



به نام خدا

استاد درس: دکتر آیتی

شبیه‌سازی اول درس کنترل تطبیقی

دانشجو: یاسین ریاضی

۸۱۰۶۰۰۲۰۳

۱۴۰۱ بهار

فهرست

1	1 شناسایی offline سیستم
1	1.1 استخراج معادلات حاکم بر سیستم
3	1.2 شناسایی پارامترهای سیستم به روش LS
5	1.3 اثر PE بودن سیگнал شناسایی ورودی های مختلف
10	1.3.1 شناسایی سیستم با استفاده از ورودی شبیه
12	1.4 اثر under-parameter و over-parameter (سیستم را با یک مدل مرتبه بالاتر و پایین تر شناسایی کرده و را بررسی کرده و نتایج خود را تحلیل کنید).
12	1.4.1 اثر کم در نظر گرفتن پارامترها
13	1.4.2 اثر زیاد در نظر گرفتن پارامترها
15	2 شناسایی On-line سیستم
15	2.1 شناسایی با حداقل مربعات بازگشته برخط
15	2.1.1 شناسایی با ورودی پالس
18	2.1.2 ورودی پله
21	2.1.3 ورودی سینوسی
24	2.1.4 ورودی شبیه
26	2.2 اثر PE بودن سیگنال های انتخابی در قسمت قبل را در شناسایی سیستم
26	2.3 اثر شرایط اولیه پارامترها و شرایط اولیه ماتریس کوواریانس
26	2.3.1 شرایط اولیه ماتریس کوواریانس
28	2.3.2 شرایط اولیه پارامترها
29	2.4 اثر مرتبه مدل انتخاب شده را در حالت underparameterize و overparameterize
29	2.4.1 underparameterize
29	2.4.2 overparameterize
31	2.5 اثر نویزهای سفید و رنگی را بر دقت شناسایی
31	2.5.1 نویز سفید
32	2.5.2 نویز رنگی
34	2.6 شناسایی ELS پارامترهای سیستم و چند جمله‌ای نویز نویز رنگی
35	2.6.1 آیا پارامترها به درستی همگرا شده‌اند؟

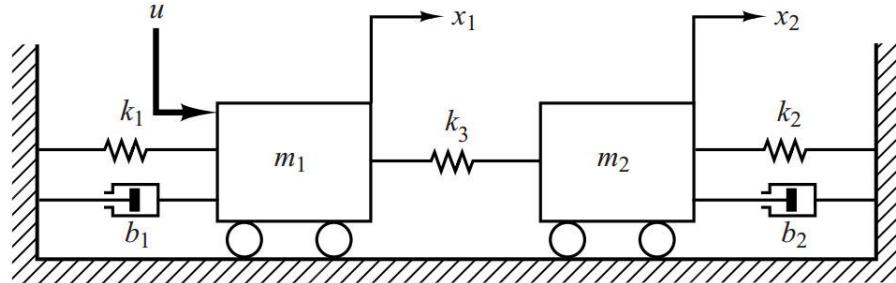
7	با تغییر پارامترهای سیستم یک بار به طور ناگهانی و یک بار به طور آرام (در حدود 10 درصد) در طول فرآیند شناسایی، از 35 مکانیزم‌های برای بهبود دقت شناسایی سیستم استفاده کنید
35 2.7.1 شناسایی با روش covariance resetting
40 2.7.2 روش Forgetting Factor
40 2.7.3 تغییرات ناگهانی
42 2.7.4 تغییرات آرام
44 RLS,LMS,PA,SA2.8
44 2.8.1 RLS شناسایی با روش
44 2.8.2 PA روش
44 2.8.3 SA روش
45 2.8.4 LMS روش
46 3 بررسی اثر فیدبک
46 3.1 با طراحی یک کنترل کننده ساده، سیستم با سه قطب و دو صفر دلخواه را تحت فیدبک قرار داده و اثر فیدبک را بر شناسایی سیستم 46 مورد نظر بررسی کنید.
47 3.1.1 شناسایی سیستم حلقه باز
48 3.1.2 شناسایی سیستم حلقه بسته
49 3.2 پارامترهای یک سیستم ناپایدار مرتبه 3 را شناسایی کنید
51 4 شناسایی با فیلتر کالمن
51 4.1 با استفاده از فیلتر کالمن پارامترهای سیستم را شناسایی کنید و نتایج را تحلیل کنید
51 4.1.1 سیستم حلقه باز
55 4.1.2 سیستم حلقه بسته با کنترل PI
56 4.2 نتایج قسمت (1-4) را با RLS مقایسه کنید
57 4.3 در پارامتر تغییرات آرام ایجاد نمایید به طوری که سیستم ناپایدار نشود. با استفاده از فیلتر کالمن پارامترها را شناسایی کنید.
60 5 شناسایی سیستم غیر خطی
64 6 پیوست 1 : کدهای متلب
64 6.1 حداقل مربعات
64 6.2 حداقل مربعات بازگشتی
64 6.3 فیلتر کالمن

شکل 1-1	نمای سوال 1
شکل 1-2	پاسخ پله برای سیستم پیوسته با رنگ آبی پیوسته و سیستم دیجیتال با رنگ آبی
شکل 1-3	: ورودی پالس برای شناسایی سیستم خارج خط
شکل 1-4	: سیستم شناسایی شده برای ورودی پالس
شکل 1-5	: ورودی پله
شکل 1-6	: نمودار بد برای سیستم و مدل شناسایی شده با ورودی پله
شکل 1-7	: ورودی سینوسی برای شناسایی سیستم
شکل 1-8	: نمودار بد برای سیستم شناسایی شده برای ورودی سینوسی
شکل 1-9	: ورودی با شب ۱.۲
شکل 1-10	: نمودار بد برای ورودی شب با ضریب ۱.۲
شکل 1-11	: نمودار بد برای حالت تخمین ۶ پارامتر
شکل 1-12	: پاسخ پله سیستم و مدل با ۶ پارامتر
شکل 1-13	: نمودار بد برای سیستم با ۴ پارامتر در صورت و ۵ پارامتر در مخرج
شکل 1-14	: ورودی پله برای سیستم و مدل شناسایی شده
شکل 2-1	سیگنال ورودی
شکل 2-2	: دیاگرام بود برای ورودی پالس
شکل 2-3	همگرایی پارامترها برای پارامترهای سیستم
شکل 2-4	مربع خط
شکل 2-5	: ورودی پله واحد
شکل 2-6	نمودار بود برای ورودی پله
شکل 2-7	دیاگرام همگرایی پارامترها
شکل 2-8	: مربع خط برای ورودی پله
شکل 2-9	: ورودی سینوسی با ۱۰ دوره تناوب کامل در زمان شبیه سازی
شکل 2-10	: دیاگرام بود برای ورودی سینوسی و با دقت مناسب تا ۱ رادیان بر ثانیه
شکل 2-11	: ورودی شب برای شناسایی سیستم
شکل 2-12	: دیاگرام بد برای ورودی شب
شکل 2-13	: مربع خط برای ورودی شب
شکل 2-14	: همگرایی پارامترها برای رورودی شب
شکل 2-15	: ورودی برای مقایسه ضرایب ماتریس P
شکل 2-16	: اثر ماتریس P
شکل 2-17	$\theta_0 = 100 \times I$
شکل 2-18	$\theta_0 = 0 \times I$
شکل 2-19	: تعداد پارامترهای صورت ۳ و تعداد پارامترهای مخرج ۳
شکل 2-20	: تعداد پارامترهای صورت ۵ و تعداد پارامترهای مخرج ۶
شکل 2-21	: تعداد پارامترهای صورت ۶ و تعداد پارامترهای مخرج ۶
شکل 2-22	: ورودی با نویز سفید

32	شکل 2-23 : دیاگرام بد برای نویز سفید.....
33	شکل 2-24 : ورودی نویز رنگی
33	شکل 2-25 : دیاگرام بود نویز رنگی
34	شکل 2-26 : ورودی ELS
34	شکل 2-27 : ورودی ELS
35	شکل 2-28 ورودی سیستم با نویز سفید
36	شکل 2-29 نغییرات آنی پارامترها
37	شکل 2-30 نمودار همگرایی پارامترهای صورت و مخرج تابع تبدیل تخمین زده شده
38	شکل 2-31 خروجی مدل و خروجی سیستم.....
39	شکل 2-32 پارامترهاس سیستم در طول زمان
39	شکل 2-33 پارامترهای شناسایی شده در طول زمان
40	شکل 2-29 نغییرات آنی پارامترها
41	شکل 2-30 نمودار همگرایی پارامترهای صورت و مخرج تابع تبدیل تخمین زده شده
42	شکل 2-31 خروجی مدل و خروجی سیستم.....
43	شکل 2-34 پارامترهاس سیستم در طول زمان
43	شکل 2-35 پارامترهای شناسایی شده با تغییرات آرام.....
47	شکل 1-3 : همگرایی پارامترهای سوال ۳.۱ برای سیستم حلقه باز در کمتر از ۲ ثانیه
48	شکل 2-3 : همگرایی پارامترها برای سیستم حقه بسته
49	شکل 3-3 : شناسایی سیستم ناپایدار.....
52	شکل 1-4 : ورودی حلقه باز برای شناسایی با فیلتر کالمن.....
53	شکل 2-4 : نمودار بود برای مدل شناسایی شده با روش فیلتر کالمن
54	شکل 3-4 : نمودار همگرایی پارامترها در فیلتر کالمن برای سیستم حلقه باز
55	شکل 4-4 : نمودار بد برای سیستم حلقه بسته
55	شکل 4-5 : همگرایی پارامترها در مدل حلقه بسته
58	شکل 4-6 : پارامترهای سیستم با تغییر آرام
58	شکل 4-7 : نمودار بد برای وضعیت‌های مختلف سیستم و مدل که بین‌ها آن‌ها شناسایی شده است.....
59	شکل 4-8 : تخمین بایاس دار پارامترها پس از تغییر پارامترها در فیلتر کالمن
61	شکل 5-1 : مقدار AIC و تعداد پارامترهای تخمین زده شده
62	شکل 5-2 مدل مرتبه 5
62	شکل 5-3 مدل مرتبه 51
63	شکل 5-4 مدل مرتبه 101

1 شناسایی offline سیستم

سیستم جرم، فنر و دمپر زیر را در نظر بگیرید.



شکل 1-1 نمای سوال 1

پارامترهای مدل را به صورت زیر در نظر بگیرید.

$$m_1 = m_2 = 0.1 \text{ kg}$$

$$k_1 = b_2 = 2k_3 = \frac{1+03}{15}$$

$$k_2 = b_1 = \frac{10+03}{100}$$

1.1 استخراج معادلات حاکم بر سیستم.

با نوشتن قانون دون نیوتون برای هر جسم معادلات دینامیک عبارت اند از

$$\begin{aligned} m_1 \ddot{x}_1 + b_1 \dot{x}_1 + (k_1 + k_3)x_1 - k_3 x_2 &= u \\ m_2 \ddot{x}_2 + b_2 \dot{x}_2 - k_3 x_1 + (k_3 + k_2)x_2 &= 0 \end{aligned}$$

$$x = \begin{bmatrix} x_1 \\ \dot{x}_1 \\ x_2 \\ \dot{x}_2 \end{bmatrix} \xrightarrow{\text{yields}} \begin{cases} \dot{x} = Ax + Bu \\ y = Cx + D \end{cases}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{(k_1 + k_3)}{m_1} & -\frac{b_1}{m_1} & \frac{k_3}{m_1} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k_3}{m_2} & 0 & -\frac{(k_3 + k_2)}{m_2} & -\frac{b_2}{m_2} \end{bmatrix}, B = \begin{bmatrix} 0 \\ u \\ 0 \\ 0 \end{bmatrix}, C^T = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

با استفاده از دستور ss2tf تابع تبدیل پیوسته سیستم محاسبه می شود.

```
clc
clear
last2digit=03
m_1=1
m_2=m_1
```

```

b_2=(1+last2digit)/15
k_1=b_2
b_1=(10+last2digit)/100
k_2=b_1
k_3=0.5*k_1
A=[0      1      0      0
 - (k_3+k_1)/m_1 - b_1/m_1 k_3/m_1      0
 0      0      0      1
 k_3/m_2      0      -(k_3+k_2)/m_2 - b_2/m_2]
B=[0 1 0 0] '
C=[0 0 1 0]
D=0
[b,a]= ss2tf(A,B,C,D)
sys=tf(b,a)

```

```

sys =

```

$$\frac{10.47}{s^4 + 24.16 s^3 + 112.7 s^2 + 701.5 s + 320.5}$$

Continuous-time transfer function.

برای محاسبه تابع تبدیل گسسته

```

fb = bandwidth(sys)
T_s=0.05*2*pi/fb
sysd = c2d(sys,T_s,'zoh')

```

```

sysd =

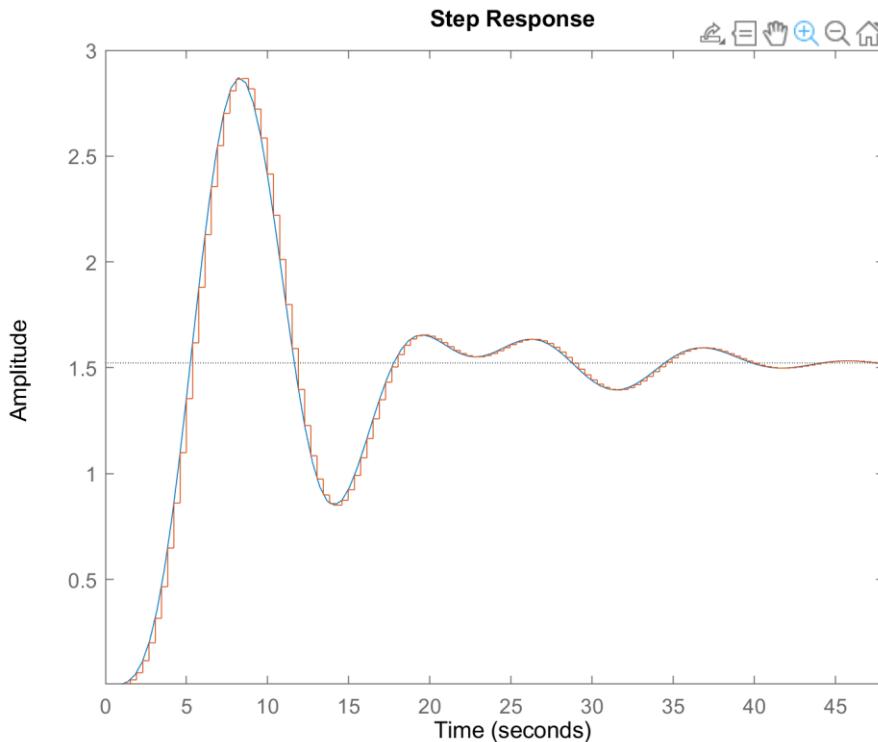
```

$$\frac{0.006452 z^3 + 0.006996 z^2 + 0.002016 z + 5.94e-06}{z^4 - 0.07735 z^3 - 0.3637 z^2 - 0.08522 z + 2.287e-07}$$

Sample time: 0.63281 seconds

Discrete-time transfer function.

پاسخ پله برای سیستم پیوسته و دیجیتال



شکل 2-1 پاسخ پله برای سیستم پیوسته با رنگ آبی پیوسته و سیستم دیجیتال با رنگ آبی

1.2 شناسایی پارامترهای سیستم به روش LS
با اجرای کد لایو اسکریپت Q1_LS.mlx شناسایی انجام می‌شود.

برای ورودی و تعداد نمونه‌ها از کد زیر استفاده شده است.

```
t = 0:T_s:tfinal;
u = gensig('sine', tfinal, tfinal, T_s);
Noise=-0.2+(0.2+0.2)*rand(numel(t),1);
u=u+Noise;
```

که شبیه از یک ضربه است و باید ورودی برای سیستم داشته باشیم نویز یا سیگنالی دیگر ورودی صفر خروجی صفر نیز خواهد داشت و محاسبات بی معنی می‌شوند.

برای شناسایی برونو خط نیاز است که خروجی سیستم را داشته باشیم برای به دست آوردن خروجی می‌توان با استفاده از خاصیت تابع تبدیل گسسته به شکل زیر خروجی را برای تابع تبدیل دلخواه استخراج کرد.

```
for temp=5:N_samples+4
    y(temp)=[-(y(temp-1:-1:temp-4)),(u(temp:-1:temp-4))]*[d(2:end),c]';
end
```

1 برای کاهش حجم کدها و پرهیز از اشتباه داده‌های اولیه درون فایل Basics.m قرار دارند که به صورت خودکار فراخوانی می‌شوند و باید در مسیر دایرکتوری متلب وجود داشته باشند.

ولی برای بیشتر شبیه‌سازی‌ها از $y = lsim(system, input, t)$ استفاده شده است.

سپس باید ماتریس رگرسور را برای تمامی سمپل تایم‌ها تشکیل داد که با استفاده از حلقه زیر صورت می‌گیرد و مرتبه مدل تخمین زده شده نیز در همین حلقه تعیین می‌شود.

این کد برای هر تعداد متغیر صورت و مخرجتابع تبدیل محاسبات را انجام می‌دهد.

```
N = numel(y) ;
Parameters_in_den=4
Parameters_in_num=4
Nv=Parameters_in_num+Parameters_in_den
phi=[];
for i=(max(Parameters_in_num,Parameters_in_den)+1):N
    phi(i,:)= [(y(i-1:-1:i-Parameters_in_den))', (u(i-1:-1:i-Parameters_in_num))'];
end
```

که در نهایت خروجی پارامترها

$\theta_{\hat{a}}$

3.4634 - 4.5497 2.6854 - 0.6010 0.0004 0.0011 - 0.0010 - 0.0003

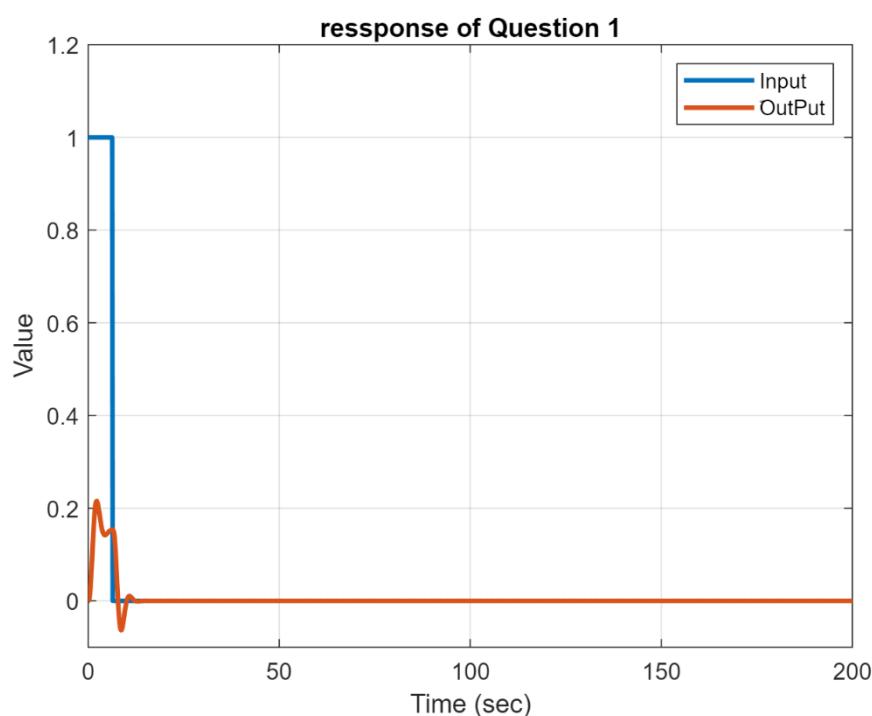
خواهد بود.

1.3 اثر PE بودن سیگنال شناسایی ورودی های مختلف

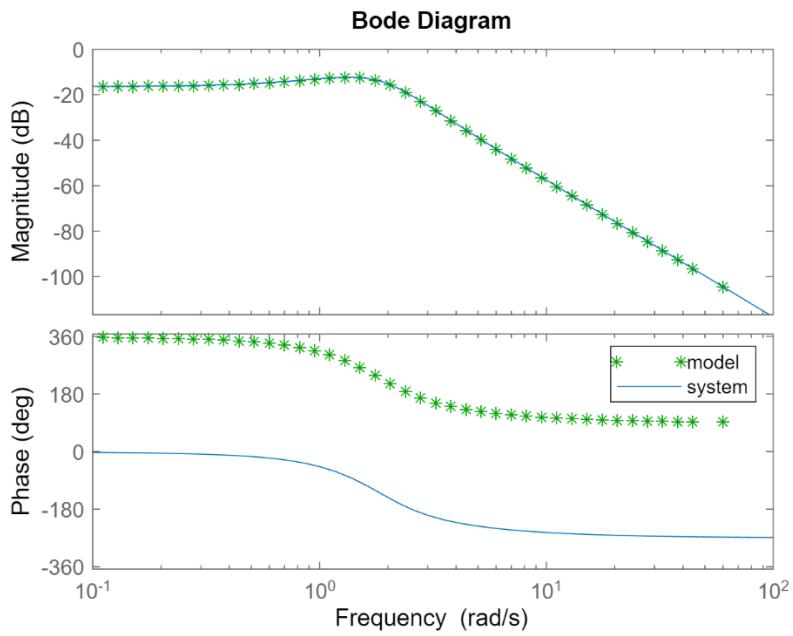
با توجه به اینکه ۸ پارامتر در سیستم داریم، برای شناسایی کامل سیستم باید سیگنال ورودی PE از مرتبه ۸ باشد. اما از آنجا که شرایط اولیه مجموعه (مقادیر توابع قبل از لحظه صفر) برابر با صفر در نظر گرفته شده است، بنابراین در شرایط خاصی (مثلاً اگر ورودی سینوسی باشد)، سیستم با یک سیگنال با PE مرتبه پایین تر هم قابل شناسایی است. اما نکته مهم این است که ای پدیده کاملاً اتفاقی بوده و در صورت تغییر در شرایط مسئله لزوماً به جواب منجر نخواهد شد. سیگنال پله یک سیگنال PE از مرتبه در LS با ورودی پله اگر فقط شرایط اولیه را در نظر بگیریم باز هم شناسایی نخواهد شد، چرا که ورودی ثابت است و باعث خواهد شد که دو ستون کاملاً برابر به دست آید.

۱-۴- شناسایی سیستم با استفاده از ورودی پالس

بر خلاف انتظار سیستم شناسایی شده است. اضافه کردن نویز در ورودی به دلیل این که مرتبه PE سیگنال را افزایش میدهد، منجر به شناسایی کامل سیستم خواهد شد. مستقل از این که شرایط اولیه در ماتریس لحاظ شده باشد یا خیر و یا این که PE سیگنال ورودی اصلی چه مقداری باشد.



شکل 1-3 : ورودی پالس برای شناسایی سیستم خارج خط



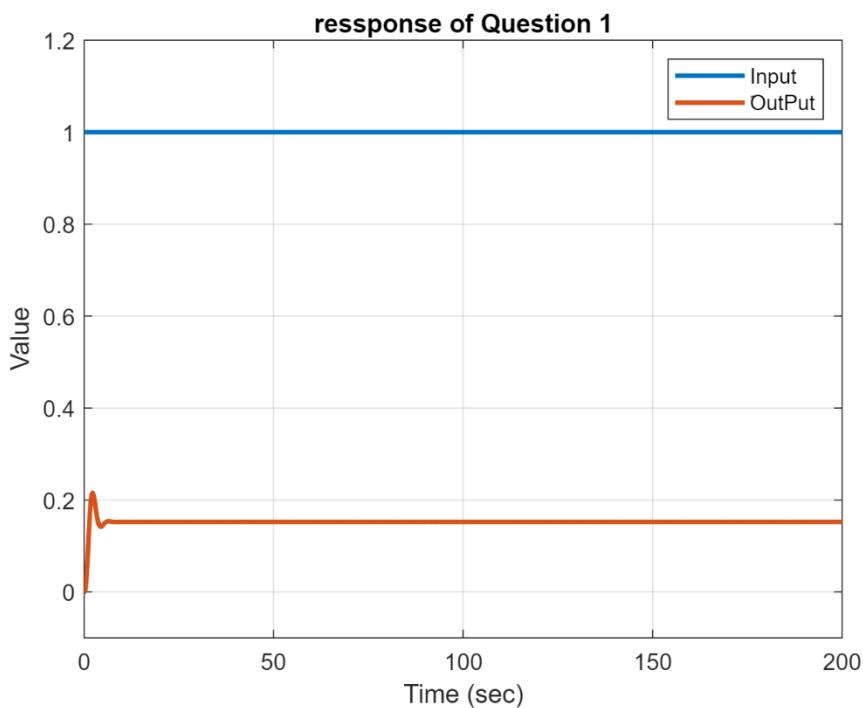
شکل ۱-۴: سیستم شناسایی شده برای ورودی پالس

	Real	Pulse
b_1	-3.4634	-3.4634
b_2	4.5497	4.5497
b_3	-2.6854	-2.6854
b_4	0.601	0.601
a_1	0.0004	0.0004
a_2	0.0011	0.0011
a_3	-0.001	-0.001
a_4	-0.0003	-0.0003

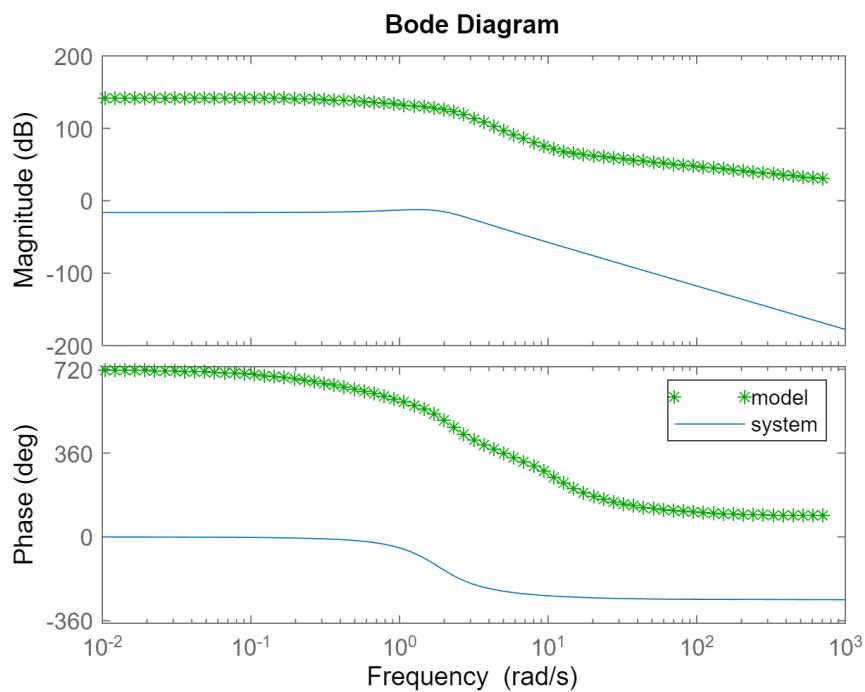
۱-۴-۲ شناسایی سیستم با استفاده از ورودی پله

سیگنال پله یک سیگنال PE از مرتبه در LS با ورودی پله حتی اگر شرایط اولیه را در نظر بگیریم باز هم شناسایی نخواهد شد، چرا که ورودی ثابت است و باعث خواهد شد که دو ستون کاملاً برابر به دست آید که مرتبه ماتریس $\phi^T \phi$ را پایین میاورد.

توجه اگر ورودی در تمامی سمپل‌ها یک نباشد شناسایی همانند پالس صورت می‌گیرد.



شکل ۱-۵: ورودی پله



شکل ۱-۶ : نمودار بد برای سیستم و مدل شناسایی شده با ورودی پله

دیاگرام بد نشان می‌دهد مدل شناسایی شده در هیچ فرکانسی رفتار سیستم را تقلید نمی‌کند.

	Real	Pulse	Step
b_1	-3.4634	-3.4634	-3.5
b_2	4.5497	4.5497	4.5
b_3	-2.6854	-2.6854	-2.7
b_4	0.601	0.601	0.6
a_1	0.0004	0.0004	183.1
a_2	0.0011	0.0011	1274.5
a_3	-0.001	-0.001	0
a_4	-0.0003	-0.0003	8192

پارامترهای مخرج به خوبی تخمین خورده‌اند ولی ۴ پارامتر صورت بسیار فاصله دارند.

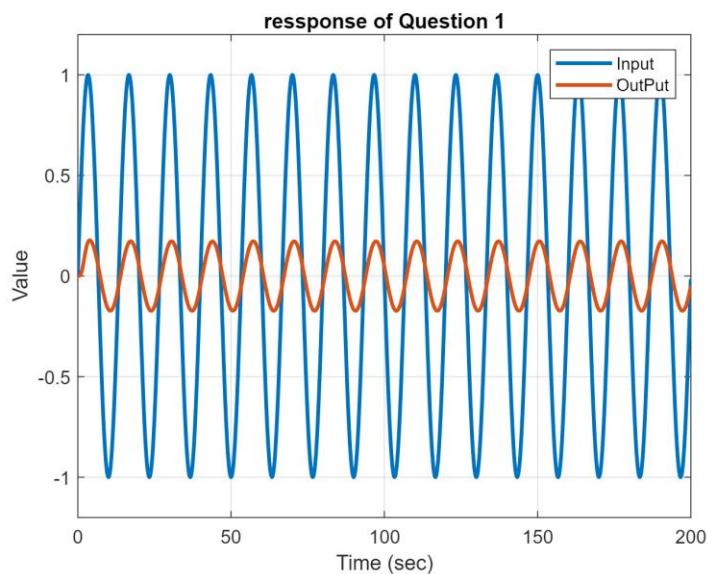
۱-۴-۳ شناسایی سیستم با استفاده از ورودی سینوسی

شناسایی سیستم با استفاده از ورودی سینوسی

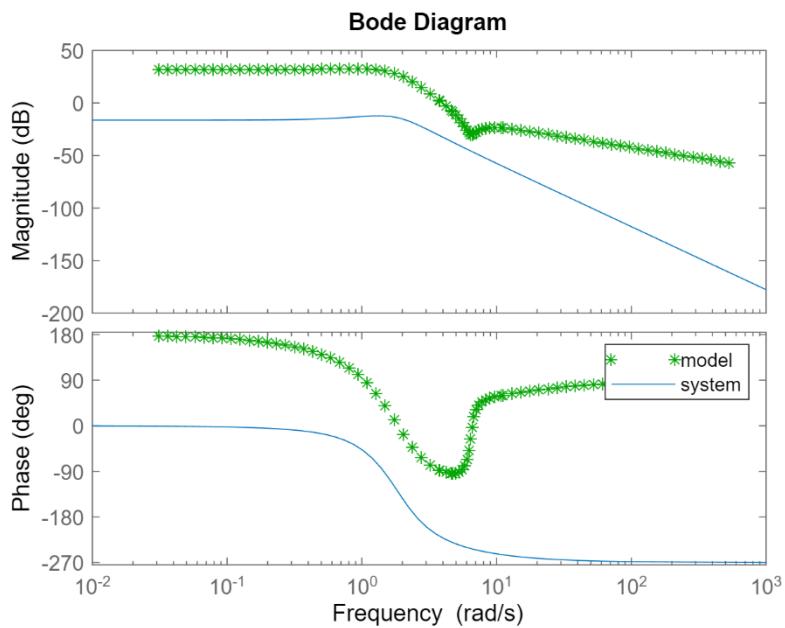
```
gensig('sine' , tfinal/15 , tfinal ,T_s)
```

ϕ و بدون اثر دادن شرایط اولیه در ماتریس.

با توجه به این که سیگنال سینوسی یک سیگنال PE از مرتبه دو است و ما ۸ پارامتر برای شناسایی داریم سیستم به صورت ناچحی شناسایی شده است.



شکل ۱-۷ : ورودی سینوسی برای شناسایی سیستم



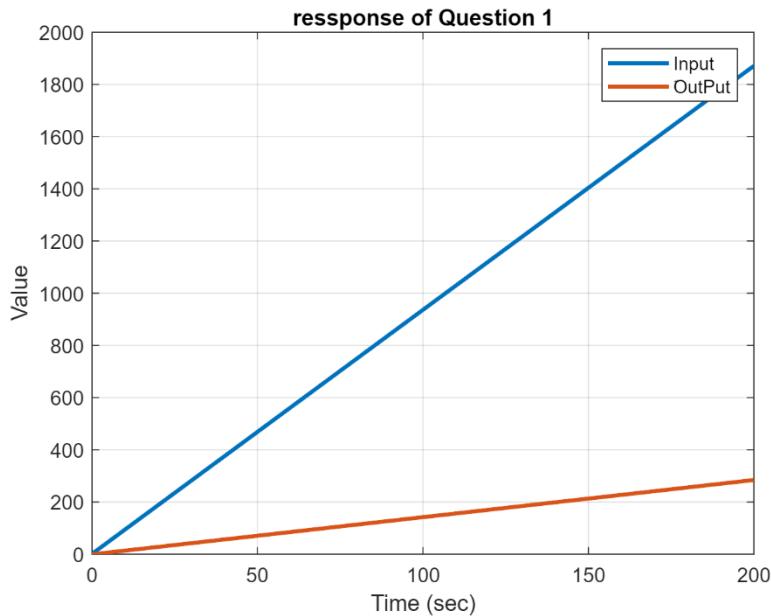
شکل 8-1: نمودار بد برای سیستم شناسایی شده برای ورودی سینوسی

همانطور که انتظار می‌رفت دیاگرام بد نسبت به ورودی پله به سیستم نزدیک تر شده است.

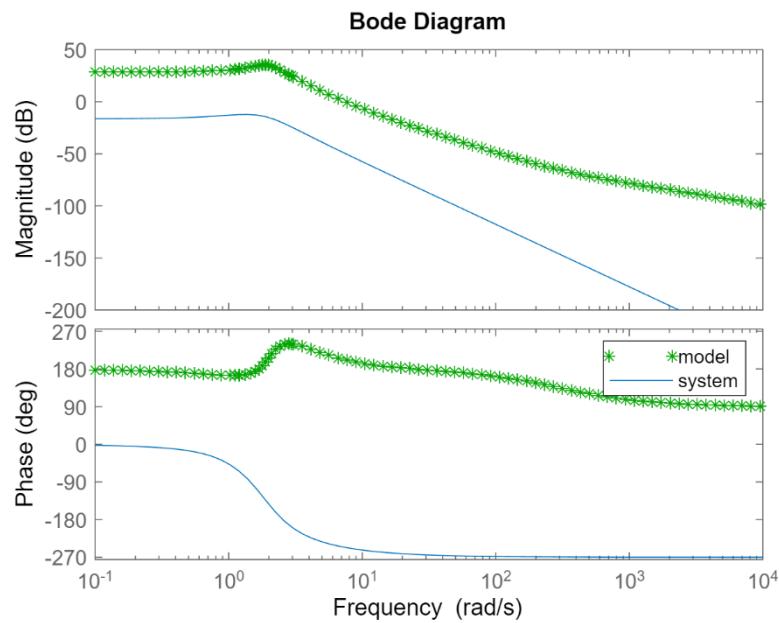
	Real	Pulse	Step	Sine
b_1	-3.4634	-3.4634	-3.5	-3.4634
b_2	4.5497	4.5497	4.5	4.5497
b_3	-2.6854	-2.6854	-2.7	-2.6854
b_4	0.601	0.601	0.6	0.601
a_1	0.0004	0.0004	183.1	-0.1562
a_2	0.0011	0.0011	1274.5	0.2383
a_3	-0.001	-0.001	0	-0.1875
a_4	-0.0003	-0.0003	8192	0.0312

شناسایی سیستم با استفاده از ورودی شبیه ۱.۳.۱

با استفاده از ورودی شبیه نتایج شبیه ورودی سینوسی می‌شوند.



شکل ۱-۹: ورودی با شبیه ۲



شکل ۱-۱۰: نمودار بد برای ورودی شبیه با ضریب ۱.۲

	Real	Pulse	Step	Sine	Ramp
b_1	-3.4634	-3.4634	-3.5	-3.4634	-3.423
b_2	4.5497	4.5497	4.5	4.5497	4.437
b_3	-2.6854	-2.6854	-2.7	-2.6854	-2.5686
b_4	0.601	0.601	0.6	0.601	0.5637
a_1	0.0004	0.0004	183.1	-0.1562	0.25
a_2	0.0011	0.0011	1274.5	0.2383	-0.3125
a_3	-0.001	-0.001	0	-0.1875	-0.375
a_4	-0.0003	-0.0003	8192	0.0312	0.1875

شناسایی با شبی ثابت مشابه حالت سینوسی است ولی دقیق شناسایی پارامتر در حالت سینوسی بیشتر است.

اثر 1.4 under-parameter و over-parameter (سیستم را با یک مدل مرتبه بالاتر و پایین‌تر شناسایی کرده و را بررسی کرده و نتایج خود را تحلیل کنید).

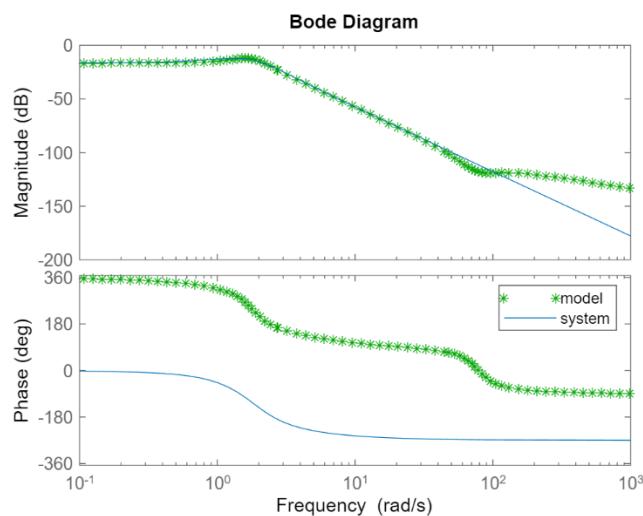
```
N = numel(y) ;
Parameters_in_den=3
Parameters_in_num=3
Nv=Parameters_in_num+Parameters_in_den
phi=[];
for i=(max(Parameters_in_num,Parameters_in_den)+1):N
    phi(i,:)= [(y(i-1:-1:i-Parameters_in_den))', (u(i-1:-1:i-Parameters_in_num))'];
end
theta_hat=((phi'*phi)^(-1))*(phi'*y)
```

کدهای شناسایی سیستم برای حالت عمومی نوشته شده‌اند و با تغییر مقدار پارامترهای زرد رنگ مراتبه مدل تغییر می‌کند.

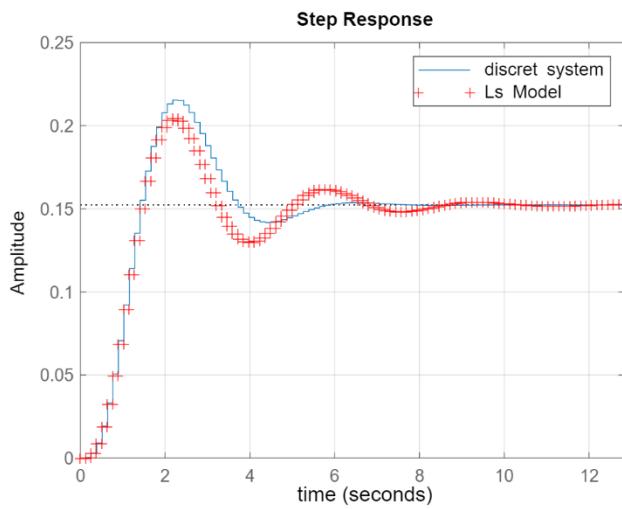
1.4.1 اثر کم در نظر گرفتن پارامترها

سیستم با استفاده از نویز و ووردی سینوسی تحریک شده است و پارامترها را شناسایی کرده است ولی به علت کم در نظر گرفتن تعداد پارامترها (۳ پارامتر در صورت و ۳ پارامتر در مخرج) مدل در فرکانس‌های بالا رفتار مناسبی را از خود نشان نمی‌دهد.

این موضوع در پاسخ پله سیستم نیز دیده می‌شود.



شکل 1-11: نمودار بد برای حالت تخمین ۶ پارامتر

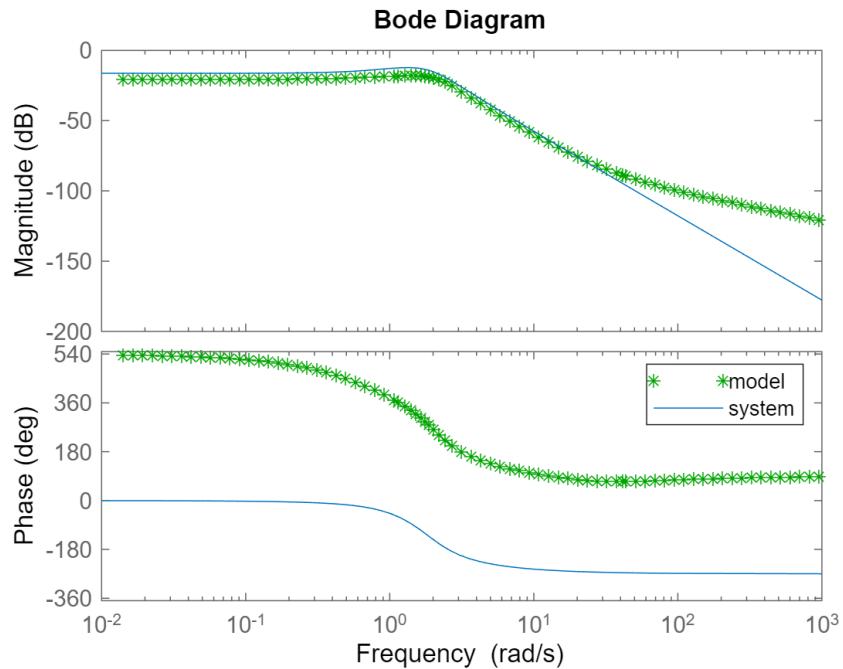


شکل ۱-۱۲: پاسخ پله سیستم و مدل با ۶ پارامتر

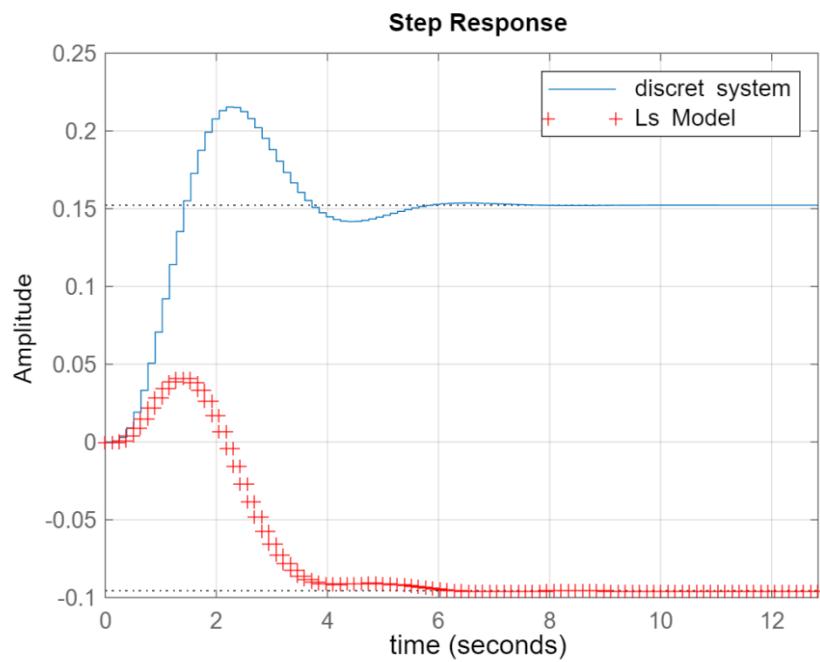
اثر زیاد در نظر گرفتن پارامترها ۱.۴.۲

برای پارامترهای بیشتر از خود سیستم تابع تبدیل ناپایدار می شود.

در این قسمت ۴ پارامتر در صورت و ۵ پارامتر در مخرج شناسایی شده اند.



شکل ۱-۱۳: نمودار بد برای سیستم با ۴ پارامتر در صورت و ۵ پارامتر در مخرج



شکل 1-14 : ورودی پله برای سیستم و مدل شناساسایی شده

2 شناسایی On-line سیستم

کدهای شناسایی برخط سیستم به شرح زیر می‌باشد.

```
N = numel(y) ;
%choose number of parameters
Parameters_in_den=4;
Parameters_in_num=4;
Nv=Parameters_in_num+Parameters_in_den;
theta(:,1:Nv) = zeros(Nv , Nv) ;
P = 1e12*eye(Nv) ;
phi=[];
for i = ( max(Parameters_in_num,Parameters_in_den)+1):N
    phi(:,i) = [(y(i-1:-1:i-Parameters_in_den))' , (u(i-1:-1:i-Parameters_in_num)) ']';
    K = P*phi(:,i)*(1+phi(:,i)']*P*phi(:,i))^(-1) ;
    P = (eye(Nv) - K*phi(:,i)')*P ;
    theta(:,i) = theta(:,i-1) + K*(y(i) - phi(:,i)']*theta(:,i-1));
end
که با تغییر
```

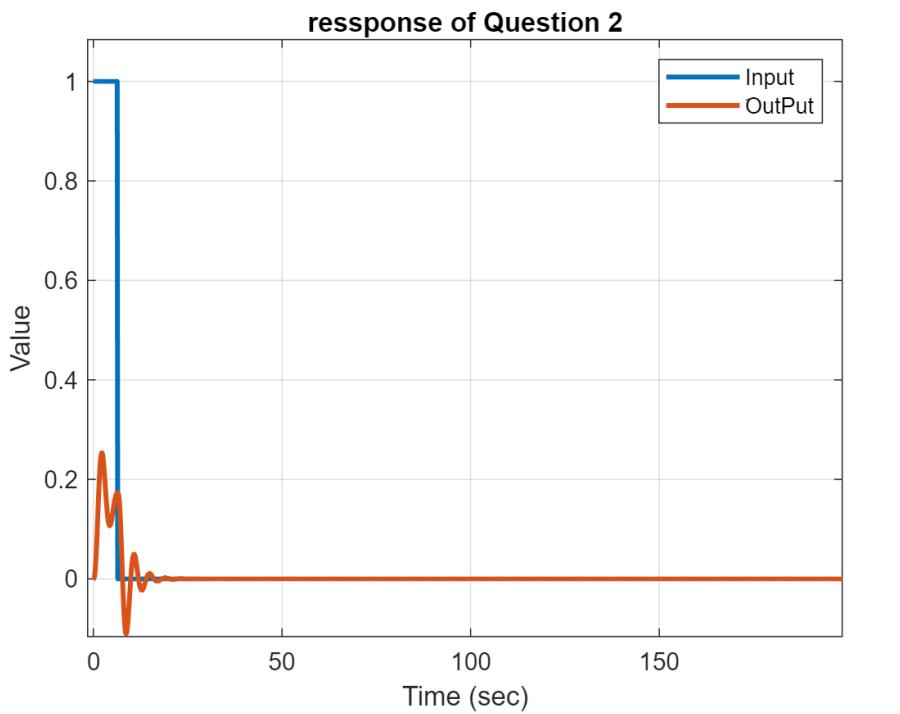
```
Parameters_in_den=4;
Parameters_in_num=4;
می‌توان مرتبه مدل را تغییر داد.
```

2.1 شناسایی با حداقل مربعات بازگشتی برخط

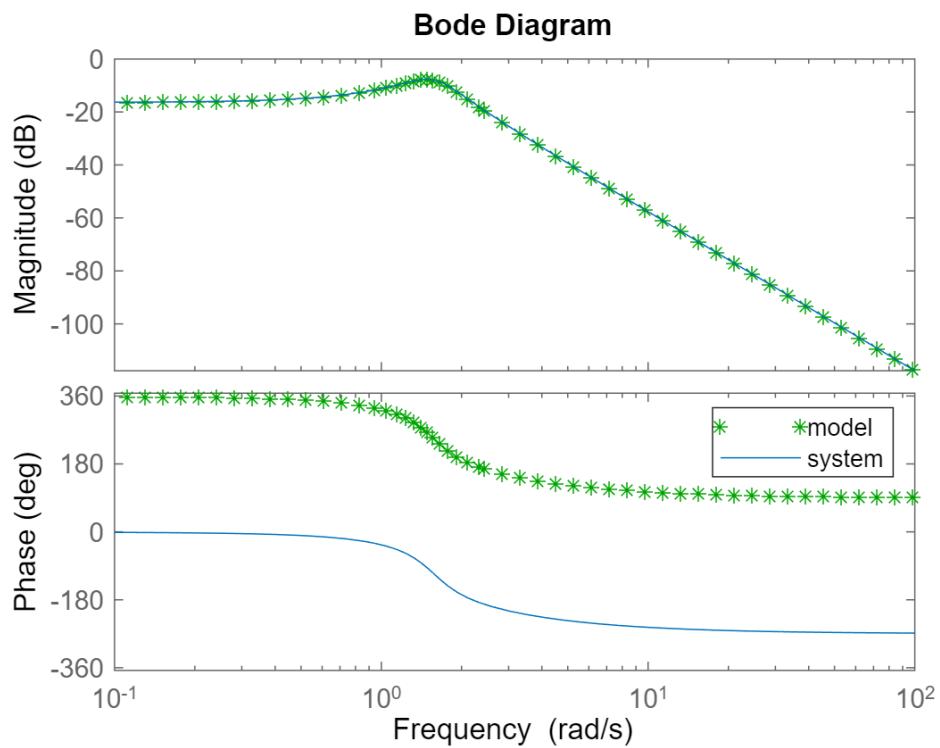
با انتخاب ورودی‌هایی از بین (پالس- پله- سینوسی و شبیب) سیستم را به صورت Online تحت شرایط ایده‌آل (با دانستن مرتبه سیستم و بدون نویز) به روش RLS شناسایی کنید.

2.1.1 شناسایی با ورودی پالس

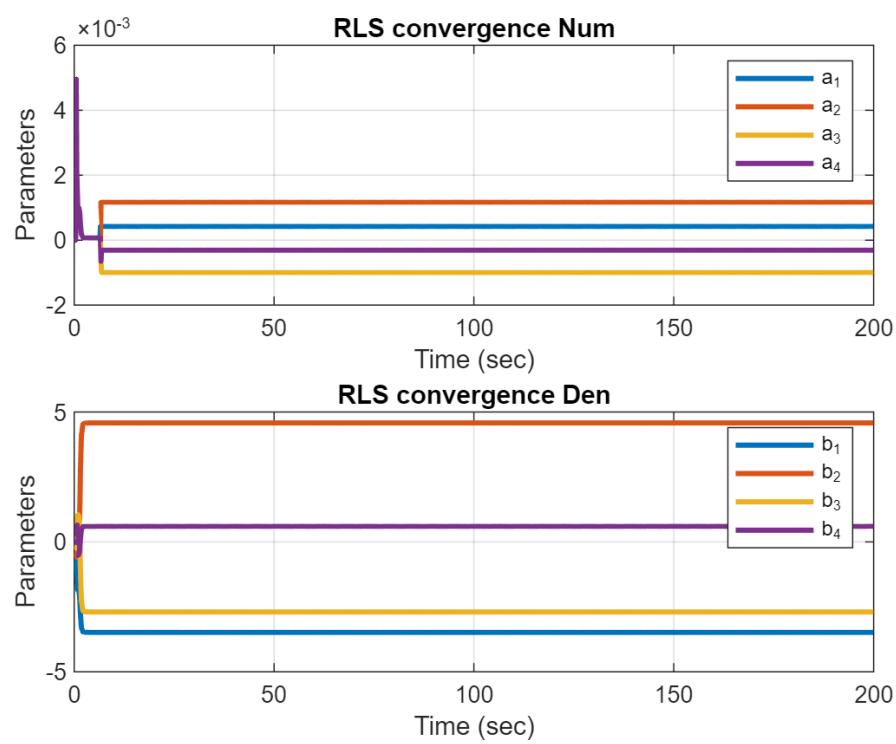
ورودی پالس همانند قسمت شناسایی حداقل مربعات پارامترها را به خوبی شناسایی کرد.



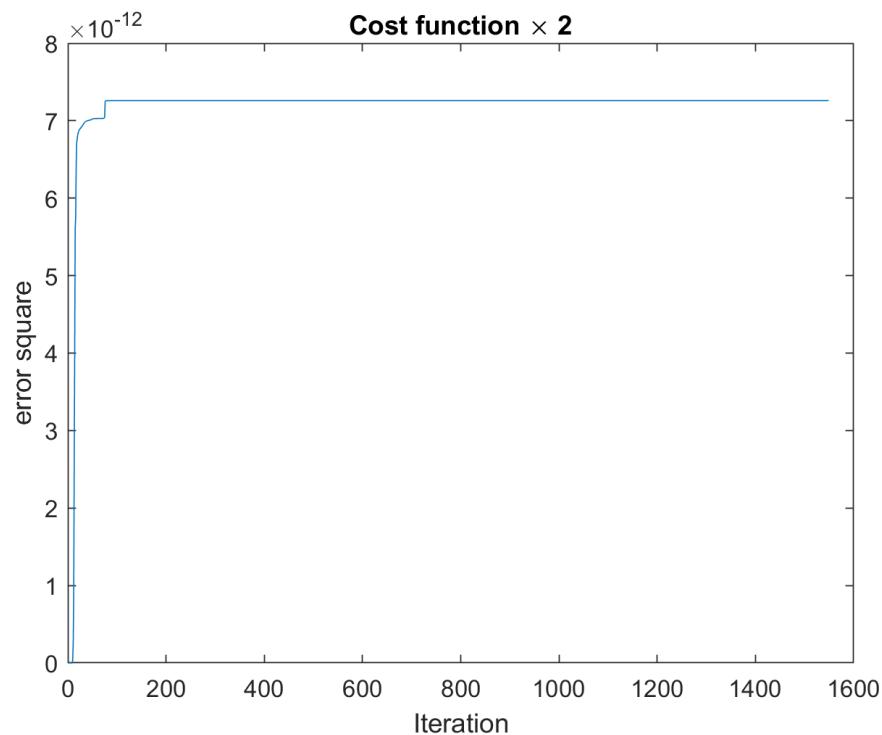
شکل ۲-۱ سیگنال ورودی



شکل ۲-۲ : دیاگرام بود برای ورودی پالس



شکل ۲-۳ همگرایی پارامترها برای پارامترهای سیستم.



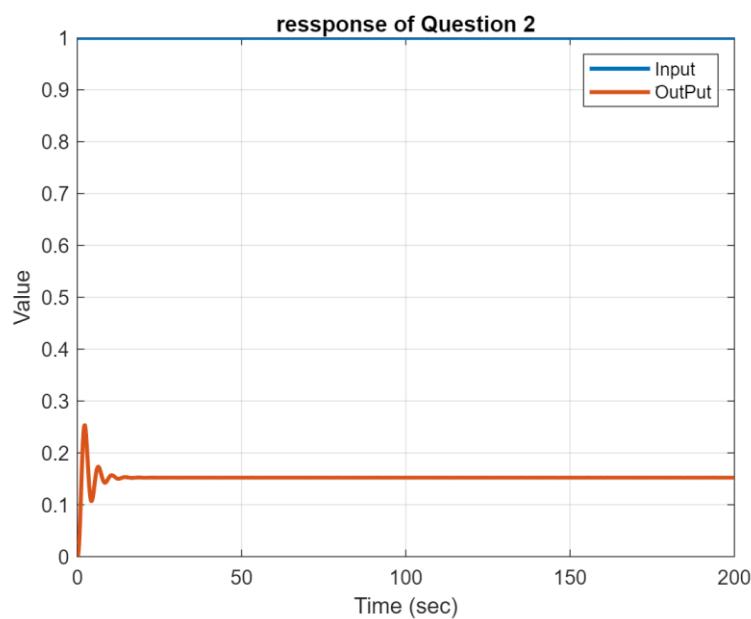
شکل ۲-۴ مربع خطأ

	Real	Pulse
b_1	-3.4634	-3.4807
b_2	4.5497	4.5801
b_3	-2.6854	-2.6967
b_4	0.601	0.5991
a_1	0.0004	0.0004
a_2	0.0011	0.0012
a_3	-0.001	-0.001
a_4	-0.0003	-0.0003

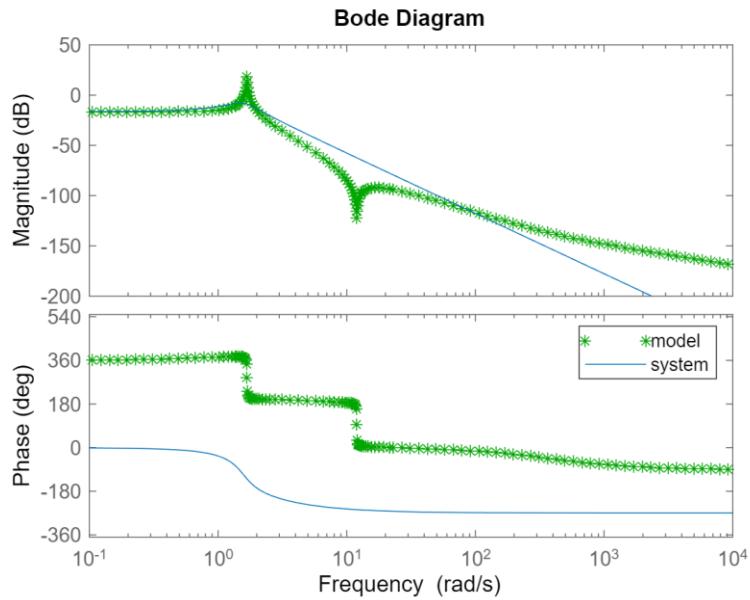
ورودی پله 2.1.2

با استفاده از پله شناسایی سریع تر انجام می شود ولی سیستم در فرکانس کمتری (حدود ۱ رادیان بر ثانیه) دقت مطلوب دارد.

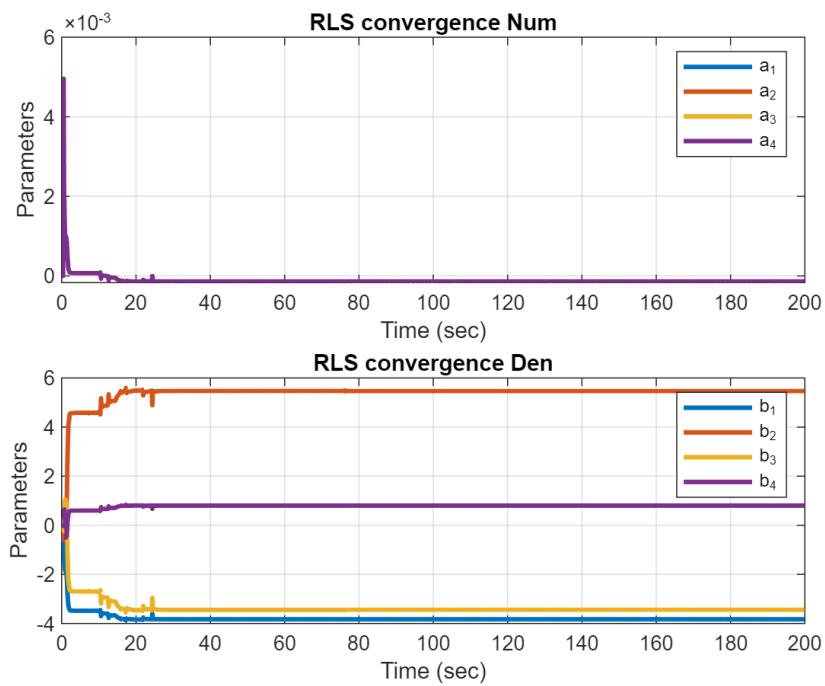
با گذشت زمان میزان خطا به مقداری ثابت همگرا می شود.



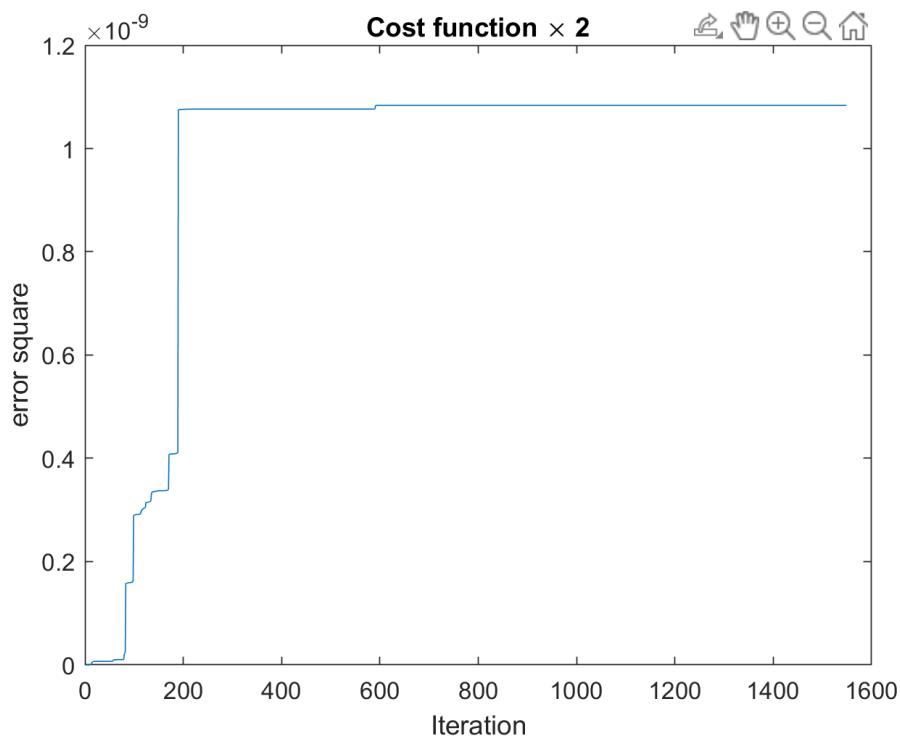
شکل 2-5 : ورودی پله واحد



شکل 2-6 نمودار بود برای ورودی پله



شکل 2-7 دیاگرام همگرایی پارامترها.

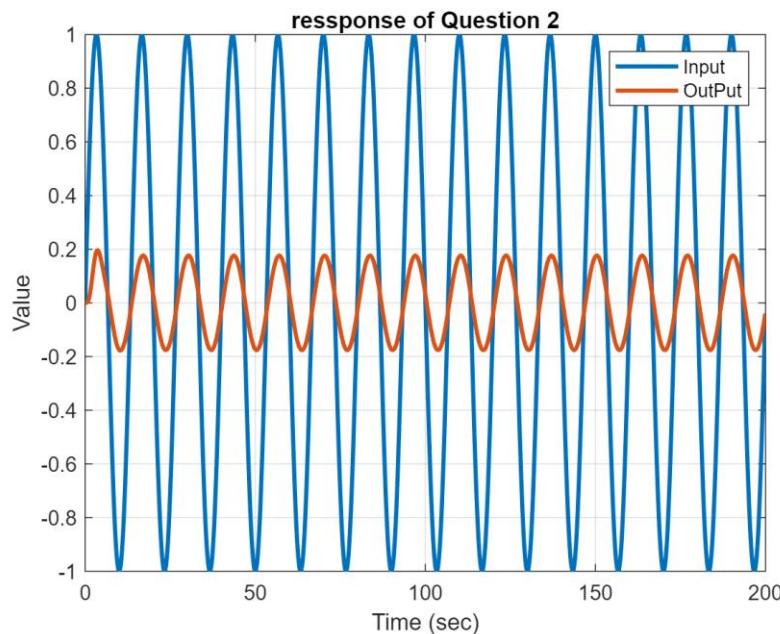


شکل 2-8 : مربع خطای ورودی پله

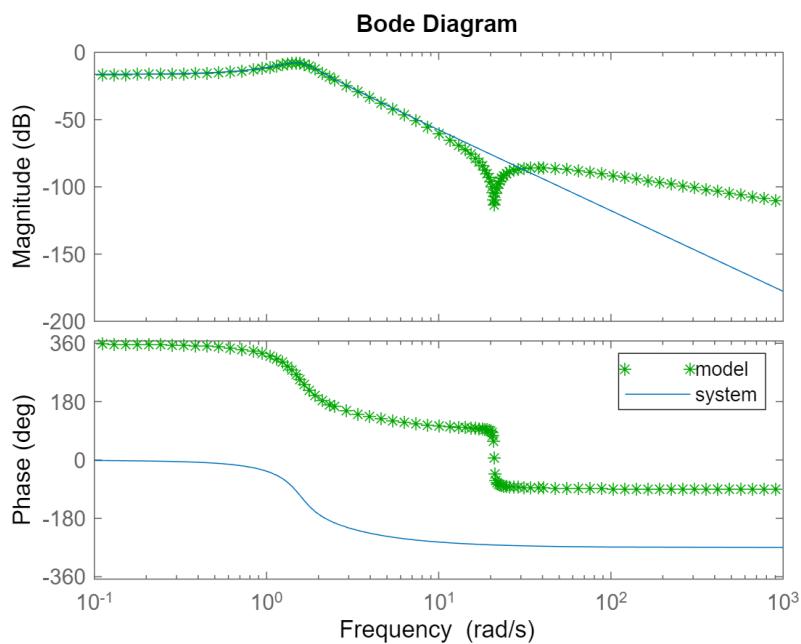
	Real	Pulse	Step
b_1	-3.4634	-3.4807	-3.8297
b_2	4.5497	4.5801	5.4663
b_3	-2.6854	-2.6967	-3.4418
b_4	0.601	0.5991	0.8015
a_1	0.0004	0.0004	-0.0001
a_2	0.0011	0.0012	-0.0001
a_3	-0.001	-0.001	-0.0001
a_4	-0.0003	-0.0003	-0.0001

پارامترهای صورت اصلاح شناسایی نشده‌اند.

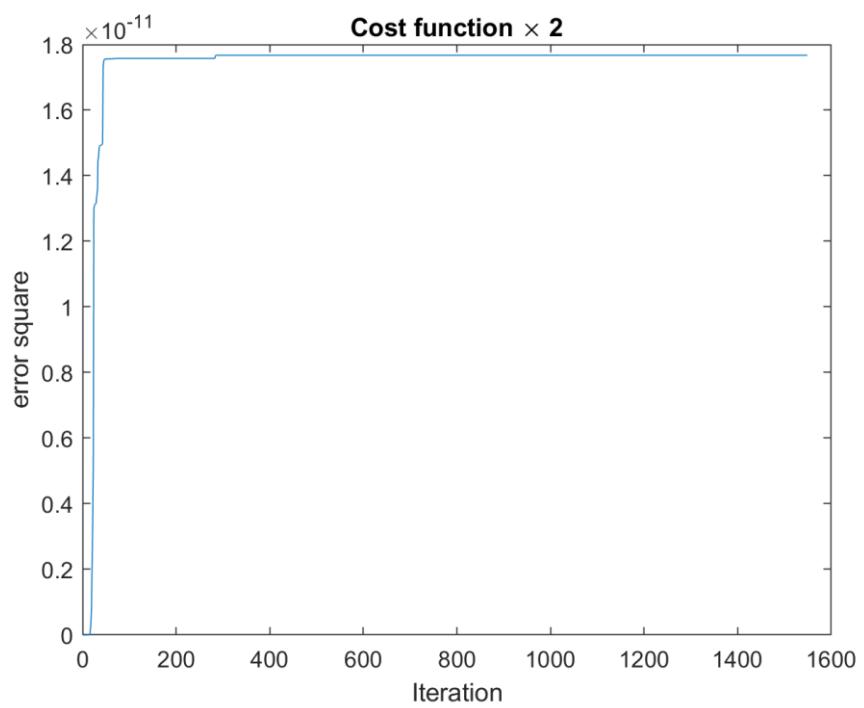
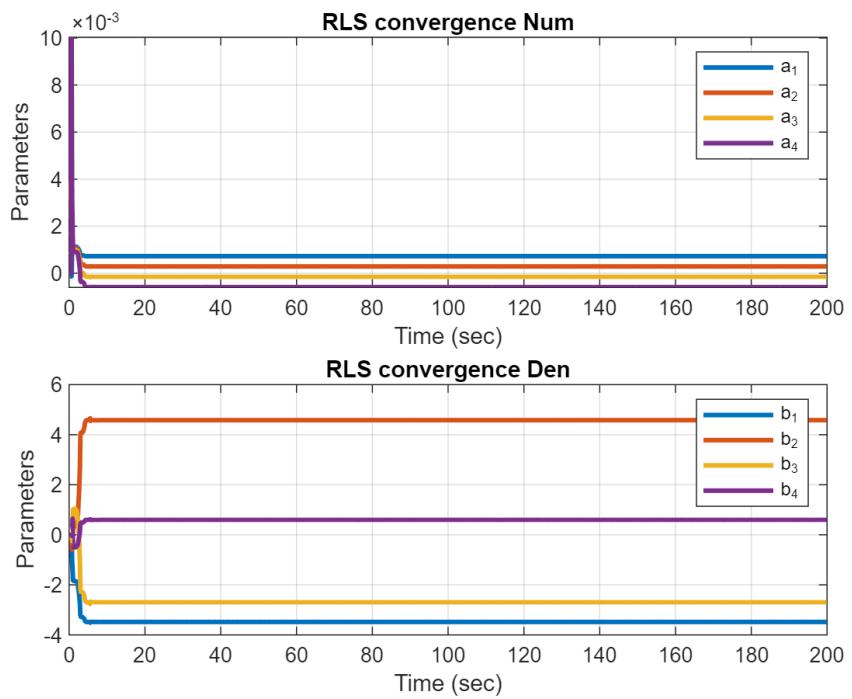
شناسایی سیستم با استفاده از سیگنال سینوسی نسبت به ورودی پله بهتر انعام شده است و نمودار بد نیز تا تطبیق تا ۱۰ رادیان بر ثانیه را نشان می‌دهد.



شکل 9-2 : ورودی سینوسی با ۱۰ دوره تناب کامل در زمان شبیه سازی



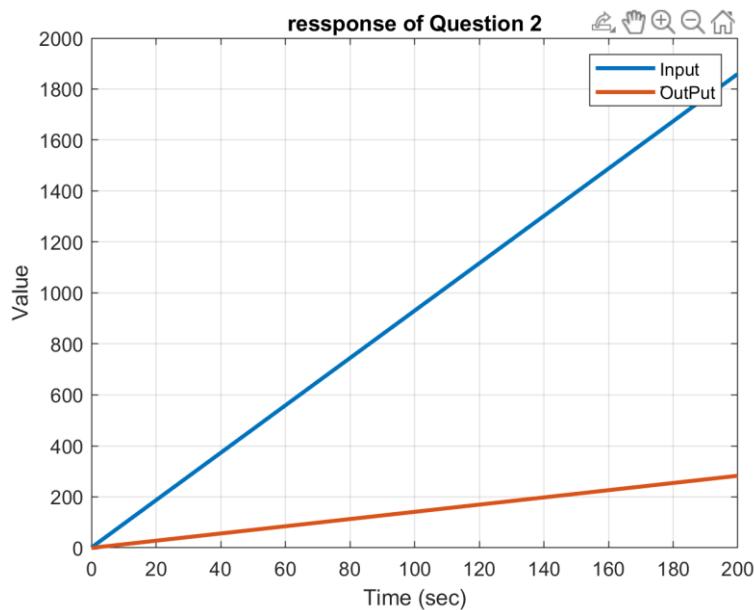
شکل 10-2 : دیاگرام بود برای ورودی سینوسی و با دقت مناسب تا ۱ رادیان بر ثانیه



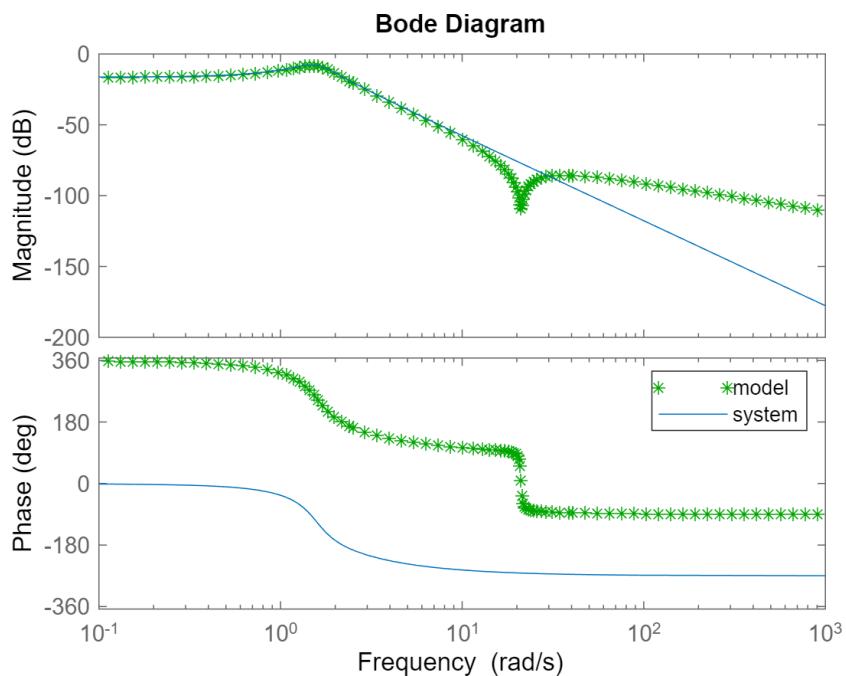
	Real	Pulse	Step	Sine
b_1	-3.4634	-3.4807	-3.8297	-3.4808
b_2	4.5497	4.5801	5.4663	4.5801
b_3	-2.6854	-2.6967	-3.4418	-2.6964
b_4	0.601	0.5991	0.8015	0.5989
a_1	0.0004	0.0004	-0.0001	0.0007
a_2	0.0011	0.0012	-0.0001	0.0003
a_3	-0.001	-0.001	-0.0001	-0.0001
a_4	-0.0003	-0.0003	-0.0001	-0.0006

سیستم در فرکانس کمتری (حدود ۱۰ رادیان بر ثانیه) دقت مطلوب دارد.

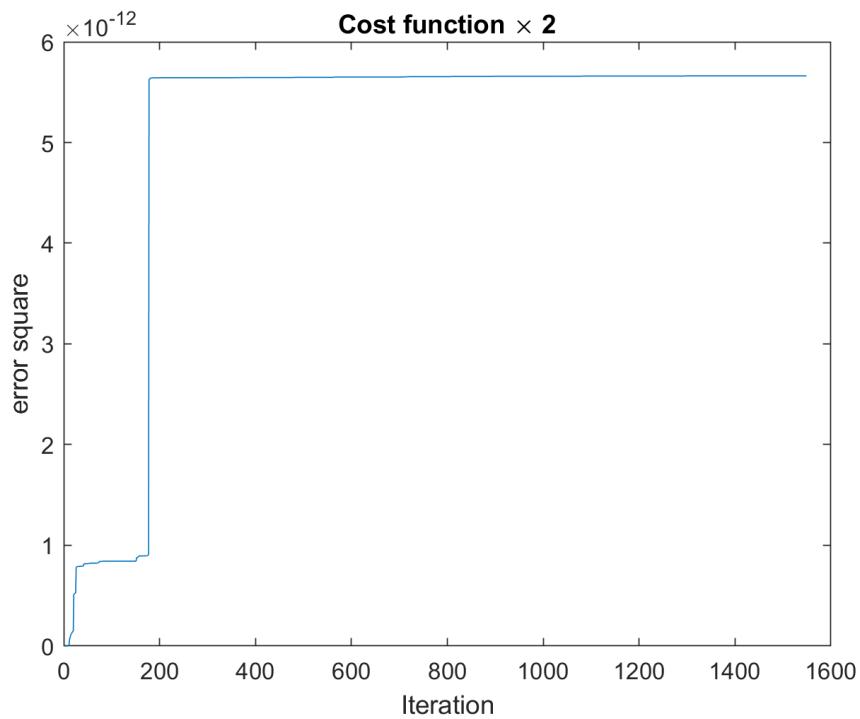
با گذشت زمان میزان خطای مقداری ثابت همگرا می‌شود.



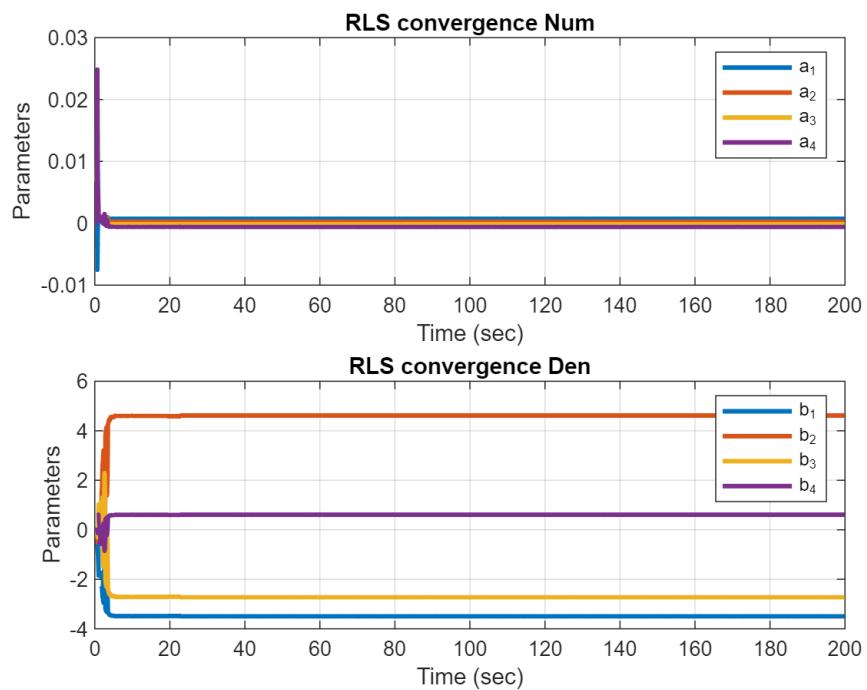
شکل 2-11 : ورودی شبیب برای شناسایی سیستم



شکل 2-12 : دیاگرام بد برای ورودی شبیب



شکل 2-13 : مریع خطای برای ورودی شبیب



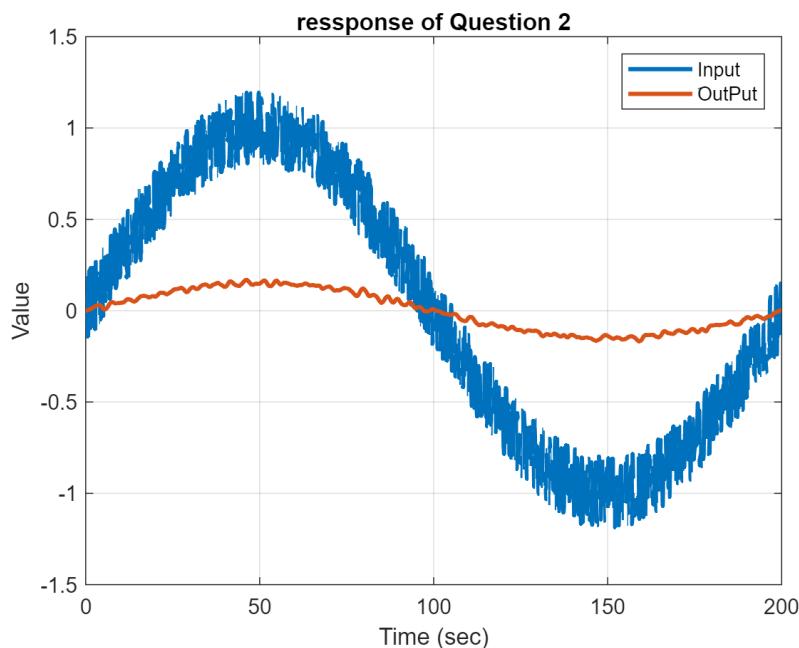
شکل 2-14 : همگرایی پارامترها برای رورودی شبیب

	Real	Pulse	Step	Sine	Ramp
b_1	-3.4634	-3.4807	-3.8297	-3.4808	-3.4948
b_2	4.5497	4.5801	5.4663	4.5801	4.6157
b_3	-2.6854	-2.6967	-3.4418	-2.6964	-2.7264
b_4	0.601	0.5991	0.8015	0.5989	0.607
a_1	0.0004	0.0004	-0.0001	0.0007	0.0007
a_2	0.0011	0.0012	-0.0001	0.0003	0.0003
a_3	-0.001	-0.001	-0.0001	-0.0001	-0.0002
a_4	-0.0003	-0.0003	-0.0001	-0.0006	-0.0006

2.2 اثر PE بودن سیگنال‌های انتخابی در قسمت قبل را در شناسایی سیستم

استفاده از ورودی‌های پالس، پله واحد، سینوسی و شب واحد برای تخمین این ۸ پارامتر مناسب نیستند چرا که مرتبه PE بودن همه آن‌ها کمتر از ۸ می‌باشند اما با این وجود ورودی‌ها متغیرهای مخرجتابع تبدیل را تا حدودی خوب تخمین می‌زنند ولی در مورد مقادیر صورت تابع تبدیل، چون‌که این مقادیر بسیار به یک دیگر نزدیک هستند و مرتبه‌ی PE بودن ورودی‌های ذکر شده به اندازه‌ای نیست که بتواند این پارامترهای نزدیک به یک دیگر را به خوبی تشخیص دهد تخمین خوبی نمی‌توانند ارایه دهنند.

2.3 اثر شرایط اولیه پارامترها و شرایط اولیه ماتریس کوواریانس



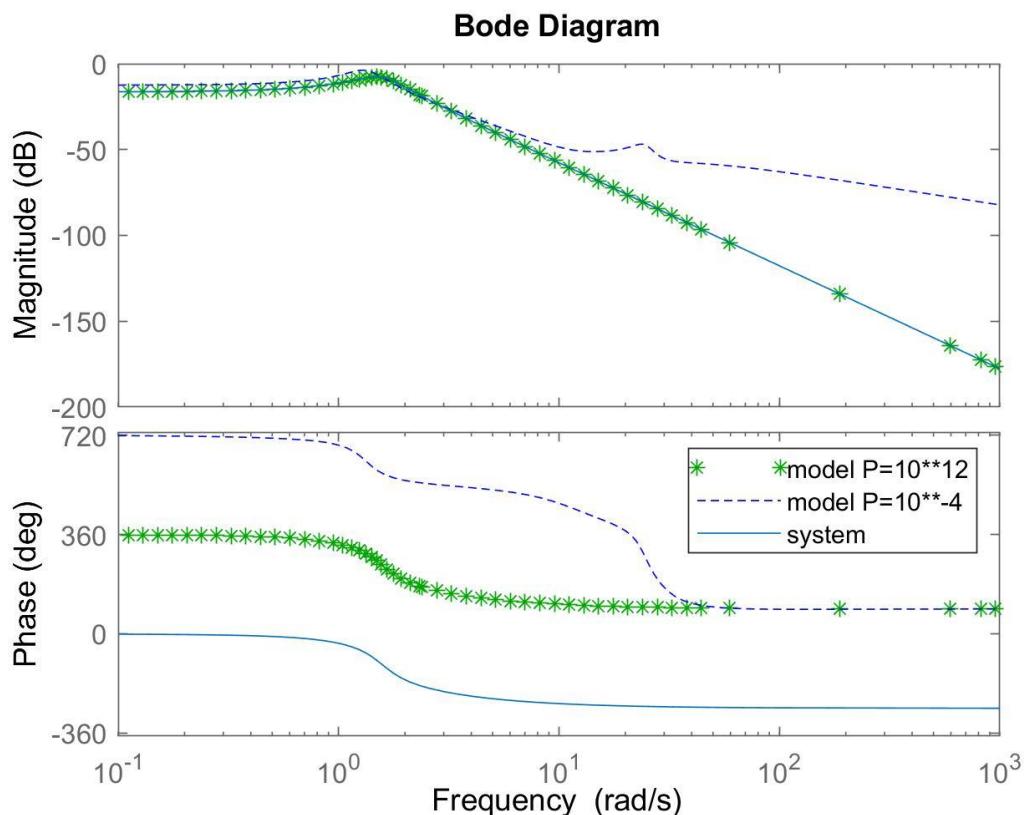
شکل 2-15 ورودی برای مقایسه ضرایب ماتریس P

شرایط اولیه ماتریس کوواریانس 2.3.1

با کاهش مقدار ماتریس P فرآیند شناسایی به شدت کند و مربع خطای بسیار بالایی دارد و حتی در صورت انتخاب ماتریس کوچک (در حدود 10^8 حتی همگرا نمی‌شود) و جواب کاملاً بی معنی می‌شود. و در رسم دیاگرام بد مجبور به افزایش مجازی درجه تابع تبدیل برای رسم ریشه‌های حقیقی منفی می‌باشیم.

در حالی که با بزرگ‌تر کردن ضریب ماتریس P سرعت و محدوده کارایی مدل تخمین زده شده تا ۱۱۰ رادیان بر ثانیه افزایش پیدا می‌کند.

در نوشتن این بخش بود که فهمیدم تمامی بخش‌های گذشته رو باید با $10^{12}I = P$ شناسایی می‌کردم نه با $I = 10^4P$ و علت شبیه نشدن نمودار بود به تابع تبدیل سیستم نیز کوچک بودن P می‌باشد.



شکل 2-16 : اثر ماتریس P

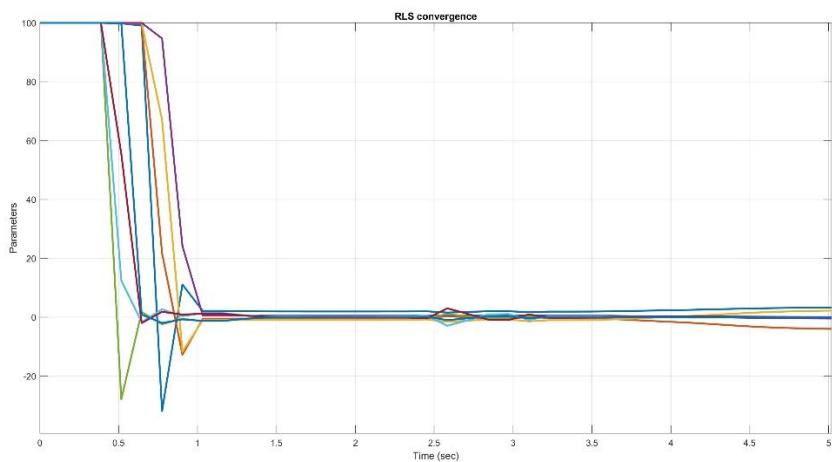
2.3.2 شرایط اولیه پارامترها

اگر بردار شرایط اولیه نزدیک به ضریب ماتریس کوواریانس انتخاب شود شکل دیاگرام بد صفر و قطب‌های اضافی نشان خواهد داد.

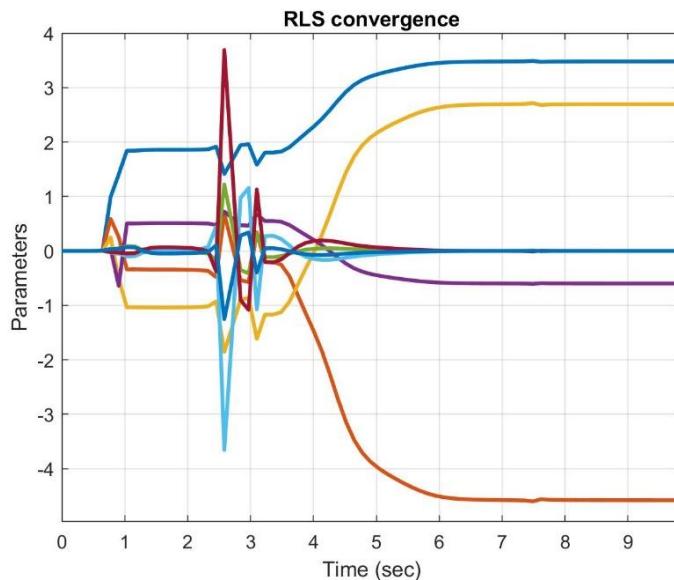
انتظار داریم که با هر بردار شرایط اولیه شناسایی سیستم همگرا شود (مقادیر ممکن فیزیکی مثلاً برای مقدار 10^5 تمامی پارامترها را صفر شناسایی می‌کند به خاطر گرد کردن‌های محاسبات عددی) و این اتفاق نیز می‌افتد.

باید در نظر داشت در چه حسابگری محاسبات شناسایی انجام می‌پذیرد چون بردار شرایط اولیه بزرگ حجم بیشتری از حافظه را به خود اختصاص می‌دهد.

ولی با بزرگ یا منفی فرض کردن بردار شرایط اولیه شناسایی پارامترها زود تر همگرا می‌شود.



شكل 2-17 $\theta_0 = 100 \times I$



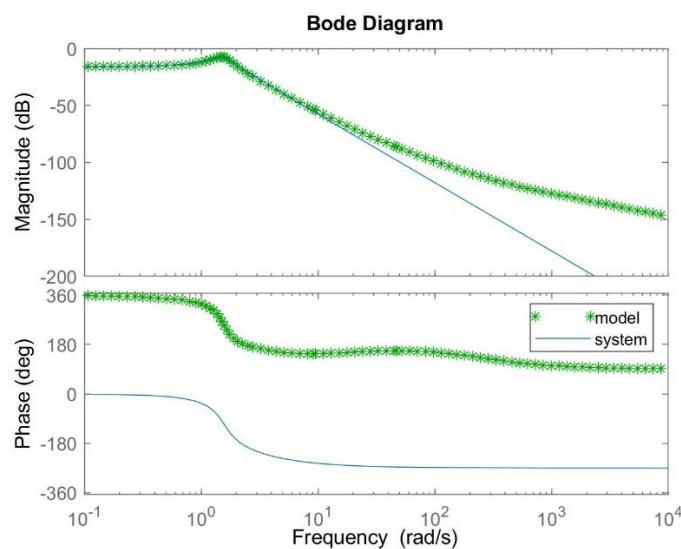
شكل 2-18 $\theta_0 = 0 \times I$

2.4 اثر مرتبه مدل انتخاب شده را در حالت underparameterize و overparameterize

underparameterize 2.4.1

در هیچ نسبتی از پارامترها نمودار بود بر روی نمودار بود سیستم منطبق نمی‌شود و در نتیجه تخمینی مناسب از سیستم را نمی‌توان انتظار داشت.

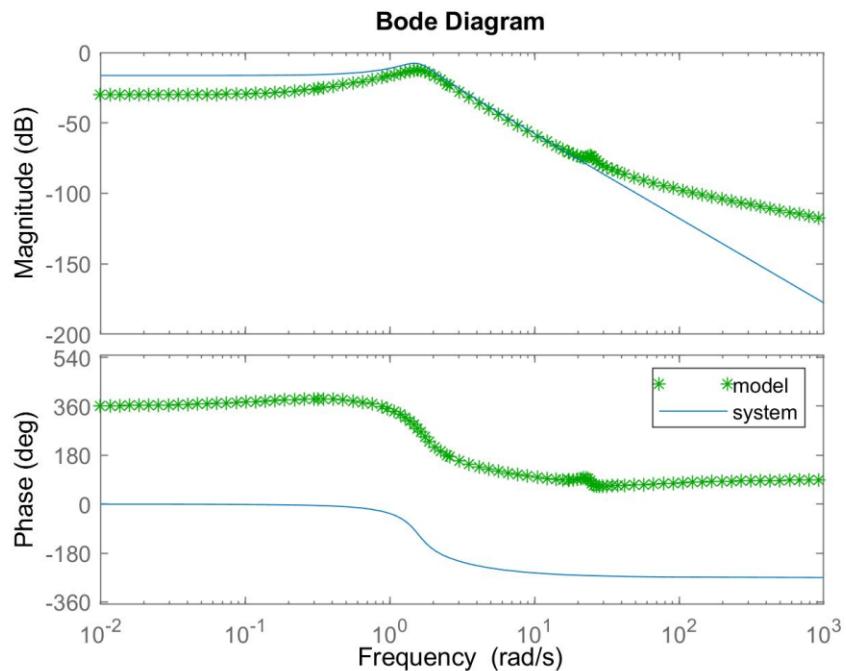
مربع خطای نیز به طور فزاینده افزایش پیدا می‌کند.



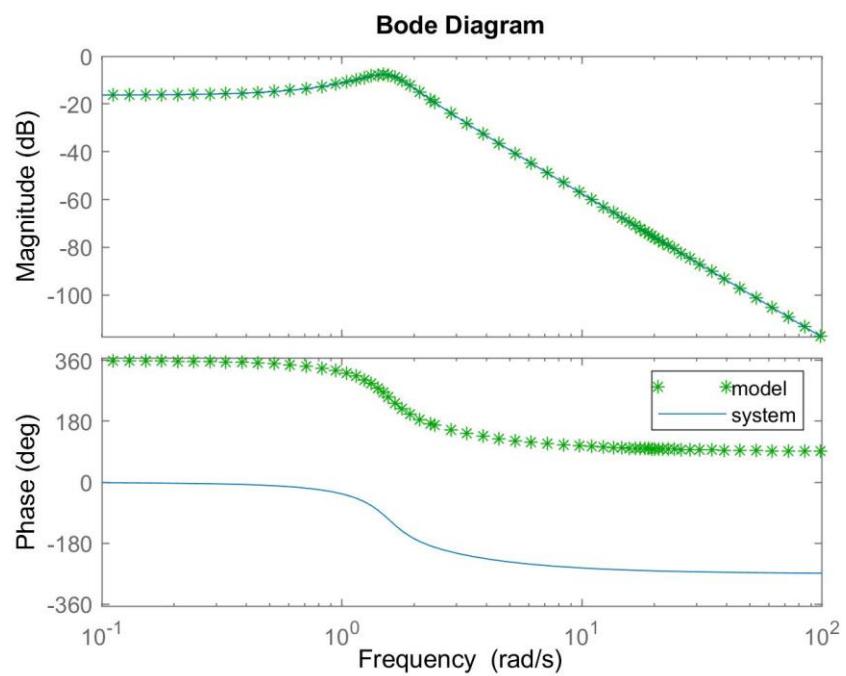
شکل 2-19 : تعداد پارامترهای صورت ۳ و تعداد پارامترهای مخرج ۳

overparameterize 2.4.2

در حالتی که نسبت پارامترهای صورت و مخرج حفظ شود سیستم ناپایدار نمی‌شود و تخمین دیاگرام بود به خوبی و بدون افزایش درجه مجازی سیستم انجام می‌شود ولی اگر نسبت تعداد پارامترها رعایت نشود افزایش درجه مجازی برای رسم ریشه‌های حقیقی منفی لازم است.



شکل 20-2 : تعداد پارامترهای صورت ۵ و تعداد پارامترهای مخرج ۶



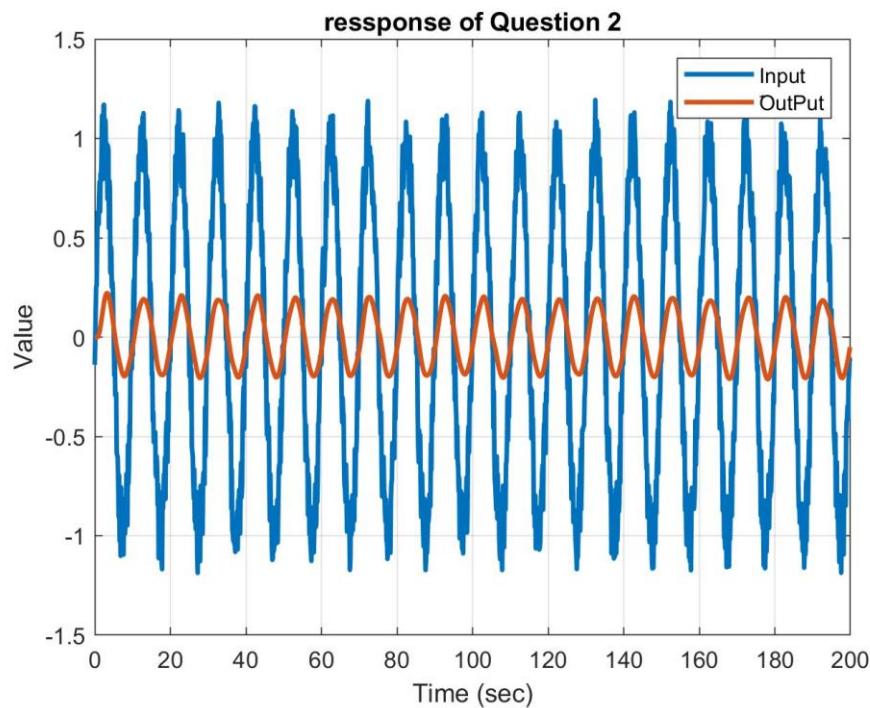
شکل 21-2 : تعداد پارامترهای صورت ۶ و تعداد پارامترهای مخرج ۶

2.5 اثر نویز های سفید و رنگی را بر دقت شناسایی

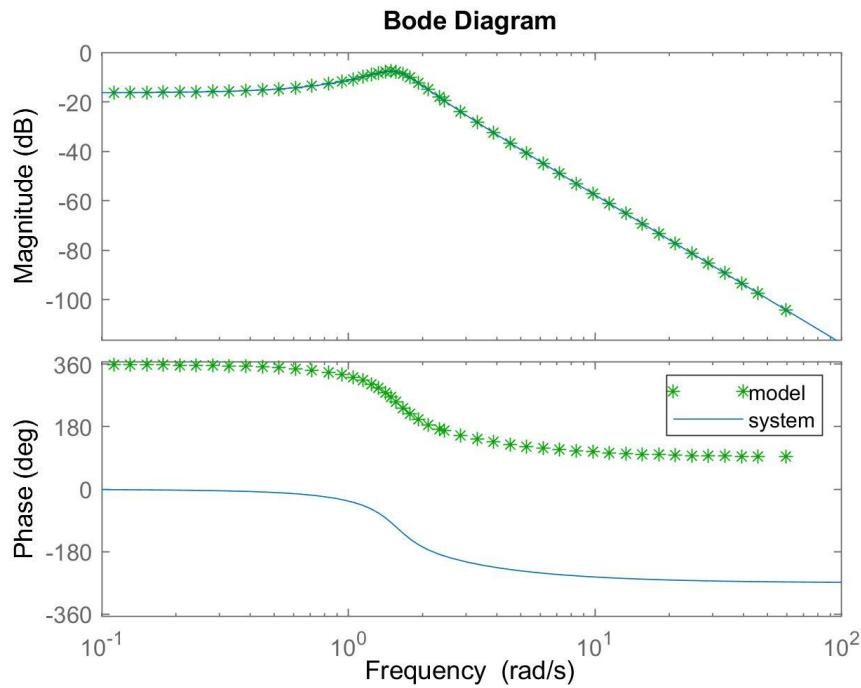
2.5.1 نویز سفید

واریانس نویز را طوری در نظر بگیرید که نسبت سیگنال خروجی به نویز حدود 10 باشد.

با استفاده از نویز سفید می‌توان با استفاده از هر سیگنال ورودی سیستم را شناسایی کرد. همانطور که در شکل زیر مشاهده می‌شود می‌توان با ورودی سینوسی با $\text{PE} = 2$ ، $\text{A} = 8$ پارامتر سیستم را شناسایی کرد.



شکل 2-22 : ورودی با نویز سفید



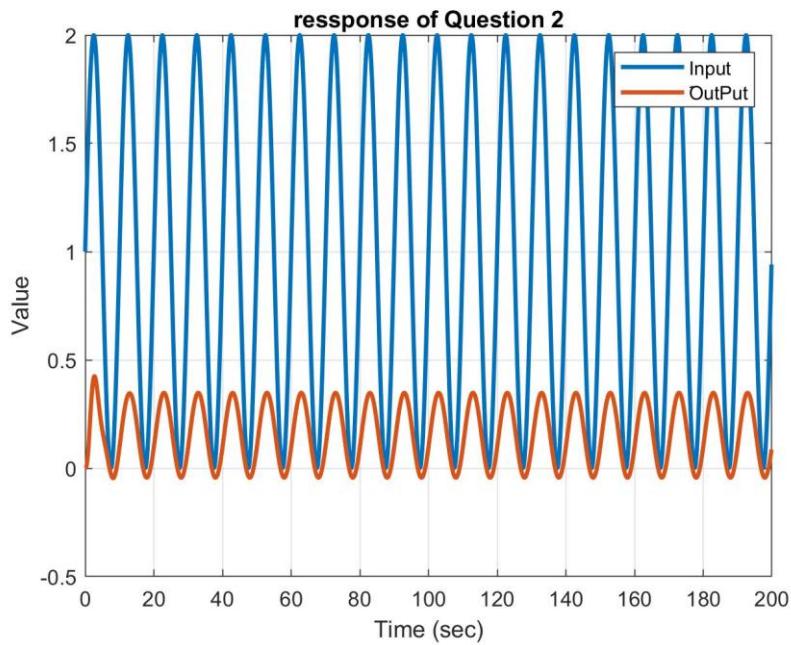
شکل 2-23 : دیاگرام بد برای نویز سفید

نویز رنگی 2.5.2

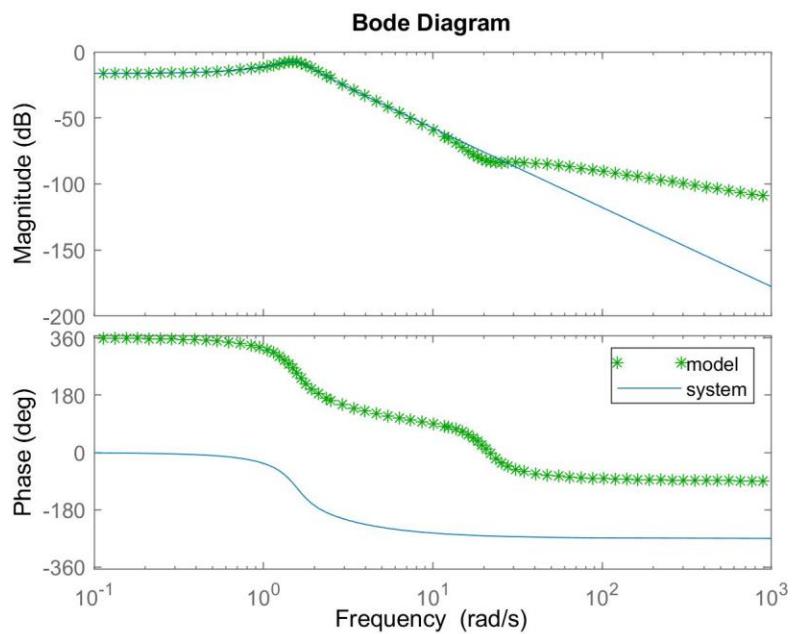
در این حالت خطا و محدوده اطمینان به شدت به ضرایب جملات نویز بستگی دارد و به خاطر وابسطه بودن داده ها به همدیگر شناسایی خیلی ضعیف صورت می گیرد.

وروهدی نویز

```
Noise(i)=+0.6*Noise(i-1)+0.75*Noise(i-2)-0.35*Noise(i-3)
```

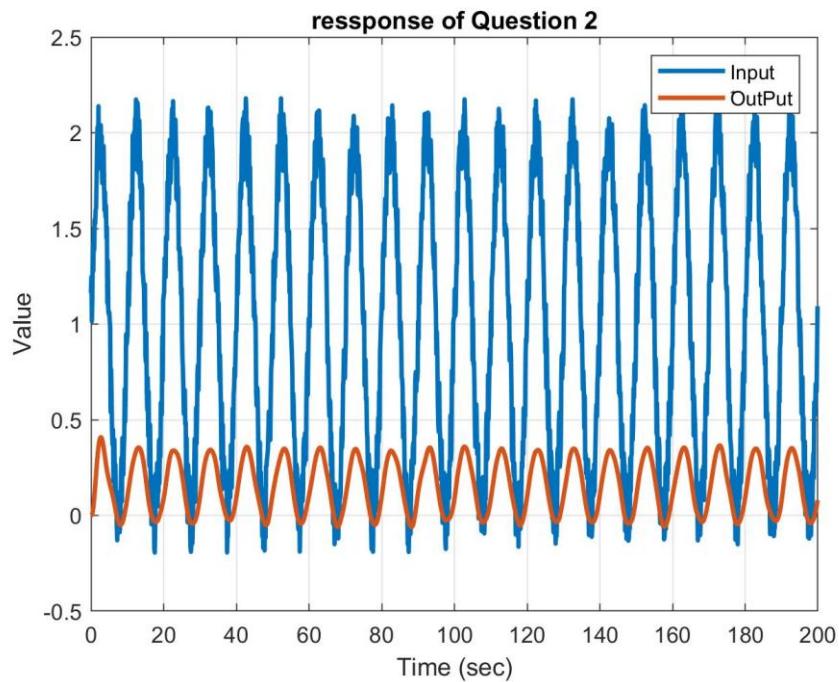


شکل 2-24: بروزی نویز رنگی

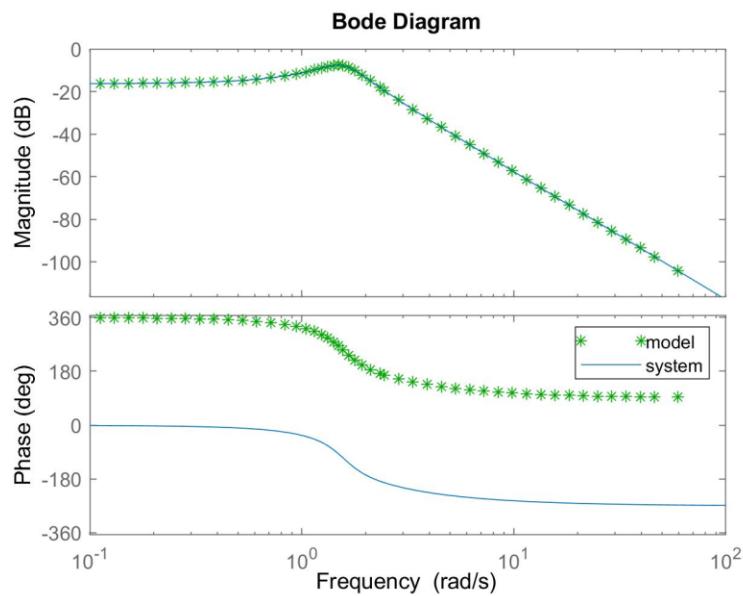


شکل 2-25: دیاگرام بود نویز رنگی

2.6 شناسایی ELS پارامترهای سیستم و چند جمله‌ای نویز نویز رنگی
با استفاده از نویز سفید و ورودی سینوسی به همراه نویز به جملات گذشته ورودی زیر تولید می‌شود.



شکل 2-26 : ورودی ELS



شکل 2-27 : ورودی ELS

2.6.1 آیا پارامترها به درستی همگرا شده‌اند؟

با ورودی سینوسی نه پارامترها و نه ضرایب نویز شناسایی نمی‌شوند ولی با اضافه کردن نویز سفید پارامترهای سیستم با تقریب خوبی شناسایی می‌شوند.

پارامترهای مجھول به آن علت که کمتر از ۱ و نزدیک به هم هستند شناسایی نمی‌شوند و مقدار آن‌ها در شناسایی ۵ شد.

2.7 با تغییر پارامترهای سیستم یک بار به طور ناگهانی و یک بار به طور آرام (در حدود ۱۰ درصد) در طول فرآیند شناسایی، از مکانیزم‌های Covariance resetting برای بهبود دقت شناسایی سیستم استفاده کنید

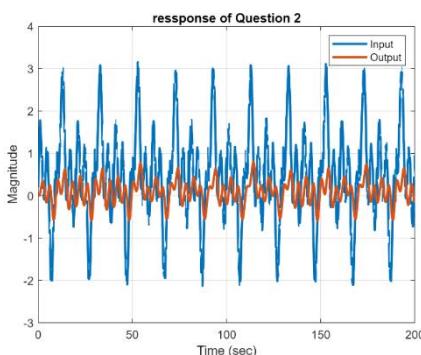
و نتایج را مقایسه کنید. با ذکر دلیل بیان نمایید که کدامیک از روش‌ها در تغییرات آرام و کدامیک در تغییرات ناگهانی پارامترها تاثیر بهتری دارند.

2.7.1 شناسایی با روش covariance resetting

این روش برای تغییر پارامتر به طور ناگهانی مورد استفاده قرار می‌گیرد. در شناسایی صورت گرفته است. در این حالت کلیه پارامترهای سیستم هر ۱۰۰ نمونه یکه بار در یک عدد تصادفی ضرب خواهند شد. روند شناسایی و تغییرات در یکی از پارامترها در شکل زیر نمایش داده شده است.

مشاهده می‌شود که این روش برای تغییرات ناگهانی خوب جواب میدهد البته زمان reset کردن پارامترها، و زمان P کردن میتواند در شناسایی اثر گذار باشد و روند متفاوتی به وجود آورد. در اینجا زمان RESET کردن P چند نمونه بعد از زمان کردن پارامترها است. حال اگر تغییر پارامترها به صورت تدریجی باشدو پس از مدتی یکتابع سینوسی با پارامترهای جمع شود به گونه‌ای که به آرامی پارامتر را تغییر دهد (حدود ۱۰ درصد) خواهیم داشت.

بعضی از پارامترهای سیستم را میتوان تخمین زد اما برخی با خطای تخمین میخورند و این مشکل با افزایش مقدار اولیه P در هر بار reset کردن حل نخواهد شد. بنابراین این روش برای تغییر پارامترها به صورت ارام نمیتواند مناسب باشد.



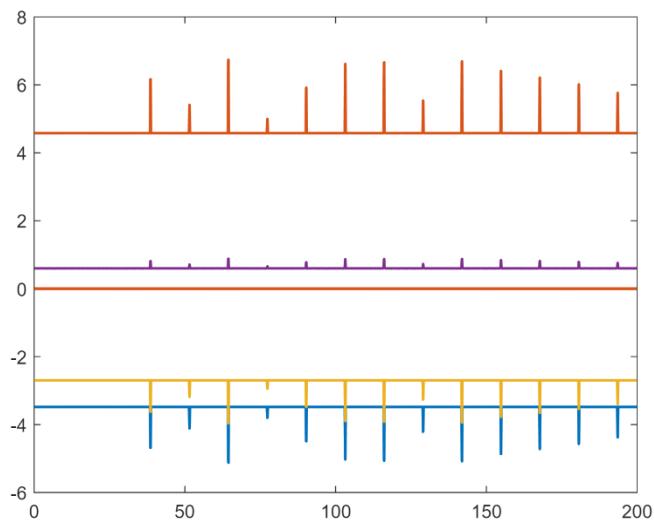
شکل 2-28 ورودی سیستم با نویز سفید

sudden parameter resetting

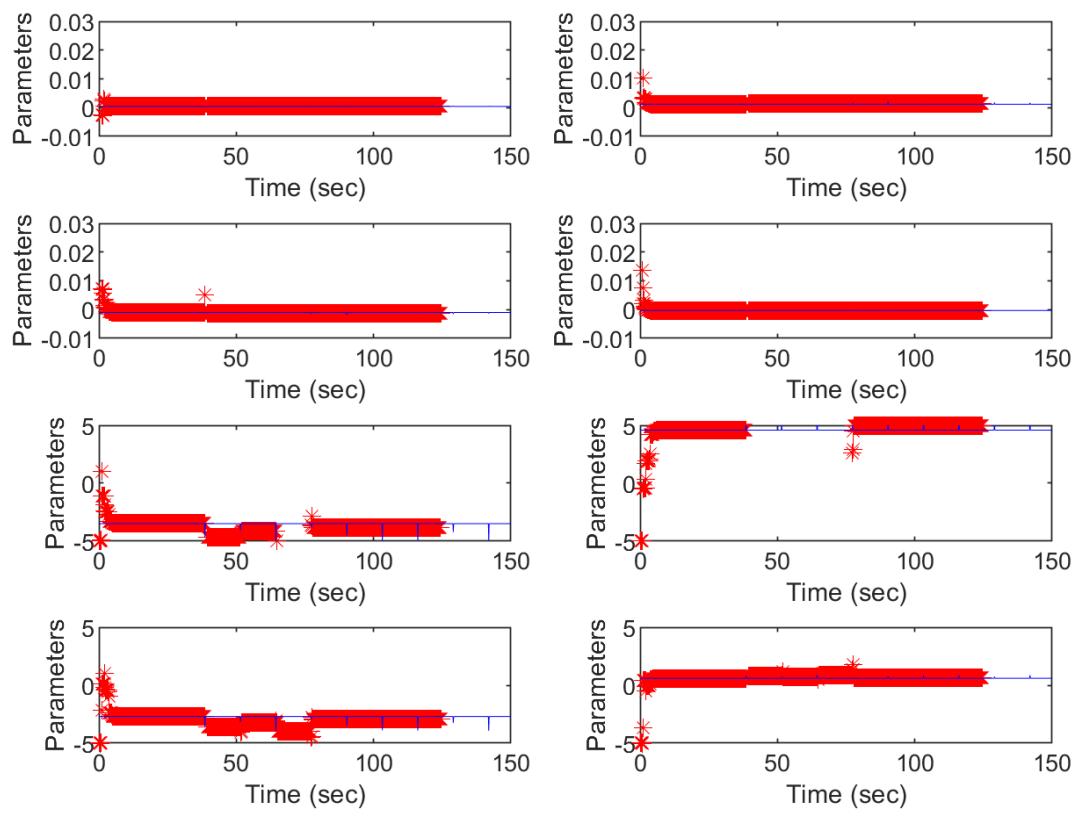
```

for temp=uu:numel(y)
    if floor(temp/100)==temp/100
        cc= c(2:end)+c(2:end)*(rand);
        dd=d(2:end)+d(2:end)*(rand)/2;
        paras(:,temp)=[dd,0,cc]';
    end
    y(temp)=[-(y(temp-1:-1:temp-4))',(u(temp-1:-1:temp-4))']* [dd,cc]';
end

```

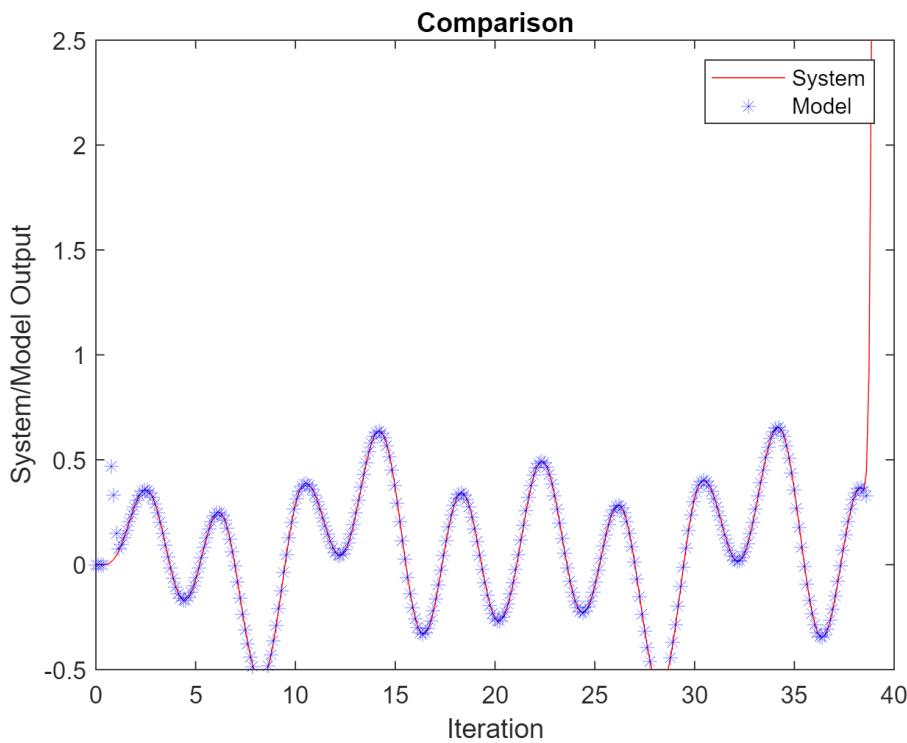


شکل 2-29 تغییرات آنی پارامترها



شکل 30-2 نمودار همگرایی پارامترهای صورت و مخرج تابع تبدیل تخمین زده شده

با تغییر پارامتر سیستم ناپایدار می شود (ثانیه ۱۵۰) و بعد از آن از آن از عدد $1e307$ خروجی بزرگتر می شود و شناسایی صورت نمی گیرد ولی تا آن جا تقریب پارامترها به خوبی صورت می گیرد.

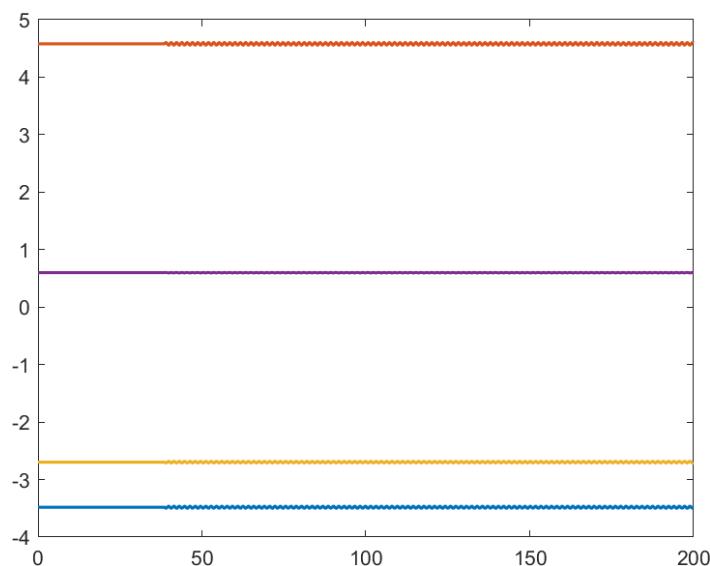


شکل 2-31 خروجی مدل و خروجی سیستم

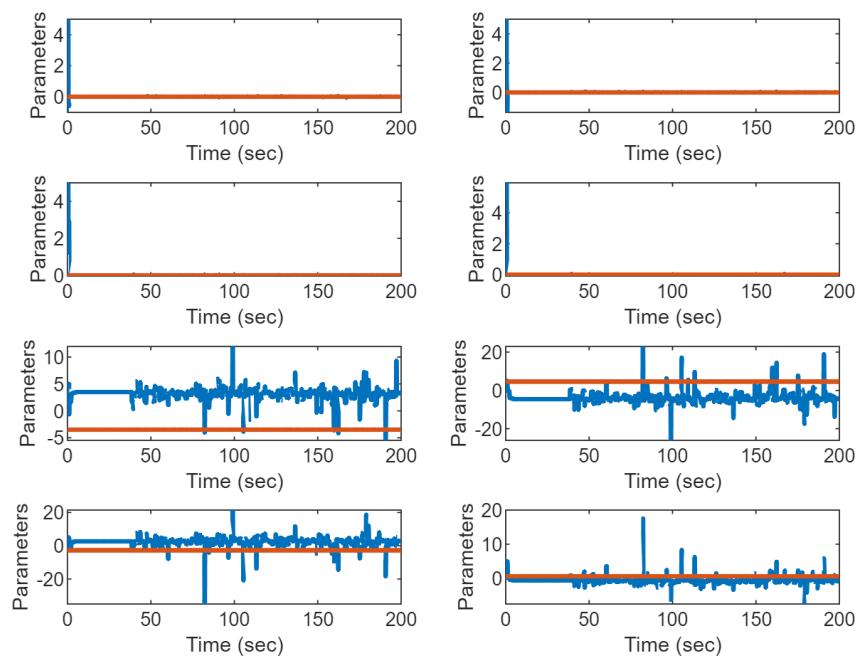
2.7.1.2 تغییرات آرام در پارامترهای سیستم

با ایجاد تغییراتی آرام در پارامترهای سیستم با دامنه 0.005 در مخرج و با فرکانس 0.5 پر صورت و با شناسایی سیستم کرده.

```
for temp=uu:numel(y)
    paras(:,temp)=[d(2:end)+d(2:end)*.005*(sin((temp-uu)/2)),(c+c*.5*(sin((temp-uu)/2)-1))]';
    y(temp)=[-(y(temp-1:-1:temp-4))',(u(temp:-1:temp-4))']*paras(:,temp);
end
```



شکل 2-32 پارامترهای سیستم در طول زمان



شکل 2-33 پارامترهای شناسایی شده در طول زمان

به وضوح مشاهد می‌شود که روش ریست کردن ماتریس کوواریانس برای تغییرات آرم مناسب نمی‌باشد.

روش 2.7.2 Forgetting Factor

در حالتی که پارامترها به آرامی تغییر میکنند، روش forgetting Factor پاسخ مناسبتری خواهد داشت. چرا که از دادههای قبلی که برای شناسایی در هنگام تغییر پارامتر مورد نیاز نیستند استفاده نمیکند. این روش برای تغییرات آرام استفاده شده است، داریم همانگونه که در شکلهای فوق مشخص است، این الگوریتم بهتر از الگوریتم قبلی، پارامترها را تخمین میزند اما با اشکالاتی نیز مواجه است و با مقداری خطأ همراه است. در اینجا مقدار ضریب لاندا (ضریب فراموشی) به صورت نمایی به یک میل میکند و در هر چند گام دوباره به مقدار اولیه 0.9048 برمیگردد.

```
lambda=exp(Step)
```

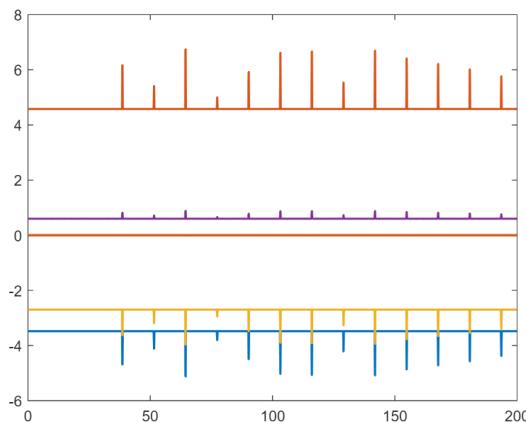
```
lambda = 1×51
0.9048    0.9066    0.9085    0.9103    0.9121    0.9139    0.9158
0.9176    0.9194    0.9213    0.9231    0.9250    0.9268    0.9287    0.9305
0.9324    0.9343    0.9361    0.9380    0.9399    0.9418    0.9436    0.9455
0.9474    0.9493    0.9512    0.9531    0.9550    0.9570    0.9589    0.9608
0.9627    0.9646    0.9666    0.9685    0.9704    0.9724    0.9743    0.9763
0.9782    0.9802    0.9822    0.9841    0.9861    0.9881    0.9900    0.9920
0.9940    0.9960    0.9980
```

در نظر گرفته شده است. موفقیت این روش به ورودی نیز وابسته است و اگر ورودی تغییرات نامناسبی داشته باشد، الگوریتم با شکست مواجه میشود اگر این روش برای تغییرات ناگهانی به کار رود نتایج حاصل عبارت خواهد بود از:

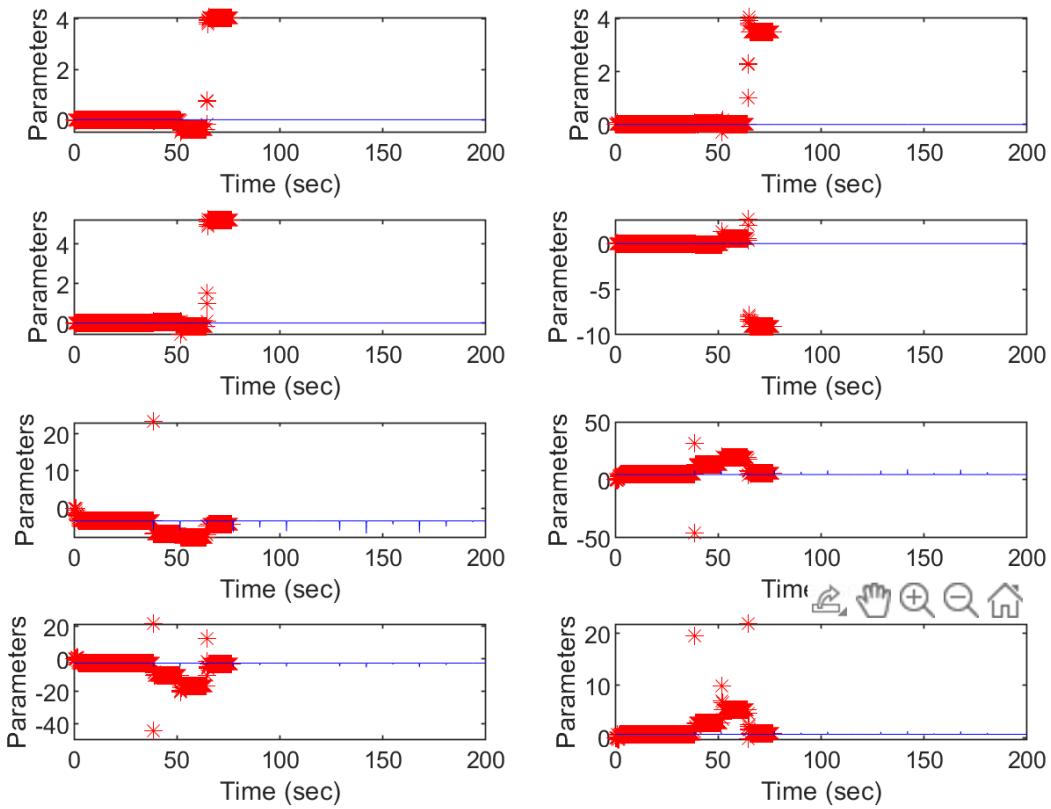
تغییرات ناگهانی 2.7.3

sudden parameter resetting

```
for temp=uu:numel(y)
    if floor(temp/100)==temp/100
        cc= c(2:end)+c(2:end)*(rand);
        dd=d(2:end)+d(2:end)*(rand)/2;
        paras(:,temp)=[dd,0,cc]';
    end
    y(temp)=[ -(y(temp-1:-1:temp-4))',(u(temp-1:-1:temp-4))' ]*[dd,cc]';
end
```



شکل 2-34 تغییرات آنی پارامترها

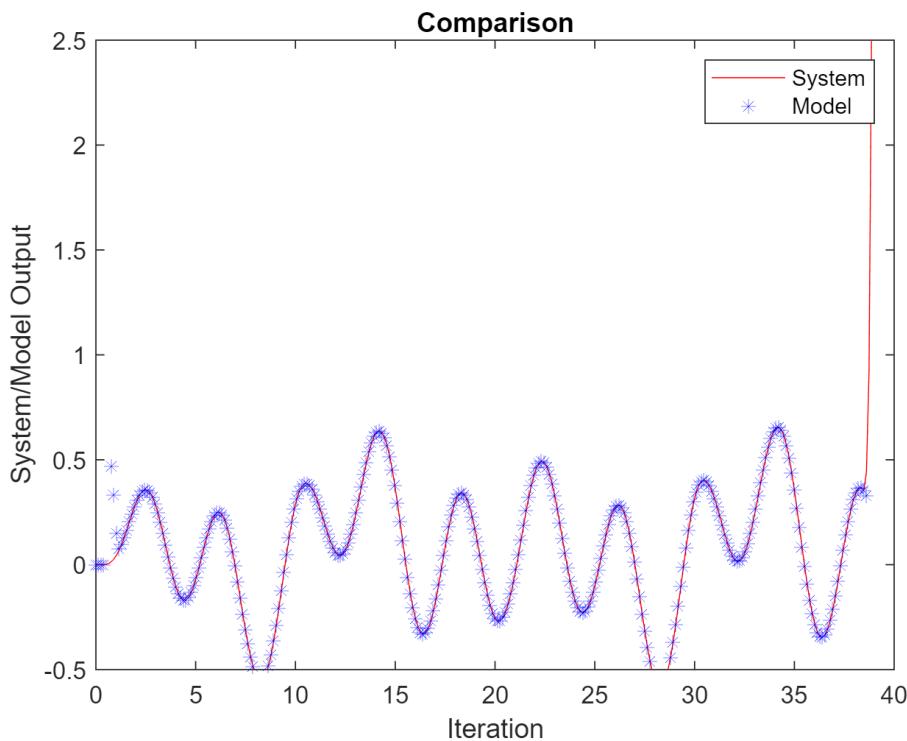


شکل 25-35 نمودار همگرایی پارامترهای صورت و مخرج تابع تبدیل تخمین زده شده

با تغییر پارامتر سیستم ناپایدار می شود (ثانیه ۱۵۰) و بعد از آن از عدد $1e307$ خروجی بزرگتر می شود و شناسایی صورت نمی گیرد ولی تا آن جا تقریب پارامترها به خوبی صورت می گیرد.

تخمین با روش لاندا تا ۷۵ ثانیه محاسبه انجام داد و بعد از آن ناپایدار شده است.

روشی مناسب برای تخمین پارامتر با تغییرات آنی نیست



شکل 2-36 خروجی مدل و خروجی سیستم

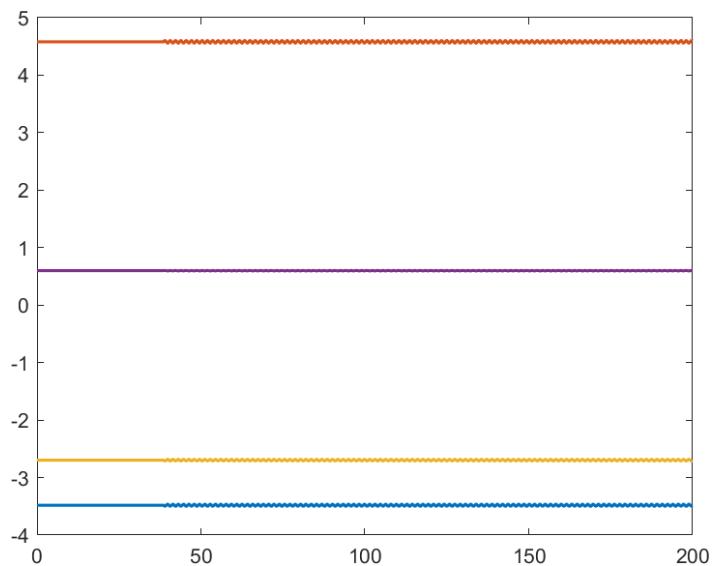
2.7.4 تغییرات آرام

با ایجاد تغییراتی آرام در پارامترهای سیستم با دامنه 0.005 در مخرج و 0.5 پر صورت و با فرکانس 0.5 اقدام به شناسایی سیستم کرد.

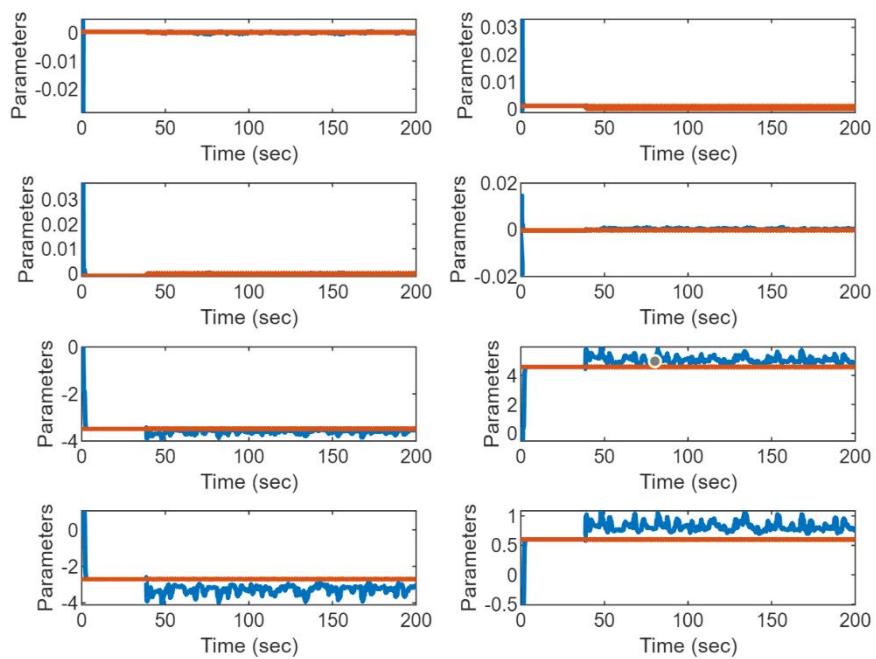
```

for temp=uu:numel(y)
    paras(:,temp)=[d(2:end)+d(2:end)*0.005*(sin((temp-uu)/2)),(c+c*.5*(sin((temp-uu)/2)-1))]';
    y(temp)=[-(y(temp-1:-1:temp-4))',(u(temp:-1:temp-4))']*paras(:,temp);
end

```



شکل 2-37 پارامتر های سیستم در طول زمان



شکل 2-38 پارامتر های شناسایی شده با تغییرات آرام

RLS,LMS,PA,SA 2.8

سیستم را با نویز سفید و با مرتبه مدل صحیح در نظر بگیرید و با روش‌های (RLS,LMS,PA,SA) شناسایی کنید و نتایج تخمین پارامترها را در شکل‌های مختلف بیاورید. همچنین مجموع مربع خطای تخمین خروجی، مجموع مربع خطای تخمین پارامترها و سرعت همگرایی روش‌های مختلف را در جدولی مقایسه نمایید.

RLS شناسایی با روش 2.8.1

اگر در خروجی نویز با دامنه $0 \dots 1$ قرار دهیم، و سیستم را با روش RLS شناسایی کنیم نتیجه حاصل عبارت خواهد بود از ان گونه که در شکل‌های فوق مشخص است، این روش پارامترها را تخمین می‌زند.

	Real	RLS
b_1	-3.4634	-3.4634
b_2	4.5497	4.5497
b_3	-2.6854	-2.6854
b_4	0.601	0.601
a_1	0.0004	0.0004
a_2	0.0011	0.0011
a_3	-0.001	-0.001
a_4	-0.0003	-0.0003

PA روش 2.8.2

ورودی با دامنه $0 \dots 1$ به دلیل این که این روش برای شناسایی در شرایط بدون نویز است، شناسایی پارامترها در این روش با موفقیت همراه نبوده است.

	Real	RLS	PA
b_1	-3.4634	-3.4634	-4.1379
b_2	4.5497	4.5497	-3.5814
b_3	-2.6854	-2.6854	-3.1379
b_4	0.601	0.601	-2.8166
a_1	0.0004	0.0004	4.0134
a_2	0.0011	0.0011	0.7907
a_3	-0.001	-0.001	-2.2782
a_4	-0.0003	-0.0003	-5.1288

SA روش 2.8.3

این روش برای سیستم با دامنه نویز $0 \dots 1$ در خروجی استفاده شده است. نتایج حاصل از این روش به شرح زیر است. از نتایج حاصله، میتوان گفت که این روش نیز برای محاسبه پارامترها در حالت وجود نویز، مناسب نمی‌باشد. در این روش مقدار P به محض استفاده به سرعت به سمت صفر می‌رود (در صورت انتخاب مقداری کوچک) و از یک طرف باعث همگرا شدن فوری جواب خواهد شد و اما از طرف دیگر اگر به موقع استفاده نشود، منجر به تخمین اشتباه می‌گردد. هم چنین این روش برای استفاده با نویز مناسب نیست.

	Real	RLS	SA	PA
b_1	-3.4634	-3.4634	-0.1173	-4.1379
b_2	4.5497	4.5497	-0.0741	-3.5814
b_3	-2.6854	-2.6854	-0.0058	-3.1379
b_4	0.601	0.601	0.0877	-2.8166
a_1	0.0004	0.0004	3.4671	4.0134
a_2	0.0011	0.0011	-0.7951	0.7907
a_3	-0.001	-0.001	-0.1503	-2.2782
a_4	-0.0003	-0.0003	-2.9851	-5.1288

LMS روش 2.8.4

با استفاده از این روش و با در نظر گرفتن مقدار $\gamma = 0.001$ نتایج زیر حاصل میشود. این الگوریتم در تخمین تابع در حضور نویز با مشکل مواجه است و به شدت به تغییرات γ و ورودی حساس است و ممکن است در مقادیر بسیار زیادی از گاما باید برای سیستمی تست شود تا این روش قابل استفاده باشد.

	Real	RLS	SA	PA	LMS
b_1	-3.4634	-3.4634	-0.1173	-4.1379	-2.9224
b_2	4.5497	4.5497	-0.0741	-3.5814	-2.5358
b_3	-2.6854	-2.6854	-0.0058	-3.1379	-2.2277
b_4	0.601	0.601	0.0877	-2.8166	-2.0043
a_1	0.0004	0.0004	3.4671	4.0134	2.7978
a_2	0.0011	0.0011	-0.7951	0.7907	0.559
a_3	-0.001	-0.001	-0.1503	-2.2782	-1.5725
a_4	-0.0003	-0.0003	-2.9851	-5.1288	-3.5516

با توجه به نمودارها، سرعت همگرایی روش PA از تمامی روشها بیشتر است. اما ممکن است به جواب صحیح همگرا نشود. البته لحظه وارد کردن پارامتر P نیز در این سرعت بی تاثیر نیست. به نظر میرسد که در صورت تنظیم شرایط روش RLS بهترین جواب را در حضور نویز به ما بدهد.

3 بررسی اثر فیدبک

3.1 با طراحی یک کنترل کننده ساده، سیستم با سه قطب و دو صفر دلخواه را تحت فیدبک قرار داده و اثر فیدبک را بر شناسایی سیستم مورد نظر بررسی کنید.

با در نظر گرفتن نویز سفید به عنوان ورودی (که باعث مستقل شدن ماتریس ϕ' خواهد شد) می‌توان هر تعداد پارامتر سیستم را شناسایی نمود

سیستم در نظر گرفته شده

```
num=[0 1 3 7];  
den=[1 8 7 7];
```

که با استفاده از کنترل کننده PI با ضرایب

```
kc=7.79967495151787+182.26982236127/s;
```

تعیین شده و سیستم پایدار است.

سیستم گسسته حلقه باز به صورت

```
sys_dis_open =  
  
0.1604 z^2 - 0.1958 z + 0.07861  
-----  
z^3 - 1.937 z^2 + 1.122 z - 0.1421
```

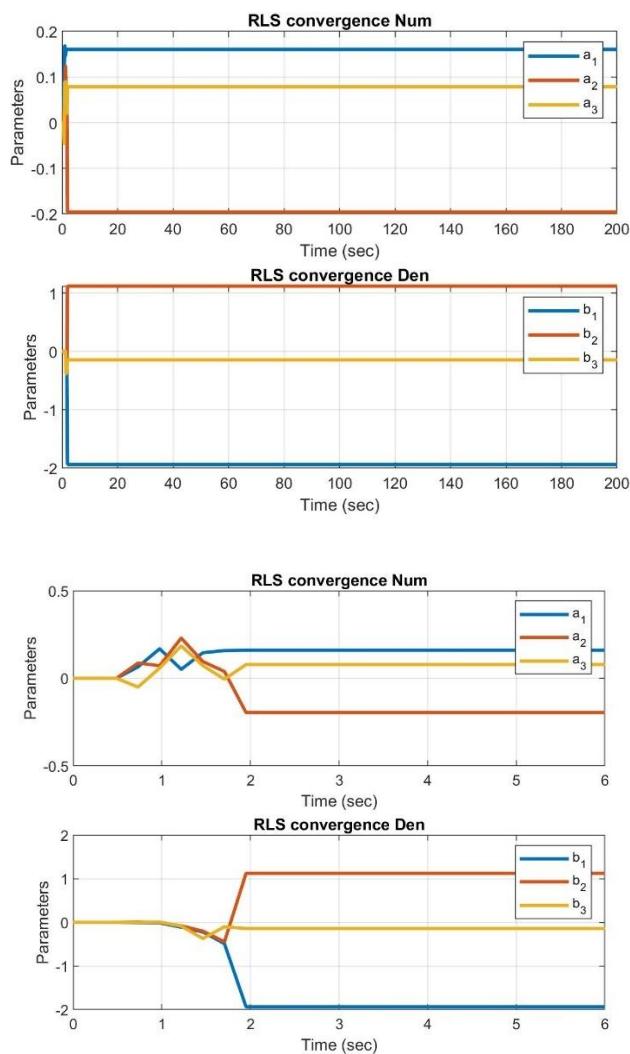
```
Sample time: 0.24388 seconds  
Discrete-time transfer function.
```

و سیستم گسسته حلقه بسته با کنترل کننده PI به صورت

```
sys_dis_close =  
  
0.1328 z^3 - 0.3505 z^2 + 0.3048 z - 0.08705  
-----  
z^4 - 3.73 z^3 + 5.24 z^2 - 3.288 z + 0.778
```

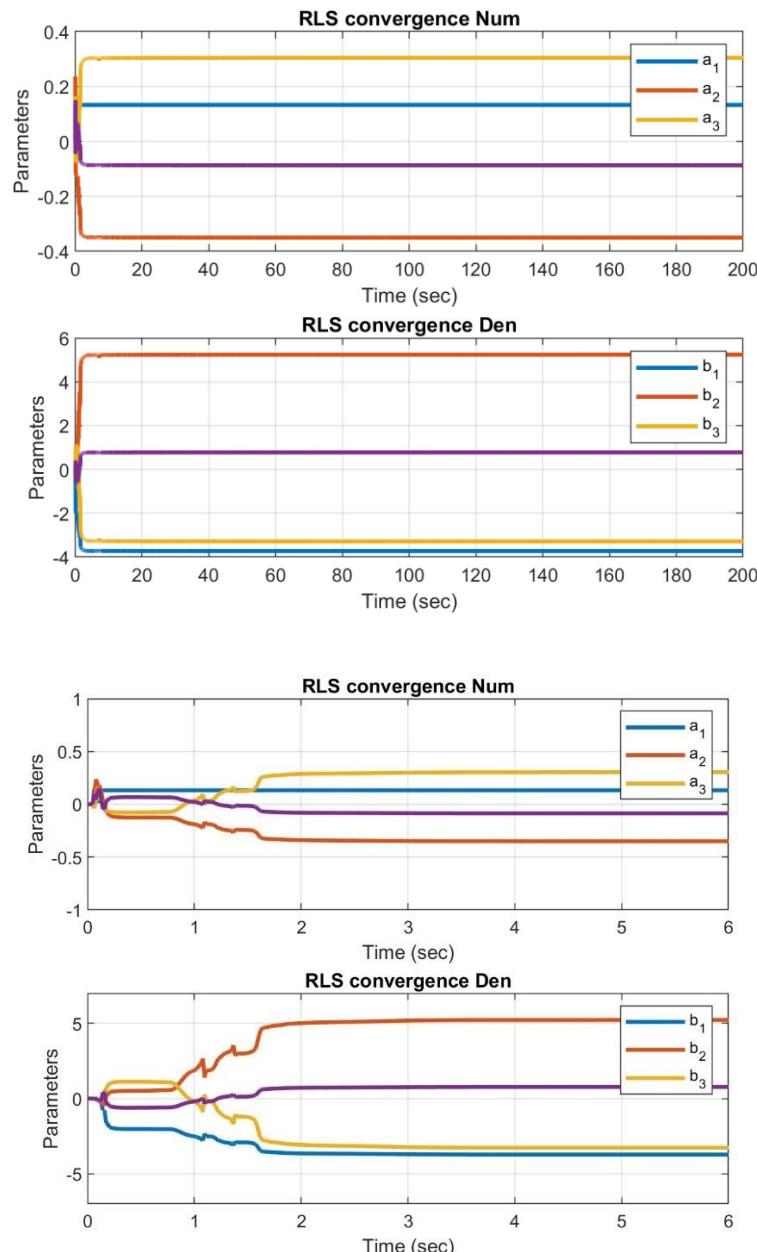
```
Sample time: 0.015887 seconds  
Discrete-time transfer function.
```

همانطور که انتظار می‌رود شناسایی سیستم حلقه باز به سادگی صورت می‌گیرد در حدود ۲ ثانیه از شروع به کار



شکل 3-1 : همگرایی پارامترهای سوال ۱.۳ برای سیستم حلقه باز در کمتر از ۲ ثانیه

برای شناسایی سیستم حلقه بسته زمان بیشتری لازم است زیرا ورودی به خروجی وابسطه می‌شود و به خاطر استفاده از نویز در ورودی بسیار محتمل است که در تخمینت‌های اولیه مقدار پارامترها اعدادی بسیار بزرگ (در حدود ۱۰۰) شوند. پارامترهای مخرج دیرتر همگرا می‌شوند.



شکل 3-2 : همگرایی پارامترها برای سیستم حلقه بسته

3.2 پارامترهای یک سیستم ناپایدار مرتبه 3 را شناسایی کنید.

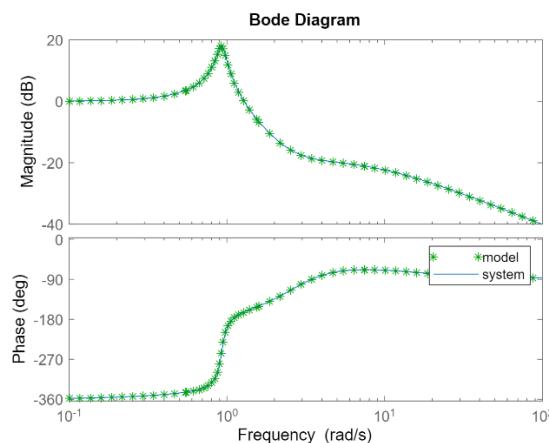
شناسایی سیستم ناپایدار در حالت کلی ممکن نیست زیرا با نامحدود شدن خروجی سیستم مقدار پارامترهای تخمین زده شده نیز به عددی همگرا نمی‌شوند و صرفه رشد می‌کنند.

اما در زمان‌ها نمونه برداری کوچک و برای سیستم‌هایی ریشه و قطب مثبت کوچک می‌توان سیستم را شناسایی کرد.

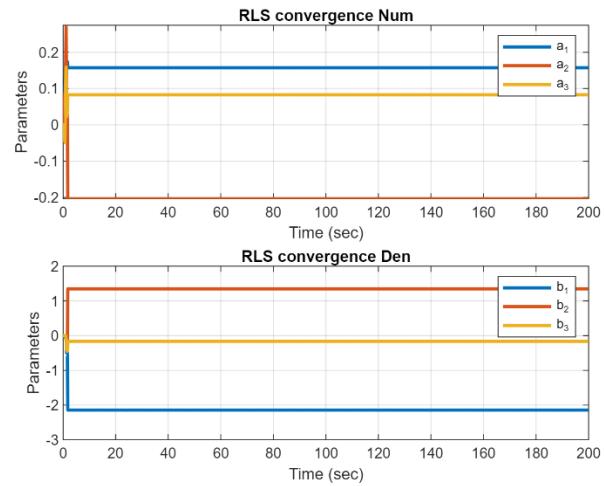
```
s=tf('s');
num=[0 1 3 7];
den=[1 8 -0.05 7];
sys_cont_open=tf(num,den)
sys_cont_open =
s^2 + 3 s + 7
-----
s^3 + 8 s^2 - 0.05 s + 7

Continuous-time transfer function.
fb_openloop = bandwidth(sys_cont_open);
T_s=0.05*2*pi/fb_openloop;
sys_dis_open = c2d(sys_cont_open, T_s, 'zoh')
sys_dis_open =
0.1572 z^2 - 0.2038 z + 0.08298
-----
z^3 - 2.145 z^2 + 1.349 z - 0.1674

Sample time: 0.22343 seconds
Discrete-time transfer function.
```



شکل 3-3 : شناسایی سیستم ناپایدار



4 شناسایی با فیلتر کالمن

سیستم قسمت (1-3) را در نظر بگیرید.

رویت پذیری سیستم باید بررسی شود.

```
[A,B,C,D]=tf2ss(c,d)
A = 3x3
1.9367 -1.1220 0.1421
1.0000 0 0
0 1.0000 0
B = 3x1
1
0
0
C = 1x3
0.1604 -0.1958 0.0786
D = 0
rank(obsv(A,C))
ans = 3
[A,B,C,D]=tf2ss(c1,d1)
A = 4x4
3.7296 -5.2395 3.2878 -0.7780
1.0000 0 0 0
0 1.0000 0 0
0 0 1.0000 0
B = 4x1
1
0
0
0
C = 1x4
0.1328 -0.3505 0.3048 -0.0871
D = 0
rank(obsv(A,C))
ans = 4
```

هر دو سیستم بسته و باز رویت پذیر هستند.

4.1 با استفاده از فیلتر کالمن پارامترهای سیستم را شناسایی کنید و نتایج را تحلیل کنید.

4.1.1 سیستم حلقه باز

سیستم مورد استفاده در این بخش

```
run("Q310_Basic mlx")
sys_dis_open =
0.1604 z^2 - 0.1958 z + 0.07861
-----
z^3 - 1.937 z^2 + 1.122 z - 0.1421
```

```

Sample time: 0.24388 seconds
Discrete-time transfer function.

sys_dis_close =

0.1328 z^3 - 0.3505 z^2 + 0.3048 z - 0.08705
-----
z^4 - 3.73 z^3 + 5.24 z^2 - 3.288 z + 0.778

```

```

Sample time: 0.015887 seconds
Discrete-time transfer function.

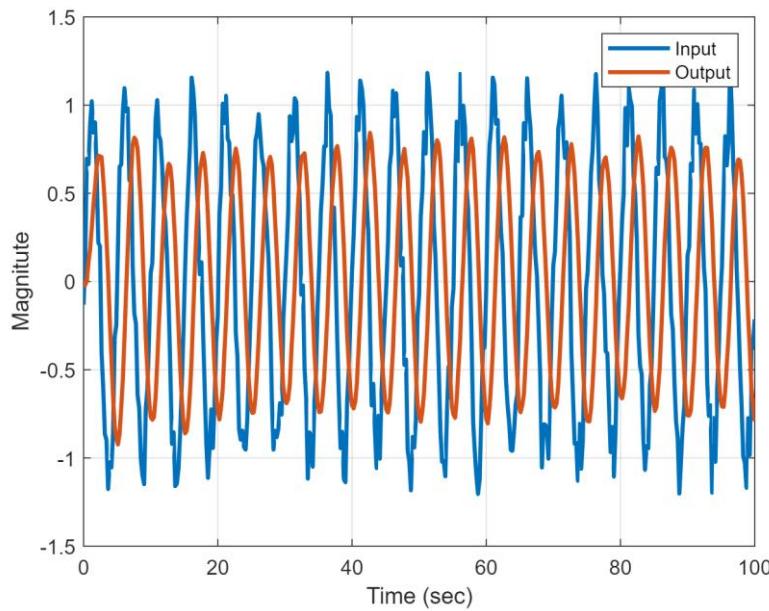
```

ورودی برای شناسایی سیستم حلقه باز

```

tfinal=100;
t = 0:T_s:tfinal;
u = gensig('sine' , tfinal/20 , tfinal ,T_s);
var_e=0.05;
e=sqrt(var_e);
Noise=(-e+(e+e)*rand(numel(t),1));
u=u+Noise;
y = lsim(sys_dis_open ,u ,t);
plot(t,u ,t , y , 'LineWidth',2) ;
xlabel('Time (sec)') ;
ylabel('Magnitude') ;
grid on
legend('Input','Output') ;

```



شکل 4-1 : ورودی حلقه باز برای شناسایی با فیلتر کالمن

معادلات فیلتر کالمن به صورت

```

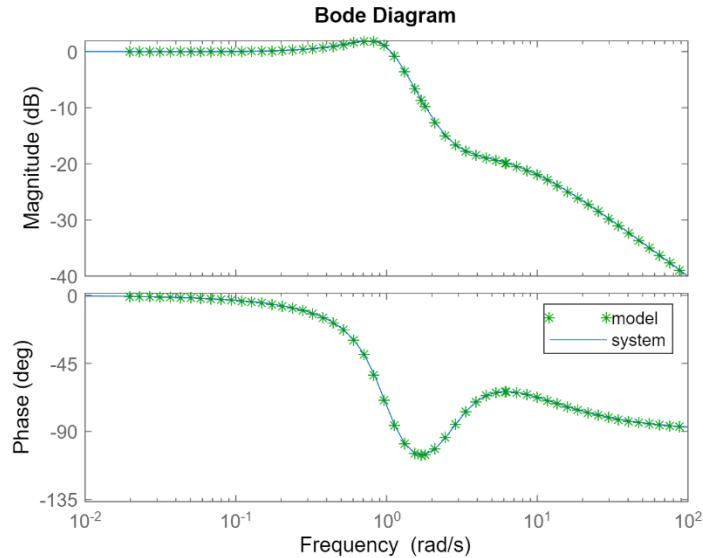
%choose number of parameters
Parameters_in_den=3
Parameters_in_num=3
% %-----%
Nv=Parameters_in_num+Parameters_in_den;

p_KF(1:Nv,1:Nv,1:N)=zeros(Nv,Nv,N);
%initial Conditions
theta_hat_KF(1:Nv,1:N)=zeros(Nv,N);
K_KF(1:Nv,1:N)=zeros(Nv,N);
var_e=0.05;

p_KF(1:Nv,1:Nv,1)=1e5*eye(Nv);p_KF(1:Nv,1:Nv,2)=p_KF(1:Nv,1:Nv,1);p_KF(1:Nv,1:Nv,
3)=p_KF(1:Nv,1:Nv,2);
for i=(max(Parameters_in_num,Parameters_in_den)+1):N
    phi_KF(:,i)=[(y(i-1:-1:i-Parameters_in_den))',(u(i-1:-1:i-Parameters_in_num))']';
    K_KF(:,i) =p_KF(:,:,i-1)*phi_KF(:,i)*(1+phi_KF(:,i)'*p_KF(:,:,i-1)*phi_KF(:,i))^-1 ;
    p_KF(:,:,i)=p_KF(:,:,i-1)-p_KF(:,:,i-1)*phi_KF(:,i)*(1+phi_KF(:,i)'*p_KF(:,:,i-
1)*phi_KF(:,i))^-1*phi_KF(:,i)'*p_KF(:,:,i-1)+var_e;
    theta_hat_KF(:,i)=theta_hat_KF(:,i-1)+K_KF(:,i)*(y(i)-phi_KF(:,i)'*theta_hat_KF(:,i-1));
end

bode(ident_analog , 'g*',sys_cont_open )
legend('model ','system')

```



شکل 2-4: نمودار بود برای مدل شناسایی شده با روش فیلتر کالمن

KF Convergence

```

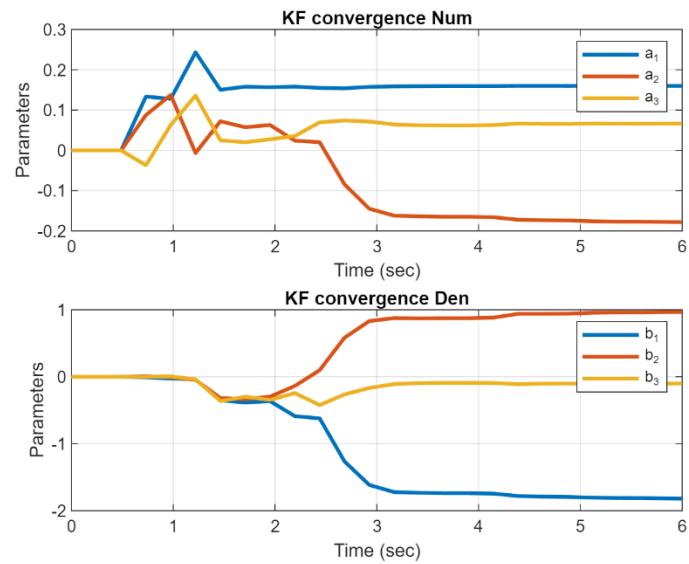
plot(t , theta_hat_KF(:,:,1) , 'LineWidth' , 2) ;
xlabel('Time (sec)');

```

```

ylabel('Parameters') ;
title('Kalman Filter convergence') ;
grid on

```



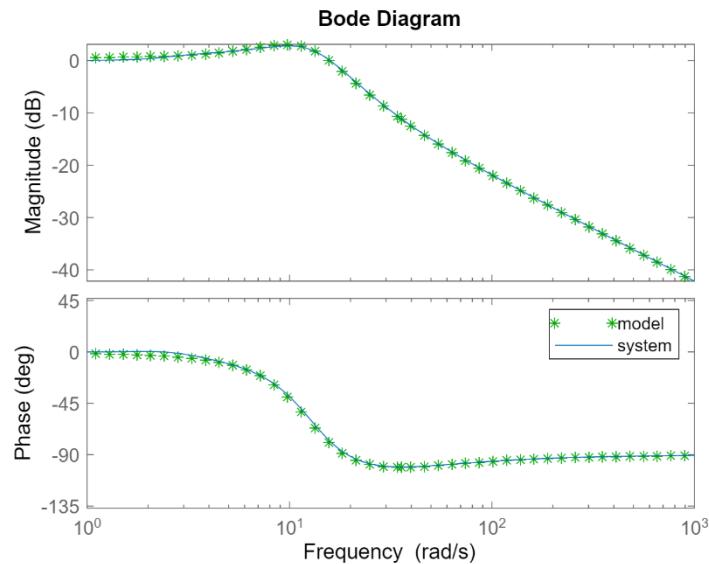
شکل 4-3: نمودار همگرایی پارامترها در فیلتر کالمن برای سیستم حلقه باز

فیلتر کالمن سیستم حلقه باز را به خوبی و با سرعتی قابل قبول شبیه حداقل مربعات بازگشتی شناسایی کرده است.

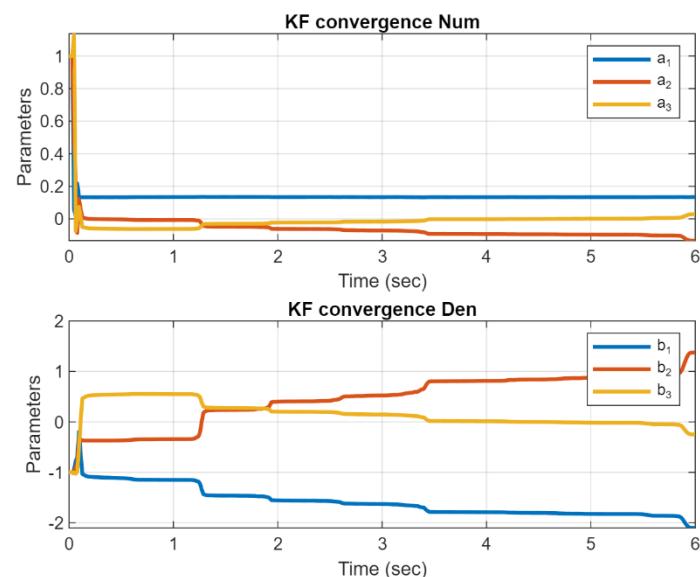
	Real	KF
b_1	-1.9367	-1.9358
b_2	1.122	1.1203
b_3	-0.1421	-0.1413
a_1	0.1604	0.1604
a_2	-0.1958	-0.1957
a_3	0.0786	0.0784

4.1.2 سیستم حلقه بسته با کنترل PI

فیلتر کالمن ۳ پارامتر از صورت و ۳ پارامتر از مخرج رامی تواند مجرأ تخمین بزند و با افزایش مرتبه مدل تمامی پارامترها به یک عدد همگرا می‌شوند.



شکل 4-4: نمودار بد برای سیستم حلقه بسته



شکل 4-5 : همگرایی پارامترها در مدل حلقه بسته

Closed Loop

	Real	KF	RLS
b_1	-3.7296	-0.0001	-3.7297
b_2	5.2395	-2.7336	5.2396
b_3	-3.2878	2.515	-3.2879
b_4	0.778	-0.7796	0.7781
a_1	0.1328	0.0001	0.1328
a_2	-0.3505	-0.1328	-0.3505
a_3	0.3048	-0.2182	0.3048
a_4	-0.0871	0.0872	-0.0871

4.2 نتایج قسمت (1-4) را با RLS مقایسه کنید.

چون روش کالمن از دسته‌ی فیلترها است دارای بایاس می‌باشد و تخمین به خود مقدار پارامتر همگرا نمی‌شود در حالت ورودی با نویز سفید اندکی عملکردی با خطای بیشتری نسبت به روش حداقل مربعات بازگشتی دارد.

روش حداقل مربعات بازگشتی در تخمین پارامترهای سیستم حلقه بسته به خوبی عمل می‌کند در حالی که تخمین کالمن ناپایدار می‌شد و انحراف بسیاری با پارامترهای حقیقی سیستم خواهد داشت.

ولی فیلتر کالمن با ورودی نویز رنگی عملکردی بهتر (حدود ۲ درصد) نسبت به روش حداقل مربعات بازگشتی دارد.

4.3 در پارامتر تغییرات آرام ایجاد نمایید به طوری که سیستم ناپایدار نشود. با استفاده از فیلتر کالمن پارامترها را شناسایی کنید.

با دو تغییر در پارامترهای سیستم به صورت سینوسی و روان قصد بررسی نحو همگرایی فیلتر کالمن را داریم.

```
tfinal=500;
t = 0:T_s:tfinal;
u = gensig('sine' , tfinal/20 , tfinal ,T_s);
var_e=0.05;
e=sqrt(var_e);
Noise=(-e+(e+e)*rand(numel(t),1));
u=u+Noise;
y = lsim(sys_dis_open ,u ,t);

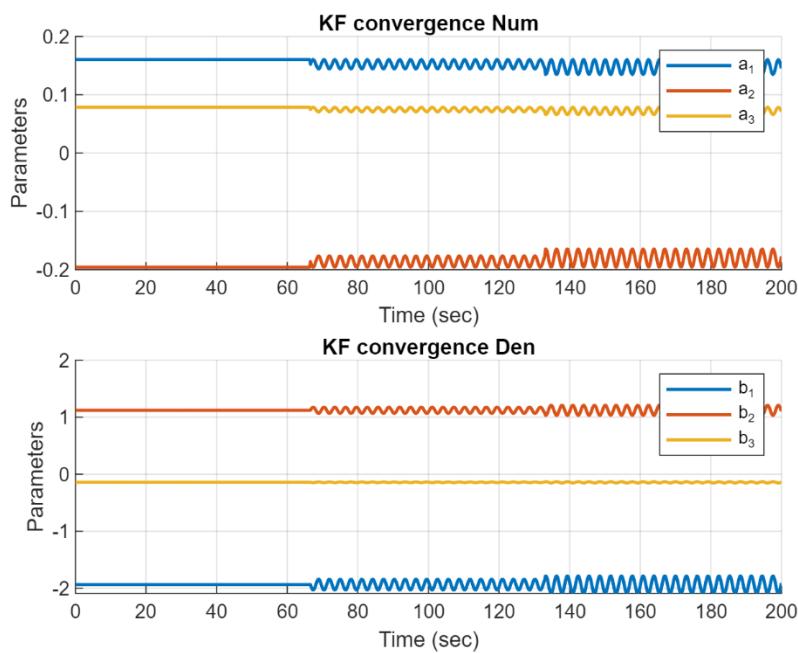
smooth parameter variation

for i=1:numel(t)
    paras(:,i)=[d(2:end),c]';
end

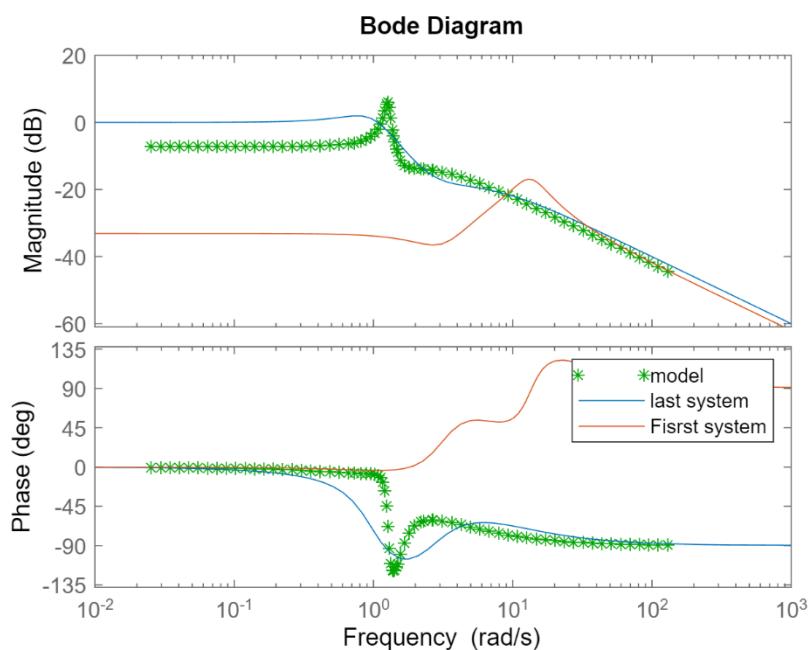
uu=400
uuu=800

for temp=uu:uuu
    paras(:,temp)=[d(2:end)+d(2:end)*.05*(sin(.5*(temp-uu))), (c+c*.05*(sin(.5*(temp-uu))-1))]';
    y(temp)=[ -(y(temp-1:-1:temp-3))',(u(temp:-1:temp-3))'*paras(:,temp)];
end

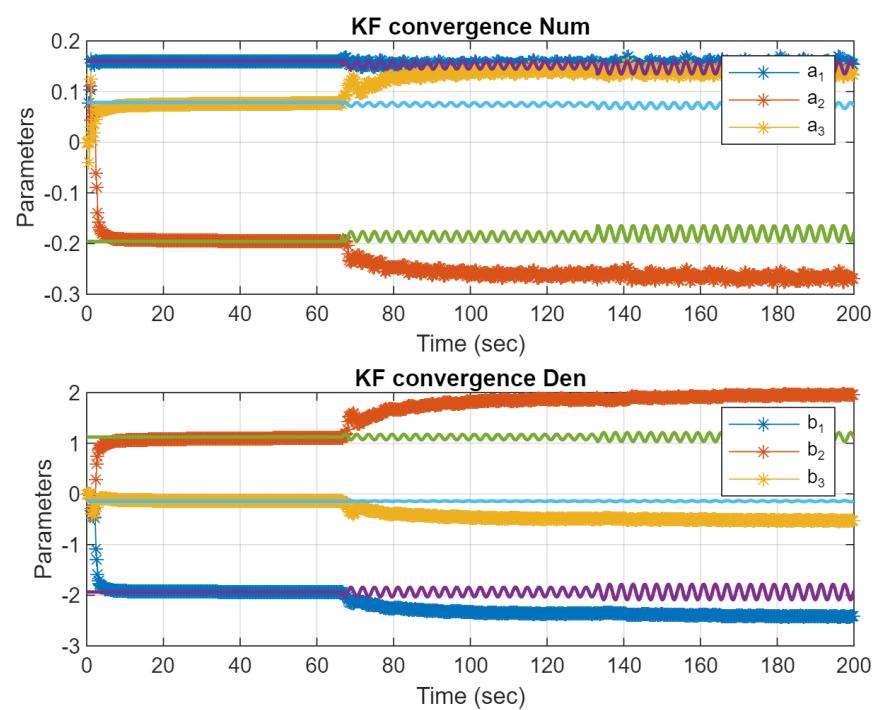
for temp=uuu:numel(y)
    paras(:,temp)=[d(2:end)+d(2:end)*.08*(sin(.5*(temp-uu))), (c+c*.08*(sin(.5*(temp-uu))-1))]';
    y(temp)=[ -(y(temp-1:-1:temp-3))',(u(temp:-1:temp-3))'*paras(:,temp)];
end
```



شکل 6-4: پارامترهای سیستم با تغییر آرام



شکل 7-4: نمودار بد برای وضعیت‌های مختلف سیستم و مدل که بین‌ها آن‌ها شناسایی شده است.



شکل ۴-۸: تخمین بایاس دار پارامترها پس از تغییر پارامترها در فیلتر کالمن

5 شناسایی سیستم غیرخطی

سیستم پاندول غیرخطی زیر را در نظر بگیرید و با استفاده از روش RLS آن را شناسایی کنید. سپس اثر تغییر مرتبه مدل در مقدار بهینه تابع هزینه را بررسی و مرتبه مدل مناسب را مشخص کنید. برای این قسمت می‌توانید از معیار $AIC = N \ln(V_N(\hat{\theta}_N)) + 2p$ هم استفاده کنید که در آن N تعداد داده‌ها و p تعداد پارامترها است.

$$\begin{aligned}x_1[k+1] &= a_{12}x_2[k] \\x_2[k+1] &= a_{21} \sin(x_1[k]) + a_{22}x_2[k] \\a_{12} &= 2, a_{21} = -\frac{g}{l} = -4, a_{22} = -\frac{k}{m} = -0,3\end{aligned}$$

برای شناسایی سیستم نیاز به داشتن داده‌های سیستم داریم

می‌توان معادله یک را در معادله دوم جایگذاری کرد و یک معادله را برای سیستم در نظر گرفت.

$$x_2[k+1] = a_{21} \sin(a_{21}x_2[k-1]) + a_{22}x_2[k] + U[k]$$

در این مسئله خواسته شده که یک سیستم غیر خطی با استفاده از مدل خطی شناسایی گردد. با توجه به این که مرتبه غیرخطی بودن سیستم خیلی بالا است برای شناسایی کامل سیستم، باید مرتبه مدل پیشنهادی را به شدت بالا برد که در شکل زیر تا مرتبه ۲۵۰ محاسبه شده است.

سیستم بتواند با دقت خوبی و تا زمانی معین (حدود ۶۰۰ برابر زمان نمونه برداری) پارامترها را شناسایی کند. اما در نهایت سیستم شناسایی شده تا بینهایت با سیستم اصلی تطبیق نخواهد داشت. اما همانگونه که در شکل های ارائه شده، مشاهده می گردد، در صورتی که ضریب بخش سینوس کم شود و یا به عبارت دیگر مرتبه غیر خطی بود سیستم کاهش یابد، مدل خطی به طور کامل سیگنال ورودی را شناسایی می‌کند.

معیار AIC چون درون خود فرمول تعداد پارامترها دخیل هستند با افزایش تعدادشان تاثیری در بهبود مدل نشان نمی‌دهند در صورتی که با افزایش رتبه سیستم به وضوح دقت مدل افزایش می‌یابد.

```
for Degree=1:10:100
    %number of parameters
    Parameters_in_den=Degree*5;
    Parameters_in_num=Degree*5+1;
    Nv=Parameters_in_num+Parameters_in_den

    phi=[];
    Eror=zeros(1,N);

    theta(1:Nv,1:Nv) = zeros(Nv,Nv) ;
    P = 1e12*eye(Nv) ;
    for i = (max(Parameters_in_num,Parameters_in_den)+1):N
        phi(:,i) = [(y(i-1:-1:i-Parameters_in_den))', (u(i-1:-1:i-Parameters_in_num))'];
        K = P*phi(:,i)*(1+phi(:,i)'*P*phi(:,i))^-(-1) ;
        P = (eye(Nv) - K*phi(:,i)')*P ;
```

```

theta(:,i) = theta(:,i-1) + K*(y(i) - phi(:,i)'*theta(:,i-1));
end
AIC(Degree)=N*log(norm(y(1:i)-theta(1:i))+2*Nv);
Trend(Nv)=AIC(end);
end

```

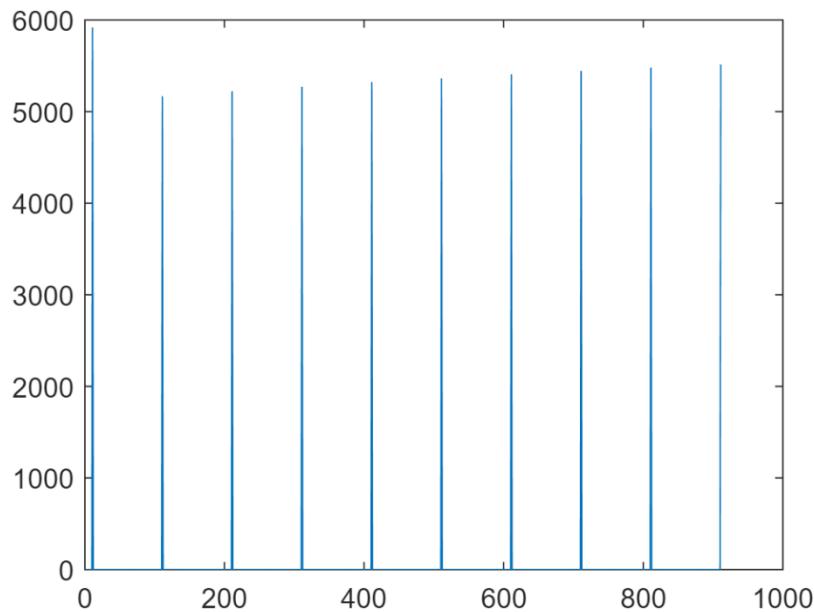
در کد بالا برای مدل‌هایی با

```

Nv = 11
Nv = 111
Nv = 211
Nv = 311
Nv = 411
Nv = 511
Nv = 611
Nv = 711
Nv = 811
Nv = 911

```

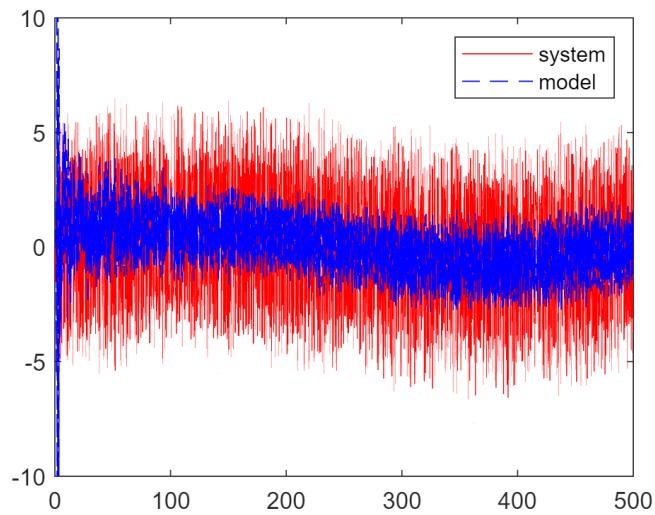
در نمودار زیر برای آن‌ها ترسیم شده است. AIC تعداد پارامتر حداقل مربعات بازگشته محاسبه شده و در اخر معیار



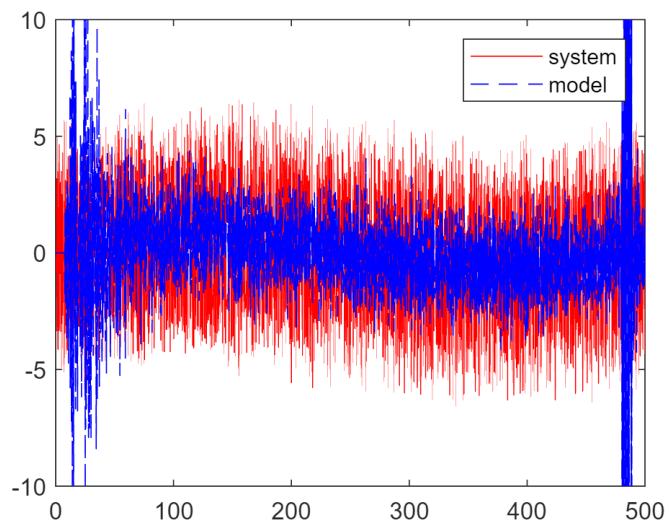
شکل 5-1: مقدار AIC و تعداد پارامترهای تخمین زده شده

تاثیر میزان افزایش مرتبه مدل و رسم خروجی از سیستم غیر خطی

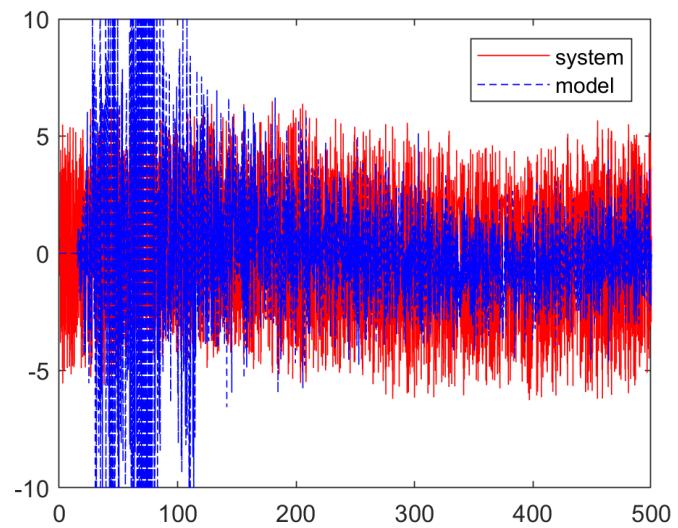
با افزایش مرتبه سیستم محدوده اطمینان تخمین افزایش می‌باید ولی هیچ وقت تمام طیف را پوشش نمی‌دهد.



شكل 2-5 مدل مرتبه 5



شكل 2-5 مدل مرتبه 51



شكل 5-4 مدل مرتبه 101

6 پیوست 1 : کدهای متلب

6.1 حداقل مربعات

```
N = numel(y) ;  
Parameters_in_den=4  
Parameters_in_num=4  
Nv=Parameters_in_num+Parameters_in_den  
phi=[];  
for i=(max(Parameters_in_num,Parameters_in_den)+1):N  
    phi(i,:)= [(y(i-1:-1:i-Parameters_in_den))' , (u(i-1:-1:i-  
Parameters_in_num))'];  
end  
theta_hat=((phi'*phi)^(-1))*(phi'*y)
```

6.2 حداقل مربعات بازگشتی

```
N = numel(y) ;  
%choose number of parameters  
Parameters_in_den=4  
Parameters_in_num=4  
Nv=Parameters_in_num+Parameters_in_den  
theta(:,1:Nv) = zeros(Nv , Nv) ;  
P = 1e12*eye(Nv) ;  
phi=[];  
Eror=zeros(1,N);  
for i = (max(Parameters_in_num,Parameters_in_den)+1):N  
    phi(:,i)= [(y(i-1:-1:i-Parameters_in_den))' , (u(i-1:-1:i-Parameters_in_num))']';  
    K = P*phi(:,i)*(1+phi(:,i)']*P*phi(:,i))^(-1) ;  
    P = (eye(Nv) - K*phi(:,i)')*P ;  
    theta(:,i) = theta(:,i-1) + K*(y(i) - phi(:,i)']*theta(:,i-1));  
    Eror(i)=(Eror(i-1)+(y(i)-phi(:,i)']*theta(:,i))^2);  
end
```

6.3 فیلتر کالمن

```
N = numel(y) ;  
% %-----%  
%choose number of parameters  
Parameters_in_den=3  
Parameters_in_num=3  
% %-----%  
Nv=Parameters_in_num+Parameters_in_den;  
p_KF(1:Nv,1:Nv,1:N)=zeros(Nv,Nv,N);  
%intitial Conditions  
theta_hat_KF(1:Nv,1:N)=zeros(Nv,N);  
K_KF(1:Nv,1:N)=zeros(Nv,N);  
var_e=0.05;
```

```

p_KF(1:Nv,1:Nv,1)=1e5*eye(Nv);p_KF(1:Nv,1:Nv,2)=p_KF(1:Nv,1:Nv,1);p_KF(1:Nv,1:Nv,3)=p_KF(1:Nv,
1:Nv,2);
for i=(max(Parameters_in_num,Parameters_in_den)+1):N
    phi_KF(:,i)=[(y(i-1:-1:i-Parameters_in_den))',(u(i-1:-1:i-Parameters_in_num))']';
    K_KF(:,i) =p_KF(:,:,i-1)*phi_KF(:,i)*(1+phi_KF(:,i)'*p_KF(:,:,i-1)*phi_KF(:,i))^-1 ;
    p_KF(:,:,i)=p_KF(:,:,i-1)-p_KF(:,:,i-1)*phi_KF(:,i)*(1+phi_KF(:,i)'*p_KF(:,:,i-
1)*phi_KF(:,i))^-1*phi_KF(:,i)'*p_KF(:,:,i-1)+var_e;
    theta_hat_KF(:,i)=theta_hat_KF(:,i-1)+K_KF(:,i)*(y(i)-phi_KF(:,i)']*theta_hat_KF(:,i-1));
end

```