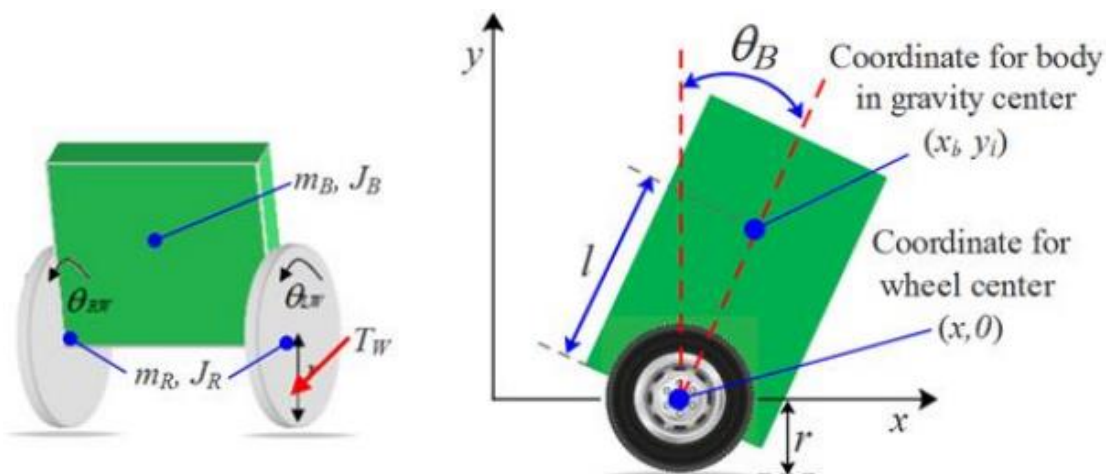


# شبیه سازی مقاله

علیرضا عباسی

# چکیده

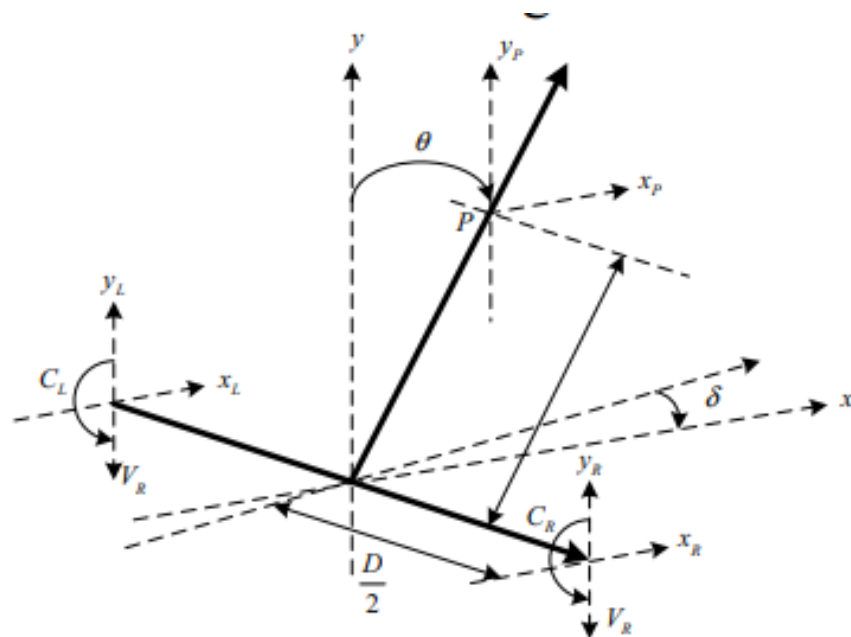
- در این مقاله، مساله ربات تعادلی با رویکرد کنترلر LQR حل شده. در ابتدا مدل ریاضی بدست آمده و به فضای حالت میرسیم. سپس کنترلر مناسب را طراحی میکنیم.



# متغیرهای حالت

♦ موقعیت و سرعت در جهت  $X$ ، زاویه پیچ و مشتق آن (سرعت زاویه‌ای)، زاویه چرخش و مشتق آن.

$$\begin{pmatrix} X_{RM} \\ V_{RM} \\ \theta_P \\ \omega_P \\ \delta \\ \dot{\delta} \end{pmatrix}$$



$X_{RM}, V_{RM}$	Level position, velocity
$\theta_P, \omega_P$	Angle, velocity of angle
$\delta$	Angle of turn

# خروجی فضای حالت

$$y = \begin{bmatrix} V_{RM} & \theta_P & \dot{\delta} \end{bmatrix}^T$$

$X_{RM}, V_{RM}$	Level position, velocity
$\theta_P, \omega_P$	Angle, velocity of angle
$\delta$	Angle of turn

# کمیت‌های ورودی فضای حالت

$$\begin{pmatrix} C_L \\ C_R \end{pmatrix}$$

$$\begin{pmatrix} C_\theta \\ C_\delta \end{pmatrix}$$

$C_R, C_L$

Driven  
torque of  
wheels

# استفاده از دیکوپلینگ

- از آنجا که  $C_L$  and  $C_R$  بطور همزمان روی خروجی اثر میگذارند. مجبوریم برای از بین بردن این مشکل از دیکوپلینگ زیر استفاده کنیم:

system,  $C_\theta$  and  $C_\delta$  are expressed as inputs that  $C_\theta$  is the input torque produced as TWSBR rotates around Z-axis and  $C_\delta$  is that produced as TWSBR rotates around Y-axis, the decoupling control elements are designed and shown in Fig.3.

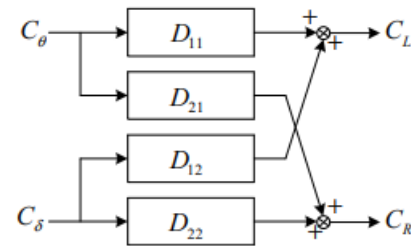


Fig.3 Decoupling Control Element of TWSBR

# معادلات ریاضی مدل

$$\ddot{X}_{RM} = \frac{1}{J_P \left( 2 \left( J_W / R^2 + M_W \right) + M_P \right) + 2 \left( J_W / R^2 + M_W \right) M_P L^2 + M_P^2 L^2 \sin^2 \theta_P} \cdot \left[ \left( J_P + M_P L^2 \right) M_P L \sin \theta_P \dot{\theta}_P^2 - M_P^2 L^2 g \cos \theta_P \sin \theta_P + \left( \frac{1}{R} \left( J_P + M_P L^2 \right) + M_P L \cos \theta_P \right) (C_R + C_L) + \left( J_P + M_P L^2 (f_{RR} + f_{RL}) \right) \right]$$

$$\ddot{\theta}_P = - \frac{1}{J_P \left( 2 \left( J_W / R^2 + M_W \right) + M_P \right) + 2 \left( J_W / R^2 + M_W \right) M_P L^2 + M_P^2 L^2 \sin^2 \theta_P} \cdot \left[ M_P^2 L^2 \sin \theta_P \cos \theta_P \dot{\theta}_P^2 + \left( 2 \left( J_W / R^2 + M_W \right) + M_P \right) M_P g L \sin \theta_P - \left( M_P L \cos \theta_P / R + 2 \left( J_W / R^2 + M_W \right) + M_P \right) (C_R + C_L) - M_P L \cos \theta_P (f_{RR} + f_{RL}) + 2 \left( J_W / R^2 + M_W \right) L \cos \theta_P f_P \right]$$

$$\ddot{\delta} = [(C_L - C_R) / R + (f_L - f_R)] / (2J_P M_W + J_W D / R^2)$$

# خطی سازی

$$\sin \theta_p \approx \theta_p, \cos \theta_p \approx 1$$

$$A_{23} = \frac{-M_p^2 L^2 g}{M_p J_p + 2(J_p + M_p L^2)(M_w + J_w/R^2)}$$

$$A_{43} = \frac{M_p^2 g L + 2M_p g L (M_w + J_w/R^2)}{M_p J_p + 2(J_p + M_p L^2)(M_w + J_w/R^2)}$$

$$B_{21} = B_{22} = \frac{(J_p + M_p L^2)/(R + M_p L)}{M_p J_p + 2(J_p + M_p L^2)(M_w + J_w/R^2)}$$

$$B_{41} = B_{42} = \frac{-(R + L) M_p / R - 2(M_w + J_w/R^2)}{M_p J_p + 2(J_p + M_p L^2)(M_w + J_w/R^2)}$$

$$B_{61} = B_{62} = \frac{D/2R}{J_p + \frac{D^2}{2R} \left( M_w R + \frac{J_w}{R} \right)}$$



# فضای حالت

$$\dot{x} = Ax + Bu = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & A_{23} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & A_{43} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} X_{RM} \\ V_{RM} \\ \theta_P \\ \omega_P \\ \delta \\ \dot{\delta} \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ B_{21} & B_{22} \\ 0 & 0 \\ B_{41} & B_{42} \\ 0 & 0 \\ B_{61} & B_{62} \end{pmatrix} \begin{pmatrix} C_L \\ C_R \end{pmatrix}$$

$$y = Cx = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_{RM} \\ V_{RM} \\ \theta_P \\ \omega_P \\ \delta \\ \dot{\delta} \end{pmatrix}$$

# فضای حالت بعد از دیکوپلینگ

$$\begin{pmatrix} \dot{X}_{RM} \\ \dot{V}_{RM} \\ \dot{\theta}_P \\ \dot{\omega}_P \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & A_{23} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & A_{43} & 0 \end{pmatrix} \begin{pmatrix} X_{RM} \\ V_{RM} \\ \theta_P \\ \omega_P \end{pmatrix} + \begin{pmatrix} 0 \\ B_{21} \\ 0 \\ B_{41} \end{pmatrix} C_\theta$$

$$\begin{pmatrix} \dot{\delta} \\ \ddot{\delta} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \delta \\ \dot{\delta} \end{pmatrix} + \begin{pmatrix} 0 \\ B_{61} \end{pmatrix} C_\delta$$

Let  $D_{11} = D_{21} = D_{12} = D_{22} = 0.5$ , then

$$\begin{pmatrix} C_L \\ C_R \end{pmatrix} = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{pmatrix} \begin{pmatrix} C_\theta \\ C_\delta \end{pmatrix} \quad ($$

# بررسی کنترل پذیری و رویت پذیری

```
%% Controllability and Observability
```

```
Ct = rank(ctrb(sys));
```

```
Ob = rank(observ(sys));
```

Ct =

4

Ob =

4

نتیجه میگیریم که سیستم بطور کامل رویت پذیر و کنترل پذیر حالت نیست

# نمایش مودها

$$\bar{A} = \begin{bmatrix} A_{no} & A_{12} \\ 0 & A_o \end{bmatrix}$$

Abaro =

0	0	0	0	0	1.0000
0	0	0	-1.0000	0	0
0	0	0	0	-60.5288	0
0	0	0	0	0	0
0	0	-1.0000	0	0	0
0	0	0	0	-6.3715	0

```
%% Observable Modes
```

```
[Abaro Bbaro Cbaro To ko] = obsvf(A, B, C);
```

```
%% Controllable Modes
```

```
[Abarc, Bbarc, Cbarc, Tc, kc] = ctrbf(A, B, C);
```

Abarc =

-0.4665	0.3202	-0.0000	-0.0000	0.0000	-0.0000
-0.6798	0.4665	-0.0000	0.0000	-0.0000	0.0000
-0.0000	0.0000	0.0000	1.0000	0.0000	0.0000
0.2461	0.3586	0.2809	0.0000	7.5671	0.0000
0.0000	-0.0000	-0.0000	-0.0000	-0.0000	1.0000
1.9593	2.8550	2.2364	0.0000	60.2479	-0.0000

$$\bar{A} = \begin{bmatrix} A_{uc} & 0 \\ A_{21} & A_c \end{bmatrix}$$

# نمایش مینیمال

3 states removed.

>> sysr

sysr =

A =

	x1	x2	x3
x1	1.878e-61	0.9979	-0.03106
x2	60.77	-1.053e-61	-1.512e-60
x3	3.451	-3.072e-62	-8.511e-62

B =

	u1	u2
x1	-3.073e-64	-3.073e-64
x2	-58	-58
x3	3.963	3.963

C =

	x1	x2	x3
y1	-4.595e-62	-0.05734	-0.8367
y2	1	-1.706e-63	-2.489e-62
y3	-2.489e-62	-0.03106	0.5469

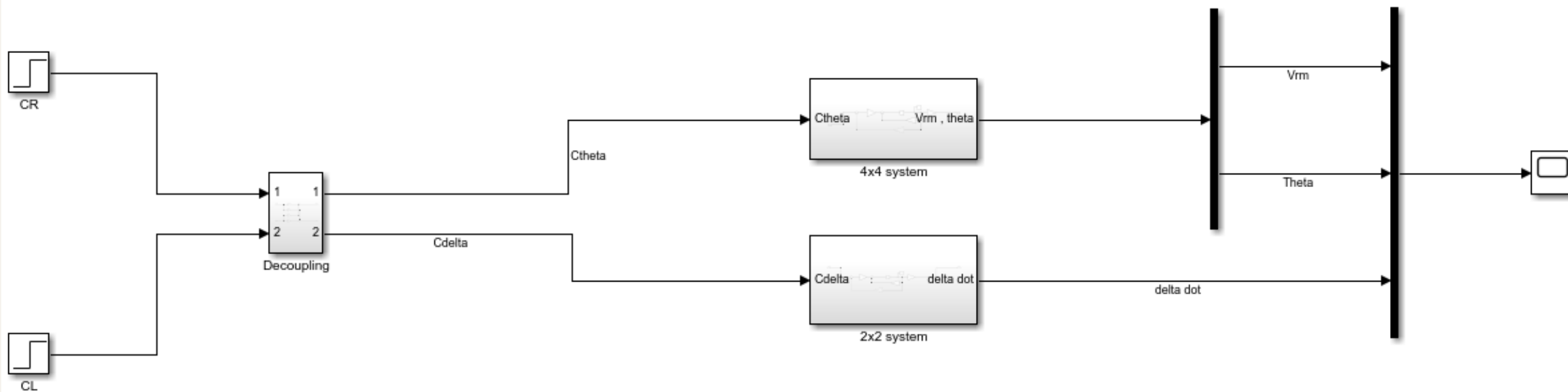
D =

	u1	u2
y1	0	0
y2	0	0
y3	0	0

Continuous-time state-space model.

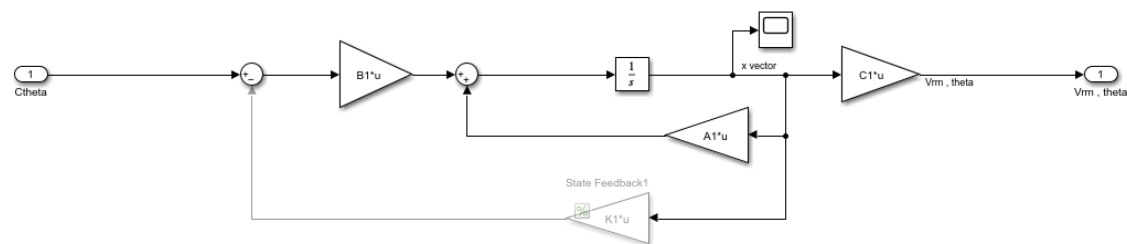
```
%% minimal form for the system  
sysr = minreal(sys);
```

# بلوک دیاگرام شبیه سازی

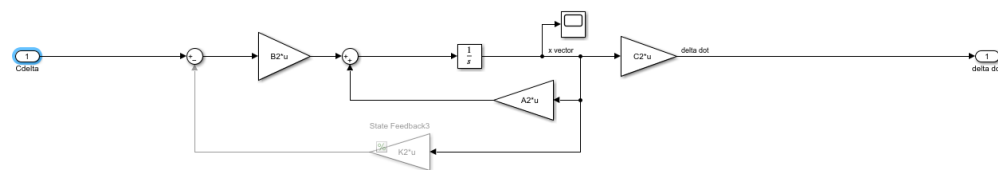


# فضاهای حالت

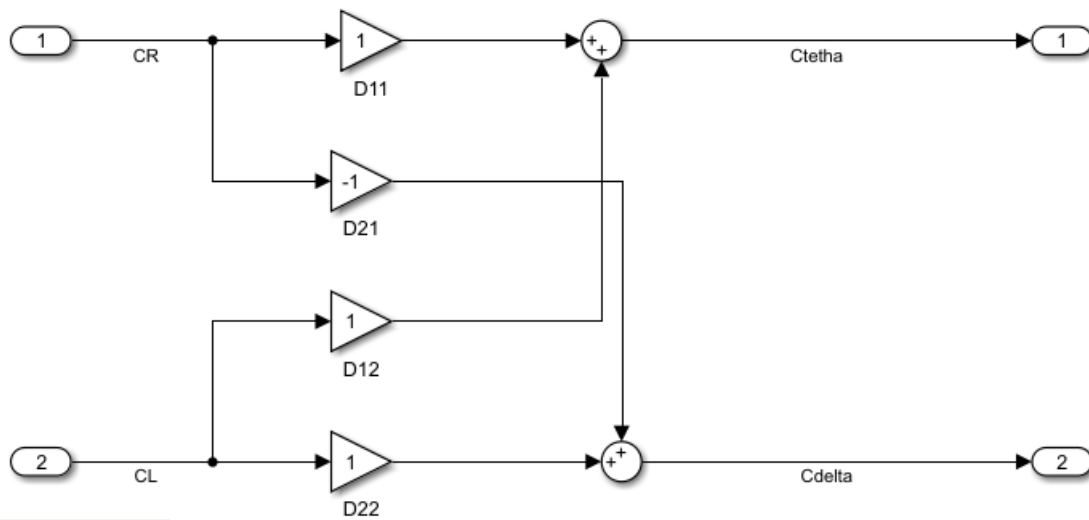
4x4 state space



2x2 state space



# دیکوپلینگ ورودی



$$\begin{bmatrix} C_L \\ C_R \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{bmatrix} \begin{bmatrix} C_\theta \\ C_\delta \end{bmatrix}$$

$$\rightarrow C_L = 0.5 C_\theta + 0.5 C_\delta \rightarrow C_\theta = 2C_L - C_\delta$$

$$C_R = 0.5 C_\theta - 0.5 C_\delta \rightarrow C_\theta = 2C_R + C_\delta$$

$$2C_L - C_\delta = 2C_R + C_\delta$$

$$\rightarrow C_\delta = C_L - C_R$$

$$\Rightarrow C_\theta = C_L + C_R$$

$$\Rightarrow \begin{bmatrix} C_\theta \\ C_\delta \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} C_L \\ C_R \end{bmatrix}$$

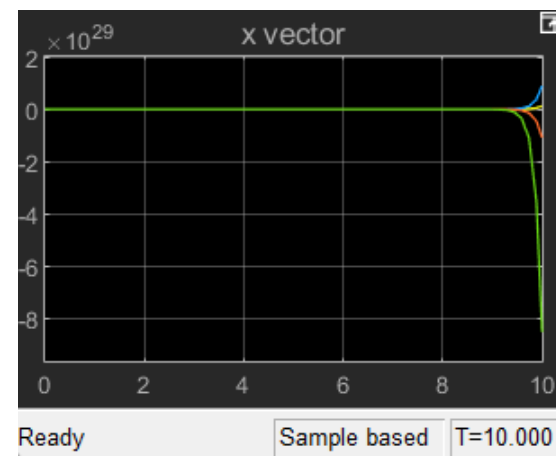
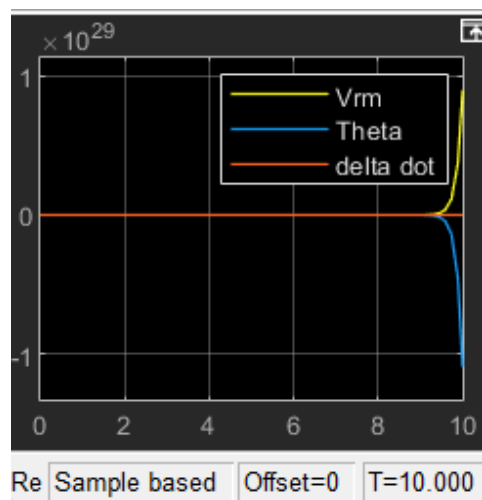
$$\Rightarrow \begin{array}{c} C_R \rightarrow \boxed{1} \rightarrow \oplus \rightarrow C_\theta \\ \quad \quad \quad \boxed{-1} \rightarrow \oplus \\ C_L \rightarrow \boxed{1} \rightarrow \oplus \rightarrow C_\delta \\ \quad \quad \quad \boxed{1} \rightarrow \oplus \end{array}$$



# پاسخ سیستم بدون فیدبک

متغیرهای حالت

خروجی



# طراحی فیدبک حالت

```
%% State Feedback
```

```
K1 = place(A1, B1, [-2 -3 -4 -5]);
```

```
K2 = place(A2, B2, [-1 -3]);
```

```
>> K1
```

```
K1 =
```

```
-15.0287 -19.2868 -58.1495 -9.0291
```

```
>> K2
```

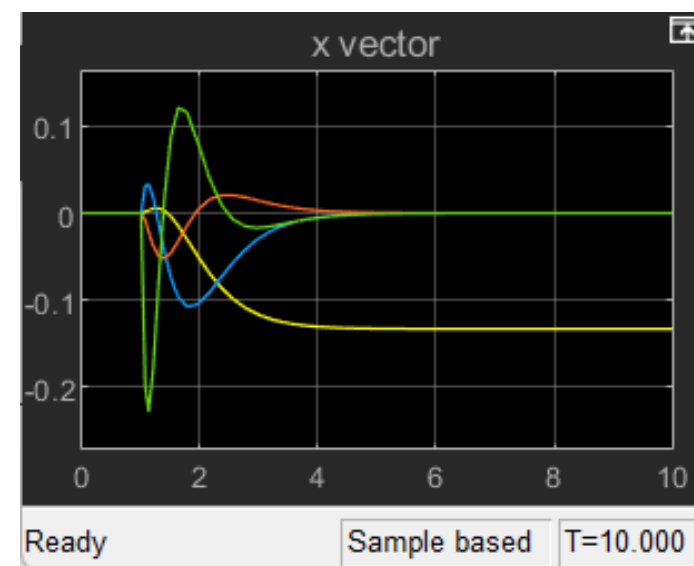
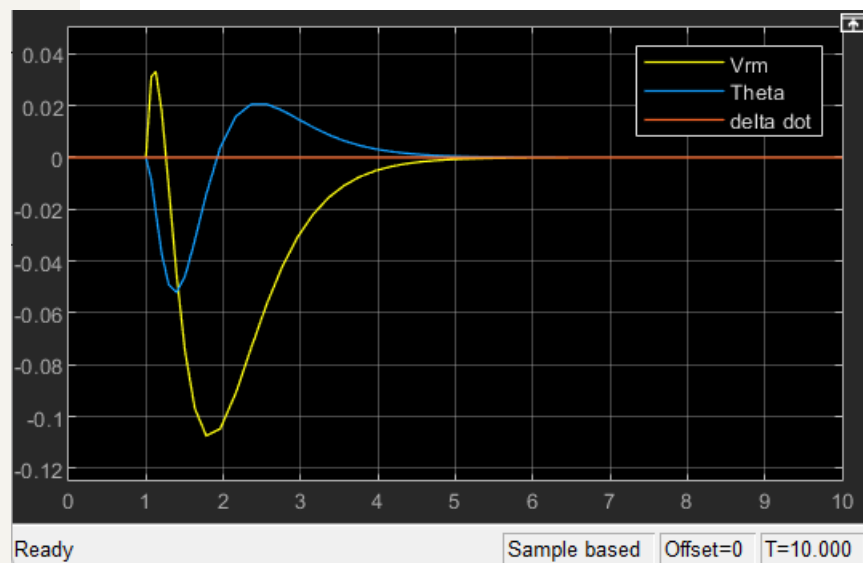
```
K2 =
```

```
5.8997 7.8663
```

# پاسخ سیستم با فیدبک

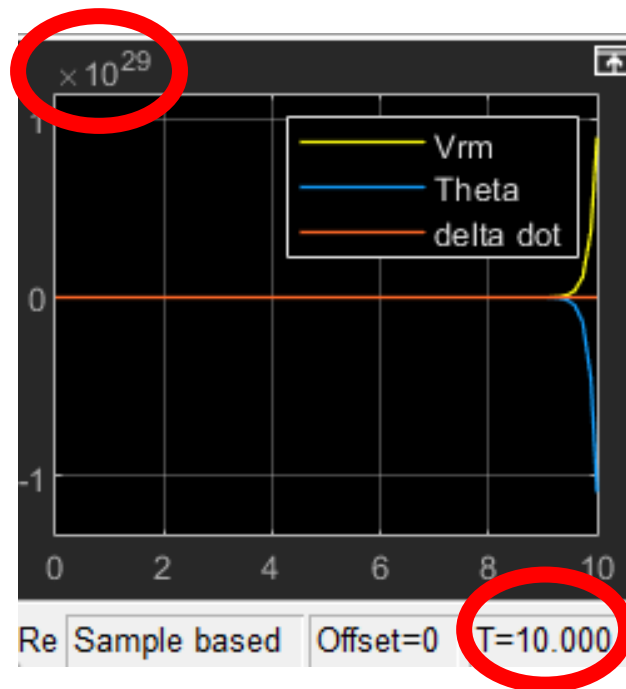
متغیرهای حالت

خروجی



# پایداری

- همانطور که مشاهده میشود سیستم نسبت به حلقه باز بسیار پایدارتر شده. چون که قطب‌های سیستم اولیه ناپایدار بودند.



# طراحی فیدبک حالت برای جانمایی قطب‌های دور

```
K1 = place(A1, B1, [-20 -30 -40 -50]);  
K2 = place(A2, B2, [-10 -30]);
```

K1 =

1.0e+05 \*

-1.5029   -0.1929   -0.2722   -0.0316

>> K2

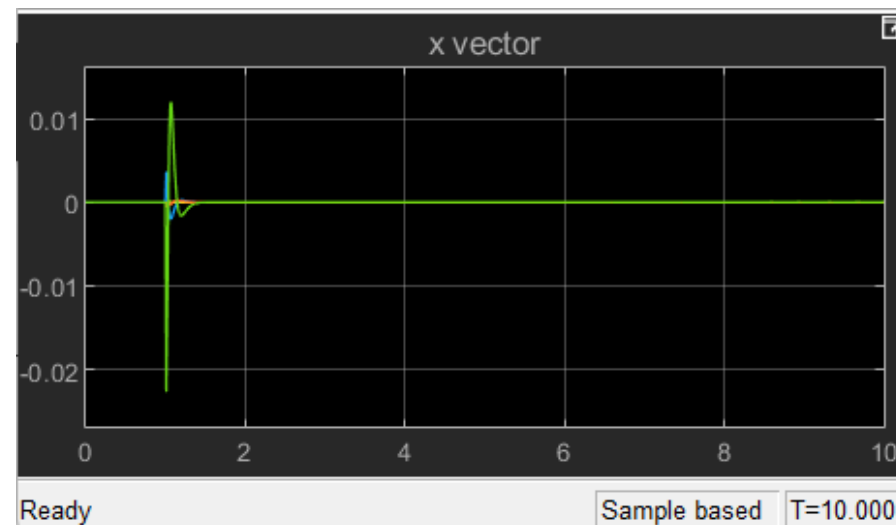
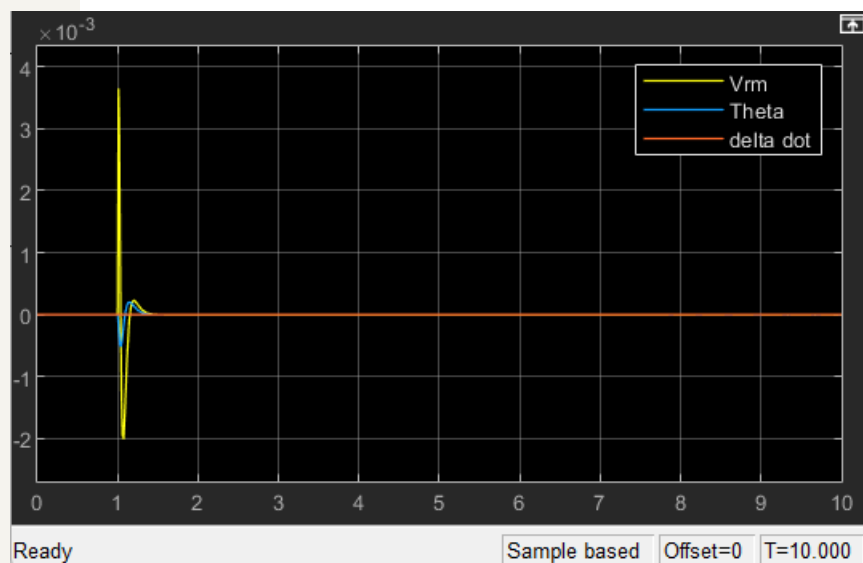
K2 =

589.9705   78.6627

# فیدبک حالت برای قطب‌های دور

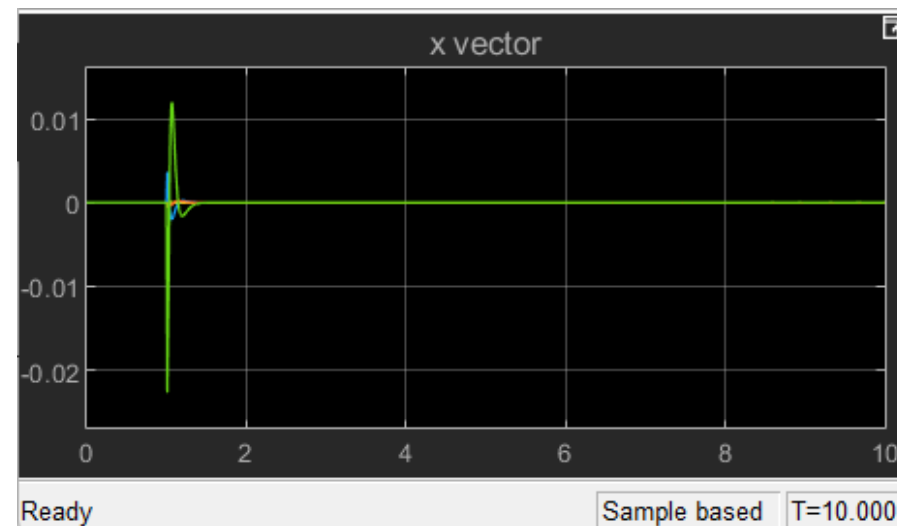
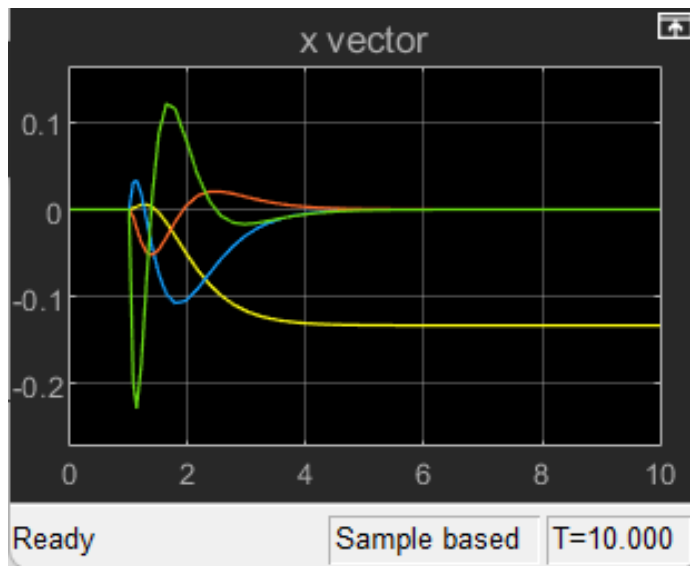
متغیرهای حالت

خروجی



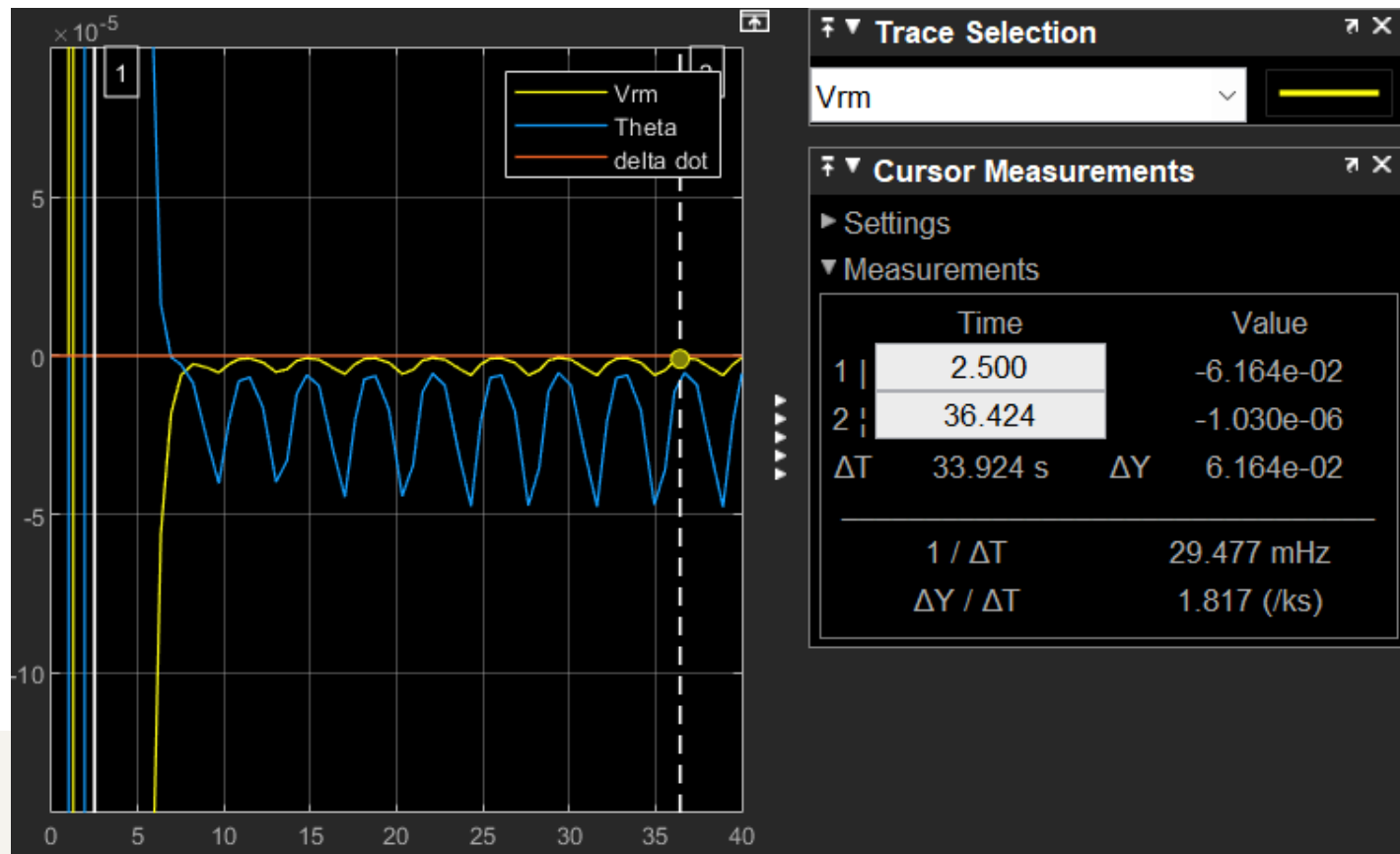
# سرعت سیستم

- همانطور که مشاهده میشود سیستم بسیار سریعتر شده و به مقدار نهایی خود رسیده. همچنین یکی از حالت ها که به مقدار صفر نرسیده بود در قطب نزدیک، با قطب دور به سرعت به steady state رسیده است.



# طراحی ردیاب استاتیک

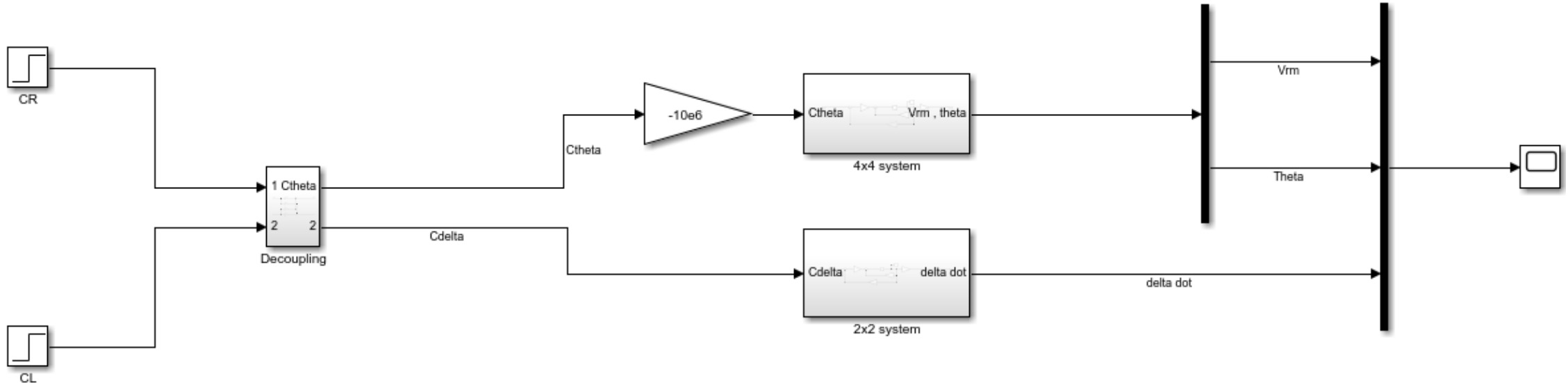
- در سیستم پایدار (با فیدبک حالت) بنظر میرسد که سیستم به حالت ثابت میرسد. با مقدار تقریبی خروجی در بینهایت، یک ردیاب استاتیک طراحی میکنیم:



- گین تقریباً برابر با  $1e6$  - در نظر میگیریم

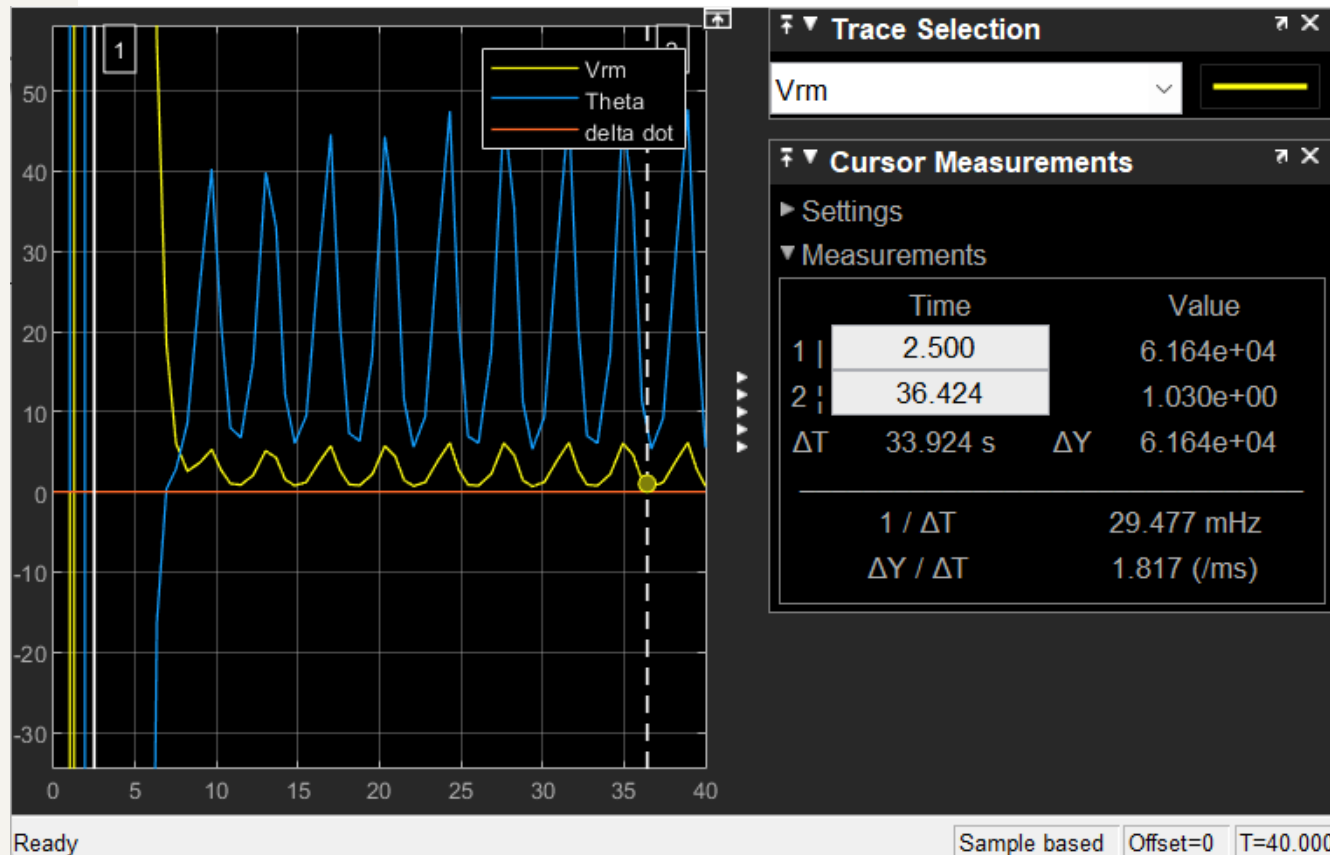


# قرار دادن ردیاب استاتیکی



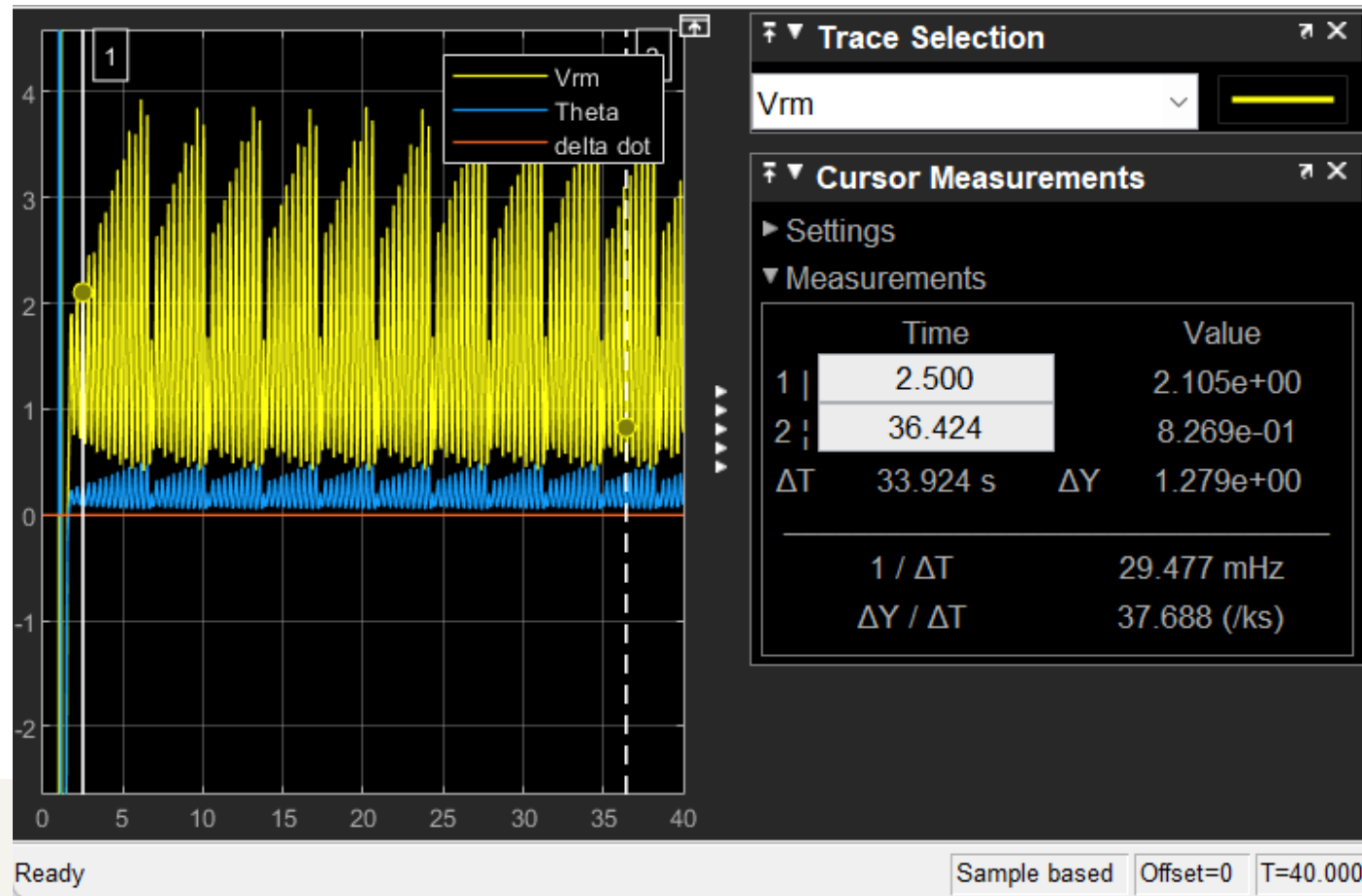
# پاسخ سیستم با ردیاب استاتیکی

♦ می‌بینیم که مساله ردیابی به خوبی تحقق نیافته است.

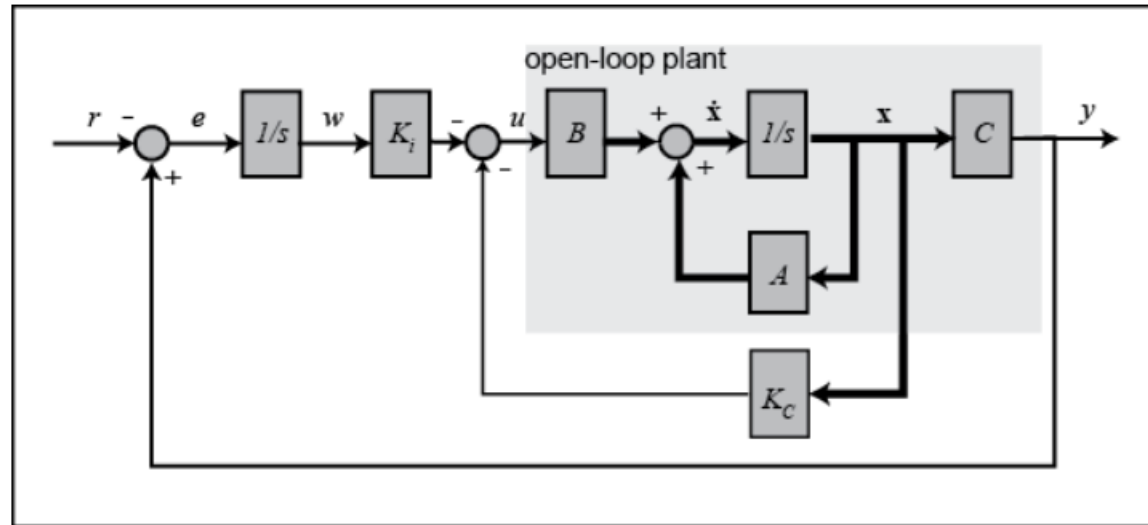


# از قطب‌های سریعتر استفاده میکنیم

- می‌بینیم که ردیابی بهتر از حالت قبل رخ داده است و با فیلتر پایین گذر احتمالاً به پاسخ دلخواه می‌توانیم برسیم



# ردیاب انتگرالی



- میبینیم که ماتریس  $A\_new$  فول رنک نیست. بنابراین نمیتوان از ردیاب انتگرالی استفاده کرد.

---

```
%% Robust Tracking  
A_new = [A1 zeros(4,2); C1 zeros(2,2)];  
B_new = [B1; zeros(2,1)];  
C_new = [C1 zeros(2,1)];  
Ki = place(A_new, B_new, [-2 -3 -4 -5 -6 -7]);
```

```
>> rank(A_new)
```

```
ans =
```

```
3
```

# طراحی تخمینگر مرتبه کامل

- مشاهده میشود چون سیستم اولیه رویت پذیر نیست، قادر به تعیین ماتریس  $L$  برای طراحی تخمینگر نیستیم. اگر رویت پذیر هم میشد باید قطب های تخمینگر را طوری جایابی میکردیم که اثرات نویز را تقویت نکند. و از فیلترهای بهینه استفاده میکردیم. (تئوری تخمین) ولی در نهایت حداقل باید ۴ الی ۵ برابر دورتر از قطب های سیستم اولیه جایابی کنیم تا سریعتر باشد.

```
127 %% full order observer
128 - L1 = place(A1', C1', [-20 -30 -40 -50]);
129 - L2 = place(A1', C1', [-5 -15]);
```

Command Window

Continuous-time state-space model.

3 states removed.

Error using **place** (line 171)

The "place" command could not place the poles at the specified locations. Probable causes include:

\* (A,B) is nearly uncontrollable

\* The specified locations are too close to each other.

Error in **Taklif** (line 128)

L1 = place(A1', C1', [-20 -30 -40 -50]);

```
>> rank(observ(A1, C1))
```

```
ans =
```

```
3
```

# طراحی تخمینگر کاهش یافته

♦ ماتریس C:

```
C1 =  
    0    1    0    0  
    0    0    1    0
```

♦ به فرم دلخواه نیست. پس، از تبدیل همانندی استفاده میکنیم:

```
P = [C1;  
      0, 0, 0, 1;  
      1, 0, 0, 0];
```

♦ ماتریس C\_new:

```
—  
C_new = C1/P;  
    1    0    0    0  
    0    1    0    0
```

# فضای حالت جدید

♦ بقیه ماتریس ها را بدست می آوریم:

```
A_new = P*A1/P;  
B_new = P*B1;
```

A\_new =

0	-6.3880	0	0
0	0	1.0000	0
0	60.6861	0	0
1.0000	0	0	0

>> B\_new

B\_new =

0.3803
0
-2.3629
0



# بدست آوردن بهره تخمینگر

♦ بعلت عدم مشاهده‌پذیری، بهره تخمینگر قابل دسترسی نیست

```
A22_new = A_new([3 4], [3 4]);  
A12_new = A_new([1 2], [3 4]);  
L = place(A22_new', A12_new', [-10 -20])';
```

Error using **place** (line 171)

The "place" command could not place the poles at the specified locations. Probable causes include:

- \* (A,B) is nearly uncontrollable
- \* The specified locations are too close to each other.

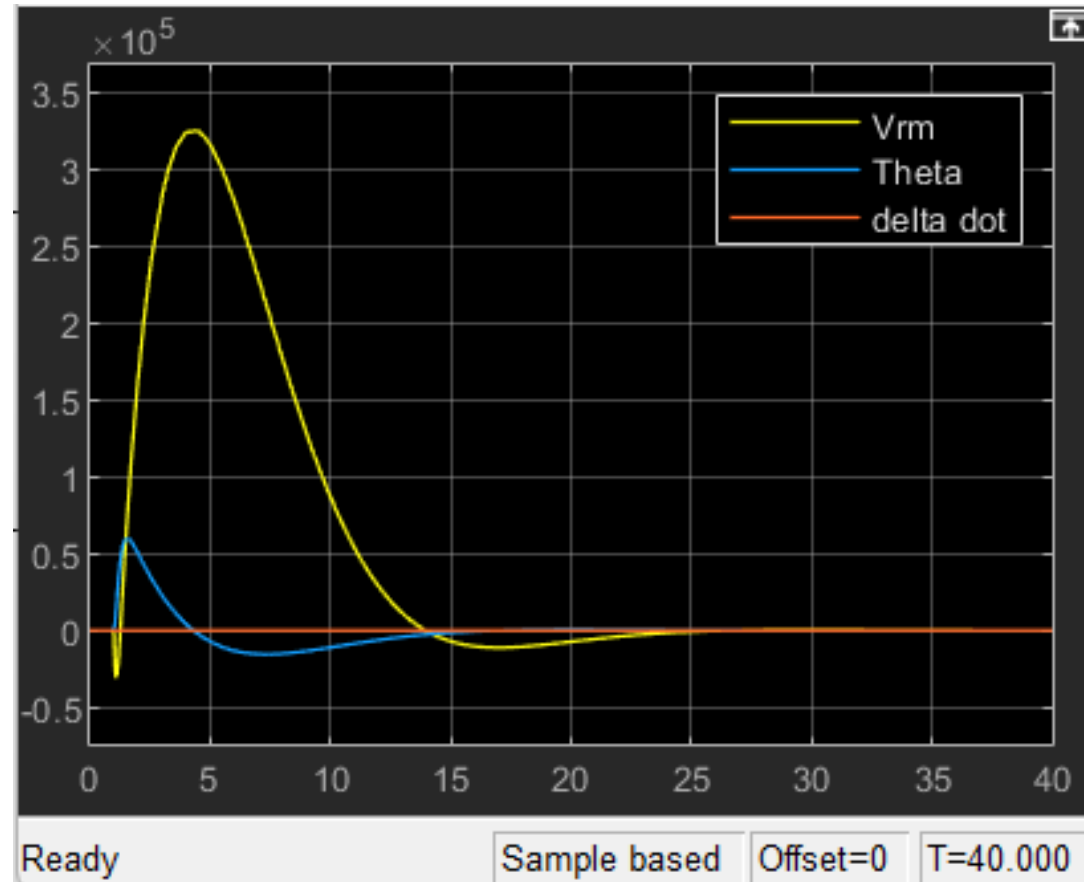
Error in **Taklif** (line 118)

```
L = place(A22_new', A12_new', [-10 -20])';
```

# Linear Quadratic Regulator

```
%% LQR state feedback
Q1 = eye(4);
R1 = 1;
K1 = lqr(A1, B1, Q1, R1);

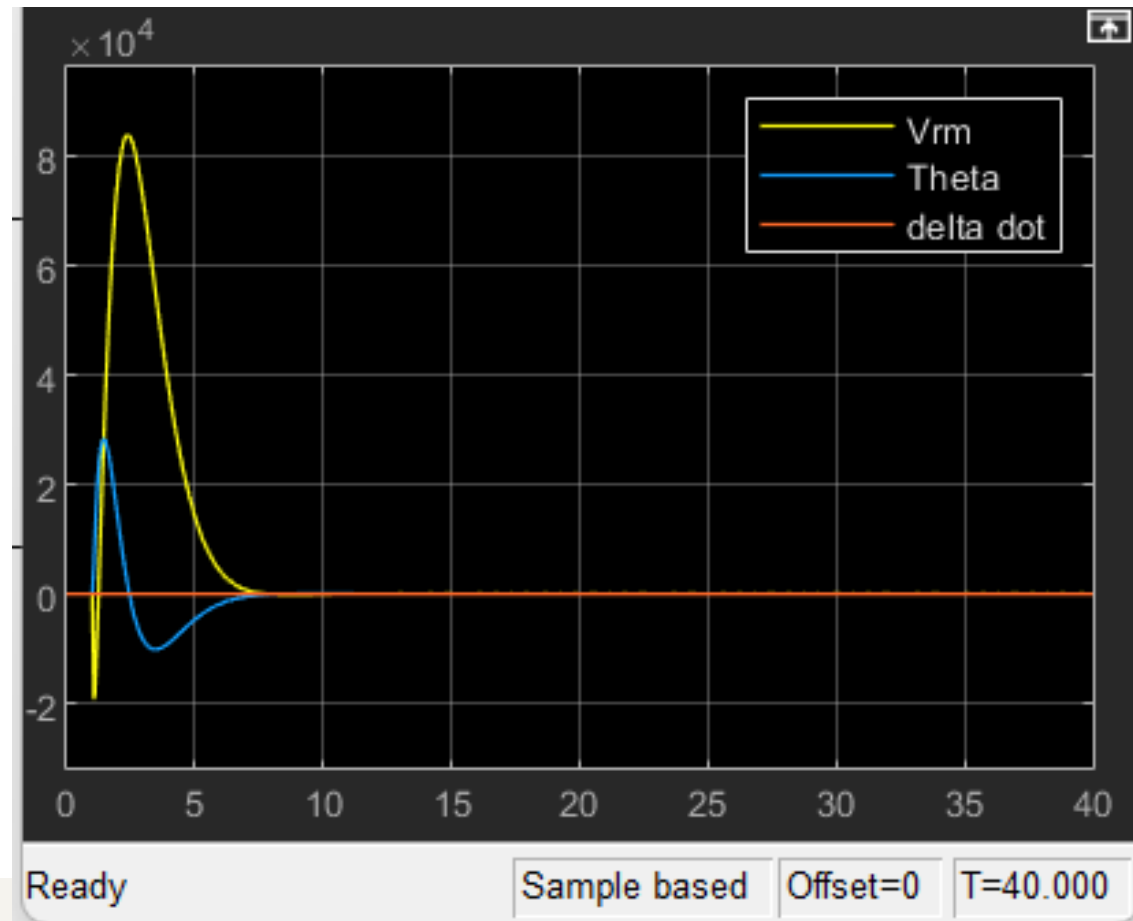
Q2 = eye(2);
R2 = 1;
K2 = lqr(A2, B2, Q2, R2);
```



# Linear Quadratic Regulator

```
%% LQR state feedback
Q1 = 100 * eye(4);
R1 = 1;
K1 = lqr(A1, B1, Q1, R1);

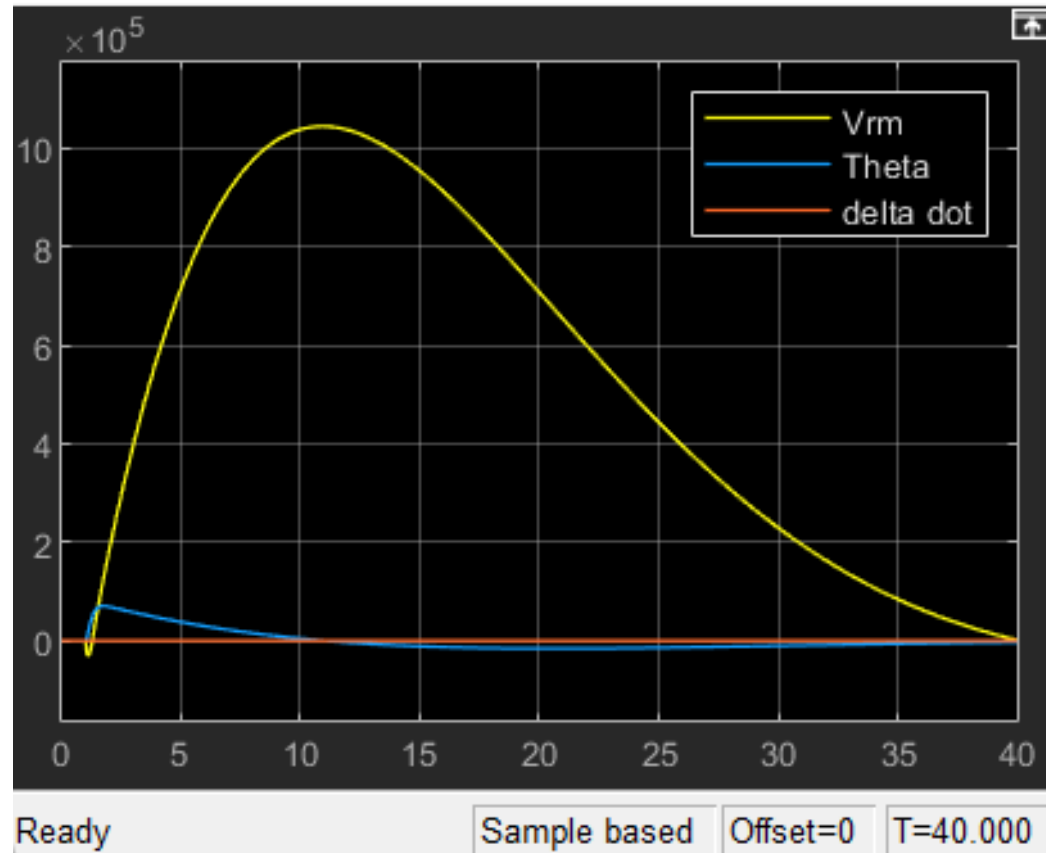
Q2 = 100 * eye(2);
R2 = 1;
K2 = lqr(A2, B2, Q2, R2);
```



# Linear Quadratic Regulator

```
%% LQR state feedback
Q1 = eye(4);
R1 = 100;
K1 = lqr(A1, B1, Q1, R1);

Q2 = eye(2);
R2 = 100;
K2 = lqr(A2, B2, Q2, R2);
```



# رویکرد مقاله برای کنترل ربات

- ♦ ابتدا به مدل خطی دست پیدا کرده
- ♦ سپس از شماتیک زیر برای کنترل بهره گرفته

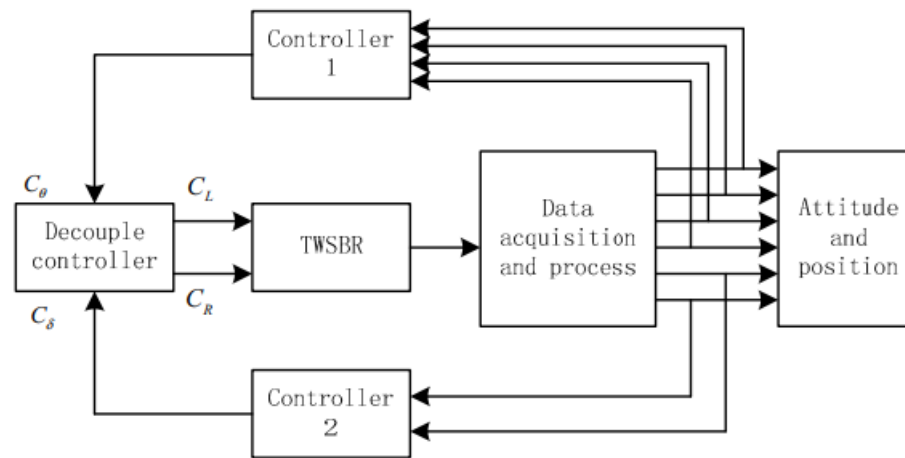
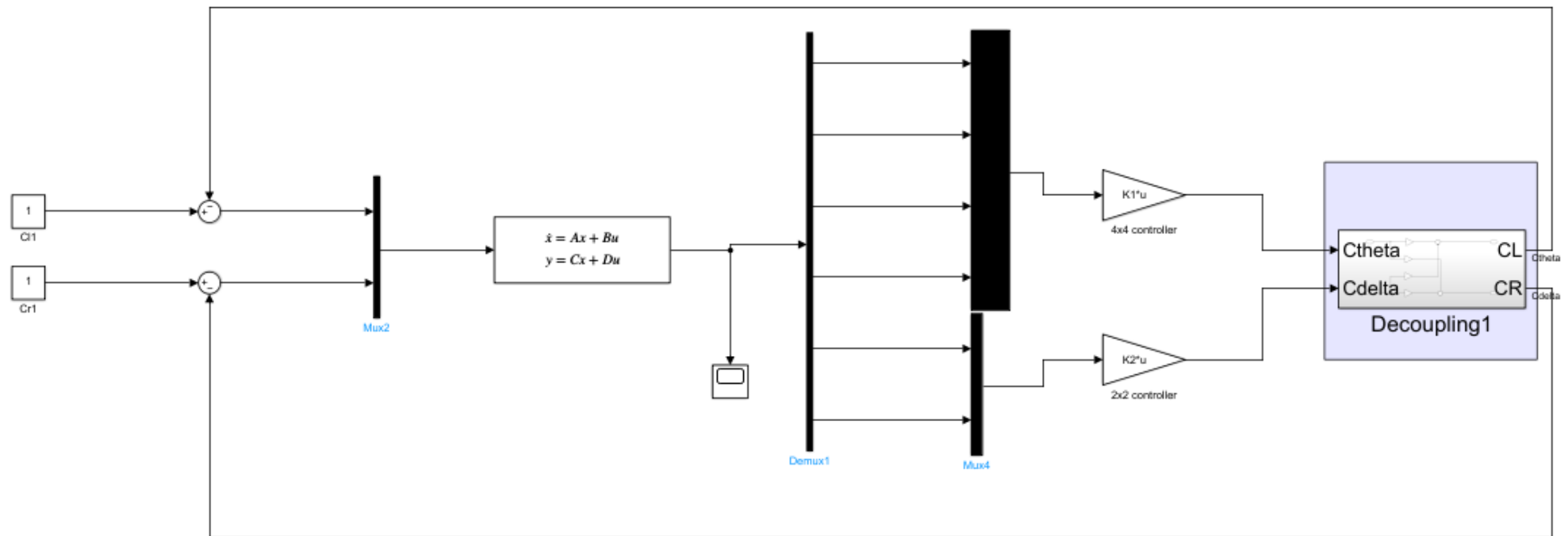


Fig.4 Decoupling Control System for TWSBR

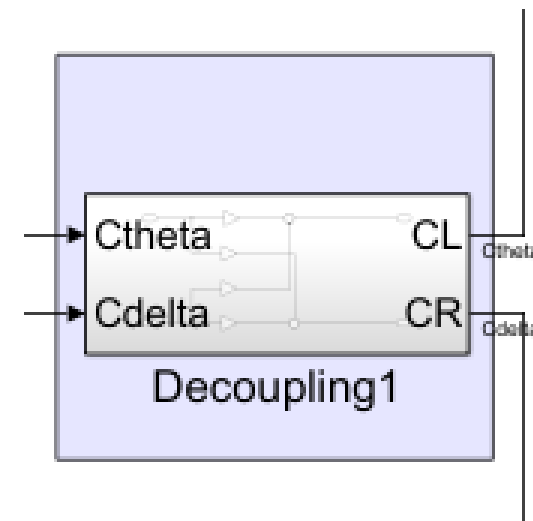
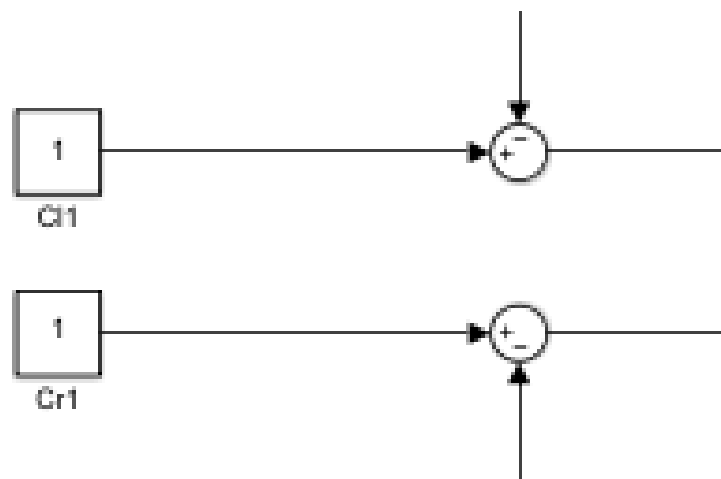
# پیاده سازی مدار کنترلی

- در سیمولینک برای شبیه سازی مراحل طی شده، بلوک دیاگرام زیر را پیاده سازی کرده ام:



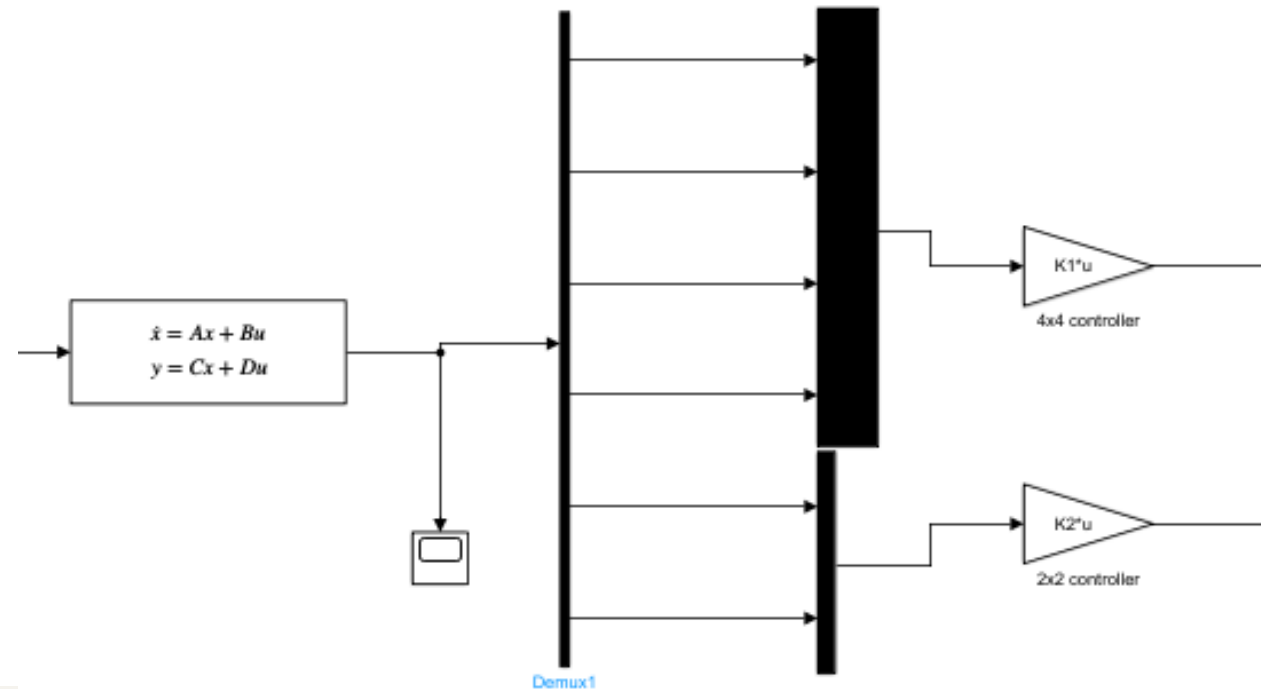
# کنترل با دی‌کوپلینگ سیگنال مرجع

- همانطور که مشاهده میکنید، دی‌کوپلینگ به این صورت است که  $C_{\theta}$  و  $C_{\delta}$  به عنوان خروجی های کنترلر به  $CL$  و  $CR$  تبدیل شده و در نهایت به عنوان فیدبک با ست‌پوینت مرجع مقایسه می‌شوند.



# تبدیل سیستم اصلی به ۲ زیر سیستم اصلی

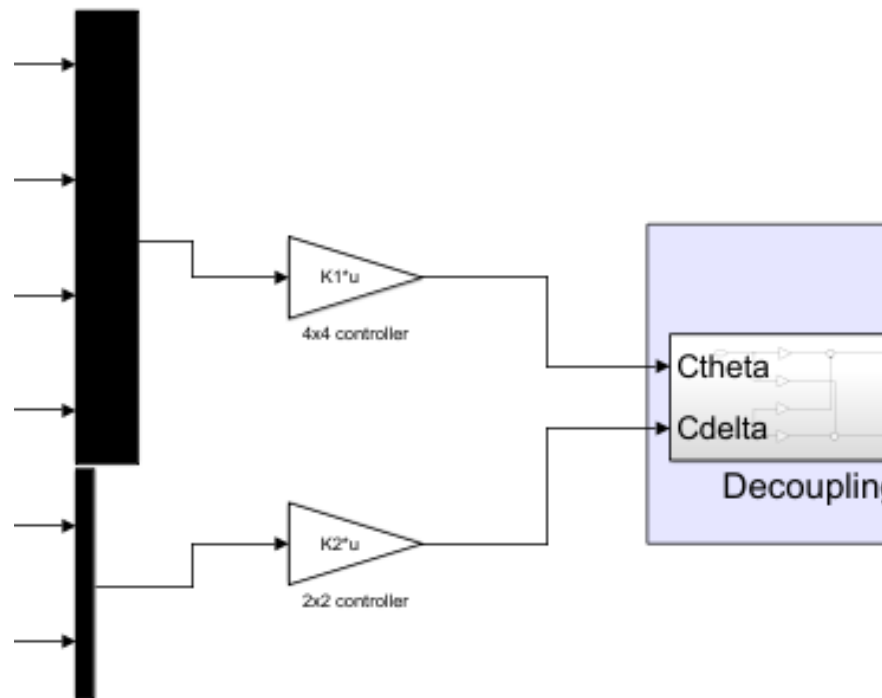
- در مقاله به علت اثرگذاری CL و CR روی یکدیگر از Cdelta و Ctheta استفاده شده که هرکدام مربوط به یک سیستم می‌باشند. بنابراین ۶ متغیر حالت اولیه به دو گروه ۴ متغیره و ۲ متغیره تقسیم شده است:



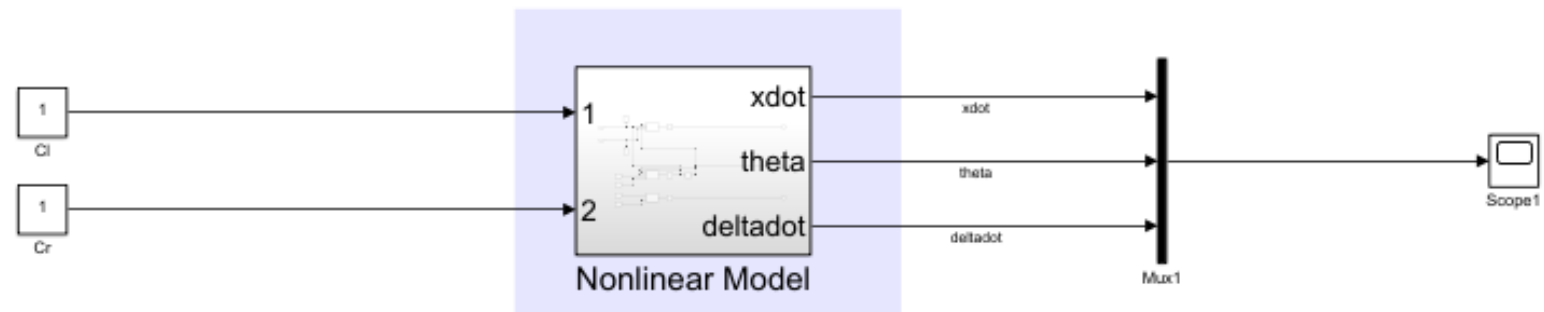


# کنترل

- در بخش کنترل نیز از متد LQR بهره گرفته شده که برای آن مقادیر  $Q$  و  $R$  در مقاله بطور خاص مشخص نشده است. برای تعیین آن باید سیگنالینگ ورودی و محدودیت‌های اورشوت در نظر گرفته شوند.



# شبیه‌سازی مدل غیرخطی

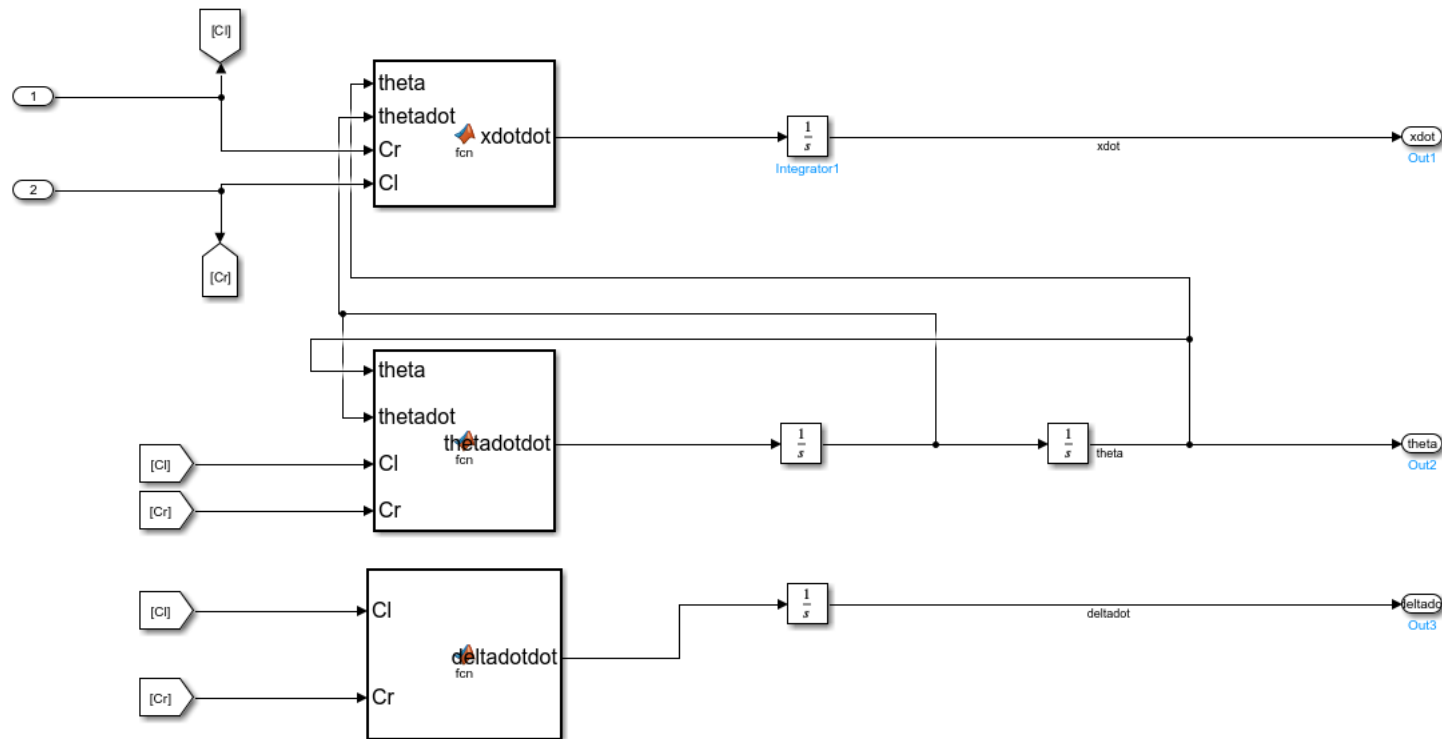


# مدل ریاضی

♦ ۳ معادله اصلی داریم که باید پیاده‌سازی کنیم:

$$\begin{aligned}\ddot{X}_{RM} &= \frac{1}{J_P \left( 2 \left( J_W / R^2 + M_W \right) + M_P \right) + 2 \left( J_W / R^2 + M_W \right) M_P L^2 + M_P^2 L^2 \sin^2 \theta_P} \\ &\quad \cdot \left[ \left( J_P + M_P L^2 \right) M_P L \sin \theta_P \dot{\theta}_P^2 - M_P^2 L^2 g \cos \theta_P \sin \theta_P \right. \\ &\quad \left. + \left( \frac{1}{R} \left( J_P + M_P L^2 \right) + M_P L \cos \theta_P \right) (C_R + C_L) \right. \\ &\quad \left. + \left( J_P + M_P L^2 (f_{RR} + f_{RL}) \right) \right] \\ \ddot{\theta}_P &= - \frac{1}{J_P \left( 2 \left( J_W / R^2 + M_W \right) + M_P \right) + 2 \left( J_W / R^2 + M_W \right) M_P L^2 + M_P^2 L^2 \sin^2 \theta_P} \\ &\quad \cdot \left[ M_P^2 L^2 \sin \theta_P \cos \theta_P \dot{\theta}_P^2 + \left( 2 \left( J_W / R^2 + M_W \right) + M_P \right) M_P g L \sin \theta_P \right. \\ &\quad \left. - \left( M_P L \cos \theta_P / R + 2 \left( J_W / R^2 + M_W \right) + M_P \right) (C_R + C_L) \right. \\ &\quad \left. - M_P L \cos \theta_P (f_{RR} + f_{RL}) + 2 \left( J_W / R^2 + M_W \right) L \cos \theta_P f_P \right] \\ \ddot{\delta} &= [(C_L - C_R) / R + (f_L - f_R)] / (2J_P M_W + J_W D / R^2)\end{aligned}$$

# پیاده سازی در سیمولینک



# توابع سیمولینک در .mfile

• Xdotdot:

```
function xdotdot = fcn(theta, thetadot, Cr, Cl)
```

```
mp = 20;  
mw = 6;  
l = 0.2;  
jpy = 1.33;  
r = 0.2;  
d = 0.5;  
jp = 0.27;  
jw = 0.12;  
g = 9.82;  
frr = 0;  
frl = 0;
```

```
xdotdot = 1/(jp*(2*(jw/r^2+mw)+mp)+2*(jw/r^2+mw)*mp*l^2+mp^2*l^2*sin(theta)^2)*((jp+mp*l^2)*mp*l*sin(th
```

# توابع سیمولینک در mfile.

Thetadotdot :

```
function thetadotdot = fcn(theta, thetadot, Cl, Cr)
mp = 20;
mw = 6;
l = 0.2;
jpy = 1.33;
r = 0.2;
d = 0.5;
jp = 0.27;
jw = 0.12;
g = 9.82;
frr = 0;
frl = 0;
fp = 0;
thetadotdot = -1/(jp*(2*(jw/r^2+mw)+mp)+2*(jw/r^2+mw)*mp*l^2+mp^2*l^2*sin(theta)^2)*(mp^2*l^2*sin(theta)
```

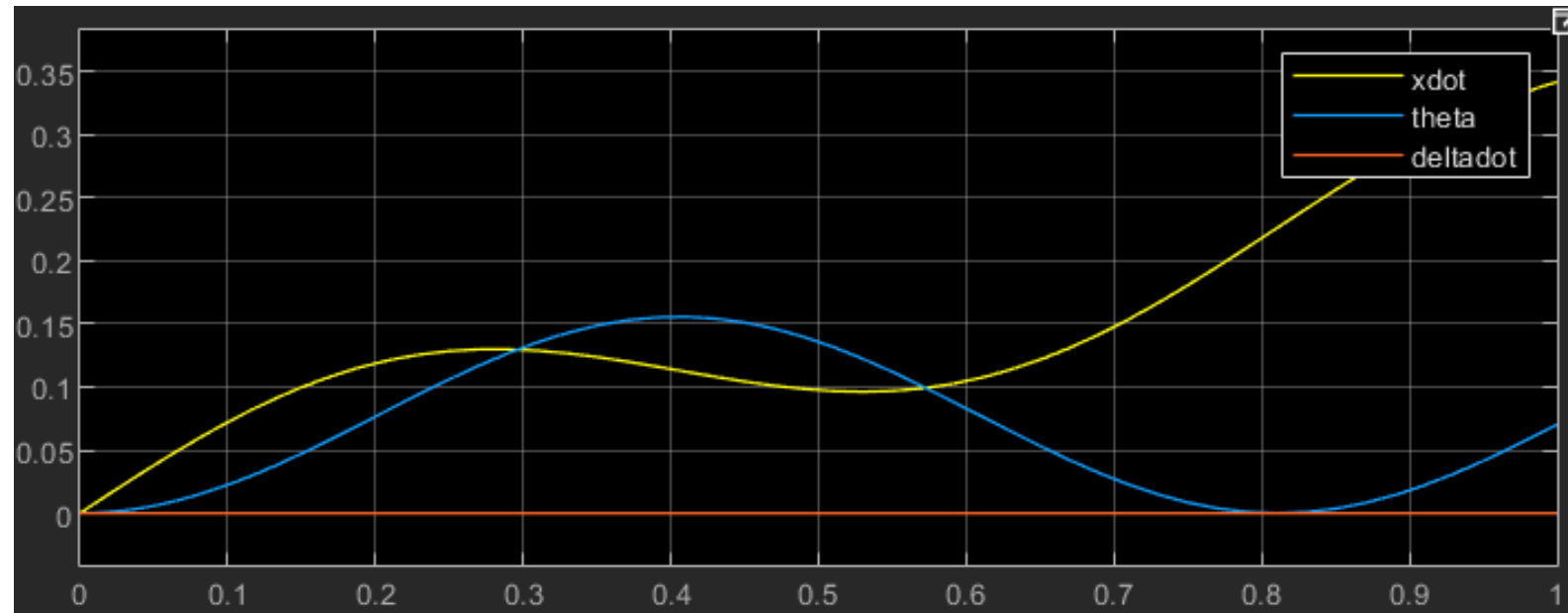
# توابع سیمولینک در .mfile

• Deltadotdot:

```
function deltadotdot = fcn(Cl, Cr)
    mp = 20;
    mw = 6;
    l = 0.2;
    jpy = 1.33;
    r = 0.2;
    d = 0.5;
    jp = 0.27;
    jw = 0.12;
    g = 9.82;
    fr = 0;
    fl = 0;

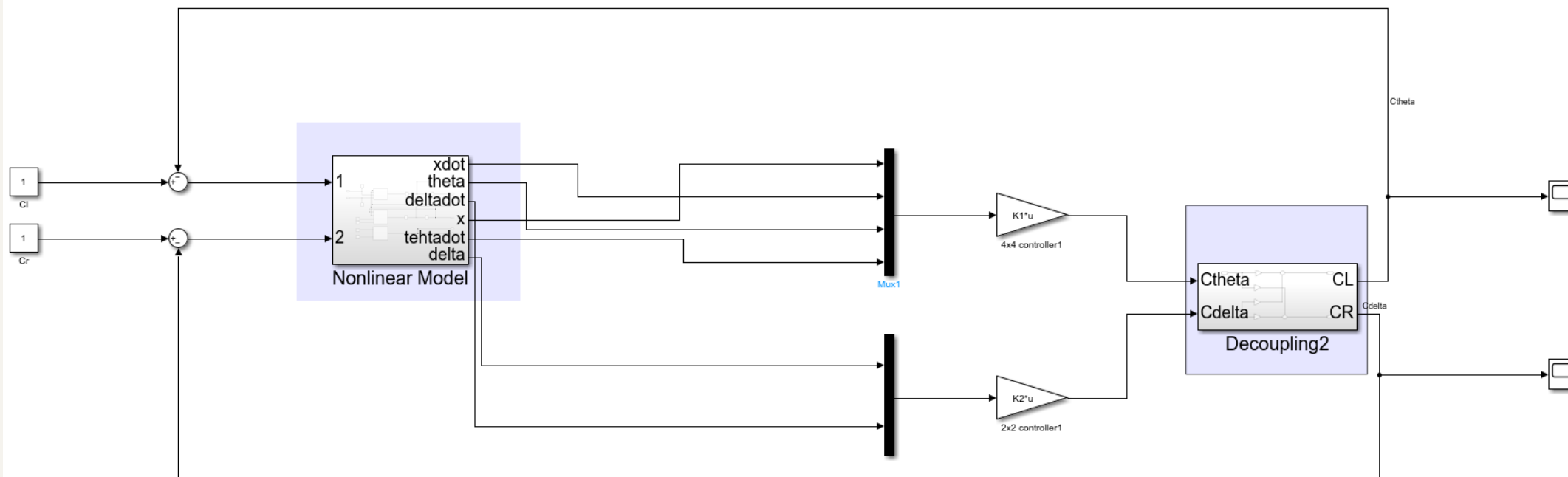
    deltadotdot = ((Cl-Cr)/r+(fl-fr))/(2*jp*mw+jw*d/r^2);
```

# پاسخ طبیعی سیستم غیر خطی





# کنترلر بهینه روی مدل غیرخطی



# توجه

- ♦ البته باید توجه شود که نمیتوان همان کنترلر بهینه را روی مدل غیرخطی پیاده‌سازی کرد، زیرا برای ورودی  $C\delta$  و  $C\theta$  طراحی شده بود. ولی سیستم ما مدل غیرخطی برای ورودی‌های  $CL$  و  $CR$  مدل‌سازی شده است.
- ♦ برای پیاده‌سازی کنترلر روی مدل غیرخطی بکمک مدل خطی باید مدل خطی کنترل‌پذیر حالت باشد. که در اینجا اینطور نیست.
- ♦ البته یک روش نیز بدست آوردن رابطه  $\ddot{x}$  و  $\ddot{\theta}$  و  $\ddot{\delta}$  بر حسب  $C\delta$  و  $C\theta$  میباشد. که در اینطور همان کنترل‌کننده قبلی پاسخ مساله را میدهد.