

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



**چکيده**

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.

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

**فهرست مطالب**

|  |  |
| --- | --- |
| عنوان | صفحه |

[فصل اول: مدل دینامیکی فضاپیما 1](#_Toc77094356)

[تعریف مسئله 1](#_Toc77094357)

[مدل دینامیکی 1](#_Toc77094358)

[طراحی کنترلکننده 1](#_Toc77094359)

[شناسایی سیستم بر اساس خطای پیشبین 2](#_Toc77094360)

[فصل دوم: شبیه‌سازی و نتایج روش خطای پیشبین 3](#_Toc77094361)

[شبیهسازی 3](#_Toc77094362)

[نتایج شبیهسازی 4](#_Toc77094363)

[نتیجه گیری 7](#_Toc77094364)

[فصل سوم: تکرار روش با شبکه عصبی چند لایه 8](#_Toc77094365)

[شناسایی با شبکه عصبی 8](#_Toc77094366)

[نتایج شبکه عصبی 11](#_Toc77094367)

[فصل چهارم: اعمال دو روش LS و RLS 14](#_Toc77094368)

[شناسایی سیستم با روش LS 14](#_Toc77094369)

[شناسایی سیستم با روش RLS 20](#_Toc77094370)

[پیوست 1 26](#_Toc77094371)

[پیوست ۲ 29](#_Toc77094372)

[پیوست ۳ 33](#_Toc77094373)

[پیوست ۴ 35](#_Toc77094374)

[پیوست ۵ 37](#_Toc77094375)

[پیوست ۶ 39](#_Toc77094376)

[منابع 42](#_Toc77094377)

# فصل اول: مدل دینامیکی فضاپیما

## تعریف مسئله

هدف این پژوهش شناسایی مدل دینامیکی یک فضاپیما حول سیارک اروس ۴۳۳ است. در این پروژه مدل دینامیکی فضاپیما در نزدیکی سیارک، 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. در ادامه با استفاده از روش تخمین مبتنی بر خطای پیش­بینی با ضریب فراموشی 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.

که موقعیت فضاپیما نسبت به دستگاه مختصات در مرکز جرم سیارک و بیانگر سرعت زاویه ای سیارک است. برای سادگی حرکت سیارک در مدار دایروی فرض شده است. نیز بیانگر نیروی های کنترلی، جرم فضاپیما، پتانسیل گرانشی سیارک و نیز نیروهای اغتشاشی هستند.

تابع پتانسیل مطابق با رابطه زیر به دست می آید.



که در آن فاصله از فضاپیما تا مرکز جرم سیارک، زاویه عرضی، زاویه طولی، جرم سیارک، شعاع مرجع سیارک، ضرایب لژاندر ، 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

که در آن خطا و ضریب وزن دهی است و با استفاده از رابطه و تعیین ورودی کنترلی انجام می­شود. با تعریف ورودی کنترلی به صورت زیر تعریف می­شود.

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 و است.

مدل سیستم به صورت پارامتری خطی مطابق رابطه­ی زیر بیان می­شود.

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

که تخمین تابع پتانسیل و اغتشاشات است که به صورت سیستم مرتبه دو مدل می­شود.

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

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

که ضریب فراموشی است و برای بهبود ردیابی پارامترها استفاده می­شود. و از رابطه زیر حاصل می­شود.

که بهره­ای است که بر سرعت همگرایی پارامترها تاثیر دارد و ضریب فراموشی ثابت است.

# فصل دوم: شبیه‌سازی و نتایج روش خطای پیش­بین

## شبیه­سازی

مقادیر پارامترهای سیستم به صورت زیر تعریف می­شود.

|  |  |  |
| --- | --- | --- |
| جرم فضاپیما |  | 150 (kg) |
| سرعت زاویه ای سیارک |  | 1.12e-7(rad/s) |
| فاصله سیارک از خورشید |  | 216696342(km) |
| جرم سیارک |  | 6.687e15(km) |
| شعاع مرجع سیارک |  | 16(km) |

شرایط اولیه نیز به صورت و در نظر گرفته می­شود و فرض می­شود اغتشاش 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 1: پارامتر جرم

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



شکل 2: 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 3: 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 4: فاصله نسبی بین فضاپیما و سیارک



شکل 5: سرعت نسبی بین فضاپیما و سیارک

## نتیجه گیری

در این پروژه مدل دینامیکی یک فضاپیما حول سیارک 433 Eros 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:

متغیرهای حالت را به صورت زیر تعریف می‌کنیم:

مدل فضای حالت به فرم زیر تبدیل می شود:

خروجی:

نیروهای پتانسیل را به صورت زیر در نظر می‌گیریم:

بنابراین:

گسسته سازی سیستم با زمان نمونه برداری T:

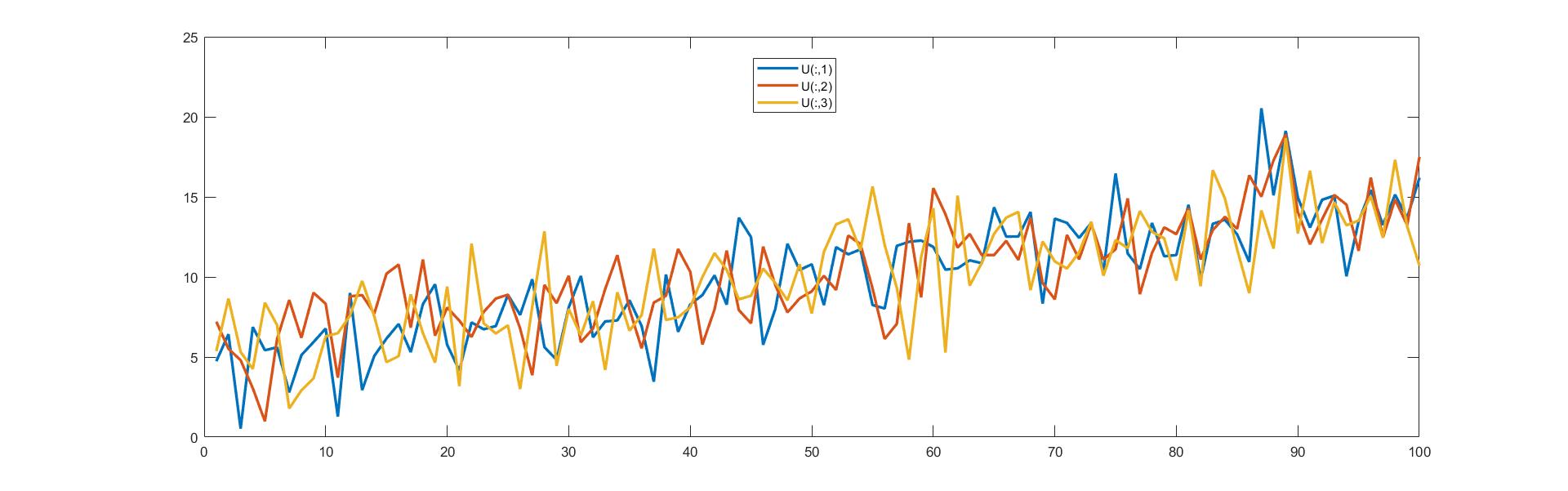
حال می‌توان شکل گسسته سازی شده معادلات حالت را نوشت:

با در نظر گرفتن پارامترها به صورت جدول قسمت قبل و همچنین در نظر گرفتن نیروهای مختلف در حد چند نیوتن و البته اغتشاش های جمع شونده با دامنه بسیار کوچک، داده های مختلف ورودی و خروجی را به دست می‌آوریم سیستم دارای سه ورودی و سه خروجی است. سپس با آموزش یک شبکه عصبی MLP یک مدل برای سیستم با استفاده از داده‌های ورودی و خروجی به دست آمده تخمین می‌زنیم.

## 

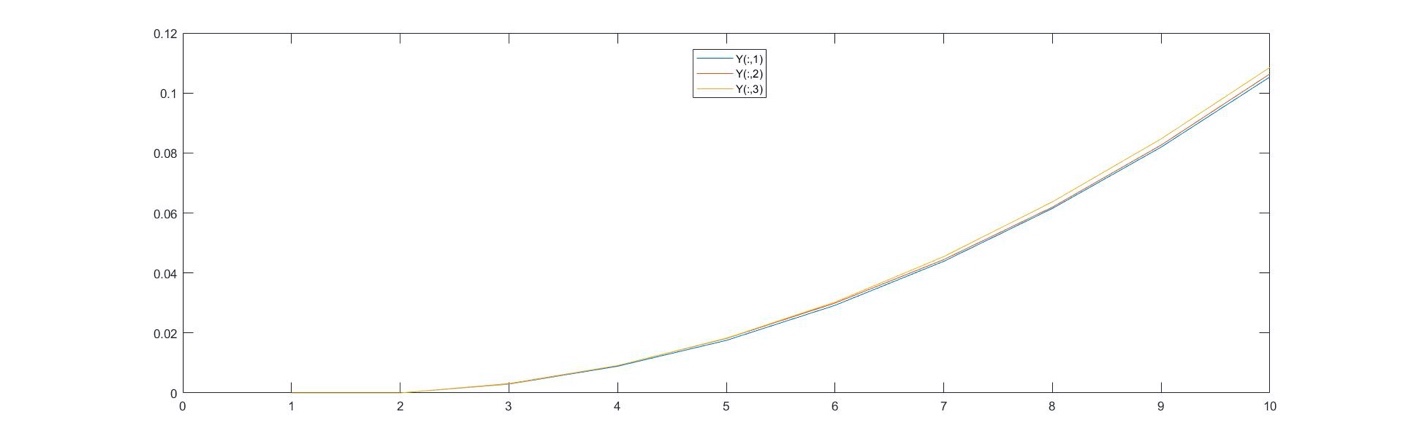
## نتایج شبکه عصبی

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.



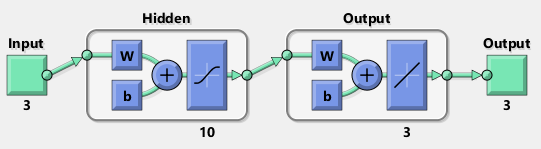
شکل ۷: خروجی‌های سیستم در نمونه‌های مختلف

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.

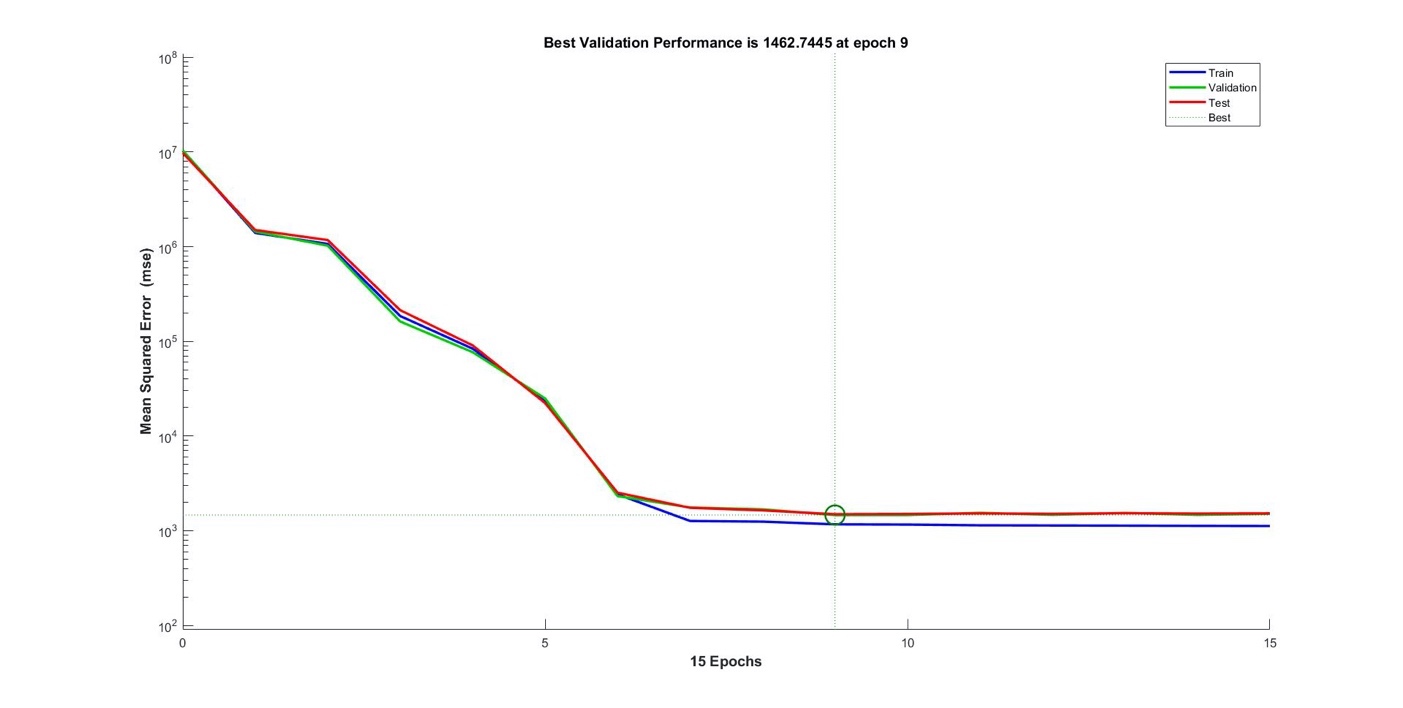
برای آموزش شبکه از ۷۰ درصد داده‌ها برای آموزش از ۱۵ درصد داده‌ها برای اعتبار سنجی و از ۱۵ درصد داده‌ها نیز برای تست شبکه پس از آموزش استفاده کرده‌ایم.

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:



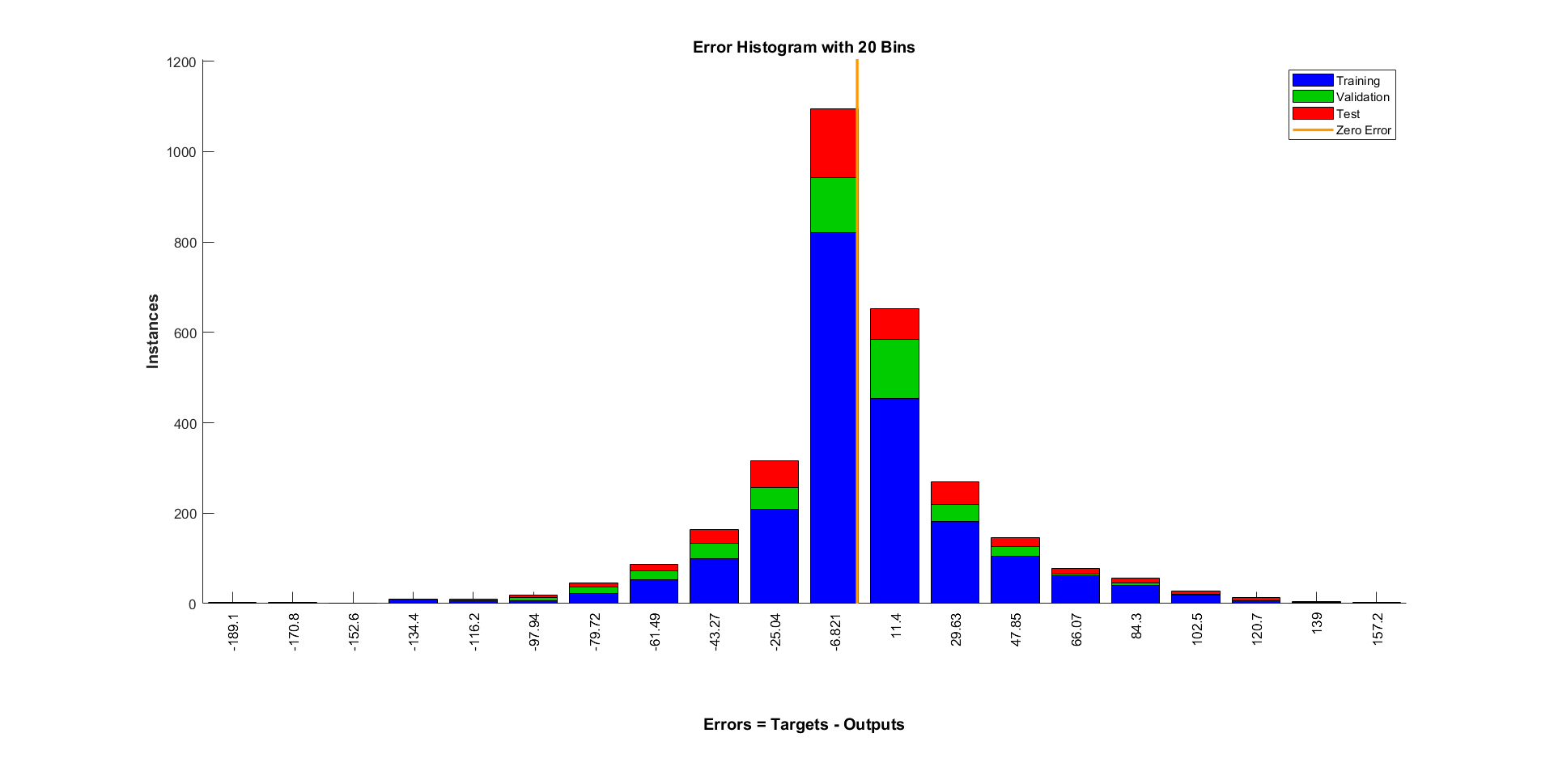
شکل ۸: شکل کلی شبکه

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



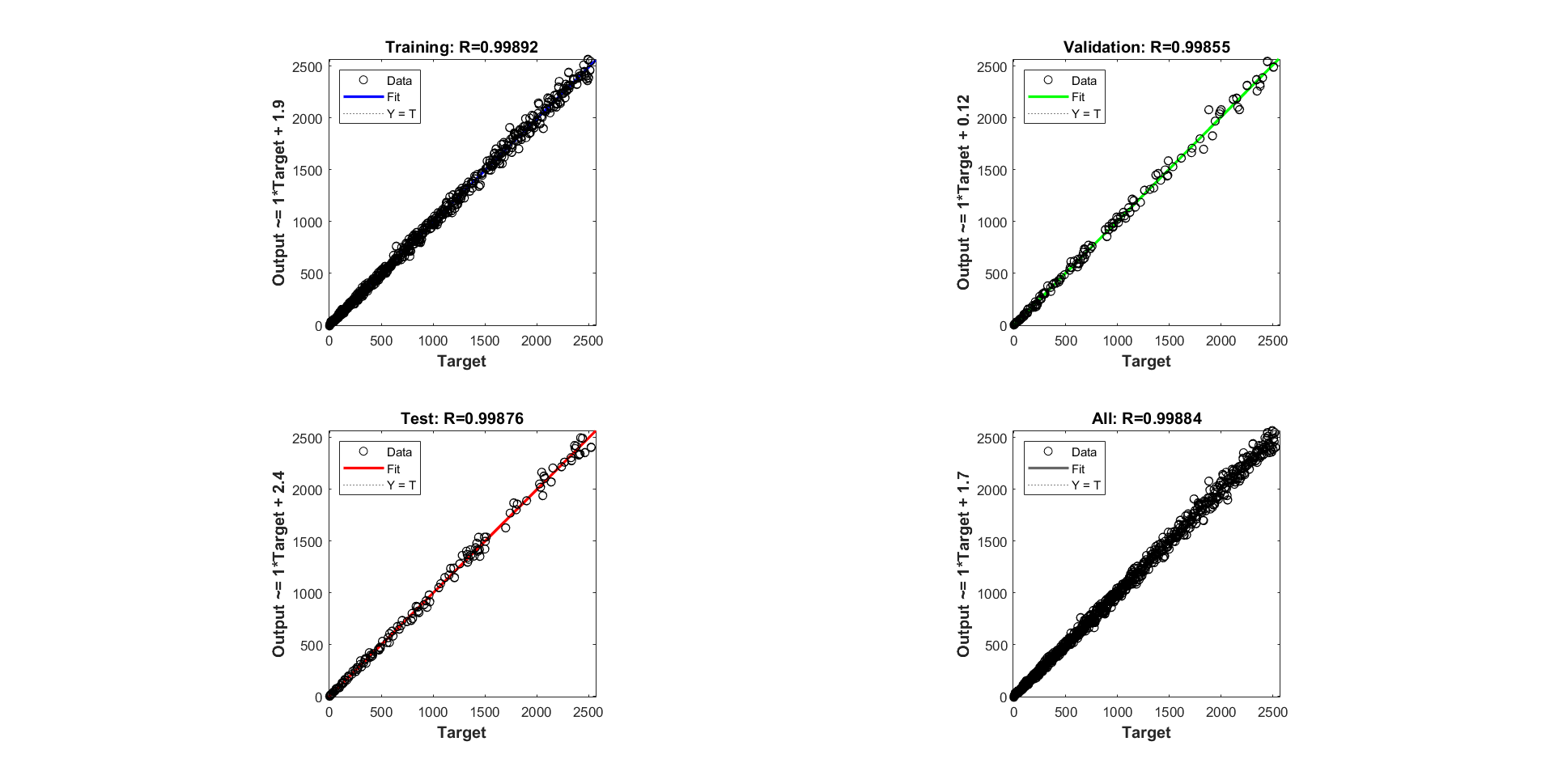
شکل ۹: عملکرد شبکه و میزان خطا

نمودار هیستوگرام خطا به صورت زیر است:



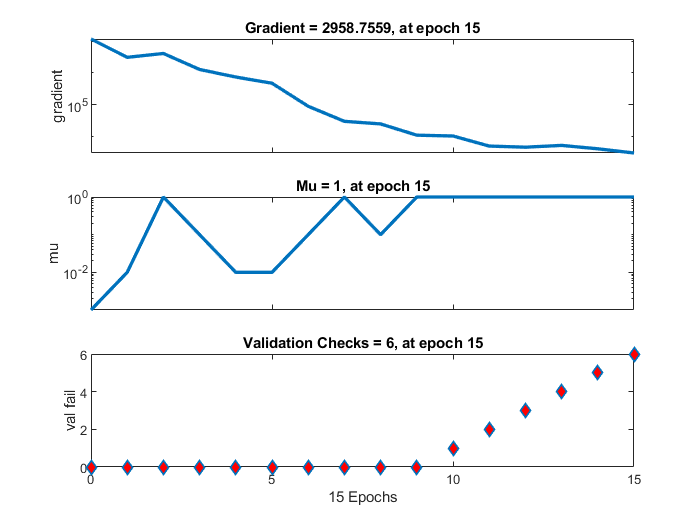
شکل۱۰: نمودار هیستوگرام خطا

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:

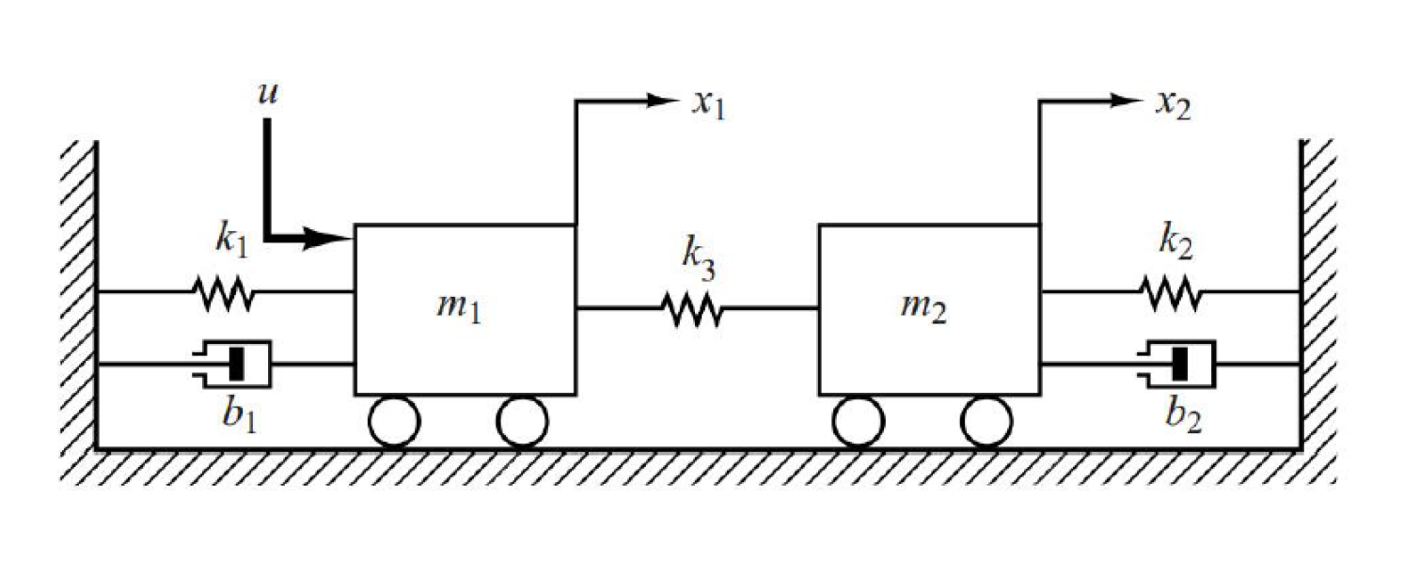
 شکل۱۲: سایز پارامترهای شبکه

# 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.**

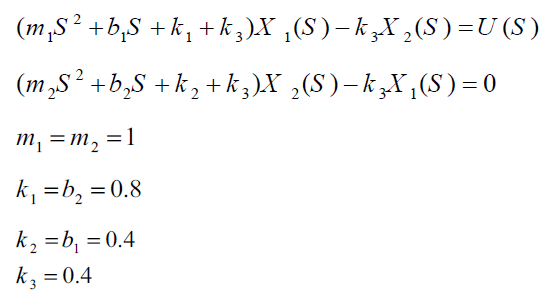
## شناسایی سیستم با روش LS

سیستم زیر را در نظر می‌گیریم.

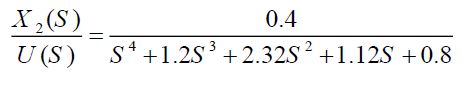


شکل۱۳: سیستم مدنظر برای بخش دوم پروژه

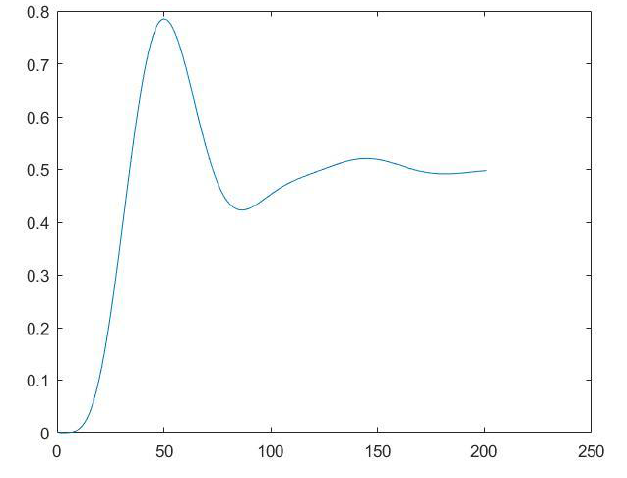
معادلات حاکم بر سیستم و همچنین مقادیر پارامترها به صورت زیر اند.



تابع تبدیل سیستم به صورت زیر به دست می‌آید:

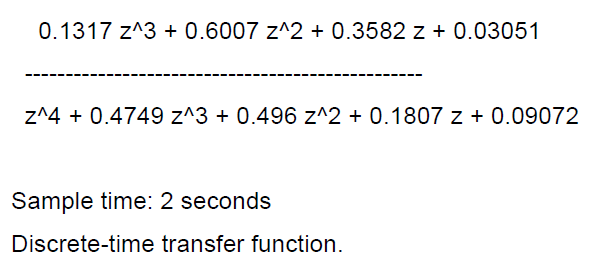


پاسخ پله زمان پیوسته این سیستم به صورت شکل زیر است:



شکل۱۴: پاسخ پله زمان پیوسته

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



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

فرم رگرسیون خطی به صورت زیر است:

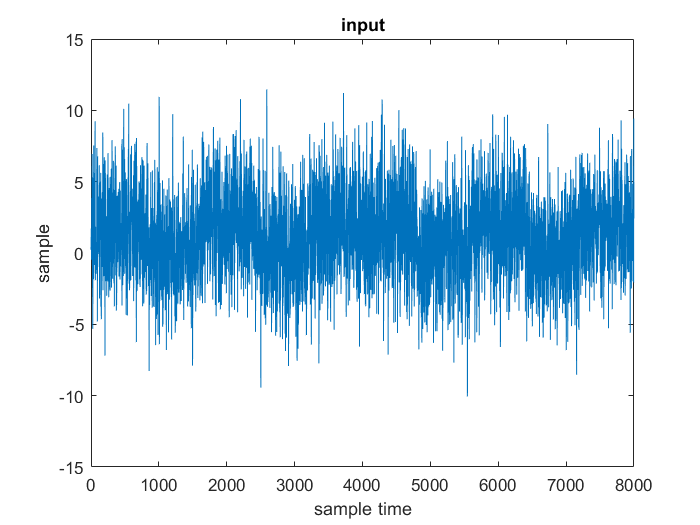
پارامترها

رگرسورها

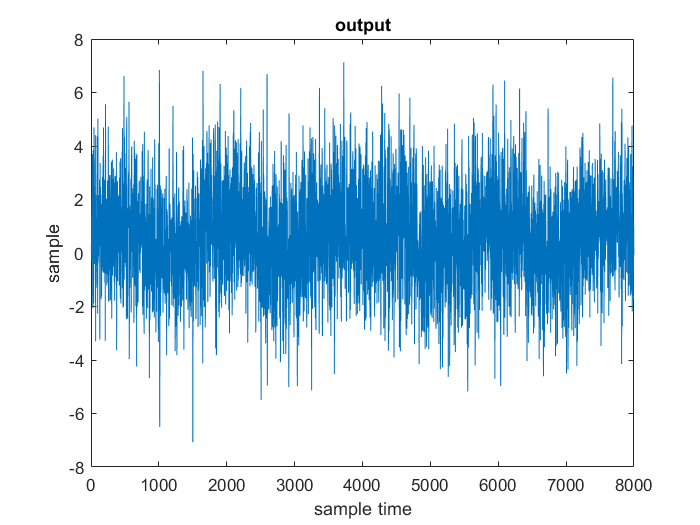
نویز سفید گوسی

پارامتر های واقعی سیستم بردار 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:

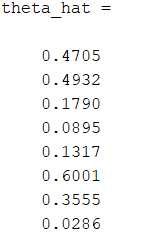


خروجی سیستم نیز به صورت زیر به دست می آید:



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

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



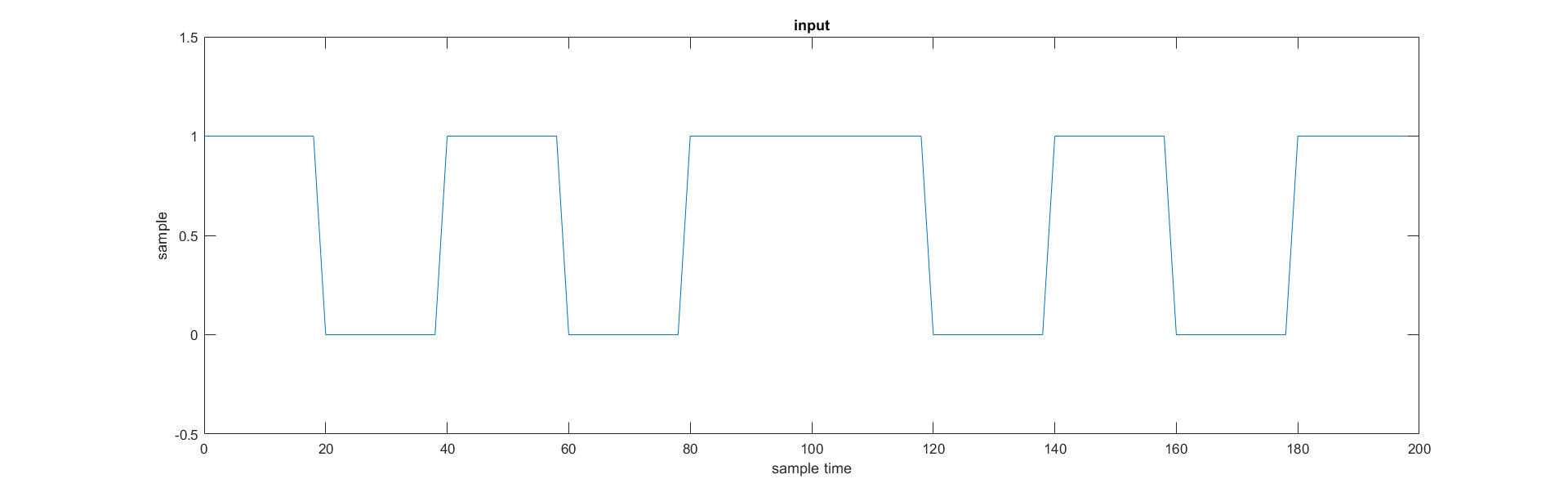
همانطور که مشاهده می شود پارامترهای تخمین زده به پارامترهای واقعی سیستم بسیار نزدیک کند پارامترهای واقعی و تخمین سیستم در جدول زیر آورده شده اند:

|  |  |  |
| --- | --- | --- |
| نام پارامتر | مقدار واقعی | مقدار تخمینی |
|  | ۰.۴۷۴۹ | ۰.۴۷۰۵ |
|  | ۰.۴۶۹ | ۰.۴۹۳۲ |
|  |  | ۰.۱۷۹۰ |
|  |  | ۰.۰۸۹۵ |
|  |  | ۰.۱۳۱۷ |
|  |  | ۰.۶۰۰۱ |
|  |  | ۰.۳۵۵۵ |
|  |  | ۰.۰۲۸۶ |

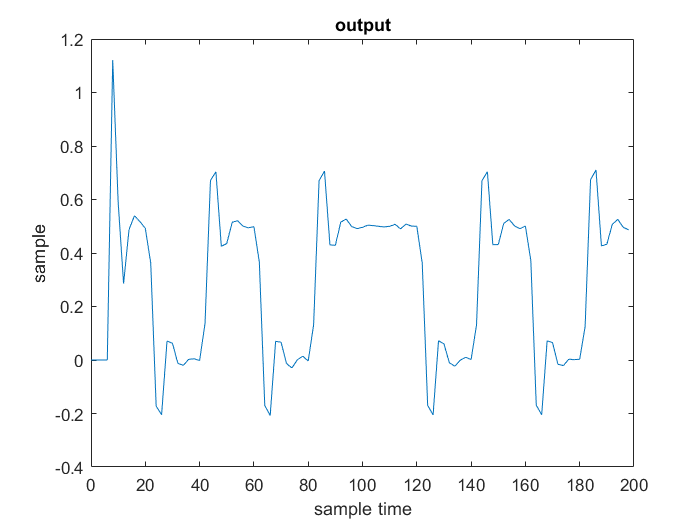
همانطور که مشاهده می‌شود مقدارهای تخمینی و مقدارهای واقعی بسیار نزدیک هستند.

## شناسایی سیستم با روش RLS

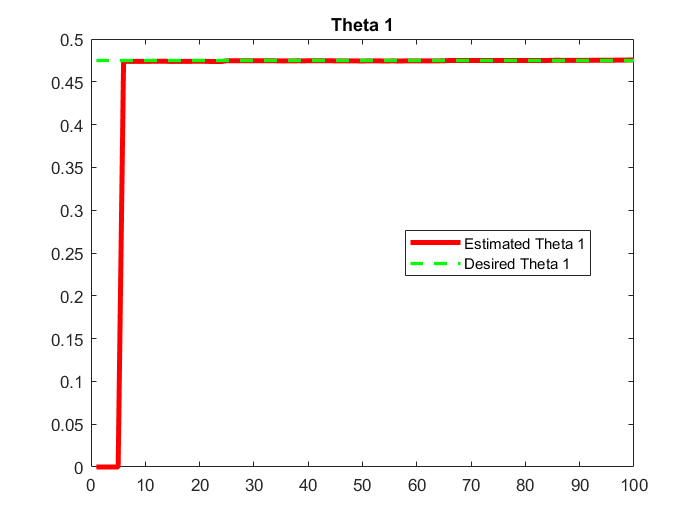
در این قسمت با روش RLS پارامترها را تخمین می‌زنیم. شکل زیر ورودی را نشان می‌دهد. ورودی سیگنال پالس می‌باشد. همچنین نویز جمع شونده گوسی نیز وجود دارد:

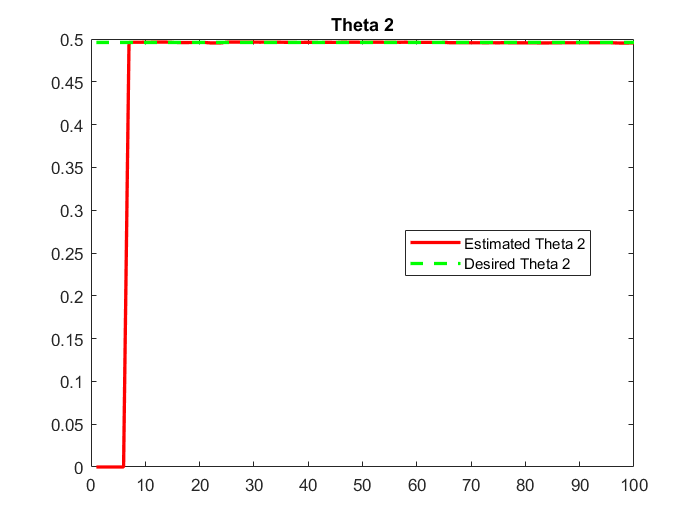


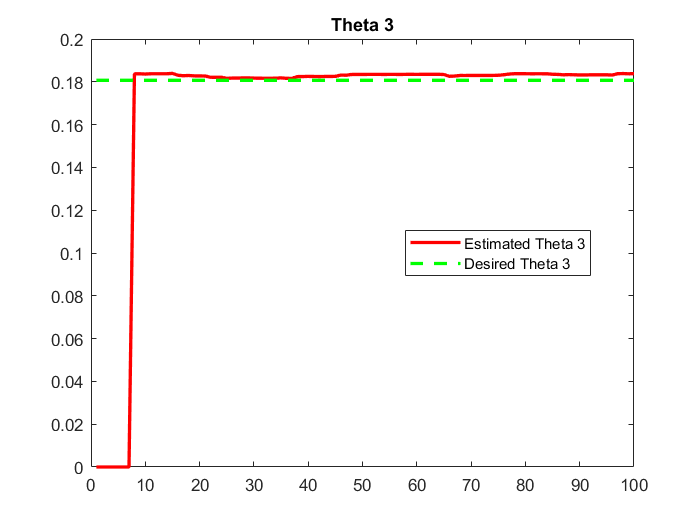
شکل زیر نیز خروجی سیستم را نشان می‌دهد:

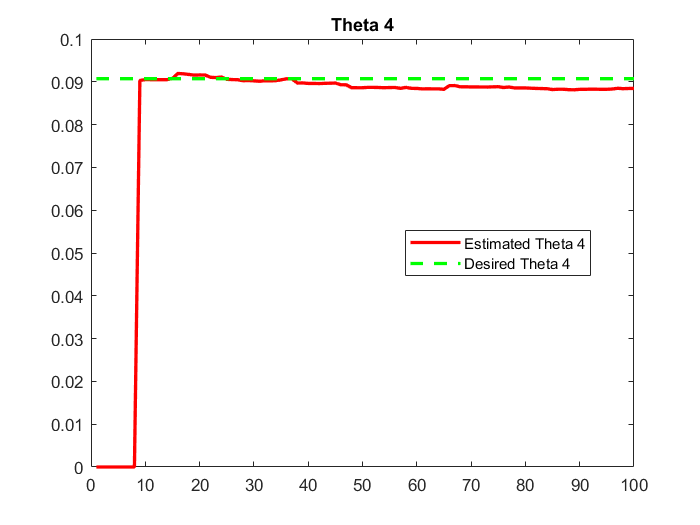


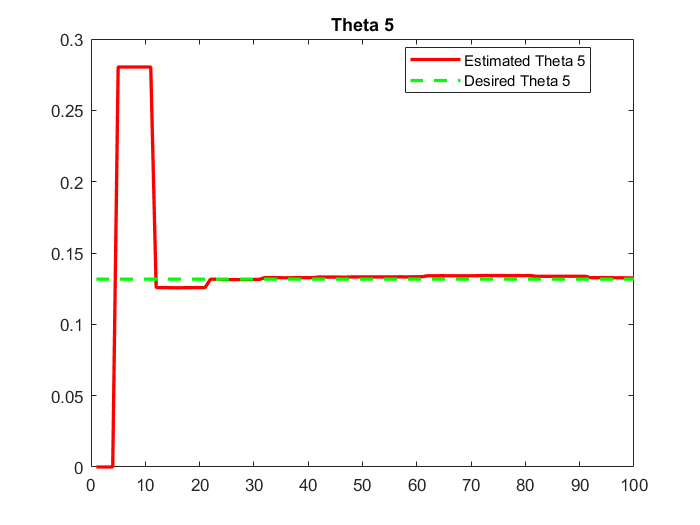
هشت شکل زیر نیز پارامترهای تخمین زده شده سیستم را در کنار پارامترهای واقعی نشان می‌دهند که برای هر پارامتر جداگانه رسم شده است.‌

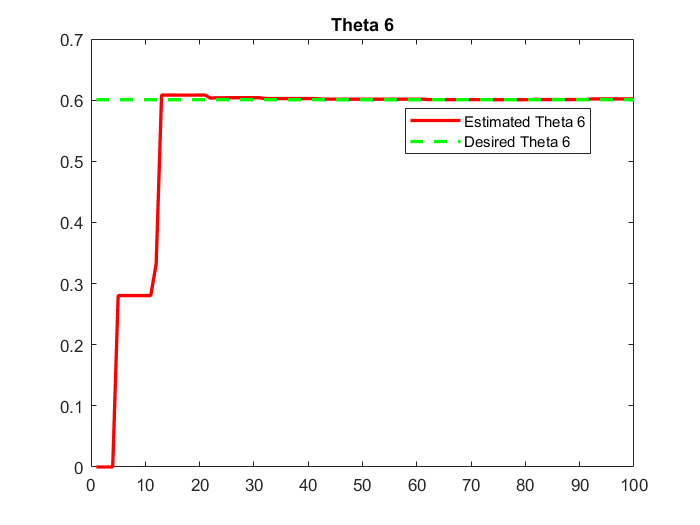
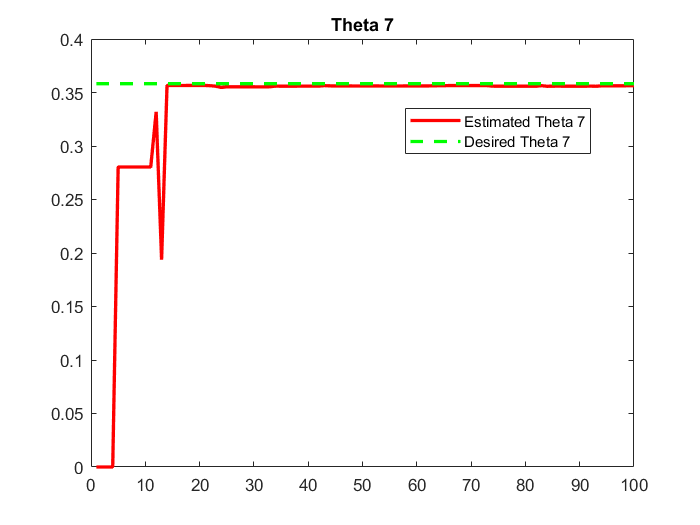


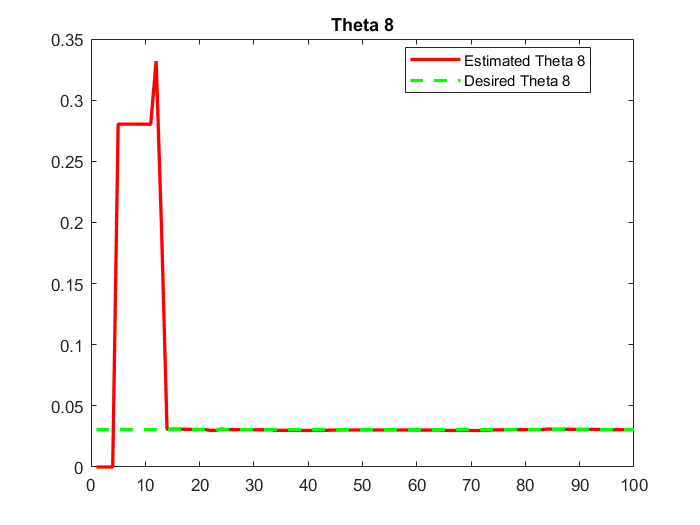












همانطور که مشاهده می‌شود مقدار پارامترها به خوبی به مقدارهای واقعی همگرا شده‌اند اگر هم جایی خطای بسیار کمی وجود دارد ناشی از وجود نویز است.

# پیوست 1

%%%%%%%%%%%%%%%%%%%% System Identification %%%%%%%%%%

%%%% 4-April-2021

clc

clear

close all

global mu W mc G ma RB

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

mc = 150; % spacecraft's mass(kg)

ma = 6.687e15; % 433 Eros mass (kg)

RB = 16; % reference radius of Eros(km)

i = 10.28\*pi/180; % inclination

hs = 218e6; % height from sun (km)

Rs = 696342; % sun radius(km)

ra = hs+Rs; % radius from the sun to asteroid

G = 6.6742e-20; % universal gravitational constant(km^3/kg.s^2)

ms = 1.989e30; % sun mass (kg)

mu = G\*(ma+ms); % gravitational parameter(km^3/s^2)

W = sqrt(mu/ra^3); %#rad/s

%#initial condition

x0 = [-5 0 -15]; dx0 = [0 40e-3 0];

ra0 = [-ra\*cos(i) 0 ra\*sin(i)];

int\_parameter = [1/130 1 1e-7 1e-14];

حل معادلات سیستم با استفاده از روش ode45:

%%% ode solving

options = odeset('RelTol',1e-3,'AbsTol',1e-5);

int = [x0 dx0 ra0 0 -sqrt(mu/ra) 0 int\_parameter 20\*ones(1,4) 1\*ones(1,4) ones(1,4) 10\*ones(1,4) x0 dx0];

[t,x]=ode45(@fun\_SI,[0 500],int);

%# plots

%%% estimation of spacecraft mass

figure

plot(t,1./x(:,13),t,mc\*ones(numel(t),1))

xlabel('t(s)'),ylabel('mc(kg)'),grid on

legend('estimation','desired')

%%% estimation of angular velocity

figure

plot(t,x(:,15),t,W\*ones(numel(t),1))

xlabel('t(s)'),ylabel('W(rad/s)'),grid on

legend('estimation','desired')

%%% relative distance

figure

plot(t,sqrt(x(:,1).^2+x(:,2).^2+x(:,3).^2)),hold on

plot(t,sqrt(x(:,33).^2+x(:,34).^2+x(:,35).^2))

xlabel('t(s)'),ylabel('R(km)'),grid on

legend('estimation','desired')

%%% relative velocity

figure

plot(t,sqrt(x(:,4).^2+x(:,5).^2+x(:,6).^2)),hold on

plot(t,sqrt(x(:,36).^2+x(:,37).^2+x(:,38).^2))

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 2-order system

for i = 1:numel(t)

[U(:,i)] = gravitational\_potential(t(i),x(i,:));

end

delta = 1e-5\*sin(0.2\*pi.\*t);

figure

plot(t,sqrt((U(1,:)+delta').^2+(U(2,:)+delta').^2+(U(3,:)+delta').^2)),hold on

f=fit(t,U(1,:)'+delta,'poly2');

f=fit(t,U(2,:)'+delta,'poly2');

f=fit(t,U(3,:)'+delta,'poly2');

W12 = (7.253e-11 \*t.^2 -1.067e-07 .\*t -5.709e-06);

W22 = (1.51e-12 \*t.^2 -1.66e-09 .\*t + 6.81e-07 ) ;

W32 = ( 2.151e-10\*t.^2 -3.006e-07.\*t -1.894e-05 ) ;

plot(t,sqrt(W12.^2+W22.^2+W32.^2)),grid on

xlabel('t(s)'),legend('$ U+\delta $','$\hat{U}+\hat{\delta}$','Interpreter','latex')

# پیوست ۲

function dx = fun\_SI(t,x)

dx = zeros(38,1);

global mu W mc

فراخوانی تابع پتانسیل:

[U] = gravitational\_potential(t,x);

طراحی کنترل کننده لغزشی:

%%% sliding mode control

landa = diag([.01 .01 .01]);k = [100 100 100]; epsilon = 0.8;

e = x(1:3);de = x(4:6);s = de+landa\*e;

Fx = mc\*(-2\*W\*x(6)-U(1)-landa(1)\*x(4))-k(1)\*satlins(s(1)/epsilon);

Fy = mc\*(W^2\*x(2)-U(2)-landa(2)\*x(5))-k(2)\*satlins(s(2)/epsilon);

Fz = mc\*(2\*W\*x(4)-3\*W^2\*x(3)-U(3)-landa(3)\*x(6))-k(3)\*satlins(s(3)/epsilon);

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

شبیه­سازی سیستم:

delta = 1e-5\*sin(0.2\*pi\*t); % disturbance

d2x = 2\*W\*x(6)+(Fx/mc+U(1)+delta);

d2y = -W^2\*x(2)+(Fy/mc+U(2)+delta);

d2z = -2\*W\*x(4)+3\*W^2\*x(3)+Fz/mc+U(3)+delta;

dx(1)=x(4); %x(1)=x

dx(2)=x(5); %x(2)=y

dx(3)=x(6); %x(3)=z

dx(4)=d2x; %x(4)=xdot

dx(5)=d2y; %x(5)=ydot

dx(6)=d2z; %x(6)=zdot

حرکت مداری سیارک(این قسمت جزء پروژه نیست.)

%%%%%%% asteroid's motion relative to sun

ra = sqrt(x(7)^2+x(8)^2+x(9)^2);

d2xa = -mu\*x(7)/(ra^3);

d2ya = -mu\*x(8)/(ra^3);

d2za = -mu\*x(9)/(ra^3);

dx(7) = x(10);

dx(8) = x(11);

dx(9) = x(12);

dx(10) = d2xa;

dx(11) = d2ya;

dx(12) = d2za;

%%%%%%%%%%%%%%%%%%%%

تشکیل ماتریس رگرسیو:

W11 = Fx;

W21 = Fy;

W31 = Fz;

%%%% gravitational potential fitting with a 2-order system

W12 = (7.253e-11 \*t^2 -1.067e-07 \*t -5.709e-06);

W22 = (1.51e-12 \*t^2 -1.66e-09 \*t + 6.81e-07 ) ;

W32 = ( 2.151e-10\*t^2 -3.006e-07\*t -1.894e-05 ) ;

W13 = 2\*x(6);

W23 = 0;

W33 = -2\*x(4);

W14 = 0;

W24 = -x(2);

W34 = 3\*x(3);

We = [W11 W12 W13 W14;W21 W22 W23 W24;W31 W32 W33 W34]; % signal vector

% phi = [1/mc 1 W W^2]'; % system's parameters

پیاده­سازی روش مبتنی بر خطای پیش بینی و تخمین پارامترها:

phi\_hat = x(13:16);

tau\_tilda = [d2x;d2y;d2z]-We\*phi\_hat;

P = [x(17) x(21) x(25) x(29);...

x(18) x(22) x(26) x(30);...

x(19) x(23) x(27) x(31);...

x(20) x(24) x(28) x(32)];% Kalman Filter matrix

landa0 = diag([.3 .01 .1 .1]); k0 = 1;

la = landa0\*(1-norm(P)/k0); %#forgetting factor

dp = la\*P-P\*(We')\*We\*P;

dphi\_hat = P\*We'\*tau\_tilda;

dx(13:16) = dphi\_hat;

dx(17:32) = dp;

شبیه سازی سیستم شناسایی شده برای مقایسه با سیستم واقعی:

%%%%%%%%%%%%%%%%%%%%%%%%% Estimation of system

landa = diag([.01 .01 .01]);k = [10 10 10]; epsilon = 0.8;

e = x(33:35);de = x(36:38);s = de+landa\*e;

Fx = (1/x(13))\*(-2\*x(15)\*x(36)-W12-landa(1)\*x(36))-k(1)\*satlins(s(1)/epsilon);

Fy = (1/x(13))\*(x(15)^2\*x(34)-W22-landa(2)\*x(37))-k(2)\*satlins(s(2)/epsilon);

Fz = (1/x(13))\*(2\*x(15)\*x(36)-3\*x(15)^2\*x(35)-W32-landa(3)\*x(38))-k(3)\*satlins(s(3)/epsilon);

d2x\_hat = 2\*x(15)\*x(38)+(Fx\*x(13)+W12);

d2y\_hat = -x(15)^2\*x(34)+(Fy\*x(13)+W22);

d2z\_hat = -2\*x(15)\*x(36)+3\*x(15)^2\*x(35)+Fz\*x(13)+W32;

dx(33)=x(36); %x(1)=x

dx(34)=x(37); %x(2)=y

dx(35)=x(38); %x(3)=z

dx(36)=d2x\_hat; %x(4)=xdot

dx(37)=d2y\_hat; %x(5)=ydot

dx(38)=d2z\_hat; %x(6)=zdot

end

# پیوست ۳

تعریف تابع پتانسیل گرانشی:

function [U] = gravitational\_potential(~,x)

global RB G ma

r = sqrt(x(1)^2+x(2)^2+x(3)^2)+RB;

phi = atan2(sqrt(x(1)^2+x(2)^2),x(3)); %# latitude angle

landa = atan2(x(2),x(1)); %# Longitudinal angle

ضرایب لژاندر:

%# Legendre coefficients

P00 = 1; P10 = sin(phi); P11 = cos(phi);

P20 = 0.5\*(3\*sin(phi)^2-1); P21 = 3\*sin(phi)\*cos(phi); P22 = 3\*cos(phi)^2\*sin(phi);

P30 = 0.5\*(5\*sin(phi)^3-3\*sin(phi)); P31 = 0.5\*cos(phi)\*(15\*sin(phi)^2-3); P32 = 15\*cos(phi)^2\*sin(phi);

P\_sinphi = [P00 P10 P11;P20 P21 P22;P30 P31 P32];

ضرایب استوکس:

%# Stokes' coefficients

C = [0 0.001175 -0.000348;-0.052851 0.000102 0.083203; -0.001747 0.004083 0.002129];

S = [0 0 0.000088;0 0.000012 -0.028023; 0 0.003404 -0.000836];

U\_bar = zeros(1,1);

%# gravitational potential function

for n = 2:3

for m = 1: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)\*(1+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=1000;

T=0.1;

Mc=150;

R=16e3;

Ma=6.687e15;

G=6.67e-11;

w=1.12e-7;

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

u1=5+2\*randn(1,N);

u2=5+2\*randn(1,N);

u3=5+2\*randn(1,N);

for i=1:N

u1(i)=u1(i)+0.1\*I;

u2(i)=u2(i)+0.1\*I;

u3(i)=u3(i)+0.1\*I;

end

مدلسازی نویز:

noise1=0.0001\*randn(1,N);

noise2=0.0001\*randn(1,N);

noise3=0.0001\*randn(1,N);

x1=zeros(1,N);

x2=zeros(1,N);

x3=zeros(1,N);

x4=zeros(1,N);

x5=zeros(1,N);

x6=zeros(1,N);

for i=1:N-1

x1(i+1)=x1(i)+T\*x2(i);

x2(i+1)=x2(i)+2\*T\*w\*x6(i)+(1/Mc)\*T\*u1(i)+T\*G\*Mc\*Ma/(R+x1(i))^2+noise1(i);

x3(i+1)=x3(i)+T\*x4(i);

x4(i+1)=x4(i)-T\*w^2\*x3(i)+(1/Mc)\*T\*u2(i)+T\*G\*Mc\*Ma/(R+x3(i))^2+noise2(i);

x5(i+1)=x5(i)+T\*x6(i);

x6(i+1)=x6(i)-2\*T\*w\*x2(i)+3\*T\*w^2\*x5(i)+(1/Mc)\*T\*u3(i)+T\*G\*Mc\*Ma/(R+x5(i))^2+noise3(i);

end

U=[u1' u2' u3'];

Y=[x1' x3' x5'];

# پیوست ۵

مدلسازی گرانش برای شبکه‌ی عصبی

function [U] = gravitational\_potential(~,x)

global RB G ma

r = sqrt(x(1)^2+x(2)^2+x(3)^2)+RB;

phi = atan2(sqrt(x(1)^2+x(2)^2),x(3)); %# latitude angle

landa = atan2(x(2),x(1)); %# Longitudinal angle

%# Legendre coefficients

P00 = 1; P10 = sin(phi); P11 = cos(phi);

P20 = 0.5\*(3\*sin(phi)^2-1); P21 = 3\*sin(phi)\*cos(phi); P22 = 3\*cos(phi)^2\*sin(phi);

P30 = 0.5\*(5\*sin(phi)^3-3\*sin(phi)); P31 = 0.5\*cos(phi)\*(15\*sin(phi)^2-3); P32 = 15\*cos(phi)^2\*sin(phi);

P\_sinphi = [P00 P10 P11;P20 P21 P22;P30 P31 P32];

%# Stokes' coefficients

C = [0 0.001175 -0.000348;-0.052851 0.000102 0.083203; -0.001747 0.004083 0.002129];

S = [0 0 0.000088;0 0.000012 -0.028023; 0 0.003404 -0.000836];

U\_bar = zeros(1,1);

%# gravitational potential function

for n = 2:3

for m = 1: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)\*(1+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 = 10;

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 = 70/100;

net.divideParam.valRatio = 15/100;

net.divideParam.testRatio = 15/100;

% 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{1};

valTargets = t .\* tr.valMask{1};

testTargets = t .\* tr.testMask{1};

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

[1] AlandiHallaj, M., & Assadian, N. (2017). Soft landing on an irregular shape asteroid using multiple-horizon multiple-model predictive control. *Acta Astronautica,* *140*, 225-234. doi:10.1016/j.actaastro.2017.08.019

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

**ممنون از تدریس خوبتان.**

**با احترام**

**مهسا آزادمنش**

**شماره دانشجویی ۹۹۰۰۴۲۶**