



به نام خدا

استاد درس: دکتر آیتی

شبیه‌سازی اول درس کنترل تطبیقی

دانشجو: یاسین ریاضی

۸۱۰۶۰۰۲۰۳

بهار ۱۴۰۱

فهرست

۱	جایابی قطب حداقل درجه با فیدبک دینامیکی	۱
۲	جایابی قطب بدون حذف صفر و قطب ، با در نظر گرفتن مکان قطبهای مطلوب	۱ - ۱
۳	Desigered System	
۴	MDPP with no zero cancelation	
۵	جایابی قطب با حذف صفر و قطب، با در نظر گرفتن مکان قطبهای مطلوب	۱ - ۲
۶	Desigered System	
۷	MDPP with zero cancelation	
۸	پاسخهای بدست آمده را با یکدیگر مقایسه نموده و تحلیل خود را بیان کنید.	۳ - ۱
۹	STR با جایابی قطب حداقل درجه برای سیستم مینیمم فاز	۲
۹	یک کنترل کننده STR به روش غیر مستقیم، بدون حذف صفر و قطب، طراحی کنید.	۲ - ۱ - ۱
۱۰	Assumption	
۱۰	RLS	۲ - ۲
۱۱	MDPP WNZC	
۱۷	یک کنترل کننده STR به روش مستقیم، بدون حذف صفر و قطب، طراحی کنید.	۲ - ۲ - ۲
۱۷	generate Data	
۱۸	Assumption	
۱۸	RLS	
۱۹	RLS System	
۲۳	generate Data	
۲۴	Assumption	
۲۴	RLS	
۲۵	RLS System	
۳۰	یک کنترل کننده STR به روش غیر مستقیم، با حذف صفر و قطب، طراحی کنید.	۳ - ۲ - ۲
۳۰	generate Data	
۳۱	Assumption	
۳۲	RLS	
۳۳	General Input v.s. Output	

۴۰.....	یک کنترل کننده STR به روش مستقیم، با حذف صفر و قطب، طراحی کنید.	۴ - ۲ - ۲
۴۰.....	generate Data	
۴۱.....	Assumption	
۴۲.....	initial parameters	
۴۲.....	main loop Direct Str algorithm	
۴۳.....	plot results	
۴۵.....	General Input v.s. Output	
۵۳.....	اثر Under parameter و Over parameter بودن مدل را در ۴ کنترل کننده فوق بررسی کنید.	۳ - ۲
۵۴.....	در این قسمت اگر شبیه سازی را برای زمانهای طولانی انجام دهید آیا ورودی کنترلی بزرگ میشود؟ بحث کنید.	۲ - ۴
۵۷.....	STR با جایابی قطب حداقل درجه برای سیستم نامینیمم فاز	۳
۵۸.....	MDPP STR (Direct)	
۵۹.....	main loop	
۵۹.....	plot results	
۶۲.....	با جایابی قطب برای سیستم پیوسته.	۴
۶۵.....	پیوست ۱ : کدهای متلب	۵
۶۵.....	دیوفانتین	۱ - ۵
۶۵.....	حداقل مربعات بازگشتی	۲ - ۵

شکل ۱-۱ :	پاسخ پله سیستم پیوسته و گسسته	۱
شکل ۲-۱ :	ورودی مرجع و خروجی پلت برای حالت بدون حذف صفر	۴
شکل ۳-۱ :	سیگنال کنترلی برای حالت بدون حذف صفر	۵
شکل ۴-۱ :	ورودی مرجع و خروجی پلت برای سیتم با حذف صفر	۷
شکل ۵-۱ :	سیگنال کنترلی برای سیتم با حذف صفر	۷
شکل ۱-۲ :	خروجی سیستم بدون کنترل کننده	۱۰
شکل ۲-۲ :	جایاب قطب غیر مستقیم-بدون نویز-ورودی مرجع و خروجی	۱۲
شکل ۳-۲ :		۱۲
شکل ۴-۲ :		۱۳
شکل ۵-۲ :		۱۳
شکل ۶-۲ :	ورودی مرجع	۱۴
شکل ۷-۲ :	جایاب قطب غیر مستقیم-با نویز رنگی-ورودی مرجع و خروجی	۱۵
شکل ۸-۲ :	جایاب قطب غیر مستقیم-با نویز رنگی-سیگنال کنترل کننده	۱۵
شکل ۹-۲ :	جایاب قطب غیر مستقیم-با نویز رنگی-همگرایی پارامترهای کنترل کننده	۱۶
شکل ۱۰-۲ :	ورودی مرجع برای حالت بدون نویز STR مستقیم	۱۷
شکل ۱۱-۲ :	ورودی مرجع و خروجی برای STR مستقیم و بدون نویز	۲۰
شکل ۱۲-۲ :	اتلاش کنترلی STR مستقیم و بدون نویز	۲۱
شکل ۱۳-۲ :	پارامترهای همگرا شده	۲۲
شکل ۱۴-۲ :	ورودی با نویز برای STR مستقیم	۲۳
شکل ۱۵-۲ :	خروجی مستقیم با نویز رنگی بدون حذف صفر	۲۶
شکل ۱۶-۲ :	خروجی کنترلی مستقیم با نویز رنگی بدون حذف صفر	۲۷
شکل ۱۷-۲ :	همگرایی مستقیم با نویز رنگی بدون حذف صفر	۲۸
شکل ۱-۴ :	شکل کلی شناسایی سیستم به صورت پیوسته در سیمولینک	۶۲
شکل ۲-۴ :	قسمت شناسایی سیستم	۶۳
شکل ۳-۴ :	خروجی سیستم و ورودی مرجع	۶۳
شکل ۴-۴ :	همگرایی حداقل مربعات بازگشتی پیوسته	۶۴
شکل ۵-۴ :	پاسخهای معادله دیوفانتین	۶۴

۱ جایابی قطب حداقل درجه با فیدبک دینامیکی

سیستم دینامیکی خطی

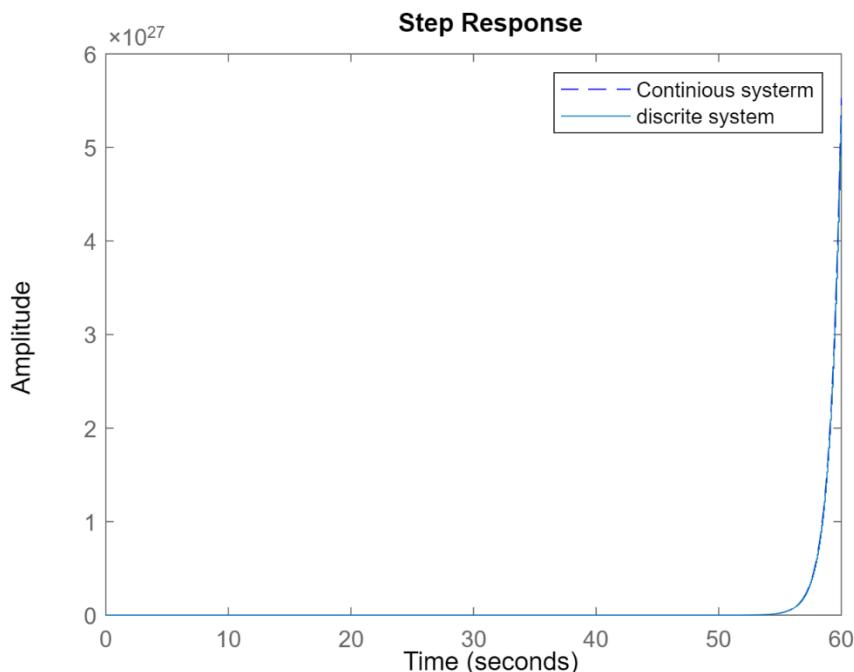
$$G = \frac{0.1(s + 1.6)(s + 3.4)}{(0.7s + 1.4)(0.3s + 1.6)(s - 1.1)} = \frac{100s^2 + 500s + 544}{210s^3 + 2089s^2 + 3848s - 704}$$

به یکی از روش‌های گسسته سازی و با انتخاب زمان نمونه برداری مناسب، مدل زمان گسسته آن را بدست آورید.

$$T_s = 0.707$$

```
sys_dis =
        0.2869 z^2 - 0.4819 z + 0.2015
-----
z^3 - 2.434 z^2 + 1.921 z - 0.4949
```

Sample time: 0.70712 seconds
Discrete-time transfer function.



شکل ۱-۱: پاسخ پله سیستم پیوسته و گسسته

راهنمایی: برای سیستم ناپایدار انتخاب شده، قطب‌های مطلوب به گونه‌ای انتخاب شوند که سیستم دارای زمان نشت ۳ ثانیه و بالازدگی ۱۰٪ باشد و قطب دیگر نسبت به قطب‌های غالب حداقل ۳.۵ برابر از محور موهومی دورتر باشد.

۱ - جایابی قطب بدون حذف صفر و قطب ، با در نظر گرفتن مکان قطبهای مطلوب

$$B = B^+ B^-$$

چون حذف صفر نداریم

$$B = 1 \cdot B^- = 0.02869 z^3 - 0.04819 z + 0.02010$$

$$\deg A_o = \deg A - \deg B^+ - 1 = 3 - 0 - 1 = 2$$

$$A_o = \text{constant}$$

$$R = R' B^+$$

$$\deg(AR') = \deg(A \cdot A_m) = 5 \xrightarrow{\text{yields}} \deg R' = 2$$

باید همواره مونیک باشد

Data: Polynomials A, B .

Specifications: Polynomials A_m, B_m , and A_o .

Compatibility Conditions:

$$\deg A_m = \deg A = 3$$

$$\deg B_m = \deg B = 2$$

$$\deg A_o = \deg A - \deg B^+ - 1 = 3 - 0 - 1 = 2$$

$$B_m = B^- B'_m =$$

ابتدا با استفاده از سیمولینک تابع تبدیل را شبیه سازی کرده و دو ایراد را متوجه شدم.

۱. به اشتباه در قسمت ضرایب تابع تبدیل محل صفرهای گسسته مدنظر خود را قرار می‌دادم و نتیجه اشتباهی به دست می‌آوردم.
۲. یکی از قطب‌های گسسته را در منفی قرار داده بودم که اورشوت بسیار زیادی را برای حالت بدون حذف صفر به وجود می‌آورد ولی قسمتی که صفر را حذف می‌کردیم به قطب منفی حساس نبوده و اورشوت اصلاً به وجود نمی‌آمد.

:کد

```
run('BASIC.m')
sys_discret =
0.017631 (z-0.3611) (z-0.6719)
-----
(z-1.365) (z-0.2212) (z-0.1986)

Sample time: 0.28285 seconds
Discrete-time zero/pole/gain model.
B = 1x3
    0.0176   -0.0182    0.0043
A = 1x4
    1.0000   -1.7848    0.6171   -0.0600
```

Desigered System

```
%run('am mlx');
%A_m=den_discret_desierd%A=den_discret
A_m=poly([0.45 0.60 0.30]);
betaa=sum(A_m)/sum(B);
B_m=B* beta;
B_plus=';
```

betaa = 41.6664

قطب ها در $[0, 0, 45, 60, 30]$ انتخاب شده است و در حالت بدون حذف صفر $B_m = \beta B$ که بنا از دست می آید.

MDPP with no zero cancelation

```
A_o=[1 0 0];
deg A_o = deg A - deg B^+ - 1 = 3 - 1 - 1 = 1
```

```
A_c=conv(A_m,A_o)
A_c = 1x6
    1.0000   -1.3400    0.5965   -0.0882       0       0
[R_prim , S] = Diophantine(A , B , A_c)
```

```
R_prim = 3x1
    1.0000
    -1.1490
    0.2871
```

```
S = 3x1
    90.4018
    -40.3827
    4.0254
```

```
T=conv(betaa,A_o)
T = 1x3
    45.5354      0      0
R=conv(R_prim,B_plus)
```

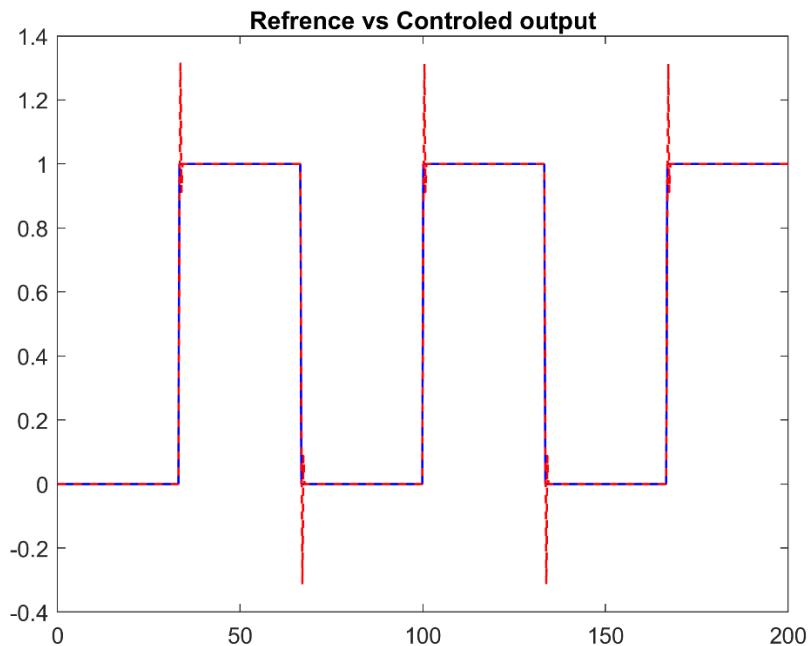
```

R = 3×1
1.0000
-1.1490
0.2871
tfinal=200;
t = 0:T_s:tfinal;
uc=gensig('square' , tfinal/3 , tfinal ,T_s);

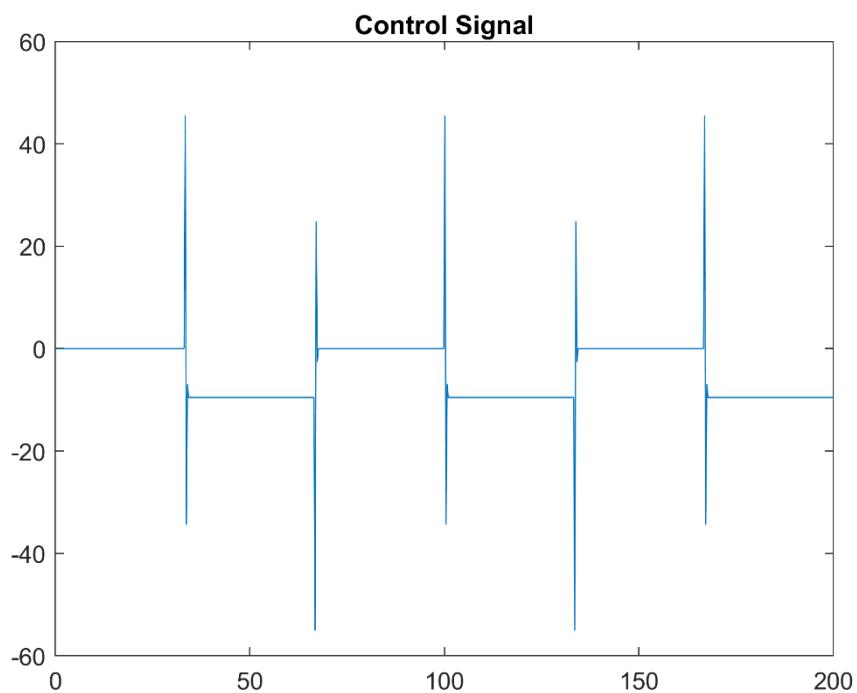
u=zeros(numel(t),1);
y=zeros(numel(t),1);
for i=10:numel(t)
    var1=conv(B,T) ; narv1=numel(var1) ;
    var2=A_c(2:end) ; narv2=numel(A_c(2:end));
    var3=conv(A,T) ; narv3=numel(var3) ;

    y(i)=var1*uc(i:-1:i-narv1+1)-var2*uc(i-1:-1:i-narv2);
    u(i)=var3*uc(i:-1:i-narv3+1)-var2*uc(i-1:-1:i-narv2);
end

```



شکل ۱-۳: ورودی مرجع و خروجی پلت برای حالت بدون حذف صفر



شکل ۱-۳ : سیگنال کنترلی برای حالت بدون حذف صفر

۱ - ۲ - جایابی قطب با حذف صفر و قطب، با در نظر گرفتن مکان قطبهای مطلوب
کد :

```
run('BASIC.m')
sys_discret =
0.017631 (z-0.3611) (z-0.6719)
-----
(z-1.365) (z-0.2212) (z-0.1986)

Sample time: 0.28285 seconds
Discrete-time zero/pole/gain model.

B = 1x3
0.0176   -0.0182   0.0043

A = 1x4
1.0000   -1.7848   0.6171   -0.0600

Titlework='Q12'
Desigered System
A_m=poly([0.2 0.2 0.02]);
```

```

betaaa=sum(A_m);
B_m=betaaa*[1,zeros(1,(numel(A)-numel(B))+1)];
B_plus =B/B(1);
B_minus= B(1);

Deg_A_o=numel(A)-numel(B_plus)-1;
A_o=[1 zeros(1,Deg_A_o)];
A_c_prim=conv(A_m,A_o);
A_c=conv(A_c_prim,B_plus);

```

MDPP with zero cancellation

```
[R,S] = Diophantine(A,B,A_c)
```

```
R = 3x1
```

```
1.0000
-1.0330
0.2426
```

```
S = 3x1
```

```
77.4135
-32.2771
3.3570
```

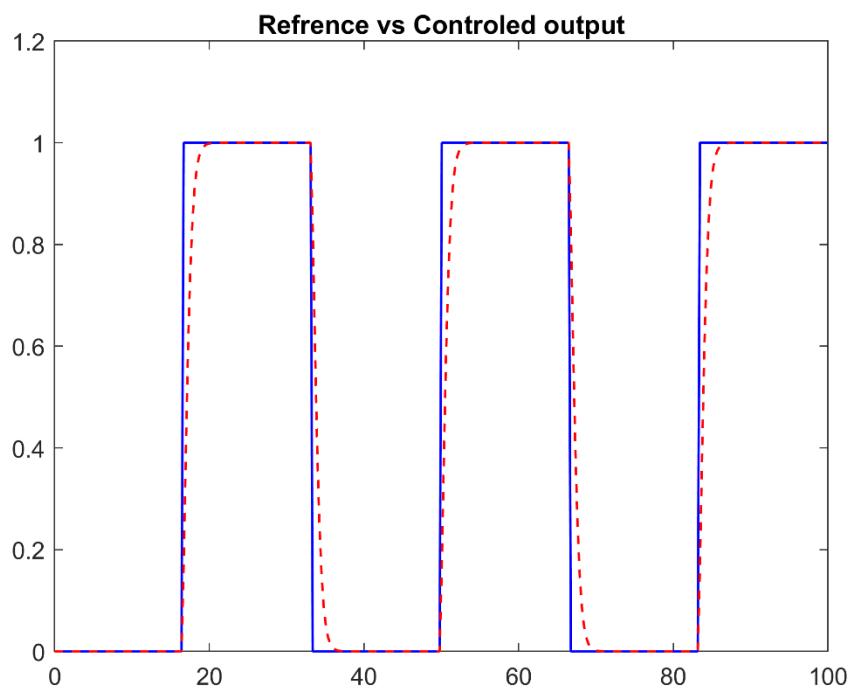
```
T=conv(A_o,B_m_prim);
```

```

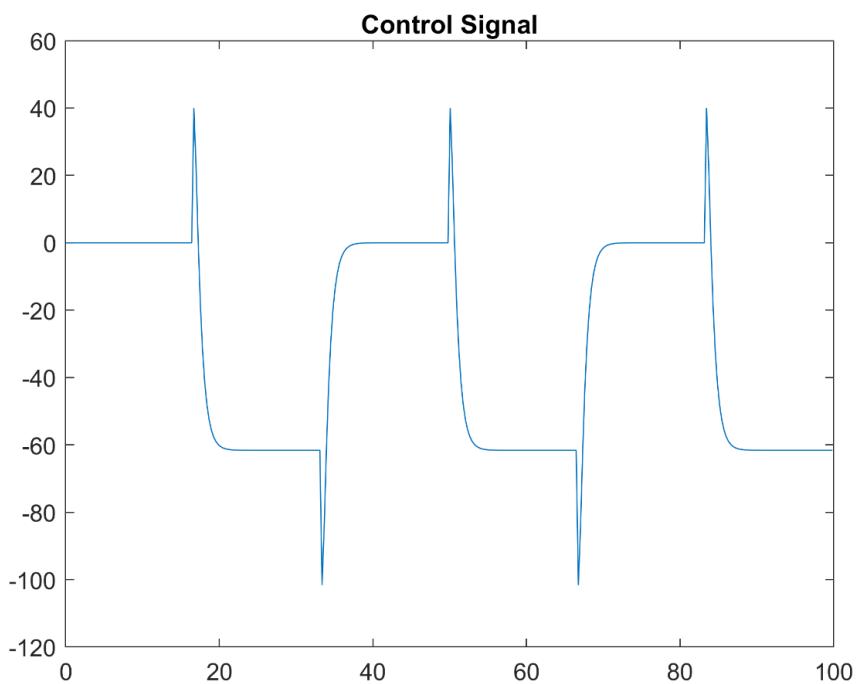
tfinal=100;
t = 0:T_s:tfinal;
uc=(gensig('square' , tfinal/3 , tfinal ,T_s));
u=zeros(numel(t),1);
y=zeros(numel(t),1);

var1=conv(B,T);      narv1=numel(var1);
var2=A_c(2:end);    narv2=numel(A_c(2:end));
var3=conv(A,T);    narv3=numel(var3);
for i=10:numel(t)
    y(i)=var1*uc(i:-1:i-narv1+1)-var2*y(i-1:-1:i-narv2);
    u(i)=var3*uc(i:-1:i-narv3+1)-var2*u(i-1:-1:i-narv2);
end

```



شکل ۱-۴ ورودی مرجع و خروجی پلنت برای سیستم با حذف صفر



شکل ۱-۵ سیگنال کنترلی برای سیستم با حذف صفر

۱ - ۳ پاسخهای بدست آمده را با یکدیگر مقایسه نموده و تحلیل خود را بیان کنید.
در استفاده از MDPP بدون حذف صفر سیگنال کنترلی بزرگ تر و کنترل اورشوت دشوار تر است در صورتی که با استفاده از MDPP با حذف صفر سیگنال کنترلی کوچکتر و

۲ با جابابی قطب حداقل درجه براى سیستم مینیمم فاز STR

الف) کنترل کننده های ذکر شده زیر a, b, c, d را طراحی کنید و عملکرد آنها را در ردیابی سیگنال ورودی مرجع مربعی در حالت بدون نویز و در حضور نویز رنگی بررسی کنید. همچنین روند همگرایی پارامترهای کنترل کننده (ضایب R و S) را رسم کنید. آیا به مقادیر مطلوب همگرا می شوند؟

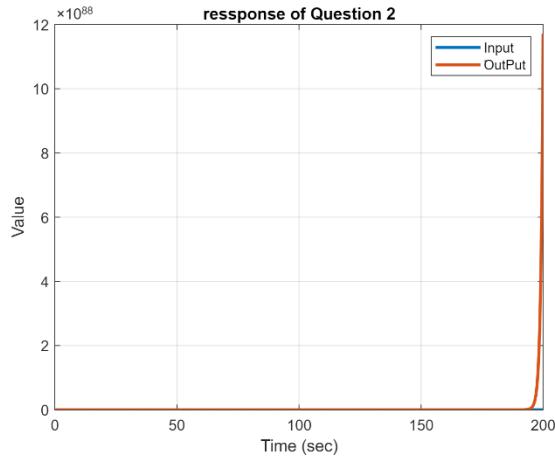
۱ - ۱ - ۲ یک کنترل کننده STR به روش غیر مستقیم، بدون حذف صفر و قطب، طراحی کنید.

```
run('BASIC.m')
sys_discret =
0.017631 (z-0.3611) (z-0.6719)
-----
(z-1.365) (z-0.2212) (z-0.1986)

Sample time: 0.28285 seconds
Discrete-time zero/pole/gain model.
B = 1x3
    0.0176   -0.0182    0.0043
A = 1x4
    1.0000   -1.7848    0.6171   -0.0600
Titlework='Q21'

tfinal=200;
t = 0:T_s:tfinal;

General Input+white Noise
u = gensig('square' , tfinal/10 , tfinal ,T_s);
y = lsim(sys_discret ,u ,t);
plot(t,u ,t , y , 'LineWidth',2) ;
xlabel('Time (sec)');
ylabel('Value') ;
title('ressponse of Question 2') ;
grid on
legend('Input' , 'OutPut') ;
```



شکل ۲-۱ خروجی سیستم بدون کنترل کننده

Assumption

```
%choose number of parameters
Parameters_in_den=3
Parameters_in_num=3
degree_cancelled_zero=0
A_m=poly([0.1 0.2 0.02]);
```

تمامی فرضیات در ۴ خط بالا انجام می‌شود به این صورت که تعداد پارامترهای صورت و مخرج تعداد صفرهای باقی مانده (به صورت ۰ و ۱) و صفرهای حلقه بسته مطلوب.

```
deg_A=Parameters_in_num+1;
deg_B=Parameters_in_den ;
Nv=Parameters_in_num+Parameters_in_den;
deg_B_plus =degree_cancelled_zero+1;
Deg_B_minus=deg_B-deg_B_plus;
Deg_A_o=deg_A-deg_B_plus-1;
A_o=[1 zeros(1,Deg_A_o)];
A_c_prim=conv(A_m,A_o);
```

شرط‌های لازم برای جایاب قطب در بالا تعریف می‌شوند.

۲ - ۲ RLS

```
theta(1:Nv,1:40) = ones(Nv,40) ;
P = 1e12*eye(Nv) ;
phi=[];
N = numel(y) ;
for i = (deg_A+deg_B):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));

A=[1 -theta(1:Parameters_in_num ,end)'];
B=theta((Parameters_in_num+1):end,end)';

```

در قسمت بالا به روش حداقل مربعات بازگشتی پارامترهای سیستم شناسایی می‌شوند.

```

if degre_canceled_zero==0
    B_plus=1;
    B_minus=B;
    betaa=sum(A_m)/sum(B);
    B_m=betaa*B;
end

```

شرط عدم حذف صفرها در اینجا بررسی می‌شوند و تمامی صفرهای شناسایی شده در صفرهای مطلوب قرار داده می‌شوند.

MDPP WNZC

```

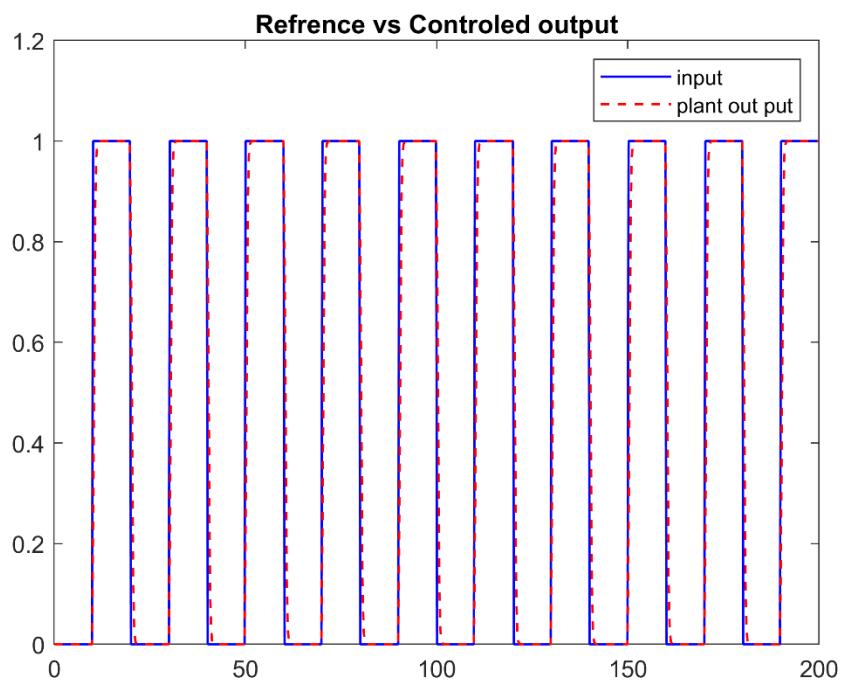
[R_prim,Si] = Diophantine(A,B ,A_c_prim);
S(i,:)=Si;
T(i,:)=conv(betaa,A_o);
R(i,:)=conv(R_prim,B_plus);

var1=conv(B,T(i,:));      narv1=numel(var1)      ;
var2=A_c(2:end) ;      narv2=numel(A_c(2:end));
var3=conv(A,T(i,:)) ;      narv3=numel(var3)      ;

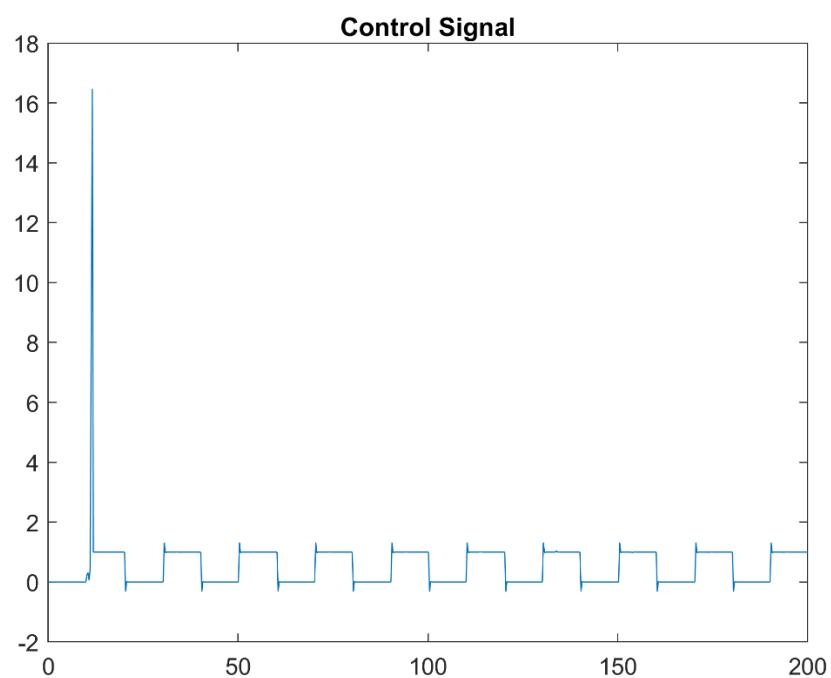
y(i)      =var1*u(i:-1:i-narv1+1)-var2*y(i-1:-1:i-narv2);
u_cont(i)=var3*u(i:-1:i-narv3+1)-var2*u(i-1:-1:i-narv2);
end

```

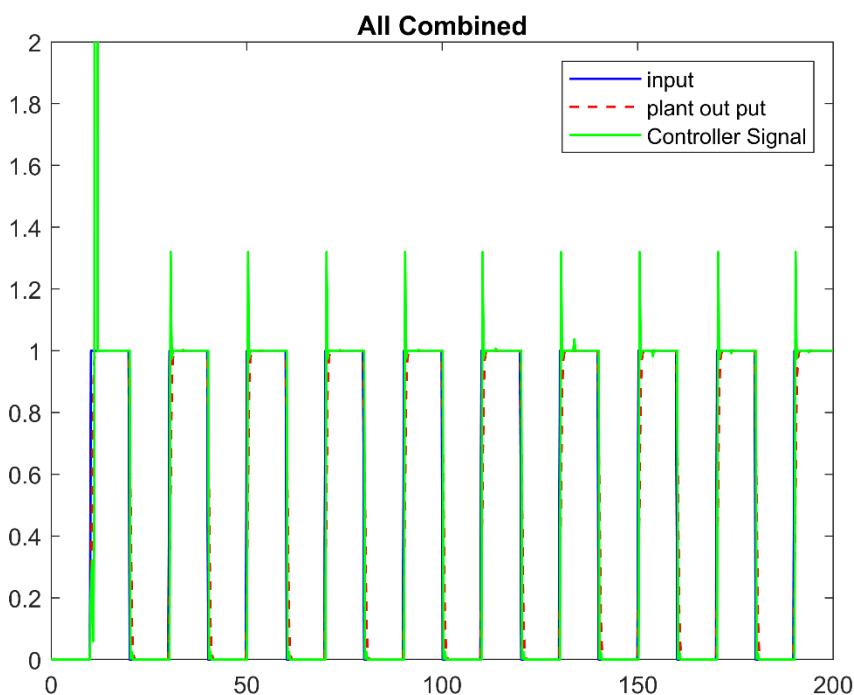
در نهایت از معادله ۱-۳ کتاب خروجی کنترل کننده و سیستم محاسبه می‌شوند.



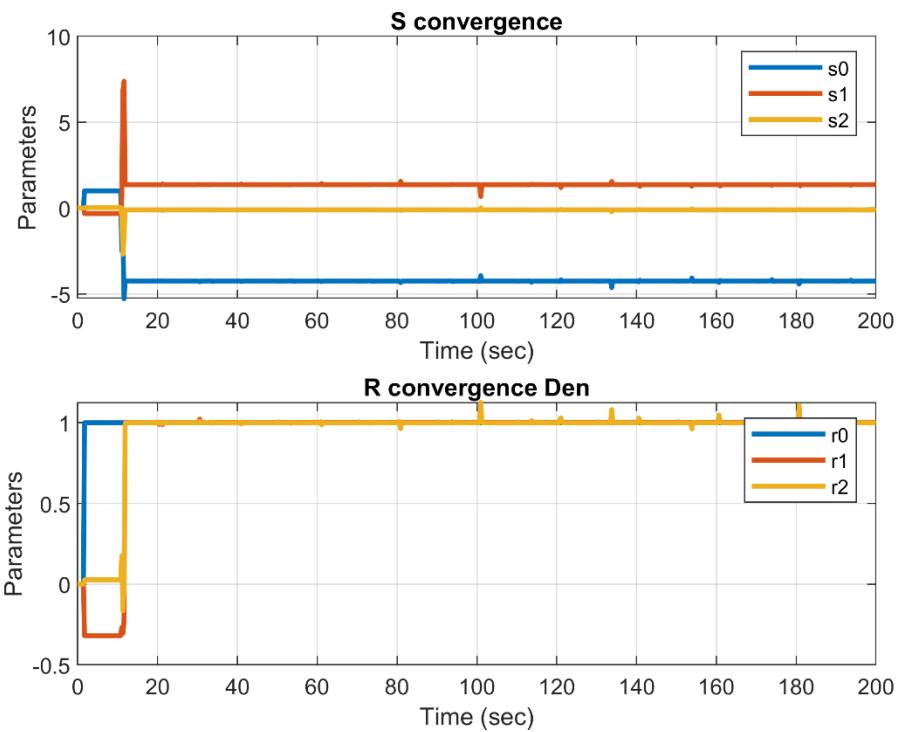
شکل ۲-۲ جایاب قطب غیر مستقیم-بدون نویز-ورودی مرجع و خروجی



شکل ۳-۲

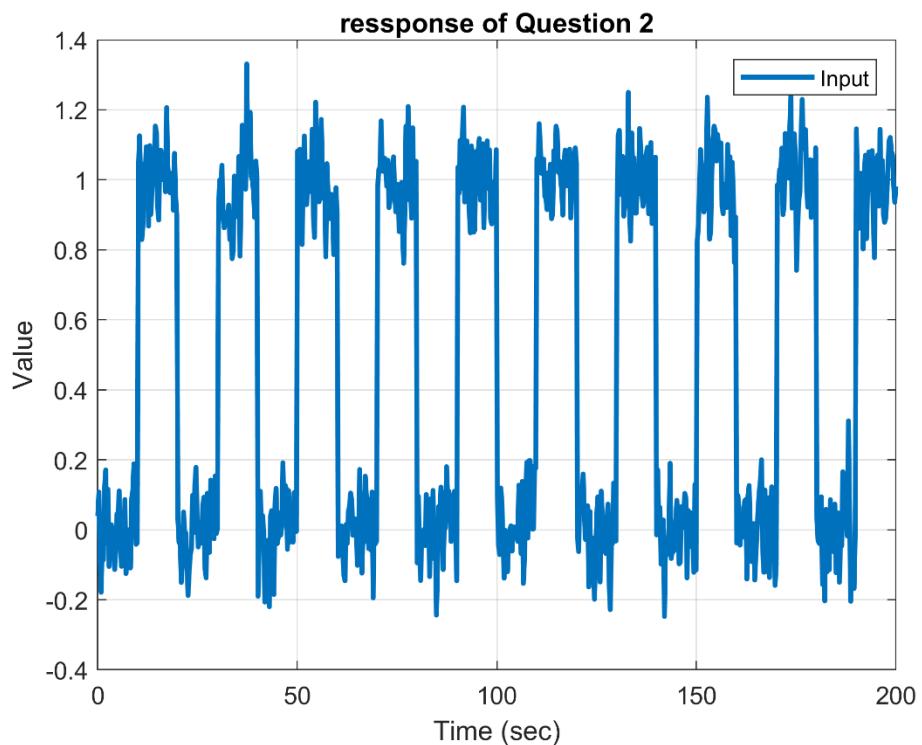


٢-٢

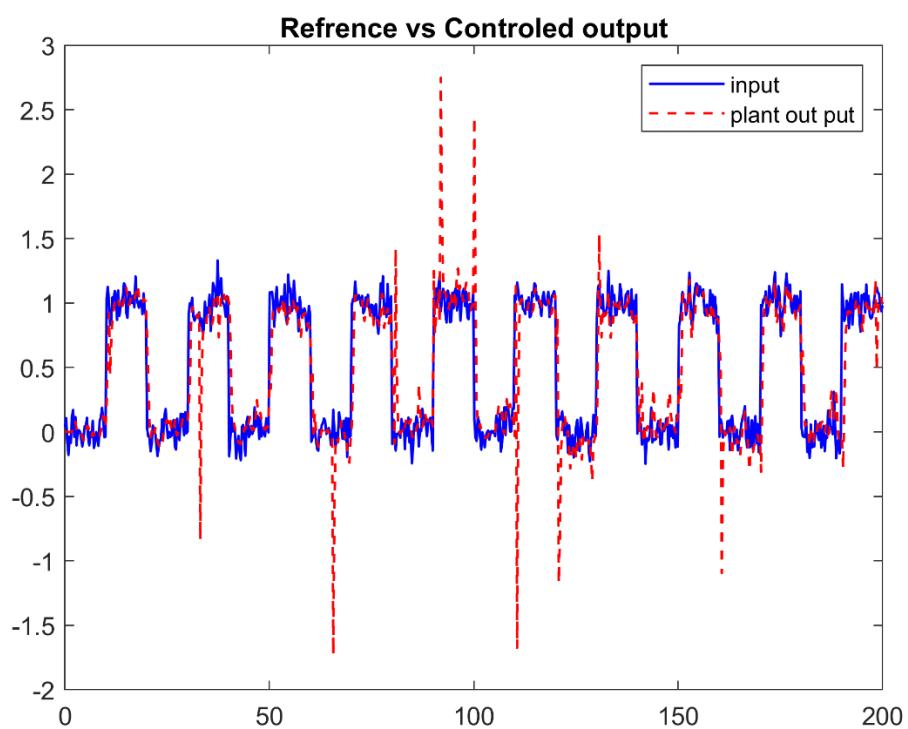


٣-٢

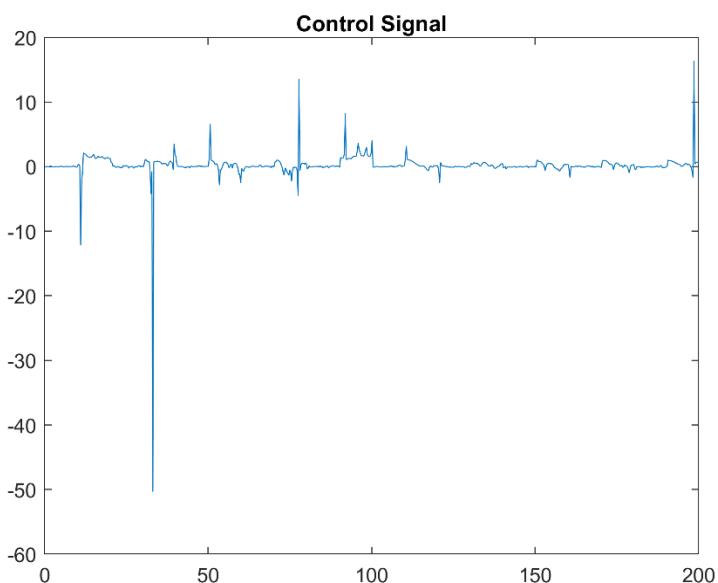
با وارد کردن نويز رنگي رفتار كنترکننده بسيار ناپايدار می شود و ديگر در زمان های طولاني حالت متناوب ندارد.



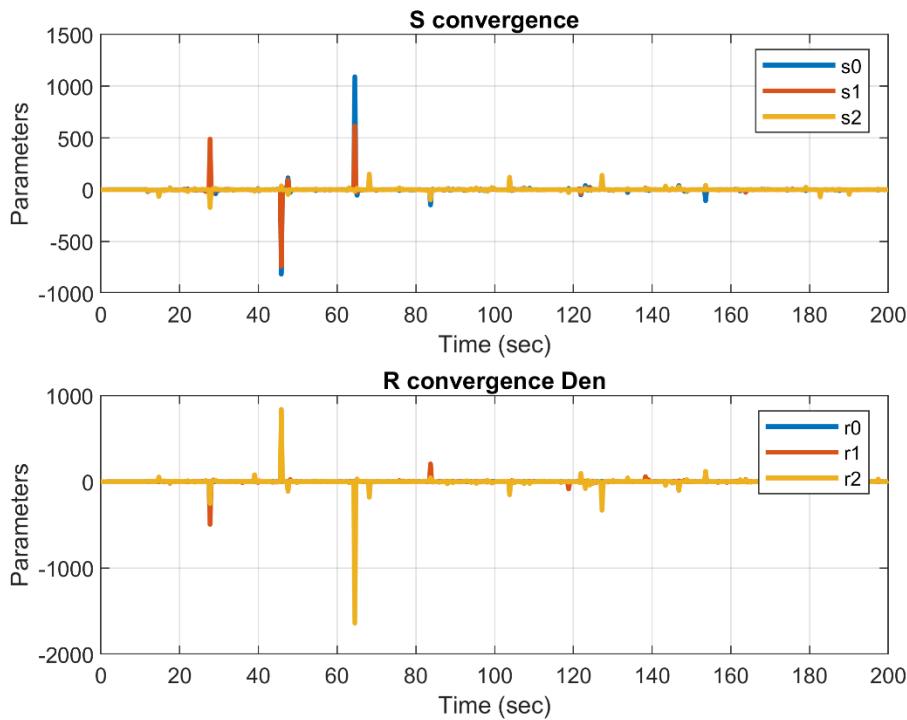
شکل ۲-۶ ورودی مرجع



شکل ۲-۲ جایاب قطب غیر مستقیم-با نویز رنگی-ورودی مرجع و خروجی



شکل ۲-۱ جایاب قطب غیر مستقیم-با نویز رنگی-سیگنال کنترل کننده



شکل ۲-۹ جایاب قطب غیر مستقیم-بدون نویز-رندگی-همگرای پارامترهای کنترل کننده

نتیجه گیری ۲-۱-۲

جایاب قطب غیر مستقیم-بدون نویز-ورودی مرجع و خروجی به خوبی کنترل را انجام می‌دهد و پارامترها نیز با سرعت مناسبی به مقدار کمی همگرا می‌شوند اما با حضور نویز رندگی رفتار پریودیک دیگر در سیگنال کنترلی دیده نمی‌شود و تلاش کنترلی زیادی برای کنترل سیستم به کار می‌رود و منشا این مشکلات در تغییرات ناگهانی در مقدار پارامترهای کنترلی است.

۲ - ۲ - ۲ یک کنترل کننده STR به روش مستقیم، بدون حذف صفر و قطب، طراحی کنید.

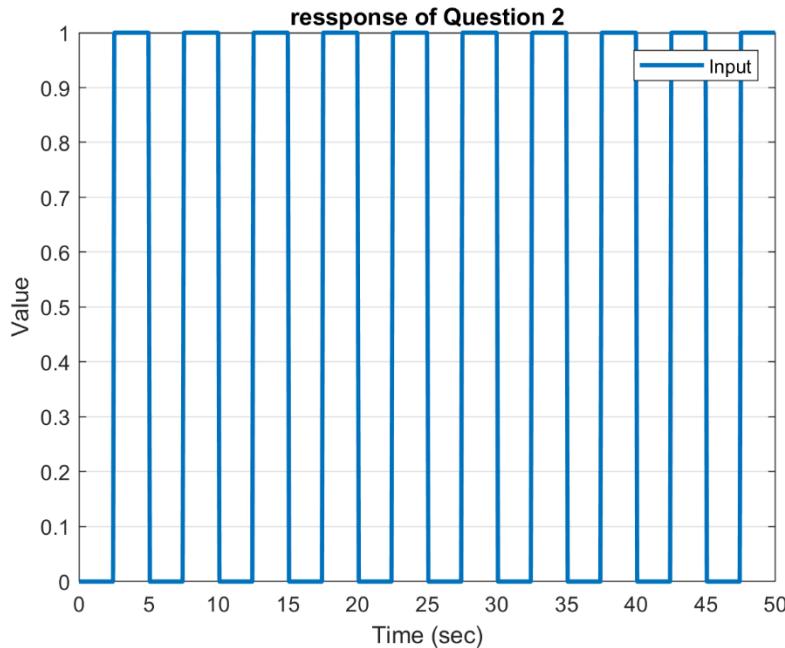
```
run('BASIC.m')
sys_discret =
0.0076672 (z-0.7239) (z-0.8621)
-----
(z-1.109) (z-0.6048) (z-0.5835)

Sample time: 0.094282 seconds
Discrete-time zero/pole/gain model.
```

generate Data

```
Question_mark='Q212';
[uc,t,Status,tfinal]=Datagen(0,T_s,200);
Status = '_No NOISE_'
Titlework=[Question_mark,Status]
Titlework = 'Q212_No NOISE_'

y = lsim(sys_discret ,uc ,t);
plot(t,uc , 'LineWidth',2) ;
xlabel('Time (sec)');
ylabel('Value') ;
title('ressponse of Question 2') ;
grid on
legend('Input' , 'OutPut') ;
print(gcf,[Titlework , num2str(plot_counter) ' Refrence .png'],'-dpng',' -r400');
```



شکل ۲-۱۰-۱ ورودی مرجع برای حالت بدون نویز STR مستقیم

```
plot_counter=plot_counter+1;
```

Assumption

با تغییر پارامترهای زیر تعداد متغیرهای شناسایی شونده برای سیستم تغییر می‌کند

```
%choose number of parameters
Parameters_in_den=3
Parameters_in_den = 3
Parameters_in_num=3
Parameters_in_num = 3
A_m=poly([0.8 0.80 0.8]);
B_m=(sum(A_m)/sum(B))*B;
sys_ref=tf(B_m,A_m,T_s)
sys_ref =
0.2101 z^2 - 0.3333 z + 0.1311
-----
z^3 - 2.4 z^2 + 1.92 z - 0.512

Sample time: 0.094282 seconds
Discrete-time transfer function.
y_ref_plant = lsim(sys_discret ,uc ,t)';
y_ref      = lsim(sys_ref , uc , t) ;

deg_A=Parameters_in_num+1;
deg_B=Parameters_in_den ;
B_m=[sum(A_m),zeros(1,(deg_A-deg_B)+1)];

A_o=[1 0 0];

dθ = deg_A-deg_B ;
A_c_prim=conv(A_m,A_o);
deg_A_c_prim=numel(A_c_prim)-1;
L = deg_A_c_prim-dθ;
```

RLS

با تغییر پارامترهای زیر تعداد متغیرهای کنترل کننده تغییر می‌کند.

```
number_R=3
number_R = 3
number_S=number_R;
L = 2 ;
Nv =number_R*2 ;

N = numel(y) ;
% initial parameters
A0 = [1 0 0];
```

```

n = numel(A)-1 ;
m = numel(B)-1 ;
d0 = n-m ;
A0Am = conv(A0 , A_m) ;
Na0am = numel(A0Am)-1 ;

theta_sys=1*ones(6,1);
teta = 10*ones(Nv,1);

u = uc ; % initial effort control
uf = uc ; % initial filtered effort control
yf = 0.1*ones(100 , 1) ; % initial filtered output
ucf= uc ; % initial filtered command signal
P_cont = 1e12*eye(Nv);
P_sys = 1e12*eye(6);
R=zeros(3,N);
S=zeros(3,N);

```

RLS System

```

for i = Nv+1:N
    if i<40
        y(i) = -A(2:end)*y(i-1:-1:i-n)+B*(u(i-1:-1:i-n)) ;

        phi_sys = [(y(i-1:-1:i-3))' , (u(i-1:-1:i-3))'];
        [theta_sys,P_sys]=RLS(theta_sys,phi_sys,P_sys,6,y(i));
        Aest=[1 -theta_sys(1:3 ,end)'];
        Best=theta_sys(4:6,end)';
    else
        y(i) = -Aest(2:end)*y(i-1:-1:i-(numel(Aest)-1))+Best*(u(i-1:-1:i-
numel(Best))) ;

        Y = y(i);%-y_ref(i);
        phi_sys = [(y(i-1:-1:i-Parameters_in_num))' , (u(i-1:-1:i-
Parameters_in_den))'];
        [theta_sys,P_sys]=RLS(theta_sys,phi_sys,P_sys,6,Y);
        Aest=[1 -theta_sys(1:3 ,end)'];
        Best=theta_sys(4:6,end');

        phi = [uf(i-d0:-1:i-number_R)' , yf(i-d0:-1:i-number_S)'];
        [teta,P_cont] =RLS(teta,phi,P_cont,Nv,Y);
        Rst = teta(1:Nv/2)' ;
        Sst = teta(Nv/2 +1:Nv)' ;
        R(1:Nv/2,i)=teta(1:Nv/2) ;
        S(1:Nv/2,i)=teta(Nv/2 +1:Nv);
        t0=sum(A_m)/sum(Best) ;
        u(i) = (-Rst(2:end)*u(i-1:-1:i-(numel(Rst(2:end))))...
            +t0*uc(i)...
            -Sst*y(i:-1:i-(numel(Sst)-1)))/Rst(1) ;
    end
end

```

```

        uf(i) = -A0Am(2:end)* uf(i-1:-1:i-Na0am) +Best*u(i:-1:i-
numel(Best)+1) ;
        yf(i) = -A0Am(2:end)* yf(i-1:-1:i-Na0am) +Best*y(i:-1:i-
numel(Best)+1) ;
        ucf(i) = -A0Am(2:end)*ucf(i-1:-1:i-Na0am) +Best*uc(i:-1:i-
numel(Best)+1) ;
    end
end

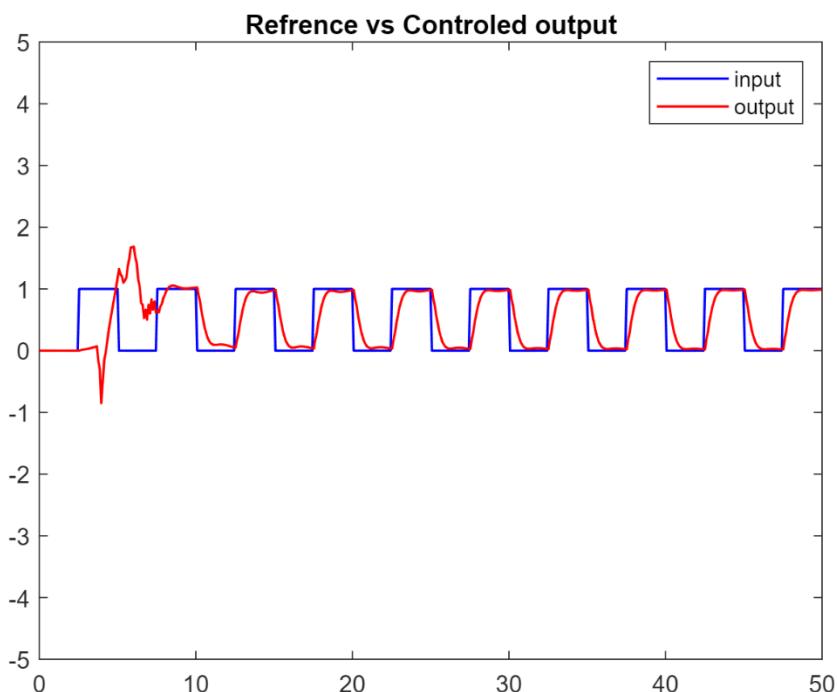
```

General Input v.s. Output

```

plot(t,uc,'b',t,y,'r','LineWidth',1)
title('Refrence vs Controled output')
legend('input','output')
xlim('auto')
ylim([-5 5])
print(gcf,[Titlework , num2str(plot_counter) ' Refrence vs Controled
output.png'], '-dpng', '-r400');

```



شکل ۱۱-۱۱ ورودی مرجع و خروجی برای STR مستقیم و بدون نویز

```

plot_counter=plot_counter+1;

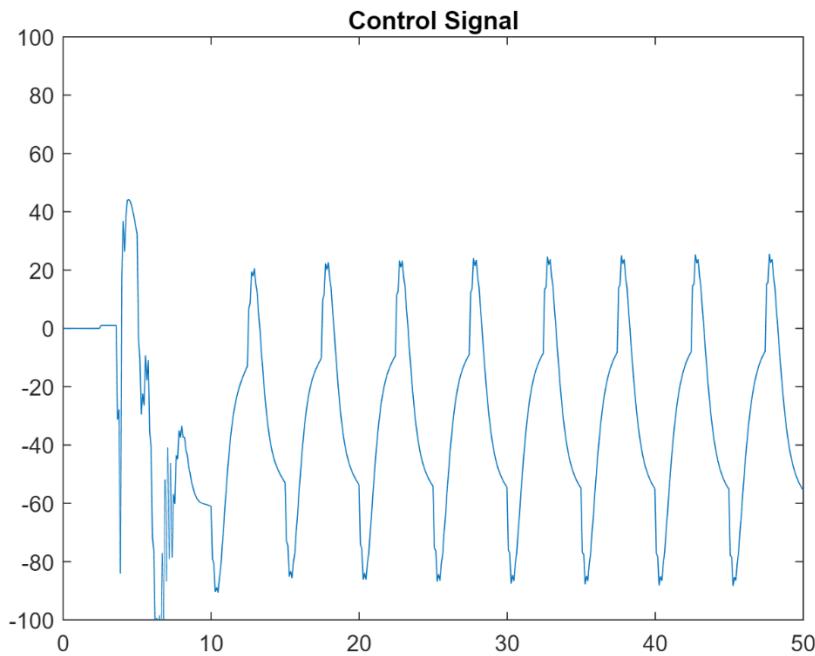
plot(t,u)
title('Control Signal')

```

```

xlim('auto')
ylim([-100 100])
print(gcf,[Titlework , num2str(plot_counter) ' Control Signal.png'], '-dpng', '-r400');

```



شکل ۲-۲ اتلاش کنترلی STR مستقیم و بدون نویز

```
plot_counter=plot_counter+1;
```

```

RLS Convergence of R UND S
subplot(2,1,1)
for i=1:number_R
    legend_names{i} = ['s' num2str(i)-1 ''];
end
plot(t ,S(:,:,1), 'LineWidth' , 2) ;
legend(legend_names)
xlabel('Time (sec)') ;
ylabel('Parameters') ;
title('S convergence') ;
grid on

```

```

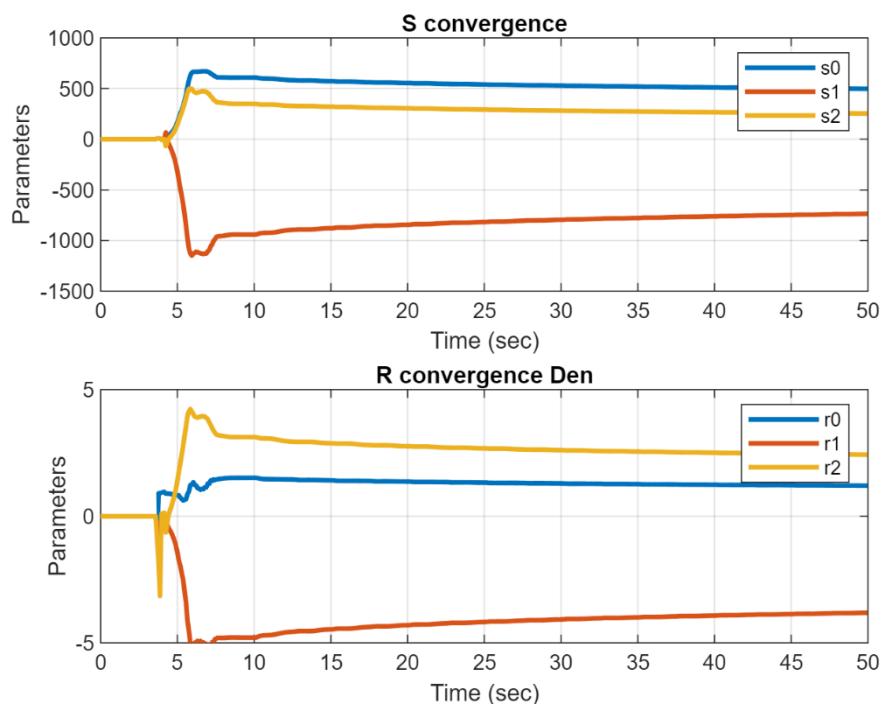
subplot(2,1,2)
for i=1:number_R
    legend_names{i} = ['r' num2str(i)-1 ''];

```

```

end
plot(t ,R(:,:,1), 'LineWidth' , 2) ;
legend(legend_names)
xlabel('Time (sec)' );
ylabel('Parameters' );
title('R convergence Den' );
xlim('auto')
ylim([-5 5])
grid on
print(gcf,[Titlework , num2str(plot_counter) ' RLS Convegence.png'], '-dpng', '-r400');

```



شکل ۲-۱۳ پارامترهای همگرا شده

```

function [teta,P_sys]=RLS(teta,phi,P_sys,Nv,Y)
K = P_sys*phi*(1+phi'*P_sys*phi)^(-1) ;
P_sys = (eye(Nv) - K*phi')*P_sys ;
teta = teta + K*(Y - phi'*teta ) ;
end

```

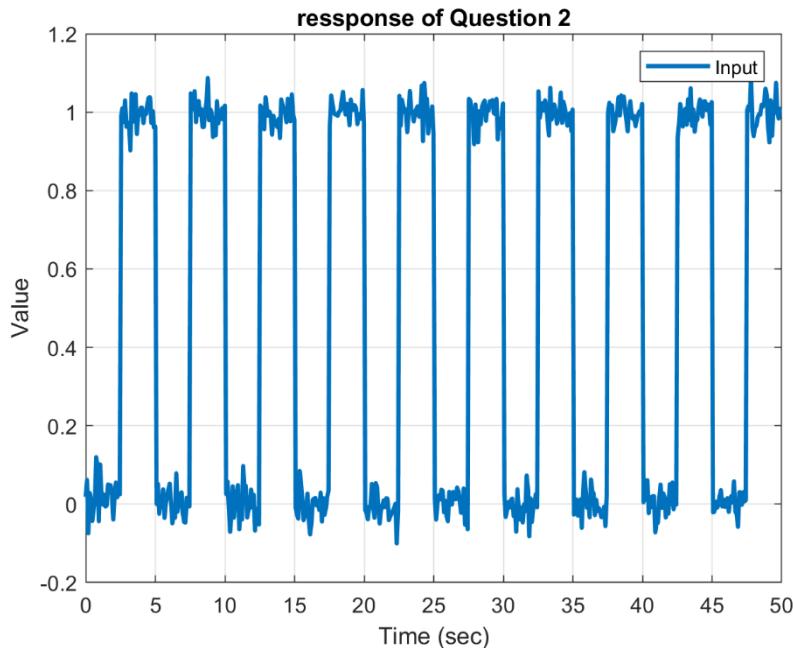
برای حالت با نویز رنگی هم کدها مشابه اند فقط قسمت هایلات شده تغییر کرده است.

```
run('BASIC.m')
```

generate Data

```
Question_mark='Q212';
[uc,t,Status,tfinal]=Datagen(1,T_s,200);
Status = '_white NOISE_'
Titlework=[Question_mark,Status]
Titlework = 'Q212_white NOISE_'

y = lsim(sys_discret ,uc ,t);
plot(t,uc , 'LineWidth',2) ;
xlabel('Time (sec)') ;
ylabel('Value') ;
title('ressponse of Question 2') ;
grid on
legend('Input' , 'OutPut') ;
print(gcf,[Titlework , num2str(plot_counter) ' Refrence .png'], '-dpng',' -r400');
```



شکل ۲-۱۴- ورودی با نویز برای STR مستقیم

```
plot_counter=plot_counter+1;
```

Assumption

```
%choose number of parameters
Parameters_in_den=3
Parameters_in_den = 3
Parameters_in_num=3
Parameters_in_num = 3
A_m=poly([0.8 0.80 0.8]);
B_m=(sum(A_m)/sum(B))*B;
sys_ref=tf(B_m,A_m,T_s)
sys_ref =
0.2101 z^2 - 0.3333 z + 0.1311
-----
z^3 - 2.4 z^2 + 1.92 z - 0.512

Sample time: 0.094282 seconds
Discrete-time transfer function.
y_ref_plant = lsim(sys_discret ,uc ,t)';
y_ref      = lsim(sys_ref , uc , t) ;

deg_A=Parameters_in_num+1;
deg_B=Parameters_in_den ;
B_m=[sum(A_m),zeros(1,(deg_A-deg_B)+1)];

A_o=[1 0 0];

d0 = deg_A-deg_B ;
A_c_prim=conv(A_m,A_o);
deg_A_c_prim=numel(A_c_prim)-1;
L = deg_A_c_prim-d0;
```

RLS

```
number_R=3
number_R = 3
number_S=number_R;
L = 2 ;
Nv =number_R*2 ;

N = numel(y) ;
%% initial parameters
A0 = [1 0 0];
n = numel(A)-1 ;
m = numel(B)-1 ;
d0 = n-m ;
A0Am = conv(A0 , A_m) ;
Na0am = numel(A0Am)-1 ;

theta_sys=1*ones(6,1);
teta = 10*ones(Nv,1);
```

```

u = uc ; % initial effort control
uf = uc ; % initial filtered effort control
yf = 0.1*ones(100 , 1) ; % initial filtered output
ucf= uc ; % initial filtered command signal
P_cont = 1e12*eye(Nv);
P_sys = 1e12*eye(6);
R=zeros(3,N);
S=zeros(3,N);

```

RLS System

```

for i = Nv+1:N
    if i<40
        y(i) = -A(2:end)*y(i-1:-1:i-n)+B*(u(i-1:-1:i-n)) ;

        phi_sys = [(y(i-1:-1:i-3))' , (u(i-1:-1:i-3))']';
        [theta_sys,P_sys]=RLS(theta_sys,phi_sys,P_sys,6,y(i));
        Aest=[1 -theta_sys(1:3 ,end)'];
        Best=theta_sys(4:6,end)';
    else
        y(i) = -Aest(2:end)*y(i-1:-1:i-(numel(Aest)-1))+Best*(u(i-1:-1:i-
numel(Best))) ;

        Y = y(i);%-y_ref(i);
        phi_sys = [(y(i-1:-1:i-Parameters_in_num))' , (u(i-1:-1:i-
Parameters_in_den))']';
        [theta_sys,P_sys]=RLS(theta_sys,phi_sys,P_sys,6,Y);
        Aest=[1 -theta_sys(1:3 ,end)'];
        Best=theta_sys(4:6,end)';

        phi = [uf(i-d0:-1:i-number_R)' , yf(i-d0:-1:i-number_S)']';
        [teta,P_cont] =RLS(teta,phi,P_cont,Nv,Y);
        Rst = teta(1:Nv/2)' ;
        Sst = teta(Nv/2 +1:Nv)' ;
        R(1:Nv/2,i)=teta(1:Nv/2) ;
        S(1:Nv/2,i)=teta(Nv/2 +1:Nv);
        t0=sum(A_m)/sum(Best) ;

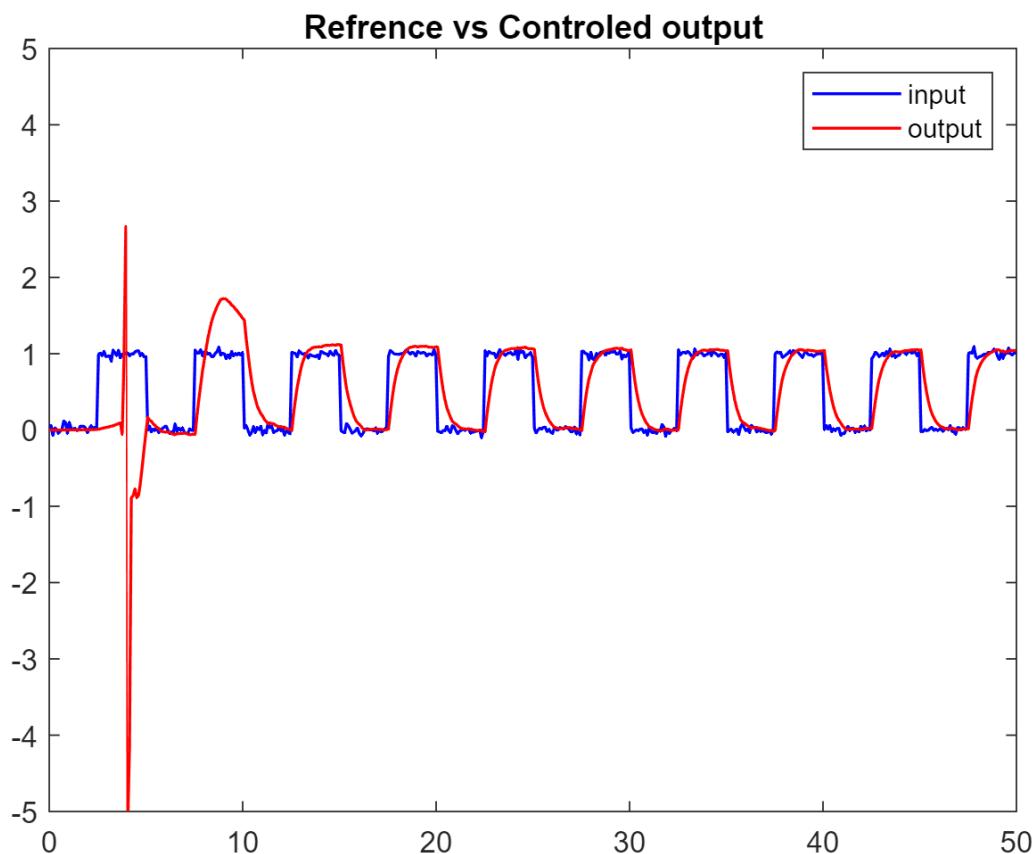
        u(i) = (-Rst(2:end)*u(i-1:-1:i-(numel(Rst(2:end))))...
            +t0*uc(i)...
            -Sst*y(i:-1:i-(numel(Sst)-1)))/Rst(1) ;

        uf(i) = -A0Am(2:end)* uf(i-1:-1:i-Na0am) +Best*u(i:-1:i-
numel(Best)+1) ;
        yf(i) = -A0Am(2:end)* yf(i-1:-1:i-Na0am) +Best*y(i:-1:i-
numel(Best)+1) ;
        ucf(i) = -A0Am(2:end)*ucf(i-1:-1:i-Na0am) +Best*uc(i:-1:i-
numel(Best)+1) ;
    end
end

```

General Input v.s. Output

```
plot(t,uc,'b',t,y,'r','LineWidth',1)
title('Refrence vs Controled output')
legend('input','output')
xlim('auto')
ylim([-5 5])
print(gcf,[Titlework , num2str(plot_counter) ' Refrence vs Controled
output.png'], '-dpng', '-r400');
```

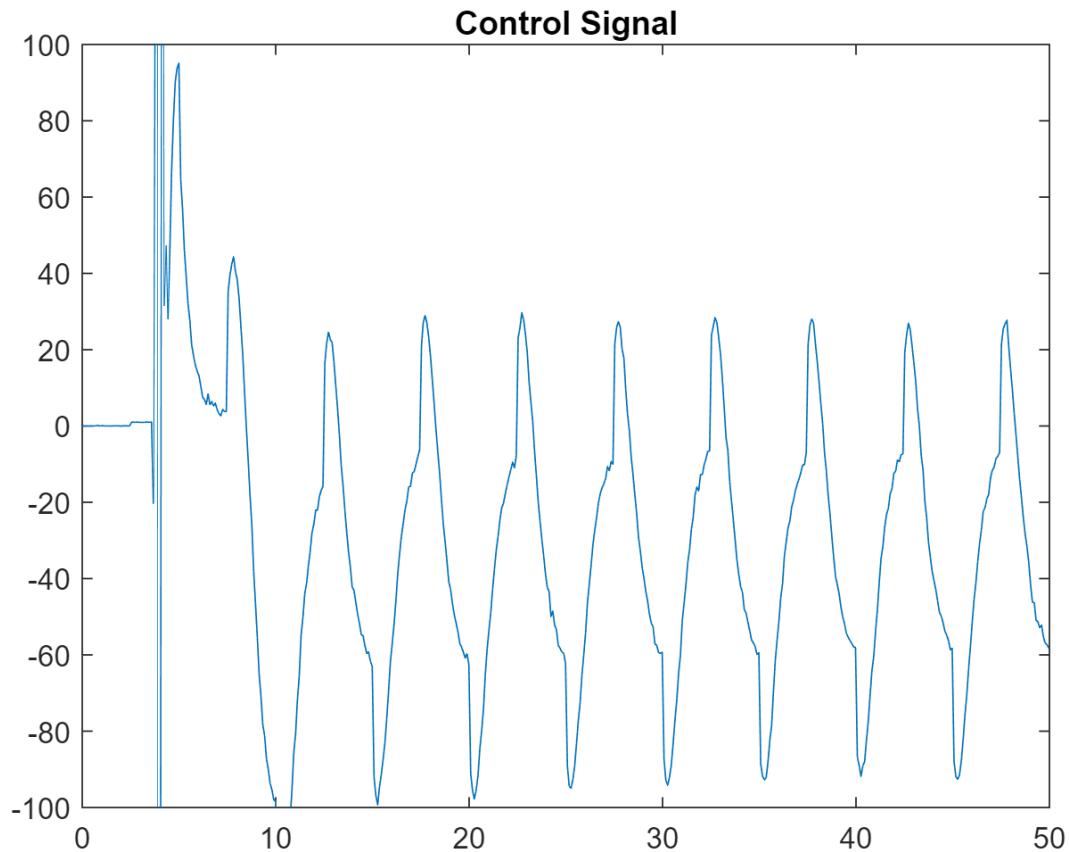


شکل ۲-۱۵ خروجی مستقیم با نویز رنگی بدون حذف صفر

```
plot_counter=plot_counter+1;

plot(t,u)
title('Control Signal')
xlim('auto')
ylim([-100 100])
```

```
print(gcf,[Titlework , num2str(plot_counter) ' Control Signal.png'], '-dpng', '-r400');
```



شکل ۳-۱۶ خروجی کنترلی مستقیم با نویز رنگی بدون حنف صفر

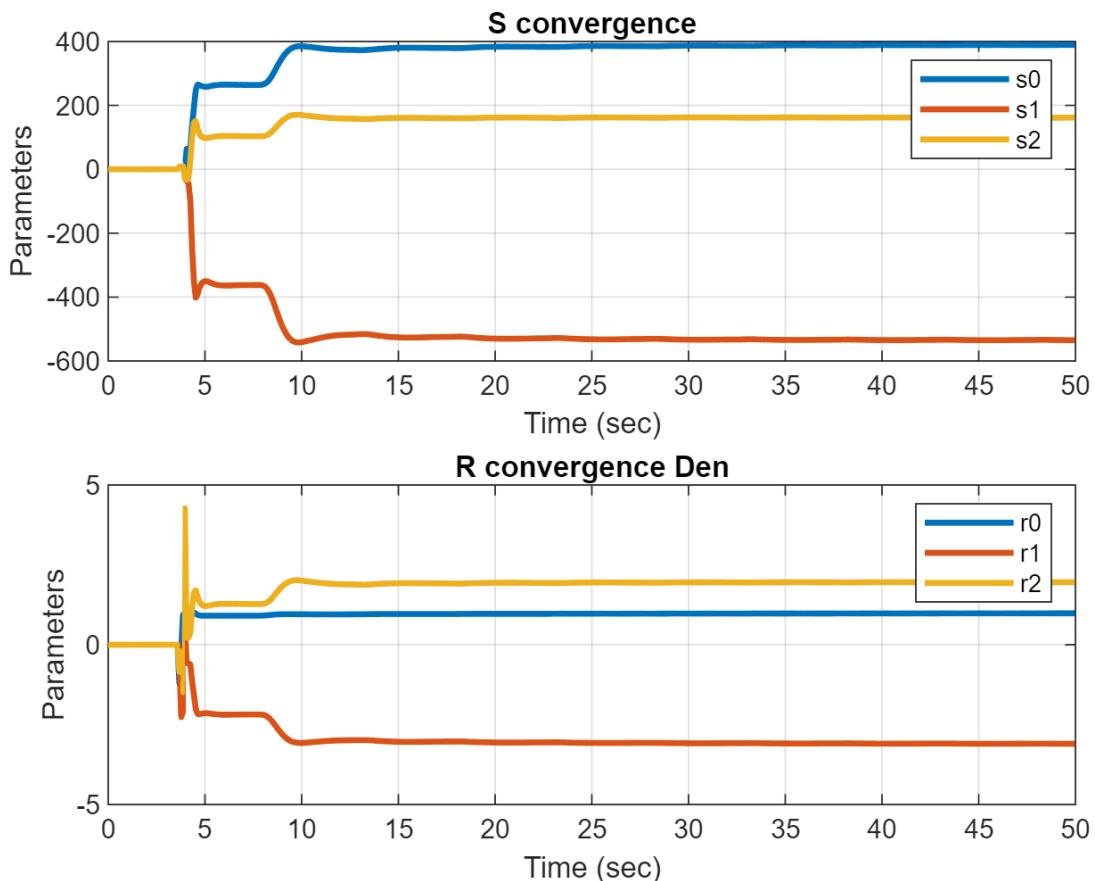
```
plot_counter=plot_counter+1;
```

```
RLS Convergence of R UND S
subplot(2,1,1)
for i=1:number_R
    legend_names{i} = [ 's' num2str(i)-1 ''];
end
plot(t ,S(:,:,1), 'LineWidth' , 2) ;
legend(legend_names)
xlabel('Time (sec)') ;
ylabel('Parameters') ;
title('S convergence') ;
grid on
```

```

subplot(2,1,2)
for i=1:number_R
    legend_names{i} = ['r' num2str(i)-1 ''];
end
plot(t ,R(:,:,i), 'LineWidth' , 2) ;
legend(legend_names)
xlabel('Time (sec)') ;
ylabel('Parameters') ;
title('R convergence Den') ;
xlim('auto')
ylim([-5 5])
grid on
print(gcf,[Titlework , num2str(plot_counter) ' RLS Convegence.png'], '-dpng', '-r400');

```



شکل ۲-۱۷ همگرایی مستقیم با نویز رنگی بدون حذف صفر

```
function [teta,P_sys]=RLS(teta,phi,P_sys,Nv,Y)
    K = P_sys*phi*(1+phi'*P_sys*phi)^(-1) ;
    P_sys = (eye(Nv) - K*phi')*P_sys ;
    teta = teta + K*(Y - phi'*teta ) ;
end
```

۳ - ۲ - ۲ یک کنترل کننده STR به روش غیر مستقیم، با حذف صفر و قطب، طراحی کنید.

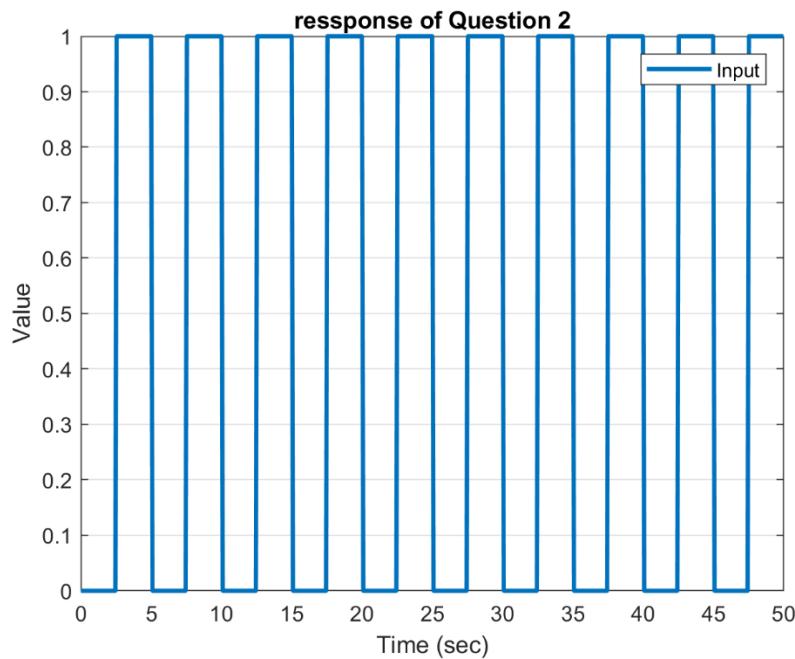
```
run('BASIC.m')
sys_discret =
0.0076672 (z-0.7239) (z-0.8621)
-----
(z-1.109) (z-0.6048) (z-0.5835)

Sample time: 0.094282 seconds
Discrete-time zero/pole/gain model.
```

generate Data

```
Question_mark='Q213';
[uc,t,Status,tfinal]=Datagen(0,T_s,200);
Status = '_No NOISE_'
Titlework=[Question_mark,Status]
Titlework = 'Q213_No NOISE_'

General Input+white Noise
y = lsim(sys_discret ,uc ,t);
plot(t,uc , 'LineWidth',2) ;
xlabel('Time (sec)') ;
ylabel('Value') ;
title('ressponse of Question 2') ;
grid on
legend('Input' , 'OutPut') ;
print(gcf,[Titlework , num2str(plot_counter) ' Refrence .png'],'-dpng',' -r400');
```



```
plot_counter=plot_counter+1;
```

Assumption

```
%choose number of parameters
Parameters_in_den=3
Parameters_in_den = 3
Parameters_in_num=3
Parameters_in_num = 3
degre_cancelled_zero=1
degre_cancelled_zero = 1
A_m=poly(linspace(0.01,0.7,(Parameters_in_num)))
A_m = 1x4
    1.0000    -1.0650     0.2591    -0.0025
B_m=sum(A_m)*[1 zeros(1,(numel(A_m)-1-1))];

deg_A=Parameters_in_num+1;
deg_B=Parameters_in_den ;

DC_gain_model=sum(A_m)/sum(B_m);           betaa=DC_gain_model
betaa = 1
Nv=Parameters_in_num+Parameters_in_den;
deg_B_plus =degre_cancelled_zero+1;
Deg_B_minus=deg_B-deg_B_plus;
```

```

Deg_A_o=0;
%A_o=[1 zeros(1,Deg_A_o)];
A_o=[1];
A_c_prim=conv(A_m,A_o);
u_cont=uc;

```

RLS

```

theta(1:Nv,1:40) = ones(Nv,40) ;
P = 1e16*eye(Nv) ;
phi=[];
N = numel(y) ;
S=zeros(Nv,Parameters_in_num);
R=zeros(Nv,Parameters_in_den);
T=zeros(Nv,Parameters_in_num);
for i = (deg_A+deg_B):N
    if i<50
        y(i) = -A(2:end)*y(i-1:-1:i-3)+B*(u_cont(i-1:-1:i-3)) ;
        phi(:,i) = [(y(i-1:-1:i-Parameters_in_den))' , (u_cont(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));

        Aest=[1 -theta(1:Parameters_in_num ,end)'];
        Best=theta((Parameters_in_num+1):end,end)';
        B_minus=Best(1);
        B_plus=B/B_minus;
    else
        y(i) = -Aest(2:end)*y(i-1:-1:i-Parameters_in_num)+Best*(u_cont(i-1:-1:i-Parameters_in_den)) ;

        phi(:,i) = [(y(i-1:-1:i-Parameters_in_den))' , (u_cont(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));

        Aest=[1 -theta(1:Parameters_in_num ,end)'];
        Best=theta((Parameters_in_num+1):end,end)';
        B_minus=Best(1);
        B_plus=Best/B_minus;

    ;
    [R_prim,Si] = Diophantine(Aest,Best ,A_c_prim);
    S(i,:)=Si;
    T(i,:)=conv(sum(A_m),[1 zeros(1,(numel(A_m)-1-1))])/Best(1);
    R(i,:)=B_plus;

```

```

A_c=conv(A_c_prim,B_plus);

var1=conv(B,T(i,:));      narv1=numel(var1)      ;
var2=A_c(2:end) ;        narv2=numel(A_c(2:end));
var3=conv(A,T(i,:)) ;    narv3=numel(var3)      ;

y(i)      =var1*uc(i:-1:i-narv1+1)-var2*y(i-1:-1:i-narv2);
u_cont(i)=var3*uc(i:-1:i-narv3+1)-var2*u_cont(i-1:-1:i-narv2);
end
end

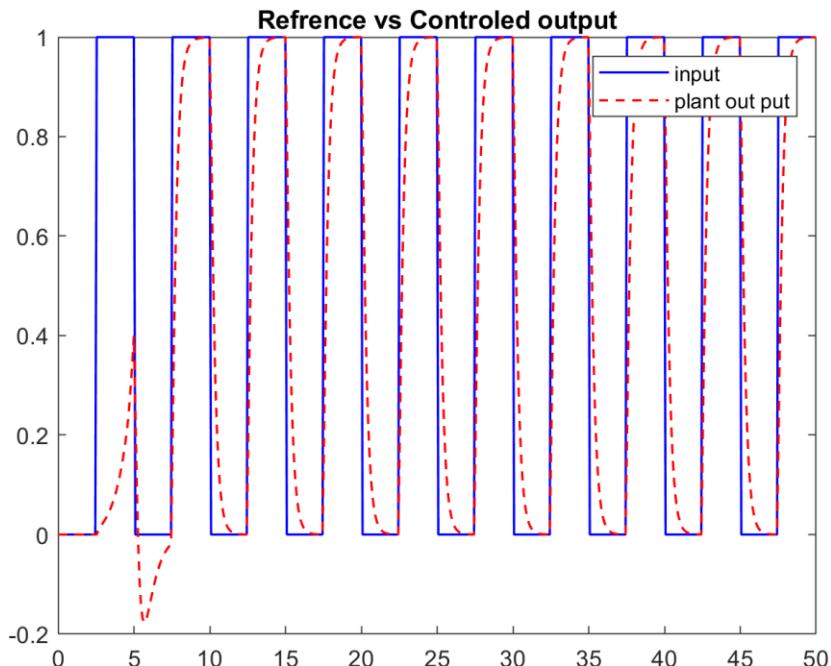
```

General Input v.s. Output

```

plot(t,uc,'b',t,y,'r--','LineWidth',1)
title('Refrence vs Controled output')
legend('input','plant out put')
print(gcf,[Titlework , num2str(plot_counter) ' Refrence vs Controled
output.png'],'-dpng','-r400');

```



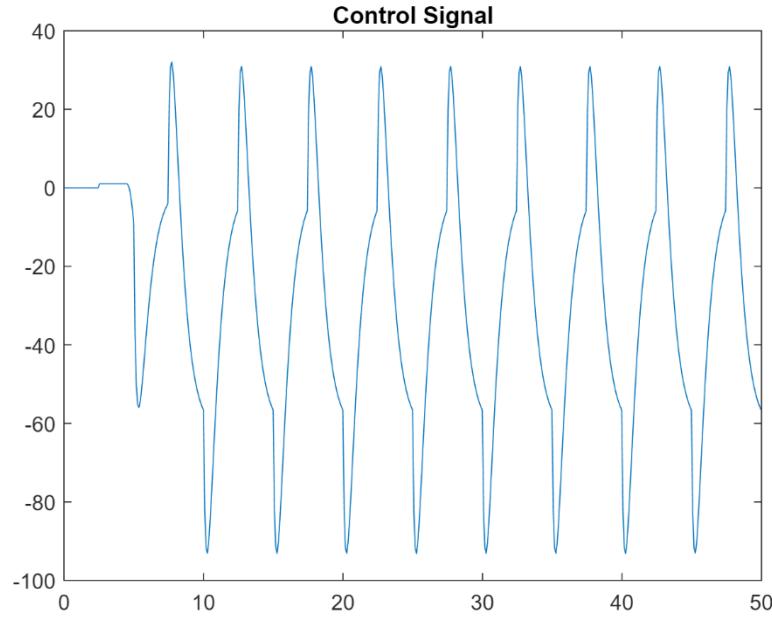
```

plot_counter=plot_counter+1;

plot(t,u_cont)
title('Control Signal')

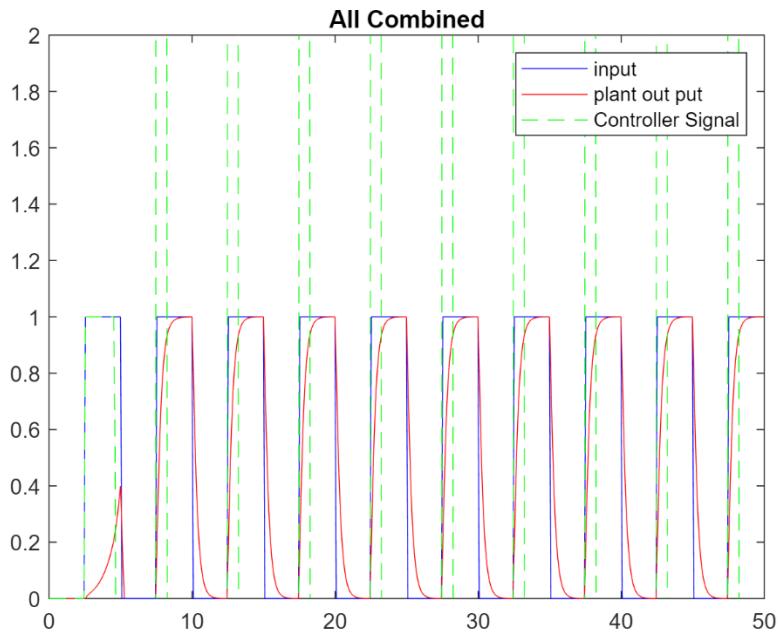
```

```
print(gcf,[Titlework , num2str(plot_counter) ' Control Signal.png'], '-dpng', '-r400');
```



```
plot_counter=plot_counter+1;

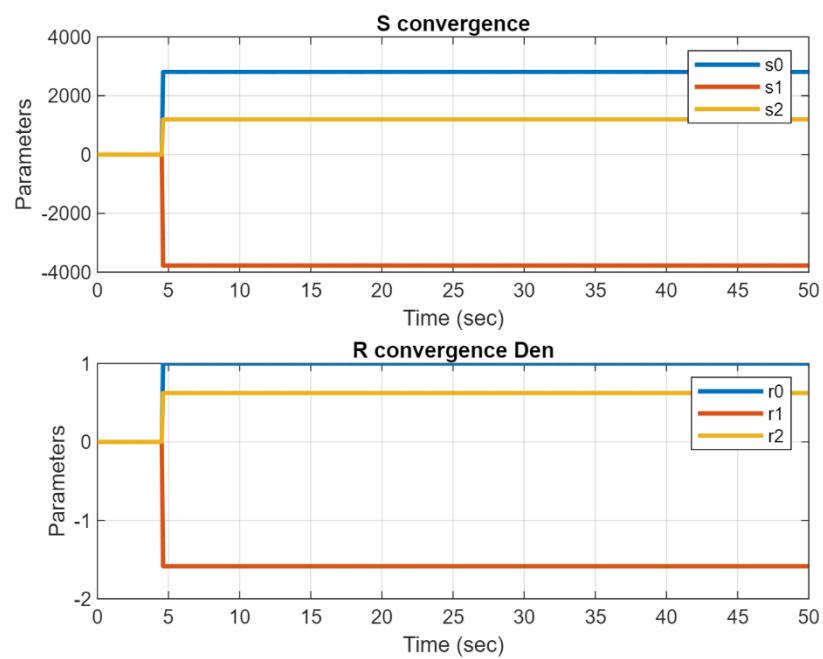
plot(t,gensig('square' , tfinal/10 , tfinal ,T_s), 'b',t,y,'r',t,u_cont,'g--'
,'LineWidth',0.25)
title('All Combined')
legend('input','plant out put','Controller Signal')
xlim([0 tfinal])
ylim([0 2])
print(gcf,[Titlework , num2str(plot_counter) ' All Combined.png'], '-dpng', '-r600');
```



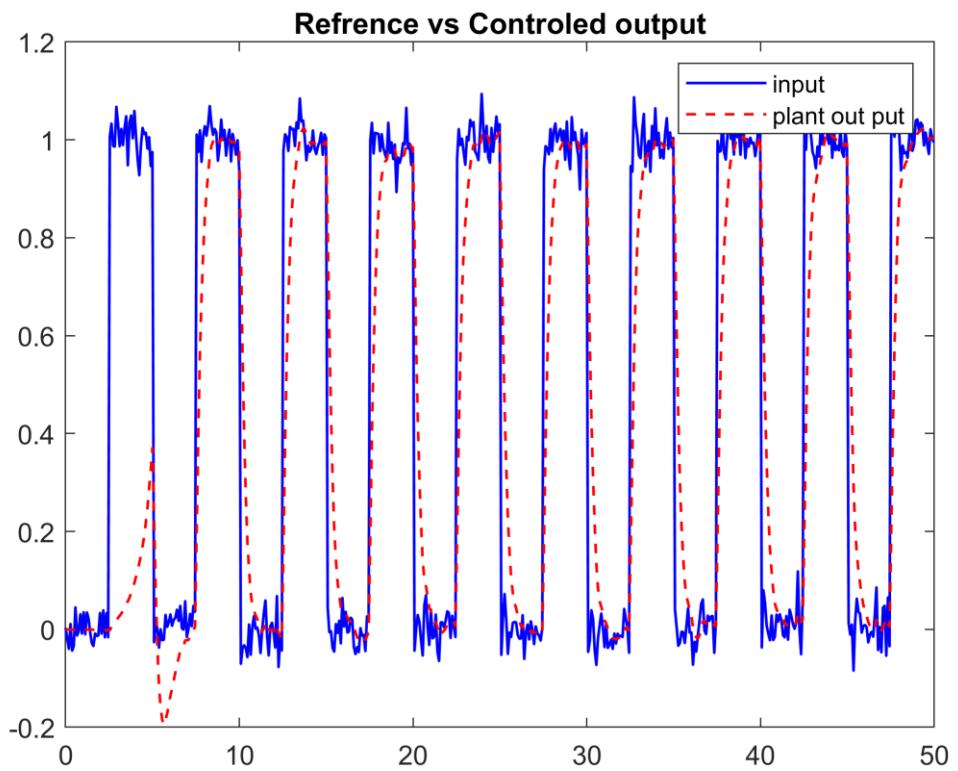
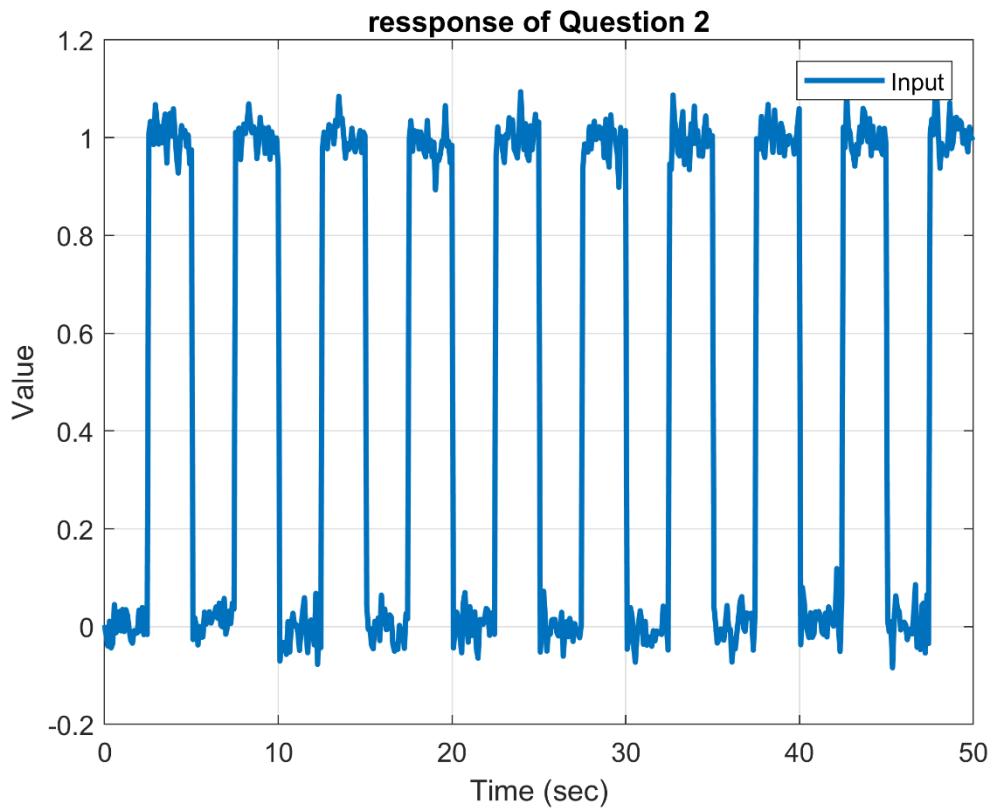
```

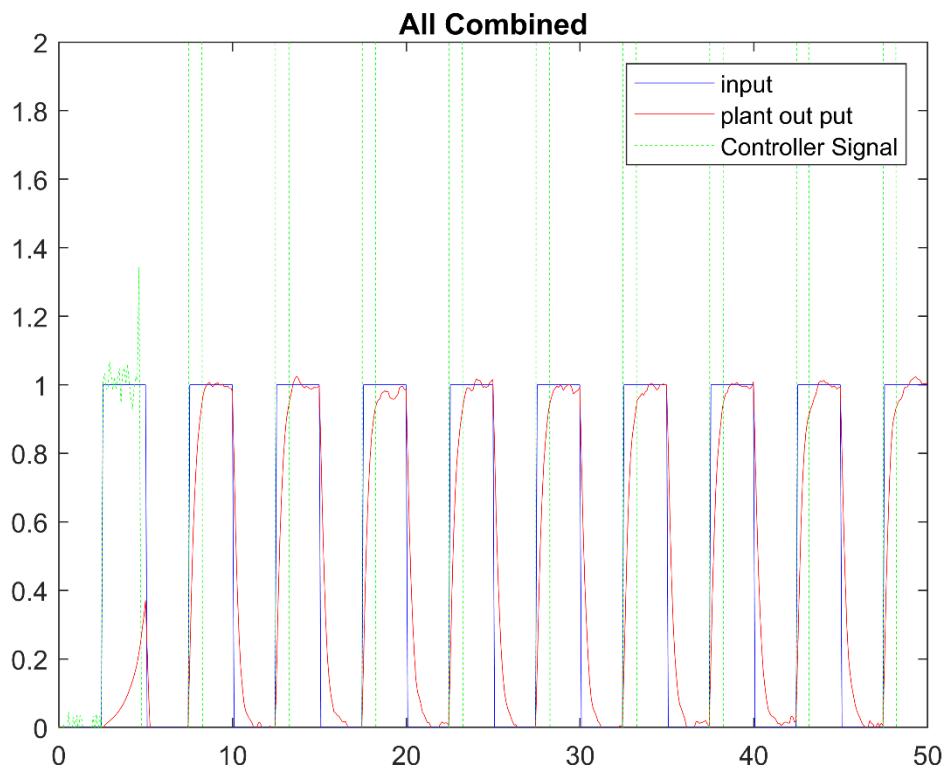
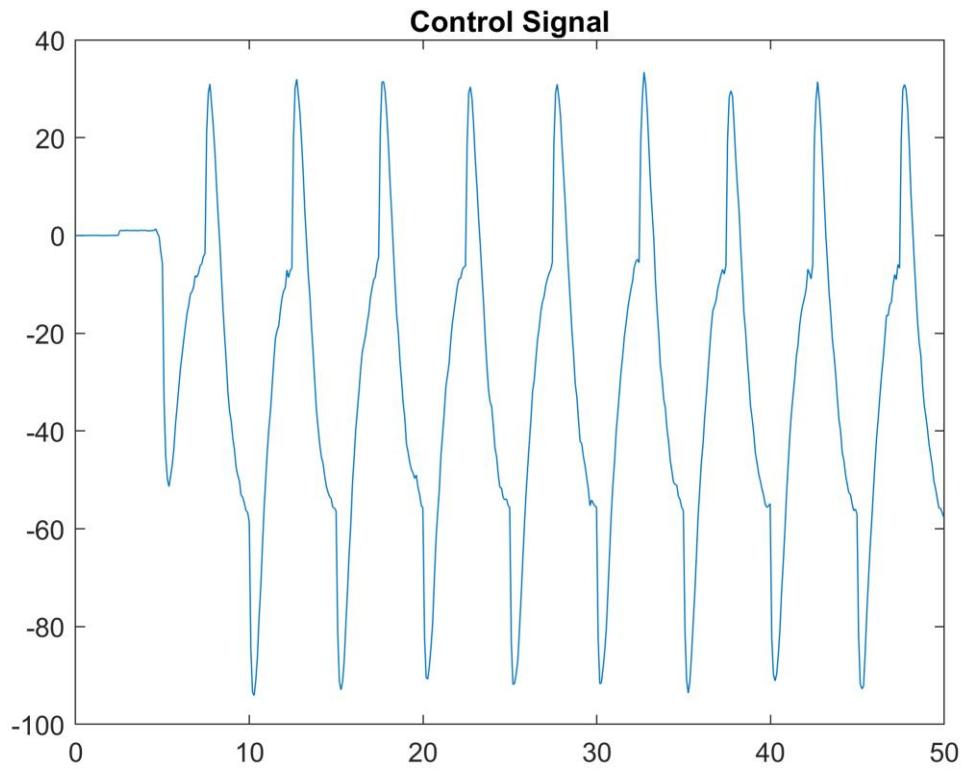
plot_counter=plot_counter+1;
RLS Convergence of R UND S
subplot(2,1,1)
for i=1:deg_A-1
    legend_names{i} = ['s' num2str(i)-1 ''];
end
plot(t ,S(:,:,1), 'LineWidth' , 2) ;
legend(legend_names)
xlabel('Time (sec)') ;
ylabel('Parameters') ;
title('S convergence') ;
grid on
subplot(2,1,2)
for i=1:deg_A-1
    legend_names{i} = ['r' num2str(i)-1 ''];
end
plot(t ,R(:,:,1), 'LineWidth' , 2) ;
legend(legend_names)
xlabel('Time (sec)') ;
ylabel('Parameters') ;
title('R convergence Den') ;
grid on
print(gcf,[Titlework , num2str(plot_counter) ' RLS Convegence.png'], '-dpng', '-r400');

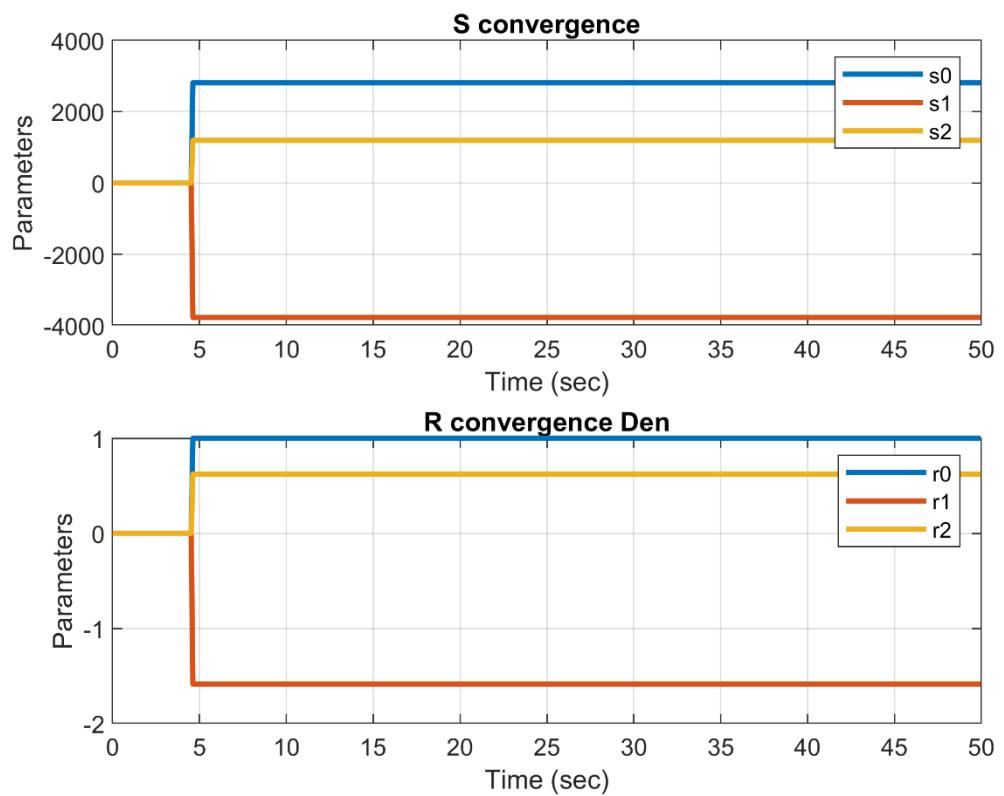
```



برای حالت با نویز رنگی







۴ - ۲ - ۲ یک کنترل کننده STR به روش مستقیم، با حذف صفر و قطب، طراحی کنید.

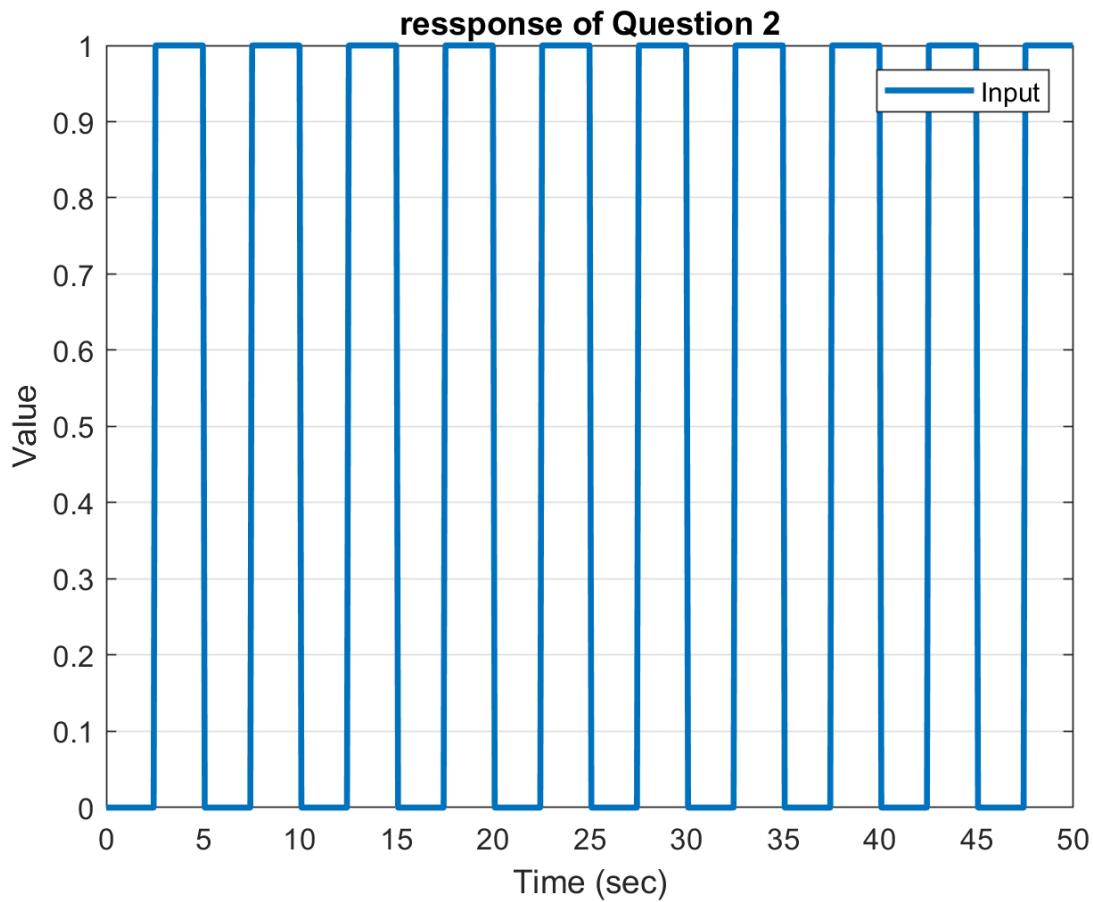
```
run('BASIC.m')
sys_discret =
0.0076672 (z-0.7239) (z-0.8621)
-----
(z-1.109) (z-0.6048) (z-0.5835)

Sample time: 0.094282 seconds
Discrete-time zero/pole/gain model.
```

generate Data

```
Question_mark='Q214';
[uc,t,Status,tfinal]=Datagen(0,T_s,200);
Status = '_No NOISE_'
Titlework=[Question_mark,Status]
Titlework = 'Q214_No NOISE_'

y = lsim(sys_discret ,uc ,t);
plot(t,uc , 'LineWidth',2) ;
xlabel('Time (sec)');
ylabel('Value') ;
title('ressponse of Question 2') ;
grid on
legend('Input' , 'OutPut') ;
print(gcf,[Titlework , num2str(plot_counter) ' Refrence .png'],'-dpng',' -r400');
```



```
plot_counter=plot_counter+1;
```

Assumption

```

Deg_R=3;
number_R=Deg_R+1;

n = numel(A)-1 ;
m = numel(B)-1 ;
d0 = n-m ;
number_S=number_R;
Nv = number_R*2 ;
%choose number of parameters
A_m=poly(linspace(0.01,0.7,number_R))
A_m = 1×5
    1.0000    -1.4200     0.6239   -0.0851    0.0008
% A_m=poly([0.3 0.4 0.2 0.1]);
B_m=sum(A_m)*[1 zeros(1,(numel(A_m)-d0-1))];
sys_ref=tf(B_m,A_m,T_s)

```

```

sys_ref =
0.1196 z^3
-----
z^4 - 1.42 z^3 + 0.6239 z^2 - 0.08506 z + 0.0007896

Sample time: 0.094282 seconds
Discrete-time transfer function.
y_ref_plant = lsim(sys_discret ,uc ,t)';
y_ref      = lsim(sys_ref , uc , t) ;

```

initial parameters

```

A_o = [1] ;

A_oA_m = conv(A_o , A_m) ;
deg_aoam = numel(A_oA_m)-1 ;

L = deg_aoam-d0 ;

N = numel(t) ;
teta = 10*ones(Nv,1);
theta_sys=1*ones(6,1);
u = uc ; % initial effort control
%y = 0.1*ones(Nv , 1) ; % initial output
uf = uc ; % initial filtered effort control
yf = 0.1*ones(Nv , 1) ; % initial filtered output
ucf= uc ; % initial filtered command signal
P_cont = 1e12*eye(Nv);
P_sys = 1e12*eye(6);
R=zeros(3,N);
S=zeros(3,N);

```

main loop Direct Str algorithm

```

for i = Nv+1:N

if i<40
    y(i) = -A(2:end)*y(i-1:-1:i-n)+B*(u(i-1:-1:i-n)) ;
else
    y(i) = -Aest(2:end)*y(i-1:-1:i-n)+Best*(u(i-1:-1:i-n)) ;
end

Y = y(i);%-y_ref(i);
phi_sys = [(y(i-1:-1:i-3))' , (u(i-1:-1:i-3))']';
[theta_sys,P_sys]=RLS(theta_sys,phi_sys,P_sys,6,Y);
Aest=[1 -theta_sys(1:3 ,end)'];
Best=theta_sys(4:6,end)';

```

```

phi = [uf(i-d0:-1:i-number_R)' , yf(i-d0:-1:i-number_S)']';
[teta,P_cont] =RLS(teta,phi,P_cont,Nv,Y);
Rst = teta(1:Nv/2)' ;
Sst = teta(Nv/2 +1:Nv)' ;
R(1:Nv/2,i)=teta(1:Nv/2) ;
S(1:Nv/2,i)=teta(Nv/2 +1:Nv);
t0=sum(A_m) ;

u(i) = (-Rst(2:end)*u(i-1:-1:i-(numel(Rst(2:end))))...
+t0*uc(i)...
-Sst*y(i:-1:i-(numel(Sst)-1)))/Rst(1) ;

uf(i) = -A_oA_m(2:end)* uf(i-1:-1:i-deg_aoam) +u(i) ;
yf(i) = -A_oA_m(2:end)* yf(i-1:-1:i-deg_aoam) +y(i) ;
ucf(i) = -A_oA_m(2:end)*ucf(i-1:-1:i-deg_aoam) +uc(i) ;

end

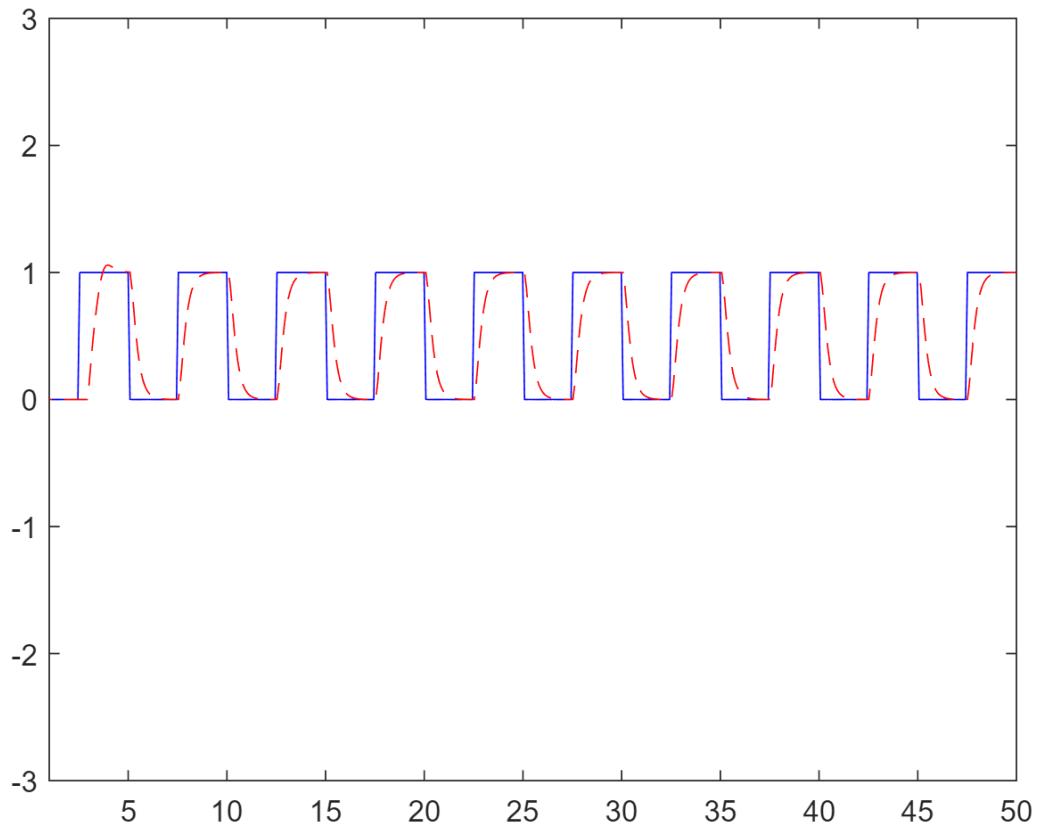
```

plot results

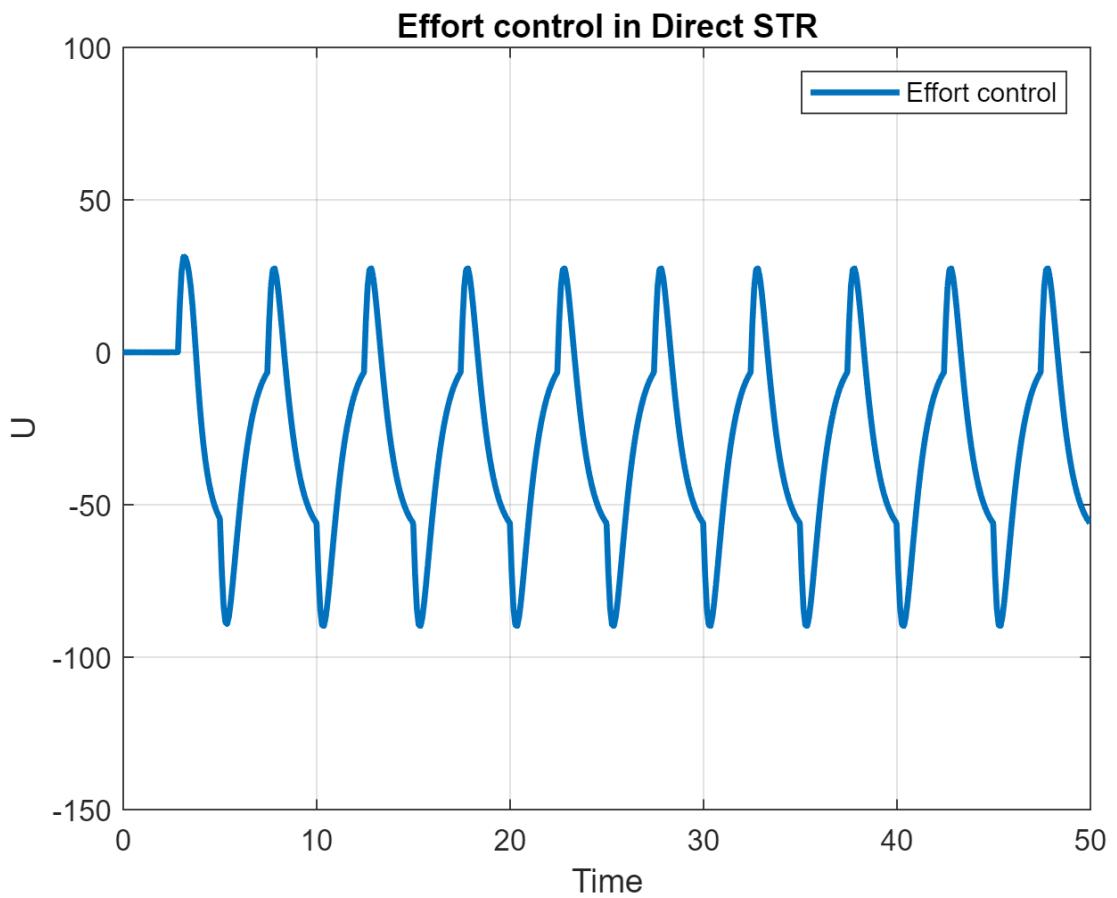
```

figure()
plot(t , uc, 'b',t , y, 'r--')
xlim([1 tfinal])
ylim([-3 3])

```

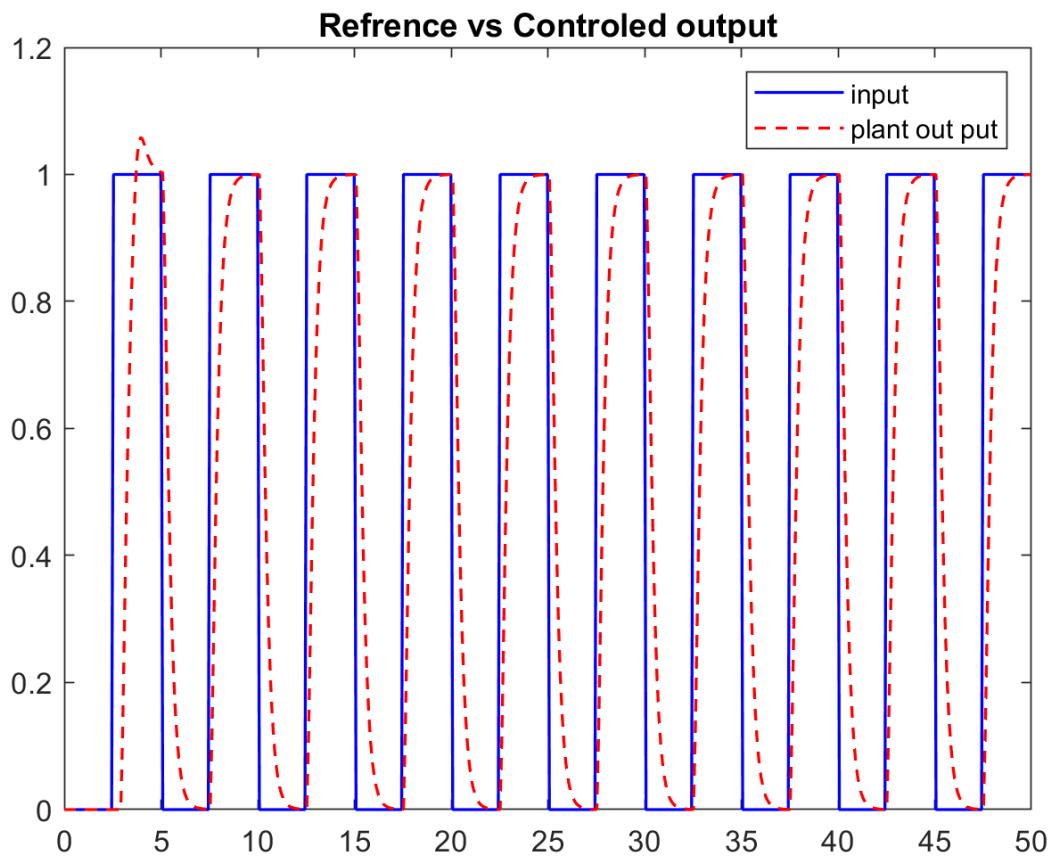


```
figure()
plot(t , u , 'LineWidth' , 2) ;
xlabel('Time') ;
ylabel('U') ;
title('Effort control in Direct STR') ;
grid on
legend('Effort control') ;
xlim('auto')
ylim([-150 100])
```



General Input v.s. Output

```
plot(t,uc,'b',t,y,'r--','LineWidth',1)
title('Refrence vs Controled output')
legend('input','plant out put')
print(gcf,[Titlework , num2str(plot_counter) ' Refrence vs Controled
output.png'], '-dpng', '-r400');
```

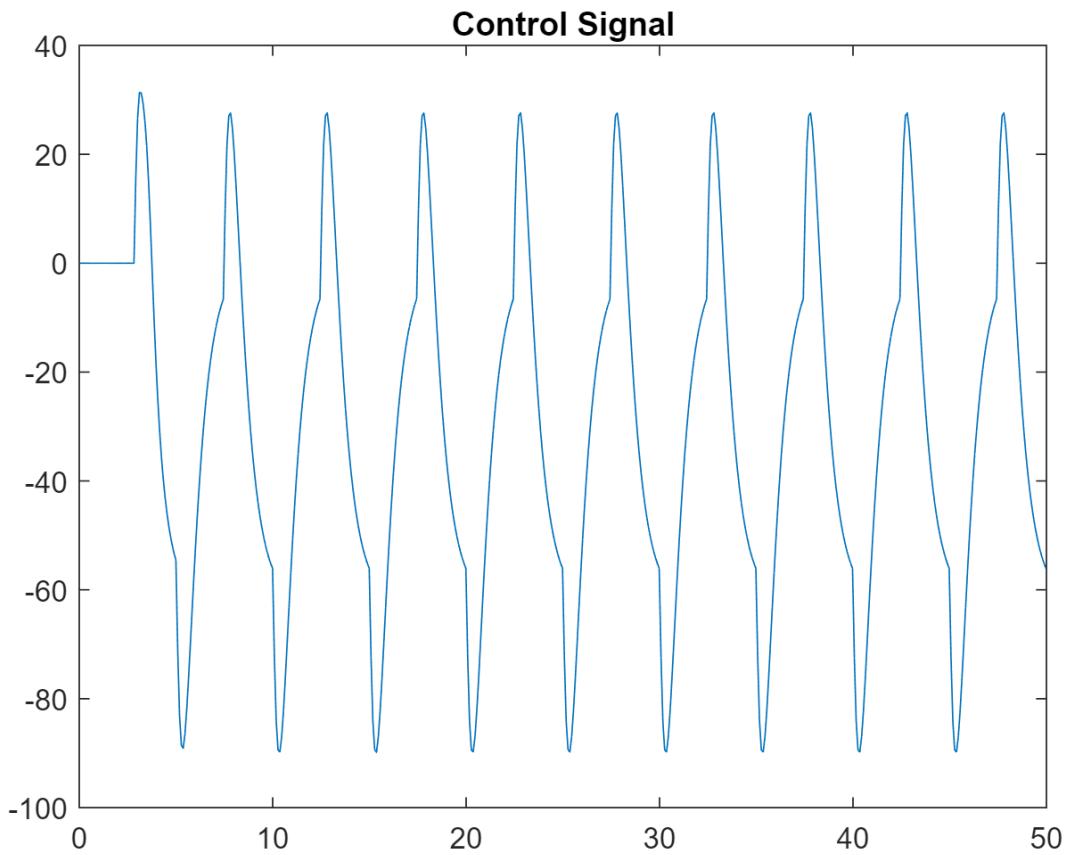


```

plot_counter=plot_counter+1;

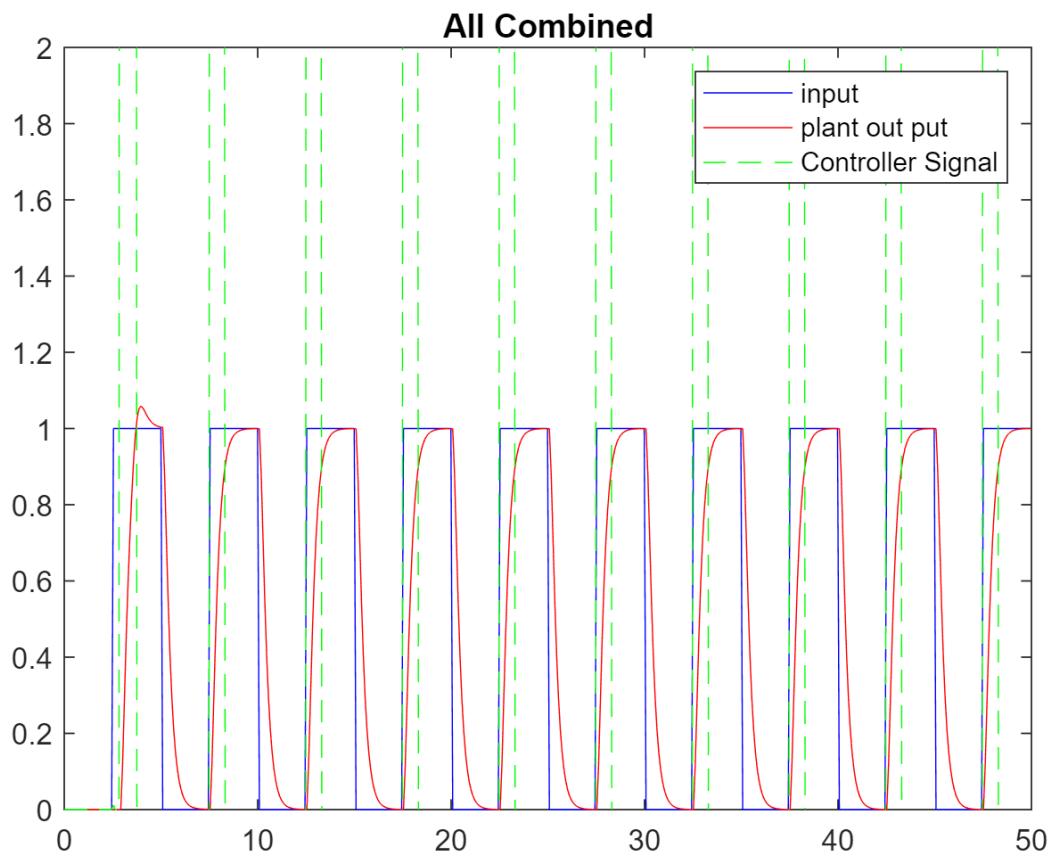
plot(t,u)
title('Control Signal')
print(gcf,[Titlework , num2str(plot_counter) ' Control Signal.png'], '-dpng', '-r400');

```



```
plot_counter=plot_counter+1;

plot(t,gensig('square' , tfinal/10 , tfinal ,T_s), 'b',t,y, 'r',t,u, 'g--'
,'LineWidth',0.25)
title('All Combined')
legend('input','plant out put','Controller Signal')
xlim([0 tfinal])
ylim([0 2])
print(gcf,[Titlework , num2str(plot_counter) ' All Combined.png'], '-dpng', '-r600');
```



```

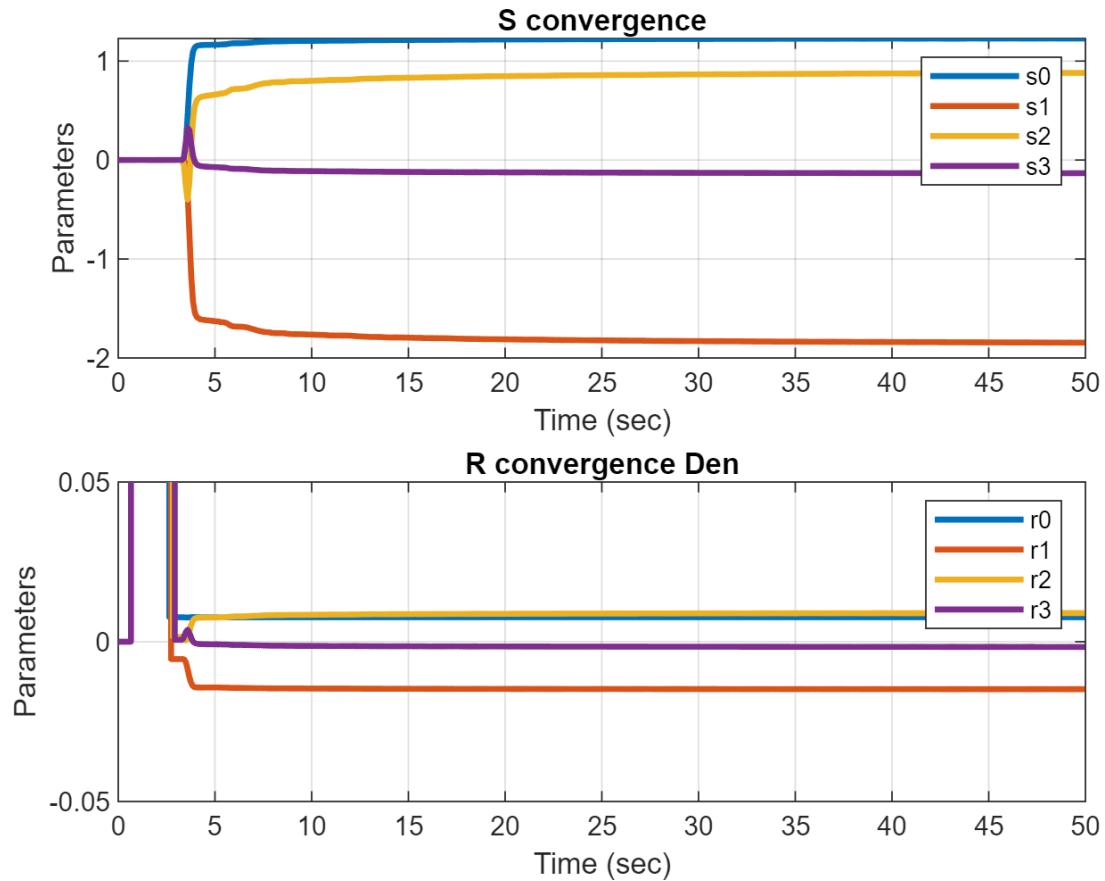
plot_counter=plot_counter+1;
RLS Convergence of R UND S
subplot(2,1,1)
for i=1:4
    legend_names{i} = ['s' num2str(i)-1 ''];
end
plot(t ,S(:,:,1), 'LineWidth' , 2) ;
legend(legend_names)
xlabel('Time (sec)') ;
ylabel('Parameters') ;
title('S convergence') ;
grid on
subplot(2,1,2)
for i=1:4
    legend_names{i} = ['r' num2str(i)-1 ''];
end
plot(t ,R(:,:,1), 'LineWidth' , 2) ;
legend(legend_names)
xlabel('Time (sec)') ;

```

```

ylabel('Parameters') ;
title('R convergence Den') ;
grid on
xlim('auto')
ylim([-0.05 0.05])
print(gcf,[Titlework , num2str(plot_counter) ' RLS Convegence.png'], '-dpng', '-r400');

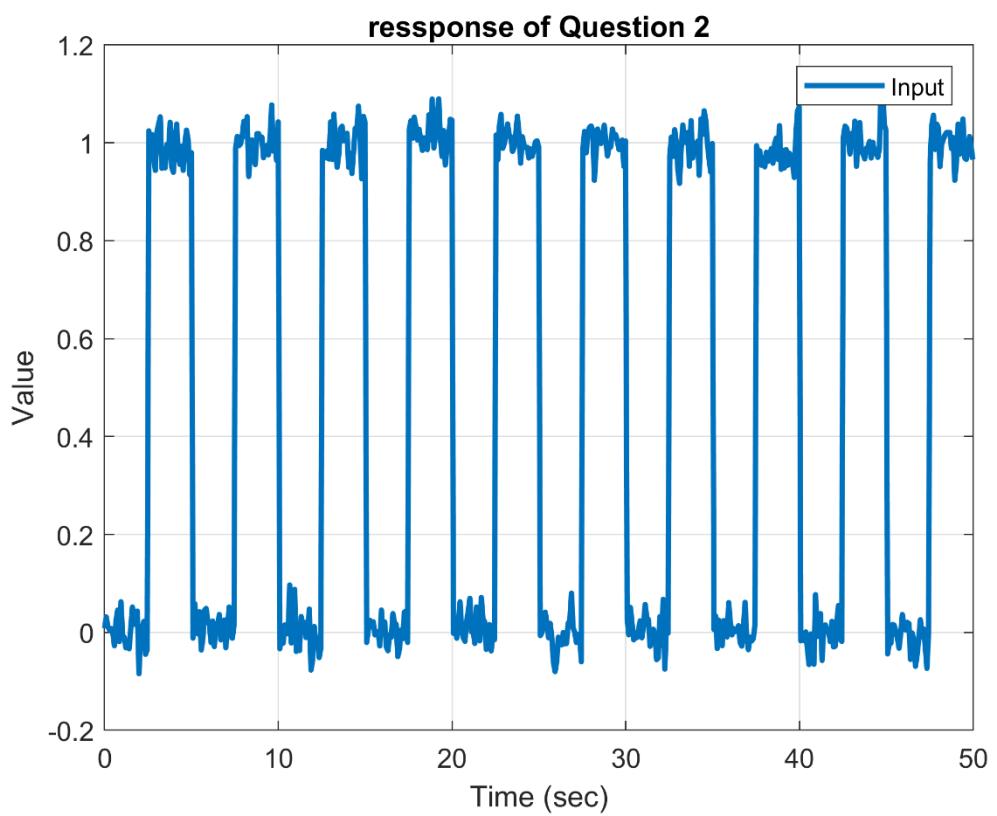
```

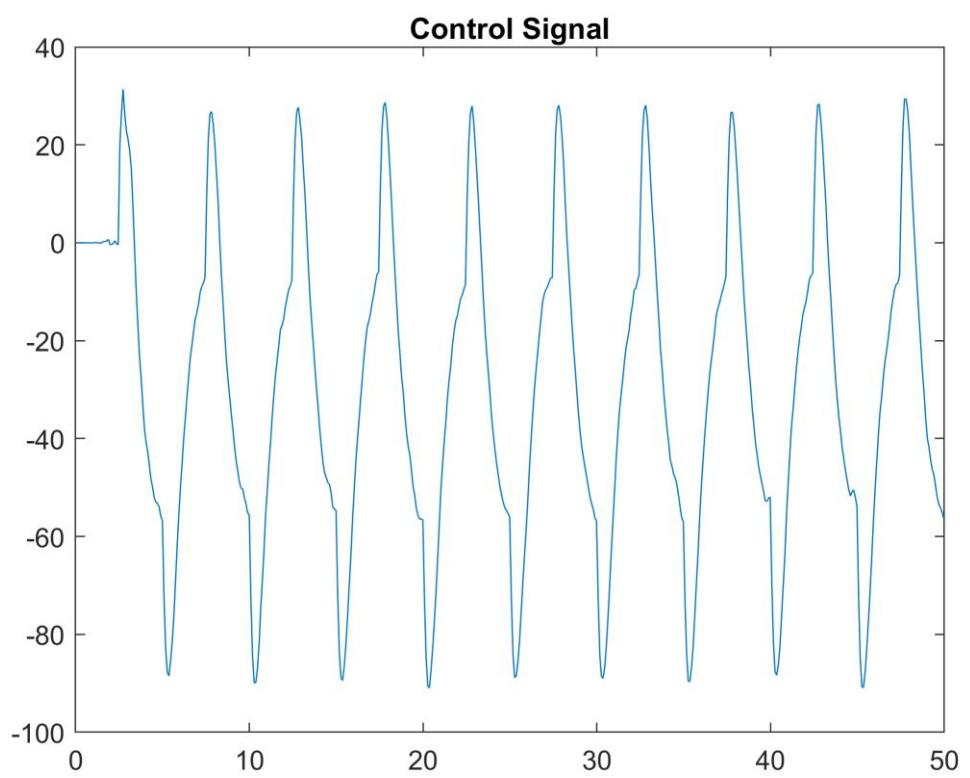
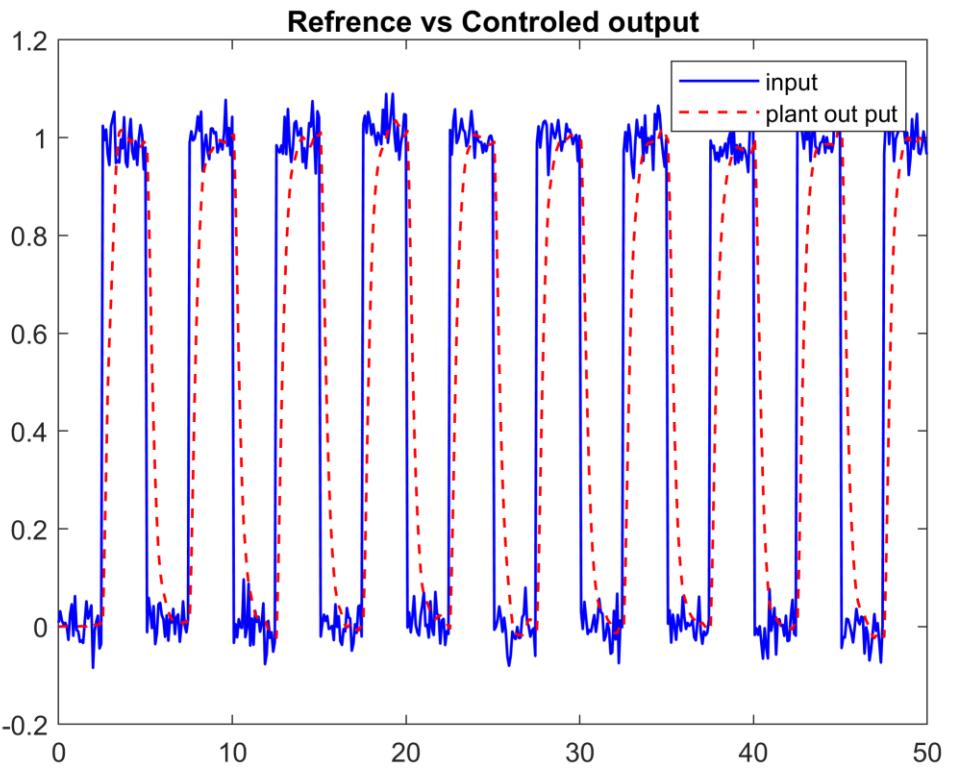


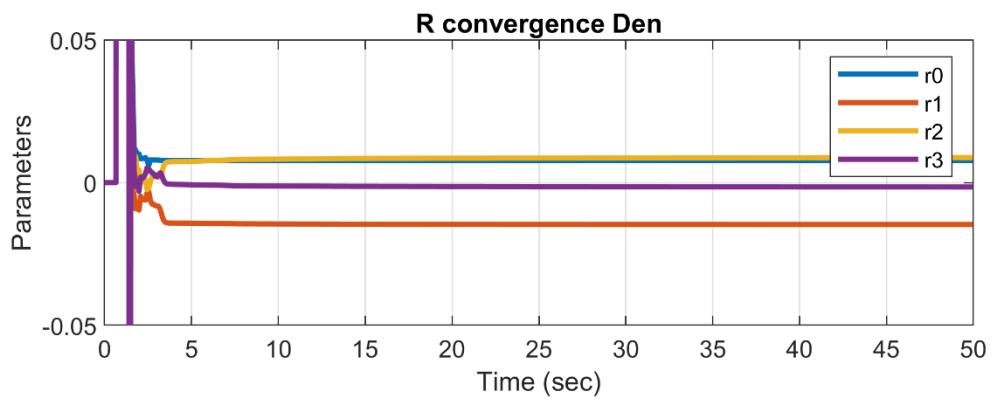
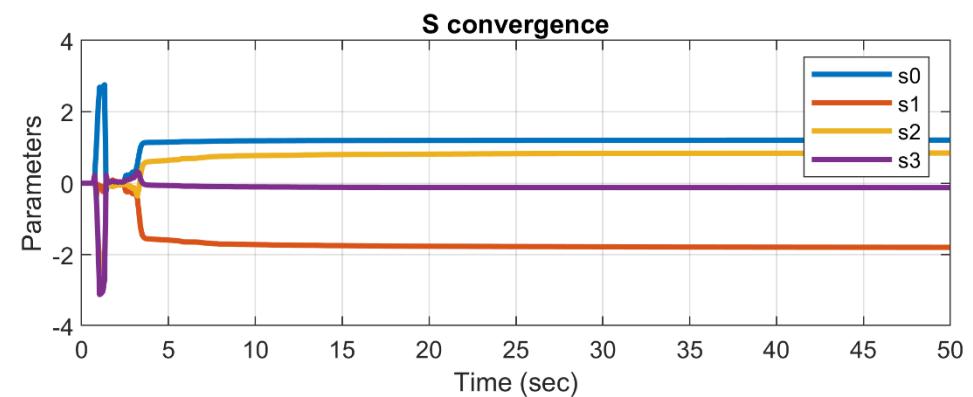
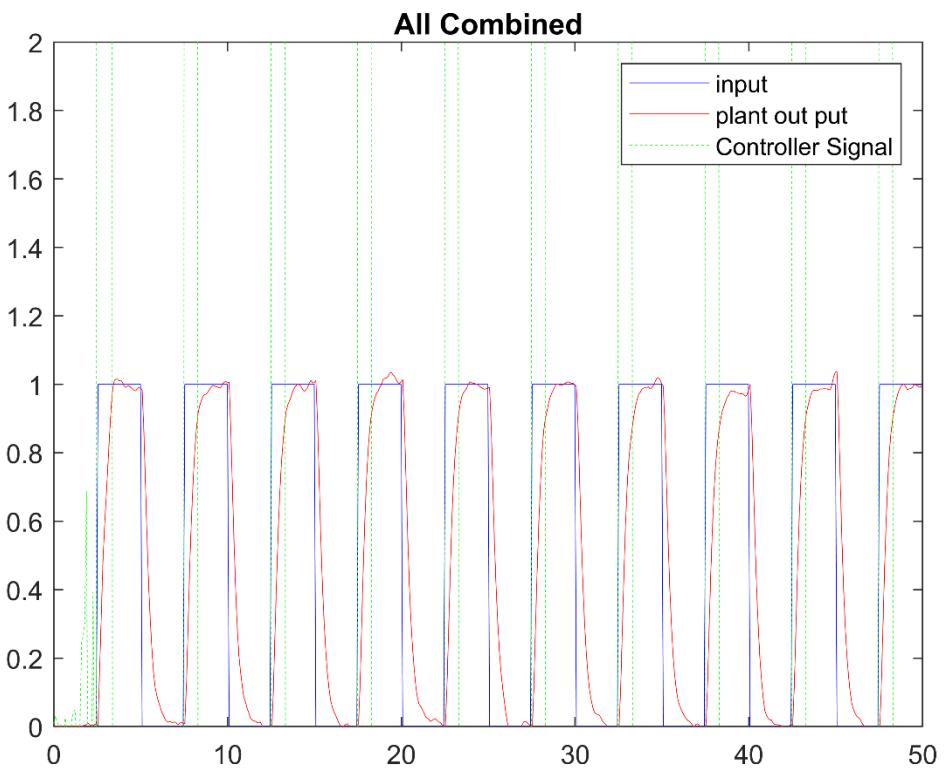
```

function [teta,P_sys]=RLS(teta,phi,P_sys,Nv,Y)
K = P_sys*phi*(1+phi'*P_sys*phi)^(-1) ;
P_sys = (eye(Nv) - K*phi')*P_sys ;
teta = teta + K*(Y - phi'*teta ) ;
end

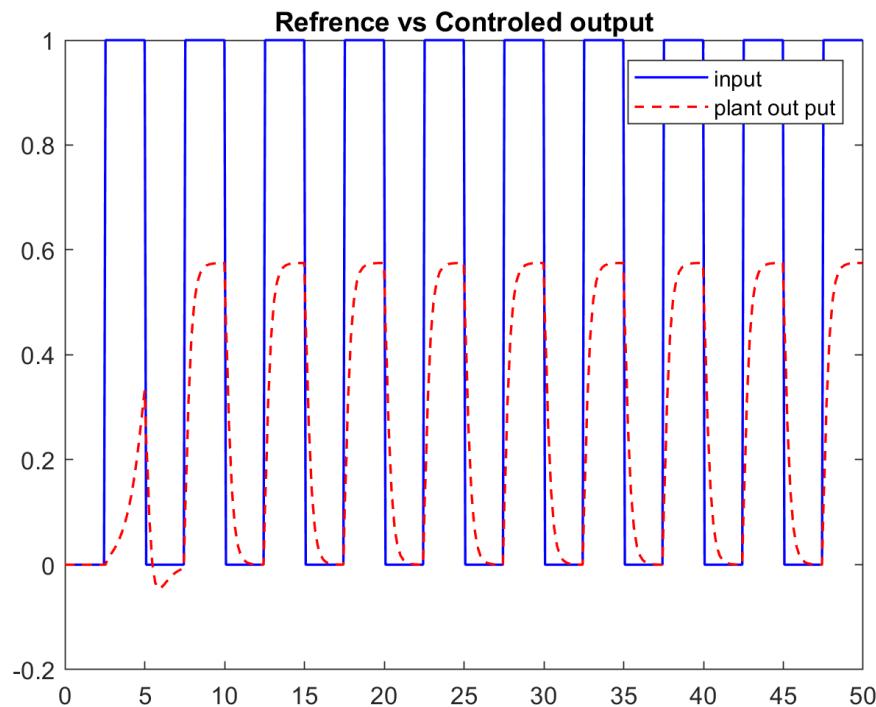
```





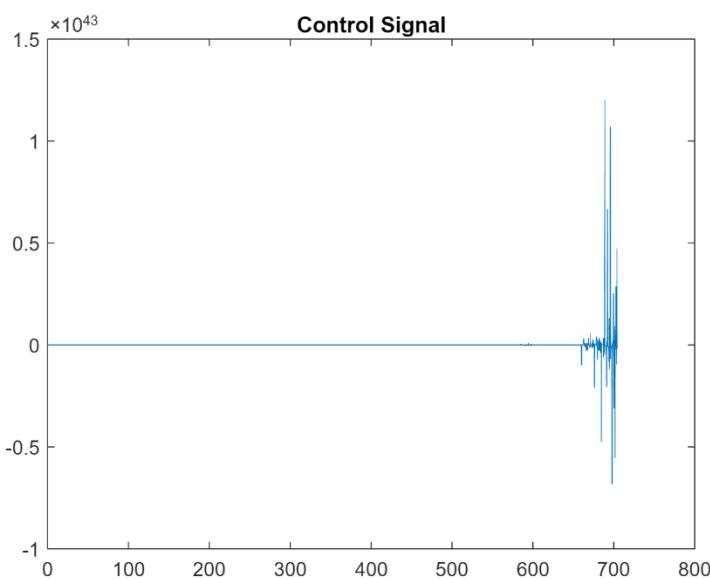
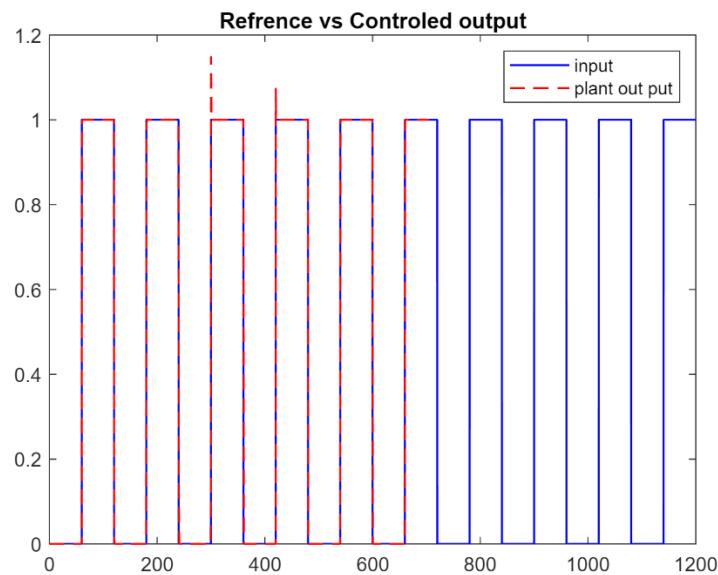


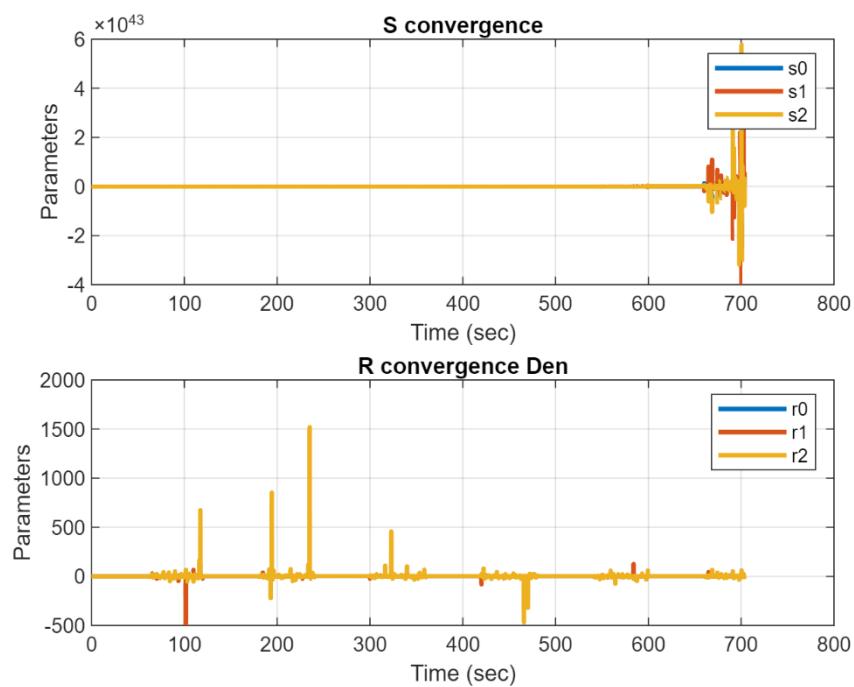
۳ - ۲ اثر Under parameter و Over parameter بودن مدل را در ۴ کنترل کننده فوق بررسی کنید.
 معمولاً با افزایش تعداد پارامترها کمی زمان همگرایی پارامترها افزایش می‌یابد ولی ترکینگ به خوبی صورت می‌گیرد به جز در افزایش مرتبه مدل در حالت غیرمستقیم بدون حذف صفر و قطب که تعقیب ورودی مرجع به خوبی صورت نمی‌گیرد.



در حالتی هم که تعداد پارامترها کمتر از مقدار صحیح باشد ترکینگ و سرعت کنترلی مناسب است ولی کمی بایاس همیشه وجود دارد.

۲ - ۴ در این قسمت اگر شبیه سازی را برای زمانهای طولانی انجام دهید آیا ورودی کنترلی بزرگ میشود؟ بحث کنید کنترل کننده STR غیر مستقیم بدون حذف صفر تا ۶۰۰ ثانیه می‌تواند کنترل انجام دهد بعد از آن پارامترهای S و اگرا می‌شوند به خاطر خطاهای تجمعی محاسباتی (به خاطر وارانهای ماتریس).

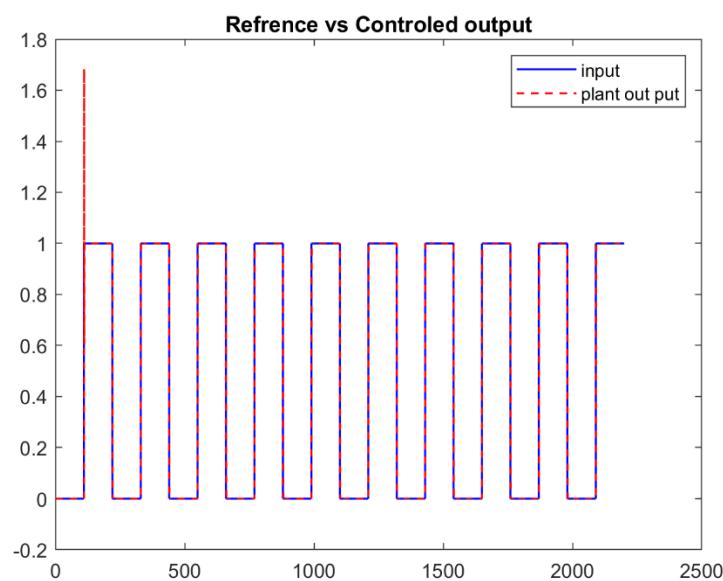


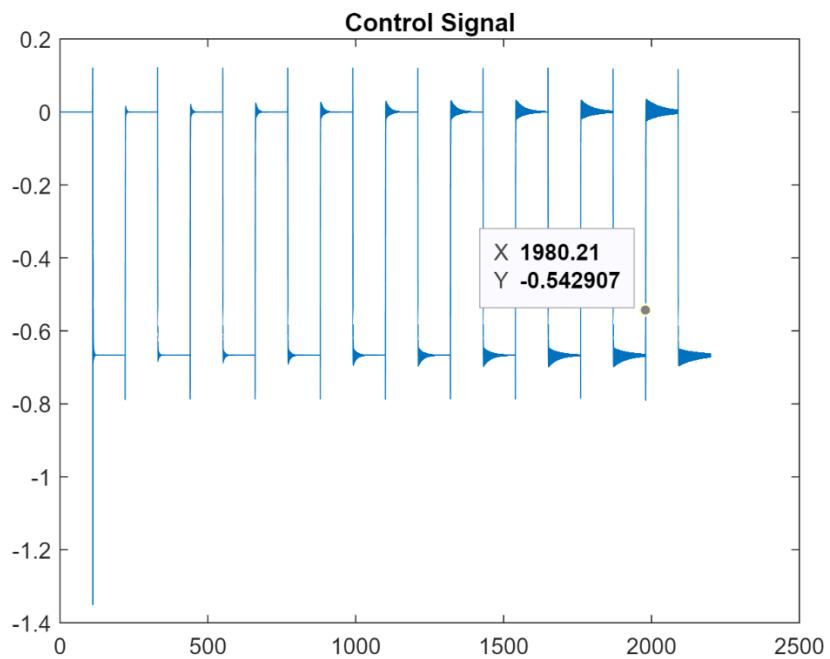


کنترل کننده STR مستقیم بدون حذف صفر با و بدون نویز به خوبی در زمان‌های طولانی خروجی را کنترل می‌کند.

کنترل کننده STR غیر مستقیم با حذف صفر با و بدون نویز به خوبی در زمان‌های طولانی خروجی را کنترل می‌کند.

کنترل کننده STR مستقیم با حذف صفر با و بدون نویز به خوبی در زمان‌های طولانی خروجی را کنترل می‌کند. اما خروجی کنترلی پدیده‌ای شبیه رینگ‌اینگ مشاهده می‌شود اما باعث توقف شبیه سازی نمی‌شود.





۳ STR با جابابی قطب حداقل درجه برای سیستم نامینیمم فاز

یک سیستم غیر مینیمم فاز انتخاب کرده (صفر سیستم دینامیکی اولیه مربوط به خود را سمت راست قرار داده تا سیستم نامینیمم فاز گردد)، با انتخاب یکی از الگوریتم‌های ۴-۳ یا ۵-۵ مطرح شده در کتاب به طراحی کنترلر STR مستقیم یا غیر مستقیم پرداخته و نتایج بدست آمده را تحلیل کنید.

برای یکی از دو الگوریتم انتخاب شده چند جمله‌ای T را به جای محاسبه با تخمین بدست آورید.

در این سوال از الگوریتم ۵-۵ استفاده شده است.

هر دو صفر سیستم به سمت راست محور موهومی برده شده اند.

```
sys_cont=zpk([+1.6 +3.4],[-4/0.7 -1.6/3 +1.1],0.1)
sys_cont =
```

$$\frac{0.1 (s-1.6) (s-3.4)}{(s+5.714) (s-1.1) (s+0.5333)}$$

Continuous-time zero/pole/gain model.

```
BW=bandwidth(sys_cont);
```

```
Discret_ratio=30; % >2 % to signal can be reconstructable we must have
```

```
2*Bandwidth
```

```
T_s=2*pi/(BW*Discret_ratio)
```

```
T_s = 0.4272
```

```
sys_discret=c2d(sys_cont,T_s, 'zoh')
```

```
sys_discret =
```

$$\frac{-0.0020776 (z-1.931) (z+8.362)}{(z-0.7963) (z-1.6) (z-0.08708)}$$

Sample time: 0.42716 seconds

Discrete-time zero/pole/gain model.

```
[num_discret,den_discret]=tfdata(sys_discret);
```

```
num_discret=cell2mat(num_discret);
```

```
num_discret=num_discret(2:end) ;B=num_discret
```

```
B = 1x3
```

$$-0.0021 \quad -0.0134 \quad 0.0335$$

```
roots(num_discret)
```

```
ans = 2x1
```

$$-8.3623$$

$$1.9307$$

هر دور صفر سیستم گسسته نیز بیرون دایره واحد هستند یعنی زمان نمونه برداری برای این سیستم گسسته مناسب است.

```
den_discret=cell2mat(den_discret) ;A=den_discret
```

```
A = 1x4
```

$$1.0000 \quad -2.4831 \quad 1.4825 \quad -0.1109$$

```

B(B==0) = [] ; % remove zeros
tfinal=1500;
t = 0:T_s:tfinal;

    General Input Noise
uc =(gensig('square' , tfinal/10 , tfinal ,T_s))';
% desire system
Am=poly([0.4 0.5 0.6])
Am = 1x4
    1.0000   -1.5000    0.7400   -0.1200
Bm=(sum(Am)/sum(B))*B;
sys_ref=tf(Bm,Am,T_s)
sys_ref =

```

$$\frac{-0.01377 z^2 - 0.08858 z + 0.2223}{z^3 - 1.5 z^2 + 0.74 z - 0.12}$$

```

Sample time: 0.42716 seconds
Discrete-time transfer function.
y_ref_plant = lsim(sys_discret ,uc ,t)';
y_ref = lsim(sys_ref , uc , t) ;

n = numel(A)-1 ;
m = numel(B)-1 ;
N = numel(t) ;

Bplus = 1;
Bminus = B;
d0 = n-m;

```

MDPP STR (Direct)

```

A0 = [1 zeros(1,numel(A)-numel(Bplus)-1)];
Ac = conv(conv(Bplus,Am),A0);

A0Am = conv(A0 , Am) ;
Na0am = numel(A0Am)-1 ;
l = Na0am-d0 ; % deg(S)=deg(R)=1
Nv = 7 ;

Phi = [];
teta = 1*ones(Nv , 1);
S_vec = [];
R_vec = [];
P = 1e16*eye(Nv) ;
lambda = 1;

Nv_sys=6;

```

```

theta=repmat([-2 1 0 0 0 0],6);
%P_sys=1e1*eye(Nv_sys);
P_sys=1e1*eye(6);

u = 0.1*ones(1,N) ; % initial effort control
y = 0.1*ones(1,N) ; % initial output
uf = 0.1*ones(1,N) ; % initial filtered effort control
yf = 0.1*ones(1,N) ; % initial filtered output
ucf= 0.1*ones(1,N) ; % initial filtered command signal

```

main loop

```

% if we does not zero cancellation the for loop should start from "d0+L+1"
iteration
% else we start from "n+1" iteration
for i = 4:N
    y(i) = [-(y(i-1:-1:i-n)),(u(i-(n-m):-1:i-n))]*[den_discret(2:end),num_discret]';
    phi(:,i) = [(y(i-1:-1:i-3)) , (u(i-1:-1:i-3))];
    K = P_sys*phi(:,i)*(1+phi(:,i)'*P_sys*phi(:,i))^-1 ;
    P_sys = (eye(6) - K*phi(:,i)')*P_sys ;
    theta(:,i) = theta(:,i-1) + K*(y(i) - phi(:,i)'*theta(:,i-1));

    A=[1 -theta(1:3 ,end)'];
    B=theta(4:6,end)';

    if i>30
        U = uf(i-d0:-1:i-3) ;
        V = [yf(i-d0:-1:i-3) , -ucf(i-d0)] ;
        Y = y(i)-y_ref(i) ;
        [teta , P] = RLS_1(U , V , Y , teta , P , Nv) ;

        R_h = teta(1:3)' ;
        S_h = teta(4:6)' ;
        t0 = teta(7)' ;
        T=t0*A0;

        u(i) = (-u (i-1:-1:i-2)*R_h(2:end)' + uc(i:-1:i-2)*T' - y(i:-1:i-2)*S_h')/R_h(1) ;
        yf(i) = -yf (i-1:-1:i-Na0am) *A0Am(2:end)' + y (i:-1:i-2)*B' ;
        uf(i) = -uf (i-1:-1:i-Na0am) *A0Am(2:end)' + u (i:-1:i-2)*B' ;
        ucf(i) = -ucf(i-1:-1:i-Na0am+2) *Am(2:end)' + uc(i:-1:i-2)*B' ;
    end
end

```

plot results

input and output

```

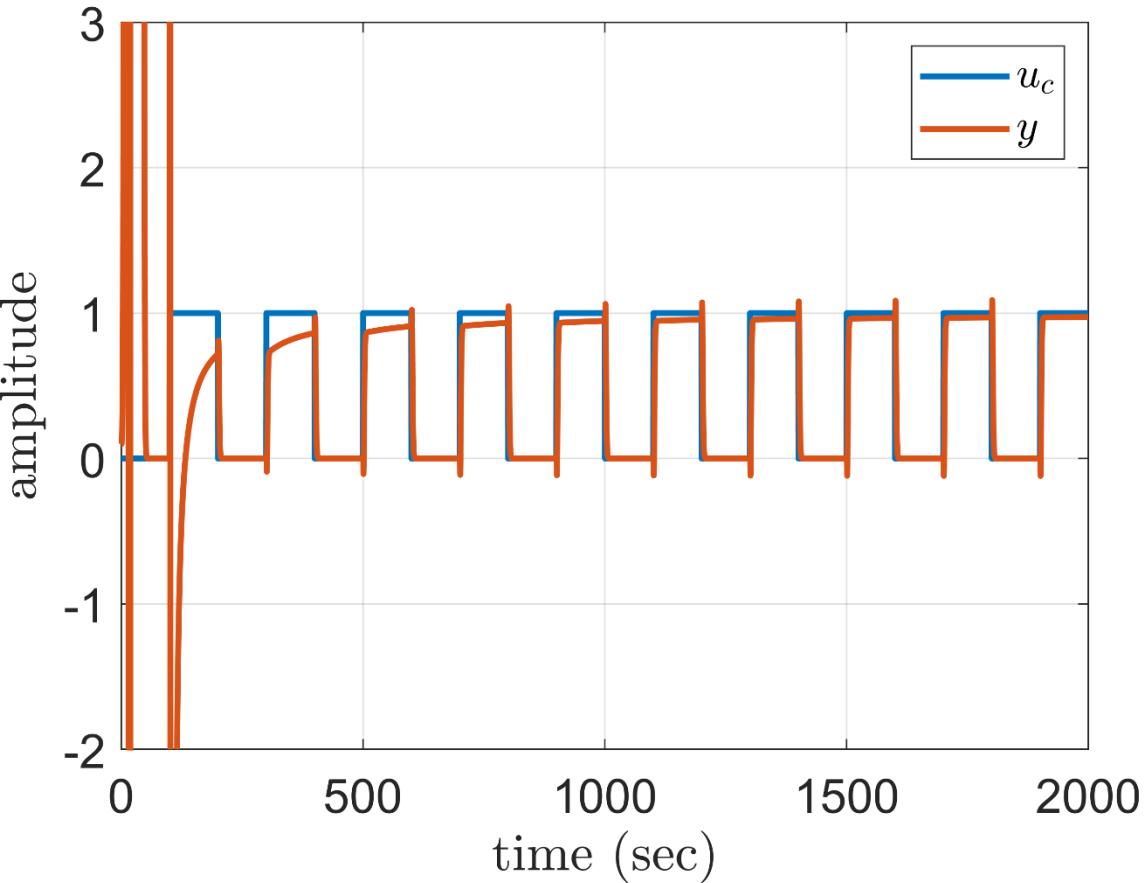
plot(t,uc,t,y , 'LineWidth' , 2) ;
xlabel('time (sec)', 'Interpreter', 'latex') ;

```

```

ylabel('amplitude','Interpreter','latex') ;
legend('$$u_c$$','$$y$$','Interpreter','latex','Location','northeast')
set(gca,'FontSize',16)
axis([0 tfinal -2 3])
grid on

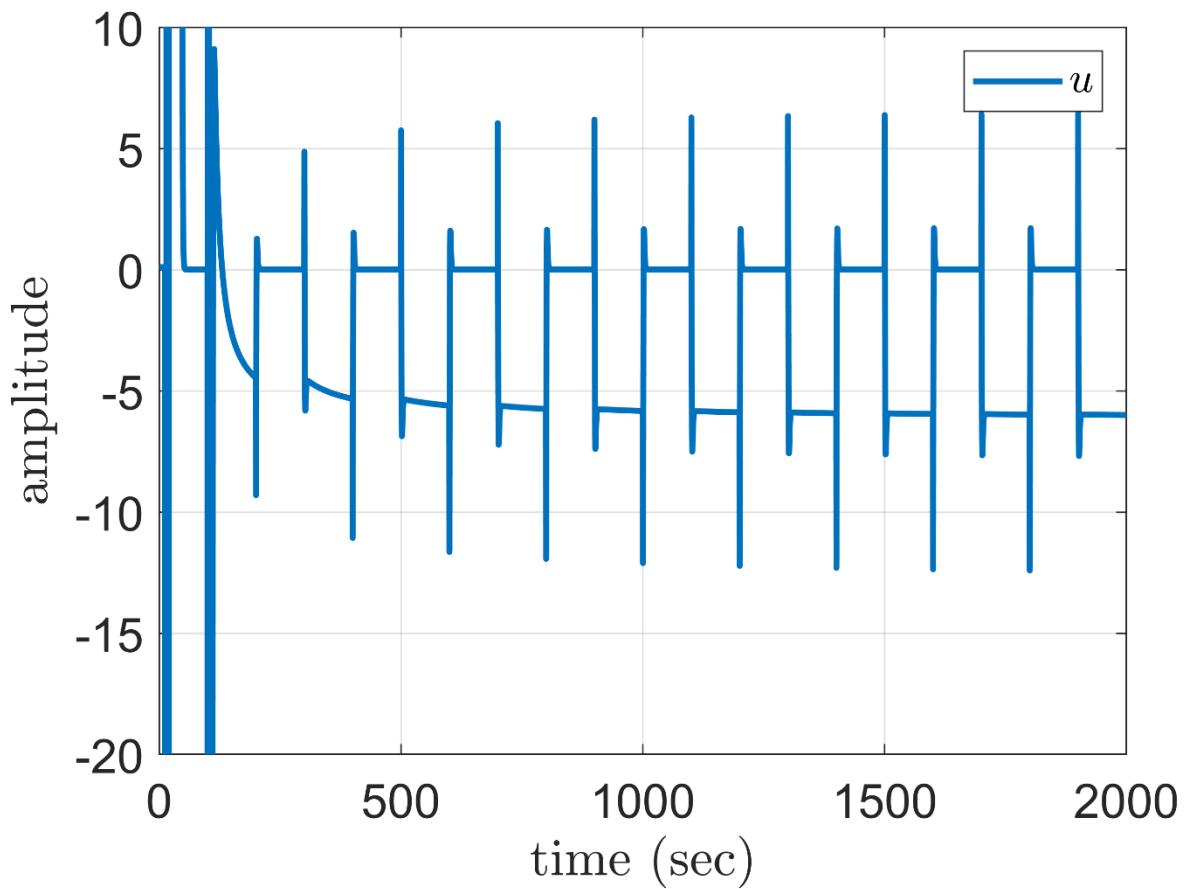
```



```

plot(t , u, 'LineWidth' , 2) ;
xlabel('time (sec)','Interpreter','latex') ;
ylabel('amplitude','Interpreter','latex') ;
legend('$$u$$','Interpreter','latex','Location','northeast')
set(gca,'FontSize',16)
axis([0 Tf -4 4])
grid on

```



```

function [teta , P] = RLS_1(U , V ,Y, teta , P , Nv)
U = U(:)' ;
V = V(:)' ;
phi = [U , V]' ;
K = P*phi*(1+phi'*P*phi)^(-1) ;
P = (eye(Nv) - K*phi')*P ;
teta = teta + K*(Y - phi'*teta ) ;
end

```

۴ STR با جایابی قطب برای سیستم پیوسته

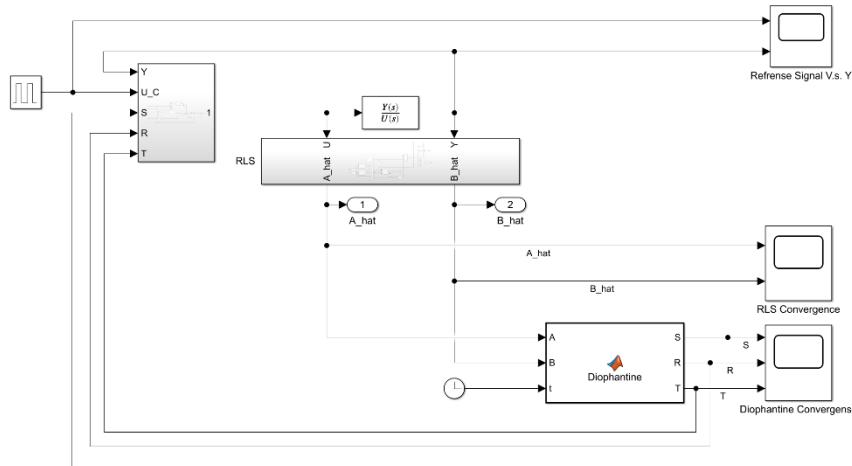
برای سیستم زیر کنترل کننده جایاب قطب تطبیقی با RLS پیوسته در محیط سیمولینک طراحی کنید. (بدون حذف صفر و قطب) (راهنمایی: از مثال‌های ۳-۳ و ۶-۳ کتاب کمک بگیرید)

$$G_s = \frac{3}{(s + 0.5)(s + 2)}$$

با اجرای $Q=10.m$ پارامترهای اولیه در ورک اسپیس ایجاد و مدل سیمولینک اجرا می‌شند.

در شکل ۱-۴ کلیت مدل سیمولیک آورده شده است.

قدم اول شناسایی پارامترهای صورت و مخرج توسط روش حداقل مربعات پوسته می‌باشد سپس پارامترهای شناسایی شده به همراه قطب‌های تابع تبدیل مطلوب به تابع حل معادله دیوفانتین فرستاده می‌شوند و در نهایت قانون کنترلی را تشکیل داده و حلقه کنترلی کامل می‌شود.



شکل ۱-۴ شکل کلی شناسایی سیستم به صورت پیوسته در سیمولینک

درون بلوك شناسایی پارامترها از روابط زیر برای تشکیل

$$A(p) = p^r + a_1 p + a_r$$

$$B(p) = b_1 p + b_r$$

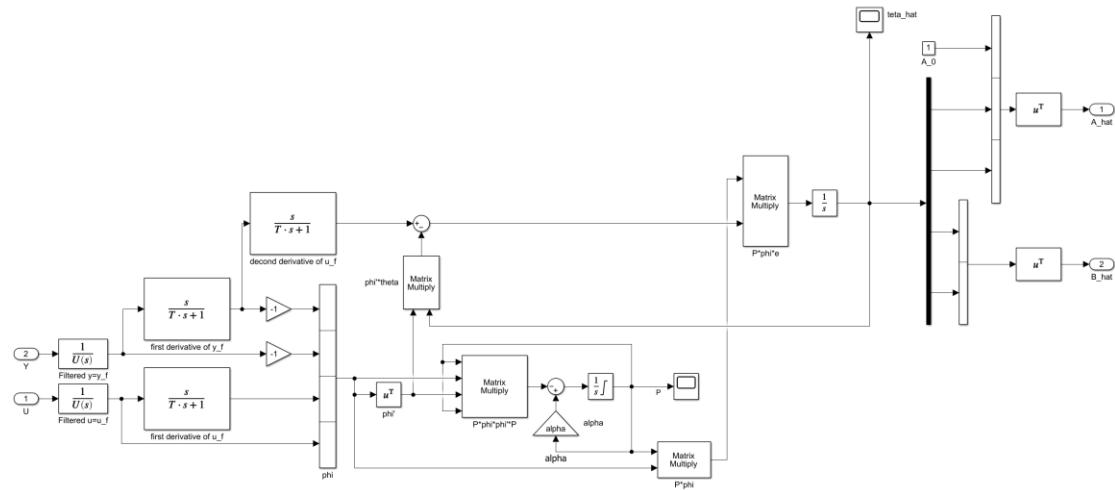
$$y_f = H_f y(t), u_f = H_f u(t), H_f = \frac{1}{A_m}$$

$$p^n y_f(t) = \phi^T(t) \theta$$

$$\frac{d\hat{\theta}(t)}{dt} = P(t)\phi(t) \left(P^n y_f(t) - \phi^T \hat{\theta}(t) \right), \frac{dP(t)}{dt} = \alpha P(t) - P(t)\phi(t)\phi^T P(t)$$

$$\theta = [a_1 \quad a_r \quad b_1 \quad b_r]$$

درون قسمت شناسایی به صورت زیر است.

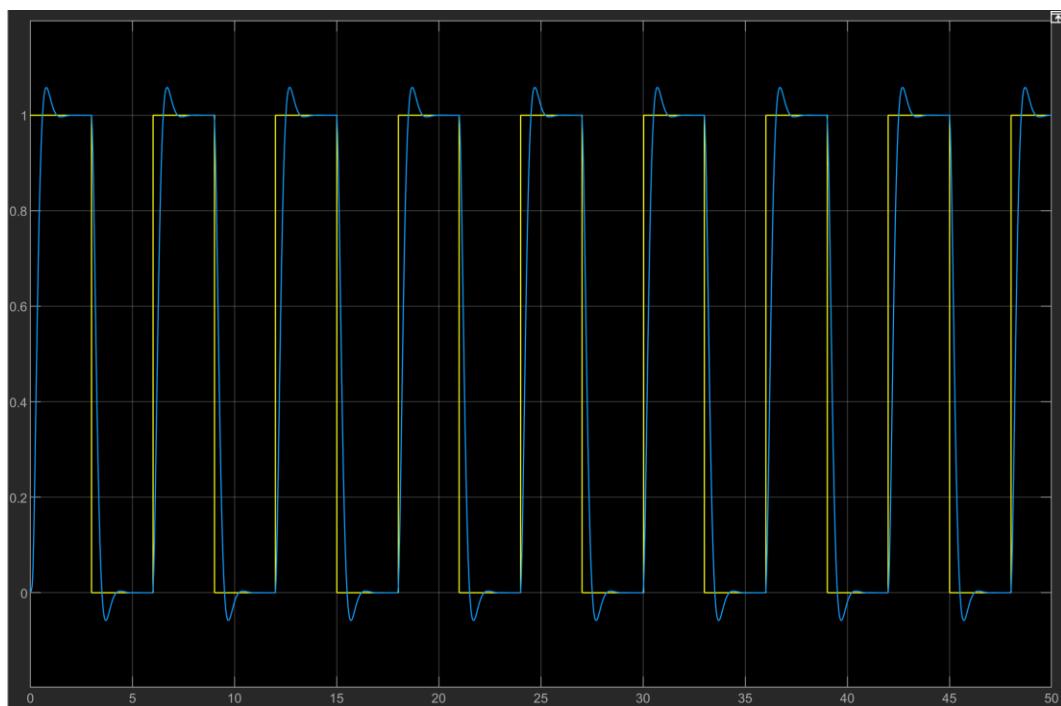


شکل ۲-۴ قسمت شناسایی سیستم.

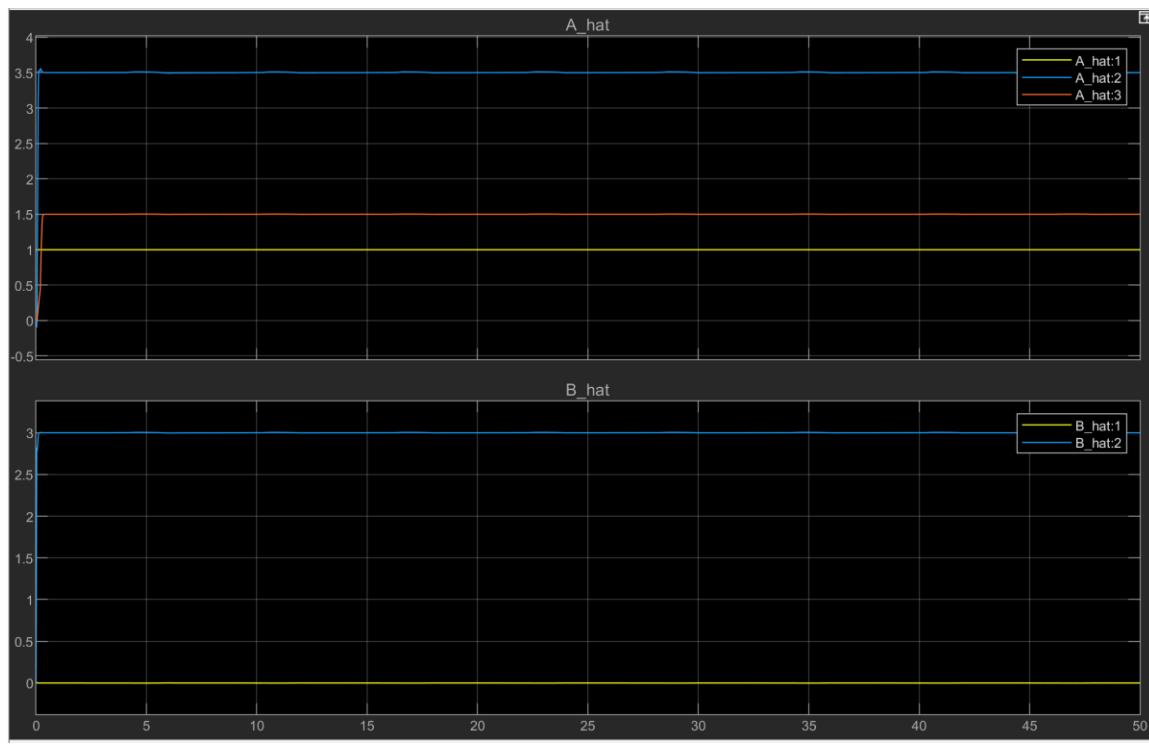
سیمولینک اجازه تعریف تابع تبدیل مشتق را نمی‌دهد و تابعی مرتبه یک با ضرب مخرج بسیار کوچک استفاده شده است.(تابع مشتق خود سیمولینک نویز را تشدید می‌کند).

$$\text{derivative} := \frac{s}{10^{-4}s + 1} \approx \frac{s}{1}$$

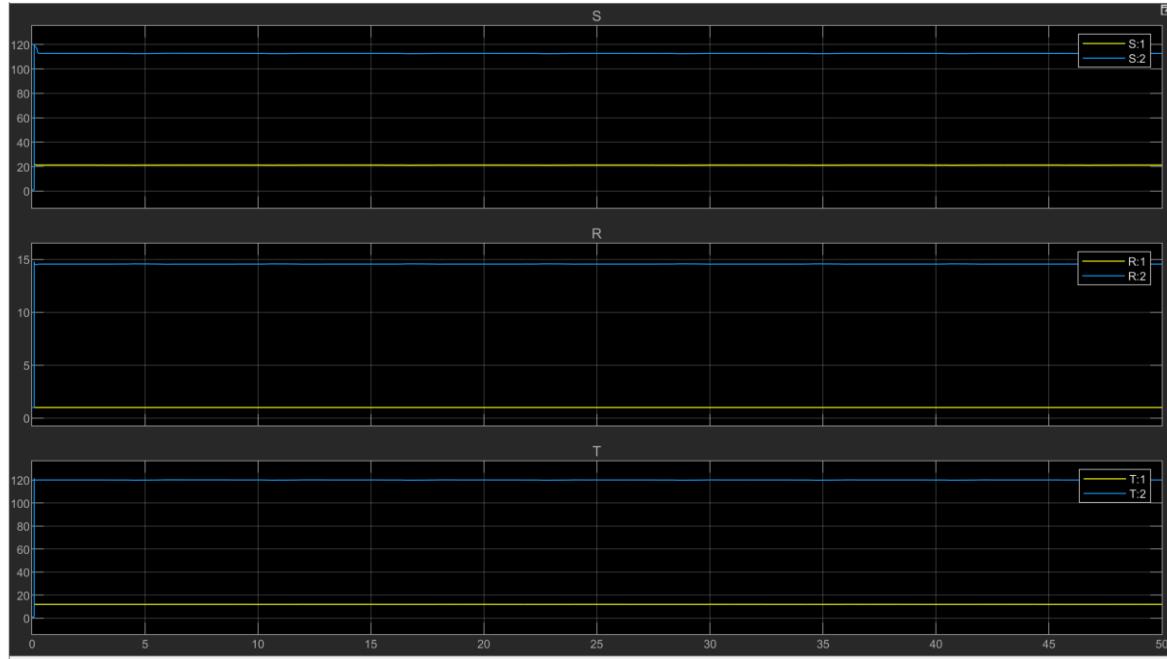
خروجی سیمولینک:



شکل ۳-۴ خروجی سیستم و ورودی مرجع



شکل ۴-۴ همگرایی حداقل مربعات بازگشتی پیوسته



شکل ۴-۵ پاسخ‌های معادله دیوفانتین

پیوست ۱ : کدهای متلب ۵

۱ - ۵ دیوفانتین

```
function [R , S ] = Diophantine(A , B , Ac)
    %%Logical Definitions
    ord_A = numel(A)-1 ;      ord_B = numel(B)-1 ;      ord_Ac = numel(Ac)-1 ;

    B = [zeros(1,ord_A-ord_B)      , B]      ;
    Ac = [zeros(1,2*ord_A-(ord_Ac+1)) , Ac]      ;
    E = zeros(2*ord_A      , 2*ord_A);
    %%      Silvester Matrix
    for i = 1:ord_A
        E(1:2*ord_A,i ) = [zeros(i-1,1); A';zeros(ord_A-i,1)];
        E(1:2*ord_A,i+ord_A) = [zeros(i-1,1); B';zeros(ord_A-i,1)];
    end
    %%      Result
    RS = E\ Ac';
    R = RS(1:ord_A);
    S = RS(ord_A+1:end);
end
```

۲ - ۵ 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
```