۱۳۰۷

دانشگاه صنعتی خواجه نصیرالدین طوسی

دانشکده‌ی مهندسی هوافضا

گزارش پروژه‌ی پایان‌ترم درس شناسایی سیستم و تخمین پارامترهای پروازی

عنوان:

شناسایی سیستم دینامیکی فضاپیما و کدنویسی مطالب تدریس شده در طول ترم

نگارش:

مهسا آزادمنش

استاد درس:

جناب آقای دکتر خوشنود

فروردین ۱۴۰۰

بسم الله الرحمن الرحیم

چکیده

در این پروژه ابتدا مدل دینامیکی فضاپیما در نزدیکی سیارک اروس ۴۳۳ با نظر گرفتن گرانش سیارک و نیروهای اغتشاشی تعریف می‌شود. هدف، شناسایی سیستم تعریف‌شده و تخمینی از پارامترها است. این امر مستلزم اعمال ورودی پایا به سیستم است. ابتدا با شبیه‌سازی در نرم‌افزار متلب ورودی و خروجی مطلوب به دست می‌آید. سپس با استفاده از روش تخمین مبتنی بر خطای پیش‌بینی با ضریب فراموشی متغیر شناسایی سیستم انجام می‌شود. روش مد نظر یک شناسایی جعبه سفید است و در آن تک‌تک پارامترها مجاسبه می‌شوند. در ادامه با در نظر گرفتن سیستم به صورت جعبه سیاه، یک‌بار دیگر کار شناسایی سیتم با شبکه عصبی مصنوعی انجام می‌گردد. از آن‌جایی که نیاز است تمام مطالب تدریس شده به نوعی در پروژه اعمال شود یک سیستم جرم و فنر نیز در بخش دوم پروژه برای اعمال مطالبی مانند تبدیل z و غیره در نظر گرفته شده است که کار نهایی با توجه به جزوه‌ی درسی تکمیل باشد.

واژه‌های کلیدی: شناسایی سیستم، اروس ۴۳۳، گرانش سیارک، اغتشاش، تخمین مبتنی بر خطای پیش‌بینی با ضریب فراموشی متغیر، تخمین پارامتر، شبکه عصبی، تبدیل z

فهرست مطالب

فصل اول: مدل دینامیکی فضاپیما

تعریف مسئله

هدف این پژوهش شـــناســـایی مدل دینامیکی یک فضـــاپیما حول ســـیارک اروس ۴۳۳ اســـت. در این پروژه مدل دینامیکی فضـاپیما در نزدیکی سـیارک، با نظر گرفتن گرانش سـیارک و نیروهای اغتشـاشـی تعریف می‌شـود. هدف، شـناسـایی سـیسـتم تعریف‌شده و تخمینی از پارامترها است. این امر مستلزم اعمال ورودی پایا به سیستم است. ابتدا با شبیه‌سازی در نرم‌افزار متلب ورودی و خروجی مطلوب به دســت می‌آید. در ادامه با اســتفاده از روش تخمین مبتنی بر خطای پیش‌بینی با ضـریب فراموشـی متغیر شناسایی سیستم انجام می‌شود. در این فصل به توضیح دینامیک سیستم و روند شناسایی سیستم پرداخته شده است.

مدل دینامیکی

مدل دینامیکی سیستم طبق رابطه‌ی زیر به دست می آید.

$$\ddot{x} = ۲\omega\dot{z} + \frac{F_x}{m_c} + U_x + \delta_x$$

$$\ddot{y} = -\omega^۲ y + \frac{F_y}{m_c} + U_y + \delta_y$$

$$\ddot{z} = -۲\omega\dot{x} + ۳\omega^۲ z + \frac{F_z}{m_c} + U_z + \delta_z$$

که x, y, z موقعیت فضـاپیما نسـبت به دسـتگاه مختصـات در مرکز جرم سـیارک و ω بیانگر سـرعت زاویه ای سـیارک اســت. برای ســادگی حرکت ســیارک در مدار دایروی فرض شــده اســت. F_x، F_y و F_z نیز بیانگر نیروی های کنترلی، m_c جرم فضاپیما، U_x، U_y و U_z پتانسیل گرانشی سیارک و δ_x، δ_y و δ_z نیز نیروهای اغتشاشی هستند.

تابع پتانسیل مطابق با رابطه زیر به دست می آید.

$$U(\boldsymbol{R}) = \frac{GM_B}{R} \sum_{l=0}^{\infty} \sum_{m=0}^{l} \left(\frac{R_B}{R}\right)^l \overline{P}_{l,m}[\sin(\varphi)] \left(\overline{C}_{l,m} \cos(m\lambda) + \overline{S}_{l,m} \sin(m\lambda)\right)$$

که در آن R فاصله از فضاپیما تا مرکز جرم سیارک، φ زاویه عرضی، λ زاویه طولی، M_B جرم سیارک، R_B شعاع مرجع سیارک، $\bar{P}_{l,m}$ ضرایب لژاندر ، $\bar{S}_{l,m}$ و $\bar{C}_{l,m}$ ضرایب استوکس هستند.

طراحی کنترل‌کننده

با استفاده از کنترل مود لغزشی که با توجه به ماهیت غیرخطی سیستم در مقابل اغتشاشات و سایر نیروهای خارجی مقاوم است خروجی مطلوب حاصل می‌شود. طراحی کنترلر با تعریف سطح لغزش

$$s = \dot{e} + \lambda_a e$$

که در آن e خطا و λ_a ضــریب وزن دهی اســت و با اســتفاده از رابطه $\dot{s} = ۰$ و تعیین ورودی کنترلی u_{eq} انجام می‌شود. با تعریف $u_r = -k * sign(s)$ ورودی کنترلی به صورت زیر تعریف می‌شود.

$$u = u_{eq} + u_r$$

در کدهای موجود در پیوست ۲ معادلات فوق بیان شده‌اند.

شناسایی سیستم بر اساس خطای پیش‌بین

ابتدا سیستم در نرم‌افزار متلب شبیه‌سازی می شود که به دلیل ناپایدار بودن سیستم با طراحی کنترل کننده مود لغزشی ورودی کنترلی به دست می آید. در ادامه با داشتن ورودی و خروجی سیستم، مدل دینامیکی شناسایی می‌شود.
یکی از روش های شناسایی تخمین پارامترهای سیستم بر اساس خطای پیش‌بینی است. در این روش مدل سیستم به صورت پارامتری خطی تبدیل می‌شود که در مورد سیستم های خطی کاربرد دارد. سیستم فوق به دلیل ترم‌های پتانسیل گرانشی دارای ماهیت غیر خطی است. اما می توان آن را به صورت رگرسیون خطی نوشت.

هدف از شناسایی به کارگیری مدلی برای شناسایی تابع پتانسیل و اغتشاشات و تخمین پارامترهای m_c و ω است.
مدل سیستم به صورت پارامتری خطی مطابق رابطه‌ی زیر بیان می‌شود.

$$y = W\varphi$$

که y بیانگر خروجی سیستم، W ماتریس رگرسیو و φ بردار پارامترهای سیستم است. بنابراین سیستم ما به صورت زیر تبدیل می شود.

$$\hat{y} = W\hat{\varphi}$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} F_x & \widehat{U}_x & ۲\dot{z} & ۰ \\ F_y & \widehat{U}_y & ۰ & -y \\ F_z & \widehat{U}_z & -۲\dot{x} & ۳z \end{bmatrix} \begin{bmatrix} \frac{۱}{\mathrm{m_c}} \\ ۱ \\ \omega \\ \omega^۲ \end{bmatrix}$$

که $\widehat{U}_x$، $\widehat{U}_y$ و $\widehat{U}_z$ تخمین تابع پتانسیل و اغتشاشات است که به صورت سیستم مرتبه دو مدل می‌شود.

$$y - \hat{y} = e = [\ddot{x}\ \ddot{y}\ \ddot{z}]^T - W\hat{\varphi}$$

که e برابر با خطا است. و بردار پارامترها طبق رابطه زیر تخمین زده می شود.

$$\dot{\hat{\varphi}} = -P(t)W^Te$$

که P ماتریس بهره تخمین‌گر است و از رابطه زیر به دست می آید.

$$\dot{P} = \lambda(t)P - PW^TWP$$

که $\lambda(t)$ ضریب فراموشی است و برای بهبود ردیابی پارامترها استفاده می‌شود. و از رابطه زیر حاصل می‌شود.

$$\lambda(t) = \lambda_. * (۱ - \frac{\|P\|}{k_.})$$

که $k_.$ بهره‌ای است که بر سرعت همگرایی پارامترها تاثیر دارد و $\lambda_.$ ضریب فراموشی ثابت است.

فصل دوم: شبیه‌سازی و نتایج روش خطای پیش‌بین

شبیه‌سازی

مقادیر پارامترهای سیستم به صورت زیر تعریف می‌شود.

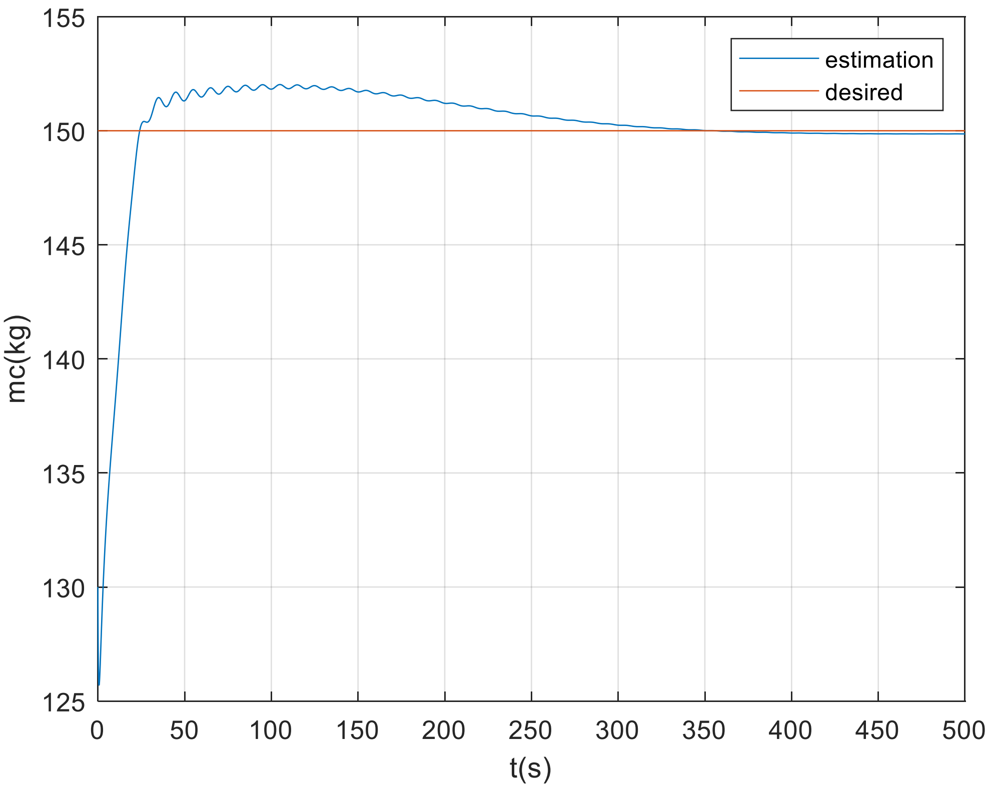
جرم فضاپیما	m_c	۱۵۰ (kg)
سرعت زاویه ای سیارک	$\omega = \sqrt{\frac{\mu}{r_a^{۳}}}$	۱.۱۲e-۷(rad/s)
فاصله سیارک از خورشید	r_a	۲۱۶۶۹۶۳۴۲(km)
جرم سیارک	m_a	۶.۶۸۷e۱۵(km)
شعاع مرجع سیارک	R_B	۱۶(km)

شرایط اولیه نیز به صورت $R_. = [-۵۰ \cdot -۱۵](km)$ و $\dot{R}_. = [۰\ ۴۰\ ۰](\frac{m}{s})$ در نظر گرفته می‌شود و فرض می‌شود اغتشاش $\delta = ۱e - ۵ * \sin(۰.۲\pi t)$ به سیستم وارد می‌شود.

نتایج شبیه‌سازی

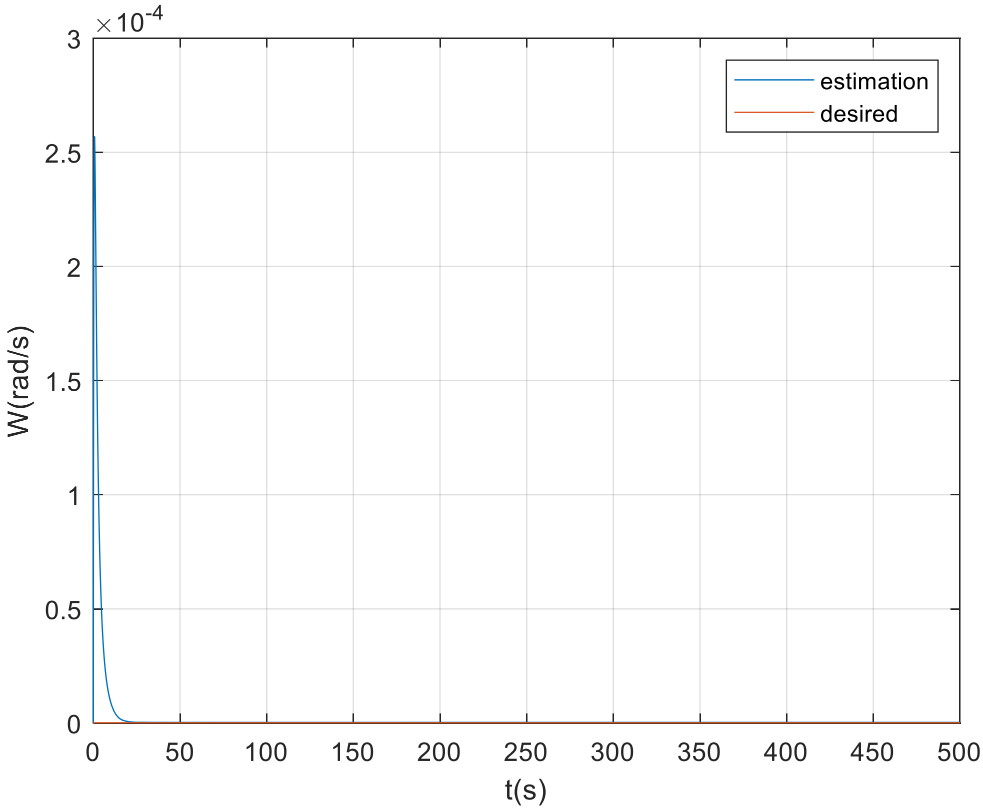
نتایج حاصل از شبیه‌سازی برای تخمین پارامترها و شناسایی سیستم در شکل‌های زیر آورده‌شده است.

مطابق با شکل (۱) پارامتر جرم به مقدار مطلوب ۱۵۰ کیلوگرم همگرا می‌شود.



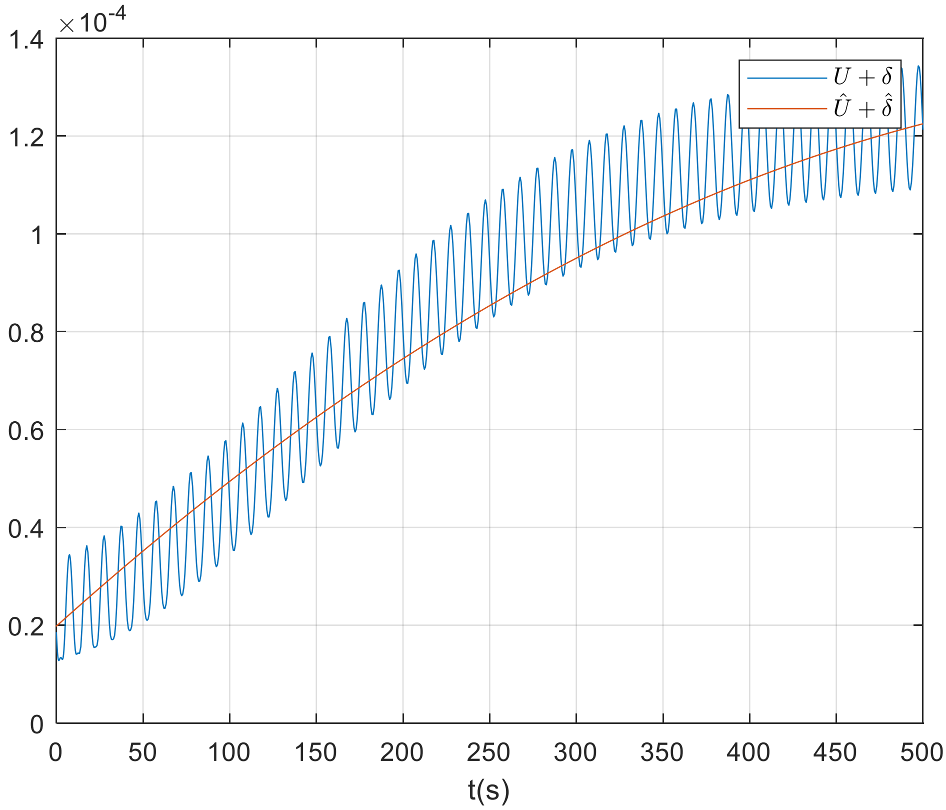
شکل ۱: پارامتر جرم

مطابق با شکل (۲) نیز پارامتر سرعت زاویه‌ای سیارک به مقدار مطلوب همگرا می‌شود.



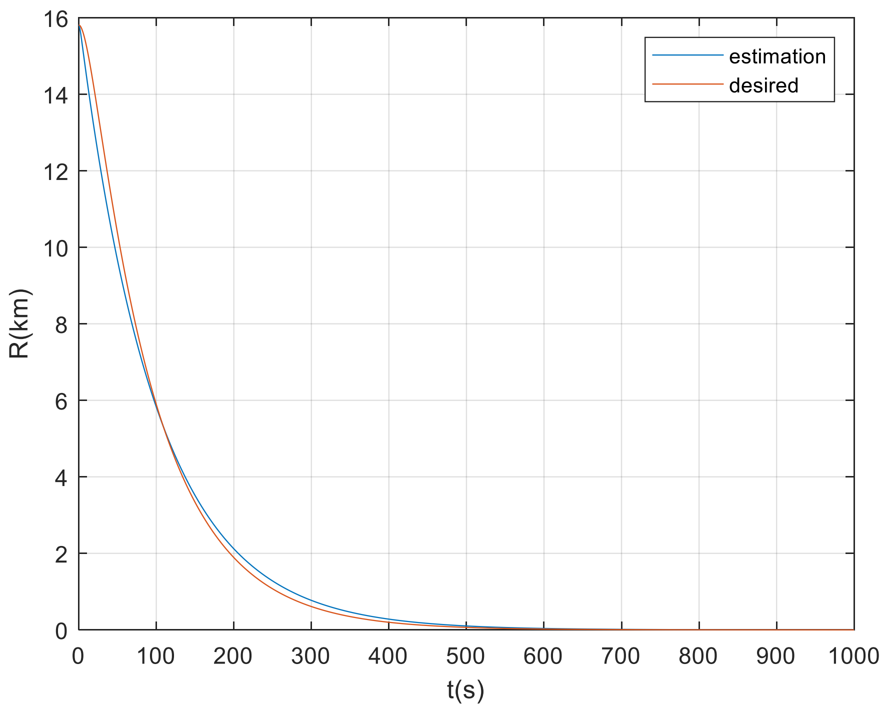
شکل ۲: پارامتر سرعت زاویه‌ای

در شکل ۳ تابع پتانسیل گرانشی و اغشاشات وارد به سیستم با یک سیستم مرتبه دو مدل شده‌است. نوسانات مشاهده شده در شکل ۳ ناشی از اعمال اغتشاش سینوسی به سیستم است.

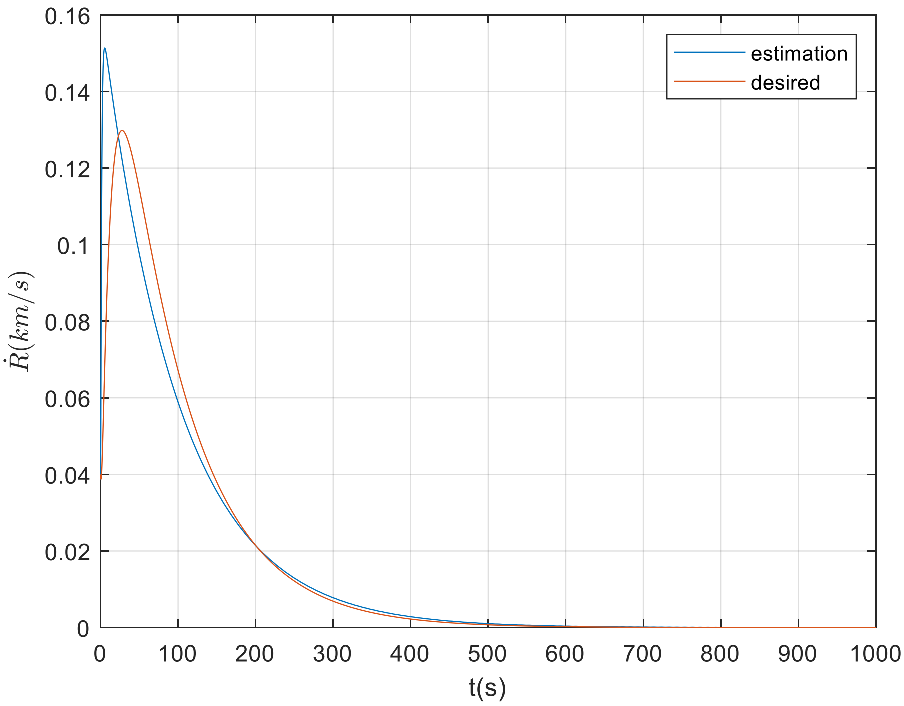


شکل ۳: برازش تابع پتانسیل گرانشی و اغتشاشات با سیستم مرتبه دو

در شکل‌های ۴ و ۵ فاصله و سرعت نسبی بین فضاپیما و سیارک حاصل از سیستم شناسایی شده به مقدار مطلوب حاصل از سیستم با کنترل‌کننده می‌رسد.



شکل ۴: فاصله نسبی بین فضاپیما و سیارک



شکل ۵: سرعت نسبی بین فضاپیما و سیارک

نتیجه گیری

در این پروژه مدل دینامیکی یک فضاپیما حول سیارک Eros ۴۳۳ با فرض حضور اغتشاشات شناسایی شده است. پارامترهای جرم فضاپیما و سرعت زاویه‌ای سیارک حول مدارش به دور خورشید تخمین زده می‌شود. و تابع پتانسیل مربوط به سیارک نیز با یک سیستم مرتبه دو تخمین زده می‌شود. نتایج همگرایی پارامترها و پاسخ سیستم نشان از شناسایی مطلوب سیستم دارد.

فصل سوم: تکرار روش با شبکه عصبی چند لایه

شناسایی با شبکه عصبی

در این قسمت قصد داریم سیستم فضاپیما حول سیارک را با استفاده از روش هوشمند شبکه عصبی مصنوعی و با فرض جعبه سیاه بودن سیستم تخمین بزنیم. مدل دینامیکی سیستم به صورت زیر است:

$$\ddot{x} = ۲\omega\dot{z} + \frac{F_x}{m_c} + U_x + \delta_x$$

$$\ddot{y} = -\omega^۲ y + \frac{F_y}{m_c} + U_y + \delta_y$$

$$\ddot{z} = -۲\omega\dot{x} + ۳\omega^۲ z + \frac{F_z}{m_c} + U_z + \delta_z$$

متغیرهای حالت را به صورت زیر تعریف می‌کنیم:

$$\begin{cases} x_۱ = x \\ x_۲ = \dot{x} \\ x_۳ = y \\ x_۴ = \dot{y} \\ x_۵ = z \\ x_۶ = \dot{z} \end{cases}$$

مدل فضای حالت به فرم زیر تبدیل می شود:

$$\dot{x}_۱ = x_۲$$

$$\dot{x}_۲ = ۲\omega x_۶ + \frac{F_x}{m_c} + U_x + \delta_x$$

$$\dot{x}_۳ = x_۴$$

$$\dot{x}_۴ = -\omega^۲ x_۳ + \frac{F_y}{m_c} + U_y + \delta_y$$

$$\dot{x}_۵ = x_۶$$

$$\dot{x}_۶ = -۲\omega x_۲ + ۳\omega^۲ x_۵ + \frac{F_z}{m_c} + U_z + \delta_z$$

خروجی:

$$\begin{bmatrix} y_۱ \\ y_۲ \\ y_۳ \end{bmatrix} = \begin{bmatrix} ۱ & ۰ & ۰ & ۰ & ۰ & ۰ \\ ۰ & ۰ & ۱ & ۰ & ۰ & ۰ \\ ۰ & ۰ & ۰ & ۰ & ۱ & ۰ \end{bmatrix} \begin{bmatrix} y_۱ \\ y_۲ \\ y_۳ \end{bmatrix}$$

نیروهای پتانسیل را به صورت زیر در نظر می‌گیریم:

$$U = G\frac{m_c m_a}{(r_a+\alpha)^۲}$$

$$G = ۶.۶۷ \times ۱۰^{-۱۱}$$

$$m_c = ۱۵۰\ Kg$$

$$m_a = ۶.۶۸۷ \times ۱۰^{۱۵}\ Kg$$

$$r_a = ۱۶\ Km = ۱۶۰۰۰\ m$$

بنابراین:

$$U = G\frac{m_c m_a}{(r_a+\alpha)^۲}$$

$$U_x = (۶.۶۷ \times ۱۰^{-۱۱})\frac{۱۵۰(۶.۶۸۷ \times ۱۰^{۱۵})}{(۱۶۰۰۰+x)^۲} = \frac{۶.۶۹ \times ۱۰^{۷}}{(۱۶۰۰۰+x)^۲}$$

$$U_y = \frac{۶.۶۹ \times ۱۰^{۷}}{(۱۶۰۰۰+y)^۲}$$

$$U_z = \frac{۶.۶۹ \times ۱۰^{۷}}{(۱۶۰۰۰+z)^۲}$$

گسسته سازی سیستم با زمان نمونه برداری T:

$$\frac{x(t+T)+x(t)}{T} = \dot{x}_t$$

$$\frac{x(K+۱)+x(K)}{T} = \dot{x}_t$$

حال می‌توان شکل گسسته سازی شده معادلات حالت را نوشت:

$$x_۱(K+۱) = x_۱(K)$$

$$x_۲(K+۱) = x_۲(K) + ۲Twx_۶(K) + \frac{T\,F_x}{m_c} + TU_x + T\delta_x$$

$$x_۳(K+۱) = x_۳(K) + Tx_۴(K+۱)$$

$$x_۴(K+۱) = x_۴(K) - ۲Tw^۲x_۲(K) + \frac{T\,F_y}{m_c} + TU_y + T\delta_y$$

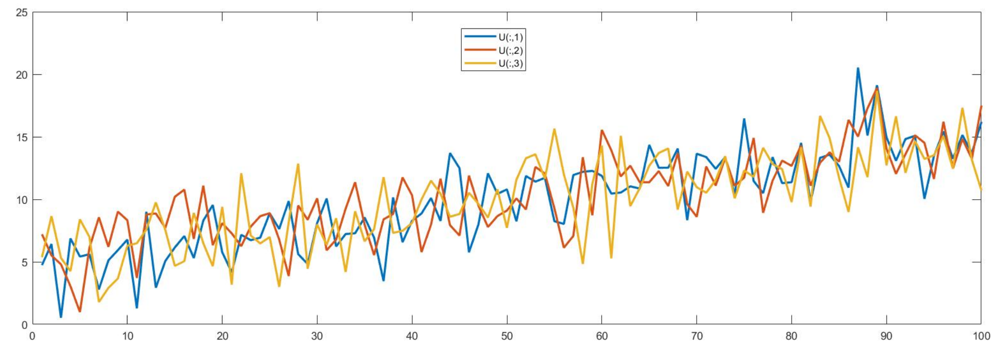
$$x_۵(K+۱) = x_۵(K) + Tx_۶(K)$$

$$x_۶(K+۱) = x_۶(K) - ۲Twx_۲(K) + ۳Tw^۲x_۵(K) + \frac{T\,F_z}{m_c} + TU_z + T\delta_z$$

با در نظر گرفتن پارامترها به صورت جدول قسمت قبل و همچنین در نظر گرفتن نیروهای مختلف در حد چند نیوتن و البته اغتشـــاش های جمع شـــونده با دامنه بســـیار کوچک، داده های مختلف ورودی و خروجی را به دست می‌آوریم سیستم دارای سه ورودی و سه خروجی است. سپس با آموزش یک شبکه عصـبی MLP یک مدل برای سـیسـتم با اسـتفاده از داده‌های ورودی و خروجی به دسـت آمده تخمین می‌زنیم.

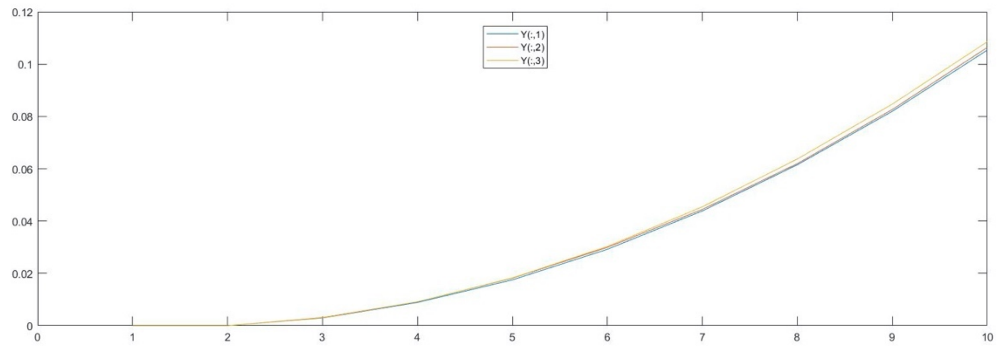
نتایج شبکه عصبی

در قسمت اول ورودی‌های تصادفی به اضافه ورودی‌های شیب را اعمال کردیم. شکل زیر سه ورودی اعمال شده این سیستم را در نمونه‌های مختلف نشان می‌دهد:



شکل ۶: سه ورودی اعمال شده در نمونه‌های مختلف

شکل زیر نیز خروجی‌های سیستم را در نمونه‌های مختلف نشان می‌دهد.



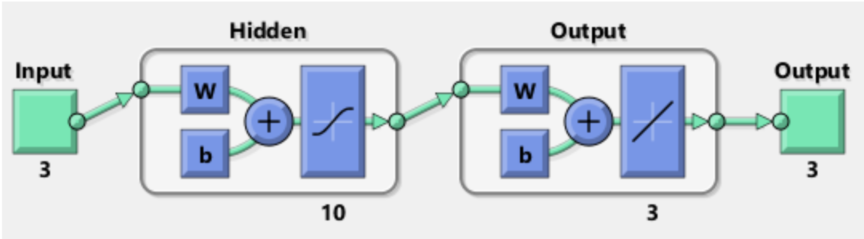
شکل ۷: خروجی‌های سیستم در نمونه‌های مختلف

البته لازم به ذکر است که در هزار نمونه مختلف داده‌های ورودی و خروجی جمع آوری شده‌اند که در دو شکل بالا فقط نمونه‌های اولیه ترسیم شده‌اند.

حال با استفاده از داده های به دست آمده در قسمت قبل قصد داریم یک شبکه عصبی MLP را آموزش دهیم.

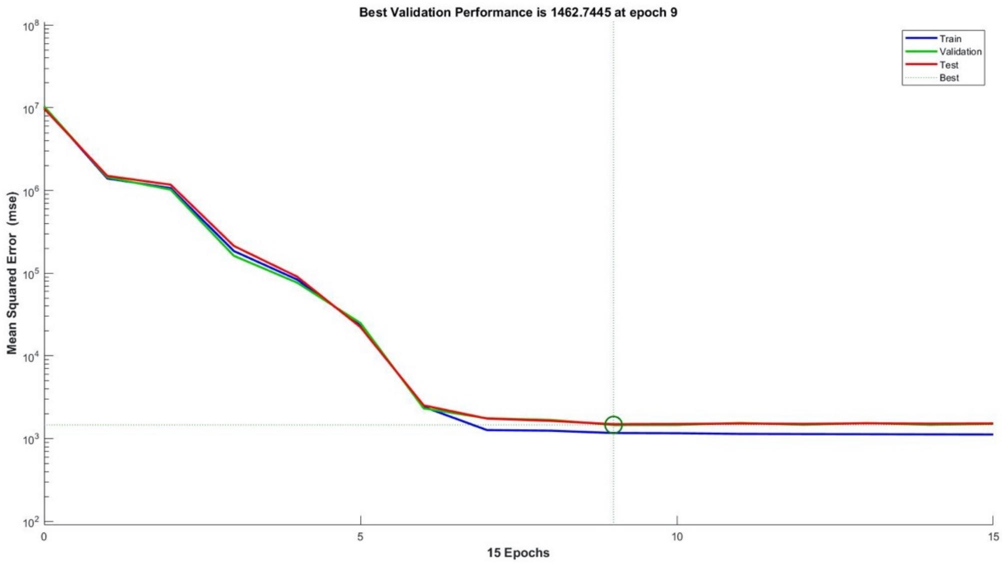
برای آموزش شبکه از ۷۰ درصد داده‌ها برای آموزش از ۱۵ درصد داده‌ها برای اعتبار سنجی و از ۱۵ درصد داده‌ها نیز برای تست شبکه پس از آموزش استفاده کرده‌ایم.

شبکه دارای دو لایه است. یک لایه ورودی که شامل ۱۰ نورون است و یک لایه خروجی که شامل سه نورون است که ۳ خروجی سیستم را نگاشت می‌کند. شکل کلی شبکه به صورت زیر است:



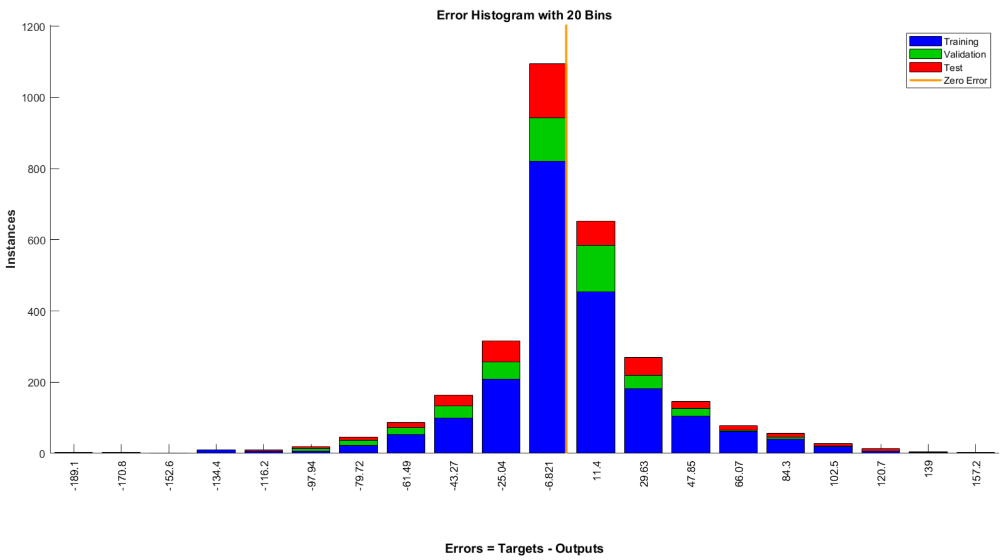
شکل ۸: شکل کلی شبکه

عملکرد شبکه و میزان خطا به صورت زیر به دست می‌آید:



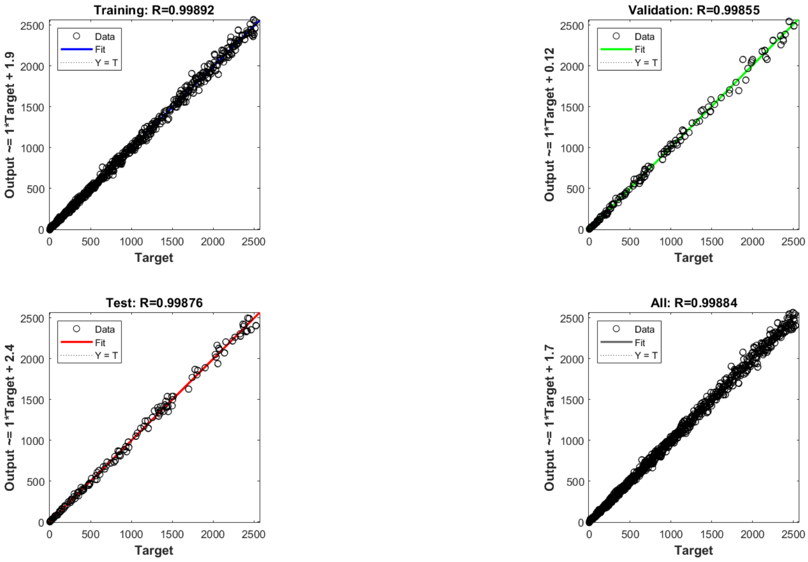
شکل ۹: عملکرد شبکه و میزان خطا

نمودار هیستوگرام خطا به صورت زیر است:



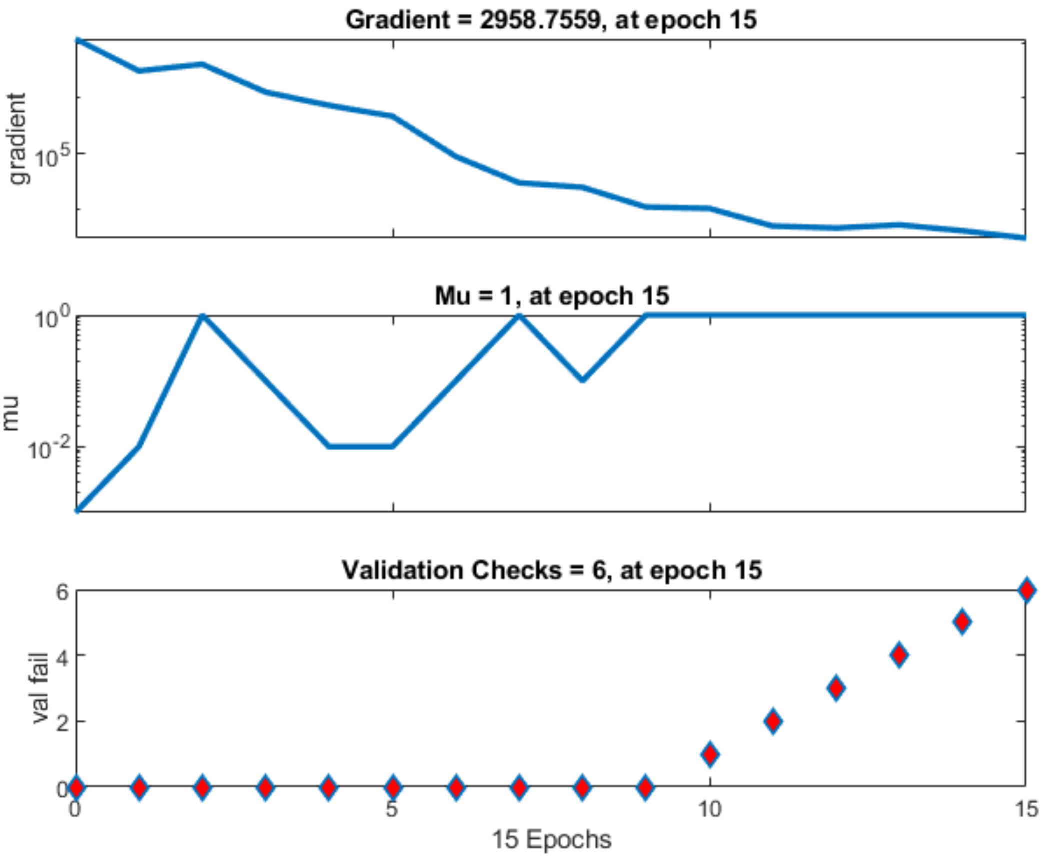
شکل ۱۰: نمودار هیستوگرام خطا

رگرسیون روی سه دسته داده آموزش، اعتبارسنجی و تست نیز به صورت زیر به دست می‌آید و مشاهده می‌شود که بسیار به عدد یک نزدیک است:



شکل۱۱: رگرسیون روی سه دسته داده آموزش، اعتبارسنجی و تست

سایز پارامترهای شبکه نیز به صورت زیر است:



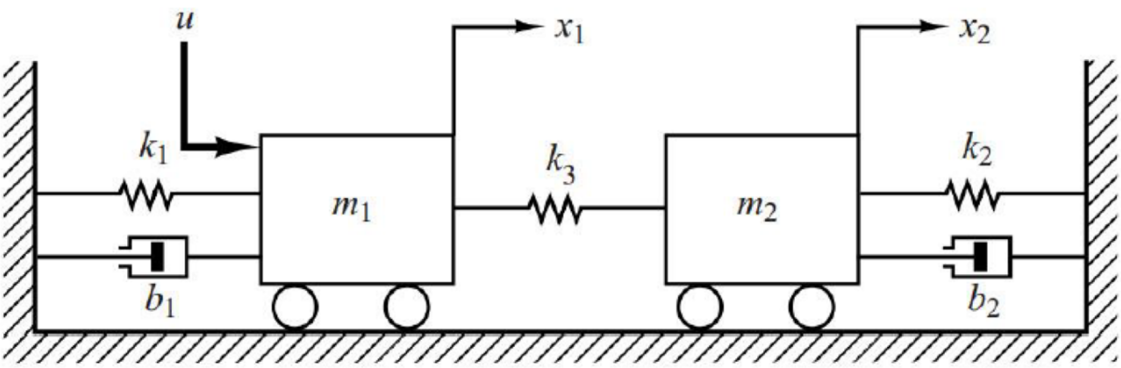
شکل۱۲: سایز پارامترهای شبکه

فصل چهارم: اعمال دو روش LS و RLS

در این قسمت از پروژه می‌خواهیم یک سیستم جرم و فنر و دمپر را با استفاده از روش‌های حداقل مربعات و حداقل مربعات بازگشتی شناسایی کرده و پارامترهای آن را تخمین بزنیم. **و هر بخشی از درس که در تا اینجا در پروژه استفاده نشده است را روی این دینامیک پیاده کنیم.**

شناسایی سیستم با روش LS

سیستم زیر را در نظر می‌گیریم.



شکل۱۳: سیستم مدنظر برای بخش دوم پروژه

معادلات حاکم بر سیستم و همچنین مقادیر پارامترها به صورت زیر اند.

$$(m_1S^2 + b_1S + k_1 + k_3)X_1(S) - k_3X_2(S) = U(S)$$

$$(m_2S^2 + b_2S + k_2 + k_3)X_2(S) - k_3X_1(S) = 0$$

$$m_1 = m_2 = 1$$

$$k_1 = b_2 = 0.8$$

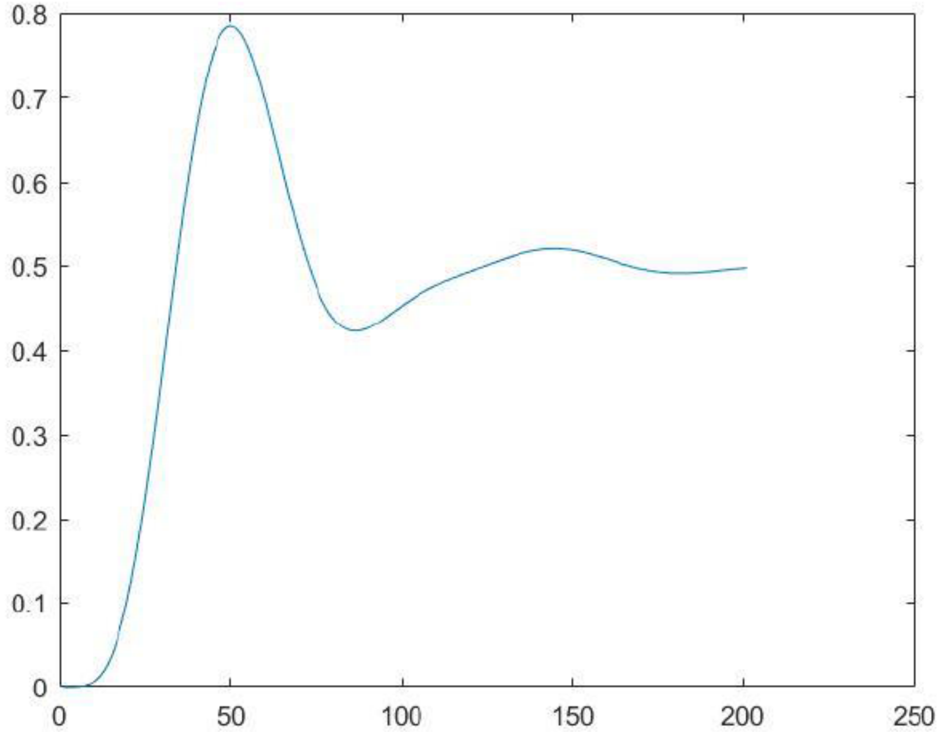
$$k_2 = b_1 = 0.4$$

$$k_3 = 0.4$$

تابع تبدیل سیستم به صورت زیر به دست می‌آید:

$$\frac{X_2(S)}{U(S)}=\frac{0.4}{S^4+1.2S^3+2.32S^2+1.12S+0.8}$$

پاسخ پله زمان پیوسته این سیستم به صورت شکل زیر است:



شکل۱۴: پاسخ پله زمان پیوسته

با توجه به زمان صـعود که حدود ۲۰ ثانیه اسـت ۱۰ نمونه از این زمان را بر می داریم بنابراین زمان نمونه برداری را برابر با ۲ ثانیه در نظر میگیریم. با گســـســـته کردن تابع تبدیل با این زمان نمونه برداری تابع تبدیل گسسته به صورت زیر به دست می‌آید.

```
  0.1317 z^3 + 0.6007 z^2 + 0.3582 z + 0.03051
  ----------------------------------------------
  z^4 + 0.4749 z^3 + 0.496 z^2 + 0.1807 z + 0.09072

Sample time: 2 seconds
Discrete-time transfer function.
```

$$\frac{Y(Z)}{U(Z)} = \frac{۰.۱۳۱۷Z^{۳} + ۰.۶۰۰۷Z^{۲} + ۰.۳۵۸۲Z + ۰.۰۳۰۵۱}{Z^{۴} + ۰.۴۷۴۹Z^{۳} + ۰.۴۹۶Z^{۲} + ۰.۱۸۰۷Z + ۰.۰۹۰۷۲}$$

$$\begin{aligned} Y(Z) + ۰.۴۷۴۹Z^{-۱}Y(Z) + ۰.۴۹۶Z^{-۲}Y(Z) + ۰.۱۸۰۷Z^{-۳}Y(Z) + ۰.۰۹۰۷۲Z^{-۴}Y(Z) \\ = ۰.۱۳۱۷Z^{-۱}U(Z) + ۰.۶۰۰۷Z^{-۲}U(Z) + ۰.۳۵۸۲Z^{-۳}U(Z) \\ + ۰.۰۳۰۵۱Z^{-۴}U(Z) \end{aligned}$$

با گرفتن وارون تبدیل معادله بالا به صورت زیر به دست می‌آید.

$$\begin{aligned} y(n) + ۰.۴۷۴۹y(n-۱) + ۰.۴۹۶y(n-۲) + ۰.۱۸۰۷y(n-۳) + ۰.۰۹۰۷۲y(n-۴) \\ = ۰.۱۳۱۷u(n-۱) + ۰.۶۰۰۷u(n-۲) + ۰.۳۵۸۲u(n-۳) \\ + ۰.۰۳۰۵۱u(n-۴) \end{aligned}$$

فرم رگرسیون خطی به صورت زیر است:

$$y = \theta^{T}\varphi + n$$

θ پارامترها

φ رگرسورها

n نویز سفید گوسی

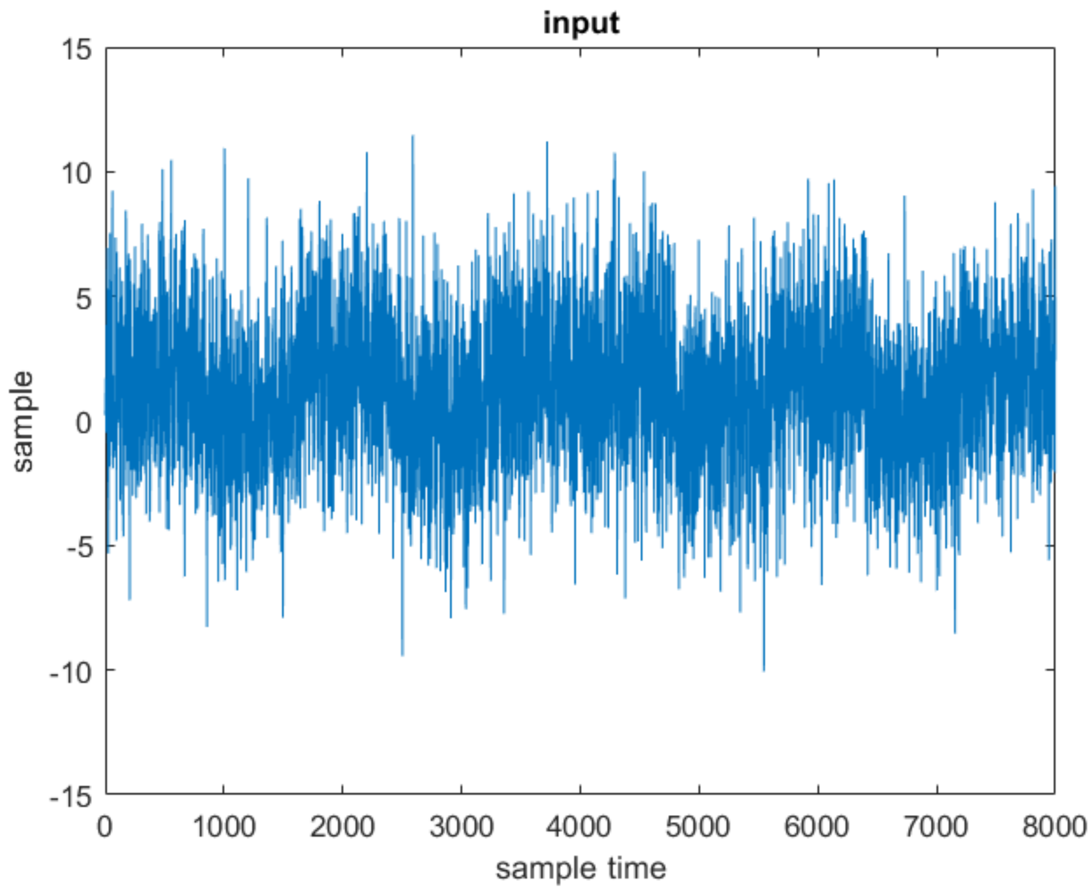
$$\theta^{T} = [۰.۴۷۴۹ \quad ۰.۴۹۶ \quad ۰.۱۸۰۷ \quad ۰.۰۹۰۷۲ \quad ۰.۱۳۱۷ \quad ۰.۶۰۰۷ \quad ۰.۳۵۸۲ \quad ۰.۰۳۰۵۱\,]$$

$$\begin{aligned} \varphi^{T} = [-y(n-۱) \quad -y(n-۲) \quad -y(n-۳) \\ -y(n-۴) \quad u(n-۱) \quad u(n-۲) \quad u(n-۳) \quad (n-۴)] \end{aligned}$$

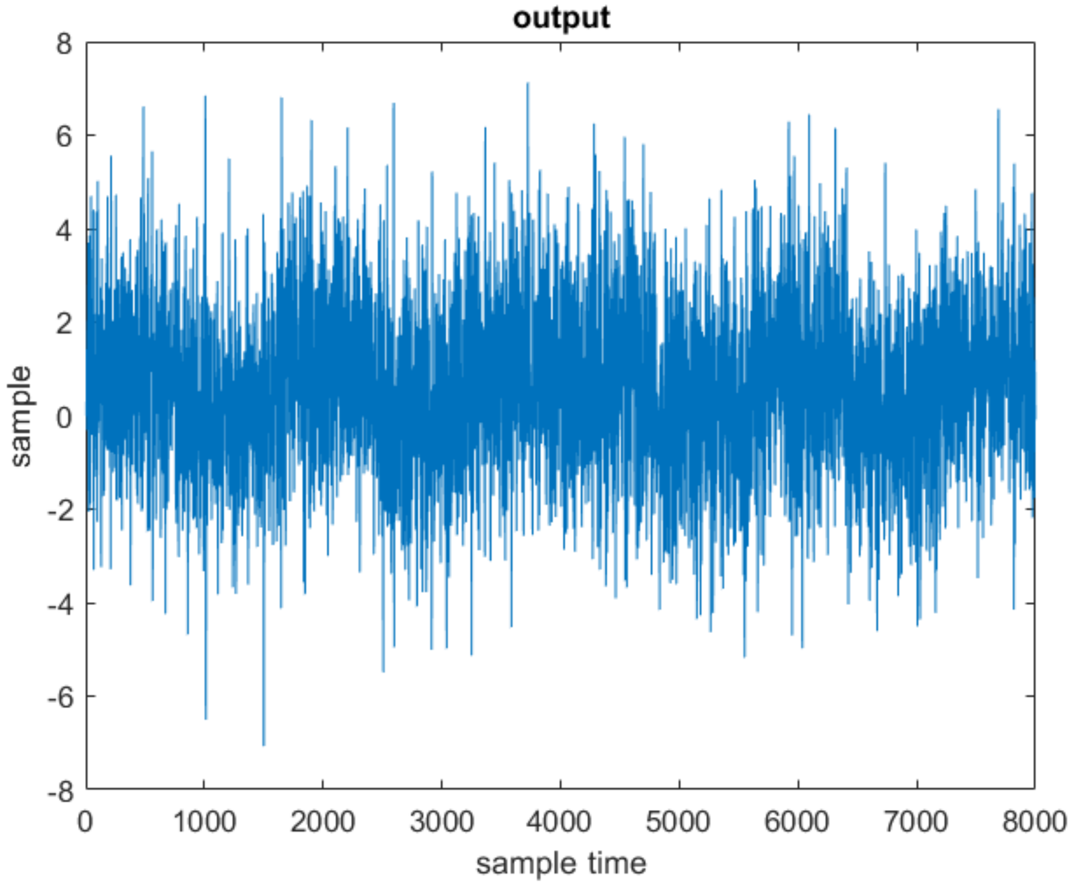
پارامتر های واقعی سیستم بردار θ اند.

ما ابتدا با توجه به پارامترهای واقعی سیستم و در حضور نویز سفید و با اعمال ورودی پالس خروجی را اندازه گیری کرده و داده های ورودی و خروجی را در اختیار داریم پس فرض می کنیم که سیستم به صورت یک جعبه خاکستری است یعنی اینکه پارامترها را نداریم ولی فرم کلی سیستم و رگرسورها و همچنین

داده‌های ورودی و خروجی را در اختیار داریم. پس با توجه به داده‌های ورودی و خروجی، پارامترهای سیستم را تخمین می‌زنیم. ورودی را به صورت جمعی از اعداد تصادفی و سیگنال پالس در نظر می‌گیریم. ورودی به صورت زیر است:



خروجی سیستم نیز به صورت زیر به دست می آید:



توسط تخمین LS یعنی با فرمول زیر پارامترها را محاسبه می‌کنیم:

$$\hat{\theta} = (\varphi^T\varphi)^{-۱}\varphi^T y$$

توسط این تخمین پارامتر ها به صورت زیر به دست می آیند.

```
theta_hat =

    0.4705
    0.4932
    0.1790
    0.0895
    0.1317
    0.6001
    0.3555
    0.0286
```

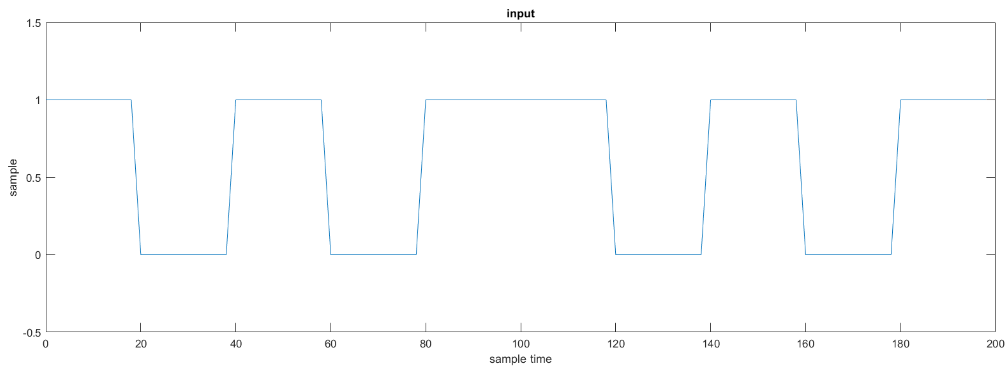
همانطور که مشاهده می شود پارامترهای تخمین زده به پارامترهای واقعی سیستم بسیار نزدیک کند پارامترهای واقعی و تخمین سیستم در جدول زیر آورده شده اند:

مقدار تخمینی	مقدار واقعی	نام پارامتر
۰.۴۷۰۵	۰.۴۷۴۹	$\theta_۱$
۰.۴۹۳۲	۰.۴۶۹	$\theta_۲$
۰.۱۷۹۰	۰.۱۸۰۷	$\theta_۳$
۰.۰۸۹۵	۰.۰۹۰۷۲	$\theta_۴$
۰.۱۳۱۷	۰.۱۳۱۷	$\theta_۵$
۰.۶۰۰۱	۰.۶۰۰۷	$\theta_۶$
۰.۳۵۵۵	۰.۳۵۸۲	$\theta_۷$
۰.۰۲۸۶	۰.۰۳۰۵۱	$\theta_۸$

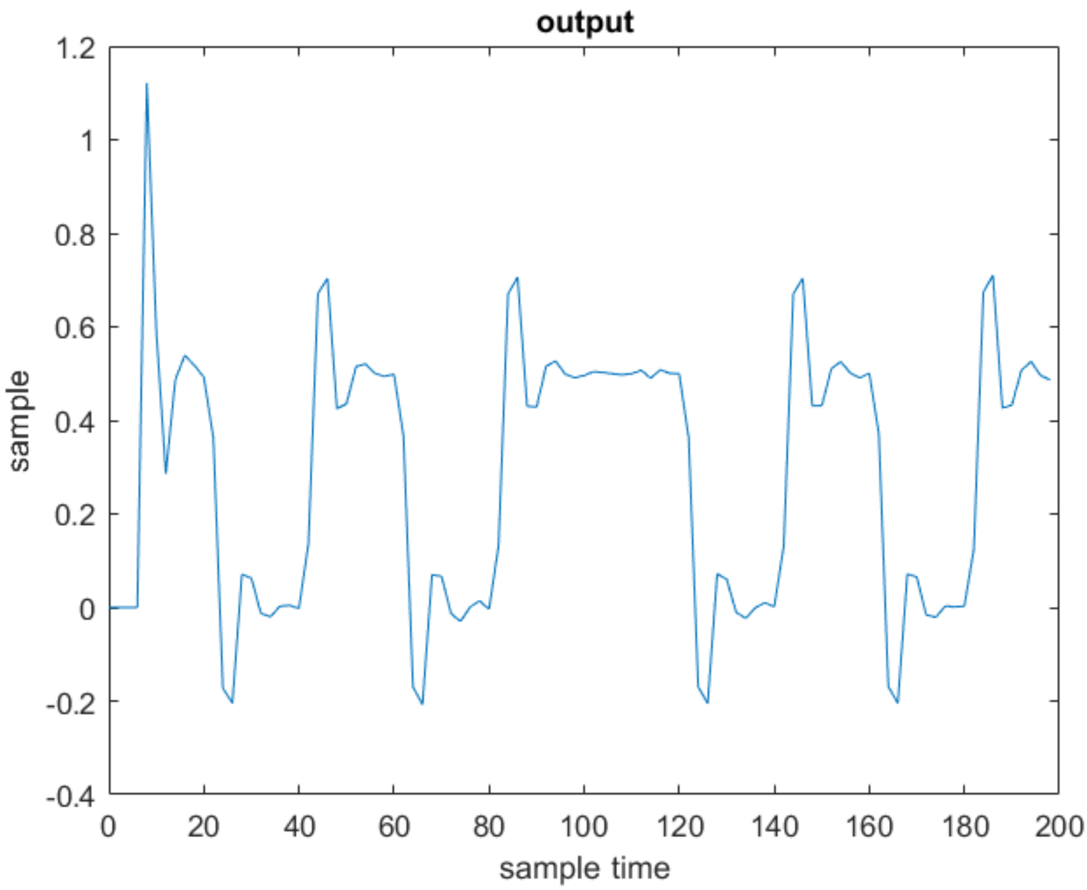
همانطور که مشاهده می‌شود مقدارهای تخمینی و مقدارهای واقعی بسیار نزدیک هستند.

شناسایی سیستم با روش RLS

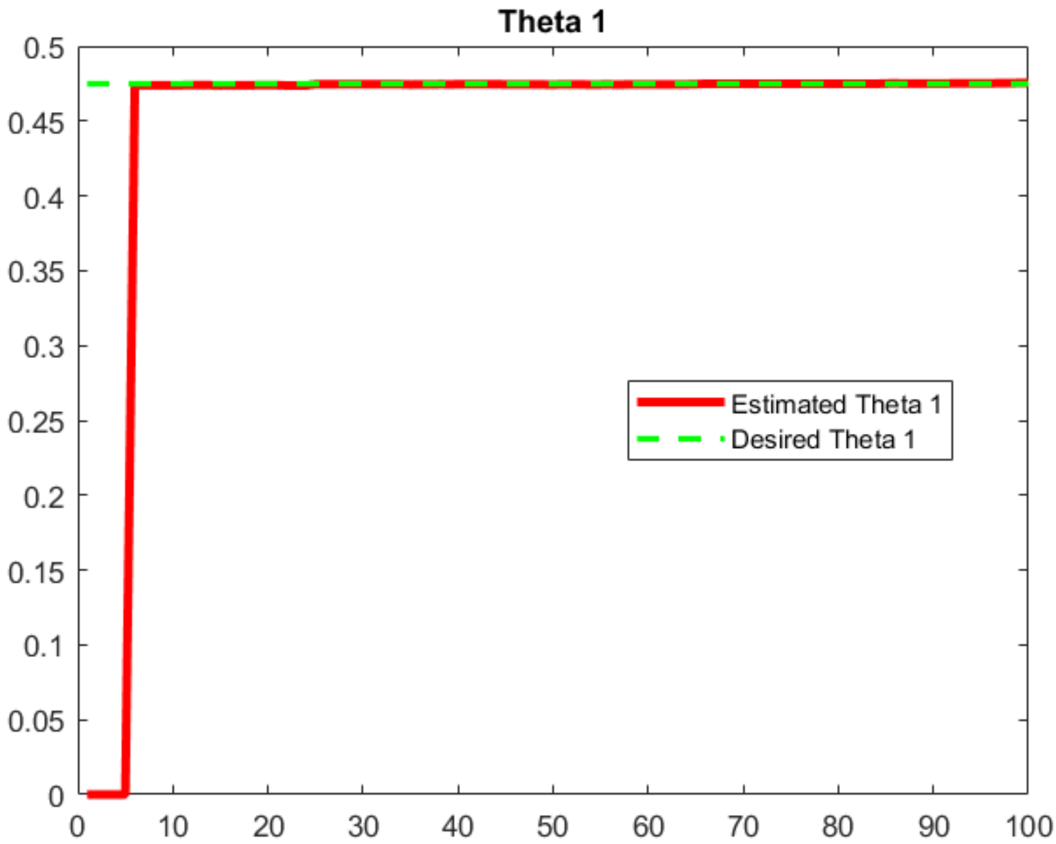
در این قسـمت با روش RLS پارامترها را تخمین می‌زنیم. شـکل زیر ورودی را نشـان می‌دهد. ورودی سـیگنال پالس می‌باشد. همچنین نویز جمع شونده گوسی نیز وجود دارد:

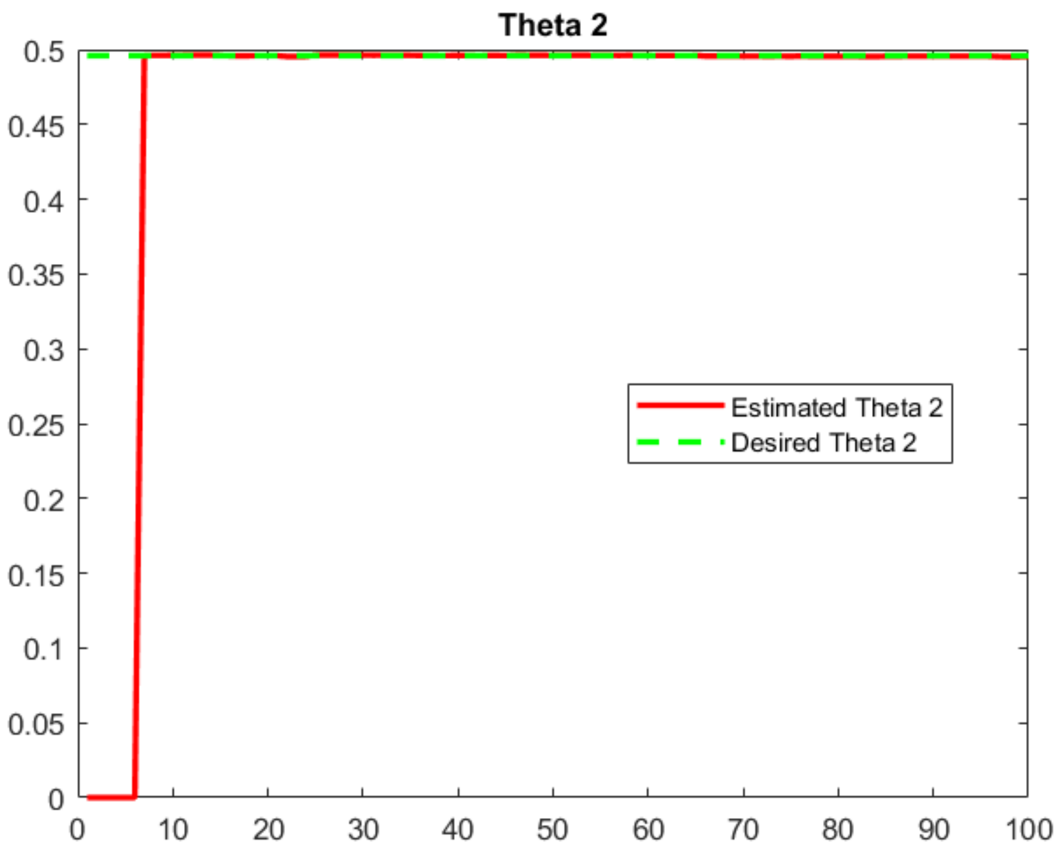


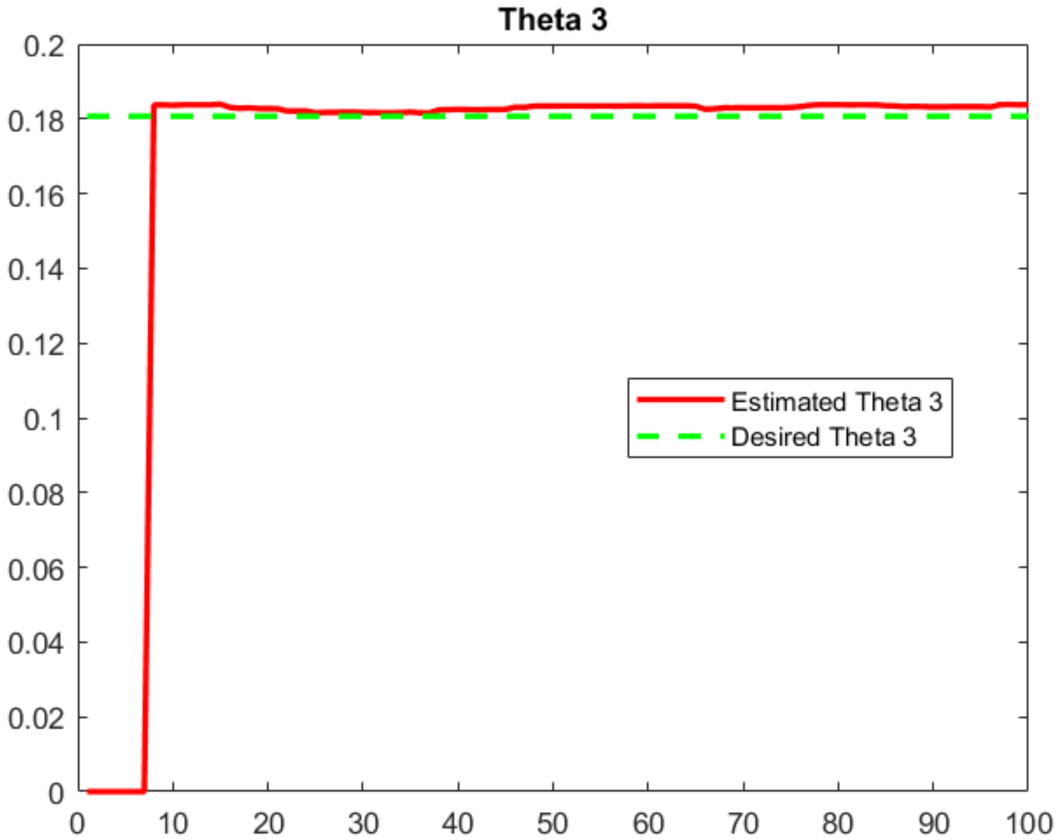
شکل زیر نیز خروجی سیستم را نشان می‌دهد:

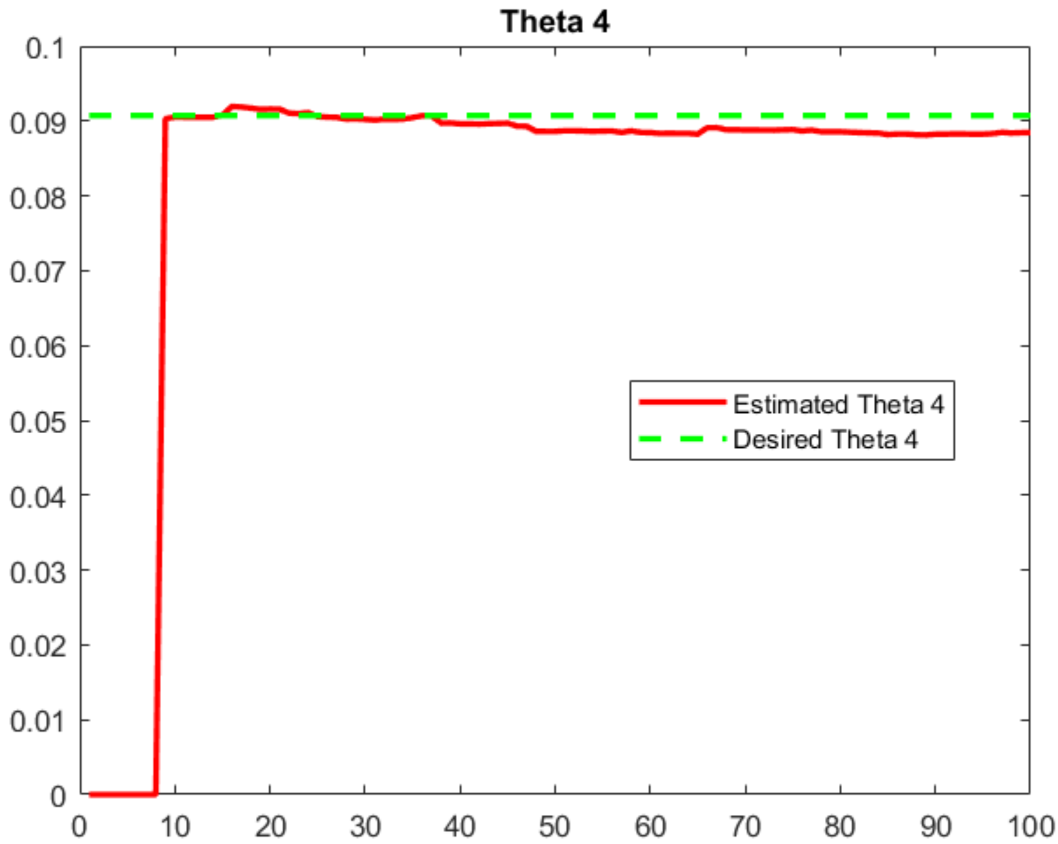


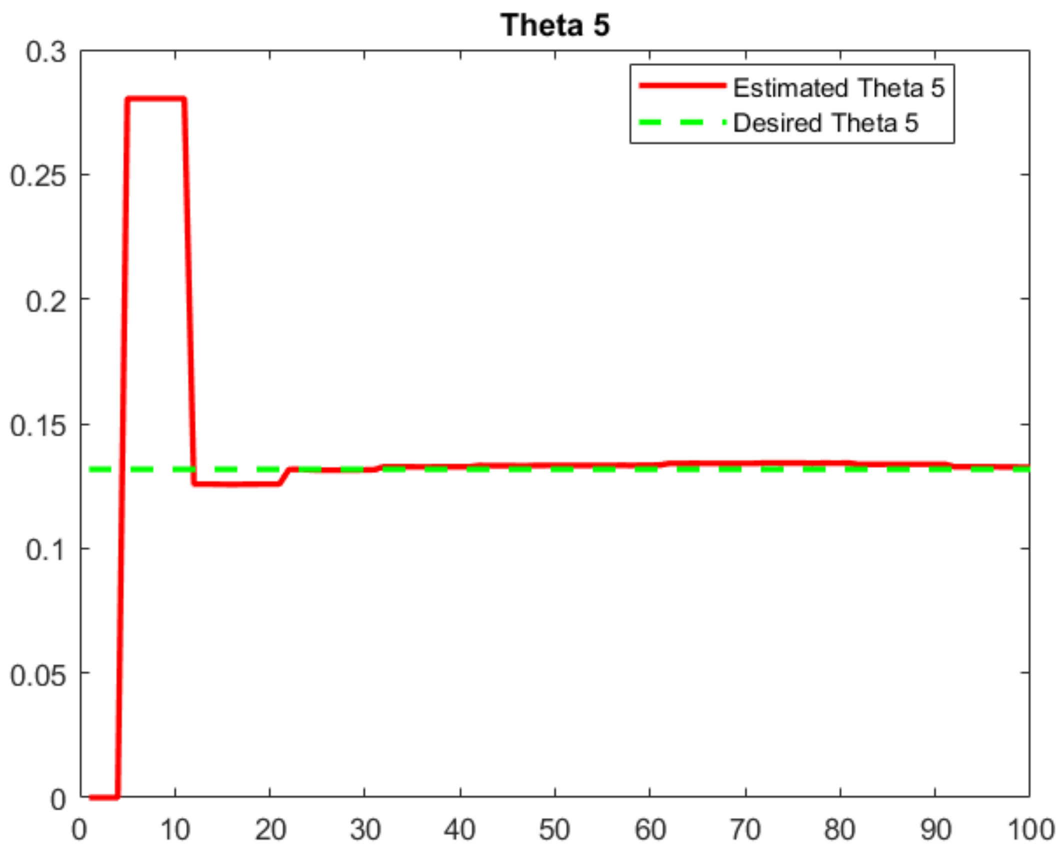
هشت شکل زیر نیز پارامترهای تخمین زده شده سیستم را در کنار پارامترهای واقعی نشان می‌دهند که برای هر پارامتر جداگانه رسم شده است.

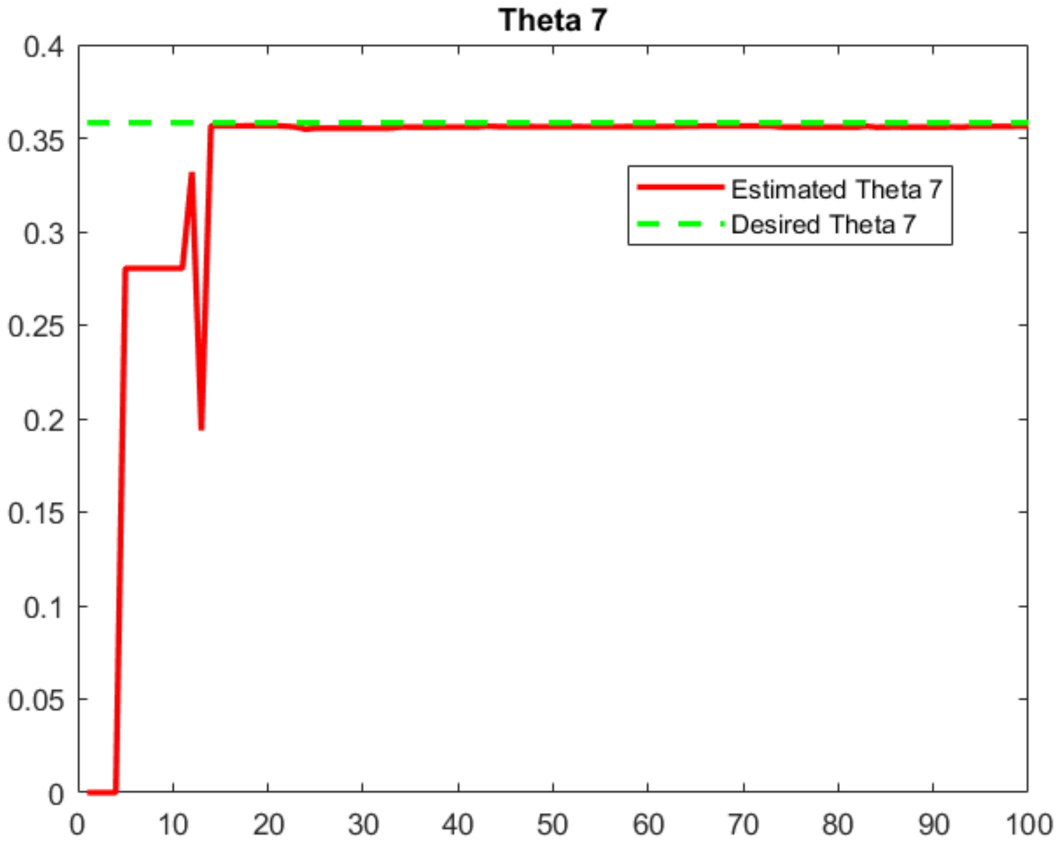

Theta 1
Estimated Theta 1
Desired Theta 1
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
10
20
30
40
50
60
70
80
90
100

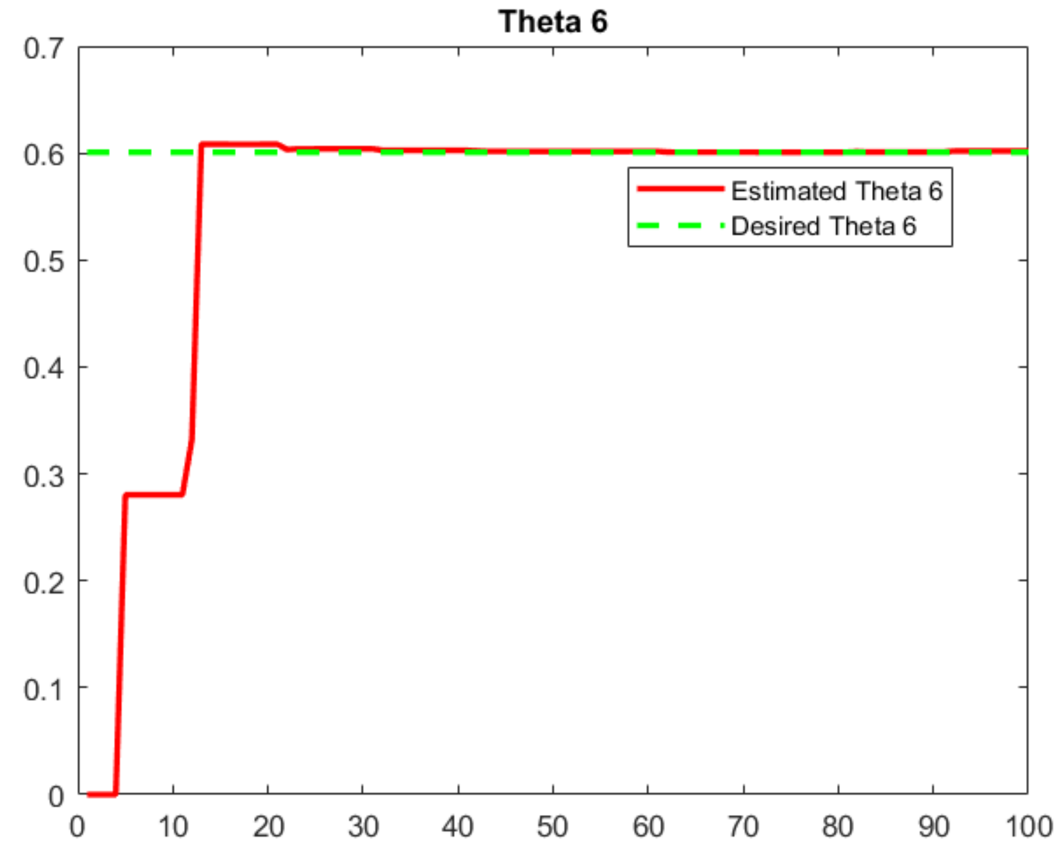

Theta 2
Estimated Theta 2
Desired Theta 2
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
10
20
30
40
50
60
70
80
90
100

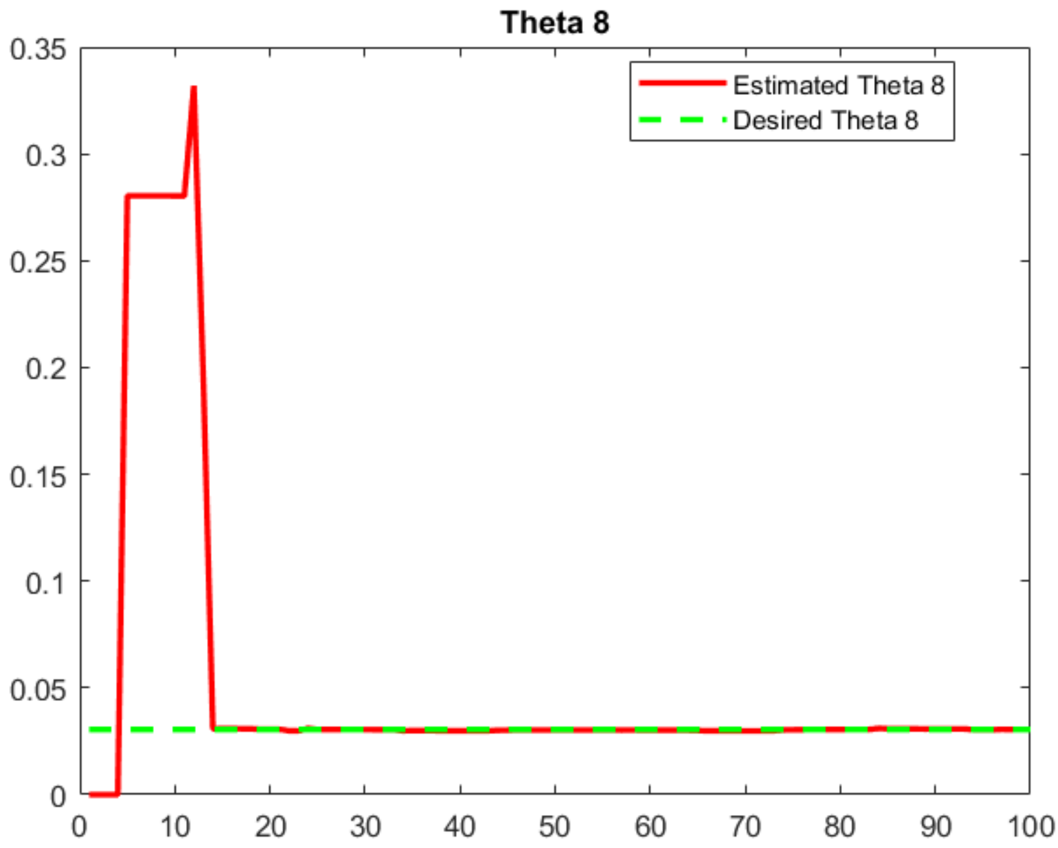

Theta 3
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0
10
20
30
40
50
60
70
80
90
100
Estimated Theta 3
Desired Theta 3


Theta 4
Estimated Theta 4
Desired Theta 4
0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0
10
20
30
40
50
60
70
80
90
100


Theta 5
Estimated Theta 5
Desired Theta 5
0.3
0.25
0.2
0.15
0.1
0.05
0
0
10
20
30
40
50
60
70
80
90
100


Theta 7
Estimated Theta 7
Desired Theta 7
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0 10 20 30 40 50 60 70 80 90 100


Theta 6
Estimated Theta 6
Desired Theta 6
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0 10 20 30 40 50 60 70 80 90 100



همانطور که مشاهده می‌شود مقدار پارامترها به خوبی به مقدارهای واقعی همگرا شده‌اند اگر هم جایی خطای بسیار کمی وجود دارد ناشی از وجود نویز است.

پیوست ۱

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%% System Identification %%%%%%%%%%%%%
%%%% ۴-April-۲۰۲۱

clc
clear
close all

global mu W mc G ma RB
```

در این قسمت پارامترهای سیستم و شرایط اولیه شبیه‌سازی تعریف شده‌اند.

```
mc = ۱۵۰;      % spacecraft's mass(kg)
ma = ۶.۶۸۷e۱۵; % ۴۳۳ Eros mass (kg)
RB = ۱۶;       % reference radius of Eros(km)
i = ۱۰.۲۸*pi/۱۸۰;     % inclination


hs = ۲۱۸e۶;       % height from sun (km)
Rs = ۶۹۶۳۴۲;      % sun radius(km)
ra = hs+Rs;       % radius from the sun to asteroid
G = ۶.۶۷۴۲e-۲۰;  % universal gravitational constant(km^۳/kg.s^۲)
ms = ۱.۹۸۹e۳۰;    % sun mass (kg)
mu = G*(ma+ms);   % gravitational parameter(km^۳/s^۲)
W = sqrt(mu/ra^۳); %#rad/s


%#initial condition
x۰ = [-۵ ۰ -۱۵]; dx۰ = [۰ ۴۰e-۳ ۰];
ra۰ = [-ra*cos(i) ۰ ra*sin(i)];
int_parameter = [۱/۱۳۰ ۱ ۱e-۷ ۱e-۱۴];
```

حل معادلات سیستم با استفاده از روش ode۴۵:

```
%%% ode solving
options = odeset('RelTol',۱e-۳,'AbsTol',۱e-۵);
int = [x۰ dx۰ ra۰ ۰ -sqrt(mu/ra) ۰ int_parameter ۲۰*ones(۱,۴) ۱*ones(۱,۴) ones(۱,۴)
۱۰*ones(۱,۴) x۰ dx۰];

    [t,x]=ode۴۵(@fun_SI,[۰ ۵۰۰],int);


%# plots
%%% estimation of spacecraft mass
figure
plot(t,۱./x(:,۱۳),t,mc*ones(numel(t),۱))
xlabel('t(s)'),ylabel('mc(kg)'),grid on
legend('estimation','desired')

%%% estimation of angular velocity
figure
plot(t,x(:,۱۵),t,W*ones(numel(t),۱))
xlabel('t(s)'),ylabel('W(rad/s)'),grid on
legend('estimation','desired')

%%% relative distance
figure
plot(t,sqrt(x(:,۱).^۲+x(:,۲).^۲+x(:,۳).^۲)),hold on
plot(t,sqrt(x(:,۳۳).^۲+x(:,۳۴).^۲+x(:,۳۵).^۲))
xlabel('t(s)'),ylabel('R(km)'),grid on
legend('estimation','desired')

%%% relative velocity
figure
plot(t,sqrt(x(:,۴).^۲+x(:,۵).^۲+x(:,۶).^۲)),hold on
plot(t,sqrt(x(:,۳۶).^۲+x(:,۳۷).^۲+x(:,۳۸).^۲))
```

```
xlabel('t(s)'),grid on
h = ylabel('$ \dot{R}(km/s) $');
set(h,'Interpreter','latex')
legend('estimation','desired')
```

برازش تابع پتانسیل با معادله مرتبه ۲:

```
%% gravitational potential fitting with a ۲-order system
for i = ۱:numel(t)
[U(:,i)] = gravitational_potential(t(i),x(i,:));
end

delta = ۱e-۵*sin(۰.۲*pi.*t);

figure
plot(t,sqrt((U(۱,:)+delta').^۲+(U(۲,:)+delta').^۲+(U(۳,:)+delta').^۲)),hold on

f=fit(t,U(۱,:)'+delta,'poly۲');
f=fit(t,U(۲,:)'+delta,'poly۲');
f=fit(t,U(۳,:)'+delta,'poly۲');


W۱۲ = (۷.۲۵۳e-۱۱  *t.^۲  -۱.۰۶۷e-۰۷ .*t -۵.۷۰۹e-۰۶);
W۲۲ = (۱.۵۱e-۱۲ *t.^۲ -۱.۶۶e-۰۹ .*t + ۶.۸۱e-۰۷ ) ;
W۳۲ = ( ۲.۱۵۱e-۱۰ *t.^۲ -۳.۰۰۶e-۰۷.*t -۱.۸۹۴e-۰۵ ) ;

plot(t,sqrt(W۱۲.^۲+W۲۲.^۲+W۳۲.^۲)),grid on
xlabel('t(s)'),legend('$ U+\delta $','$\hat{U}+\hat{\delta}$','Interpreter','latex')
```

پیوست ۲

```
function dx = fun_SI(t,x)
dx = zeros(۳۸,۱);

global mu W mc
```

فراخوانی تابع پتانسیل:

```
  [U] = gravitational_potential(t,x);
```

طراحی کنترل کننده لغزشی:

```
%%% sliding mode control
landa = diag([.۰۱ .۰۱ .۰۱]);k = [۱۰۰ ۱۰۰ ۱۰۰]; epsilon = ۰.۸;
e = x(۱:۳);de = x(۴:۶);s = de+landa*e;
Fx = mc*(-۲*W*x(۶)-U(۱)-landa(۱)*x(۴))-k(۱)*satlins(s(۱)/epsilon);
Fy = mc*(W^۲*x(۲)-U(۲)-landa(۲)*x(۵))-k(۲)*satlins(s(۲)/epsilon);
Fz = mc*(۲*W*x(۴)-۳*W^۲*x(۳)-U(۳)-landa(۳)*x(۶))-k(۳)*satlins(s(۳)/epsilon);

%%%%%%%% %%%%%% spacecraft dynamics system
```

شبیه‌سازی سیستم:

```
delta = ۱e-۵*sin(۰.۲*pi*t);  % disturbance



d۲x = ۲*W*x(۶)+(Fx/mc+U(۱)+delta);
d۲y = -W^۲*x(۲)+(Fy/mc+U(۲)+delta);
d۲z = -۲*W*x(۴)+۳*W^۲*x(۳)+Fz/mc+U(۳)+delta;

dx(۱)=x(۴);    %x(۱)=x
dx(۲)=x(۵);    %x(۲)=y
dx(۳)=x(۶);    %x(۳)=z
dx(۴)=d۲x;     %x(۴)=xdot
dx(۵)=d۲y;     %x(۵)=ydot
dx(۶)=d۲z;     %x(۶)=zdot
```

حرکت مداری سیارک(این قسمت جزء پروژه نیست.)

```
%%%%%%%% asteroid's motion relative to sun
ra = sqrt(x(۷)^۲+x(۸)^۲+x(۹)^۲);
d۲xa = -mu*x(۷)/(ra^۳);
d۲ya = -mu*x(۸)/(ra^۳);
d۲za = -mu*x(۹)/(ra^۳);
dx(۷) = x(۱۰);
dx(۸) = x(۱۱);
dx(۹) = x(۱۲);
dx(۱۰) = d۲xa;
dx(۱۱) = d۲ya;
dx(۱۲) = d۲za;

%%%%%%%%%%%%%%%%%%%%%%%%%
```

تشکیل ماتریس رگرسیو:

```
W۱۱ = Fx;
W۲۱ = Fy;
W۳۱ = Fz;

%%%%% gravitational potential fitting with a ۲-order system
W۱۲ = (۷.۲۵۳e-۱۱  *t^۲  -۱.۰۶۷e-۰۷ *t -۵.۷۰۹e-۰۶);
W۲۲ = (۱.۵۱e-۱۲ *t^۲ -۱.۶۶e-۰۹ *t + ۶.۸۱e-۰۷ )  ;
W۳۲ = ( ۲.۱۵۱e-۱۰*t^۲ -۳.۰۰۶e-۰۷*t -۱.۸۹۴e-۰۵ ) ;


W۱۳ = ۲*x(۶);
W۲۳ = ۰;
W۳۳ = -۲*x(۴);

W۱۴ = ۰;
W۲۴ = -x(۲);
W۳۴ = ۳*x(۳);
```

```
We = [W۱۱ W۱۲ W۱۳ W۱۴;W۲۱ W۲۲ W۲۳ W۲۴;W۳۱ W۳۲ W۳۳ W۳۴];   % signal vector

% phi = [۱/mc ۱ W W^۲]';     % system's parameters
```

پیاده‌سازی روش مبتنی بر خطای پیش بینی و تخمین پارامترها:

```
phi_hat = x(۱۳:۱۶);
tau_tilda =  [d۲x;d۲y;d۲z]-We*phi_hat;

P = [x(۱۷) x(۲۱) x(۲۵) x(۲۹);...
    x(۱۸) x(۲۲) x(۲۶) x(۳۰);...
    x(۱۹) x(۲۳) x(۲۷) x(۳۱);...
    x(۲۰) x(۲۴) x(۲۸) x(۳۲)];% Kalman Filter matrix

landa۰ = diag([.۳ .۰۱ .۱ .۱]); k۰ = ۱;
la = landa۰*(۱-norm(P)/k۰);  %#forgetting factor
dp = la*P-P*(We')*We*P;
dphi_hat = P*We'*tau_tilda;

dx(۱۳:۱۶) = dphi_hat;
dx(۱۷:۳۲) = dp;
```

شبیه سازی سیستم شناسایی شده برای مقایسه با سیستم واقعی:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Estimation of system
landa = diag([.۰۱ .۰۱ .۰۱]);k = [۱۰ ۱۰ ۱۰]; epsilon = ۰.۸;
e = x(۳۳:۳۵);de = x(۳۶:۳۸);s = de+landa*e;
Fx = (۱/x(۱۳))*(-۲*x(۱۵)*x(۳۶)-W۱۲-landa(۱)*x(۳۶))-k(۱)*satlins(s(۱)/epsilon);
Fy = (۱/x(۱۳))*(x(۱۵)^۲*x(۳۴)-W۲۲-landa(۲)*x(۳۷))-k(۲)*satlins(s(۲)/epsilon);
Fz = (۱/x(۱۳))*(۲*x(۱۵)*x(۳۶)-۳*x(۱۵)^۲*x(۳۵)-W۳۲-landa(۳)*x(۳۸))-k(۳)*satlins(s(۳)/epsilon);


d۲x_hat = ۲*x(۱۵)*x(۳۸)+(Fx*x(۱۳)+W۱۲);
```

```
d۲y_hat = -x(۱۵)^۲*x(۳۴)+(Fy*x(۱۳)+W۲۲);

d۲z_hat = -۲*x(۱۵)*x(۳۶)+۳*x(۱۵)^۲*x(۳۵)+Fz*x(۱۳)+W۳۲;


dx(۳۳)=x(۳۶);    ٪x(۱)=x

dx(۳۴)=x(۳۷);    ٪x(۲)=y

dx(۳۵)=x(۳۸);    ٪x(۳)=z

dx(۳۶)=d۲x_hat;    ٪x(۴)=xdot

dx(۳۷)=d۲y_hat;    ٪x(۵)=ydot

dx(۳۸)=d۲z_hat;    ٪x(۶)=zdot



end
```

پیوست ۳

تعریف تابع پتانسیل گرانشی:

```
function [U] = gravitational_potential(~,x)

global RB G ma
r = sqrt(x(۱)^۲+x(۲)^۲+x(۳)^۲)+RB;
phi = atan۲(sqrt(x(۱)^۲+x(۲)^۲),x(۳));  ٪# latitude angle
landa = atan۲(x(۲),x(۱));   ٪# Longitudinal angle
```

ضرایب لژاندر:

```
٪# Legendre coefficients
P۰۰ = ۱; P۱۰ = sin(phi); P۱۱ = cos(phi);
P۲۰ = ۰.۵*(۳*sin(phi)^۲-۱); P۲۱ = ۳*sin(phi)*cos(phi); P۲۲ = ۳*cos(phi)^۲*sin(phi);
P۳۰ = ۰.۵*(۵*sin(phi)^۳-۳*sin(phi)); P۳۱ = ۰.۵*cos(phi)*(۱۵*sin(phi)^۲-۳); P۳۲ =
۱۵*cos(phi)^۲*sin(phi);
P_sinphi = [P۰۰ P۱۰ P۱۱;P۲۰ P۲۱ P۲۲;P۳۰ P۳۱ P۳۲];
```

ضرایب استوکس:

```
٪# Stokes' coefficients
C = [۰ ۰.۰۰۱۱۷۵ -۰.۰۰۰۳۴۸;-۰.۰۵۲۸۵۱ ۰.۰۰۰۱۰۲ ۰.۰۸۳۲۰۳; -۰.۰۰۱۷۴۷ ۰.۰۰۴۰۸۳ ۰.۰۰۲۱۲۹];
S = [۰ ۰ ۰.۰۰۰۰۸۸;۰ ۰.۰۰۰۰۱۲ -۰.۰۲۸۰۲۳; ۰ ۰.۰۰۳۴۰۴ -۰.۰۰۰۸۳۶];


U_bar = zeros(۱,۱);
٪# gravitational potential function
for n = ۲:۳
for m = ۱:n
U_bar = U_bar +(RB/(r))^n*P_sinphi(n,m)+C(n,m)*cos(m*landa)+S(n,m)*sin(m*landa);
end
end

U = (G*ma/r)*(۱+U_bar);


ux = U*sin(phi)*cos(landa);
```

```
uy = U*sin(phi)*sin(landa);
uz = U*cos(phi);
U = [ux uy uz];


end
```

پیوست ۴

```
clear;
clc;
close all;
```

داده‌های مساله

```
N=۱۰۰۰;
T=۰.۱;
Mc=۱۵۰;
R=۱۶e۳;
Ma=۶.۶۸۷e۱۵;
G=۶.۶۷e-۱۱;
w=۱.۱۲e-۷;
```

داده‌دهی رندوم برای ورودی کنترلی f که در تابع پتانسیل گرانشی اعمال می‌شود:

```
u۱=۲+۵*randn(۱,N);
u۲=۲+۵*randn(۱,N);
u۳=۲+۵*randn(۱,N);

for i=۱:N
  u۱(i)=u۱(i)+۰.۱*I;
  u۲(i)=u۲(i)+۰.۱*I;
  u۳(i)=u۳(i)+۰.۱*I;
end
```

مدلسازی نویز:

```
noise۱=۰.۰۰۰۱*randn(۱,N);
noise۲=۰.۰۰۰۱*randn(۱,N);
noise۳=۰.۰۰۰۱*randn(۱,N);


x۱=zeros(۱,N);
x۲=zeros(۱,N);
x۳=zeros(۱,N);
x۴=zeros(۱,N);
```

```
x۵=zeros(۱,N);
x۶=zeros(۱,N);

for i=۱:N-۱
   x۱(i+۱)=x۱(i)+T*x۲(i);
   x۲(i+۱)=x۲(i)+۲*T*w*x۶(i)+(۱/Mc)*T*u۱(i)+T*G*Mc*Ma/(R+x۱(i))^۲+noise۱(i);
   x۳(i+۱)=x۳(i)+T*x۴(i);
   x۴(i+۱)=x۴(i)-T*w^۲*x۳(i)+(۱/Mc)*T*u۲(i)+T*G*Mc*Ma/(R+x۳(i))^۲+noise۲(i);
   x۵(i+۱)=x۵(i)+T*x۶(i);
   x۶(i+۱)=x۶(i)-
۲*T*w*x۲(i)+۳*T*w^۲*x۵(i)+(۱/Mc)*T*u۳(i)+T*G*Mc*Ma/(R+x۵(i))^۲+noise۳(i);

end

U=[u۱' u۲' u۳'];
Y=[x۱' x۳' x۵'];
```

پیوست ۵

مدلسازی گرانش برای شبکه‌ی عصبی

```
function [U] = gravitational_potential(~,x)

global RB G ma
r = sqrt(x(١)^٢+x(٢)^٢+x(٣)^٢)+RB;
phi = atan٢(sqrt(x(١)^٢+x(٢)^٢),x(٣));  ٪# latitude angle
landa = atan٢(x(٢),x(١));   ٪# Longitudinal angle

٪# Legendre coefficients
P٠٠ = ١; P١٠ = sin(phi); P١١ = cos(phi);
P٢٠ = ٠/٥*(٣*sin(phi)^٢-١); P٢١ = ٣*sin(phi)*cos(phi); P٢٢ = ٣*cos(phi)^٢*sin(phi);
P٣٠ = ٠/٥*(٥*sin(phi)^٣-٣*sin(phi)); P٣١ = ٠/٥*cos(phi)*(١٥*sin(phi)^٢-٣); P٣٢ =
١٥*cos(phi)^٢*sin(phi);
P_sinphi = [P٠٠ P١٠ P١١;P٢٠ P٢١ P٢٢;P٣٠ P٣١ P٣٢];

٪# Stokes' coefficients
C = [٠ ٠/٠٠١١٧٥ _٠/٠٠٠٣٤٨;_٠/٠٥٢٨٥١ ٠/٠٠٠١٠٢ ٠/٠٨٣٢٠٣; _٠/٠٠١٧٤٧ ٠/٠٠٤٠٨٣ ٠/٠٠٢١٢٩];
S = [٠ ٠ ٠/٠٠٠٠٨٨;٠ ٠/٠٠٠٠١٢ _٠/٠٢٨٠٢٣; ٠ ٠/٠٠٣٤٠٤ _٠/٠٠٠٨٣٦];


   U_bar = zeros(١'١);
٪# gravitational potential function
for n = ٢:٣
   for m = ١:n
      U_bar = U_bar +(RB/(r))^n*P_sinphi(n,m)+C(n,m)*cos(m*landa)+S(n,m)*sin(m*landa);
   end
end

      U = (G*ma/r)*(١+U_bar);
```

```
ux = U*sin(phi)*cos(landa);

uy = U*sin(phi)*sin(landa);

uz = U*cos(phi);

U = [ux uy uz];


end
```

پیوست ۶

شبکه عصبی چند لایه

```
٪ Solve an Input-Output Fitting problem with a Neural Network
٪ This script assumes these variables are defined:
٪   U - input data.
٪   Y - target data.

x = U';
t = Y';

٪ Choose a Training Function
٪ For a list of all training functions type: help nntrain
٪ 'trainlm' is usually fastest.
٪ 'trainbr' takes longer but may be better for challenging problems.
٪ 'trainscg' uses less memory. Suitable in low memory situations.
trainFcn = 'trainlm';  ٪ Levenberg-Marquardt backpropagation.

٪ Create a Fitting Network
hiddenLayerSize = ۱۰;
net = fitnet(hiddenLayerSize,trainFcn);

٪ Choose Input and Output Pre/Post-Processing Functions
٪ For a list of all processing functions type: help nnprocess
net.input.processFcns = {'removeconstantrows','mapminmax'};
net.output.processFcns = {'removeconstantrows','mapminmax'};

٪ Setup Division of Data for Training, Validation, Testing
٪ For a list of all data division functions type: help nndivision
net.divideFcn = 'dividerand';  ٪ Divide data randomly
net.divideMode = 'sample';  ٪ Divide up every sample
net.divideParam.trainRatio = ۱۰۰/۷۰;
```

```
net.divideParam.valRatio = ۱۰۰/۱۵;
net.divideParam.testRatio = ۱۰۰/۱۵;

٪ Choose a Performance Function
٪ For a list of all performance functions type: help nnperformance
net.performFcn = 'mse';  ٪ Mean Squared Error

٪ Choose Plot Functions
٪ For a list of all plot functions type: help nnplot
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
    'plotregression', 'plotfit'};

٪ Train the Network
[net,tr] = train(net,x,t);

٪ Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)

٪ Recalculate Training, Validation and Test Performance
trainTargets = t .* tr.trainMask{۱};
valTargets = t .* tr.valMask{۱};
testTargets = t .* tr.testMask{۱};
trainPerformance = perform(net,trainTargets,y)
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)

٪ View the Network
view(net)

٪ Plots
٪ Uncomment these lines to enable various plots.
```

```
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotregression(t,y)
%figure, plotfit(net,x,t)

% Deployment
% Change the (false) values to (true) to enable the following code blocks.
% See the help for each generation function for more information.
if (false)
   % Generate MATLAB function for neural network for application
   % deployment in MATLAB scripts or with MATLAB Compiler and Builder
   % tools, or simply to examine the calculations your trained neural
   % network performs.
   genFunction(net,'myNeuralNetworkFunction');
   y = myNeuralNetworkFunction(x);
end
if (false)
   % Generate a matrix-only MATLAB function for neural network code
   % generation with MATLAB Coder tools.
   genFunction(net,'myNeuralNetworkFunction','MatrixOnly','yes');
   y = myNeuralNetworkFunction(x);
end
if (false)
   % Generate a Simulink diagram for simulation or deployment with.
   % Simulink Coder tools.
   gensim(net);
end
```

منابع

[۱] AlandiHallaj, M., & Assadian, N. (۲۰۱۷). Soft landing on an irregular shape asteroid using multiple-horizon multiple-model predictive control. *Acta Astronautica, ۱۴۰*, ۲۲۵-۲۳۴. doi:۱۰.۱۰۱۶/j.actaastro.۲۰۱۷.۰۸.۰۱۹

[۲] جزوه‌ی درسی. درس شناسایی سیستم و تخمین پارامترهای پروازی. استاد درس: دکتر خوشنود. ۱۴۰۰-۱۳۹۹

ممنون از تدریس خوبتان.

با احترام
مهسا آزادمنش
شماره دانشجویی ۹۹۰۰۴۲۶