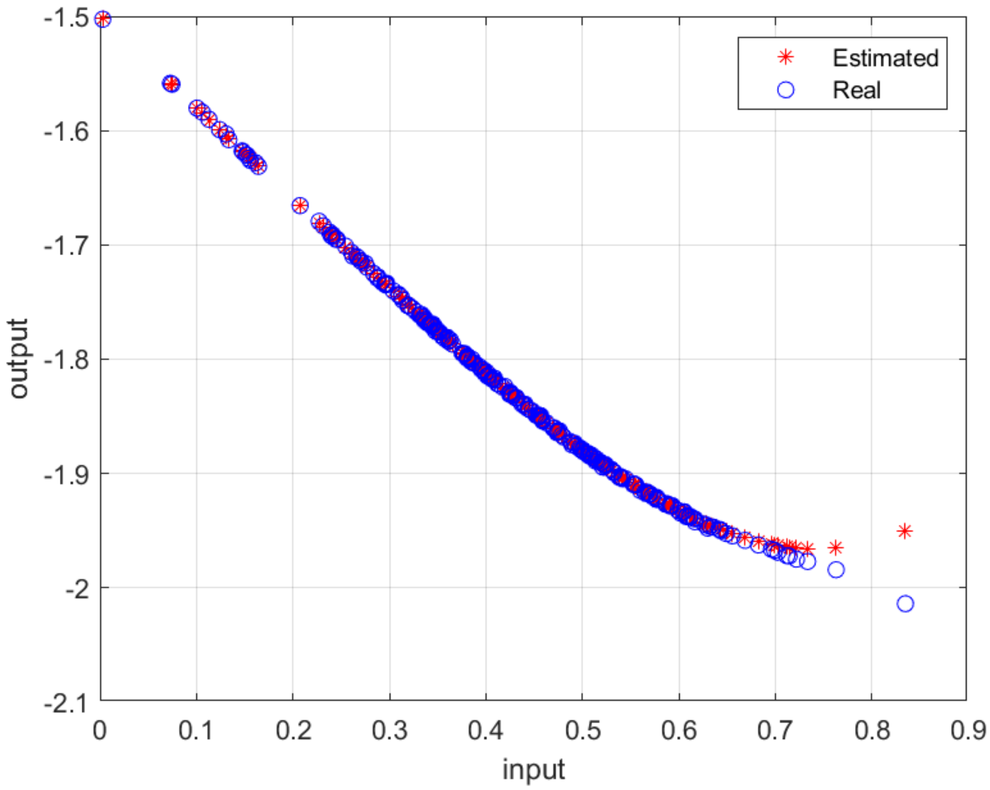
البته در انتخاب n باید به اصل خصت نیز توجه کرد و اگر کاهش مقدار n باعث ایجاد خطا زیادی نشود میتوانیم مقدار n را کمتر در نظر بگیریم تا مدل ساده تری داشته باشیم

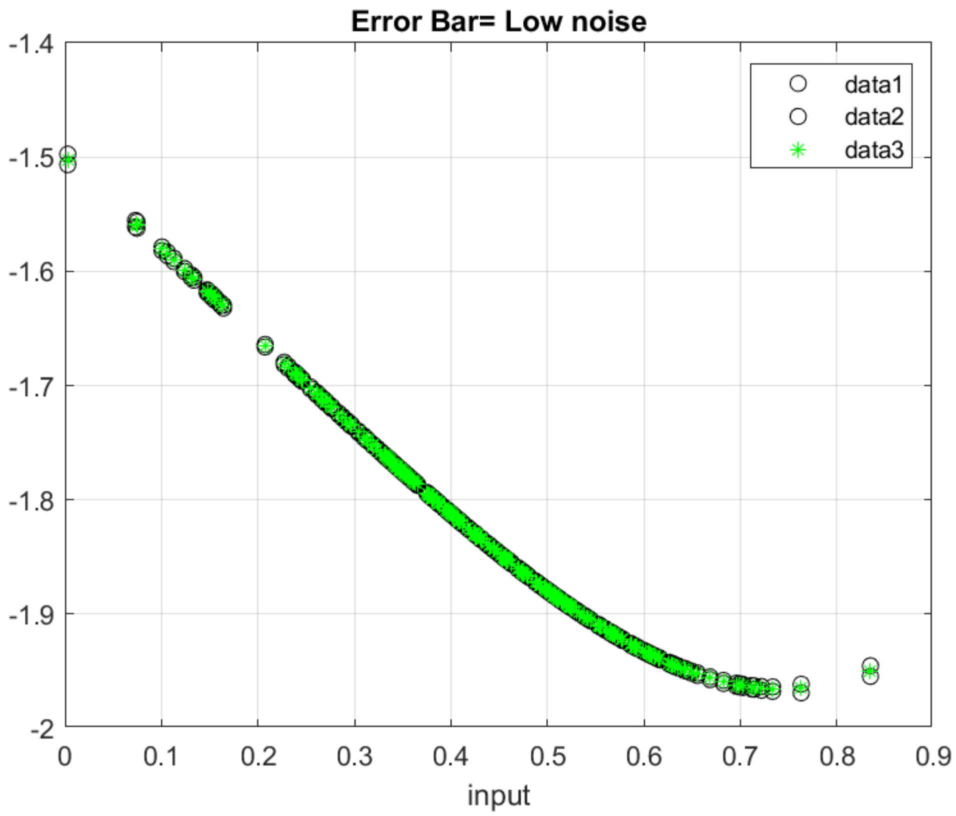
قسمت ه

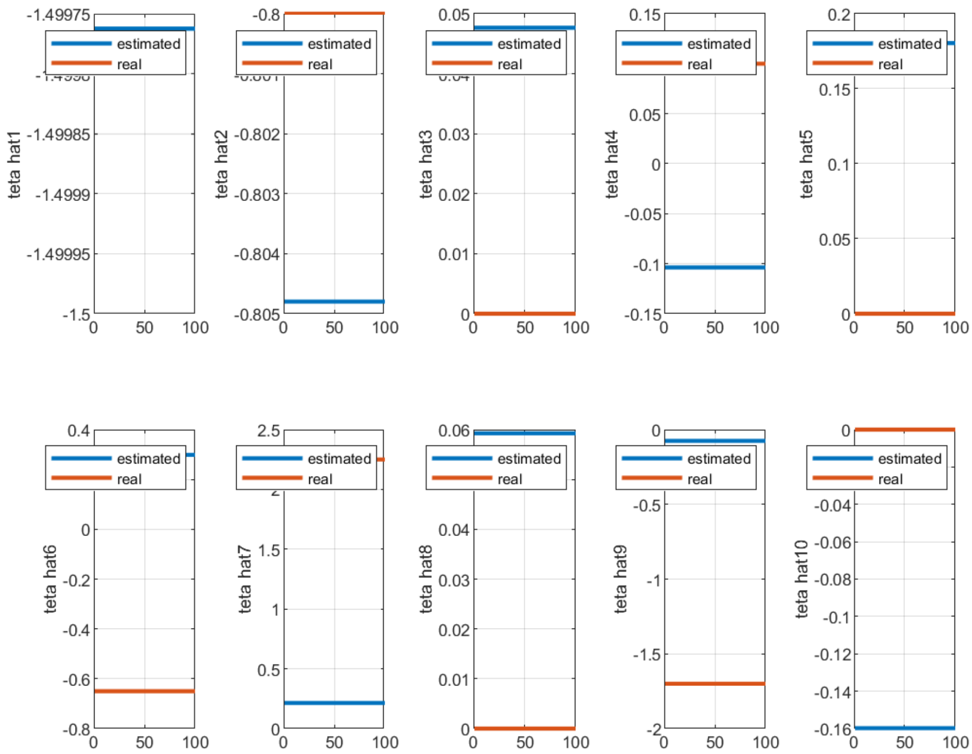
با توجه به مدل داده شده در صورت سوال میتوانیم نتیجه بگیریم که رگرسور های با توان کمتر مهم تر میباشد به طور مثال رگرسور هایی که توان صفر یا یک دارند در شناسایی این سیستم ارزش بیشتری دارند و مهم تر اند. زیرا در تشکیل خروجی سهم بیشتری دارند.

سوال دوم

حالت نویز کم

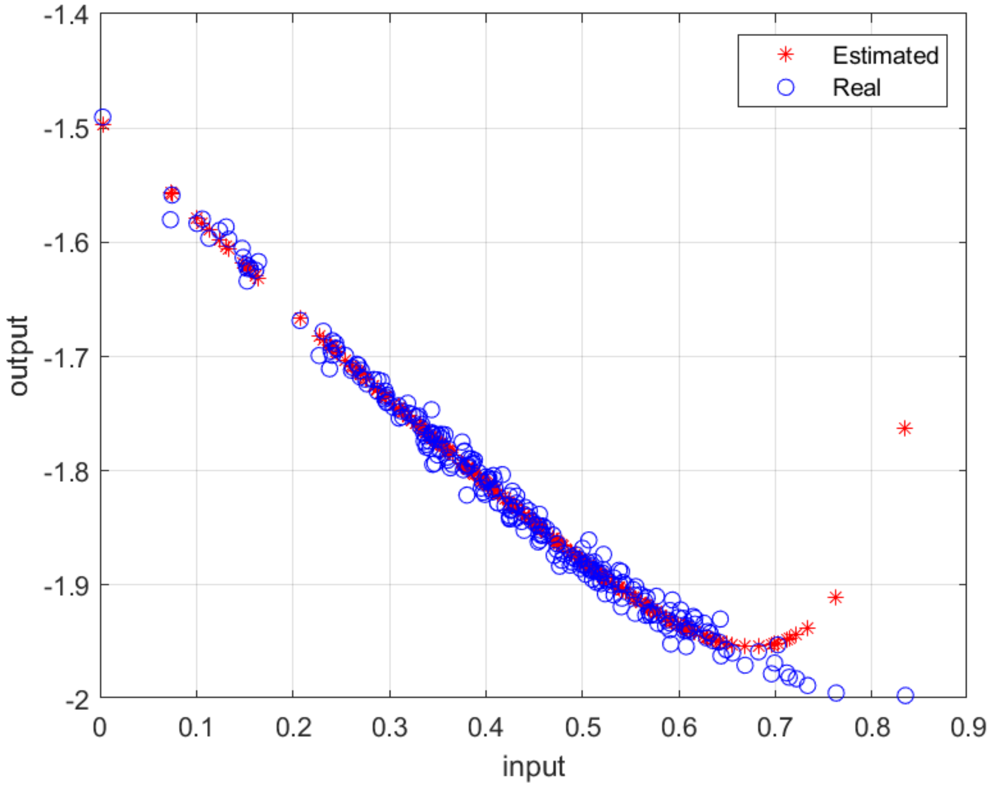

Estimated
Real
output
input
-1.5
-1.6
-1.7
-1.8
-1.9
-2
-2.1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

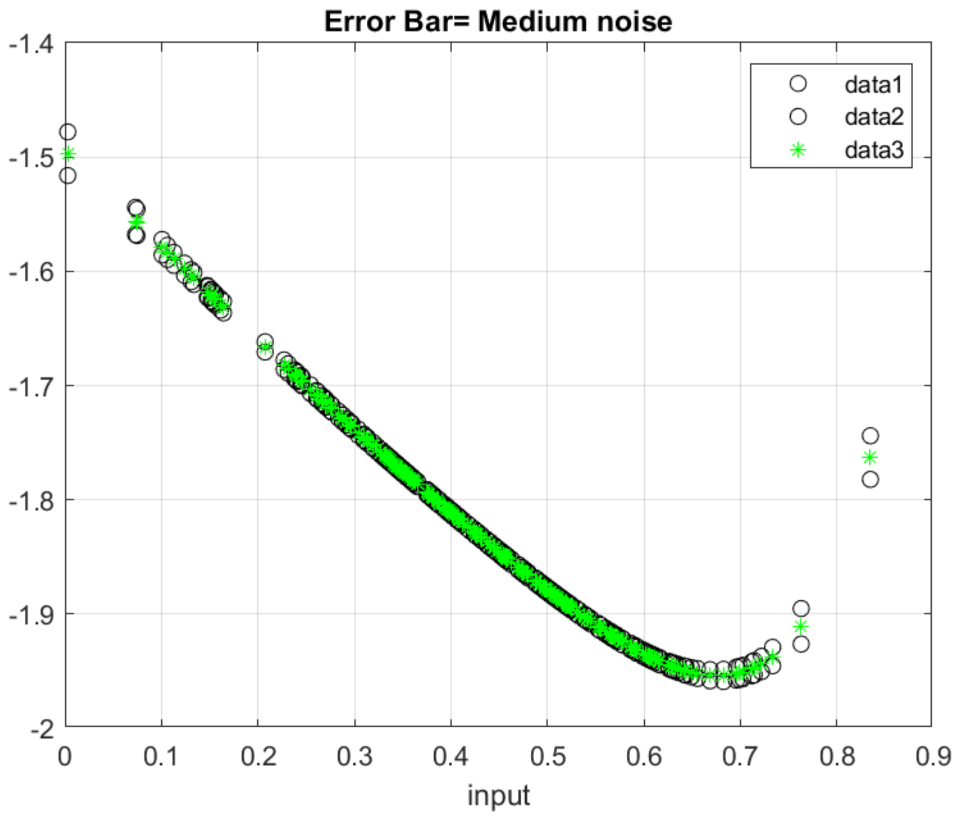

Error Bar= Low noise
data1
data2
data3
input
-1.4
-1.5
-1.6
-1.7
-1.8
-1.9
-2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

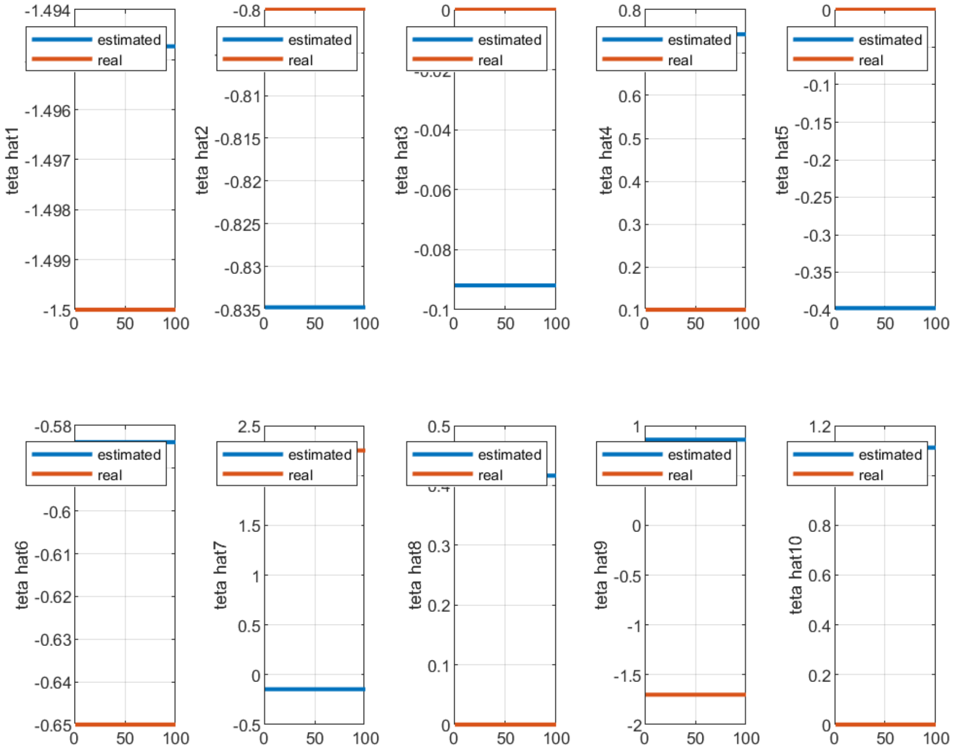



```
noise =

    'Low noise'


Mean squared normalized error  =

   1.9643e-05
```

حالت نویز متوسط

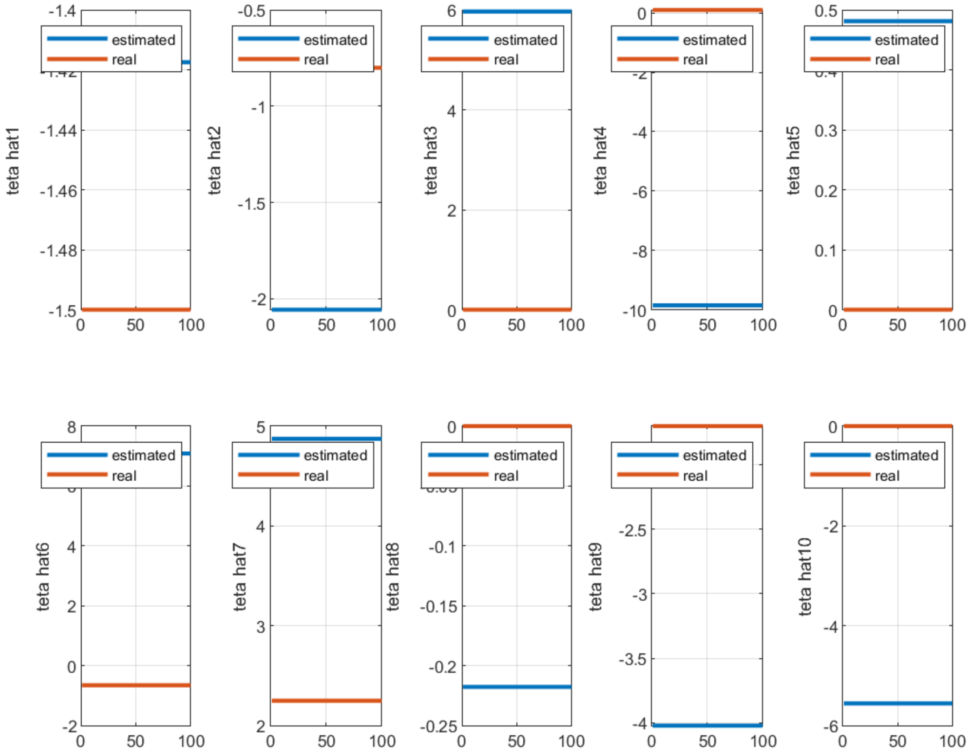

Estimated
Real
output
input
-1.4
-1.5
-1.6
-1.7
-1.8
-1.9
-2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

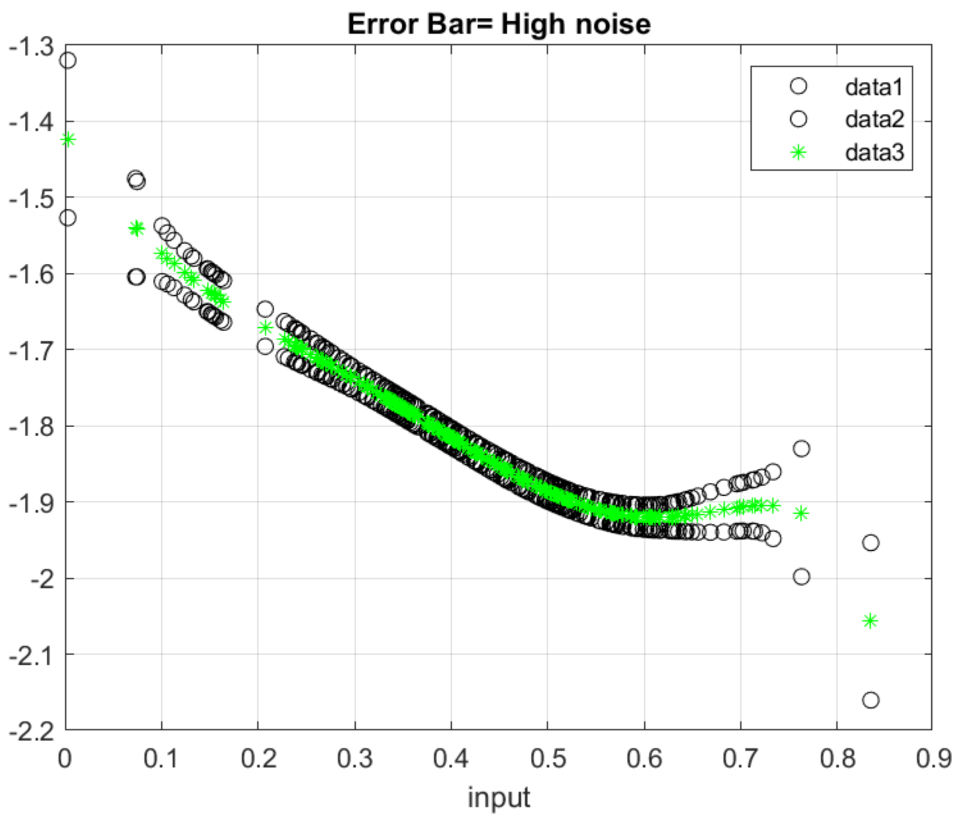

Error Bar= Medium noise
data1
data2
data3
input
-1.4
-1.5
-1.6
-1.7
-1.8
-1.9
-2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

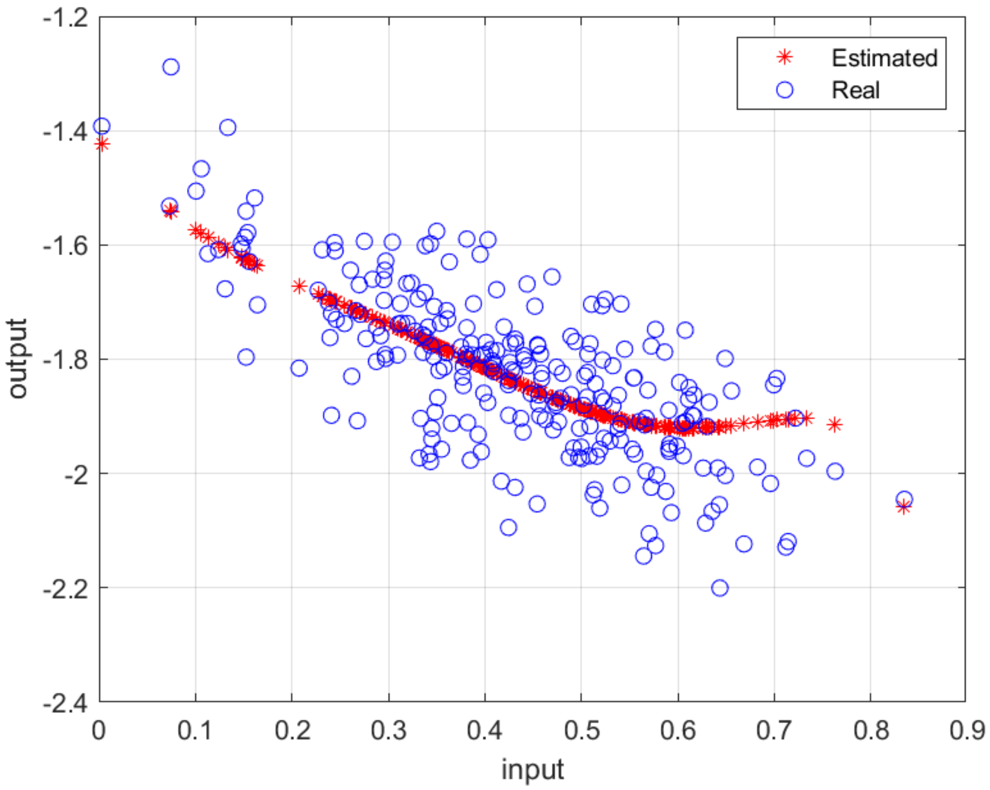



```
noise =

    'Medium noise'


Mean squared normalized error  =

   3.5085e-04
```

حالت نویز زیاد


teta hat1
teta hat2
teta hat3
teta hat4
teta hat5
teta hat6
teta hat7
teta hat8
teta hat9
teta hat10
estimated
real
0
50
100


Error Bar= High noise
data1
data2
data3
input
-1.3
-1.4
-1.5
-1.6
-1.7
-1.8
-1.9
-2
-2.1
-2.2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9




```
noise =

    'High noise'


Mean squared normalized error  =

    0.0228
```

نتیجه گیری

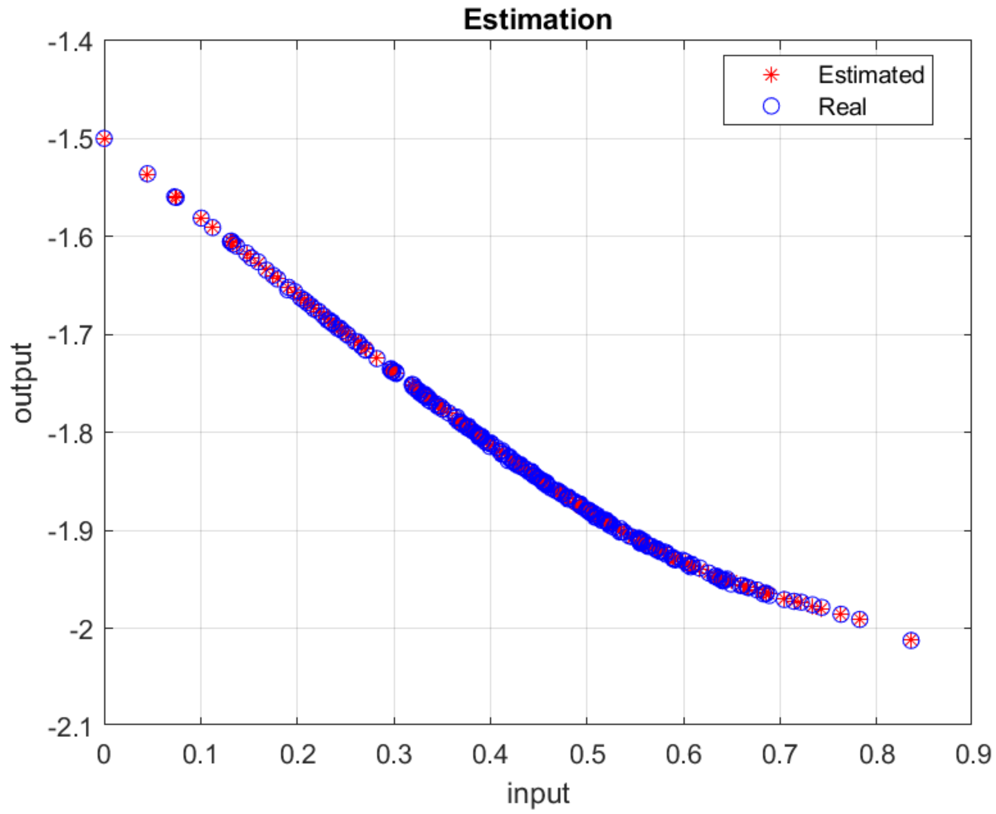
	حالت اول داده های اموزش در بازه [0 1]	حالت دوم داده های اموزش در بازه [0 0.8]
نویز کم	1.4 e-6	1.96 e-5
نویز متوسط	1.34 e-4	3.50 e-4
نویز زیاد	0.0105	0.0228

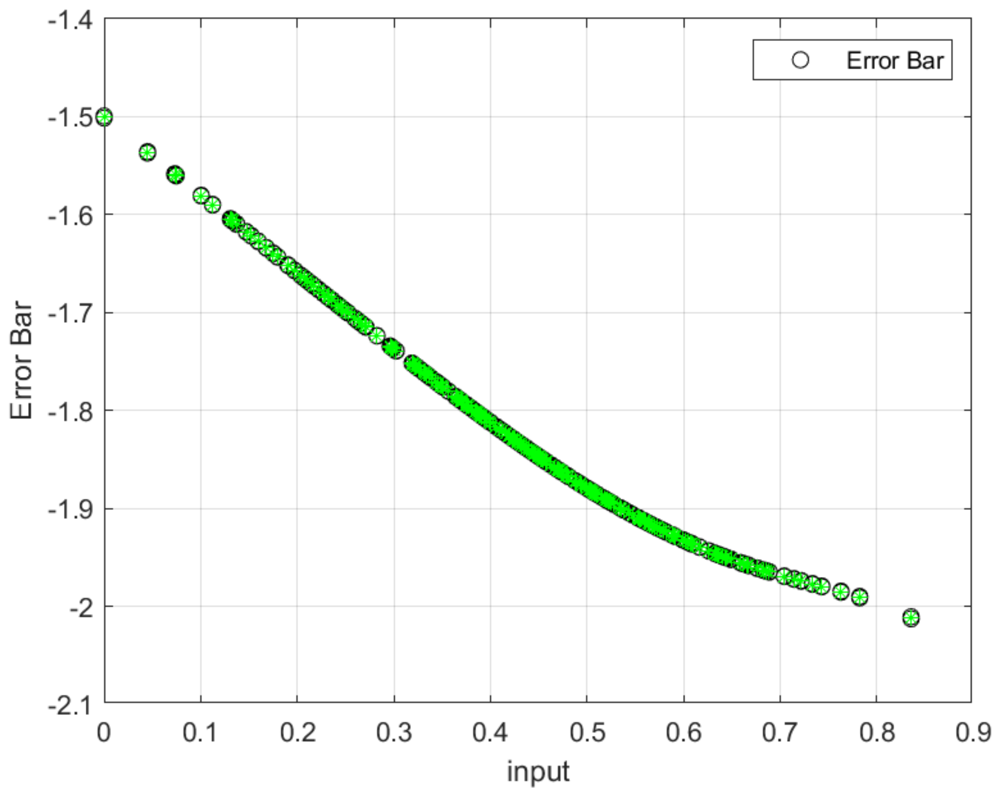
شکل 2

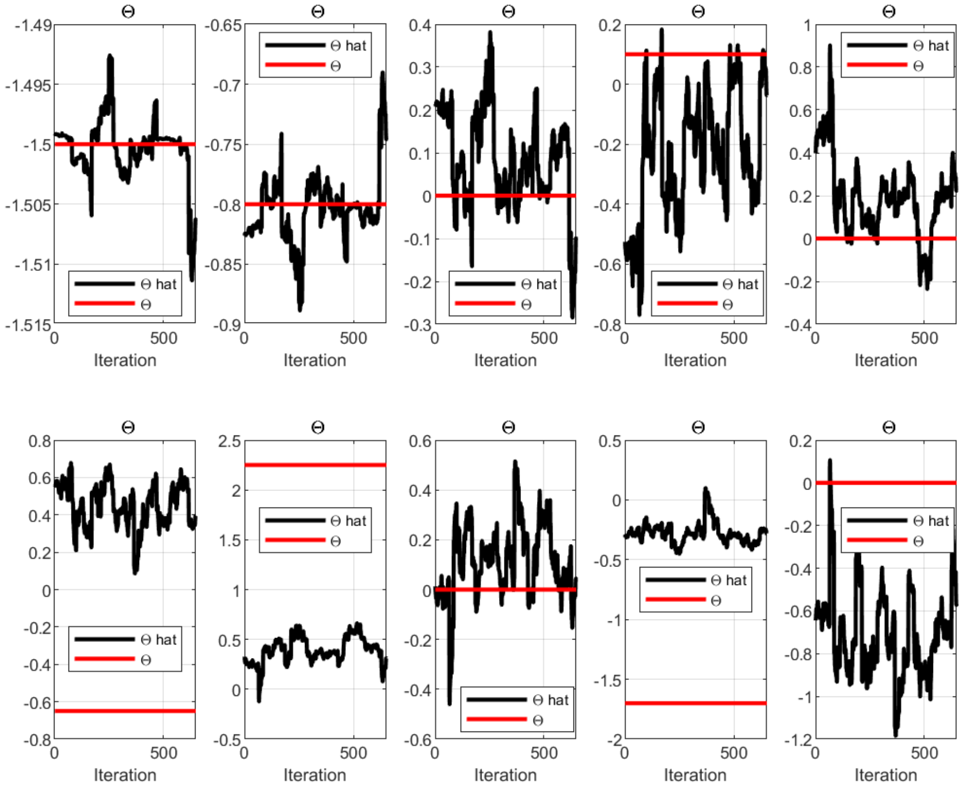
همان گونه که مشاهده میشود مقدار خطا نسبت به حالت قبلی افزایش پیدا کرده است. همچنین در نمودار های Error Bar در بازه [0.8 1] بزرگتر شده است به دلیل اینکه هیچ داده ای برای اموزش وجود نداشته در نتیجه مدل تخمین زده شده در این بازه خطا بسیار بیشتری نسبت به بقیه بازه که در فرایند اموزش شرکت داشته اند دارد و به بیان دیگر مدل شناسایی شده قابلیت برون یابی مناسبی نداشته و مقدار خطا برای ان ناحیه افزایش یافته است.

سوال سوم

حالت اول نویز کم

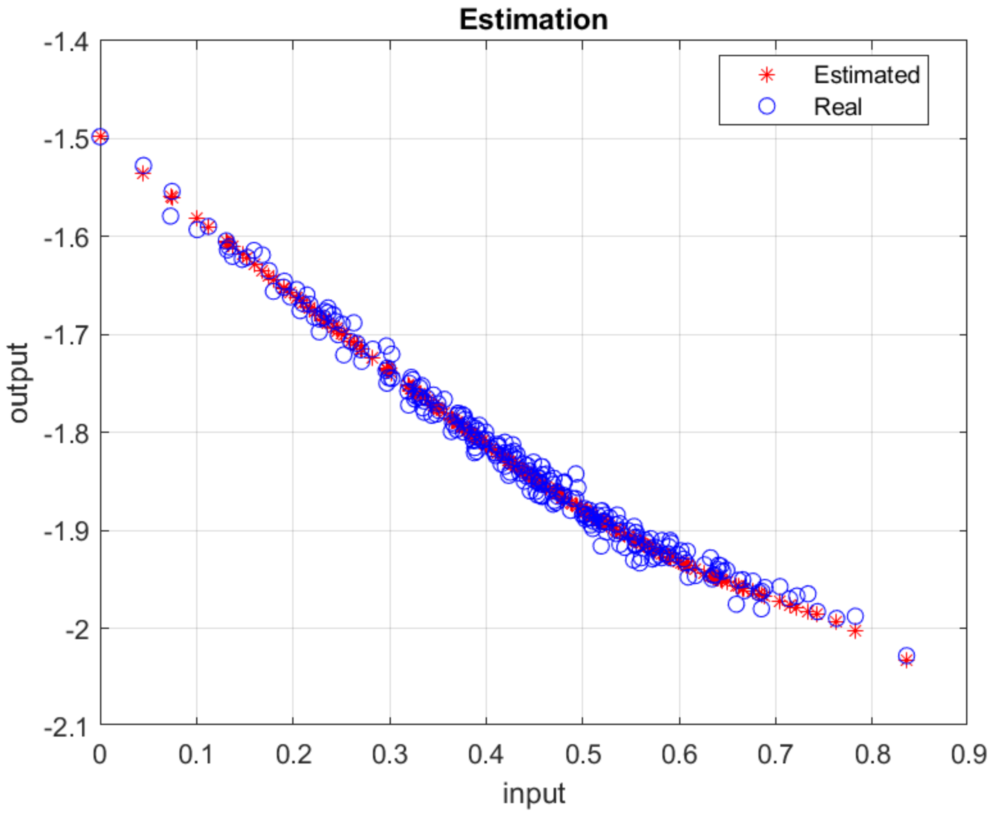

Estimation
Estimated
Real
output
input
-1.4
-1.5
-1.6
-1.7
-1.8
-1.9
-2
-2.1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

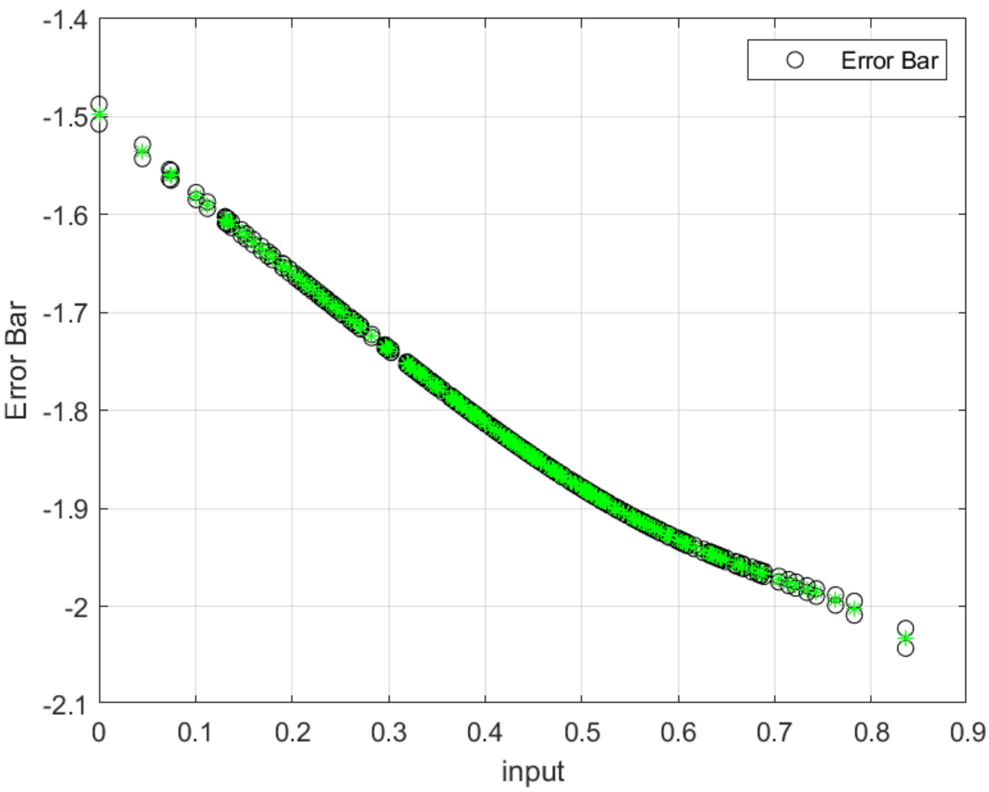

Error Bar
Error Bar
input
-1.4
-1.5
-1.6
-1.7
-1.8
-1.9
-2
-2.1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

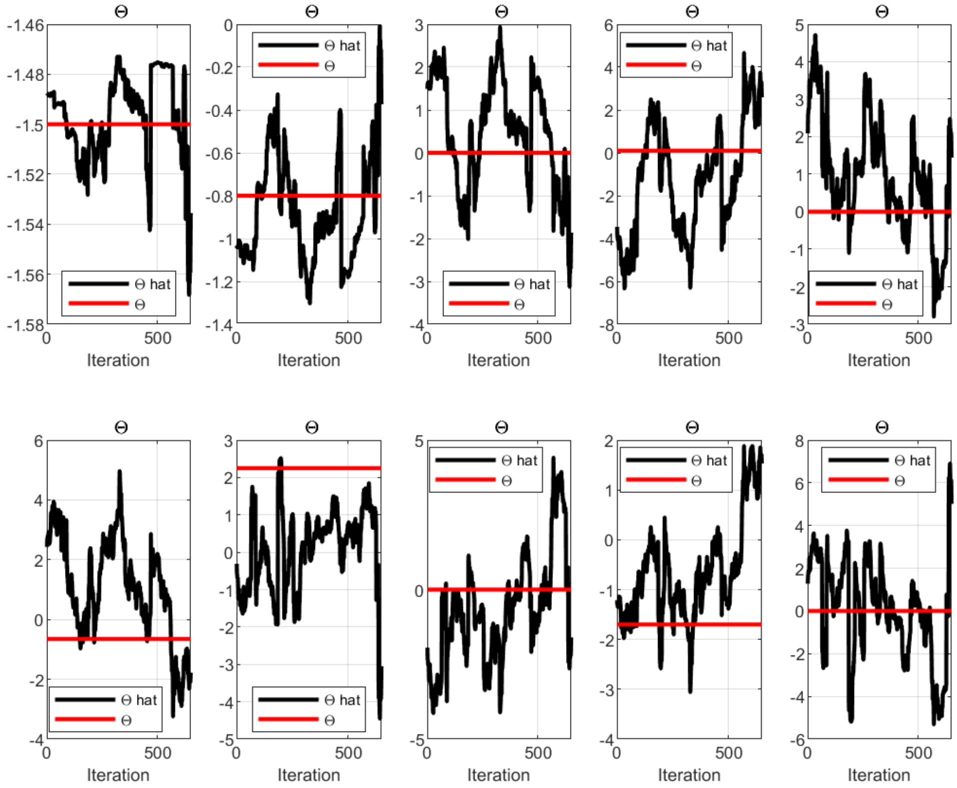



```
noise =

    'Low noise'


Mean squared normalized error  =

   1.1079e-06
```

حالت دوم نویز متوسط

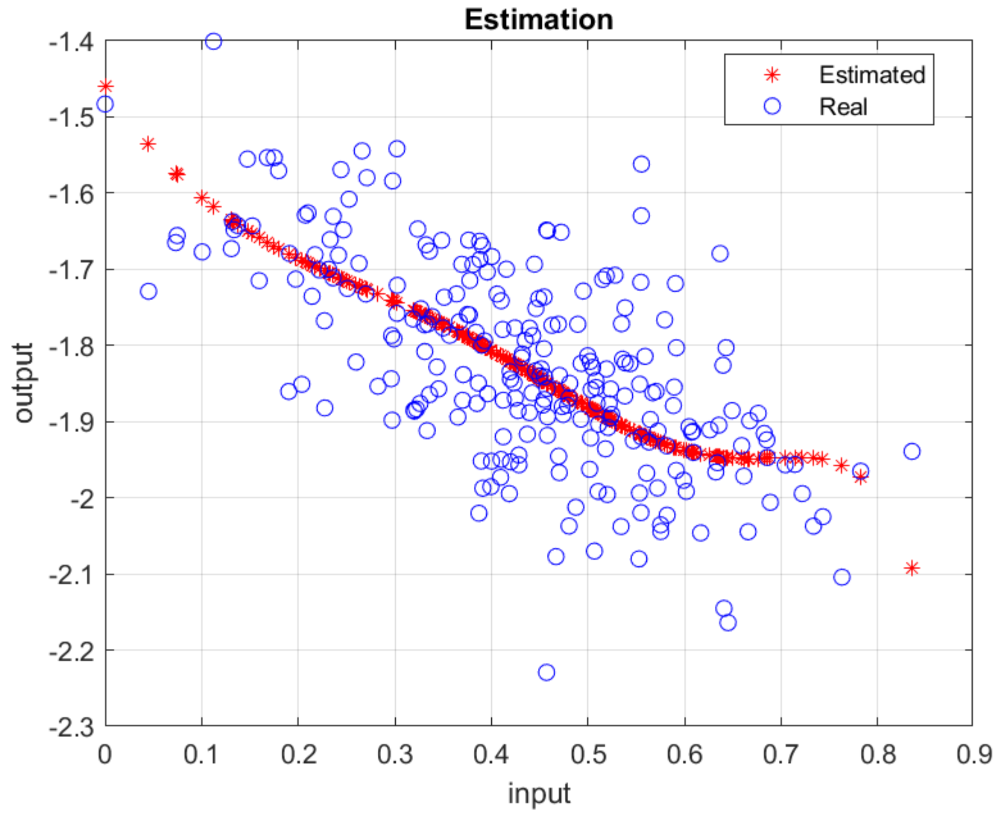

Estimation
Estimated
Real
output
input
-1.4
-1.5
-1.6
-1.7
-1.8
-1.9
-2
-2.1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9


Error Bar
Error Bar
input
-1.4
-1.5
-1.6
-1.7
-1.8
-1.9
-2
-2.1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

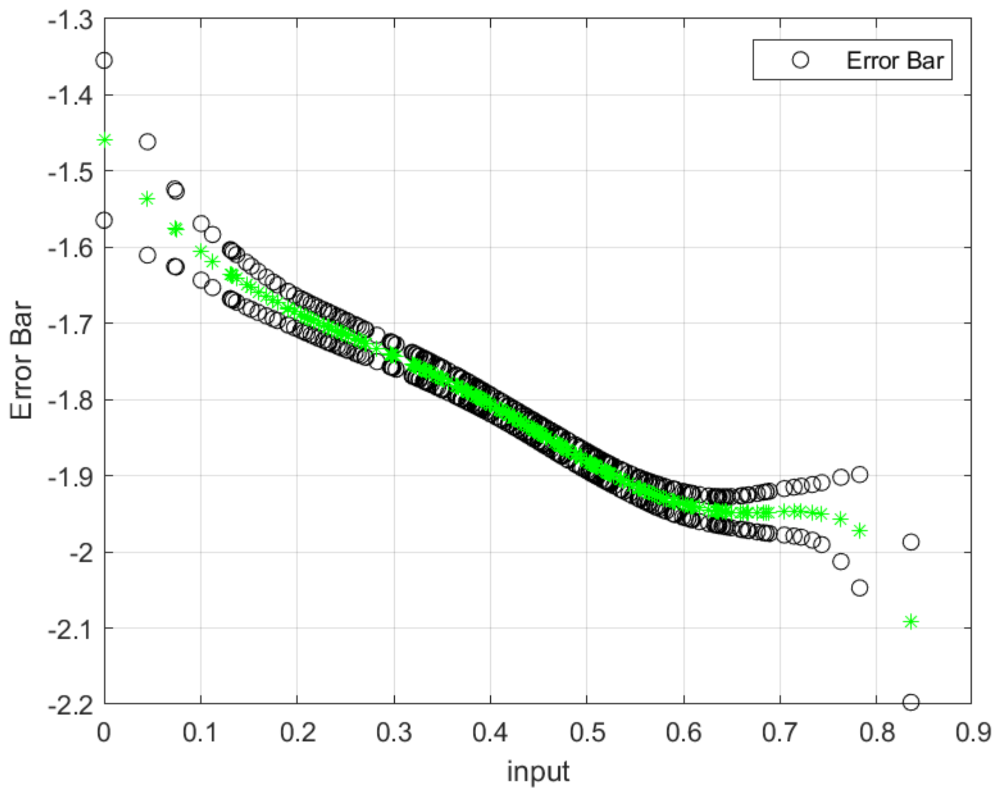



```
noise =

    'Medium noise'


Mean squared normalized error   =

   9.8750e-05
```

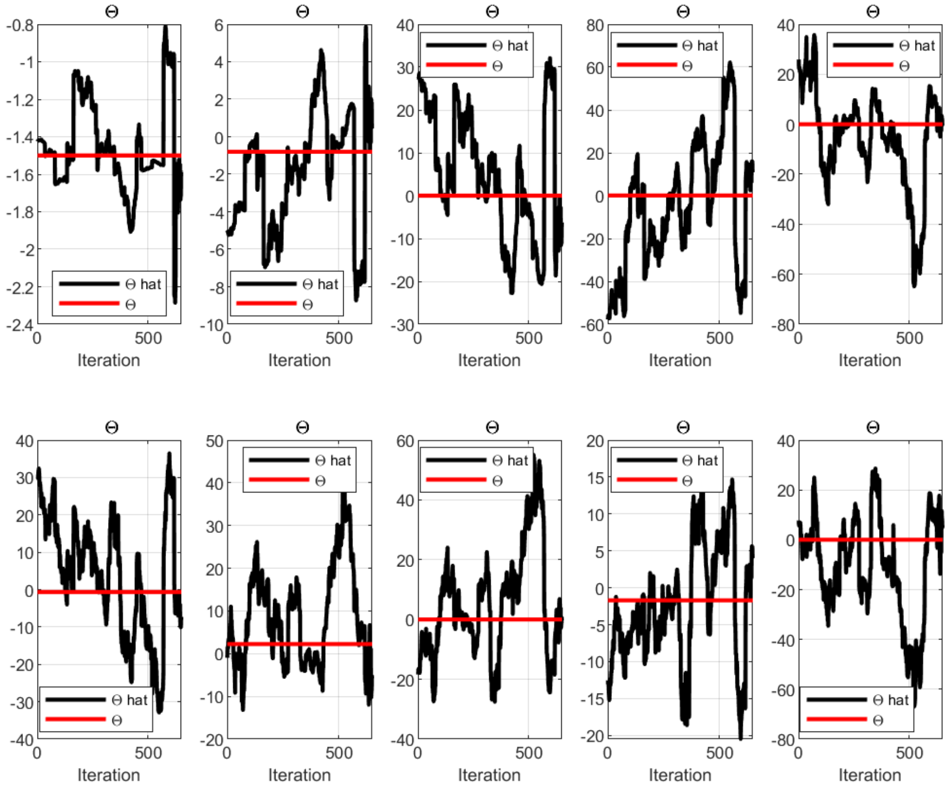
حالت سوم نویز زیاد



Estimation
Estimated
Real
output
input
-1.4
-1.5
-1.6
-1.7
-1.8
-1.9
-2
-2.1
-2.2
-2.3
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9



Error Bar
Error Bar
input
-1.3
-1.4
-1.5
-1.6
-1.7
-1.8
-1.9
-2
-2.1
-2.2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9




```
noise =

    'High noise'

Mean squared normalized error  =

    0.0107
```

نتیجه گیری

مقدار تتا تخمین زده شده با لغزش پنجره در هر مرحله تغییر میکند اما میانگین تتا بدست امده نزدیک به مقدار تتا واقعی است. البته چون ما از ترم Regularization استفاده کرده ایم یک مقدار ناچیزی بایاس در حالت اول که مقدار نویز کم است بر روی مقادیر تتا تخمین زده شده وجود دارد اما این مقدار ناچیز خطا در این حالت قابل قبول است زیرا اگر ترم Regularization وجود نداشته باشد زمانی که نویز زیاد میشود پارامتر های تتا تخمین زده شده بسیار بزرگ میشوند.

همچنین تتا های بدست امده مانند حالت قبلی ثابت نیستند و با لغزش پنجره به سمت جلو و امدن داده های جدید به این الگوریتم تتا های جدید تخمین زده میشوند. با توجه به اینکه نویز برای هر پنجره متفاوت است و محدود بودن طول پنجره نتایج تتا های بدست امده در هر مرحله متفاوت اند

در این سوال اگر از پنجره های کوچک تر استفاده کنیم نتایج به دست امده بدتر میشود و تخمین به دست امده دقت کمتری دارد چون از داده های کمتری استفاده میکند اما دارای سرعت بیشتری هستند و به صورت انلاین میتوانیم از این روش استفاده کنیم و بلعکس هر چه طول پنجره بیشتر باشد نتایج خروجی دقت بیشتری دارند اما مدت زمان بیشتری برای حل نیاز دارند

سوال 1-4

روش حداقل مربعات وزن دار شده به این صورت است که اگر در بین دادههای آموزش، دادههایی وجود داشته باشد که قابل اطمینان نیستند یا کم ارزش اند، با در نظر گرفتن این داده ها و حداقل کردن اهمیت آن ها با دادن وزن کمتر و کمتر کردن خطا می توان به جواب بهتری رسید.

برای انتخاب وزن های مناسب دو معیار زیر در نظر گرفته شده اند

حالت اول

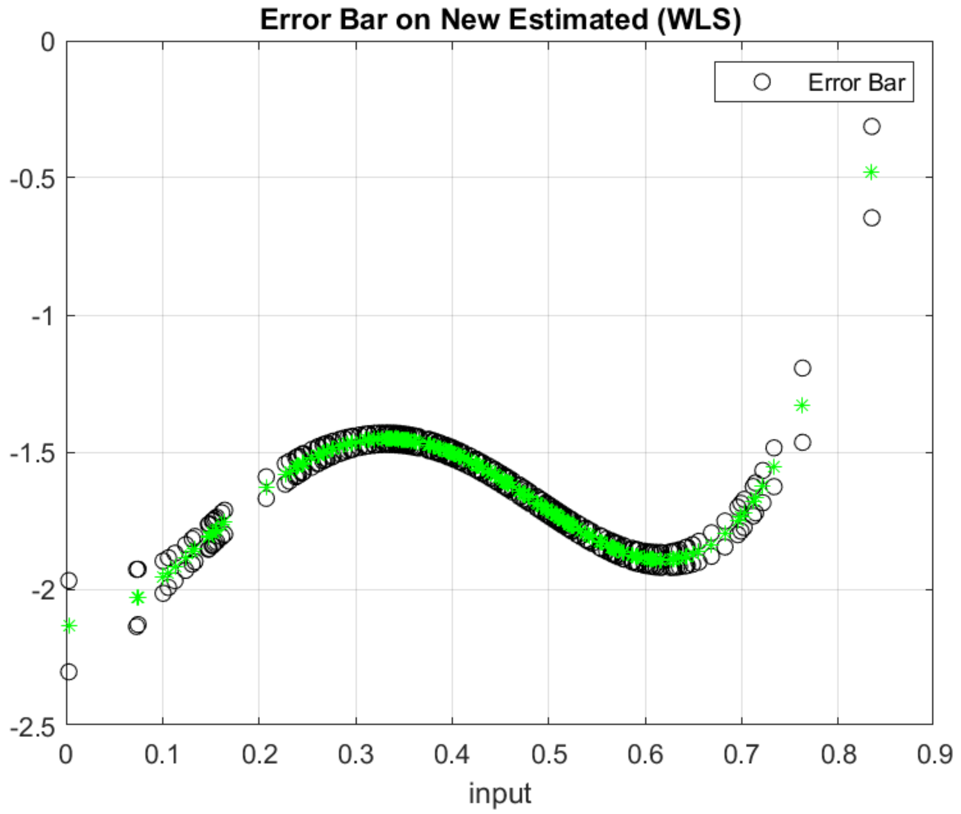
در این حالت میخواهیم با استفاده از وزن مناسب P تاثیر داده های ورودی پرت را کمتر کنیم برای این کار در ابتدا به 10 داده اخر هر دسته در مقدار ورودی مقداری بایاس به صورت رندوم بین صفر تا 5 و به مقدار خروجی متناظر با ان مقداری بایاس به صورت رندون بین صفر تا 10 میدهیم در این صورت ده داده اخر ارزش کمتری دارند و باید در محاسبات وزن کوچیکتری داشته باشند برای این کار وزن را به صورت زیر تعریف میکنیم

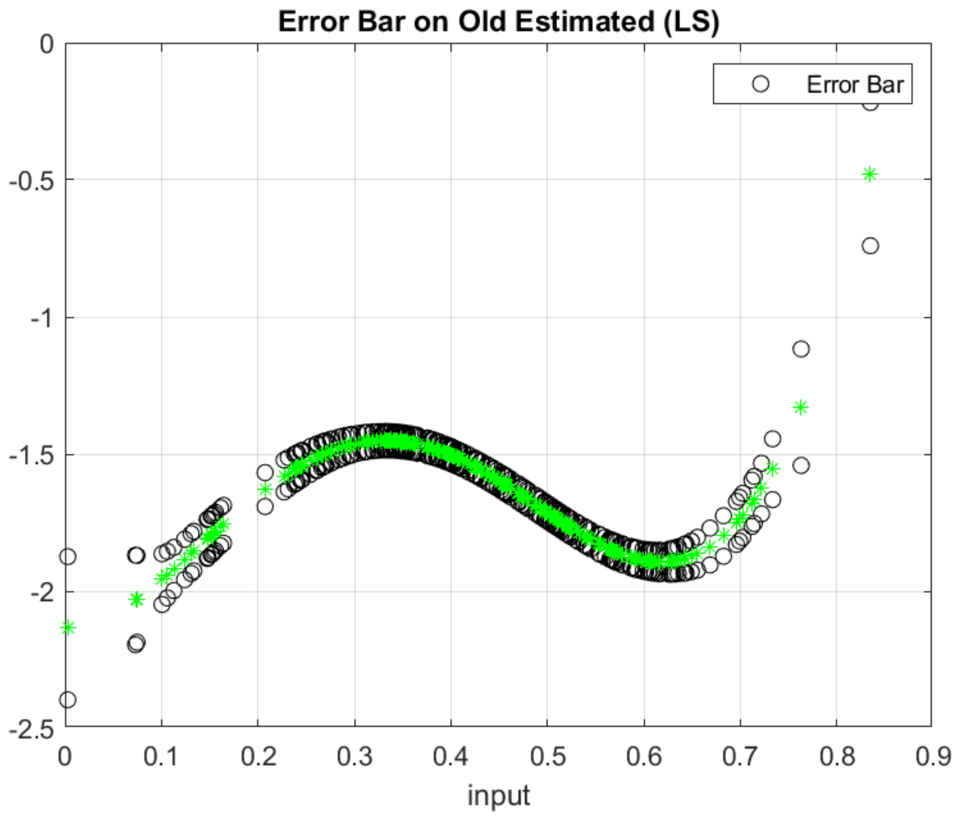
$$P(i,i) = exp(-1 * norm(\mathrm{xtrain(i,:)} - \mathrm{meanX}))$$

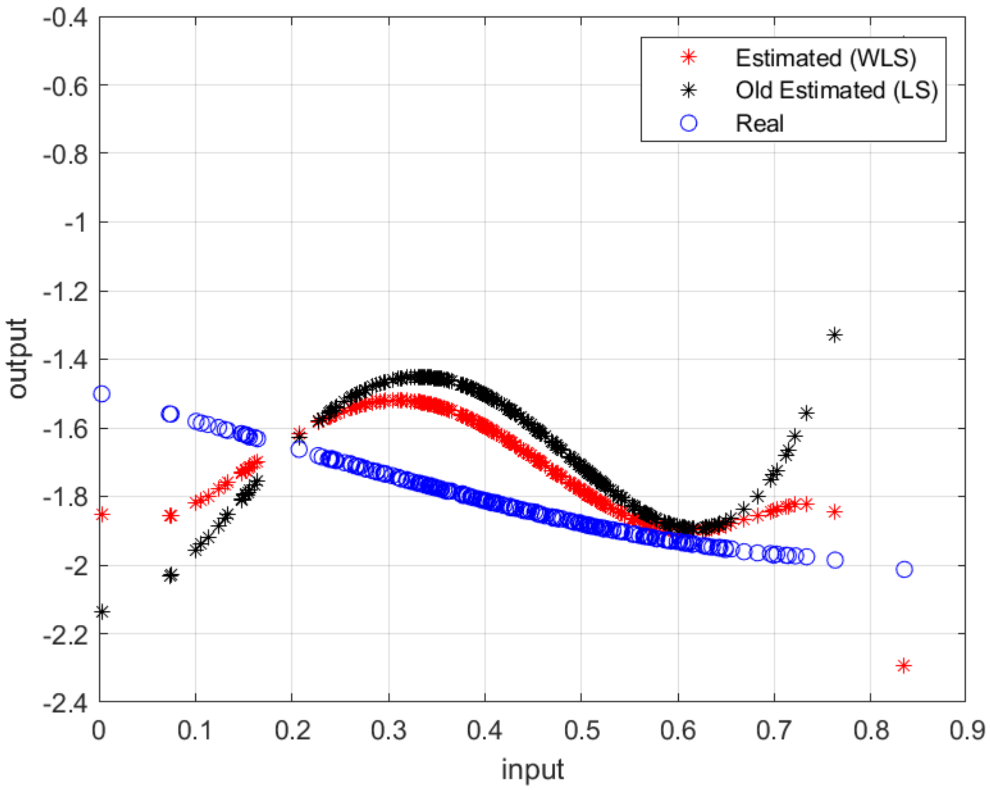
با این وزن ها داده هایی که از میانگین فاصله دارند مقدار وزن کمتر در تخمین پیدا میکنند و داده هایی که شبیه به میانگین هستند در فرایند تخمین بدون تغییر باقی میمانند

نتایج شبیه سازی ها به صورت زیر میباشد به ازای همه نویز های مختلف روش WLS نسبت به روش LS معمولی نتایج بهتری بدست میاورد.

حالت اول نویز کم

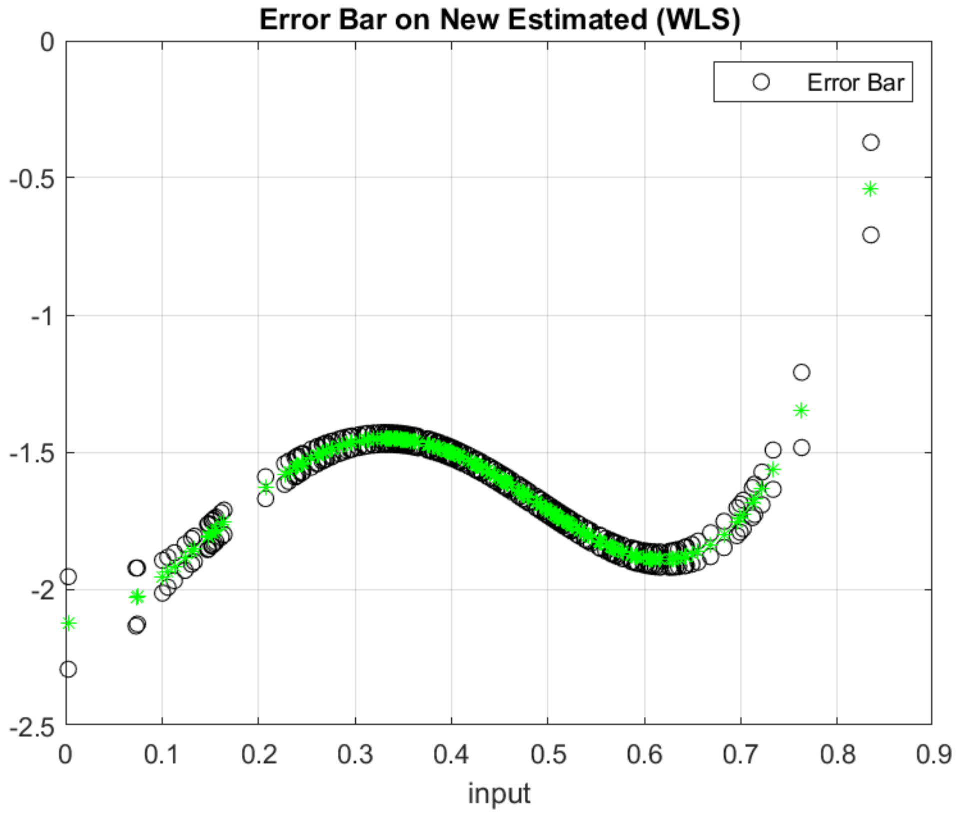

Error Bar on New Estimated (WLS)
Error Bar
0
-0.5
-1
-1.5
-2
-2.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
input

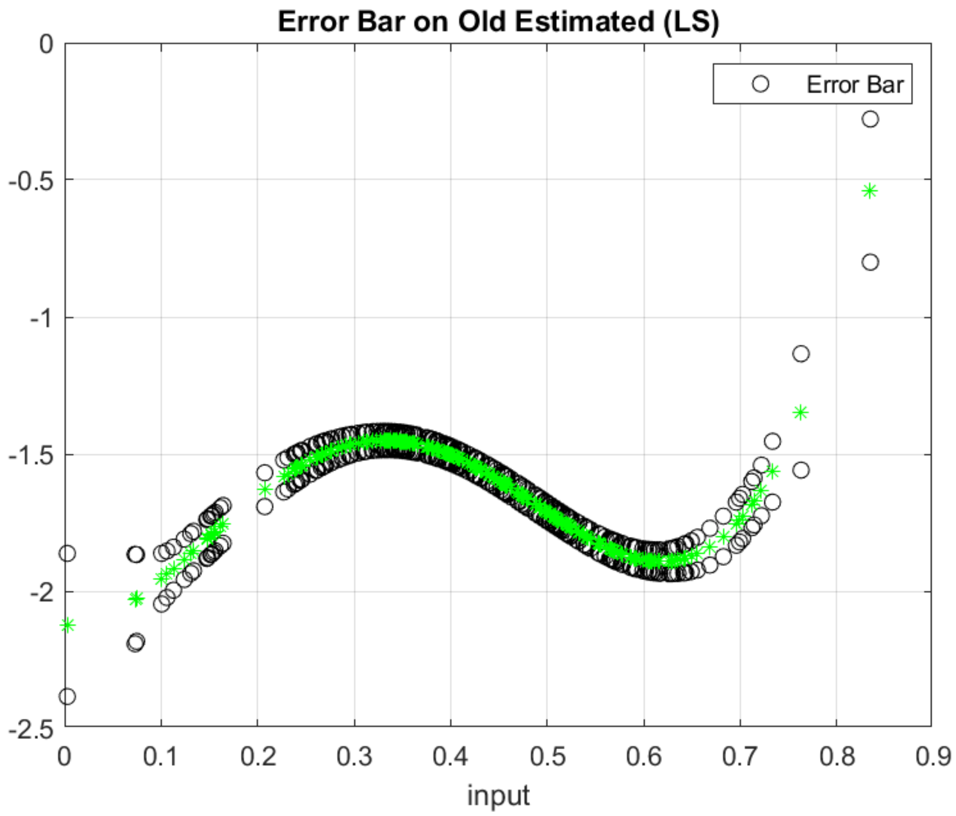

Error Bar on Old Estimated (LS)
Error Bar
0
-0.5
-1
-1.5
-2
-2.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
input

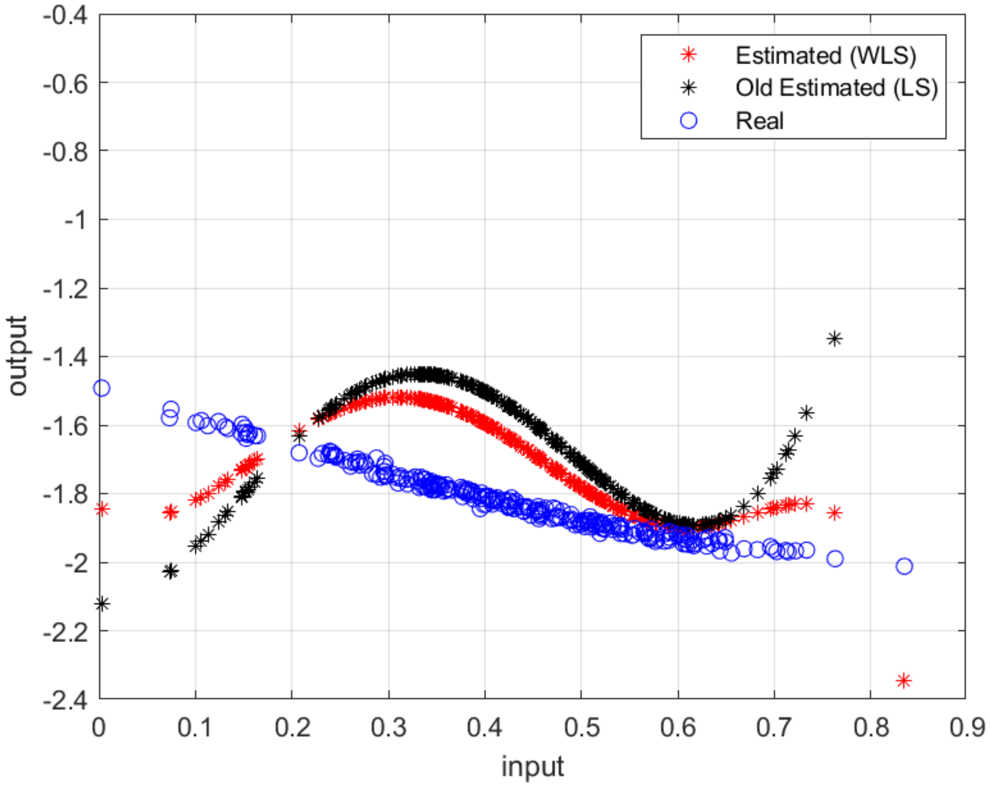



```
noise =

    'Low noise'


Mean squared normalized old error (LS)

    0.0656


Mean squared normalized new error (WLS)

    0.0267
```

حالت دوم نویز متوسط

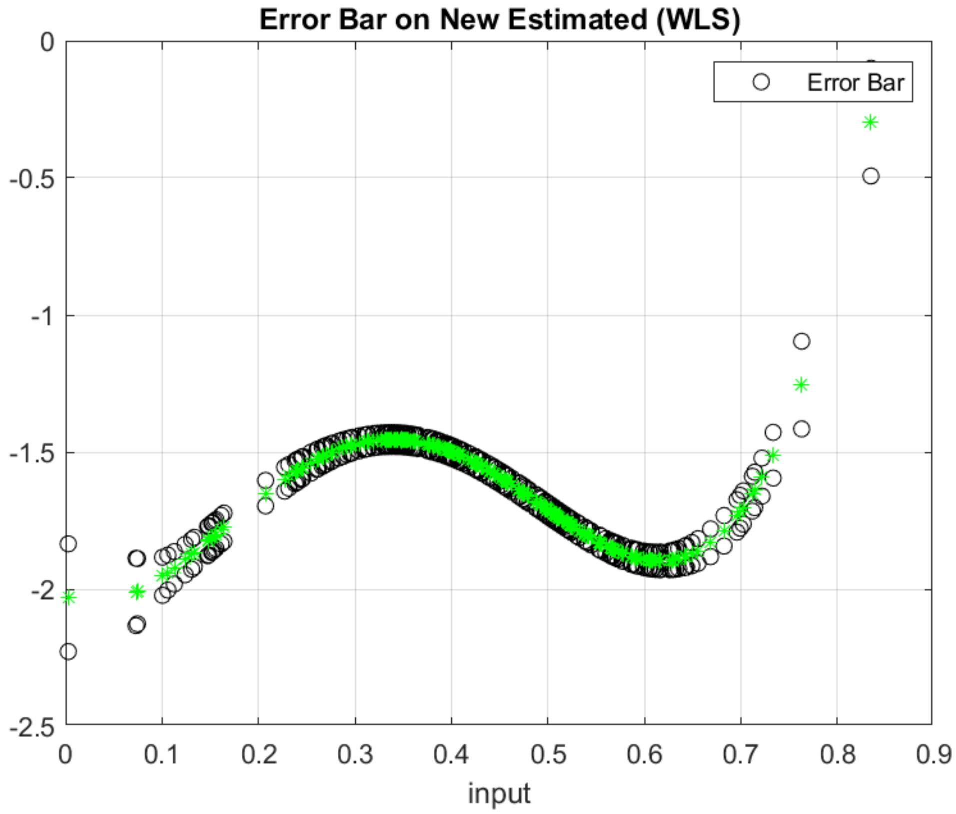

Error Bar on New Estimated (WLS)
Error Bar
input
0
-0.5
-1
-1.5
-2
-2.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

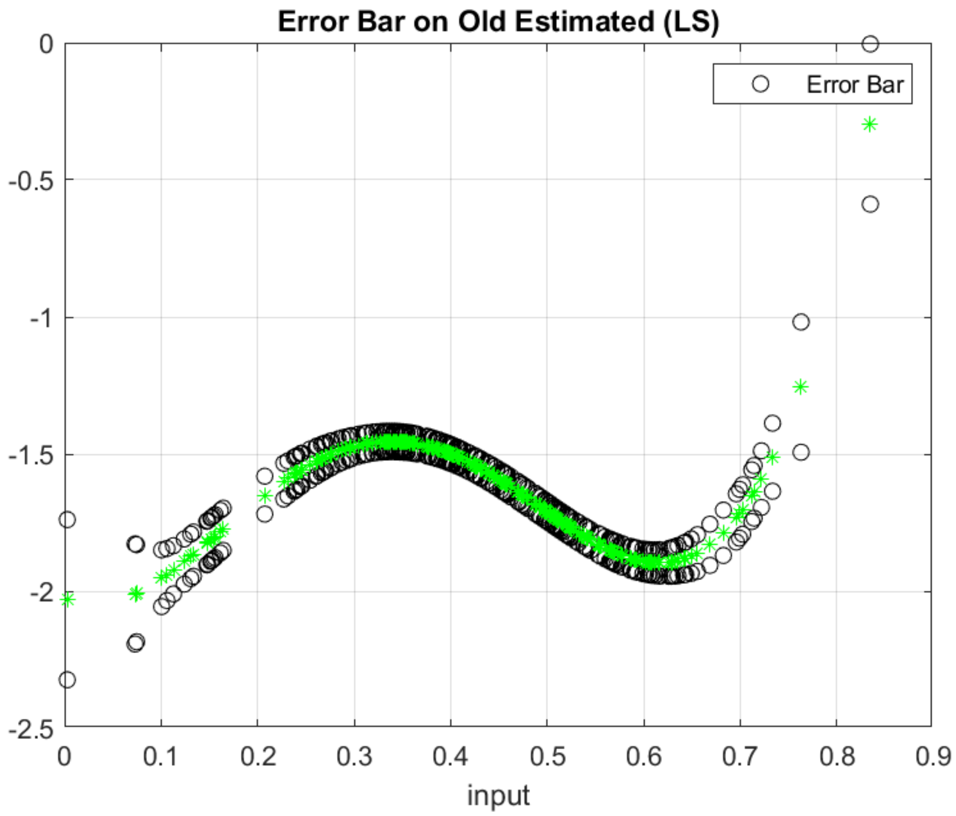

Error Bar on Old Estimated (LS)
Error Bar
input
0
-0.5
-1
-1.5
-2
-2.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

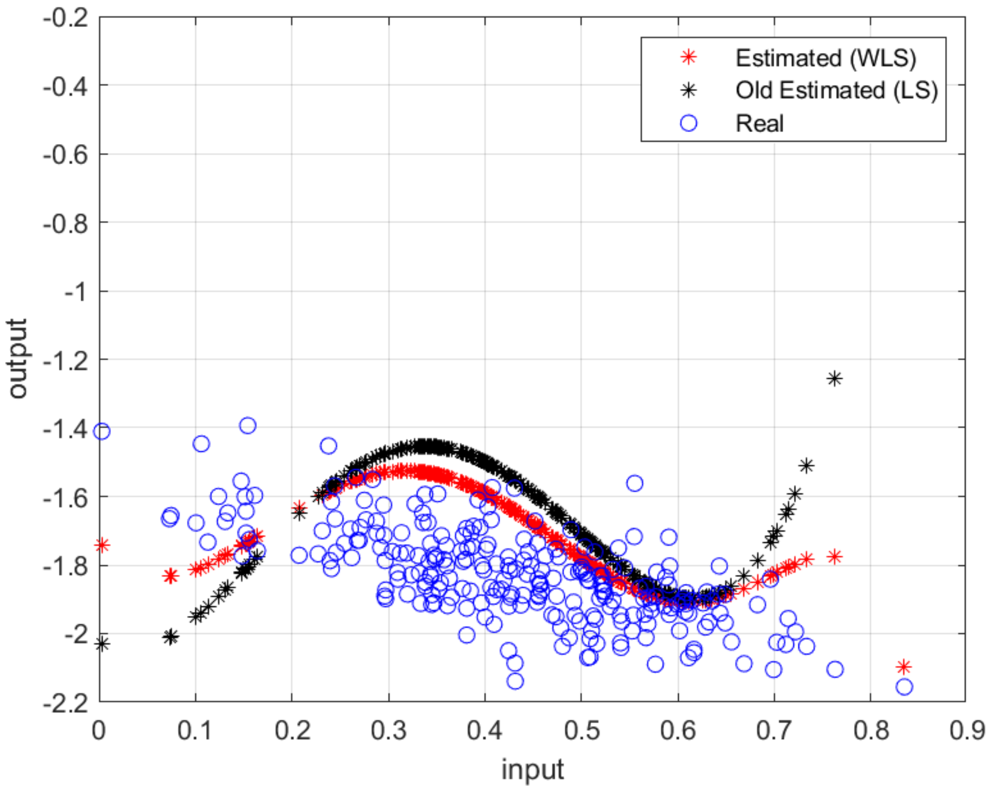



```
noise =

    'Medium noise'


Mean squared normalized old error (LS)

    0.0655


Mean squared normalized new error (WLS)

    0.0274
```

حالت سوم نویز زیاد


Error Bar on New Estimated (WLS)
Error Bar
0
-0.5
-1
-1.5
-2
-2.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
input


Error Bar on Old Estimated (LS)
Error Bar
0
-0.5
-1
-1.5
-2
-2.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
input




```
noise =

    'High noise'


Mean squared normalized old error (LS)

    0.0821


Mean squared normalized new error (WLS)

    0.0371
```

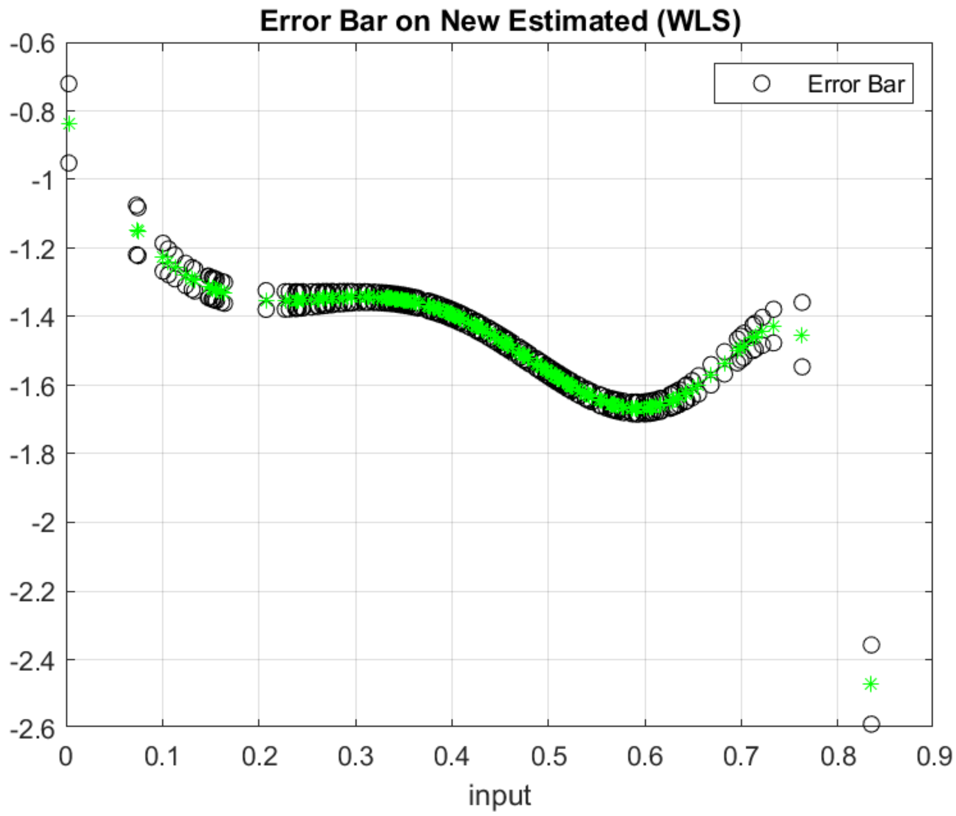
حالت دوم

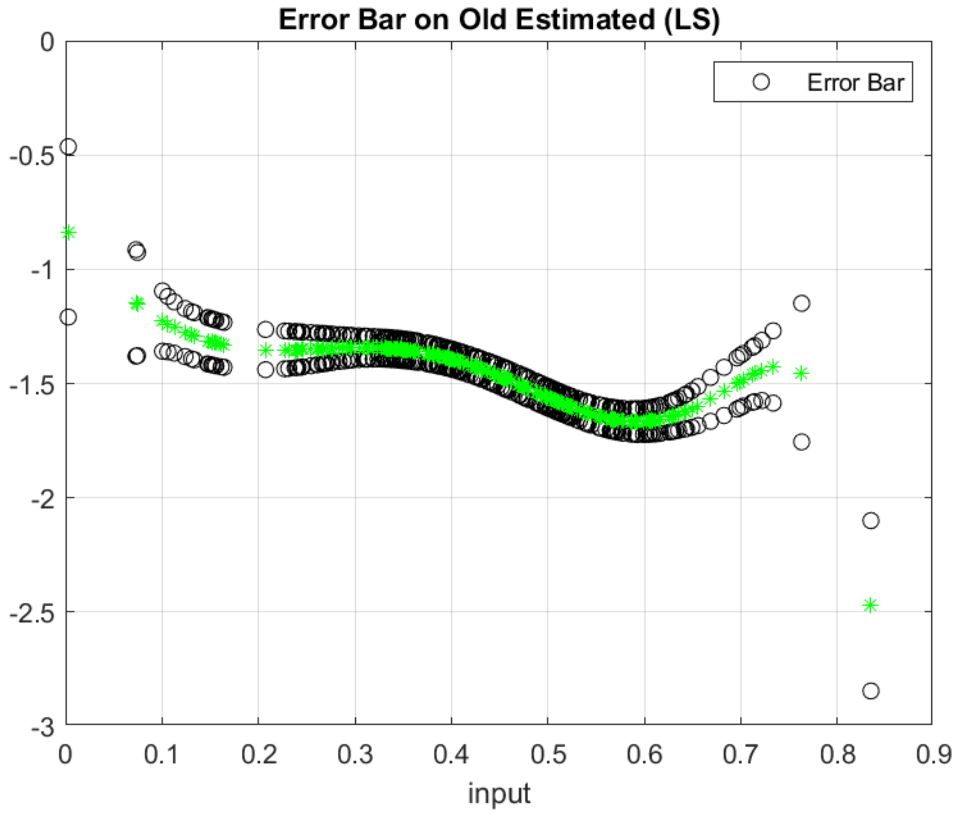
در این حالت با استفاده از خطا LS ساده به انتخاب وزن های هر مرحله میپردازیم به این ترتیب که خطا متناظر برای هر داده را با استفاده از فرمول زیر در وزن مربوط به ان داده تاثیر میدهیم

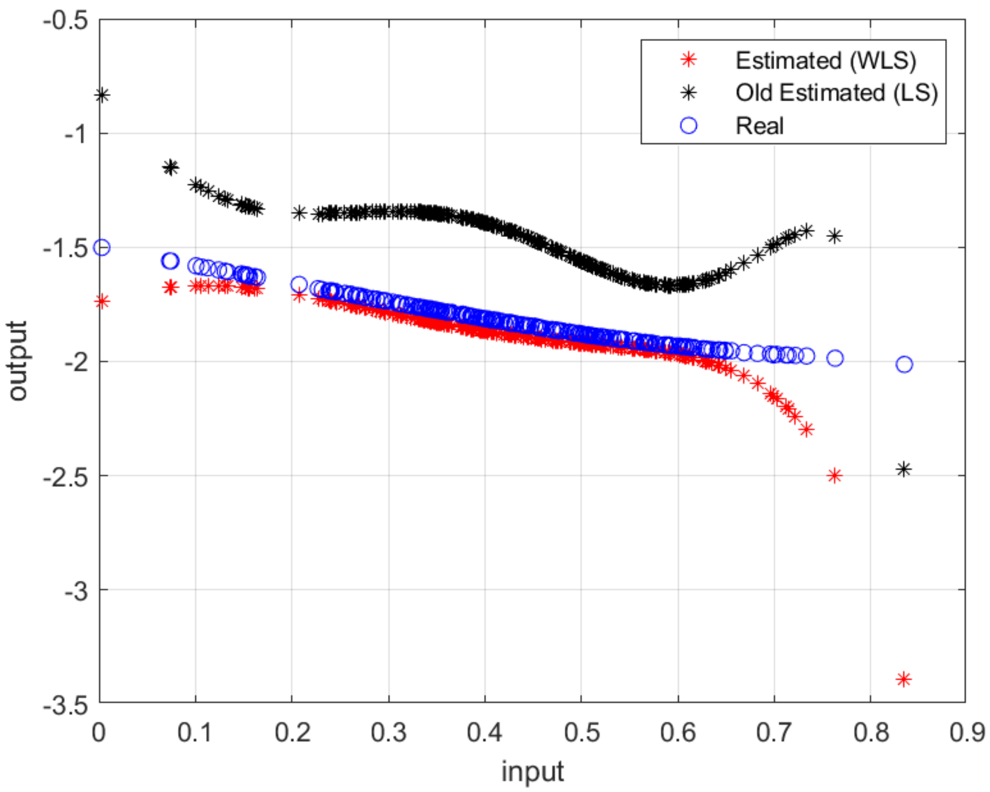
```
Q(r,r)=1/(1+10*e2(r,:))
```

نتایج شبیه سازی ها به صورت زیر میباشد به ازای همه نویز های مختلف روش WLS نسبت به روش LS معمولی نتایج بهتری بدست میاورد. و این نشان دهنده کارامد بودن وزن های پیشنهاد داده شده است

حالت اول نویز کم

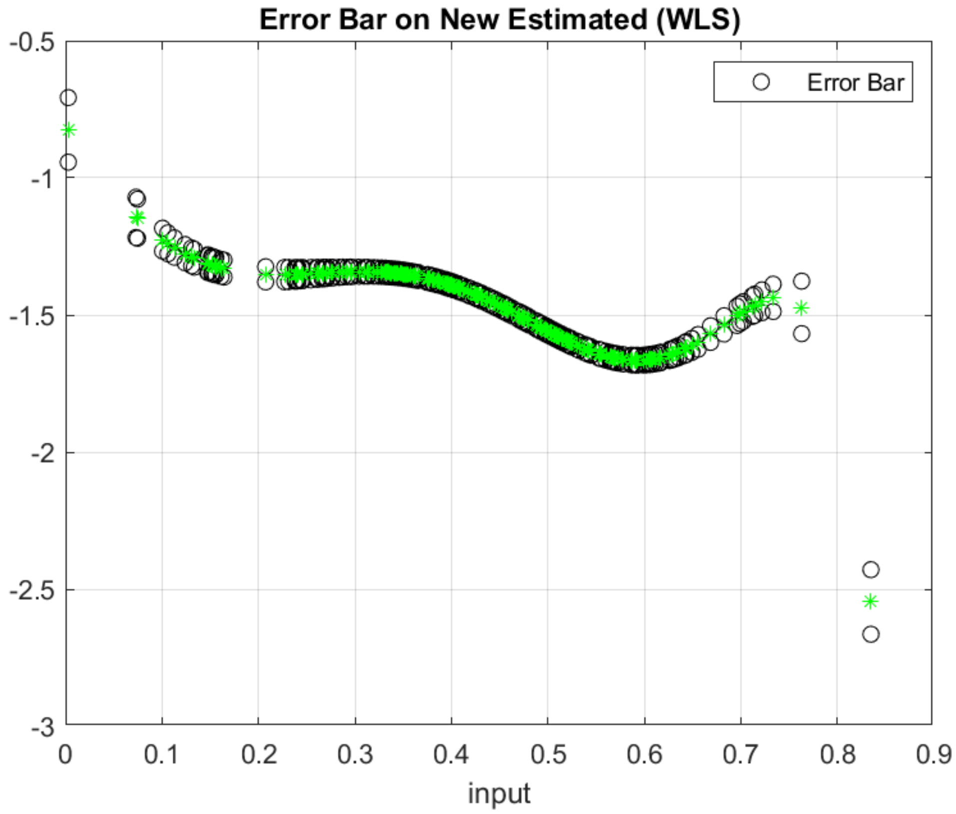


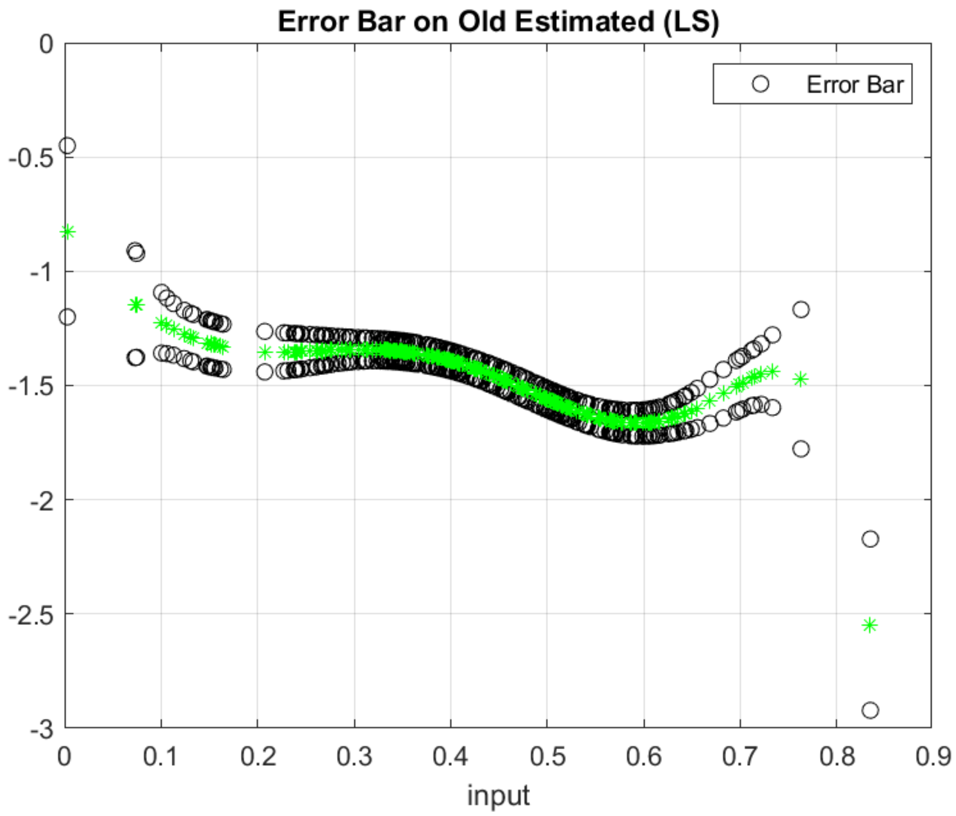

Error Bar on Old Estimated (LS)
Error Bar
input

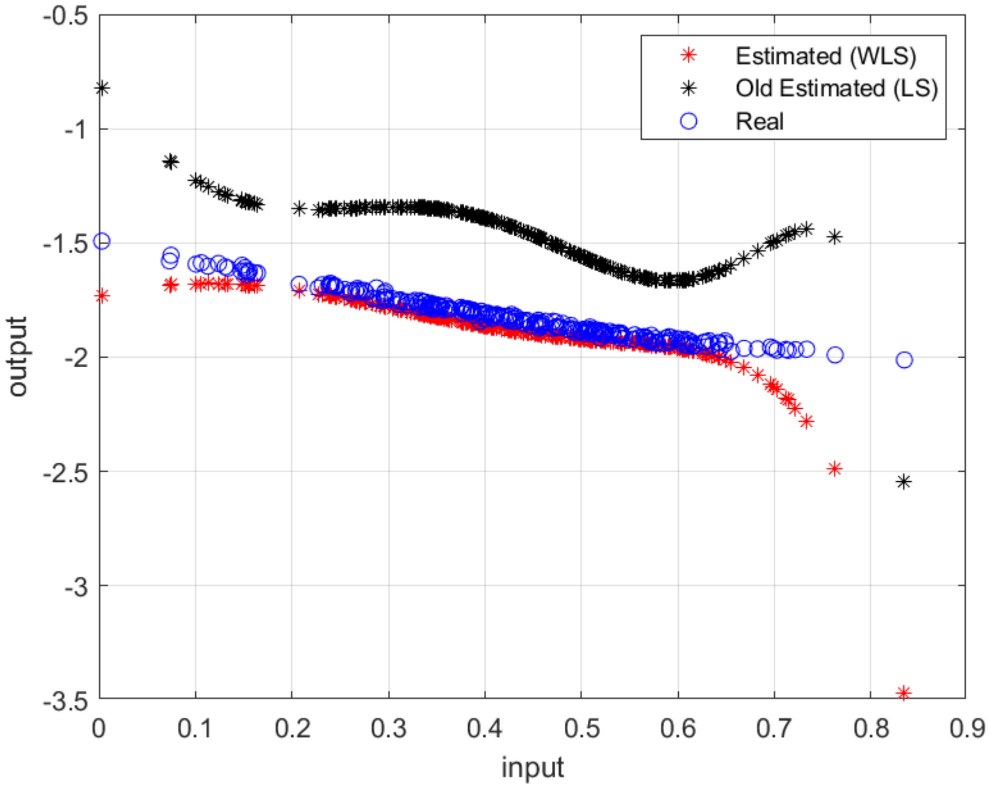

Estimated (WLS)
Old Estimated (LS)
Real
output
input

```
noise =

    'Low noise'


Mean squared normalized old error (LS)

    0.1337


Mean squared normalized new error (WLS)

    0.0128
```

حالت دوم نویز متوسط

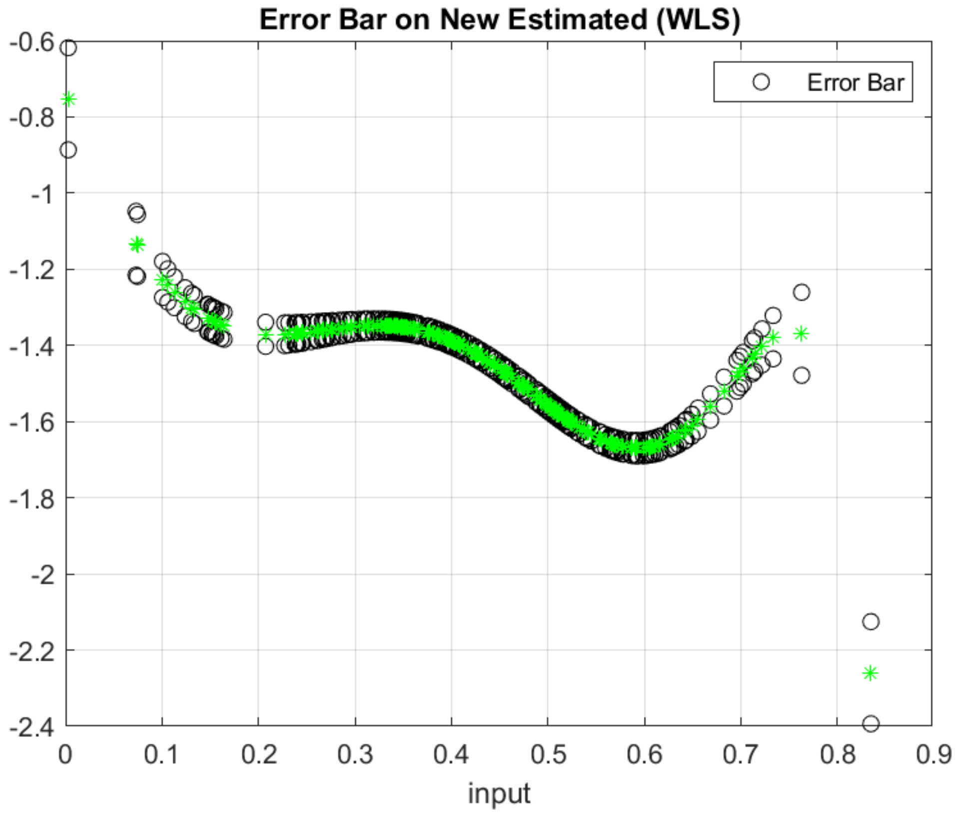

Error Bar on New Estimated (WLS)
Error Bar
input
-0.5
-1
-1.5
-2
-2.5
-3
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

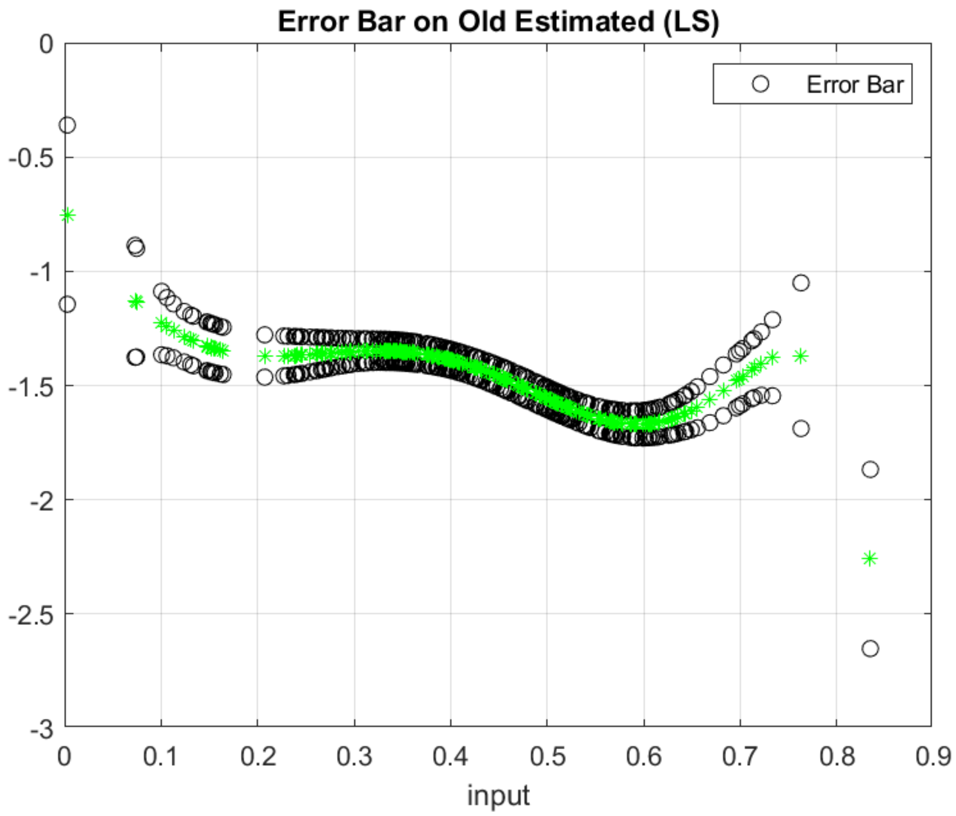

Error Bar on Old Estimated (LS)
Error Bar
input
0
-0.5
-1
-1.5
-2
-2.5
-3
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

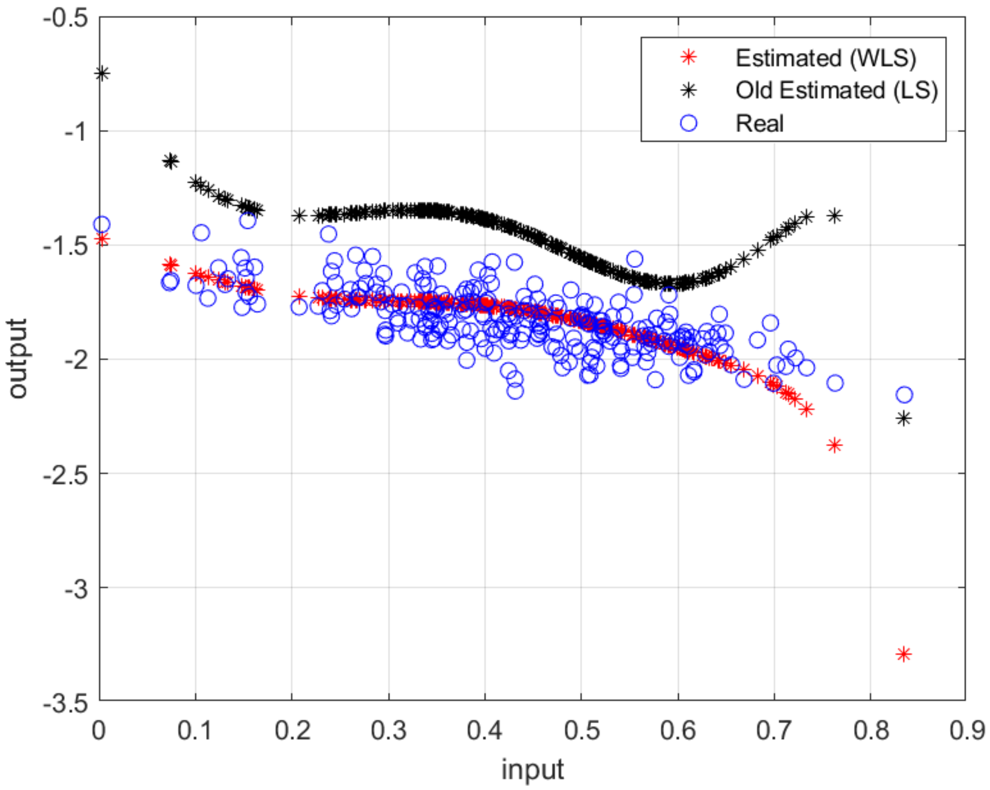



```
noise =

    'Medium noise'


Mean squared normalized old error (LS)

    0.1352


Mean squared normalized new error (WLS)

    0.0133
```

حالت سوم نویز زیاد


Error Bar on New Estimated (WLS)
Error Bar
-0.6
-0.8
-1
-1.2
-1.4
-1.6
-1.8
-2
-2.2
-2.4
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
input


Error Bar on Old Estimated (LS)
Error Bar
0
-0.5
-1
-1.5
-2
-2.5
-3
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
input



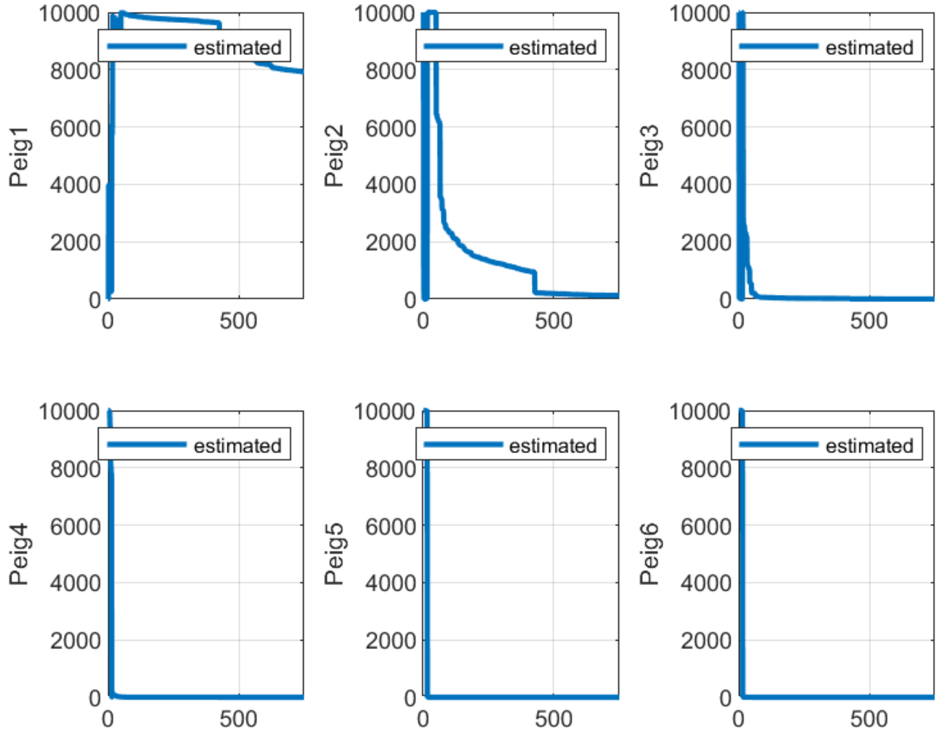

```
noise =

    'High noise'


Mean squared normalized old error (LS)

    0.1479


Mean squared normalized new error (WLS)

    0.0173
```

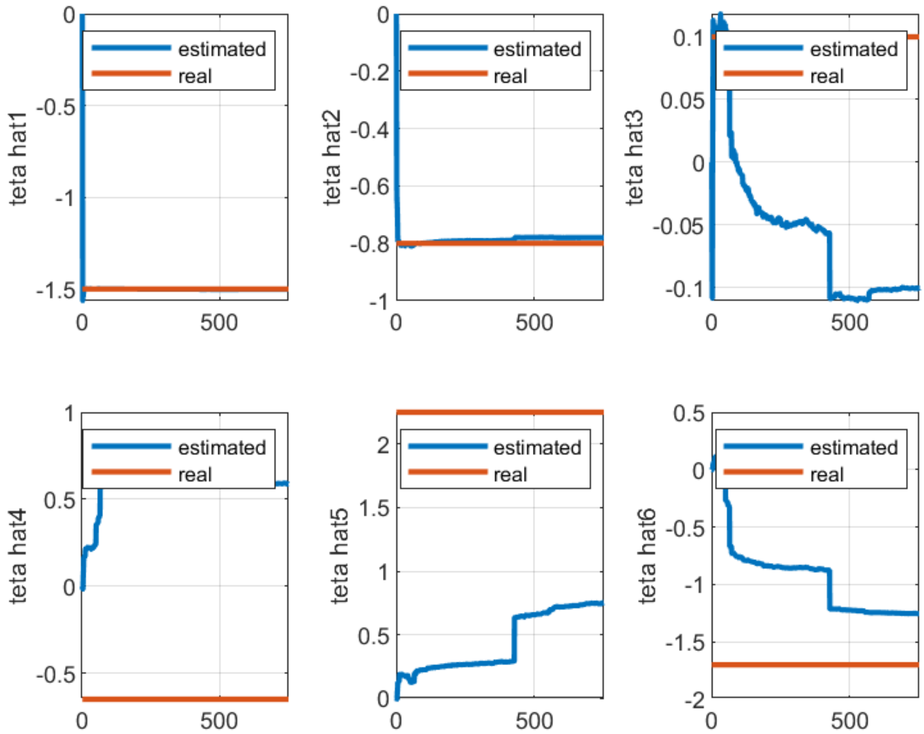
سوال 1-5

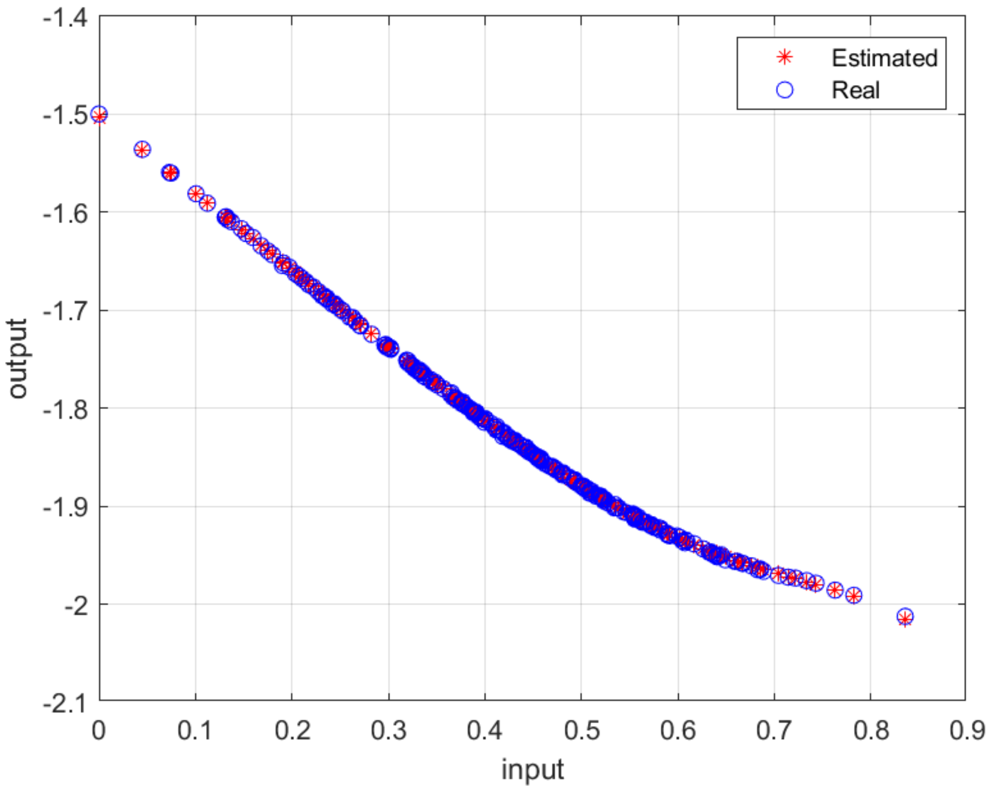
در این سوال از الگوریتم RLS استفاده شده است

در حالتی که کوواریانس نویز کمتر است الگوریتم نتایج بهتری داشته است و پارامتر های تخمین زده شده تغییرات کمتری داشته اند اما در حالتی که کوواریانس نویز زیاد است تغییرات پارامتر های تخمین زده شده بیشتر میشود

حالت اول نویز کم

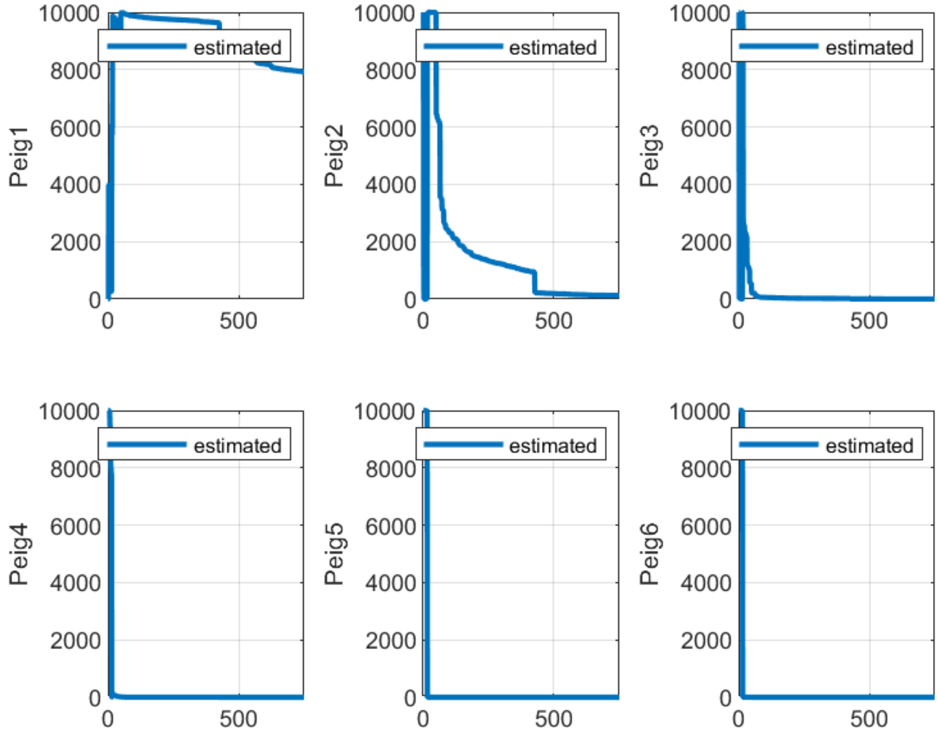

Peig1
Peig2
Peig3
Peig4
Peig5
Peig6
estimated
10000
8000
6000
4000
2000
0
500

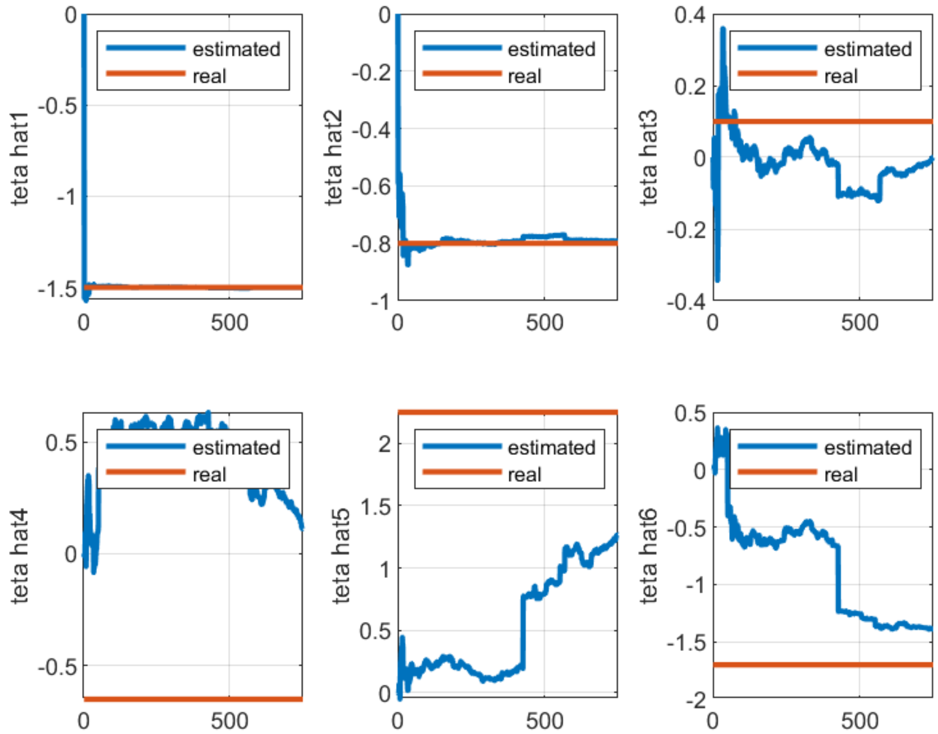

teta hat1
teta hat2
teta hat3
teta hat4
teta hat5
teta hat6
estimated
real
500

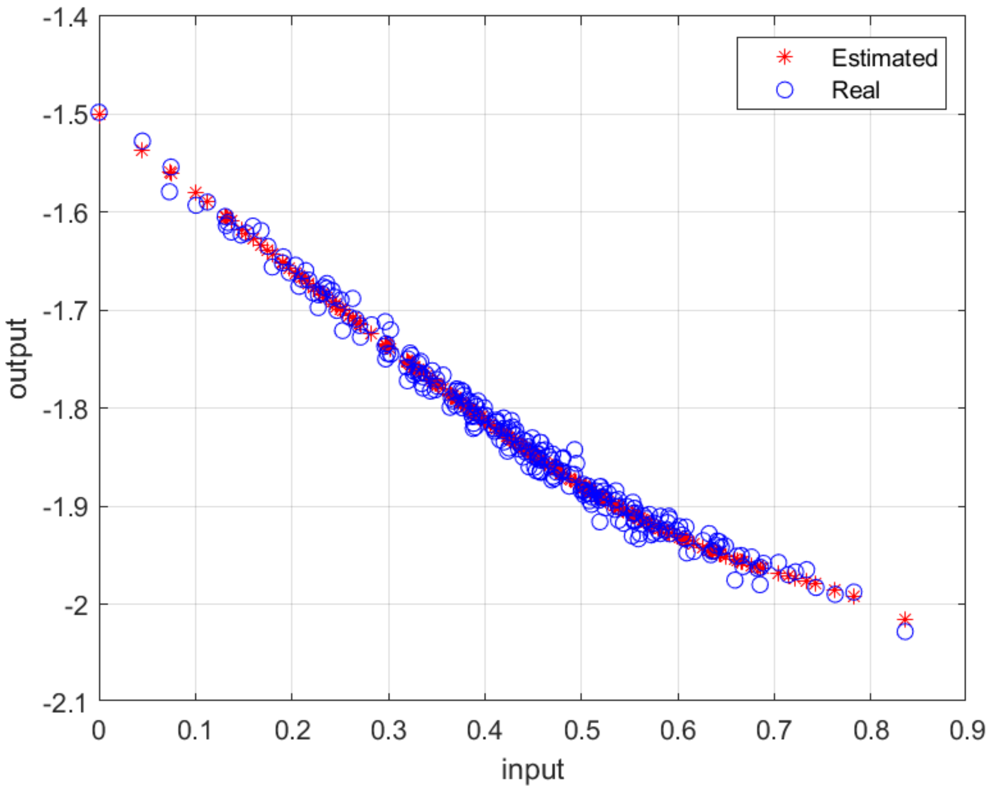



```
Mean squared normalized error  =

   1.3669e-06
```

حالت دوم نویز متوسط

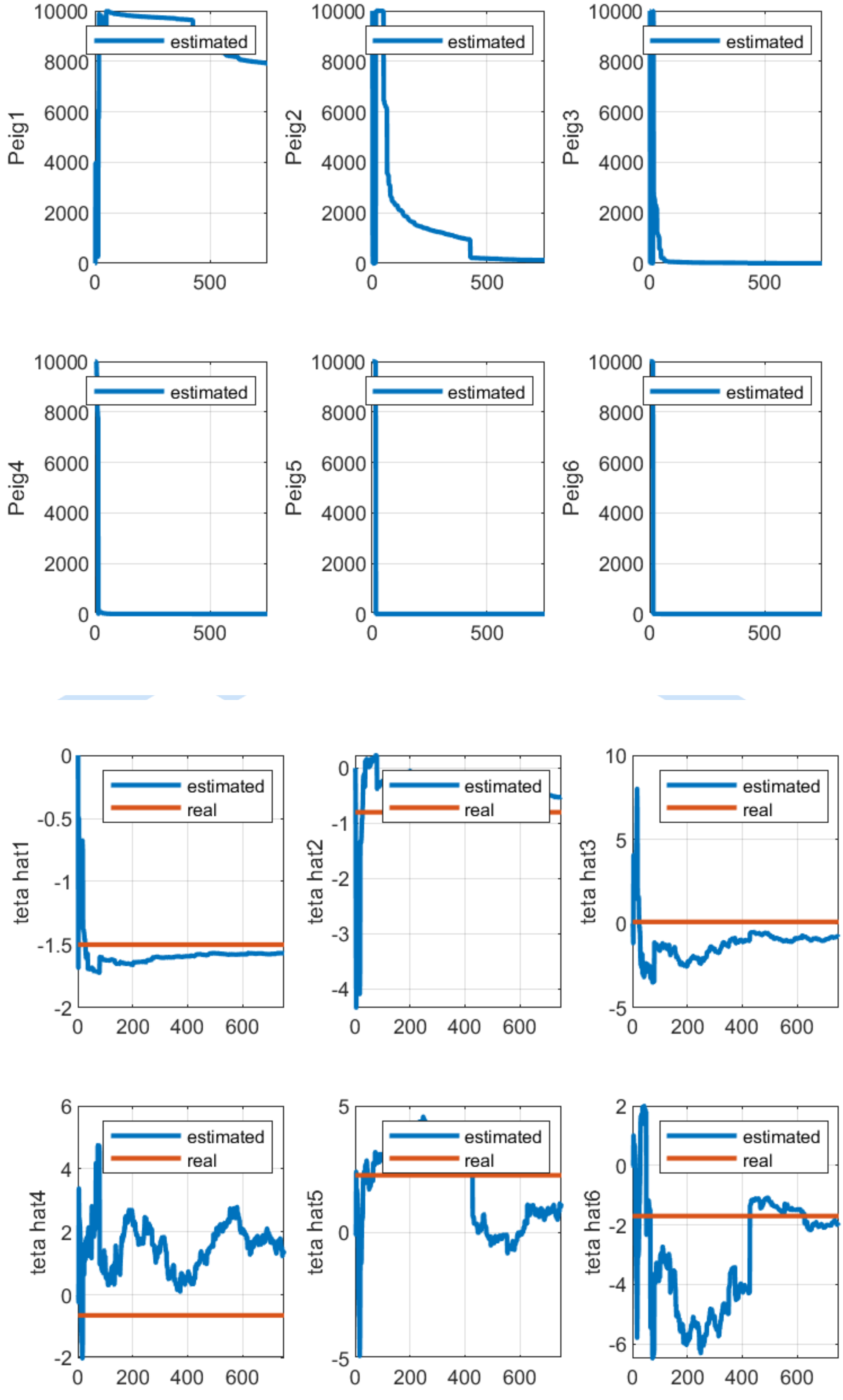

Peig1
Peig2
Peig3
Peig4
Peig5
Peig6
estimated


teta hat1
teta hat2
teta hat3
teta hat4
teta hat5
teta hat6
estimated
real

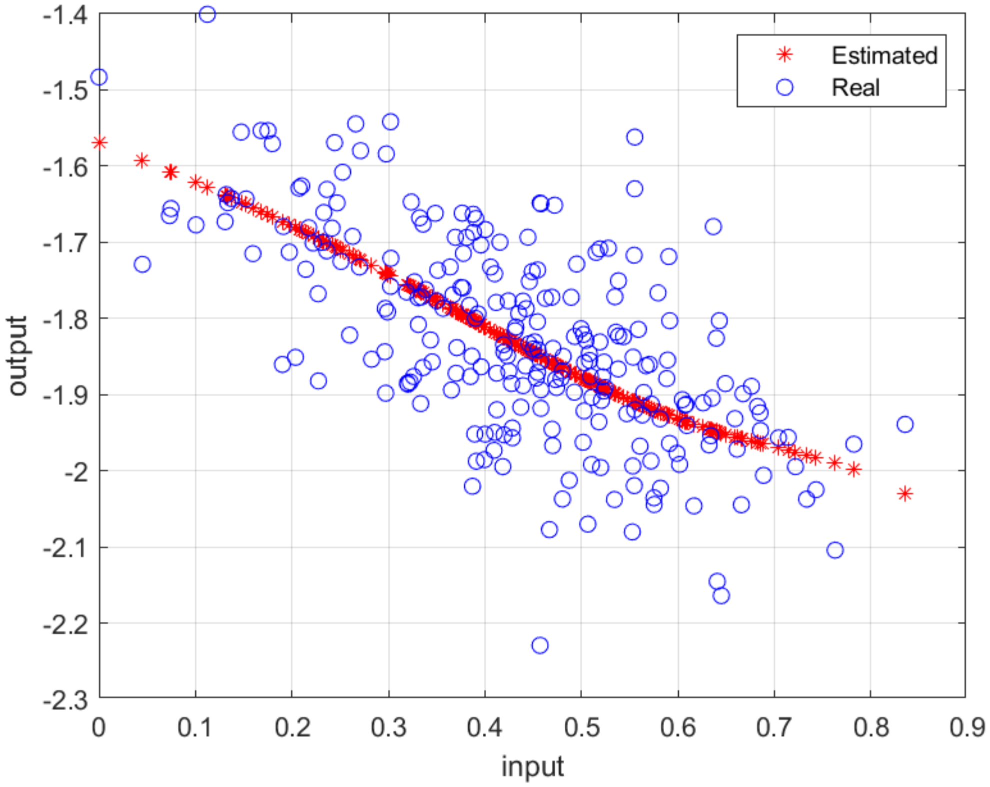



```
noise =

    'Medium noise'


Mean squared normalized error  =

   9.6095e-05
```

حالت سوم نویز زیاد



Peig1
Peig2
Peig3
Peig4
Peig5
Peig6
estimated
teta hat1
teta hat2
teta hat3
teta hat4
teta hat5
teta hat6
estimated
real



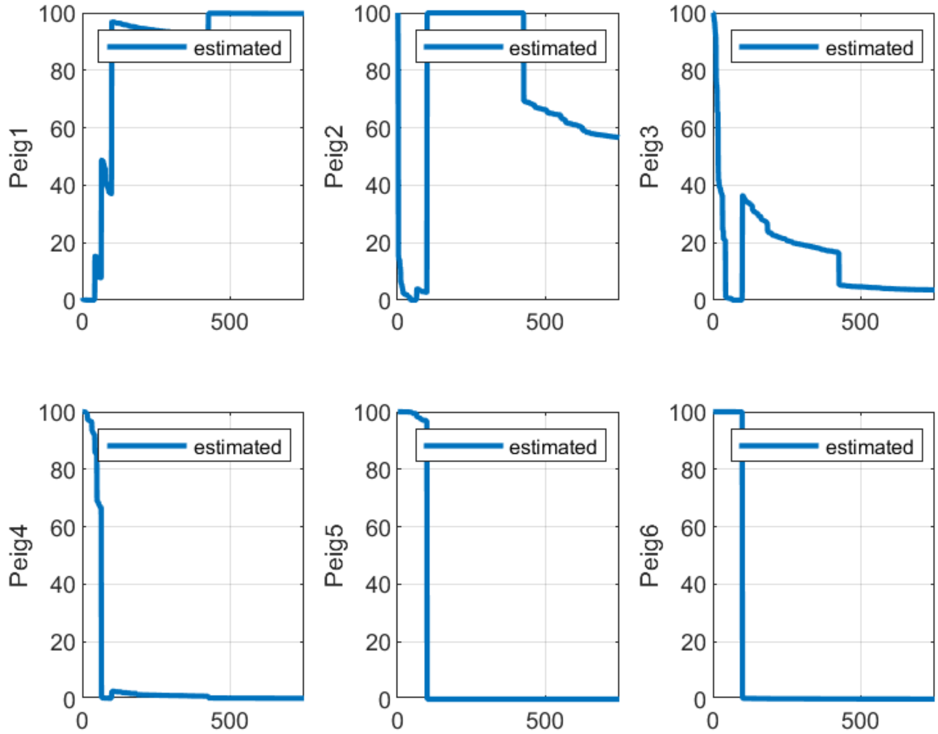

```
noise =

    'High noise'


Mean squared normalized error   =

    0.0104
```

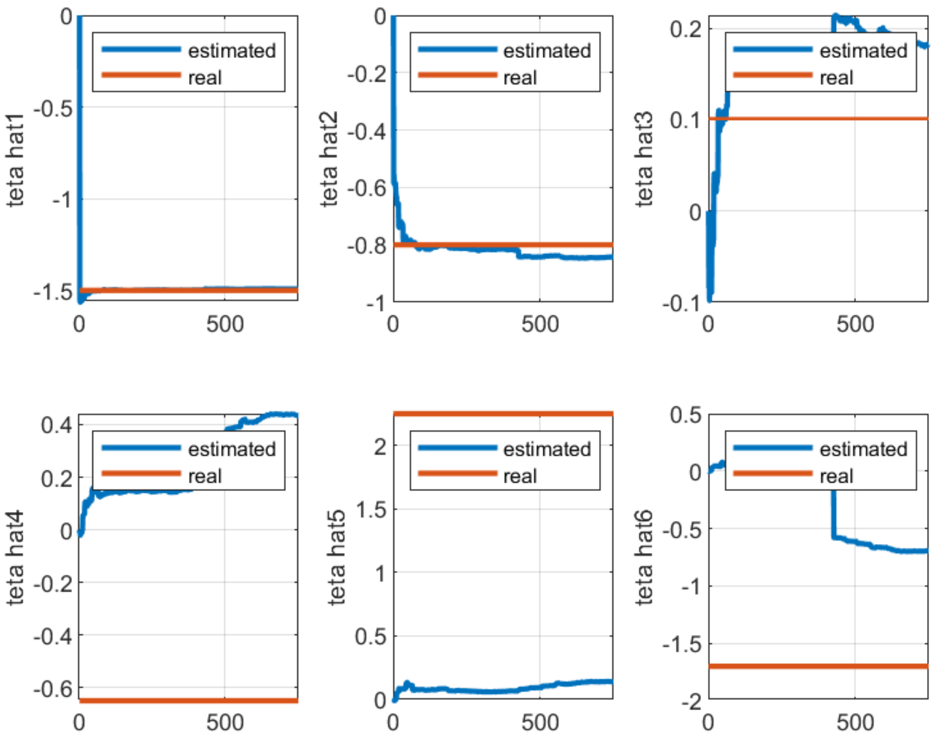
برای نویز متوسط تغیرات ماتریس کوواریانس رو برای حالت های مختلف بررسی میکنیم

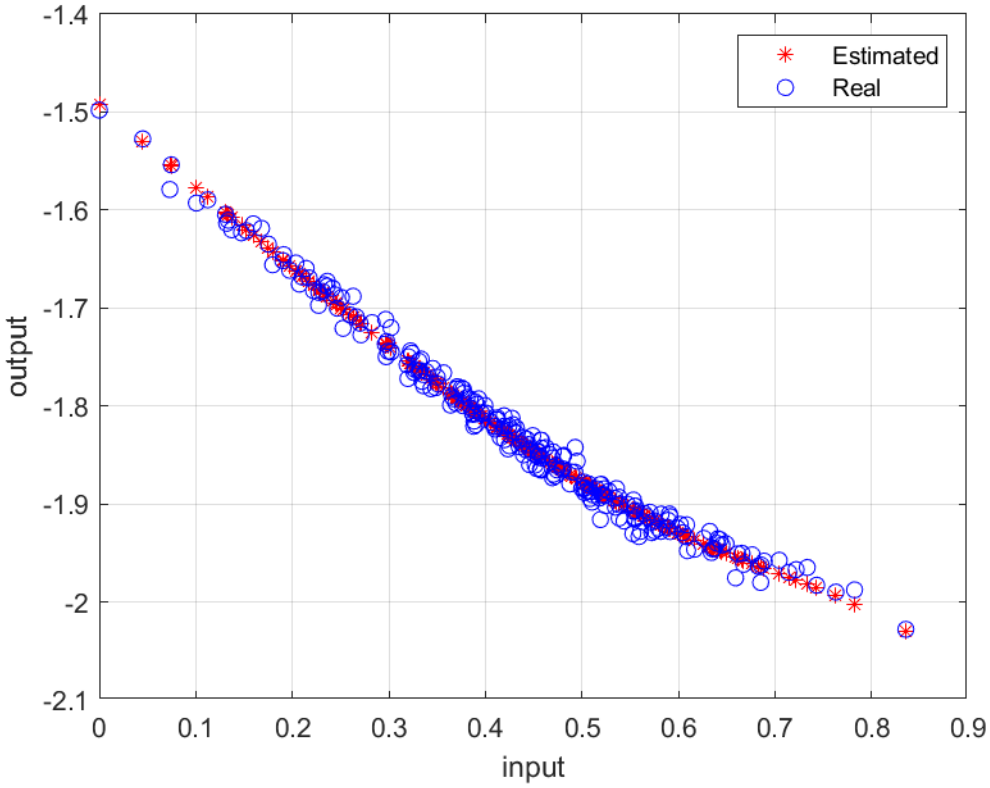
هر چه مقدار P بیشتر شود مقادیر تتا و مقادیر ویژه ماتریس P سریعتر همگرا میشود و همچنین مقدار تغییرات تتا بیشتر و تغییرات مقادیر ویژه ماتریس P بزرگتر میشود.

در این سوال بنده کد خواسته شده در صورت سوال را پیاده سازی کرده ام اما در این سوال تعداد داده ها کم میباشد در نتیجه تتا های تخمین زده شده همگرایی خوبی به مقدار تتا های واقعی ندارند اگر تعداد داده ها را بیشتر کنیم این الگوریتم بهتر جواب میدهد و همگرایی پارامتر های تخمین بهتر رخ میدهد. همچنین اگر از پارامتر نرخ فراموشی استفاده کنیم. باز هم نتیجه بدست امده بهتر میشود

با کاهش مقدار اولیه P به مقدار 100 نتایج زیر را بدست میاوریم

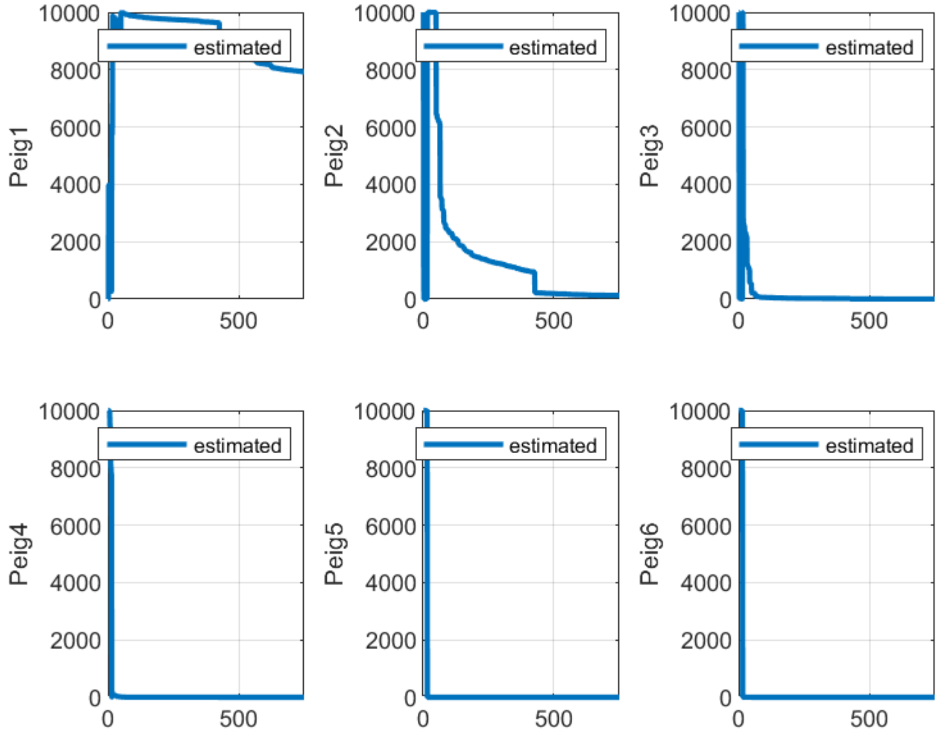


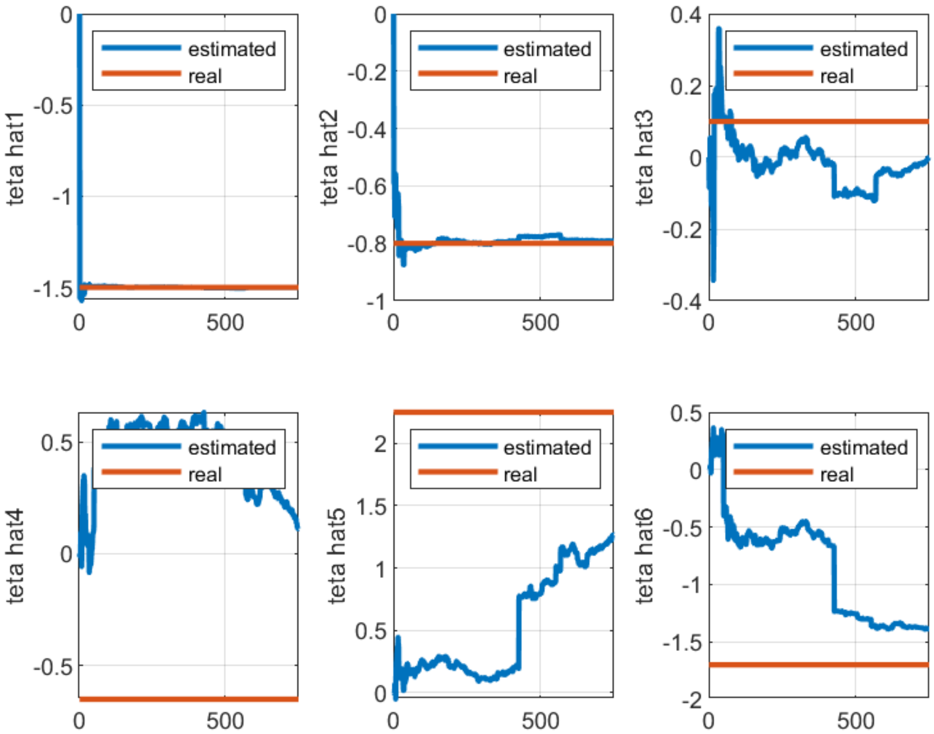


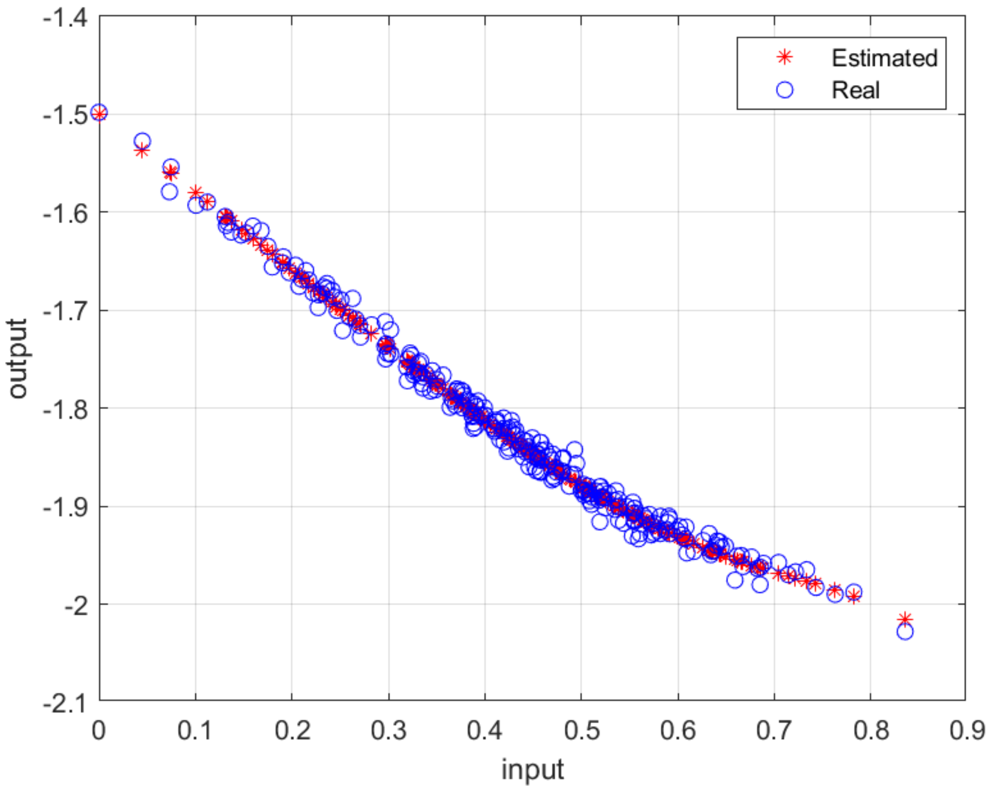



```
noise =

    'Medium noise'


Mean squared normalized error  =

   9.9392e-05
```

با مقدار اولیه 10 به توان 4 برای ماتریس P نتایج زیر را داریم

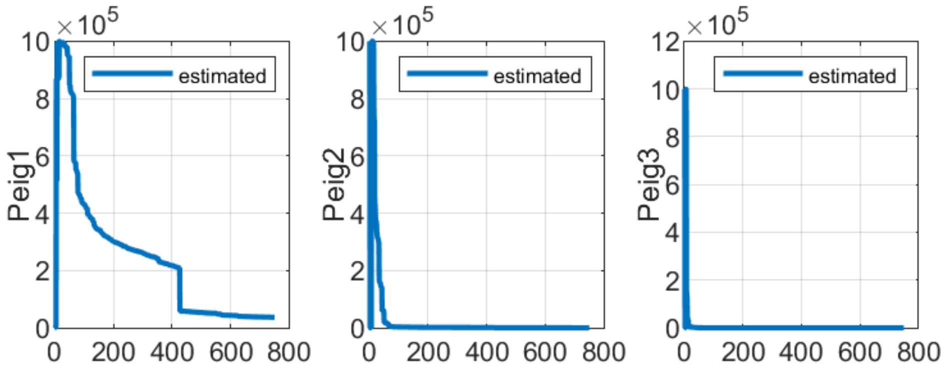

Peig1
Peig2
Peig3
Peig4
Peig5
Peig6
estimated
10000
8000
6000
4000
2000
0
500

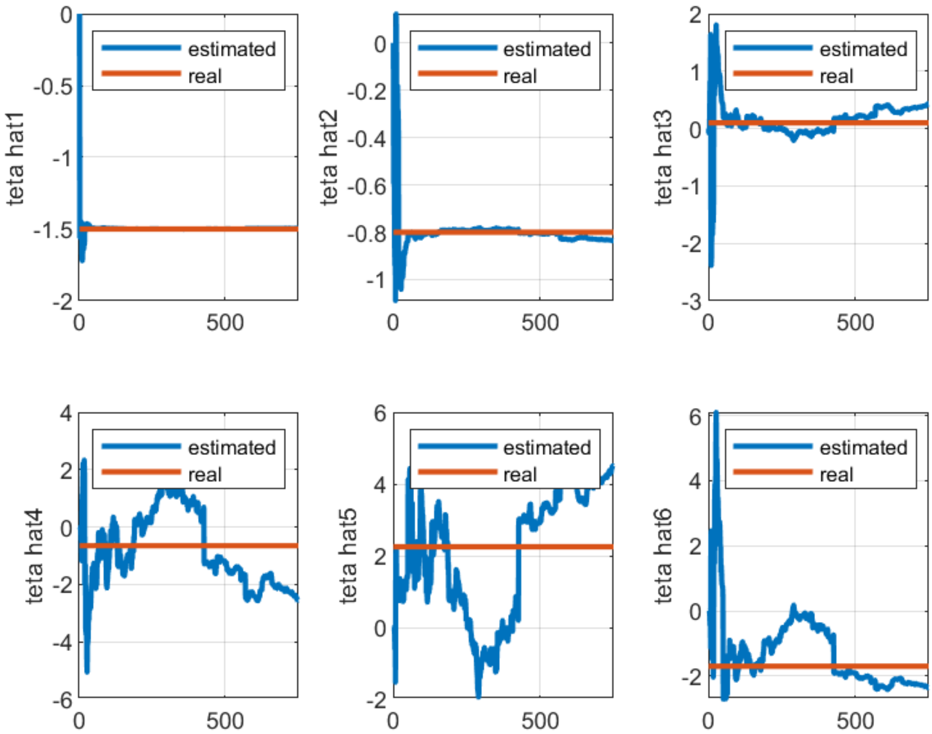

teta hat1
teta hat2
teta hat3
teta hat4
teta hat5
teta hat6
estimated
real

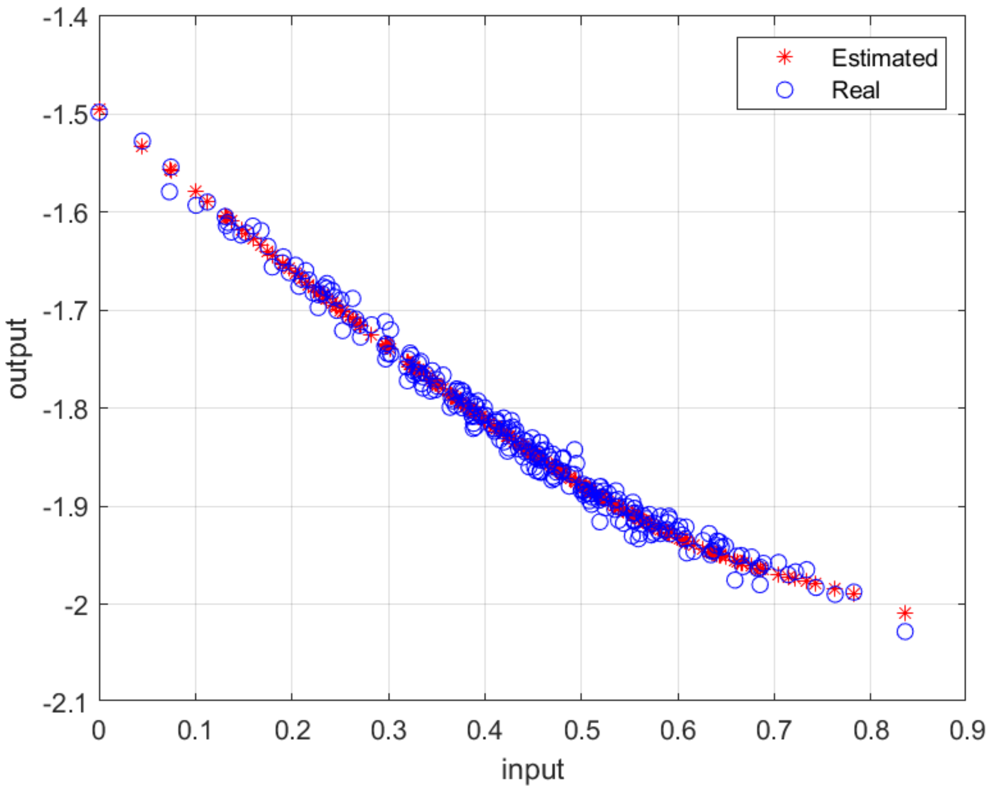



```
noise =

    'Medium noise'


Mean squared normalized error  =

   9.6095e-05
```

حال مقدار اولیه ماتریس P را بیشتر میکنیم به طور مثال مقدار ان را به 10 به توان 6 می رسانیم


Peig1
Peig2
Peig3
Peig4
Peig5
Peig6
×10^5
estimated
0 200 400 600 800


teta hat1
teta hat2
teta hat3
teta hat4
teta hat5
teta hat6
estimated
real
0 500




```
noise =

    'Medium noise'

Mean squared normalized error  =

   9.7908e-05
```

سوال 1-6

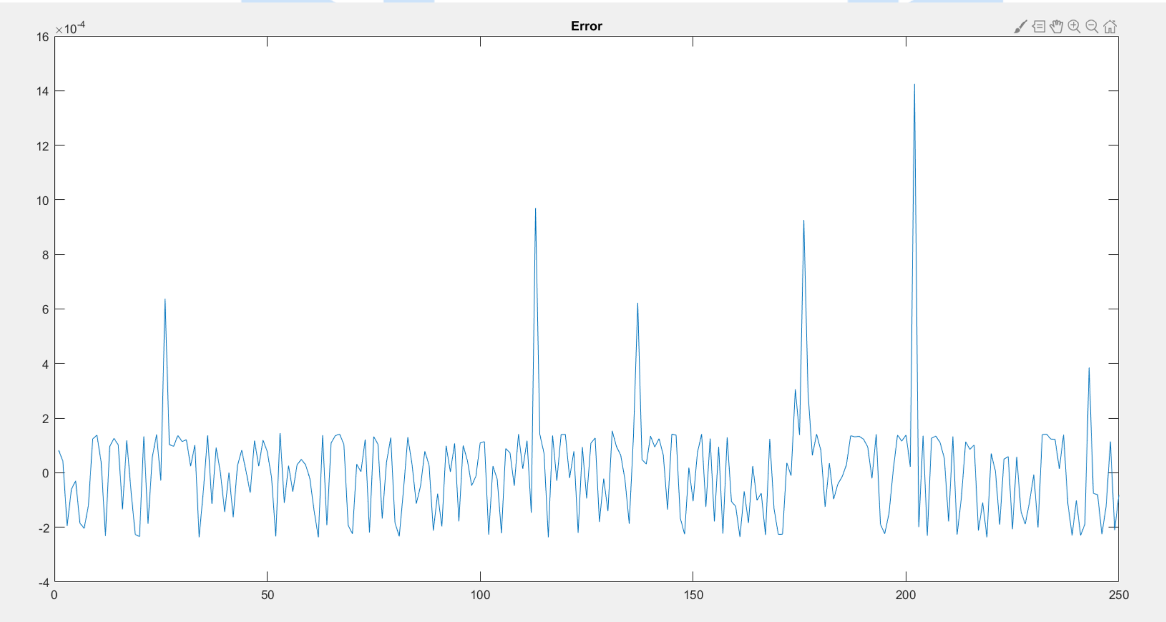
با توجه به اینکه داده های ورودی متعامد نیستند. پس روش های انتخاب پیش رونده (forward selection) و حذف عقب گرد (backward elimination) با احتمال زیادی نتایج مشابه ندارند. زیرا ممکن است یک رگرسور به تنهایی جواب خوبی داشته باشد اما در ترکیب با بقیه رگرسور ها جواب مناسبی نداشته باشد. تحلیل دقیق تر این مسئله بر اساس تصویر هر رگرسور در فضای برداری برروی رگرسور های دیگر و بردار y در فضای برداری رگرسور های نامتعامد است

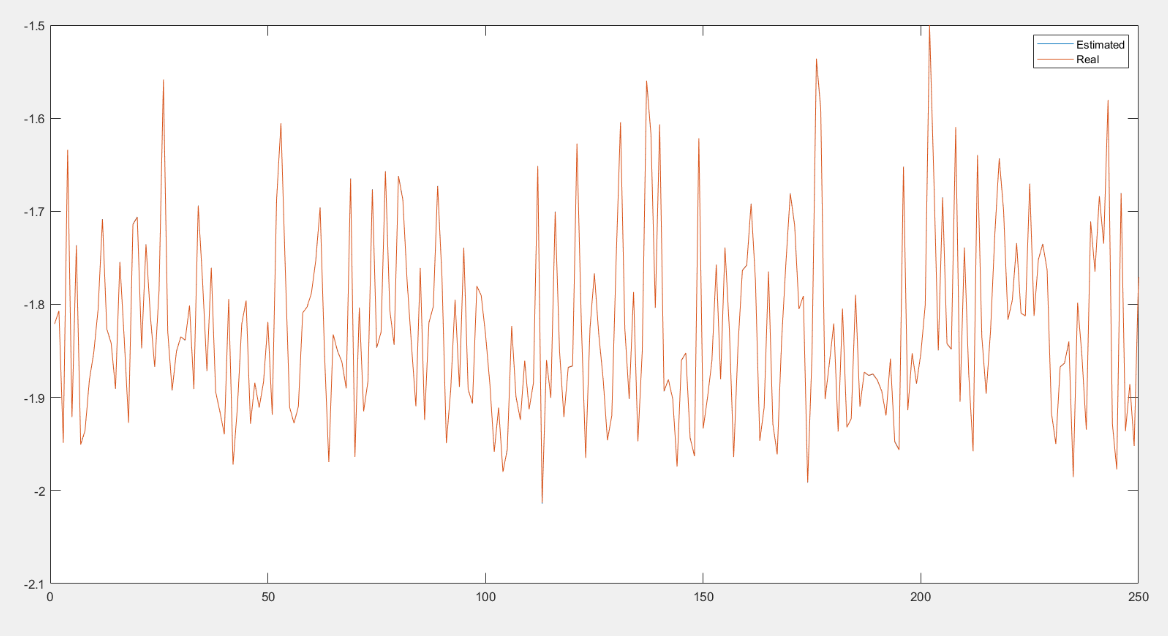
در این سوال با استفاده از 4 رگرسور بهتر به شناسایی سیستم پرداخته شده است و نمودار خروجی تخمین و خروجی واقعی سیستم و همچنین خطا این دو حالت رسم شده است

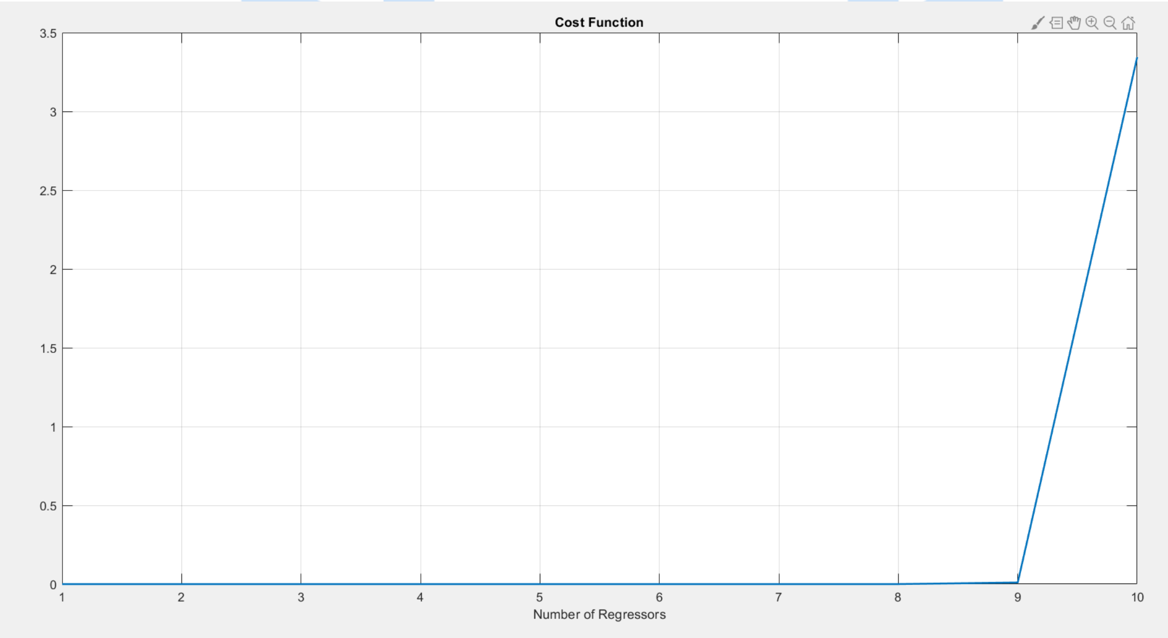
BE

برای روش BE نتایج زیر را بدست میاوریم

```
noise =

    'without noise'

Order of delete Regressors in Backward Elimination =5    8    3   10    4    6    9    7    2    1

MSE of Error in Test Data =

   3.6502e-08
```



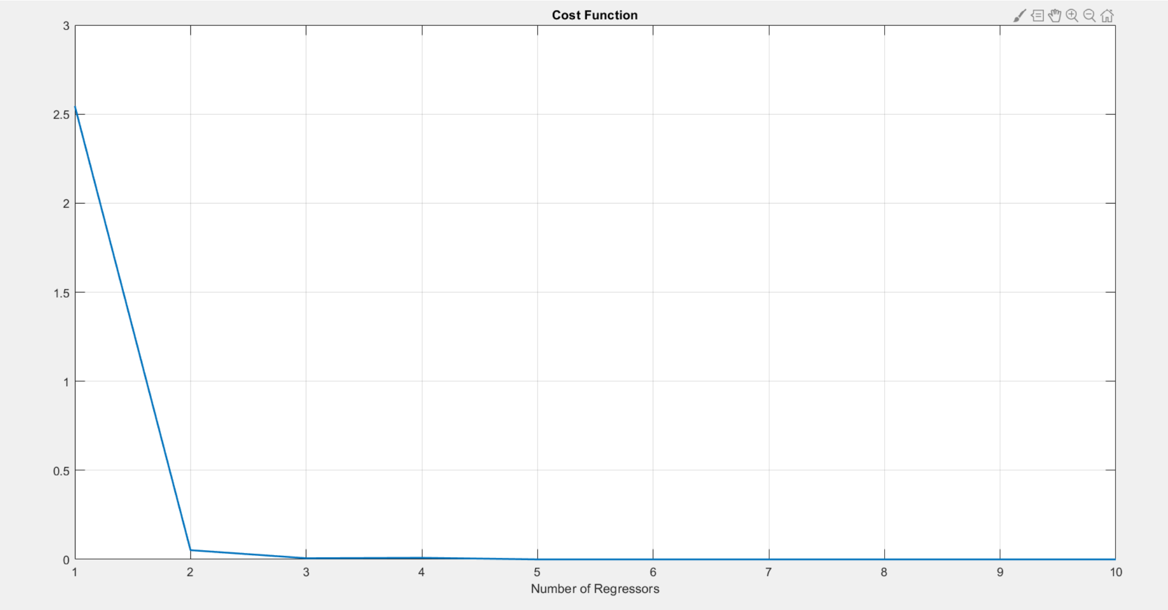


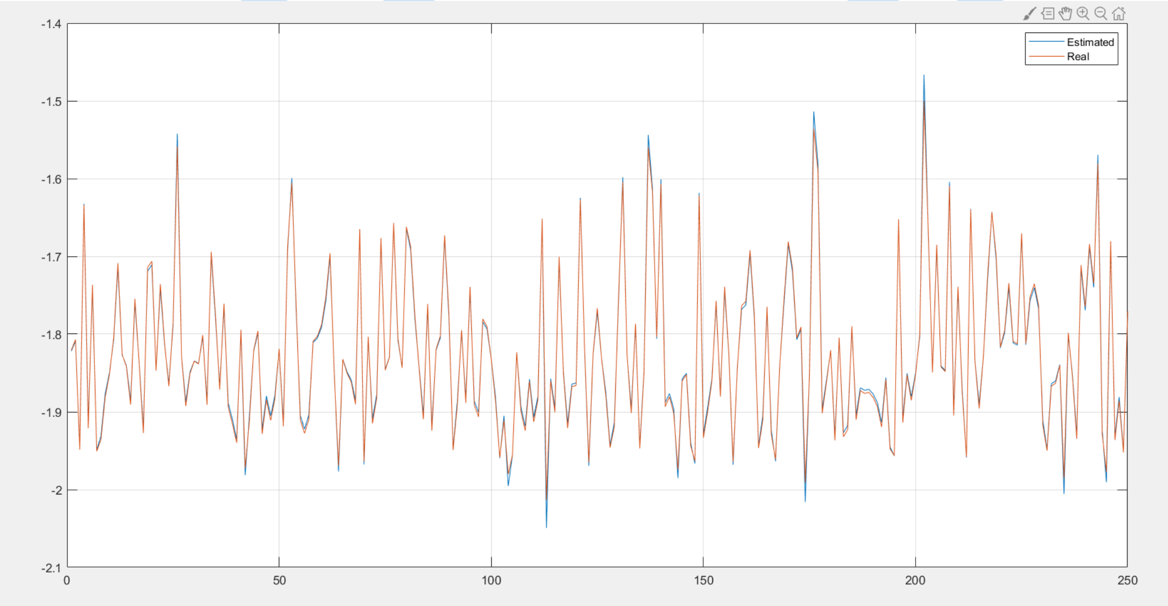


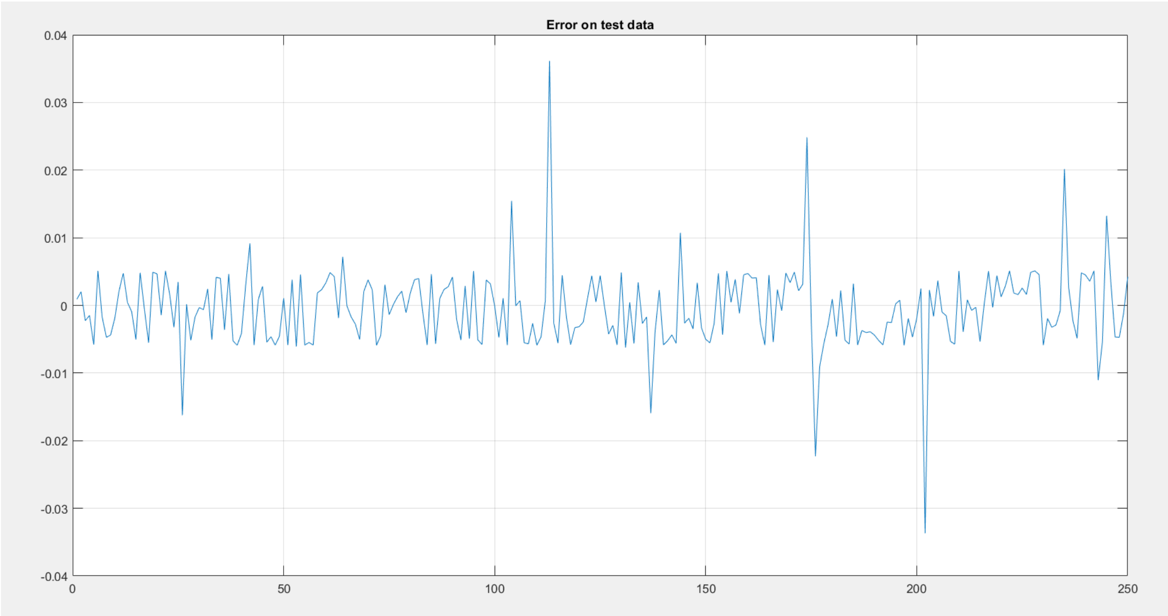
برای روش FS نتایج زیر را بدست میاوریم.

FS

```
noise =

    'without noise'

Order Of Important Regressors Forward Selection =1    2    3   10    7    9    8    4    5    6

MSE of Error in Test Data =

   3.5413e-05
```


Cost Function
3
2.5
2
1.5
1
0.5
0
1
2
3
4
5
6
7
8
9
10
Number of Regressors


Estimated
Real
-1.4
-1.5
-1.6
-1.7
-1.8
-1.9
-2
-2.1
0
50
100
150
200
250



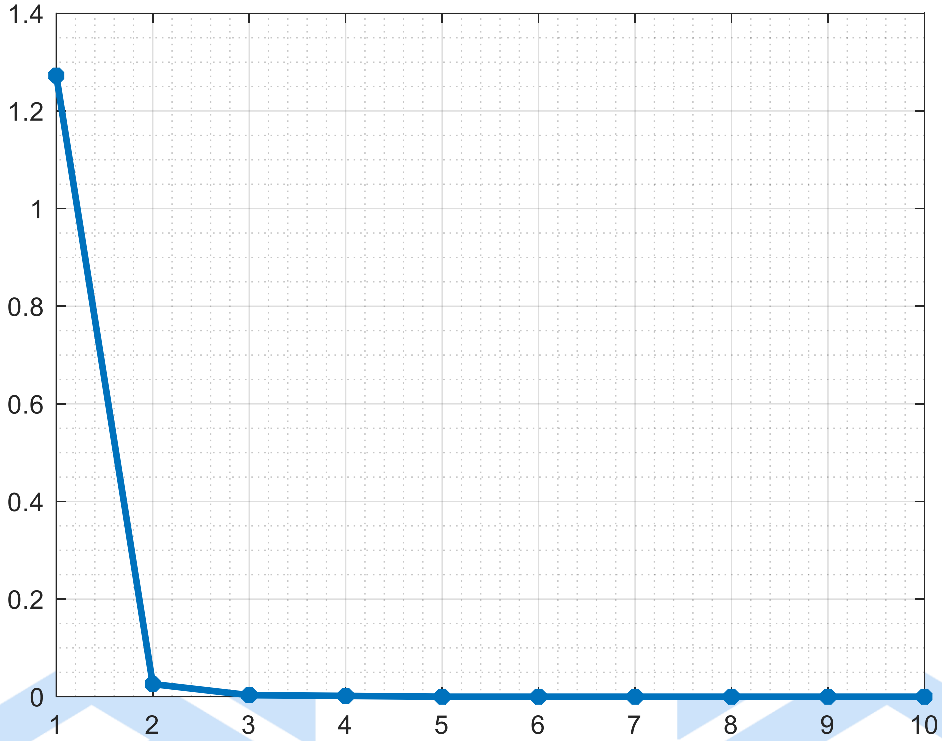
سوال 1-7

در روش متعامد سازی در ابتدا داده های ورودی را نسبت به یک دیگر متعامد میکنیم سپس روش LS را انجام میدهیم در این قسمت با توجه به اینکه داده ها را متعامد سازی کرده ایم نتایج روش B.E و روش F.S با یک دیگر یکسان شده اند

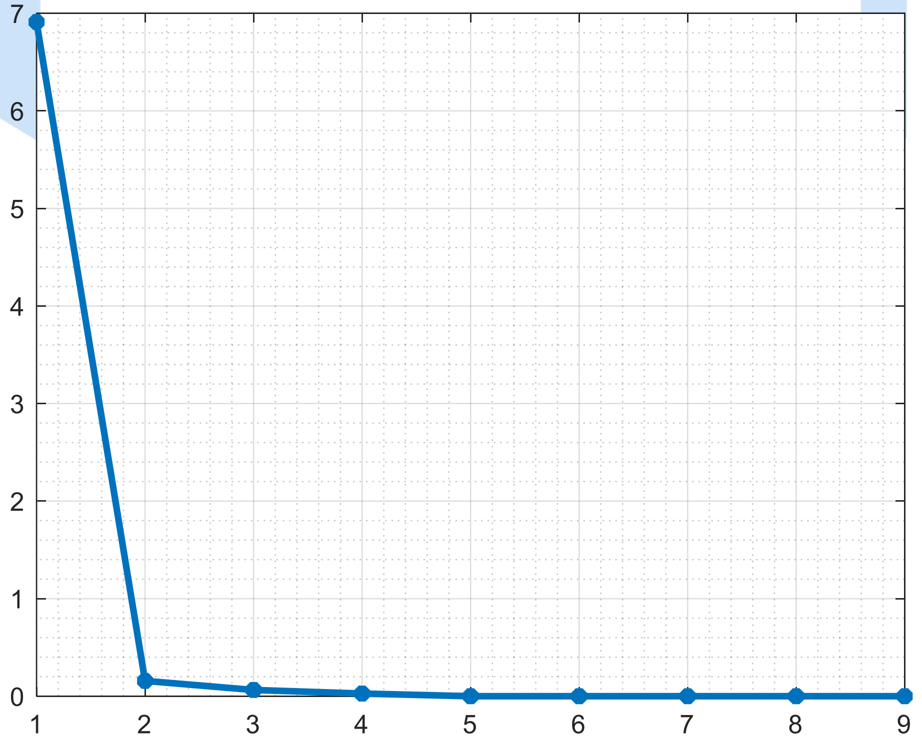
در شکل زیر روش F.S و B.E زده شده است و رگرسور های با ارزش به ترتیب نوشته شده اند همچنین نمودار تابع هزینه در هر مرحله نشان داده شده است

0
1
2
3
4
5
6
7
8
9

تابع هزینه روش F.S



تابع هزینه روش B.E



سوال 8-1

در این روش انتخاب تابع الفا و همچنین بازه تغییرات و اندازه هر گام تغییر در نتیجه خروجی بسیار تاثیر گذار است و در صورت تغییر هر یک از سه پارامتر بالا نتیجه خروجی تغییر پیدا خواهد کرد.

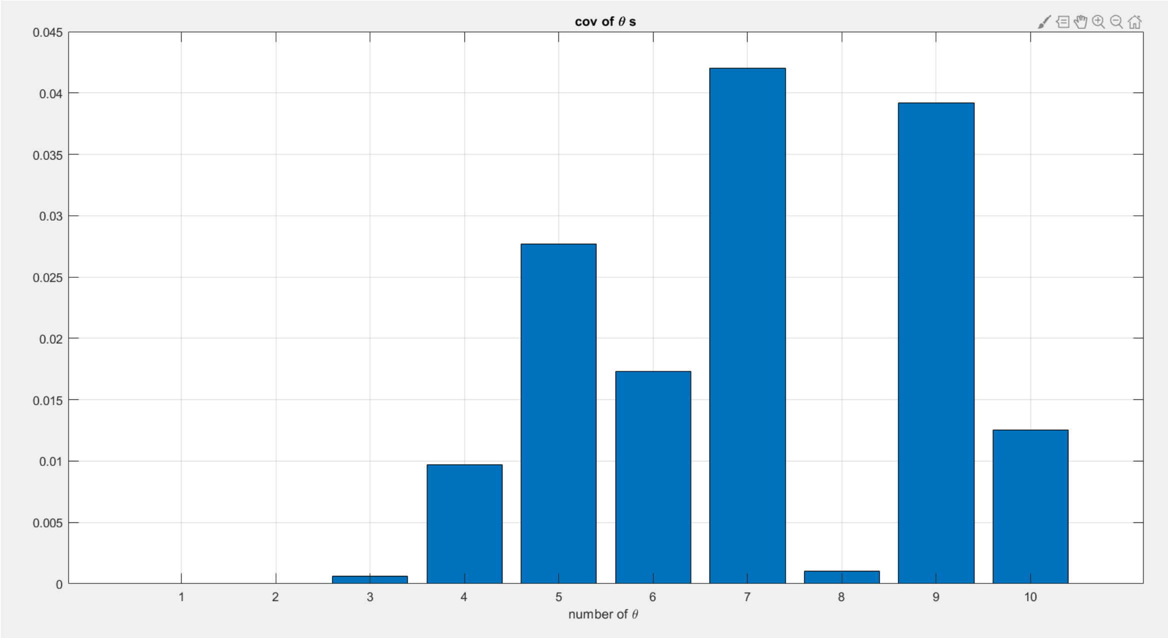
در اینجا مقدار الفا با استفاده از راهنمایی استاد به صورت زیر ساخته ایم در بازه صفر تا 0.000001 با step های 0.00000001 مقادیر الفا را ساخته ایم

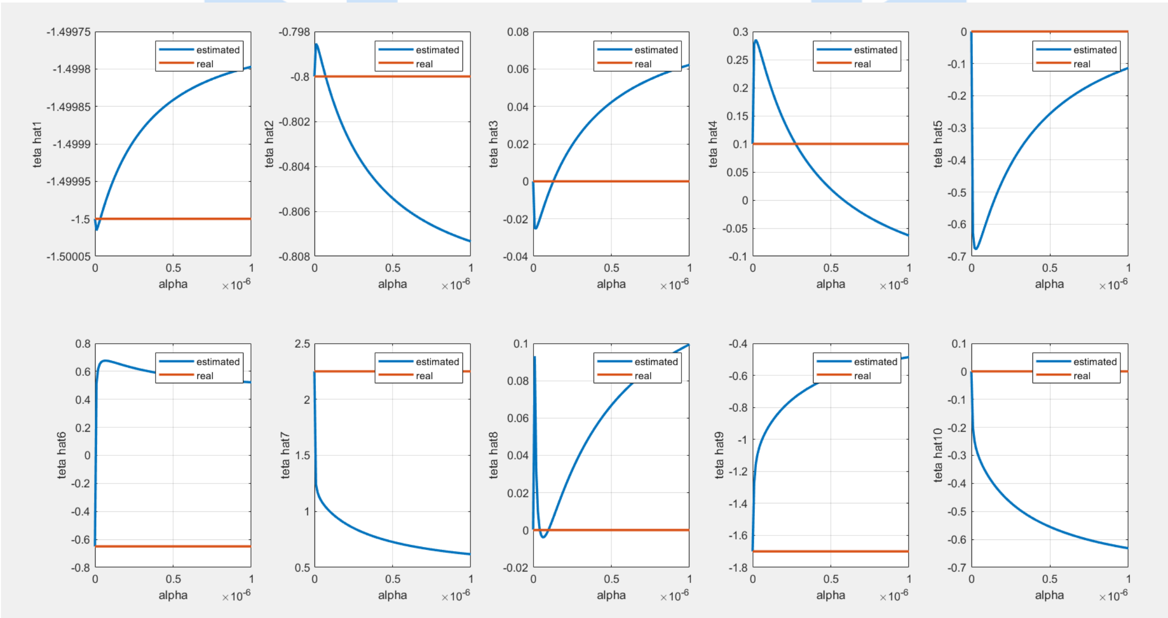
برای انتخاب مقادیر رگرسور نمودار تغییرات رگرسور را با تغییرات الفا رسم کرده ایم و کوواریانس تغییرات رگرسور ها را به عنوان یک معیار در نظر گرفته ایم

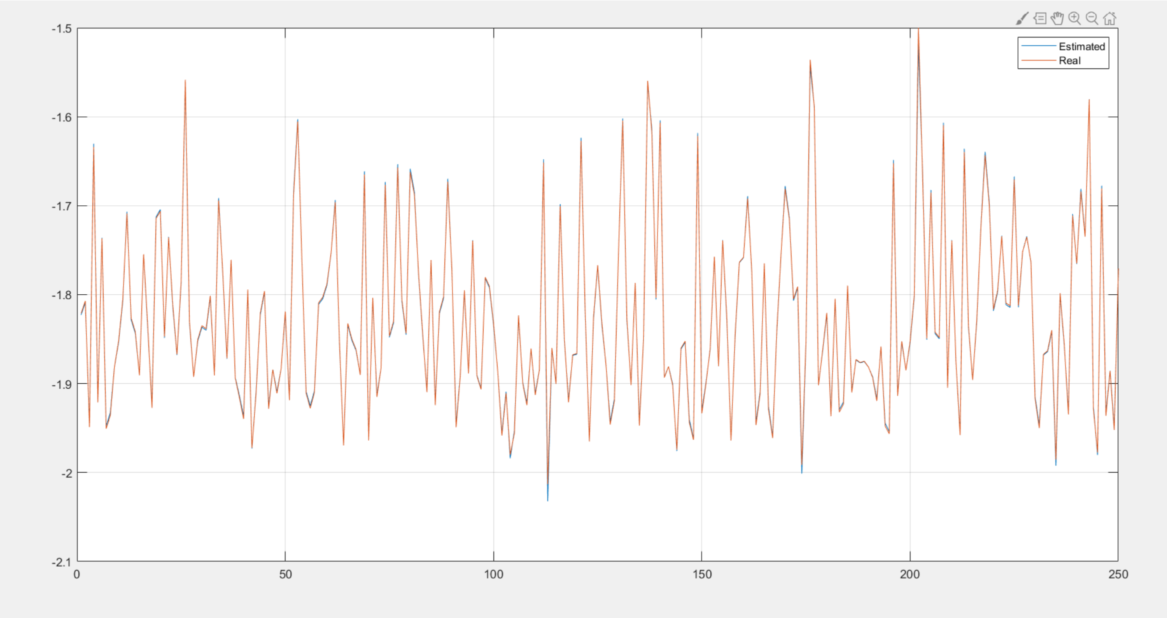
همچنین با استفاده از 5 رگرسور بهتر به شناسایی سیستم پرداخته ایم

رگرسور های مهم به ترتیب برابر اند با

	1
1	1
2	2
3	3
4	8
5	4
6	10
7	6
8	5
9	9
10	7


cov of θ s
number of θ
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
1
2
3
4
5
6
7
8
9
10

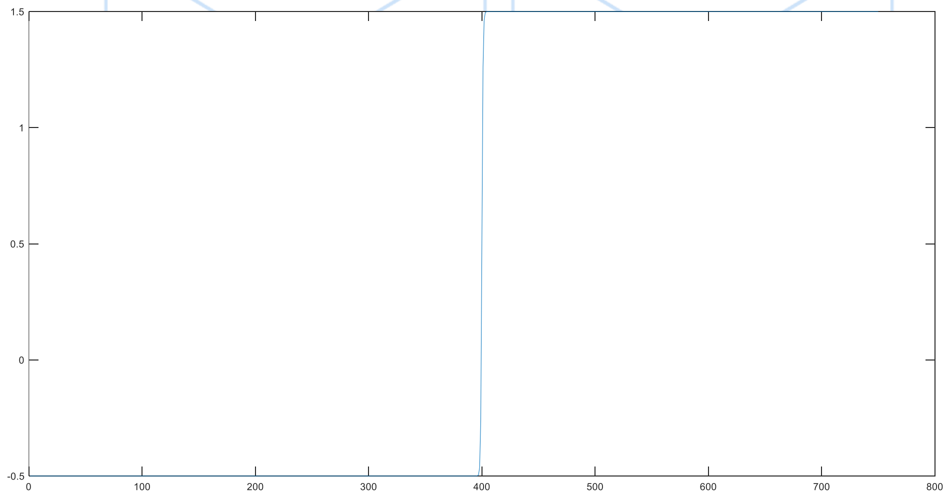

teta hat1
teta hat2
teta hat3
teta hat4
teta hat5
teta hat6
teta hat7
teta hat8
teta hat9
teta hat10
estimated
real
alpha
×10^-6



سوال دو

قسمت 1-2

در این سوال یکی از پارامتر های سیستم متغیر با زمان است. برای حالت g برابر با یک تغییرات این متغیر به شکل زیر است.

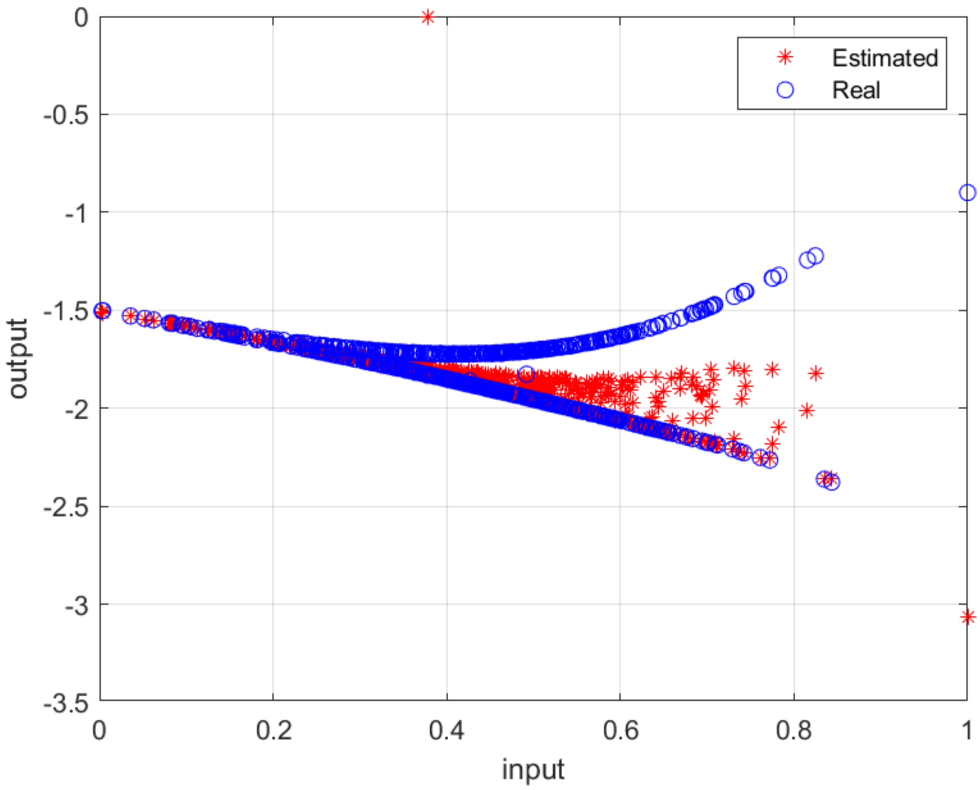


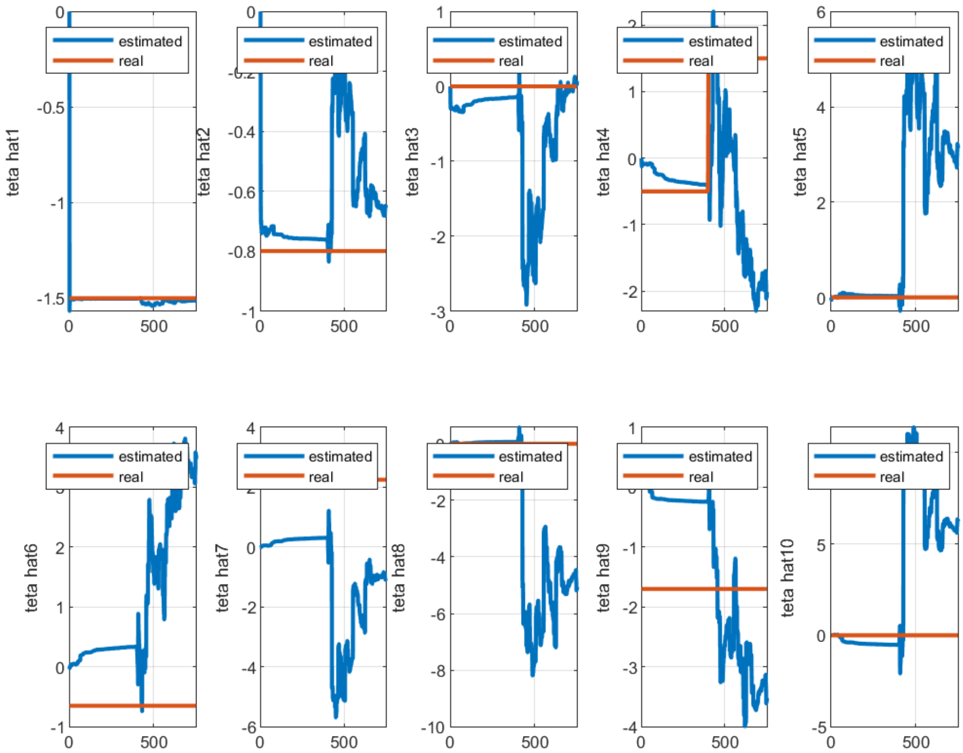
در این سوال میخواهیم عملکرد الگوریتم RLS در حالتی که یکی از متغییر ها با زمان تغییر میکند را بررسی کنیم در این مسئله قبل از تغییر کردن تتا سه پارامتر های تخمین زده شده از سیستم در حال همگرا شدن هستند به مقدار واقعی خود هستند و مقادیر ویژه ماتریس P به سمت صفر میل کرده اند اما زمانی که پارامتر تتا سه به صورت سریع تغییر میکند مقادیر تخمین زده شده برای سیستم تغییرات شدیدی پیدا میکنند و از مقدار واقعی سیستم دورتر میشوند و همچنین مقادیر ویژه ماتریس P بزرگ میشوند سپس با امدن داده های جدید مقادیر تخمین زده شده سعی میکنند به مقادیر واقعی سیستم شبیه شوند. اما به دلیل تغییرات شدیدی که پیدا کرده اند دیرتر به حالت نهایی خود همگرا شده اند و در برخی موارد نیز هنوز به همگرایی نرسیده اند. البته اگر از تعداد داده های بیشتری استفاده کنیم این همگرایی بیشتر قابل مشاهده است.

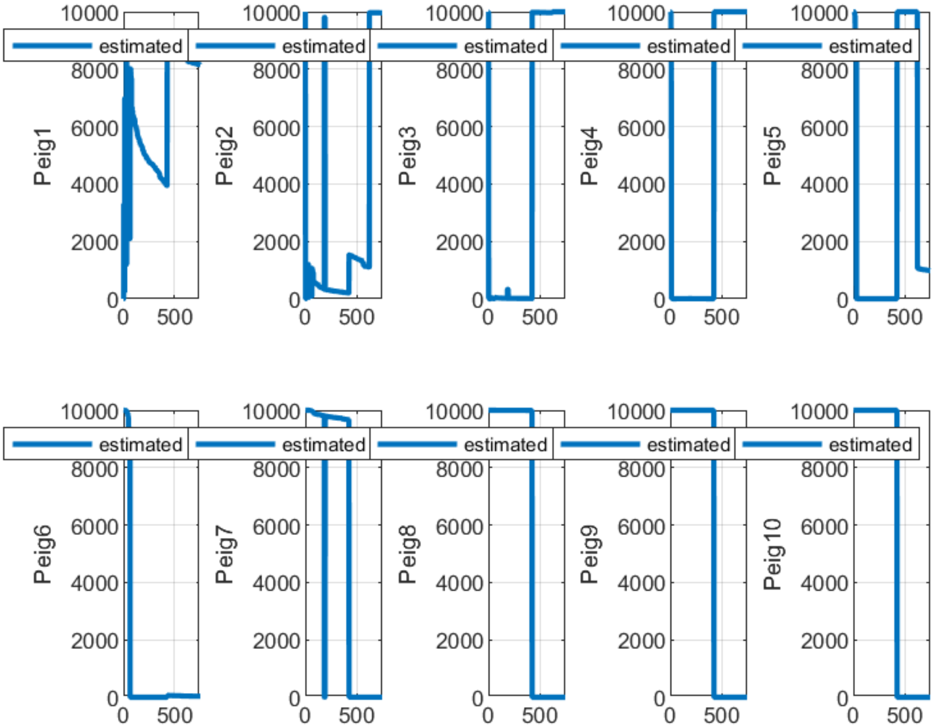
در حالتی که کوواریانس نویز کمتر است الگوریتم نتایج بهتری داشته است و پارامتر های تخمین زده شده تغییرات کمتری داشته اند اما در حالتی که کوواریانس نویز زیاد است تغییرات پارامتر های تخمین زده شده بیشتر میشود

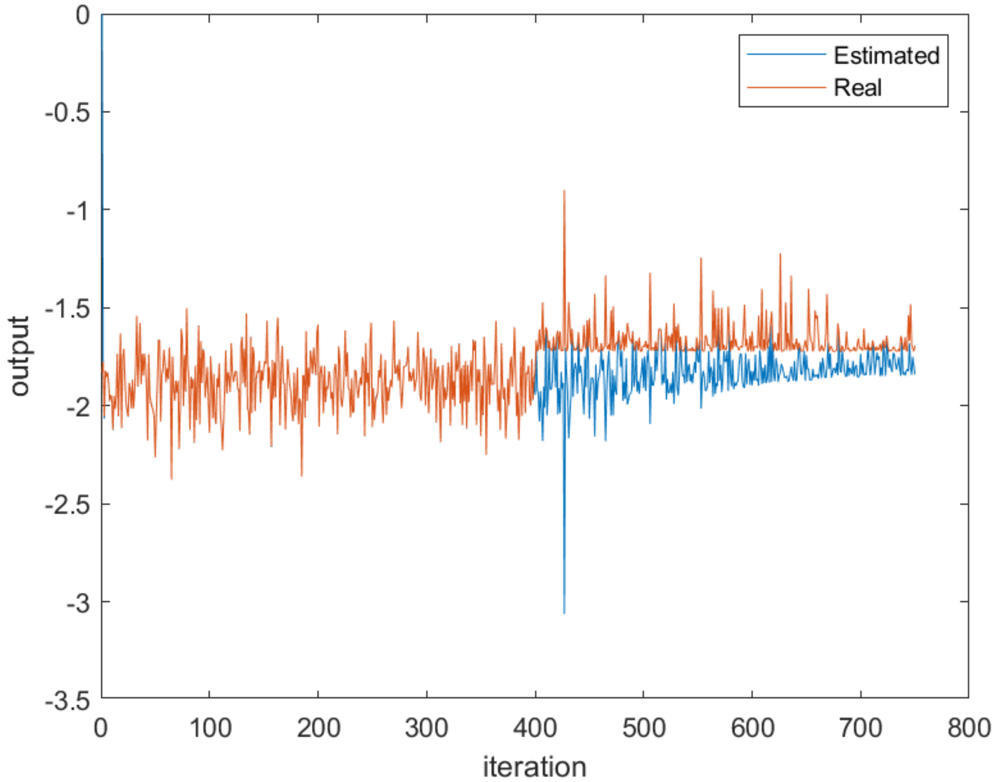
همچنین در هیچ یک از حالت ها الگوریتم نتوانسته تغییرات رخ داده در پارامتر تتا سه را ردیابی کند

حالت اول نویز کم

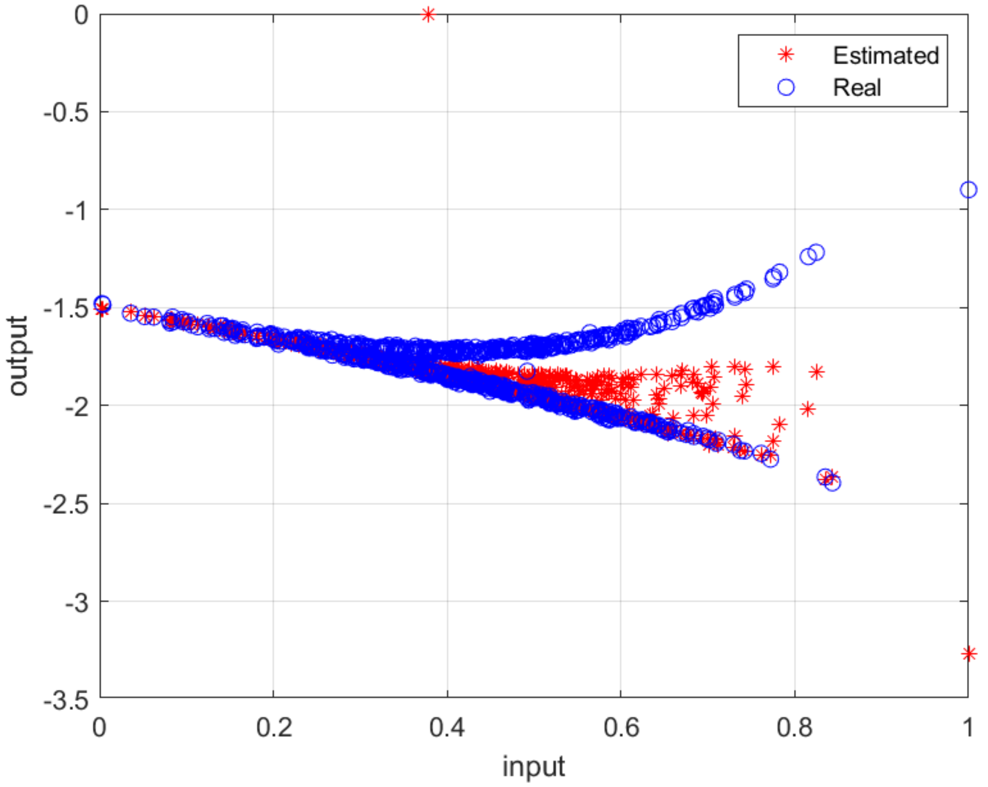


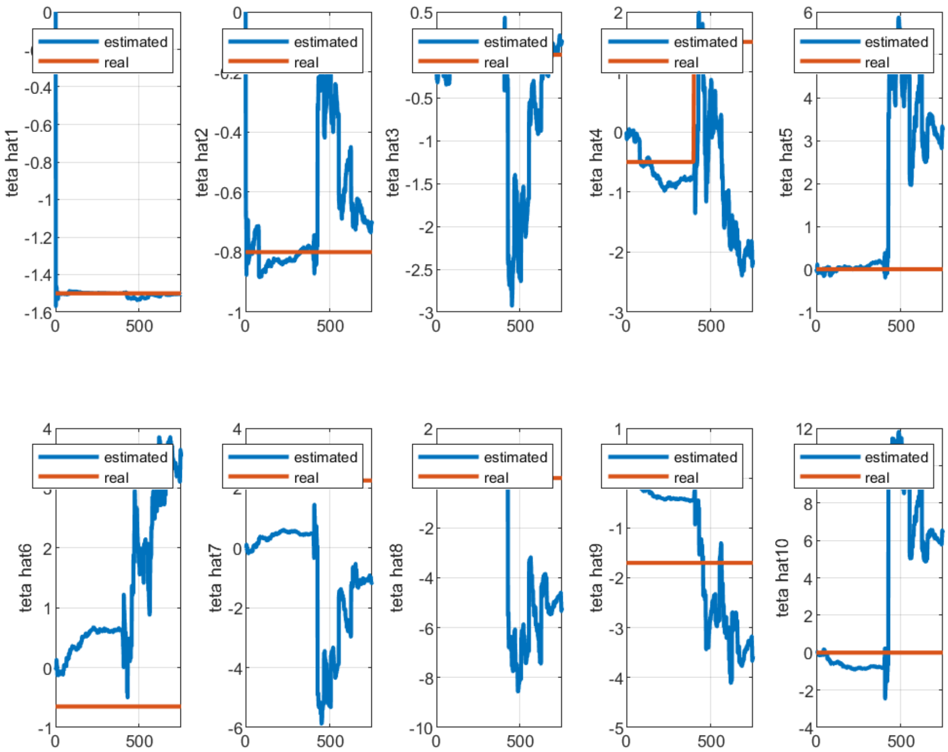


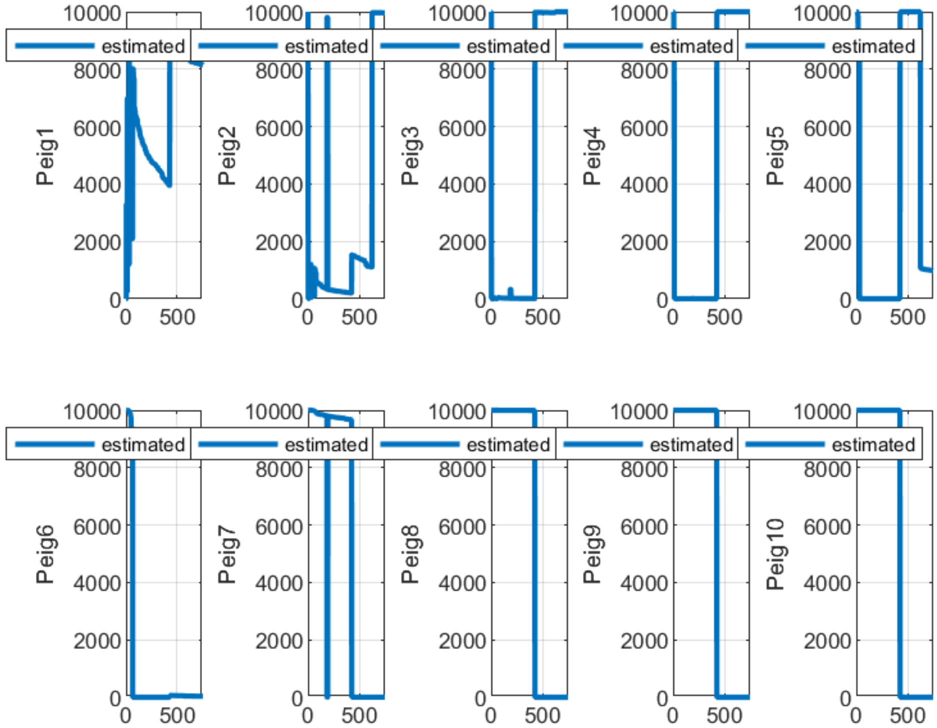

Peig1
Peig2
Peig3
Peig4
Peig5
Peig6
Peig7
Peig8
Peig9
Peig10
estimated
10000
8000
6000
4000
2000
0
0
500

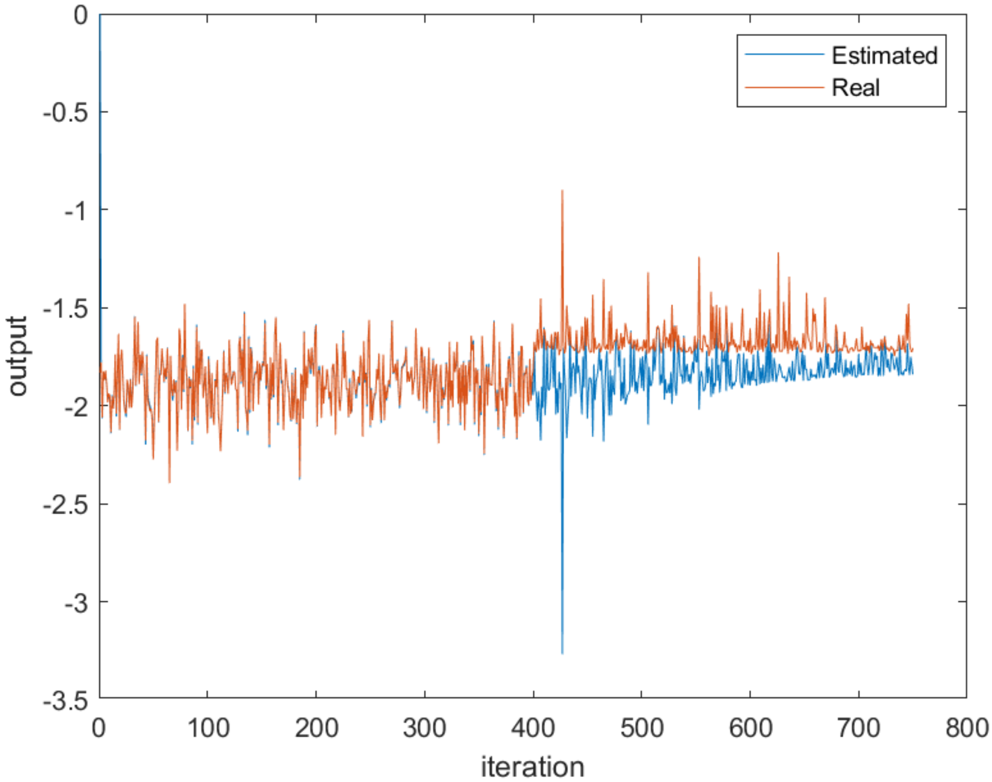

Estimated
Real
output
iteration
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
0
100
200
300
400
500
600
700
800

حالت دوم نویز متوسط

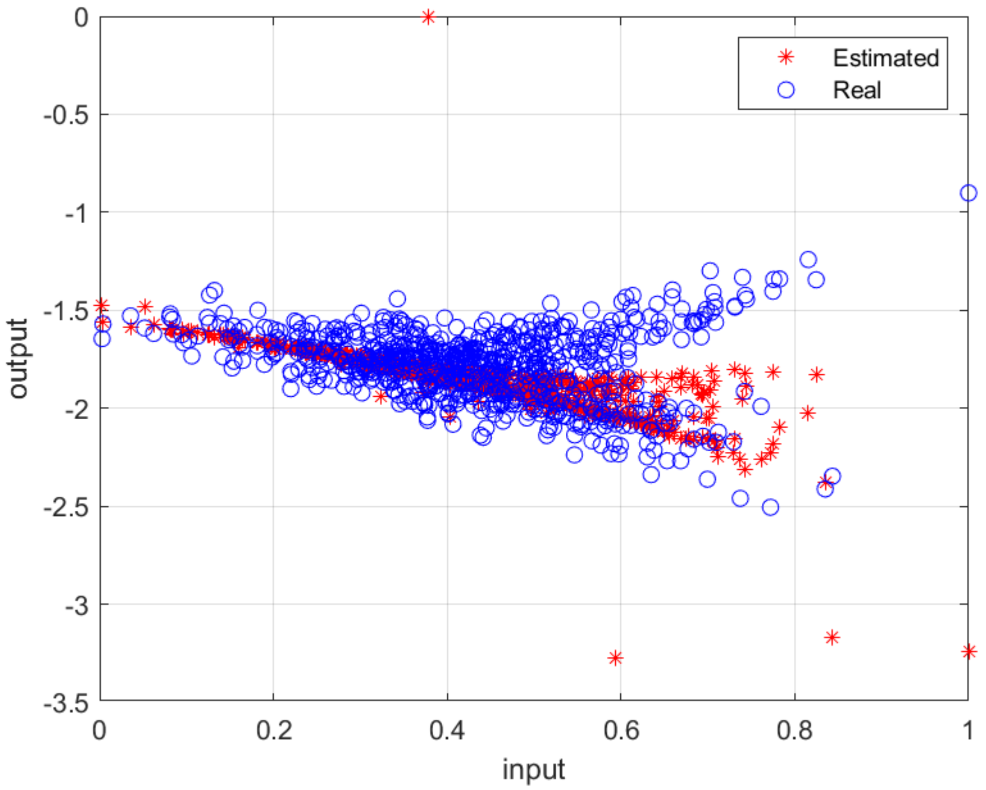


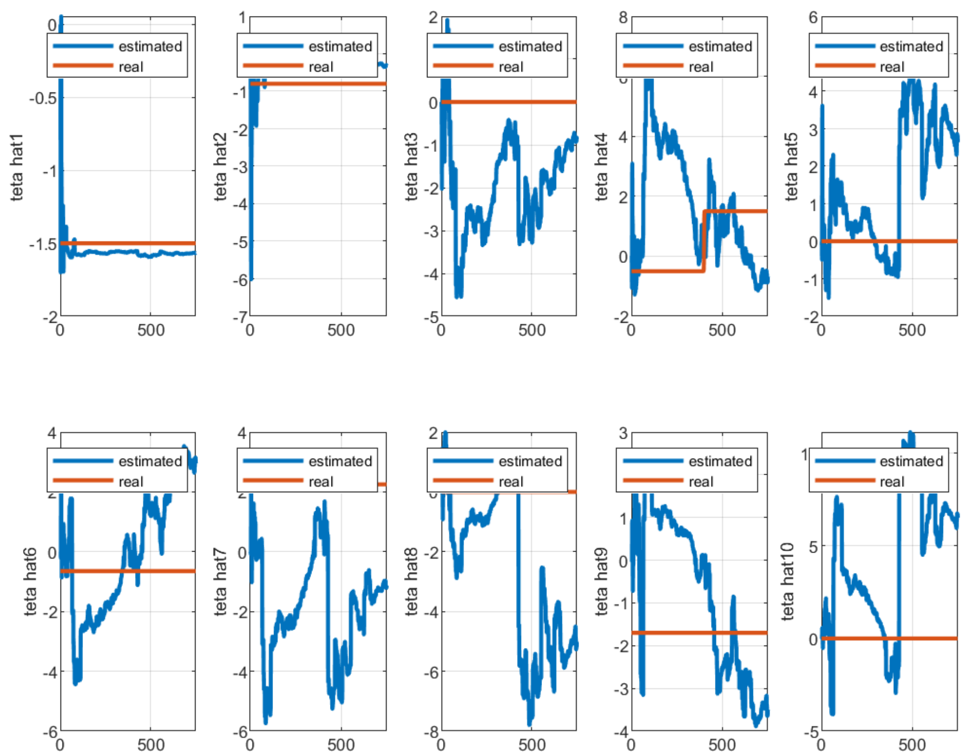


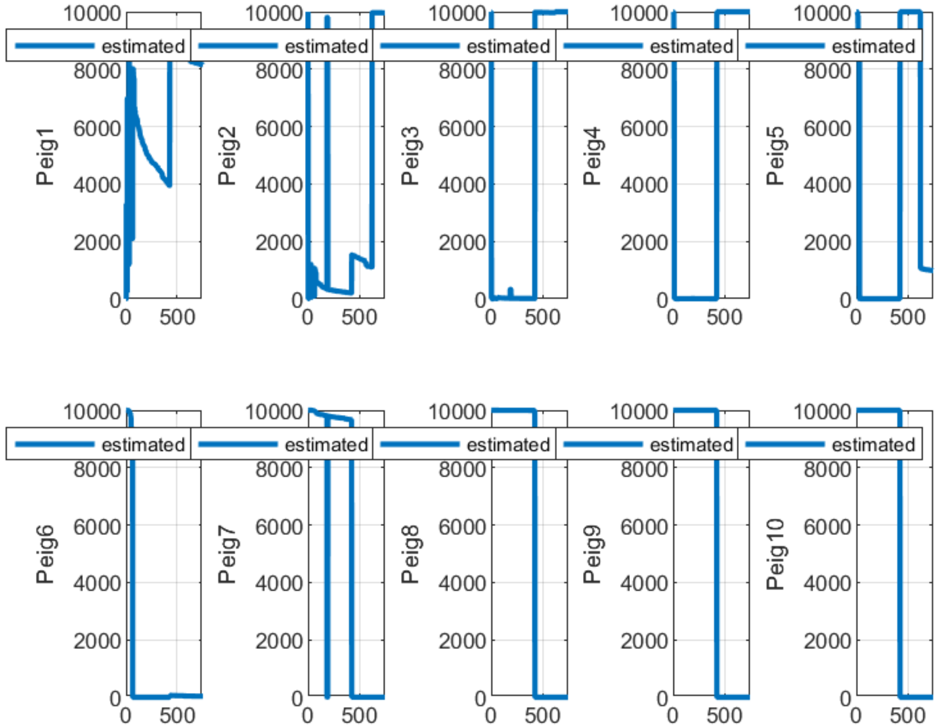


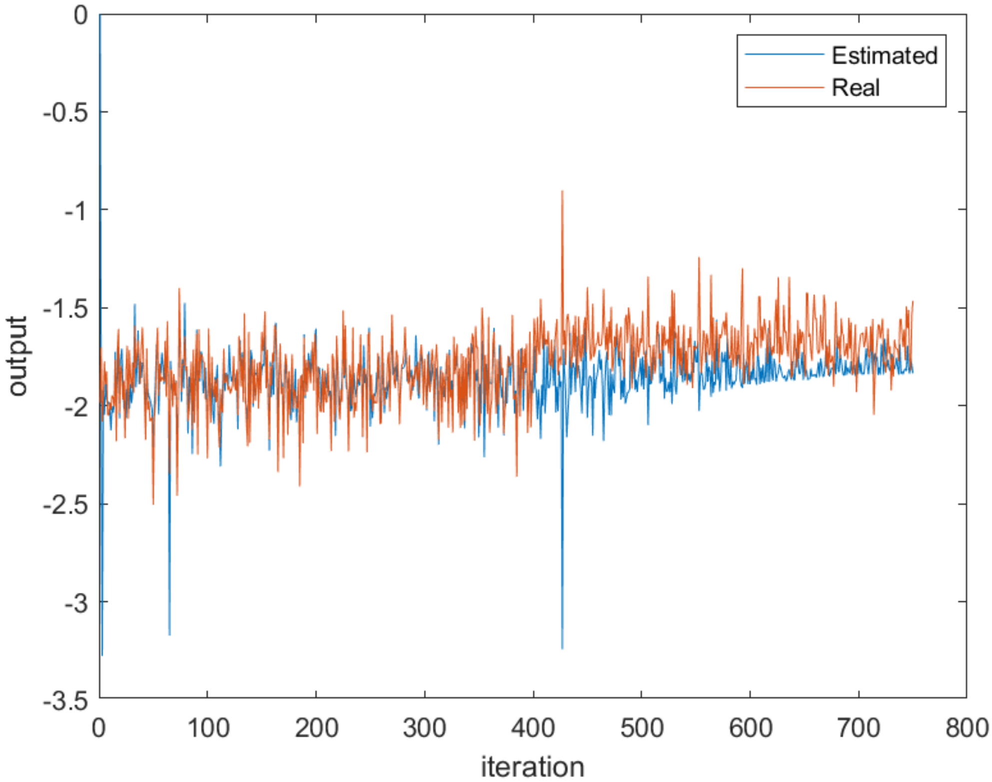


حالت سوم نویز زیاد


Estimated
Real
output
input
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
0
0.2
0.4
0.6
0.8
1

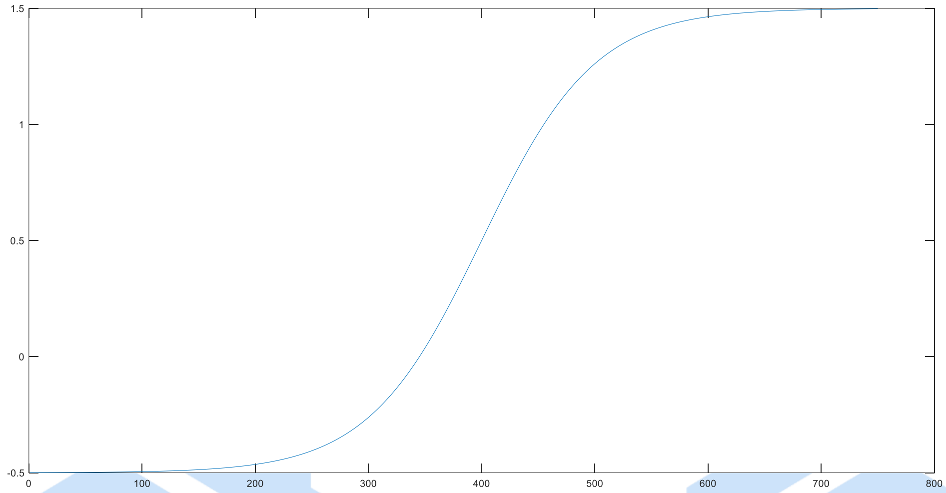

estimated
real
teta hat1
teta hat2
teta hat3
teta hat4
teta hat5
teta hat6
teta hat7
teta hat8
teta hat9
teta hat10
500


estimated
Peig1
Peig2
Peig3
Peig4
Peig5
Peig6
Peig7
Peig8
Peig9
Peig10
10000
8000
6000
4000
2000
0
500


Estimated
Real
output
iteration
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
100
200
300
400
500
600
700
800

قسمت 2-2

در سوال قبلی حالت g برابر با یک را بررسی کردیم در این سوال مقدار g را برابر با 0.01 قرار میدهیم در این حالت تغییرات پارامتر تتا برابر با شکل زیر است

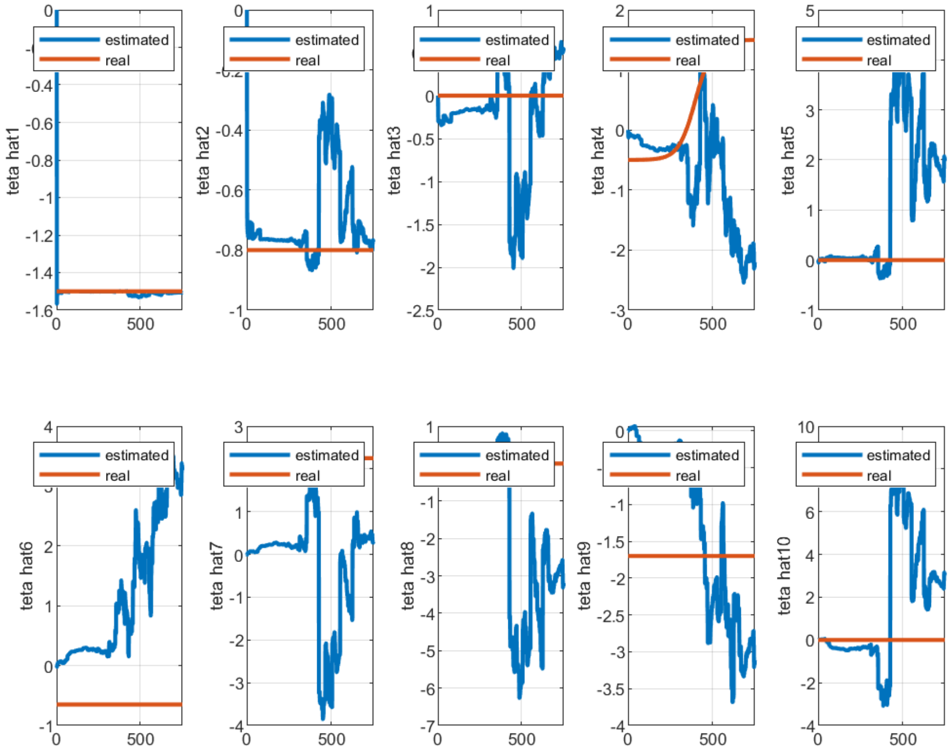


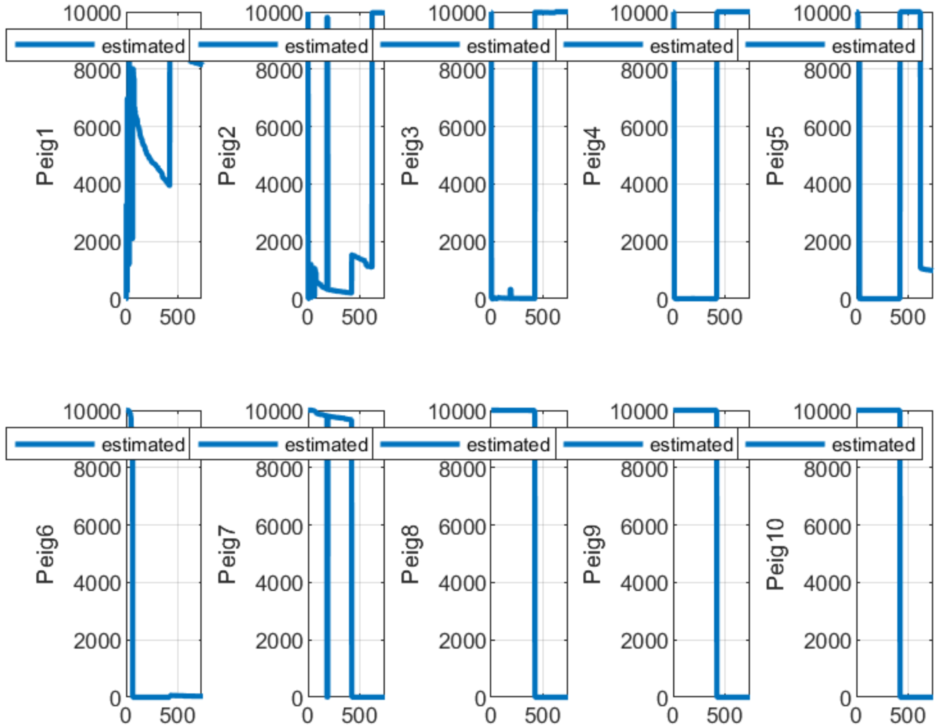
در این حالت نیز مشابه حالت قبلی الگوریتم RLS نمیتواند تغییرات پارامتر تتا سه را دنبال کند اما تفاوتی که این حالت با حالت قبلی دارد به دلیل اینکه تغییرات تتا سه در این حالت به صورت نرم و هموار است و به صورت ضربه ای نمیباشد پارامتر های تتا تخمین زده شده در سیستم نیز تغییرات کمتری نسبت به حالت قبل پیدا کرده اند.

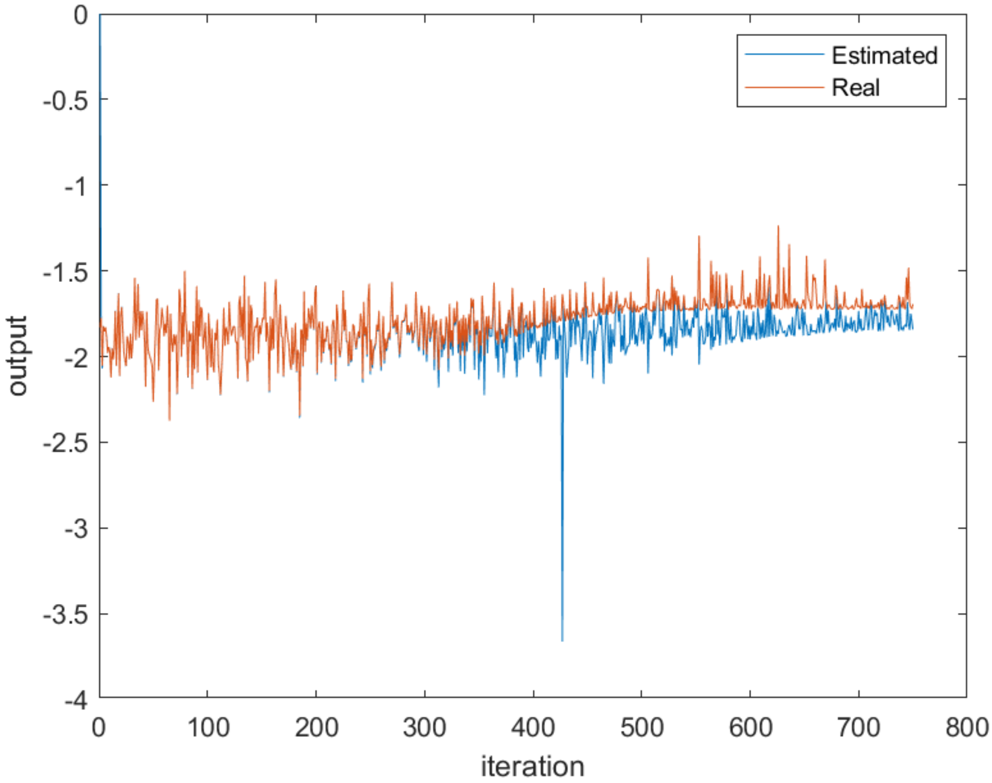
به طور مثال در این بخش که g برابر با 0.01 است برای حالت نویز کم پارامتر تتا دو بعد از اعمال تغییرات در سیستم از 0.8- تا 0.4- بالا رفته است اما در بخش قبلی که g برابر با یک بود این متغییر از 0.8- تا

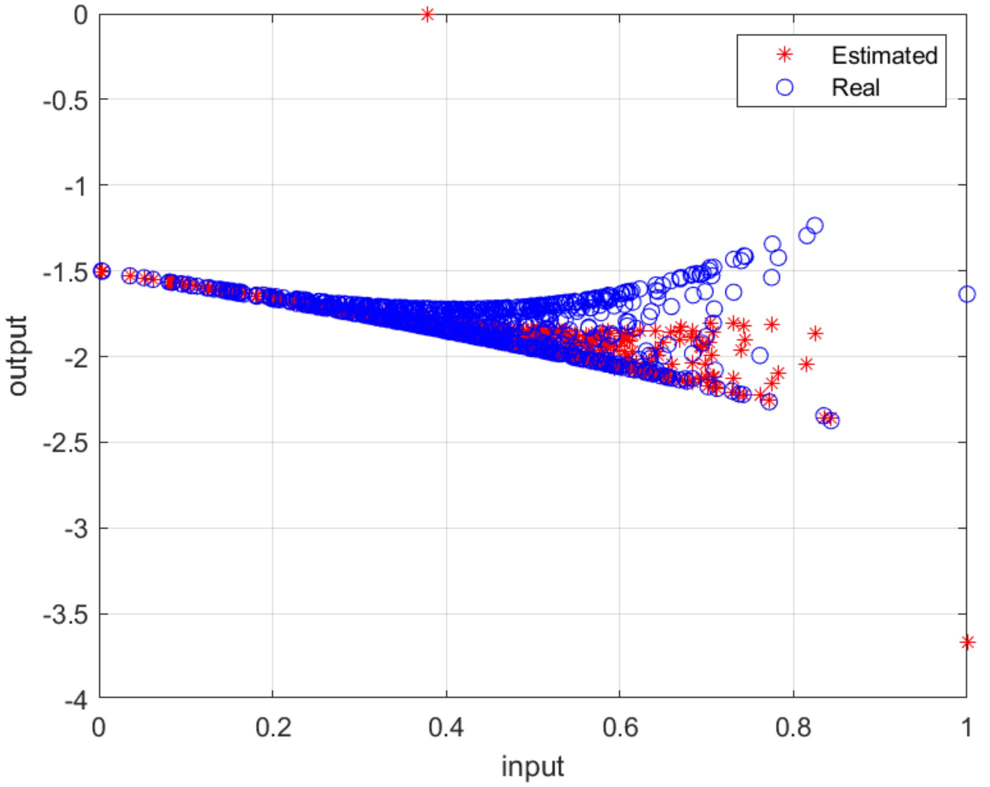
0.2- بالا رفته است و این نشان دهنده کمتر بودن نرخ تغییرات در این حالت میباشد و این به این خاطر است که در این حالت تغییرات با شیب کمتری صورت گرفته است

حالت اول نویز کم

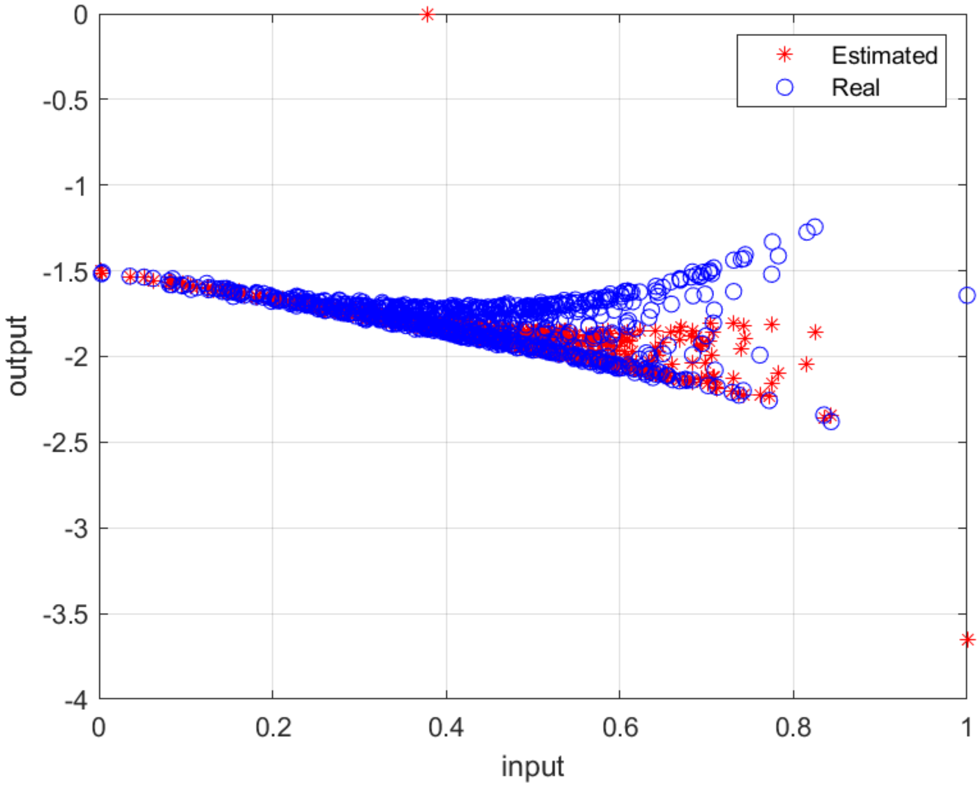


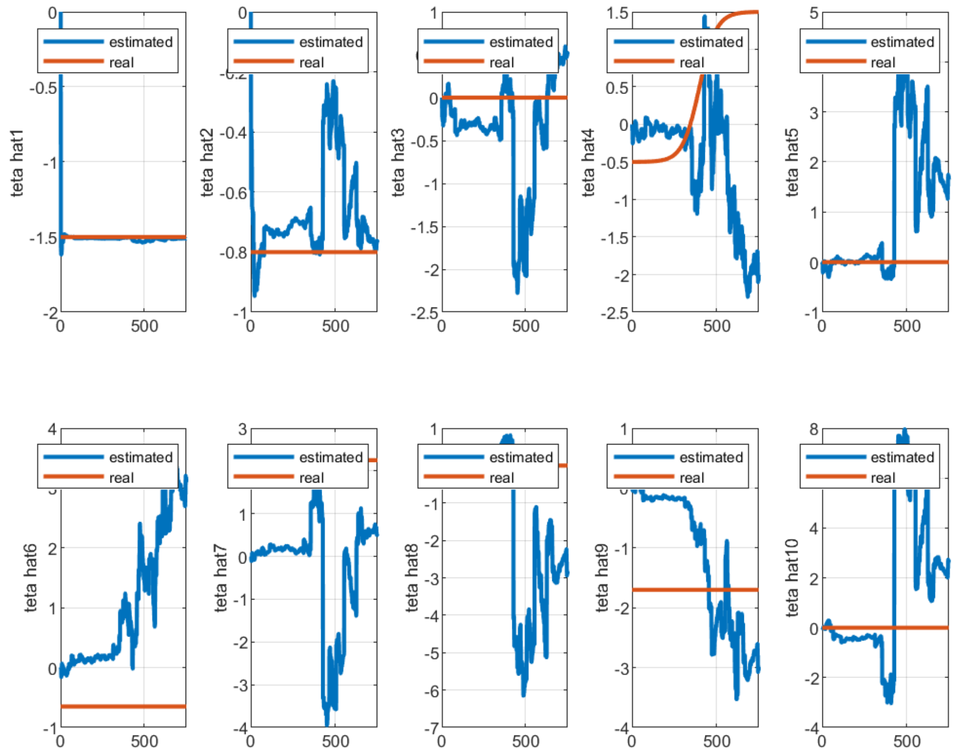


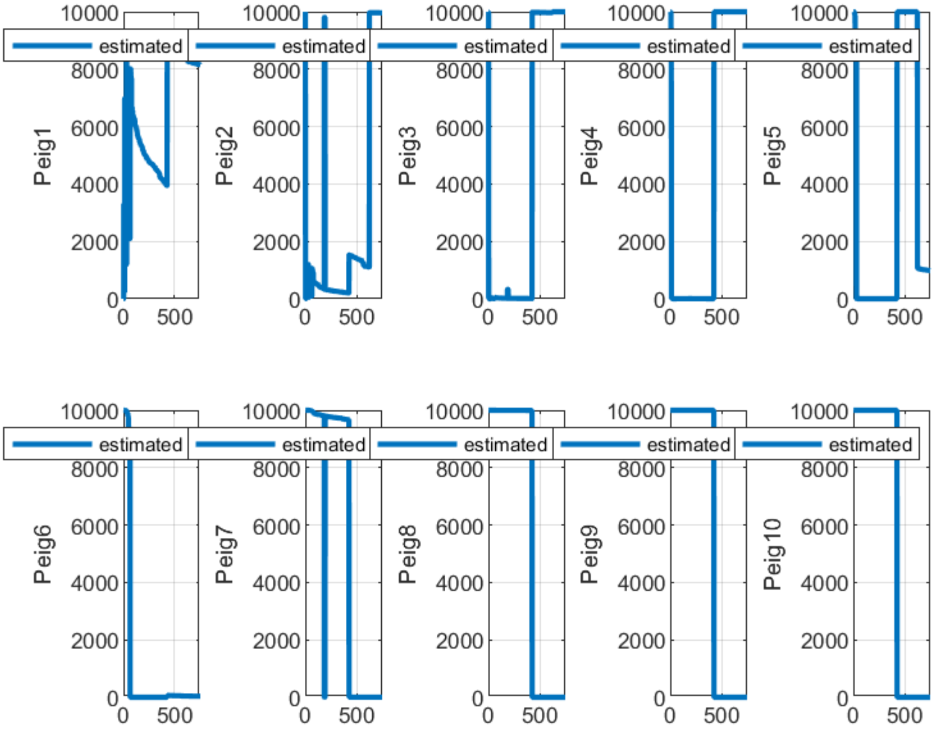

Estimated
Real
output
iteration
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
-4
0
100
200
300
400
500
600
700
800

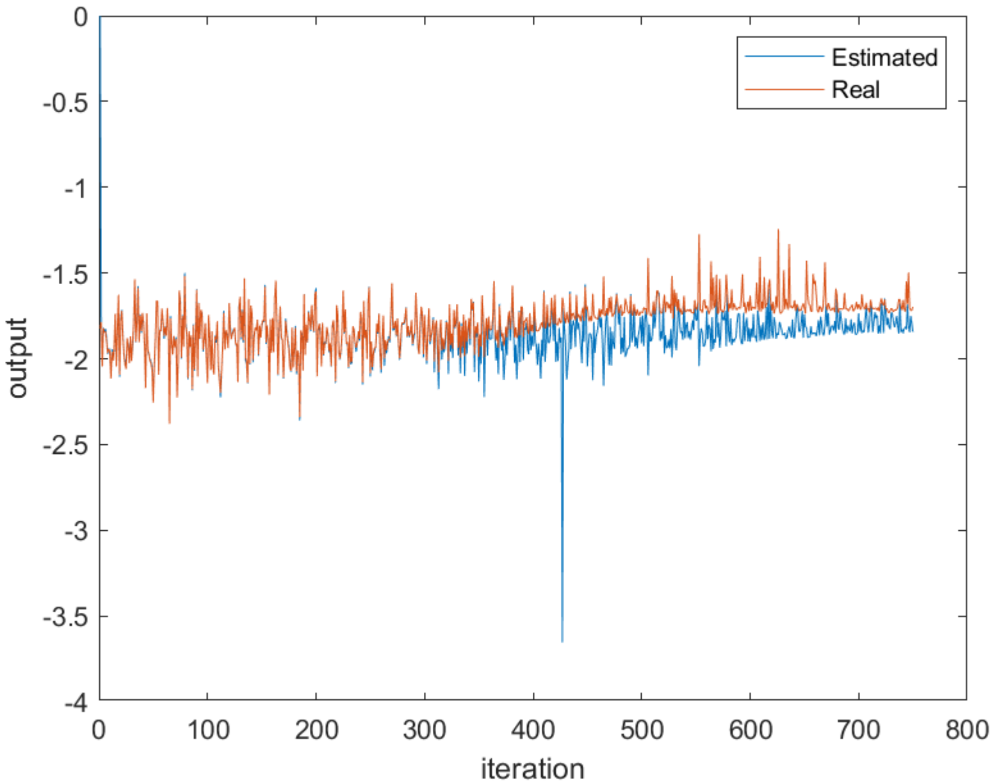

Estimated
Real
output
input
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
-4
0
0.2
0.4
0.6
0.8
1

حالت دوم نویز متوسط

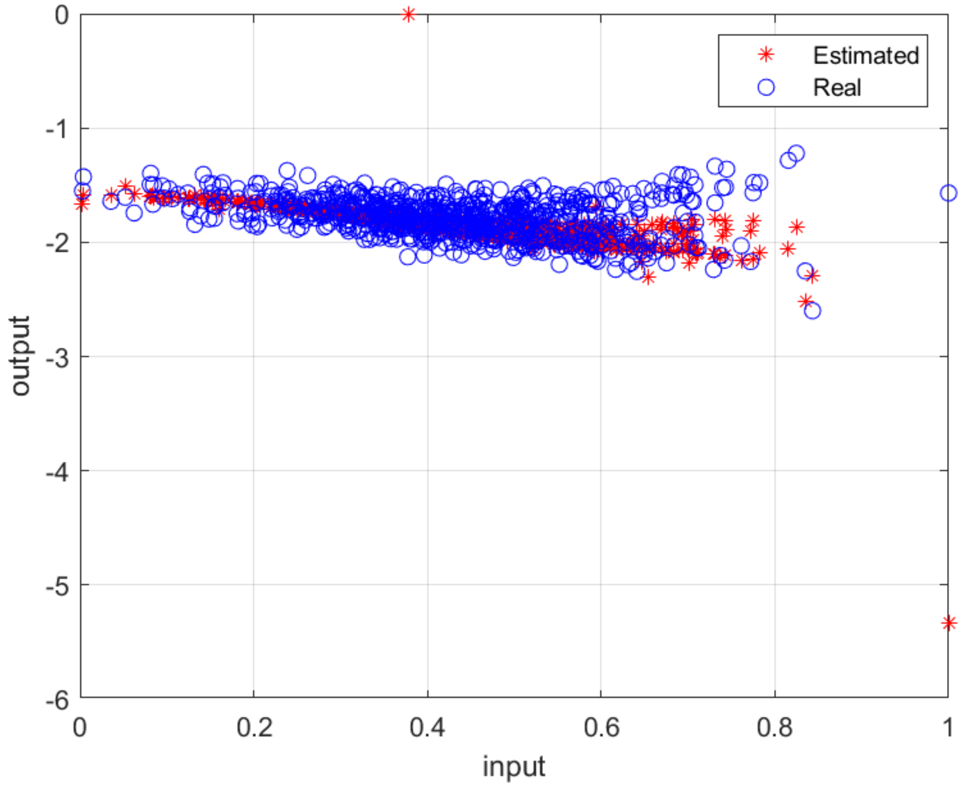


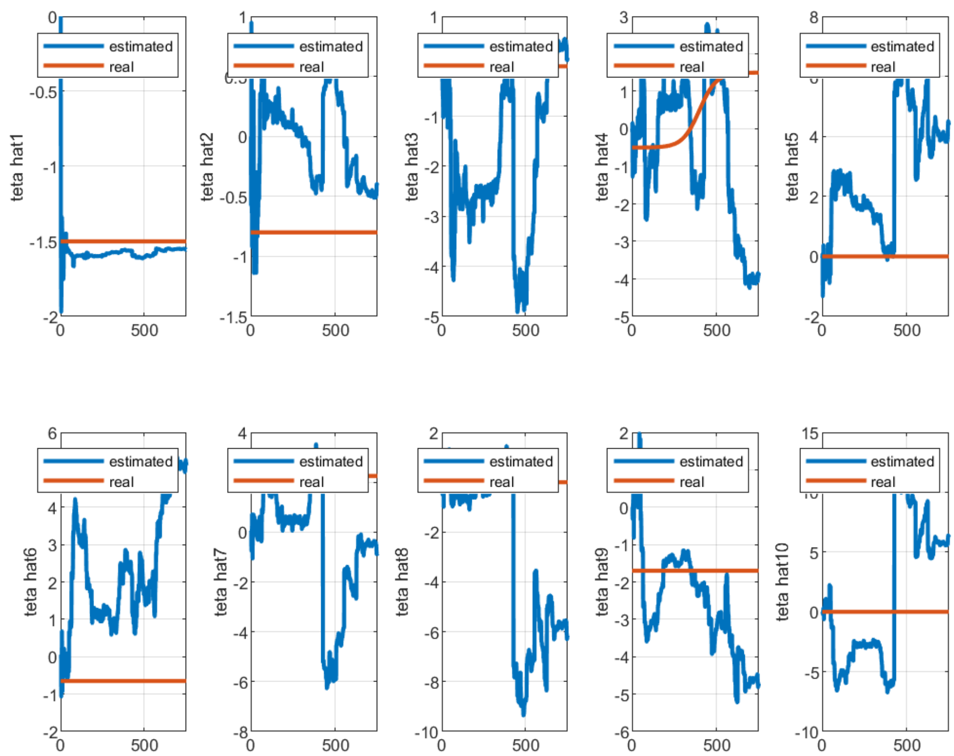


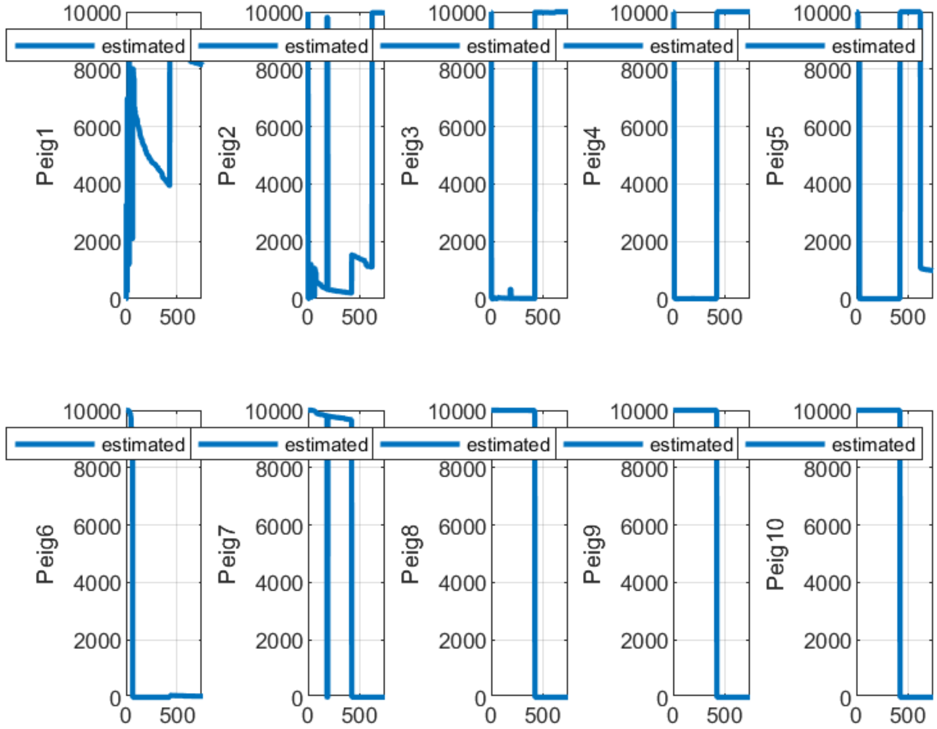

estimated
Peig1
Peig2
Peig3
Peig4
Peig5
Peig6
Peig7
Peig8
Peig9
Peig10
10000
8000
6000
4000
2000
0
0
500

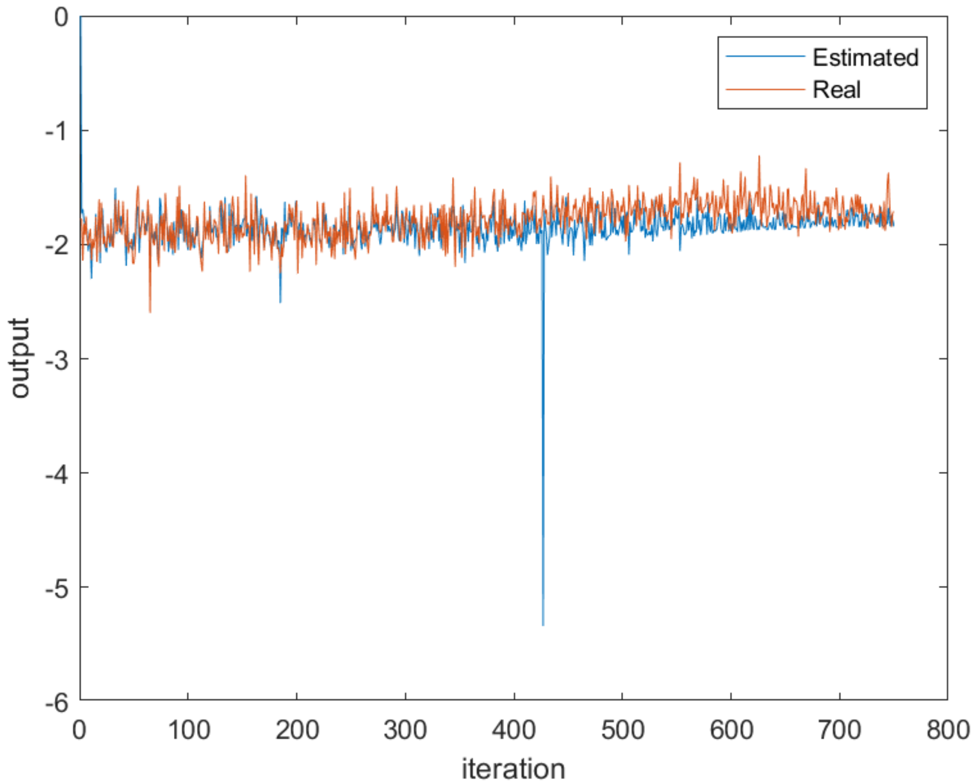

Estimated
Real
output
iteration
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
-4
0
100
200
300
400
500
600
700
800

حالت سوم نویز زیاد






estimated
Peig1
Peig2
Peig3
Peig4
Peig5
Peig6
Peig7
Peig8
Peig9
Peig10
10000
8000
6000
4000
2000
0
0
500


Estimated
Real
output
iteration
0
-1
-2
-3
-4
-5
-6
0
100
200
300
400
500
600
700
800

سوال 2-3

در این روش الگوریتم با استفاده از یک نرخ فراموشی به داده هایی جدید تر وزن و ارزش بیشتری میدهد و داده هایی قدیمی تر را با توجه به نرخ فراموشی تعیین شده وزن کمتری بهشان اختصاص داده و میتوان گفت که بعد از چندین مرحله انها را فراموش میکند.

در اینجا فاکتور فراموشی را برابر 0.99 گرفته ایم اگر فاکتور فراموشی را کمتر از یک در نظر بگیریم الگوریتم به داده هایی که جدید تر به سیستم وارد شده اند وزن بیشتری میدهد این کار برای سیستم هایی که تغییرات دارند مناسب است و باعث میشود که دقت سیستم بیشتر شود اما در سیستم هایی که تغییرات ندارند و تتا های سیستم ثابت اند استفاده از فاکتور فراموشی دقت سیستم را افزایش نمیدهد و شاید حتی باعث کاهش دقت سیستم شود

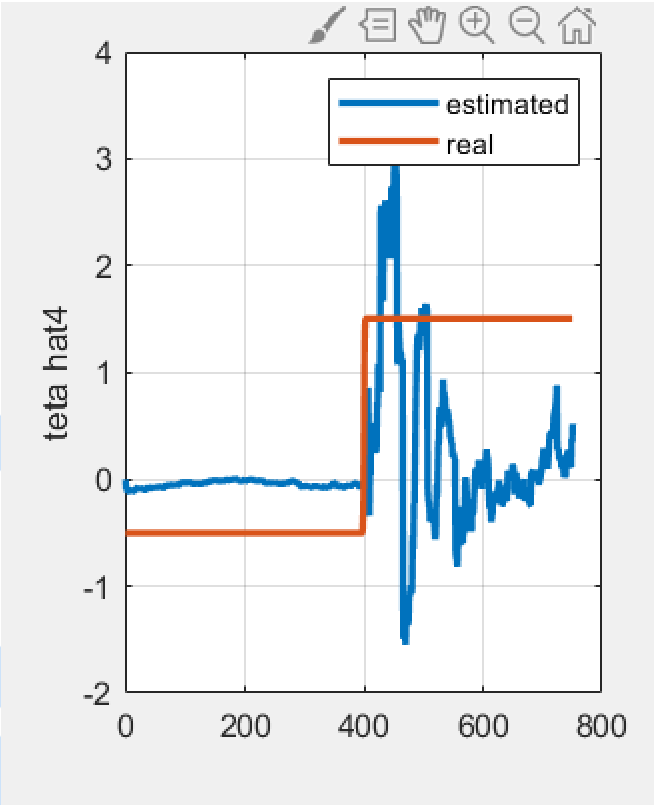
همچنین در سیستم هایی که نیز تغییرات دارند نباید فاکتور فراموشی را خیلی خیلی کم در نظر بگیریم یعنی نباید خیلی از محدوده یک تا 0.9 کمتر کنیم به طور مثال اگر فاکتور فراموشی را بین 0.1 تا 0 در نظر بگیریم سیستم به سرعت داده های قبلی را فراموش میکند و این باعث میشود دقت خروجی خیلی کم شود پس برای مقدار فاکتور فراموشی یک مقدار بهینه ای را متناسب با نرخ تغییرات سیستم تعیین کنیم.

باتوجه به توضیحات استاد و ازمون و خطا مقدار فاکتور فراموشی را برابر با 0.99 در نظر گرفته ایم.

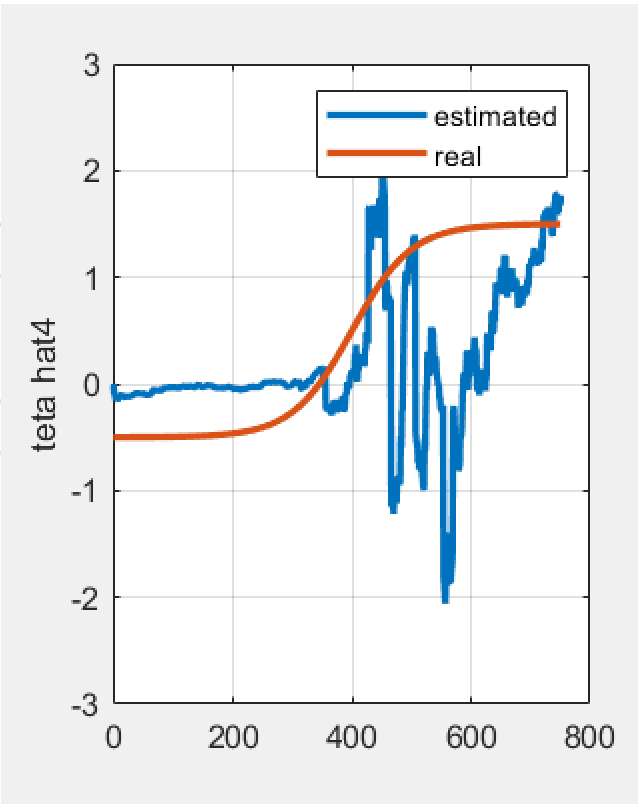
در این بخش نتایج بدست امده تنها برای حالت نویز متوسط در گزارش کار اورده شده است. و نتایج برای نویز کم و نویز زیاد نیز مشابه همین نتایج میباشد و تنها تفاوت انها این است که انها با افزایش نویز مقدار تغییرات تخمین یک مقدار بیشتر میشود و تفاوت قابل توجهی ندارند.

الگوریتم FFRLS نسبت به الگوریتم RLS به نتایج بهتری برای این سوال میرسد اما هنوز نمیتواند به طور کامل تغییرات داده شده را دنبال کند اما نسبت به الگوریتم RLS پارامتر های تخمین زده شده تغییرات کمتری دارند و به نسبت سریعتر همگرا میشوند.

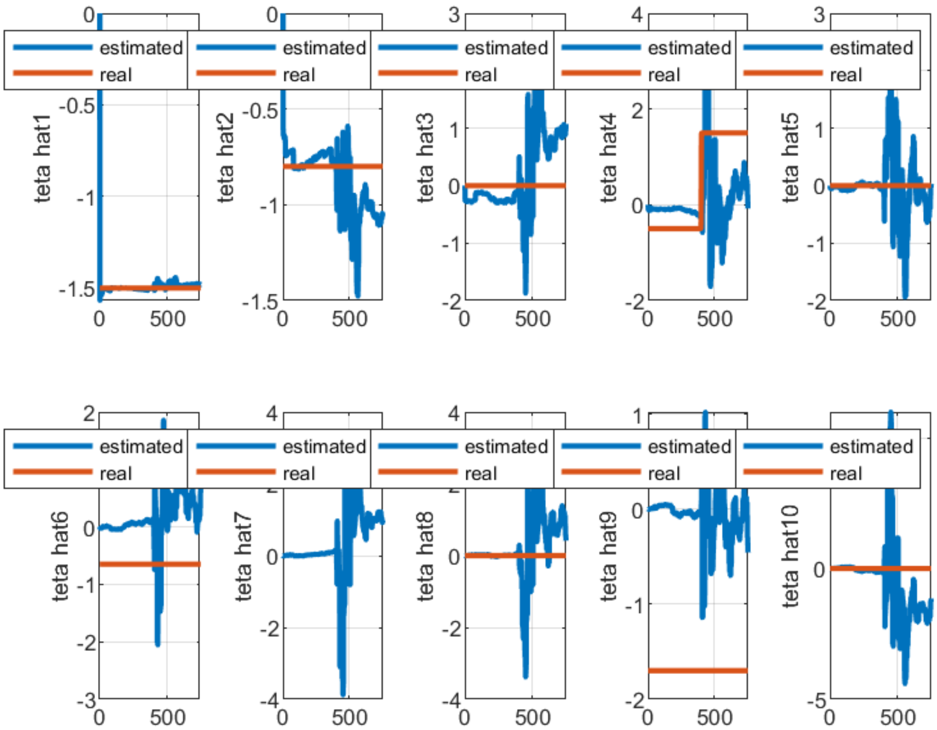
همچنین در حالت g برابر با یک این الگوریتم نمیتواند تغییرات را به خوبی دنبال کند

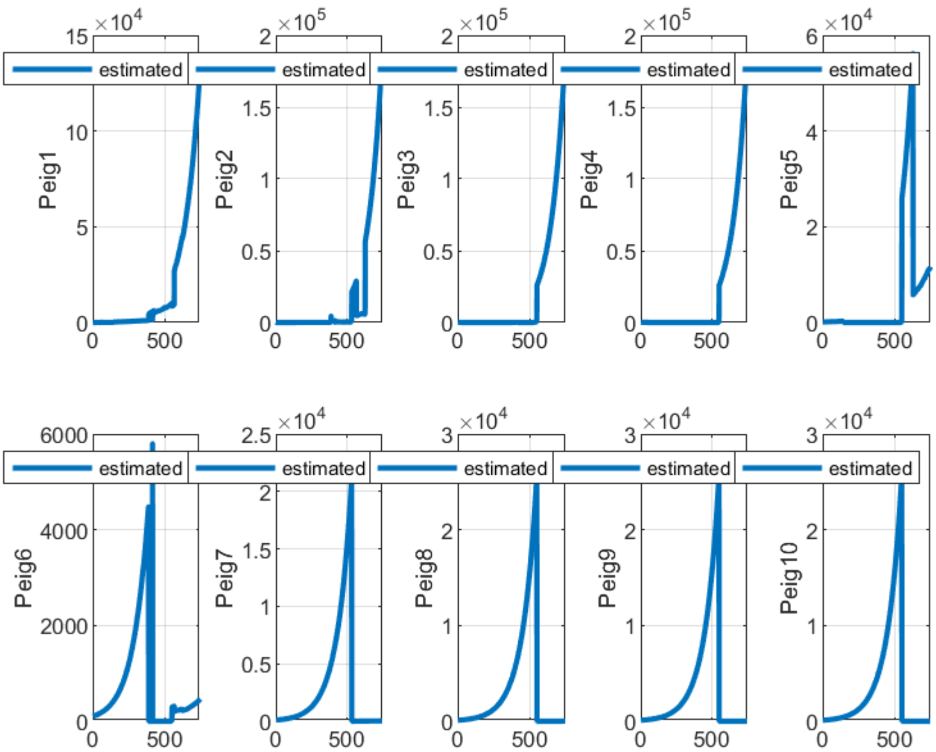


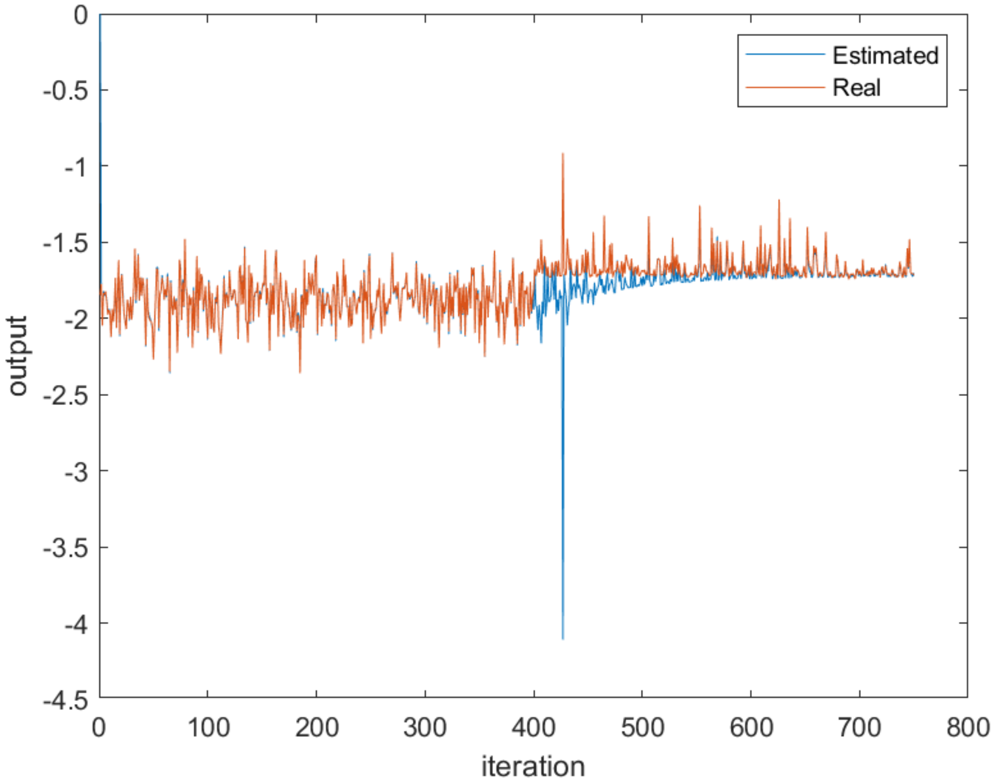
ولی در حالت g برابر با 0.01 این الگوریتم نسبت به الگوریتم RLS نتایج بهتری را دارد و توانسته تا حدودی تغییرات داده شده را دنبال کند.



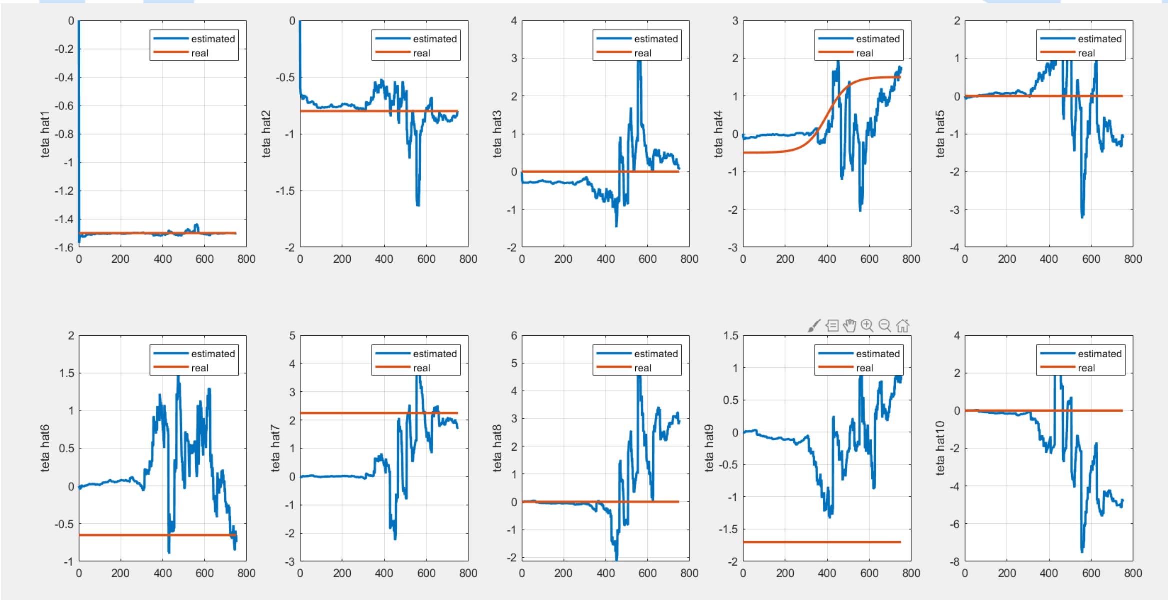
حالت اول $g=1$

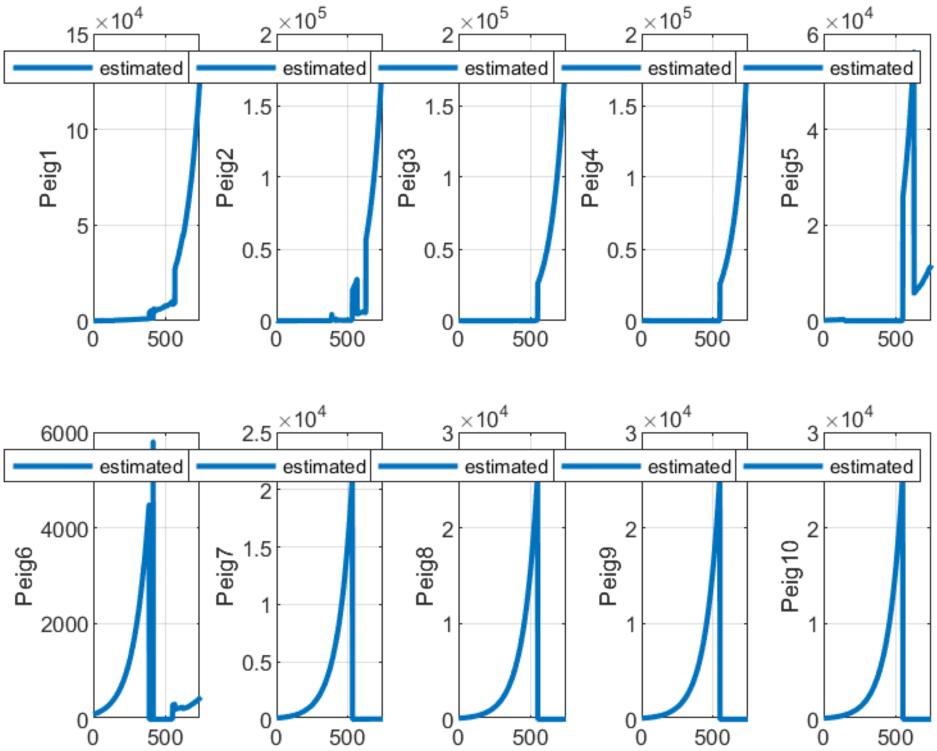

estimated
real
teta hat1
teta hat2
teta hat3
teta hat4
teta hat5
teta hat6
teta hat7
teta hat8
teta hat9
teta hat10

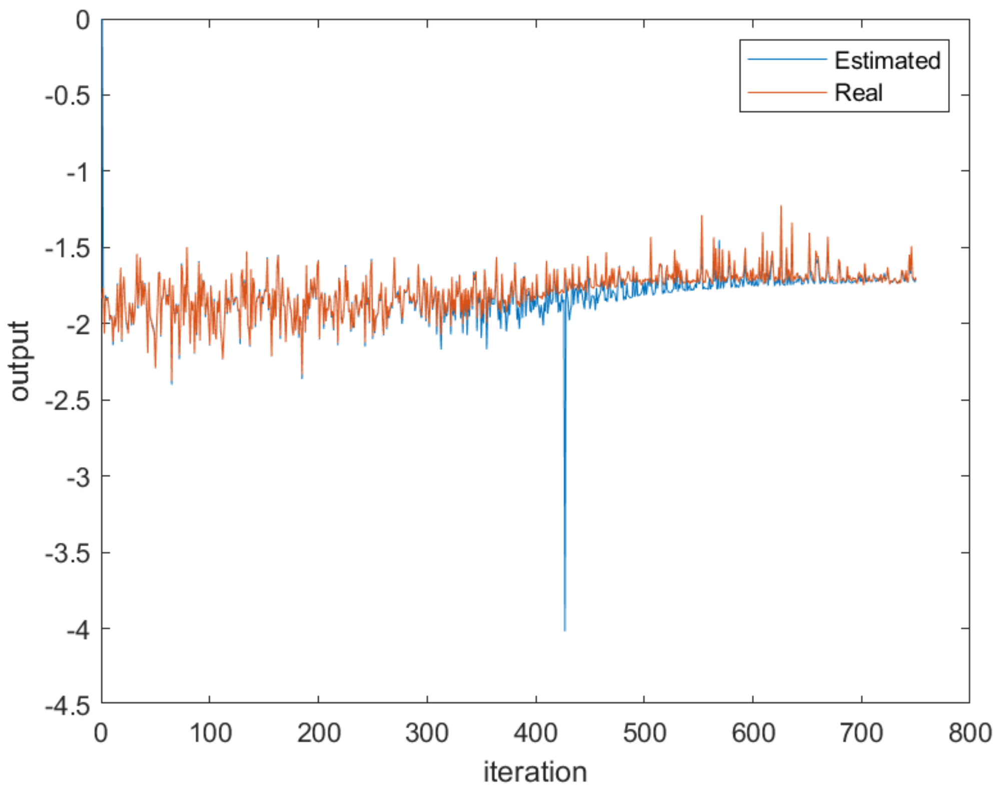

estimated
Peig1
Peig2
Peig3
Peig4
Peig5
Peig6
Peig7
Peig8
Peig9
Peig10



حالت دوم *g=0.01*




estimated
Peig1
Peig2
Peig3
Peig4
Peig5
Peig6
Peig7
Peig8
Peig9
Peig10


Estimated
Real
output
iteration

سوال 2-4

فیلتر کالمن بهترین رویتگر است که با وجود نویز های اندازگیری و نویز فرایند کار میکند البته برای نویز ها شرط های مختلفی مانند گوسی بودن قرار داده است این فیلتر یک روش بازگشتی است در درس شناسایی با استفاده از مفهوم فیلتر کالمن سعی میکنیم که مسئله خود را به این مسئله تبدیل کنیم تا بتوانیم از خواص ان بهره بگیریم

روابط موجود در این بخش به دیدن صورت است

$$\underline{\hat{\theta}}(k) = \underline{\hat{\theta}}(k-1) + \underline{\gamma}(k)e(k), \quad e(k) = y(k) - \underline{x}^T(k)\underline{\hat{\theta}}(k-1) \tag{3.94a}$$

$$\underline{\gamma}(k) = \frac{1}{\underline{x}^T(k)\underline{P}(k-1)\underline{x}(k)+1}\underline{P}(k-1)\underline{x}(k) \tag{3.94b}$$

$$\underline{P}(k) = \left(\underline{I} - \underline{\gamma}(k)\underline{x}^T(k)\right)\underline{P}(k-1) + \underline{W}. \tag{3.94c}$$

وزن های لازم برای این روش به این صورت انتخاب میشوند که هر کدام از متغییر های تتا که تغییرات بیشتری داشته باشند وزن بیشتری میگیرند و متغییر های تتا که تغییرات کمتری داشته باشند وزن کمتری میگیرند به این ترتیب این ضرایب از صفر شدن ماتریس P برای متغییر هایی که تغییر زیادی دارند جلوگیری میکنند همچنین متغییر هایی که تغییر کمی دارند را اجازه میدهند که ضرایب ماتریس P صفر شوند.

نتایج شبیه سازی در زیر نشان داده شده است

این الگوریتم به خوبی توانسته با تغییرات متغییر تتا سه مقاوم باشد و تخمین خود را متناسب با ان به روز رسانی کند همچنین بقیه متغییر های تتا نیز با اینکه تغییرات داشته اند اما تغییرات انها محدود بوده است و به سمت همگرایی میل میکند این نتایج در عکس های زیر قابل مشاهده اند.

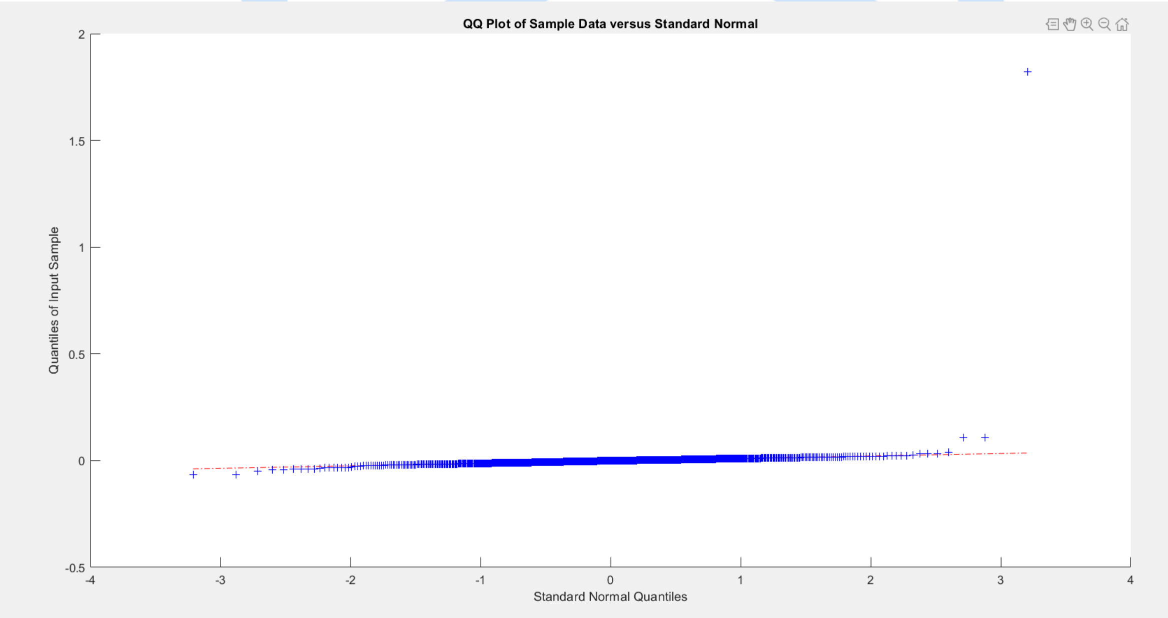
خروجی تخمین بدست امده در این روش را میتوان با استفاده از دو معیار بررسی کرد

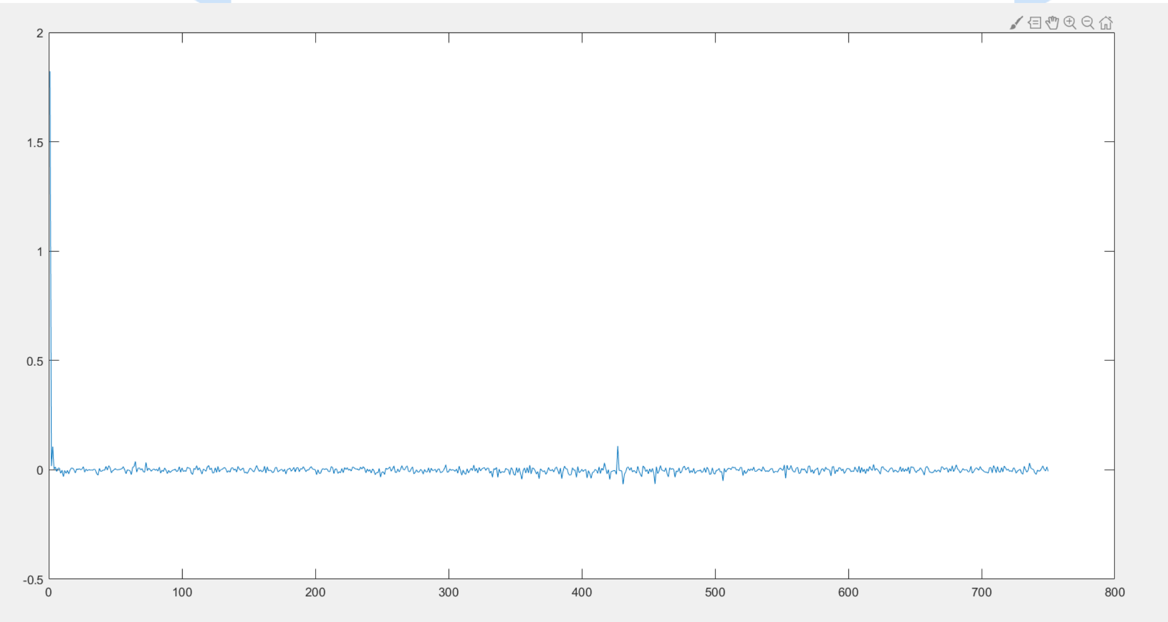
یک با توجه به اینکه مقادیر تتا تخمین زده شده با تقریب خوبی به مقادیر تتا واقعی همگرا شده اند در نتیجه میتوان گفت که تخمین زده شده دقت بالایی داشته است

راه دوم چک کردن مقدار خطا بدست امده در این روش است اگر خطا بدست امده در این روش مشابه با نویز باشد میتوانیم نتیجه بگیریم که تخمین زده شده دقت بالایی دارد و به حالت ایده ال رسیده ایم

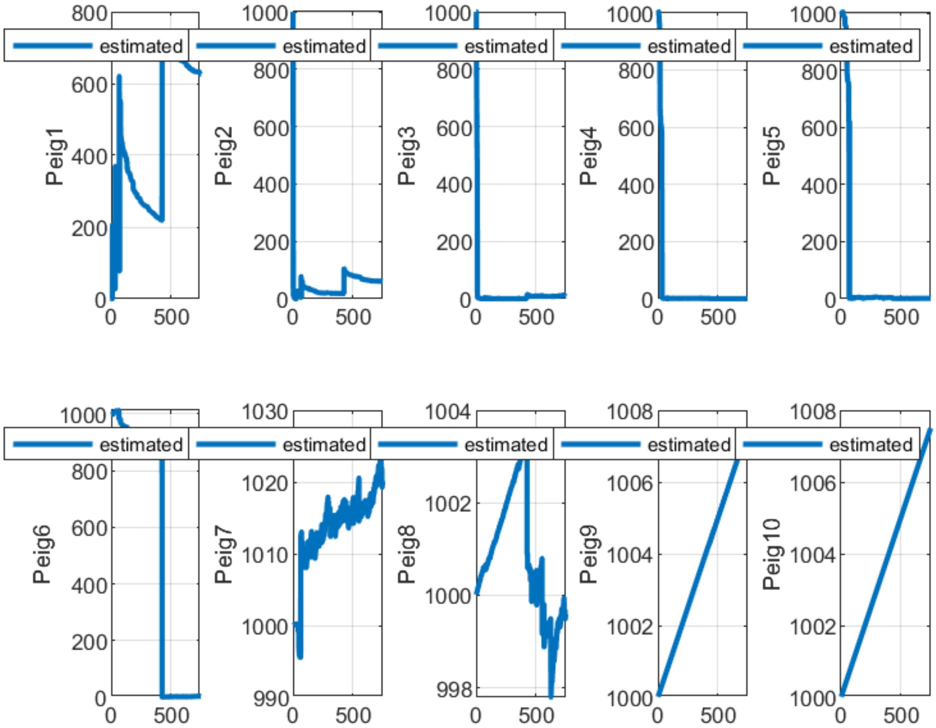
به طور مثال برای نویز متوسط(با کوواریانس 0.01) نتایج زیر را داشته ایم همان گونه که مشاهده میشود مقدار خطا توزیع ای مشابه با مقدار نویز داشته است.

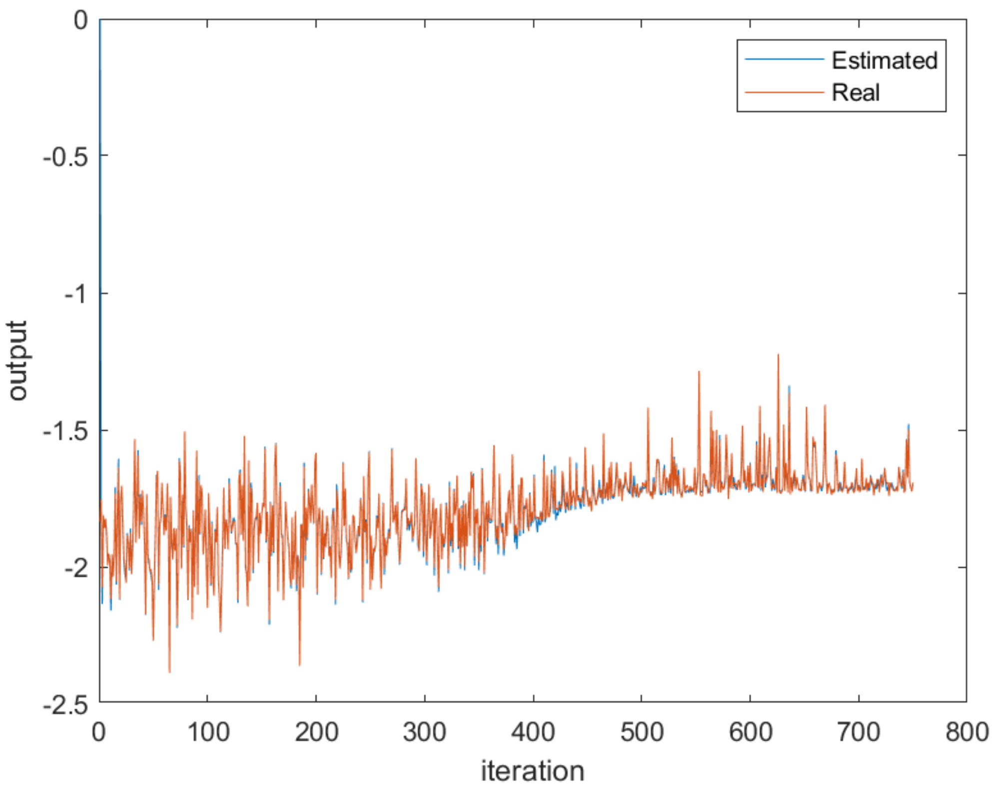
البته این الگوریتم چون به صورت بازگشتی است در نقطه شروع مقدار خطا بیشتر میباشد ولی در بعد از چند مرحله خطا خروجی به مقدار نویز میل میکند.

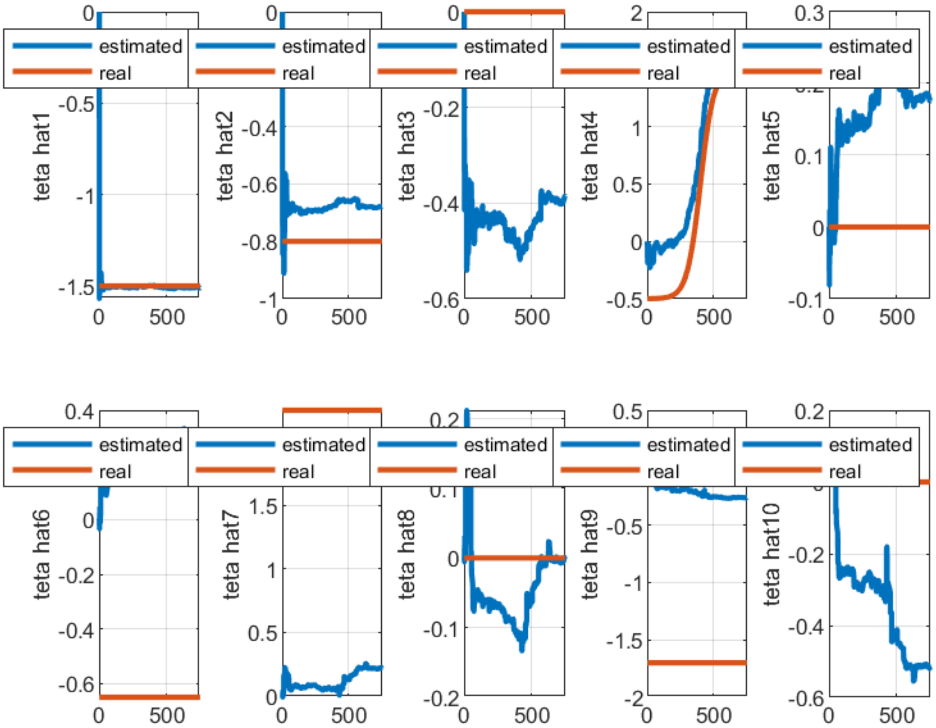


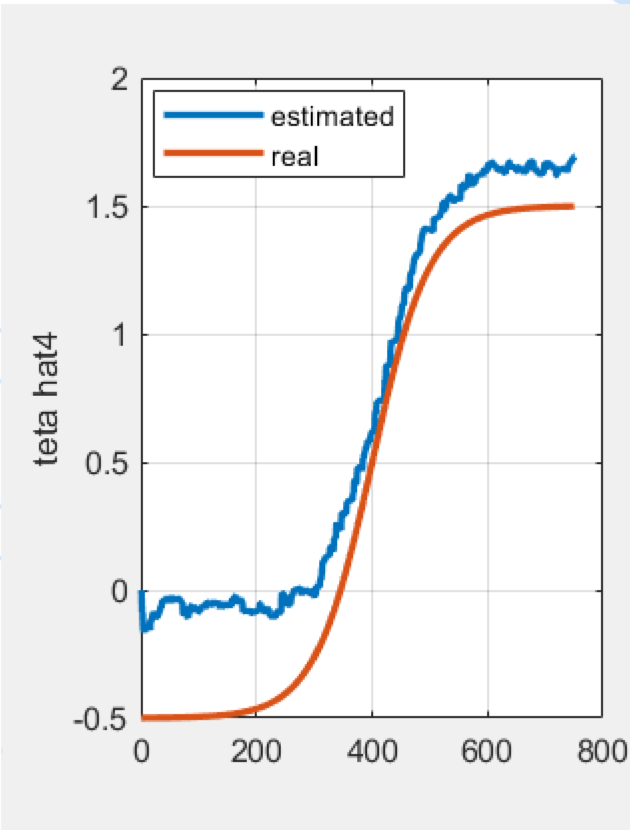


برای $g=0.01$

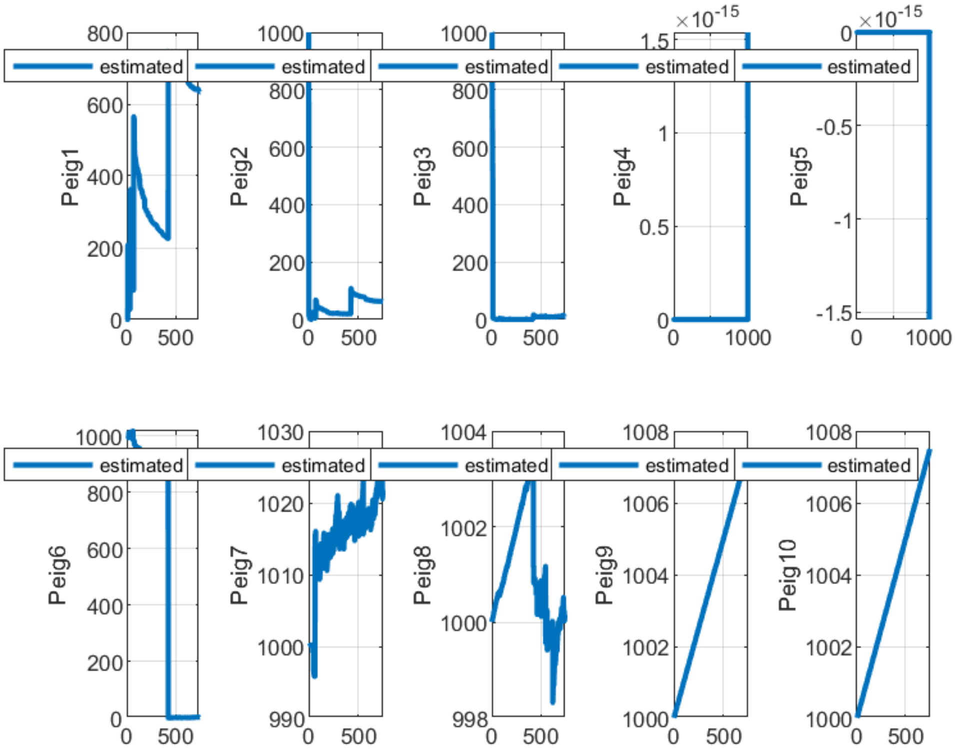

estimated
Peig1
Peig2
Peig3
Peig4
Peig5
Peig6
Peig7
Peig8
Peig9
Peig10

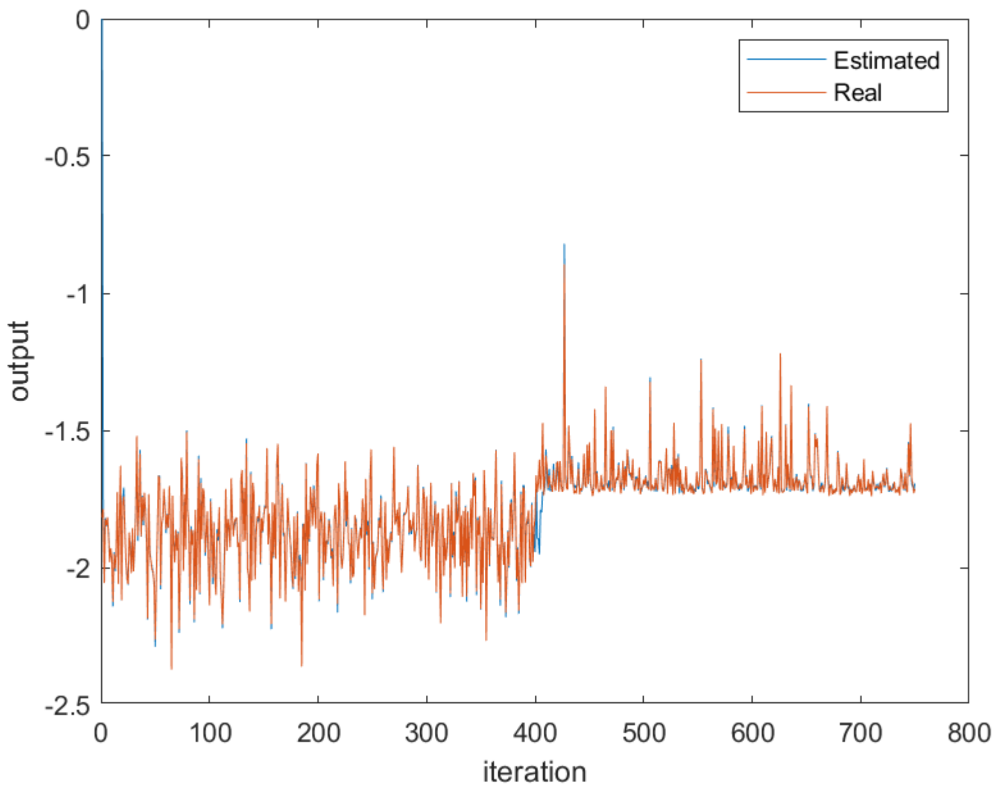

Estimated
Real
output
iteration

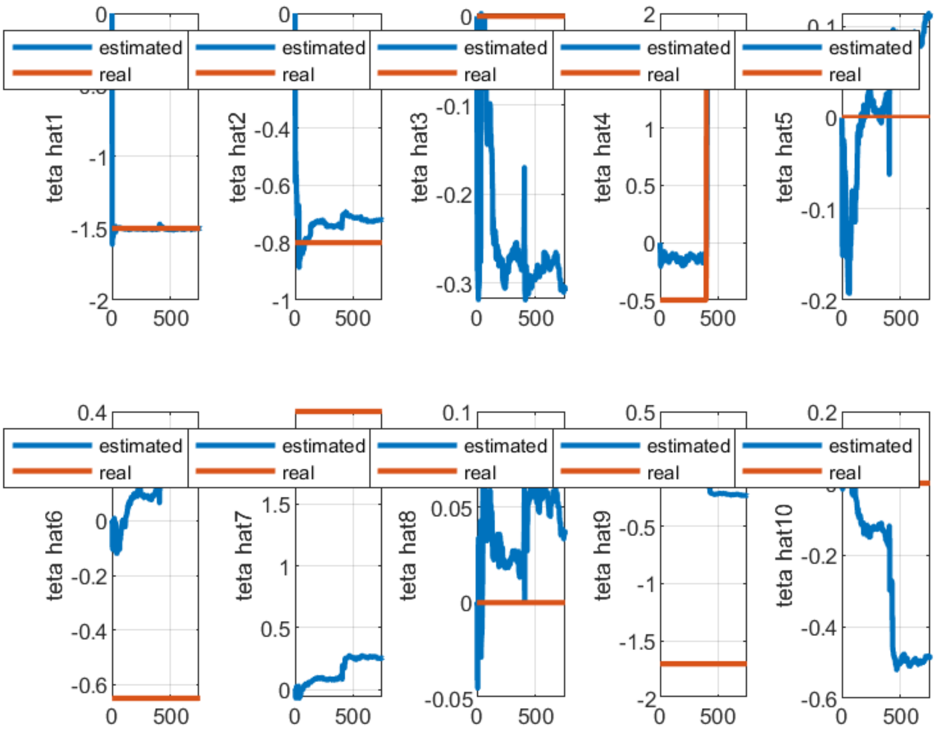

estimated
real
teta hat1
teta hat2
teta hat3
teta hat4
teta hat5
teta hat6
teta hat7
teta hat8
teta hat9
teta hat10

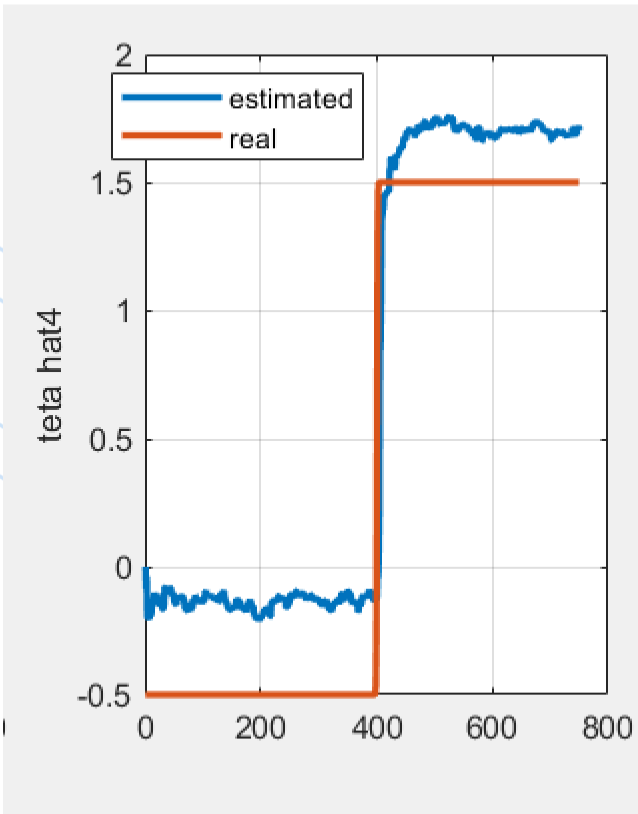

estimated
real
teta hat4
2
1.5
1
0.5
0
-0.5
0
200
400
600
800

برای g=1


estimated
Peig1
Peig2
Peig3
Peig4
Peig5
Peig6
Peig7
Peig8
Peig9
Peig10
$\times 10^{-15}$


Estimated
Real
output
iteration


estimated
real
teta hat1
teta hat2
teta hat3
teta hat4
teta hat5
teta hat6
teta hat7
teta hat8
teta hat9
teta hat10


estimated
real
teta hat4

سوال سه

قسمت اول

منطقی بودن مقدار تتا به این معنی است که

قسمت دوم

در نمودار های زیر مقدار error2 تفاوت مقدار خروجی بدون نویز با خروجی تخمین زده شده را نمایش میدهد

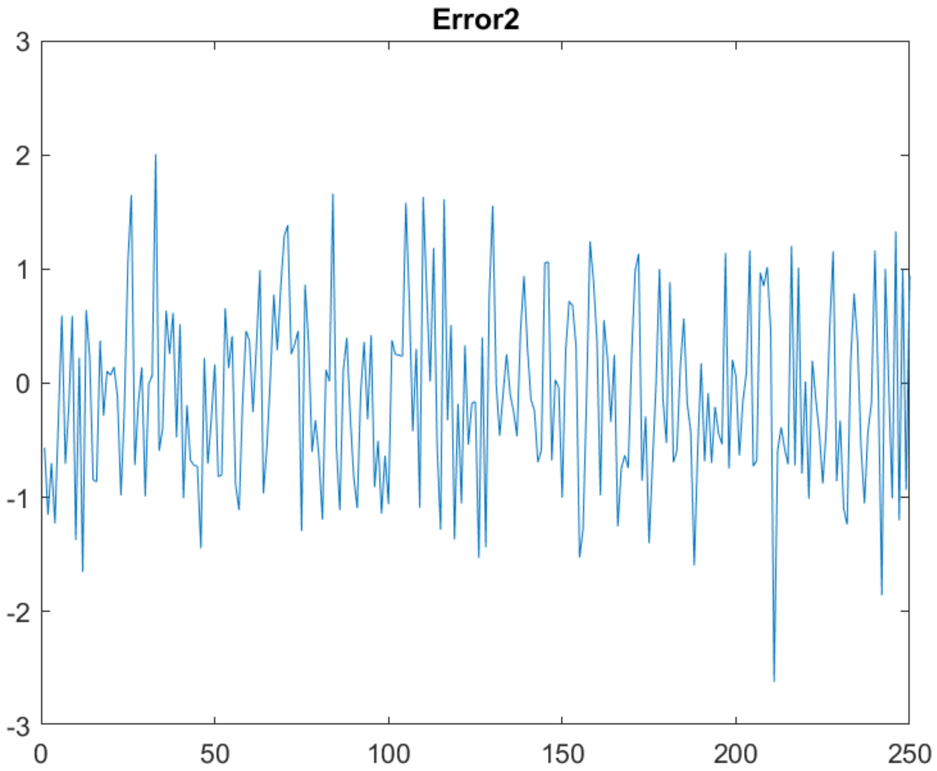
در نمودار های زیر مقدار error1 تفاوت مقدار خروجی با نویز با خروجی تخمین زده شده را نمایش میدهد

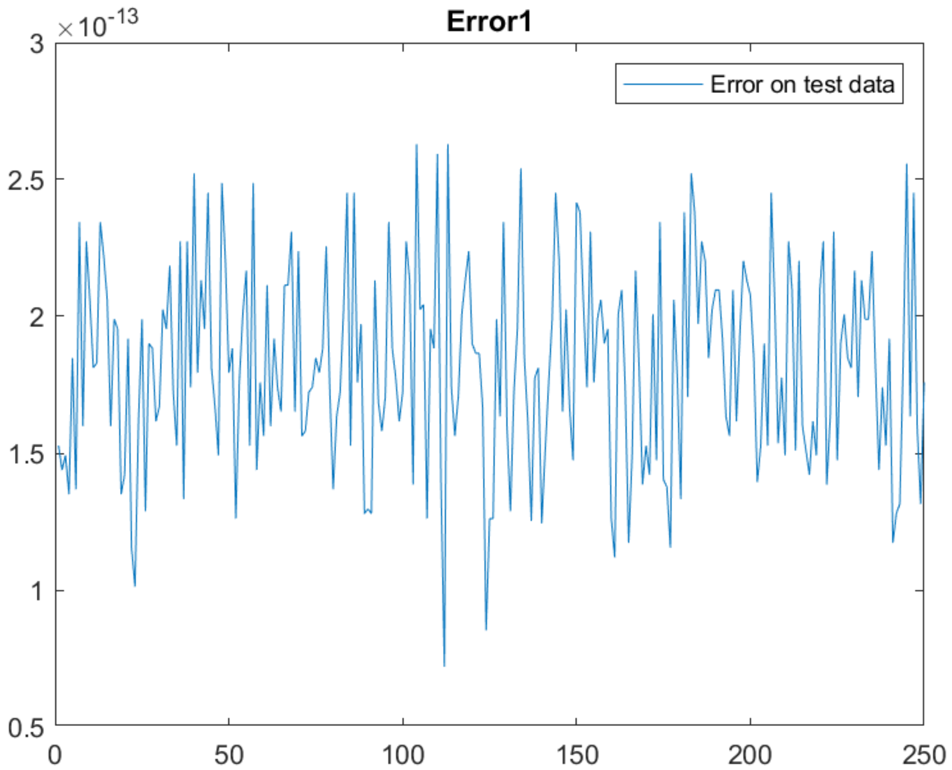
همچنین در نمودار سوم مقدار خروجی با نویز و خروجی بدون نویز و خروجی تخمین زده شده را نشان داده شده است

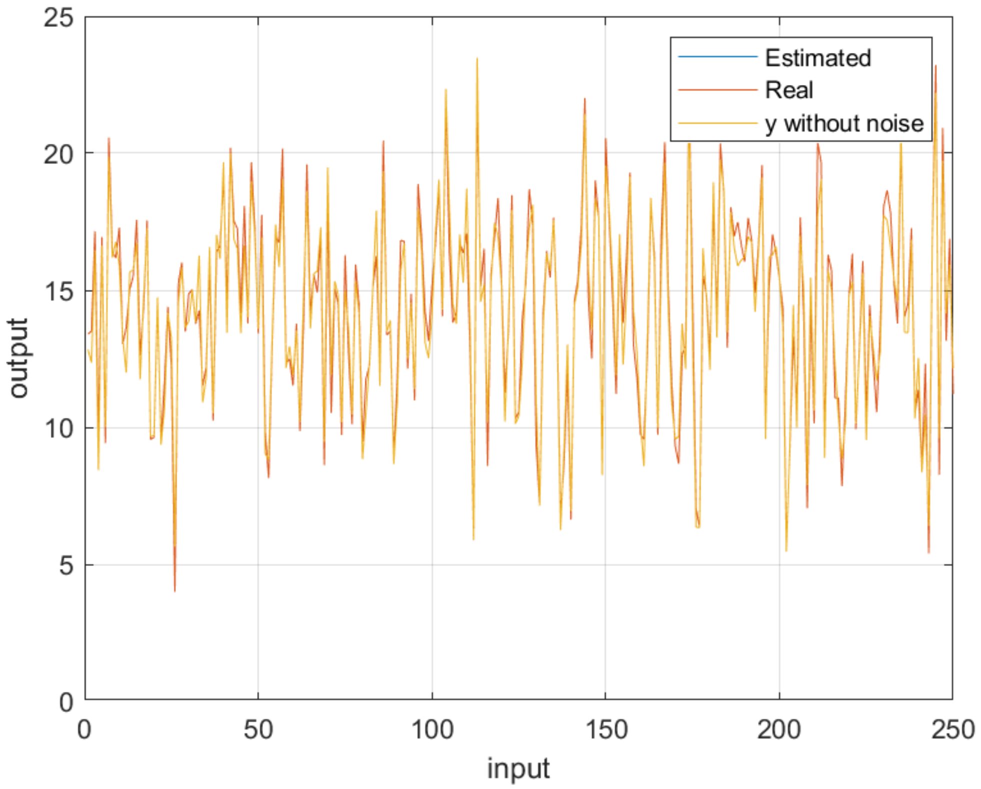
در این مسئله داده های ورودی سوم به داده های ورودی اول و دوم وابسته اند به همین خاطر نمیتوانیم سه رگرسور مستقل داشته باشیم در واقع رگرسور سوم نویز را دنبال میکند به همین خاطر دقت خروجی شبکه خیلی خوب است و مقدار تتا های تخمین زده شده مشابه تتا های واقعی است اما سیستم به درستی شناسایی نشده است و مقدار y بدون نویز با مقدار y تخمین زده شده بسیار تفاوت دارد

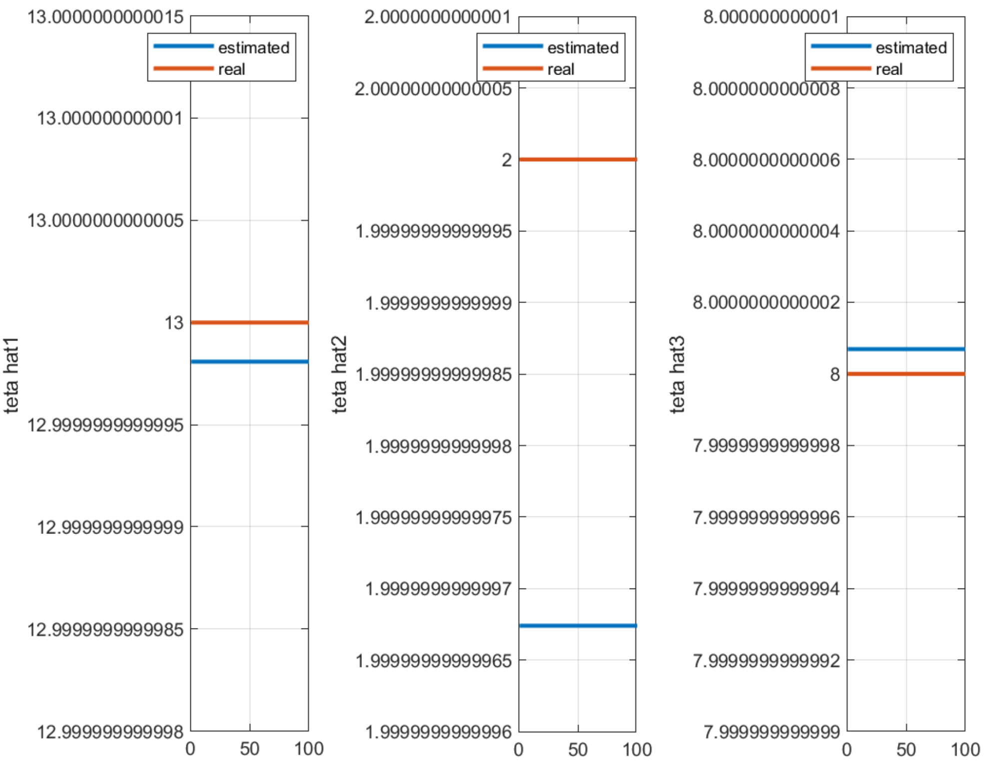
به بیان دیگر در این سیستم y تخمین زده شده به مقدار y با نویز شبیه شده است اما مقدار y تخمین زده شده با مقدار y بدون نویز بسیار متفاوت است.

نویز کم

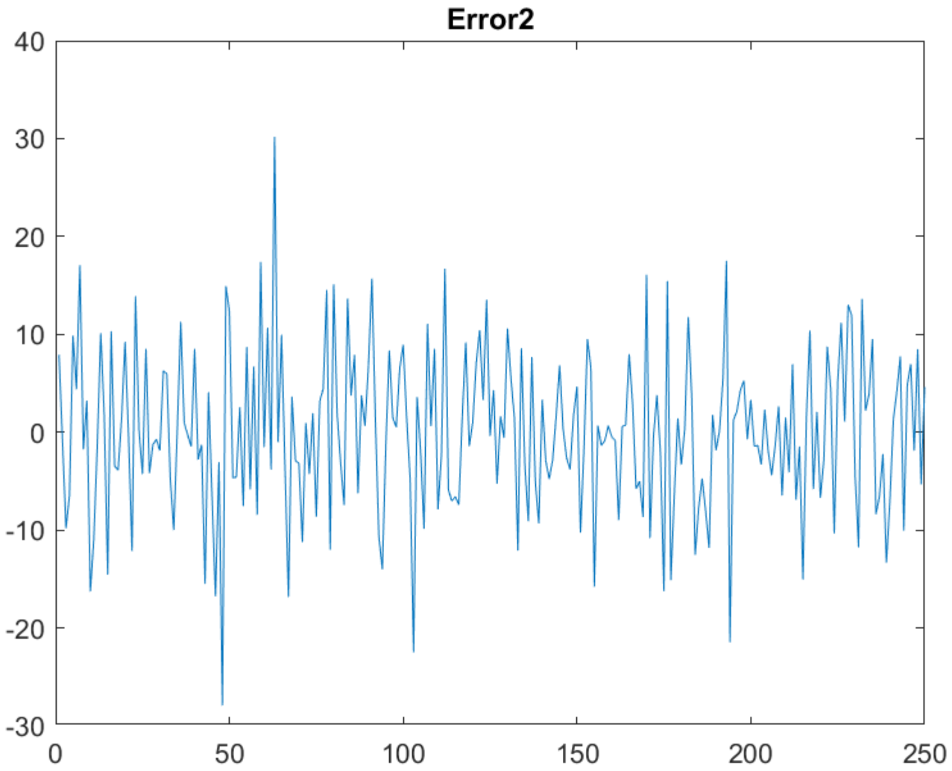

Error2
3
2
1
0
-1
-2
-3
0
50
100
150
200
250

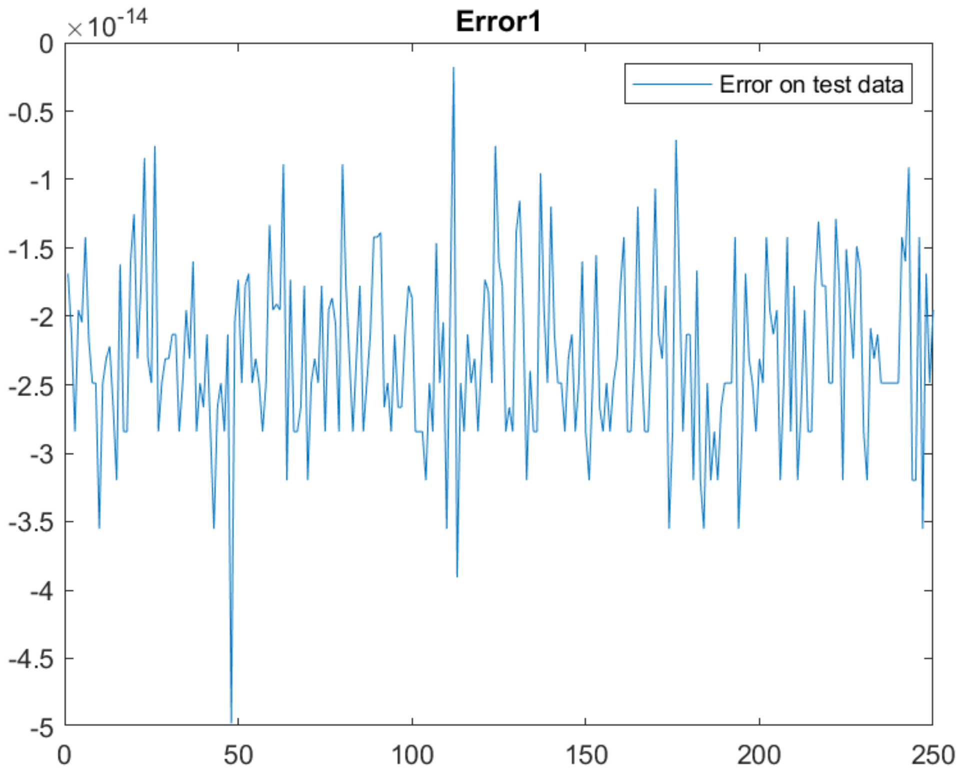

Error1
×10^-13
Error on test data
3
2.5
2
1.5
1
0.5
0
50
100
150
200
250

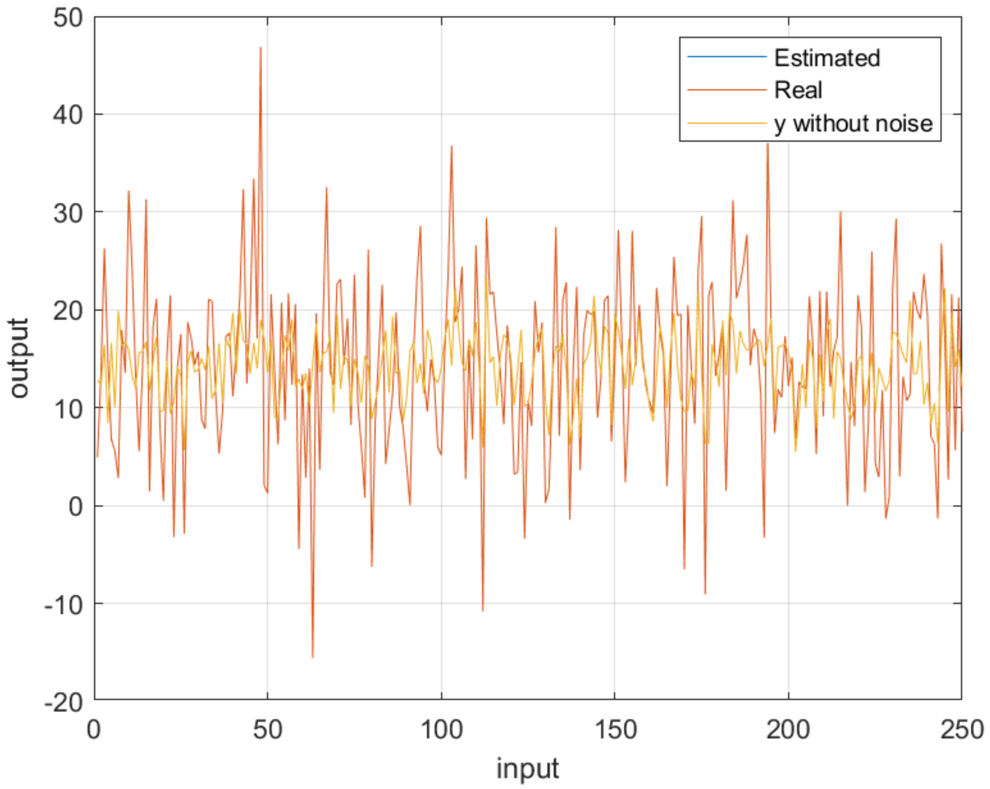

Estimated
Real
y without noise
output
input
0
5
10
15
20
25
0
50
100
150
200
250

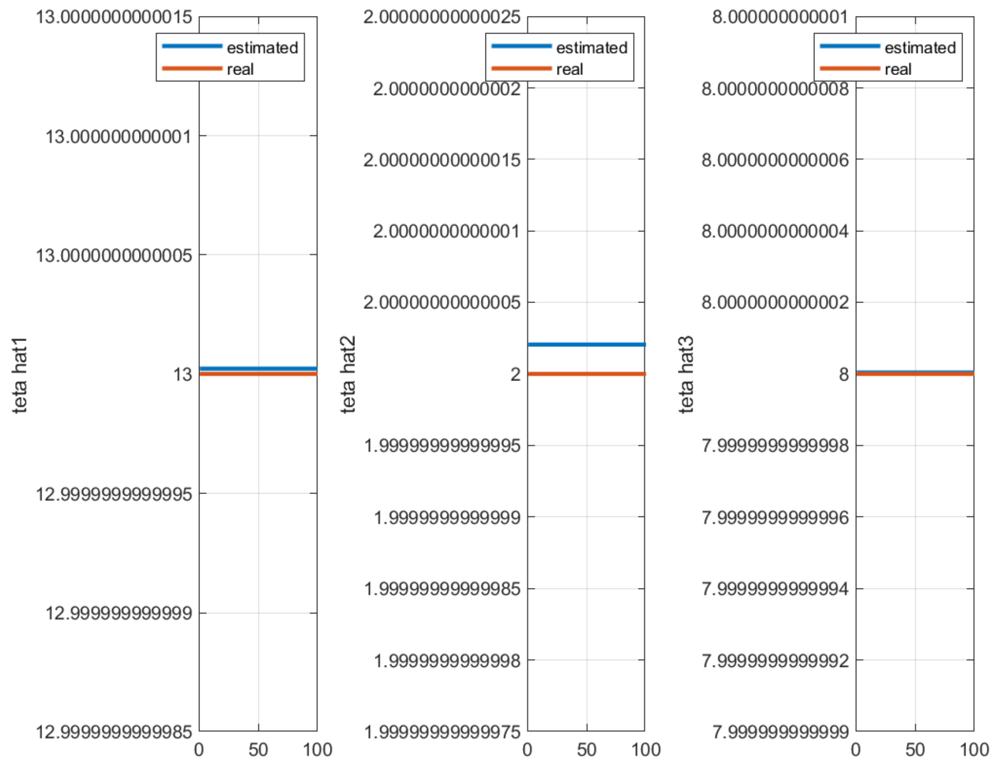

estimated
real
teta hat1
13.0000000000015
13.000000000001
13.0000000000005
13
12.9999999999995
12.999999999999
12.9999999999985
12.999999999998
teta hat2
2.0000000000001
2.00000000000005
2
1.99999999999995
1.9999999999999
1.99999999999985
1.9999999999998
1.99999999999975
1.9999999999997
1.99999999999965
1.9999999999996
teta hat3
8.000000000001
8.0000000000008
8.0000000000006
8.0000000000004
8.0000000000002
8
7.9999999999998
7.9999999999996
7.9999999999994
7.9999999999992
7.999999999999
0
50
100

نویز متوسط

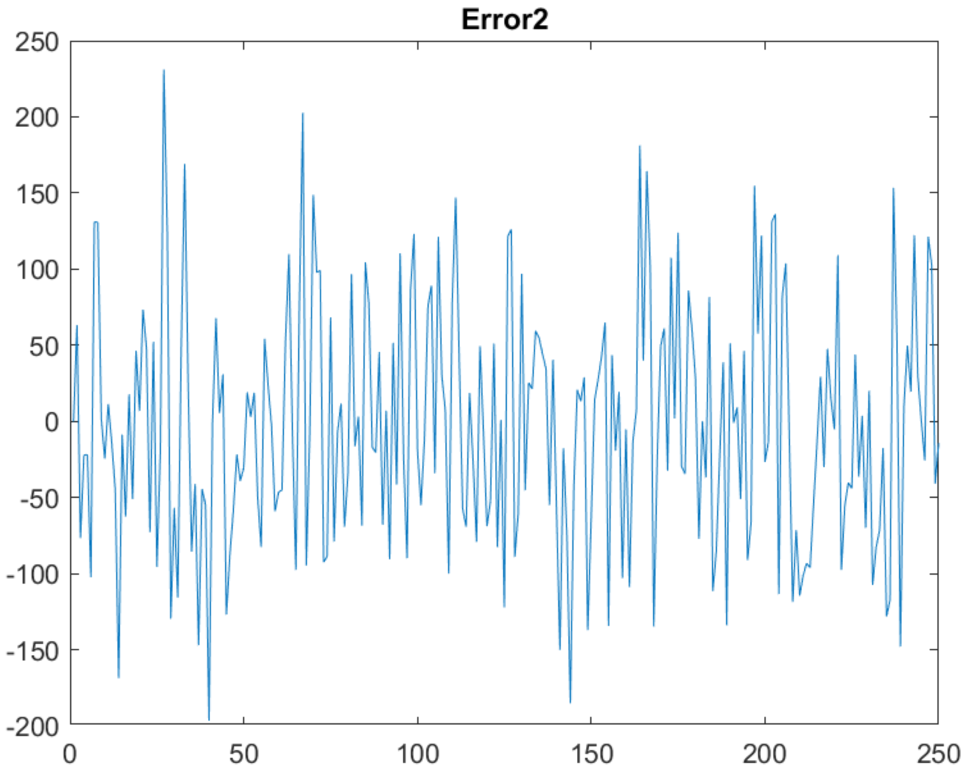

Error2
40
30
20
10
0
-10
-20
-30
0
50
100
150
200
250

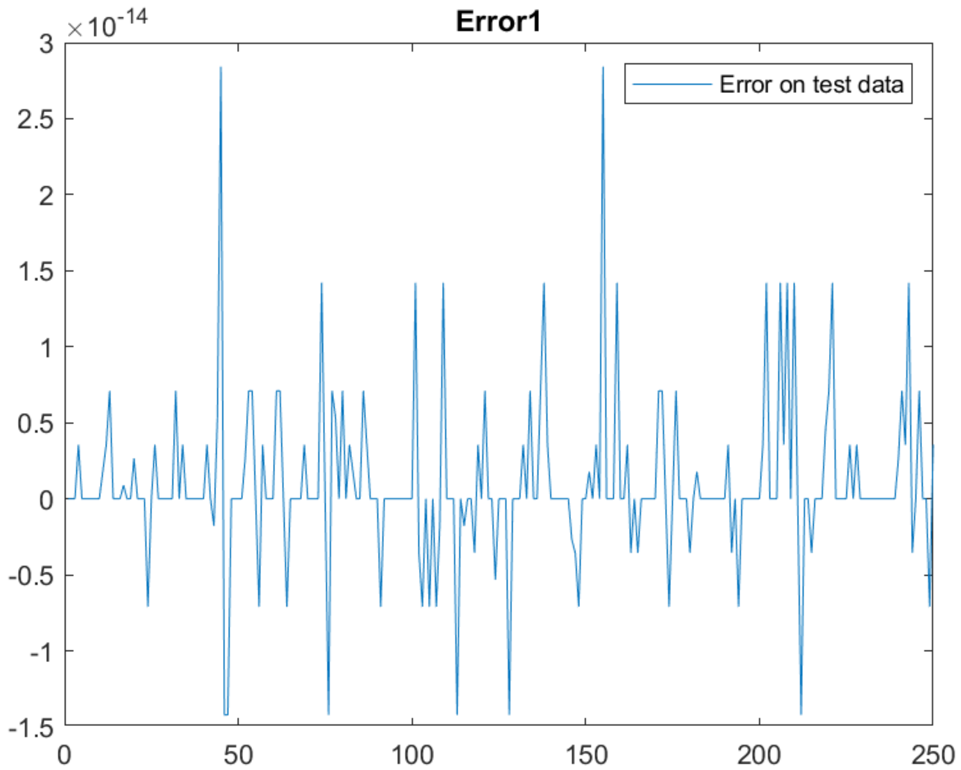

Error1
×10^-14
Error on test data
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
-4
-4.5
-5
0
50
100
150
200
250

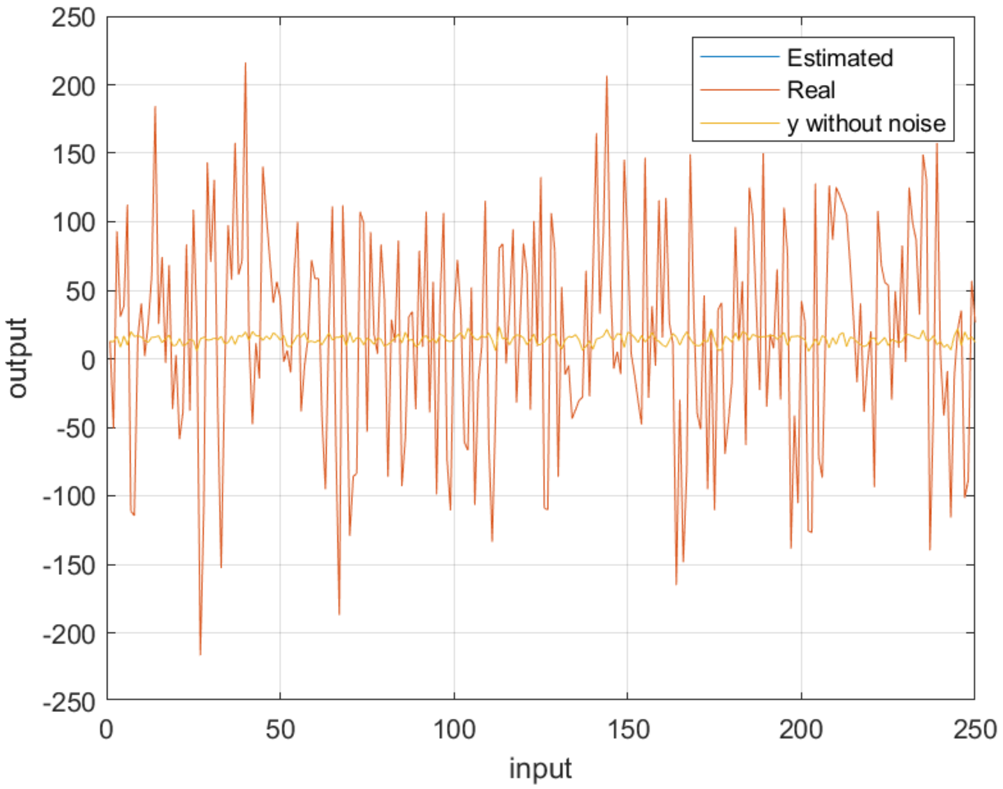

Estimated
Real
y without noise
output
input

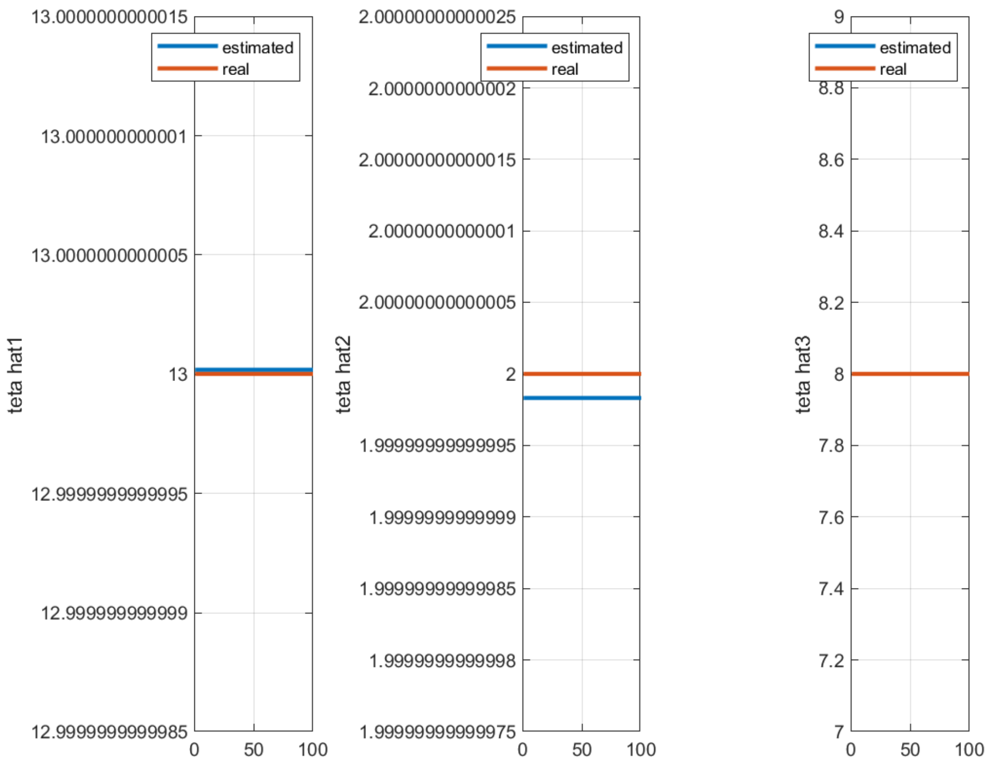

estimated
real
teta hat1
teta hat2
teta hat3

نویز زیاد


Error2
250
200
150
100
50
0
-50
-100
-150
-200
0
50
100
150
200
250

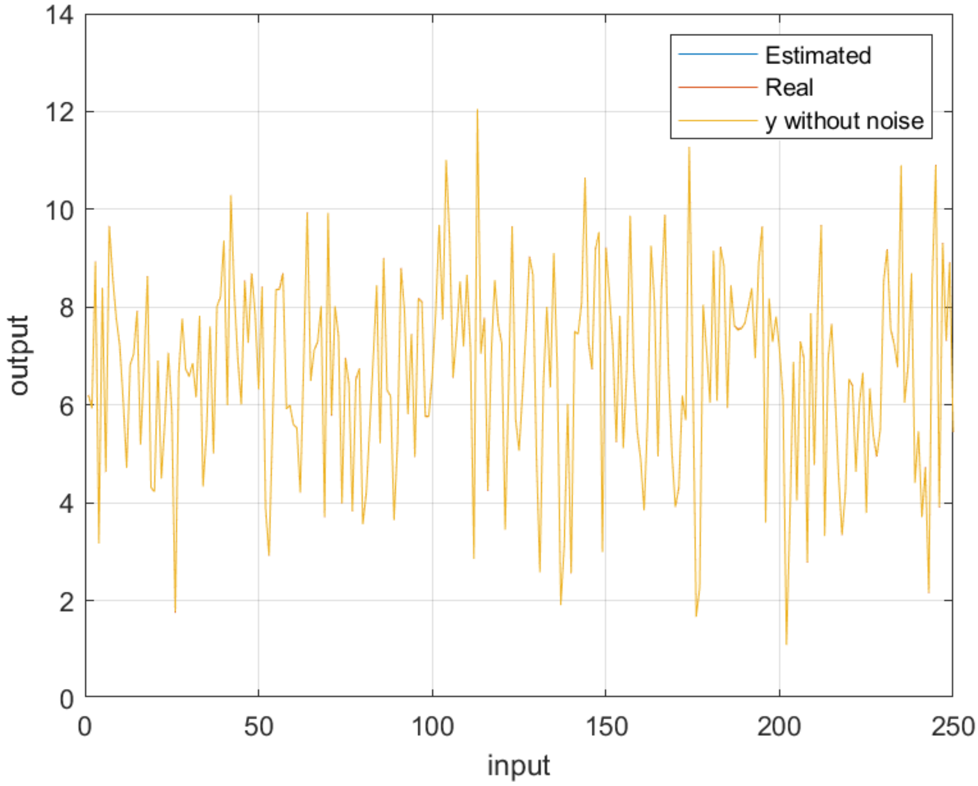

Error1
×10^-14
3
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
0
50
100
150
200
250
Error on test data


250
200
150
100
50
0
-50
-100
-150
-200
-250
output
0
50
100
150
200
250
input
Estimated
Real
y without noise

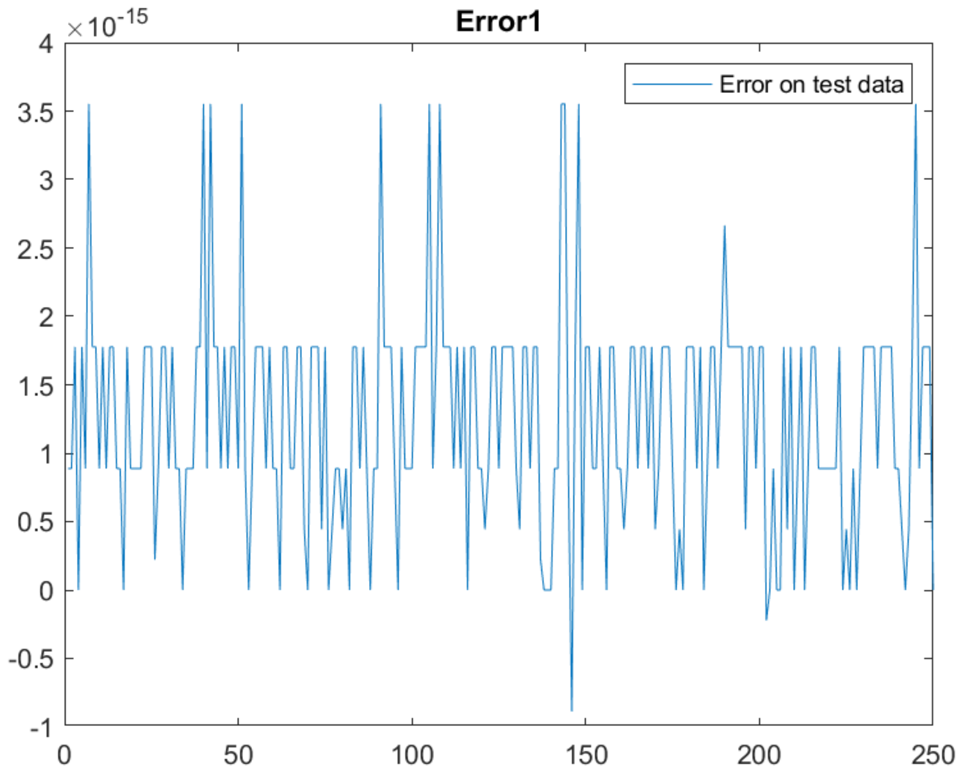

13.0000000000015
13.000000000001
13.0000000000005
13
12.9999999999995
12.999999999999
12.9999999999985
teta hat1
estimated
real
2.00000000000025
2.0000000000002
2.00000000000015
2.0000000000001
2.00000000000005
2
1.99999999999995
1.9999999999999
1.99999999999985
1.9999999999998
1.99999999999975
teta hat2
9
8.8
8.6
8.4
8.2
8
7.8
7.6
7.4
7.2
7
teta hat3
0
50
100

برای حل این مشکل باید به تعداد مناسب رگرسور انتخاب کرد این مسئله با انتخاب 2 رگرسور داریم همانگونه که مشاهده میشود مقدار خطا نسبت به حالت قبلی بسیار کمتر شده است.

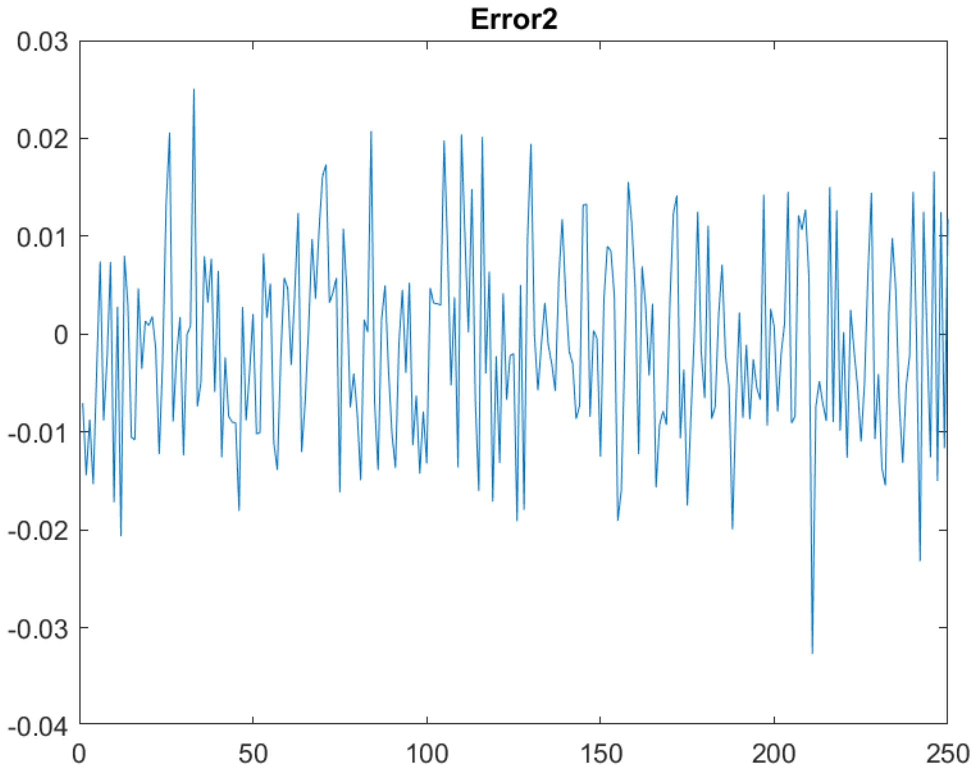
برای نویز کم

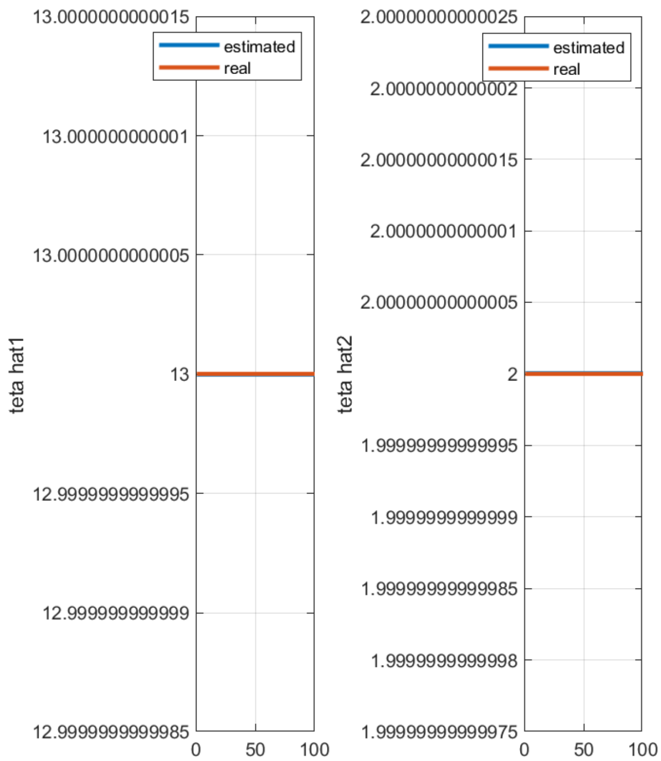


Estimated
Real
y without noise
output
input

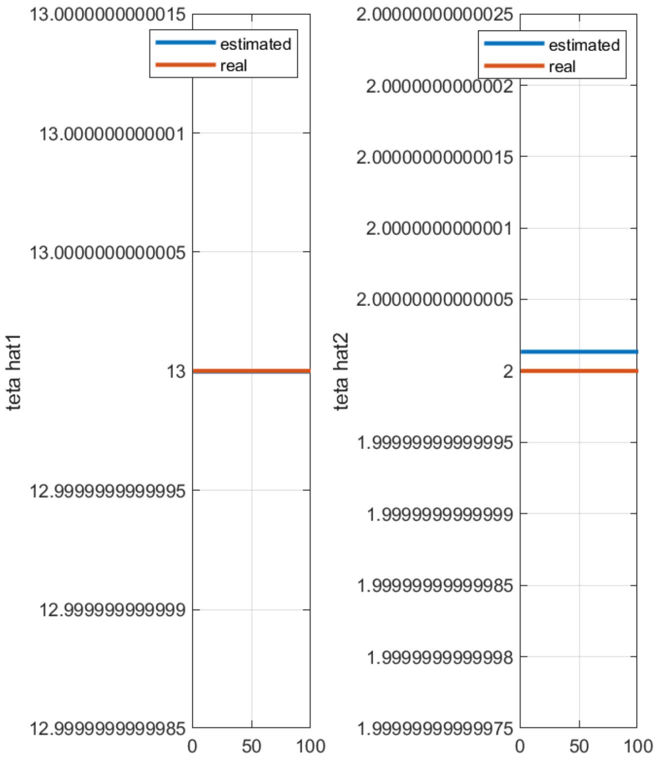


Error1
Error on test data

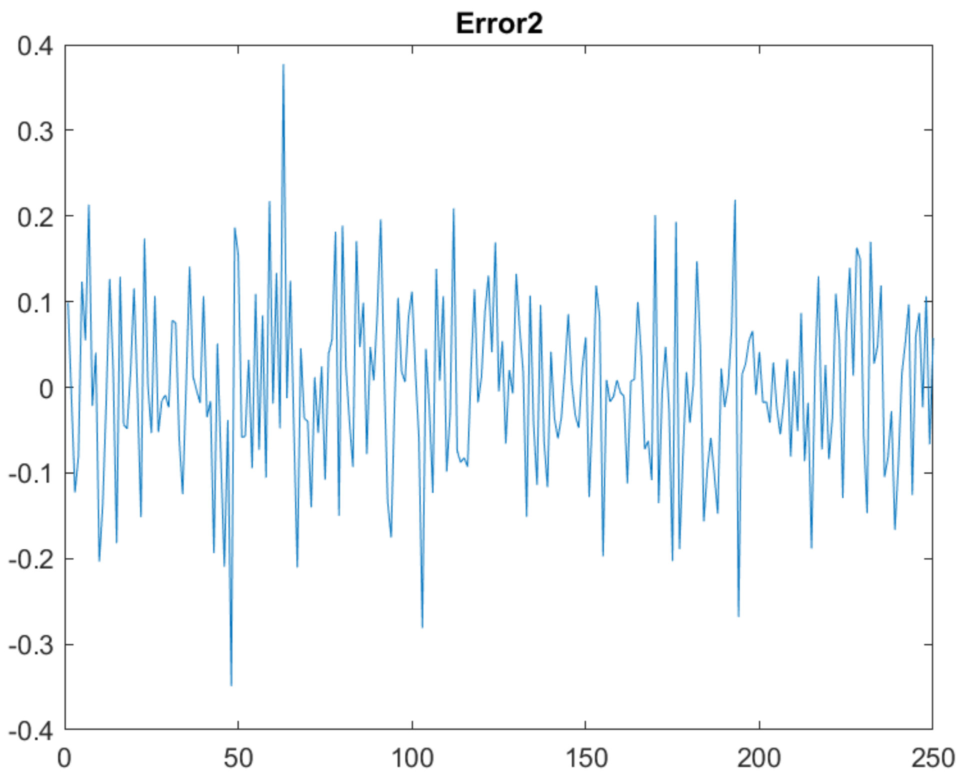

Error2
0.03
0.02
0.01
0
-0.01
-0.02
-0.03
-0.04
0
50
100
150
200
250


estimated
real
13.0000000000015
13.000000000001
13.0000000000005
13
12.9999999999995
12.999999999999
12.9999999999985
teta hat1
0
50
100
estimated
real
2.00000000000025
2.0000000000002
2.00000000000015
2.0000000000001
2.00000000000005
2
1.99999999999995
1.9999999999999
1.99999999999985
1.9999999999998
1.99999999999975
teta hat2
0
50
100

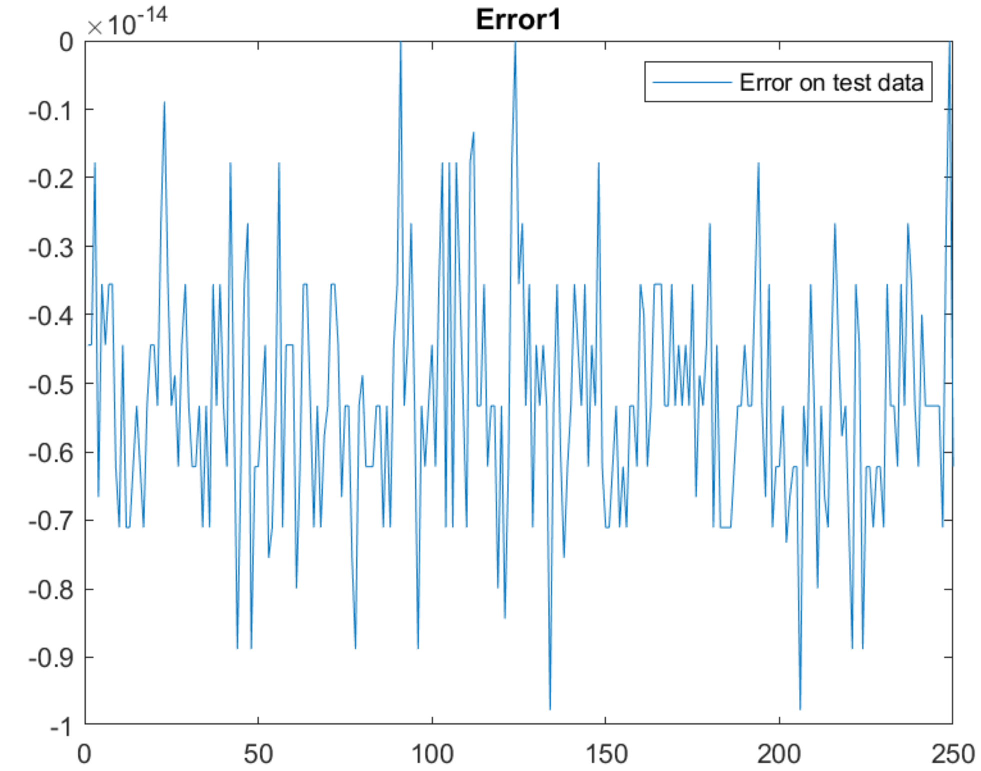
نویز متوسط



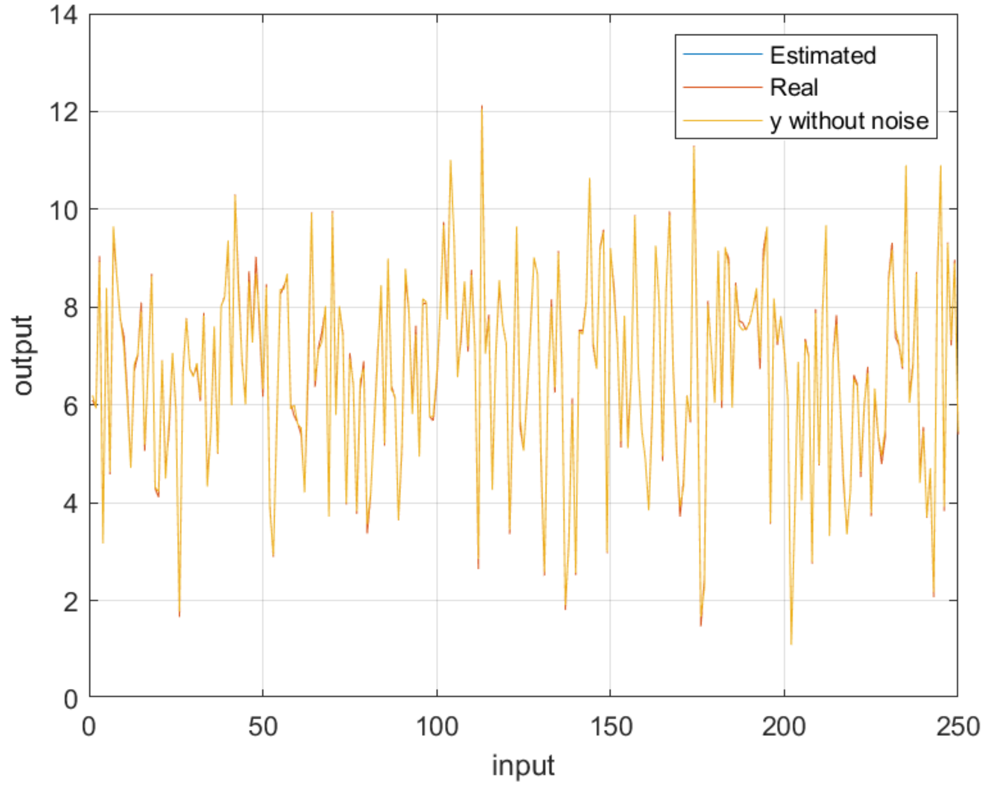
13.0000000000015
13.000000000001
13.0000000000005
13
12.9999999999995
12.999999999999
12.9999999999985
teta hat1
estimated
real
0
50
100
2.00000000000025
2.0000000000002
2.00000000000015
2.0000000000001
2.00000000000005
2
1.99999999999995
1.9999999999999
1.99999999999985
1.9999999999998
1.99999999999975
teta hat2



Error2
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
0
50
100
150
200
250

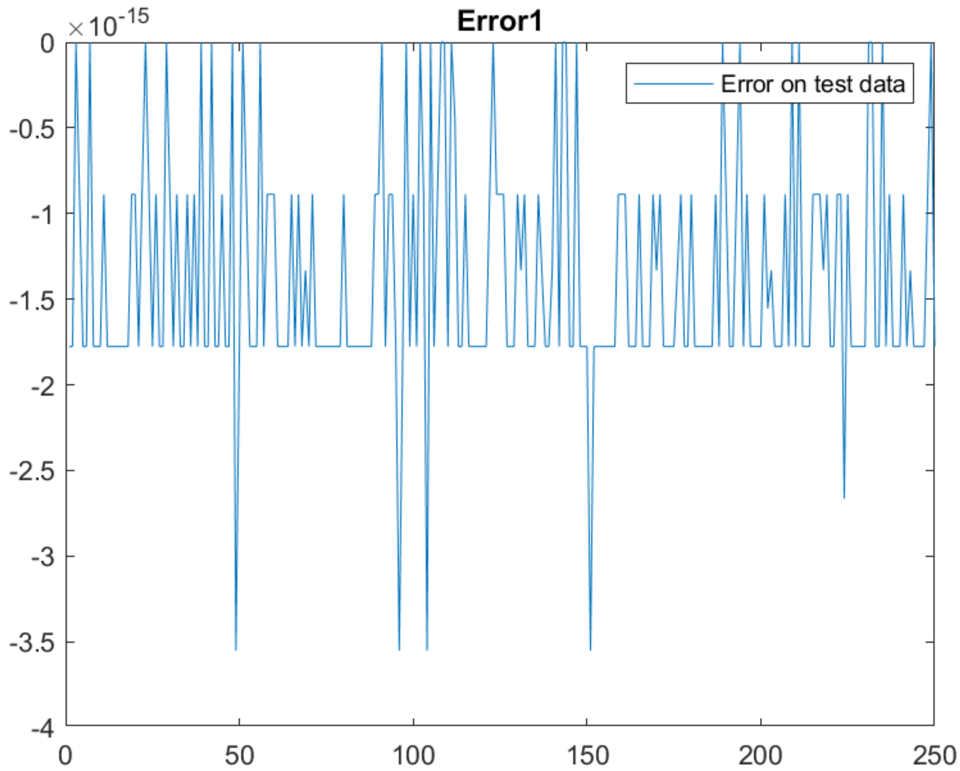


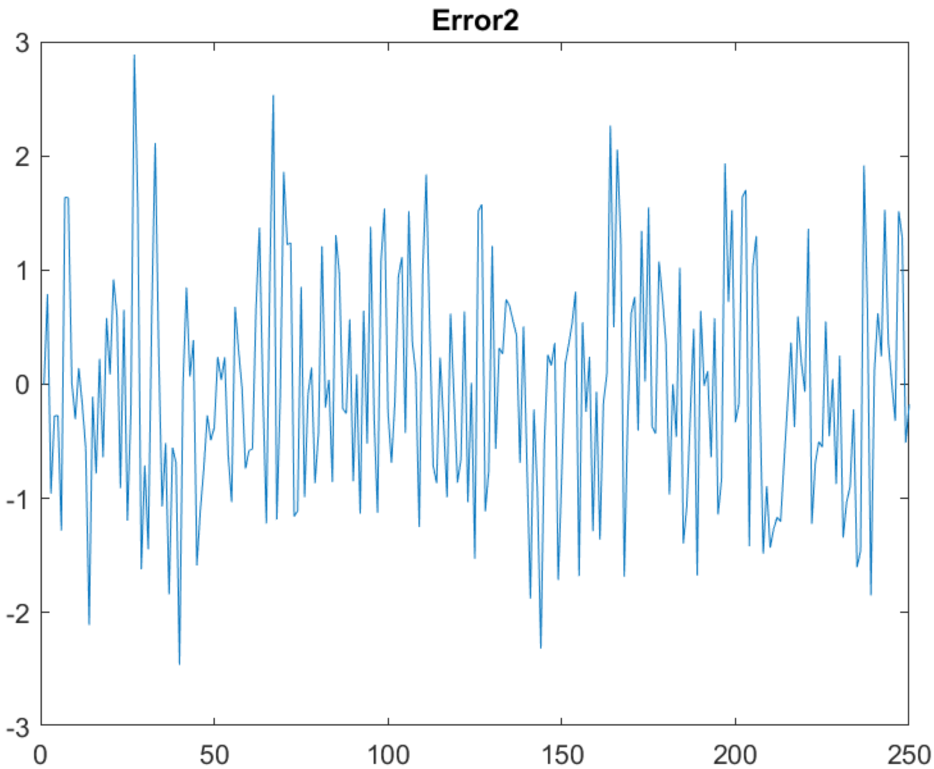
Error1
×10^-14
Error on test data
0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.6
-0.7
-0.8
-0.9
-1
0
50
100
150
200
250

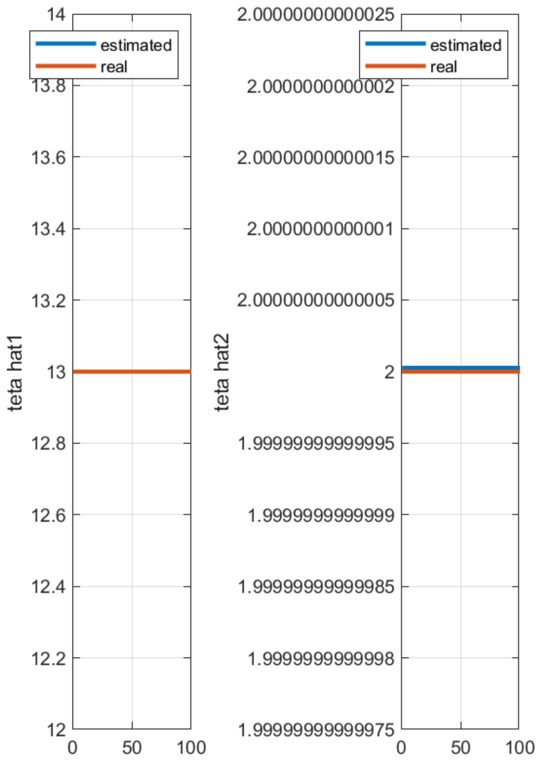


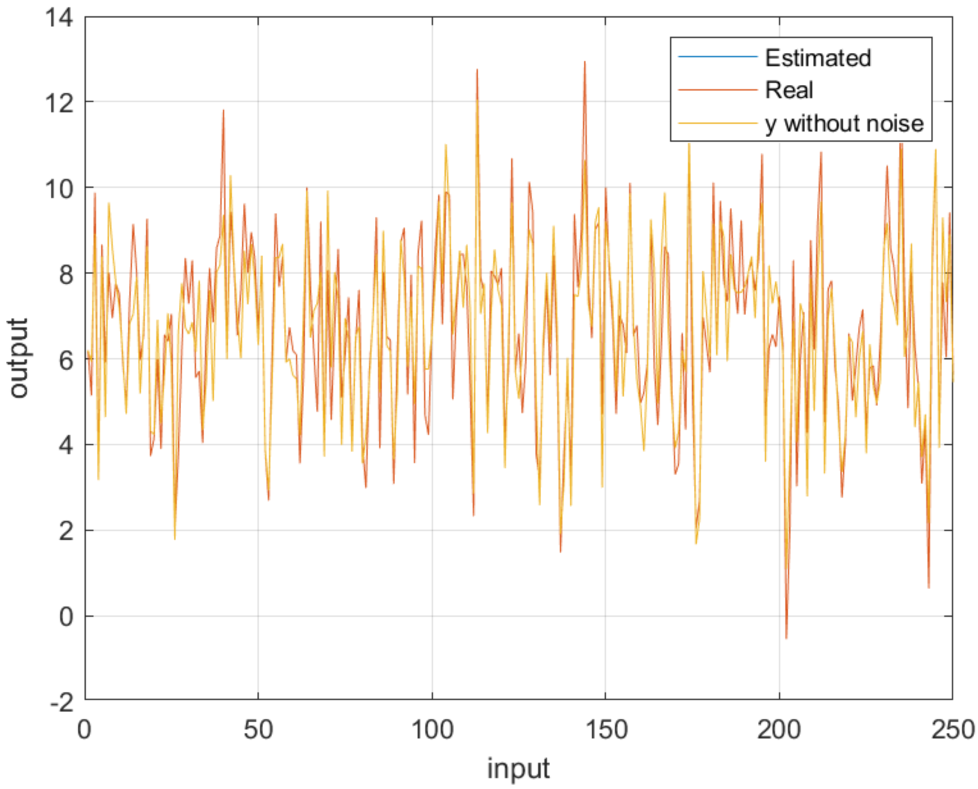
Estimated
Real
y without noise
output
input
14
12
10
8
6
4
2
0
0
50
100
150
200
250

برای نویز زیاد






estimated
real
teta hat1
14
13.8
13.6
13.4
13.2
13
12.8
12.6
12.4
12.2
12
0
50
100
estimated
real
teta hat2
2.00000000000025
2.0000000000002
2.00000000000015
2.0000000000001
2.00000000000005
2
1.99999999999995
1.9999999999999
1.99999999999985
1.9999999999998
1.99999999999975
0
50
100


Estimated
Real
y without noise
output
input
14
12
10
8
6
4
2
0
-2
0
50
100
150
200
250