

دانشگاه خواجه نصیرالدین طوسی
دانشکده برق - گروه مکترونیک

تمرین درس کنترل مبتنی بر پیش بینی مدل دوره کارشناسی ارشد

رشته مهندسی مکترونیک

عنوان

تمرین درس کنترل مبتنی بر پیش بینی مدل

نگارش

علیرضا امیری

آذر ۱۴۰۳

پاسخ سوالات سری چهارم

پاسخ سوال ۱

بخش یکم

در سوال اول با توجه به معادلات حالت سیستم که به صورت زیر داده شده است، می توانیم درکی از سیستم به دست

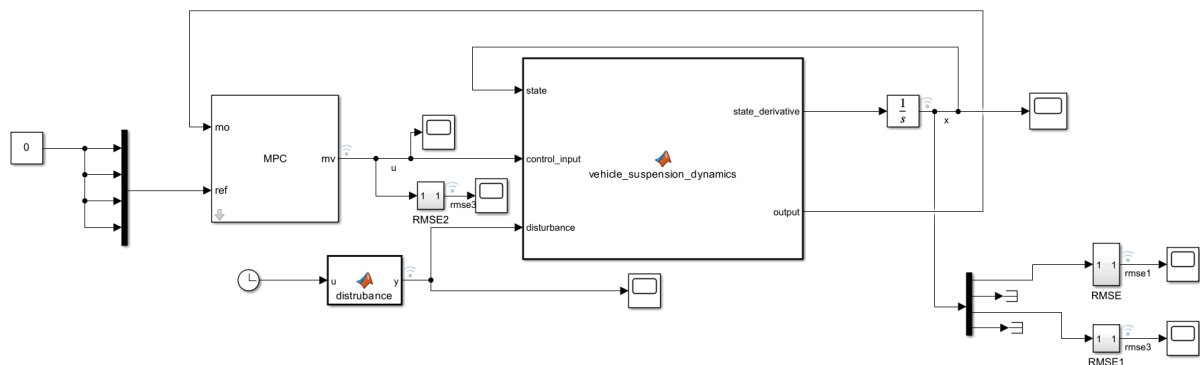
بیاوریم:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{c_S}{M_S} & \frac{k_S}{M_S} & \frac{c_S}{M_S} \\ 0 & 0 & 1 & 0 \\ -\frac{k_U}{M_U} & \frac{c_S}{M_U} & \frac{k_S+k_U}{M_U} & \frac{-c_S+c_U}{M_U} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{c_S c_U}{M_S M_U} \\ -\frac{c_U}{M_U} \\ \frac{c_U}{M_U} \left(\frac{k_U}{c_U} - \frac{c_S}{M_U} - \frac{c_U}{M_U} \right) \end{bmatrix} d + \begin{bmatrix} 0 \\ \frac{1}{M_S} \\ 0 \\ -\frac{1}{M_U} \end{bmatrix} u$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} d + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} u$$

مشاهده می شود که سیستم دارای چهار حالت و سه خروجی مشاهده پذیر است. ساختار کنترلی پیش بین خطی

برای این سوال به صورت زیر طراحی می شود.



شکل ۱: بلوک دیاگرام LMPC

لازم به ذکر است که با توجه به محدودیت های متلب برای ابعاد ورودی ها، خروجی ها و حالت های سیستم، چهار خروجی برای سیستم تعریف شده است، اما خروجی چهارم که نیازی نیست در نظر گرفته شود با مقدار ۰ در نظر گرفته می شود. معادلات سیستم را به صورت زیر در یک سیستم تعریف می کنیم.

```
function [state_derivative, output] = vehicle_suspension_dynamics(state, control_input, disturbance)
% Parameters of the suspension system
sprung_mass = 2500; % Sprung mass (kg)
unsprung_mass = 320; % Unsprung mass (kg)
spring_constant_s = 80000; % Spring constant of the suspension (N/m)
spring_constant_u = 500000; % Spring constant of the tire (N/m)
damping_coefficient_s = 350; % Damping coefficient of the suspension (Ns/m)
damping_coefficient_u = 15020; % Damping coefficient of the tire (Ns/m)

% State-space matrices
% State vector: [x_s; v_s; x_u; v_u]
% x_s: Displacement of the sprung mass
% v_s: Velocity of the sprung mass
% x_u: Displacement of the unsprung mass
% v_u: Velocity of the unsprung mass

A_matrix = [
    0, 1, 0, 0;
    -damping_coefficient_s/sprung_mass, -spring_constant_s/sprung_mass, damping_coefficient_s/sprung_mass, 0;
    0, 0, 0, -1;
    -spring_constant_u/unsprung_mass, damping_coefficient_s/unsprung_mass, (spring_constant_s + spring_constant_u)/unsprung_mass, -(damping_coefficient_s + damping_coefficient_u)/unsprung_mass
];

% Input matrices
B_disturbance = [
    0;
    damping_coefficient_s * damping_coefficient_u / (sprung_mass * unsprung_mass);
    -damping_coefficient_u / unsprung_mass;
    damping_coefficient_u / unsprung_mass * (spring_constant_u / damping_coefficient_u - damping_coefficient_s / unsprung_mass - damping_coefficient_u / unsprung_mass)
];

B_control = [
    0;
    1 / sprung_mass;
    0;
    -1 / unsprung_mass
];

% Compute state derivative
state_derivative = A_matrix * state + B_disturbance * disturbance + B_control * control_input;

% Compute output vector
% Output vector is [x_s; v_s; x_u; placeholder], where placeholder is fixed at 0
output = zeros(4, 1);
output(1) = state(1); % Sprung mass displacement
output(2) = state(2); % Sprung mass velocity
output(3) = state(3); % Unsprung mass displacement
output(4) = 0; % Placeholder
end
```

شکل ۲: کد پلنت سیستم

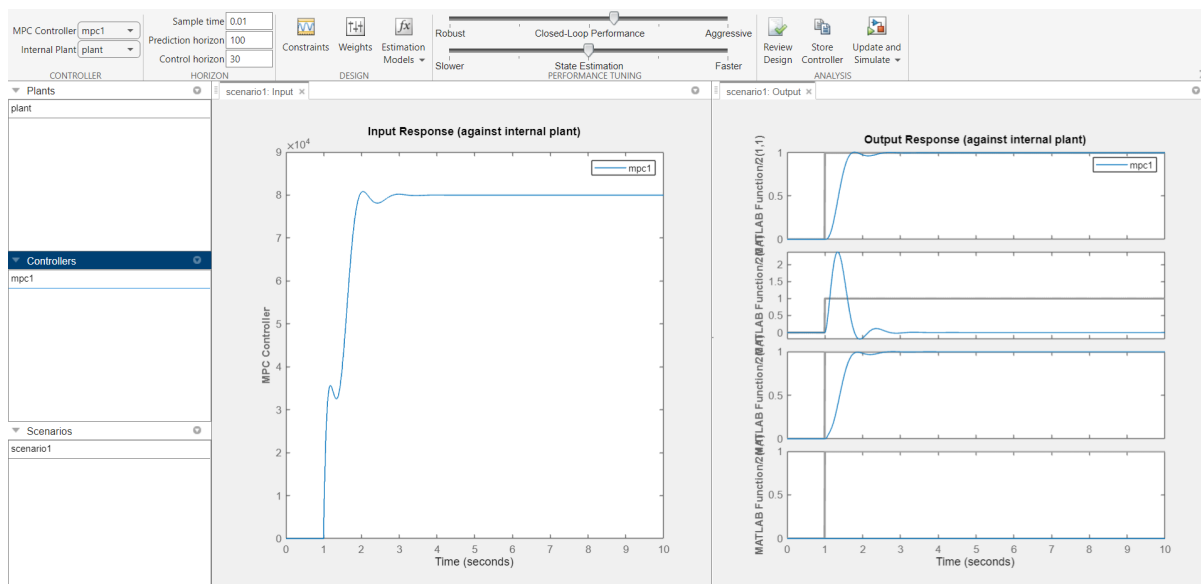
همچنین، اغتشاش وارد شده به سیستم توسط کد زیر ایجاد می شود.

```
function y = distrubance(u)

if u>=4 & u<5
    y = (u-4);
elseif u>=5 & u<= 7
    y = 1;
elseif u>7 & u<8
    y = -u+8;
else
    y = 0;
end
```

شکل ۳: اغتشاش

در ادامه، با طراحی کنترلر MPC برای این سیستم با تنظیمات زیر، می توانیم نتایج به دست آمده از سیستم را مشاهده کنیم:



شکل ۴: design LMPC

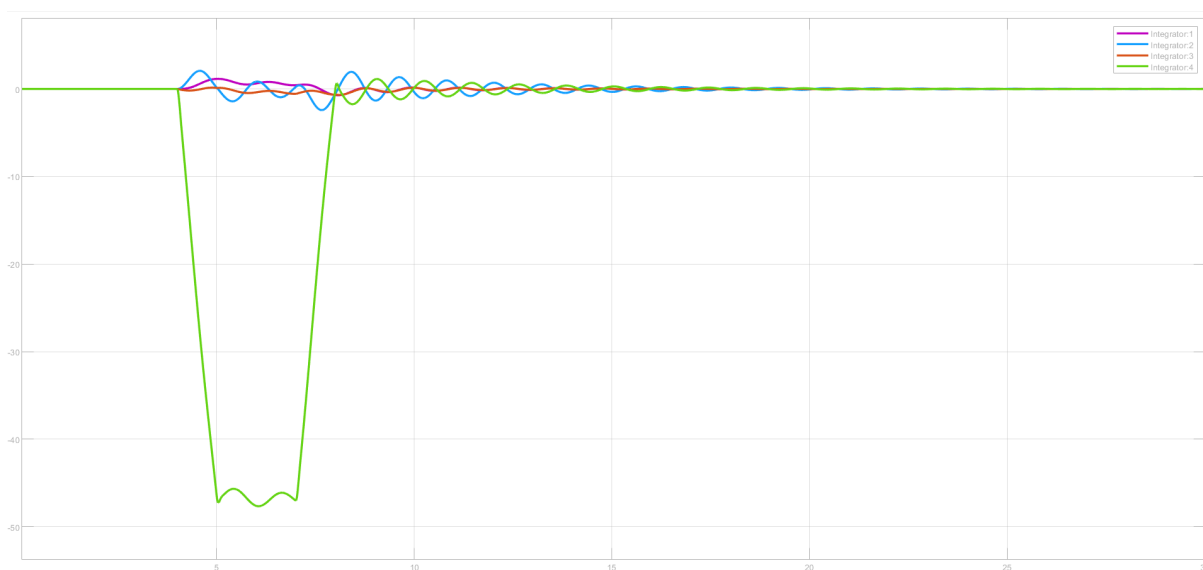
همچنین باید توجه شود که قید هایی برای حالت اول سیستم در نظر گرفته شده است تا مقدار آن از ۱ بیشتر نشود.

Input and Output Constraints					
Channel	Type	Min	Max	RateMin	RateMax
▼ Inputs					
$u(1)$	MV	-Inf	Inf	-Inf	Inf
▼ Outputs					
$y(1)$	MO	-1	1		
$y(2)$	MO	-Inf	Inf		
$y(3)$	MO	-Inf	Inf		
$y(4)$	MO	-Inf	Inf		

Equal Constraint Relaxation (ECR)					
Channel	Type	MinECR	MaxECR	RateMinECR	RateMaxECR
▼ Inputs					
$u(1)$	MV	0	0	0	0
▼ Outputs					
$y(1)$	MO	1	1		
$y(2)$	MO	1	1		
$y(3)$	MO	1	1		
$y(4)$	MO	1	1		

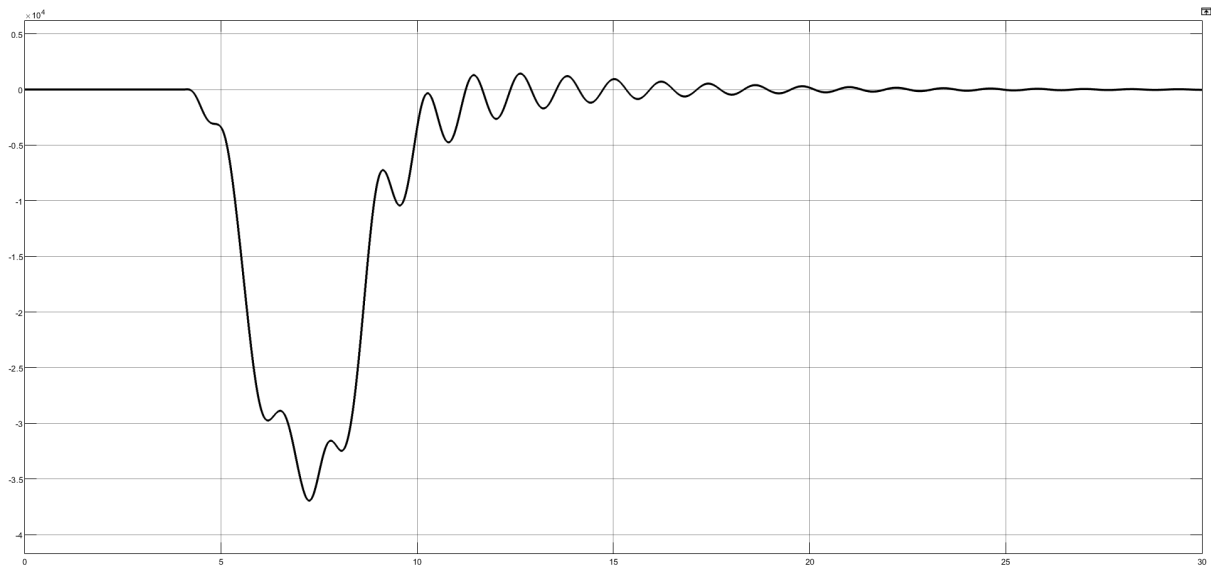
شکل ۵: configuration constraints LMPC

با استفاده از سیستم تعریف شده، پاسخ های سیستم به صورت زیر به دست می آید.



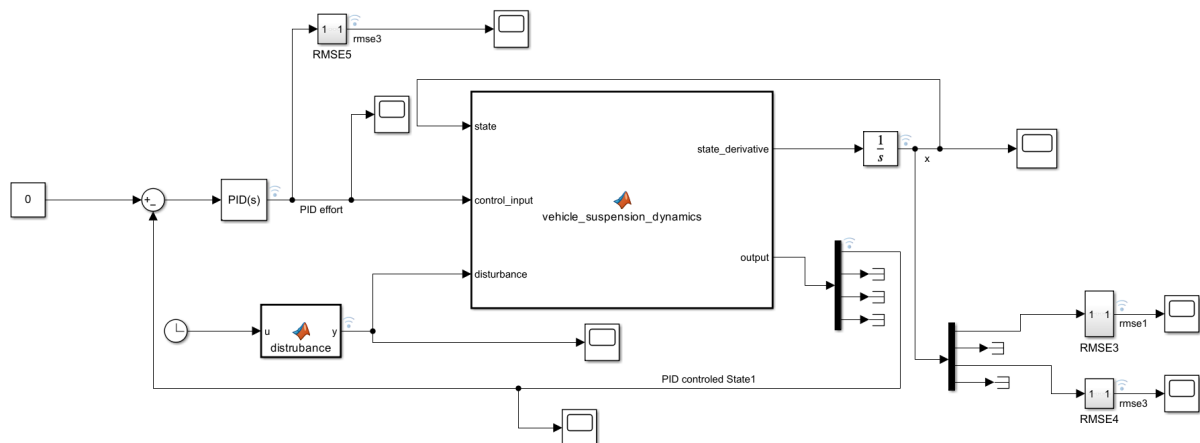
شکل ۶: output states LMPC

همچنین نمودار تلاش کنترلی سیستم به صورت زیر به دست می آید.



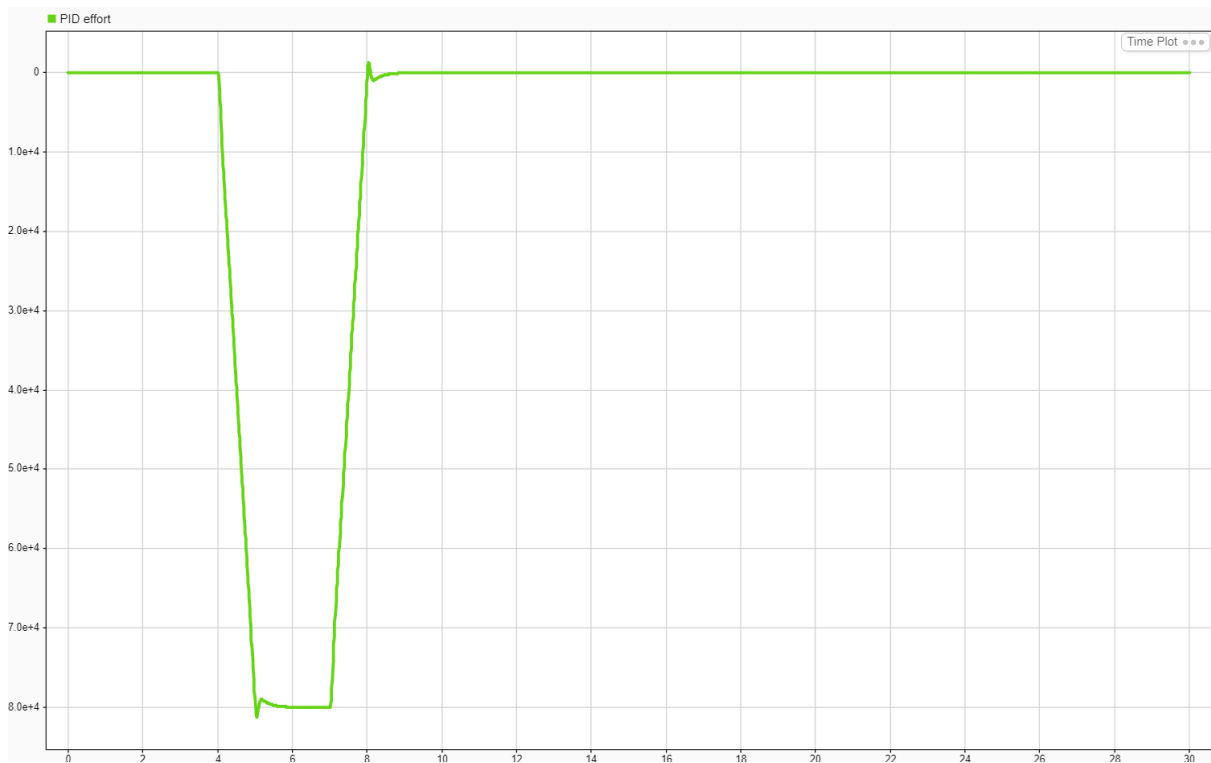
شکل ۷: effort control LMPC

با در اختیار داشتن نتایج به دست آمده و به منظور تحلیل این نتایج، کنترلر PID نیز برای سیستم طراحی و بررسی می شود. دیاگرام این سیستم به شرح زیر خواهد بود:



شکل ۸: diagram PID

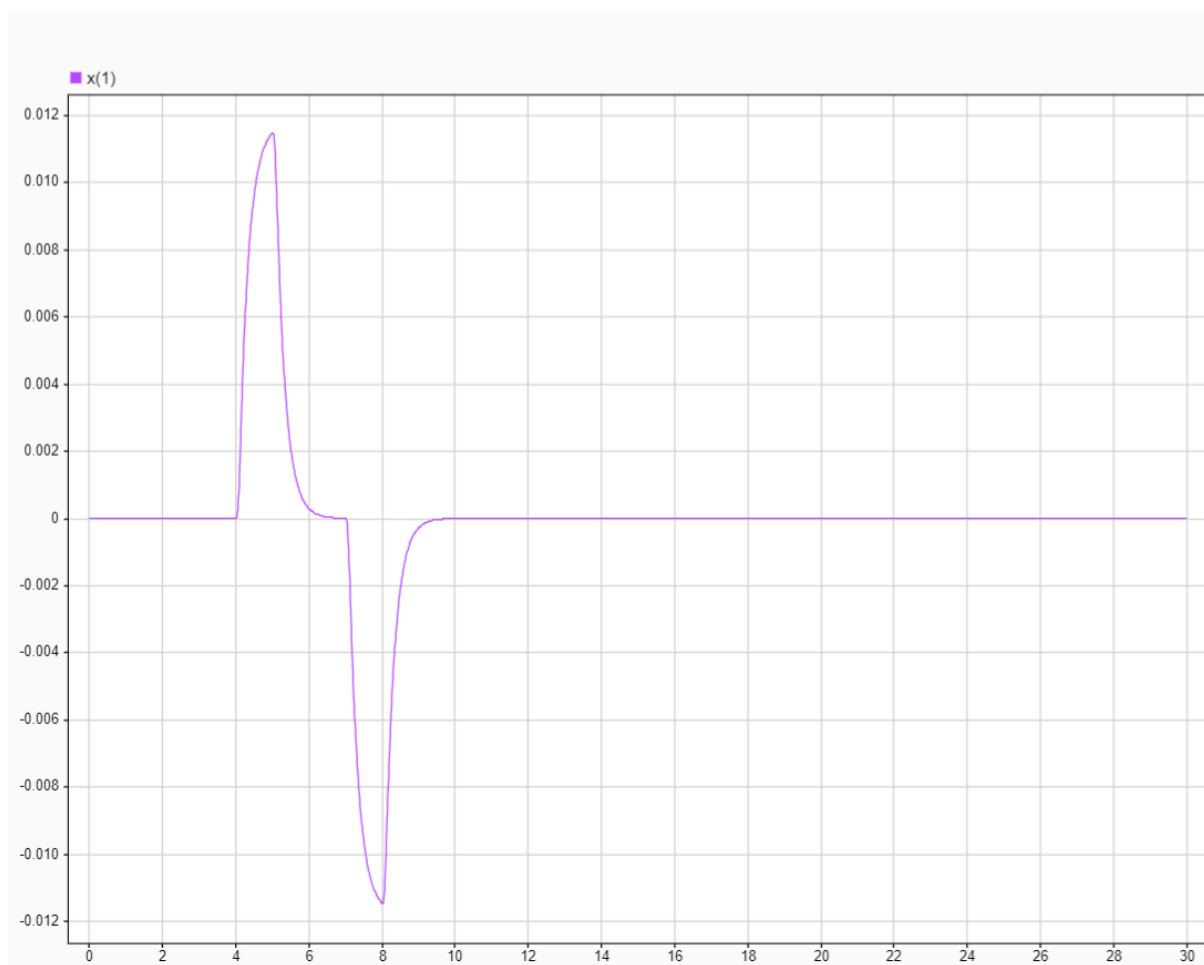
لازم به ذکر است که برای این سیستم، تنها حالت اول به عنوان خروجی سیستم در نظر گرفته شده است. پاسخ سیستم به صورت زیر به دست می آید و مشاهده می شود که می تواند به خوبی اغتشاش وارد شده را رفع کند. اما با در نظر گرفتن ضرایب PID و همچنین نمودار تلاش کنترلی مشاهده می شود که در نهایت این کنترلر در سیستم های واقعی قابل پیاده سازی نمی باشد.



شکل ٩: PID effort control

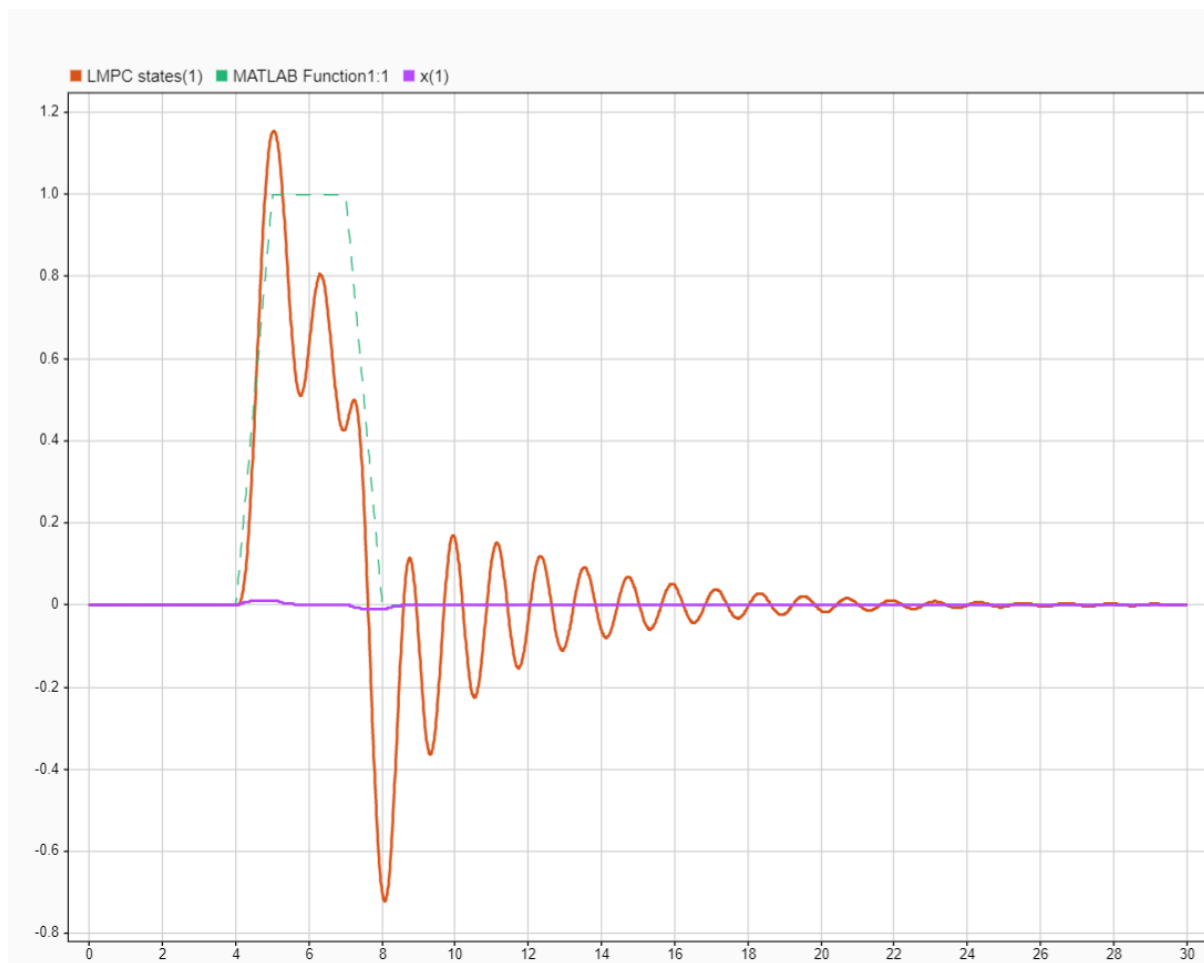
Proportional (P):	2115167.33224621	2.1152e+06	:
Integral (I):	6808255.34812067	6.8083e+06	:
	<input type="checkbox"/> Use I*Ts (optimal for codegen)		
Derivative (D):	134270.145058429	1.3427e+05	:
Filter coefficient (N):	315.344677005256	315.34	:
	<input checked="" type="checkbox"/> Use filtered derivative		

شکل ١٠

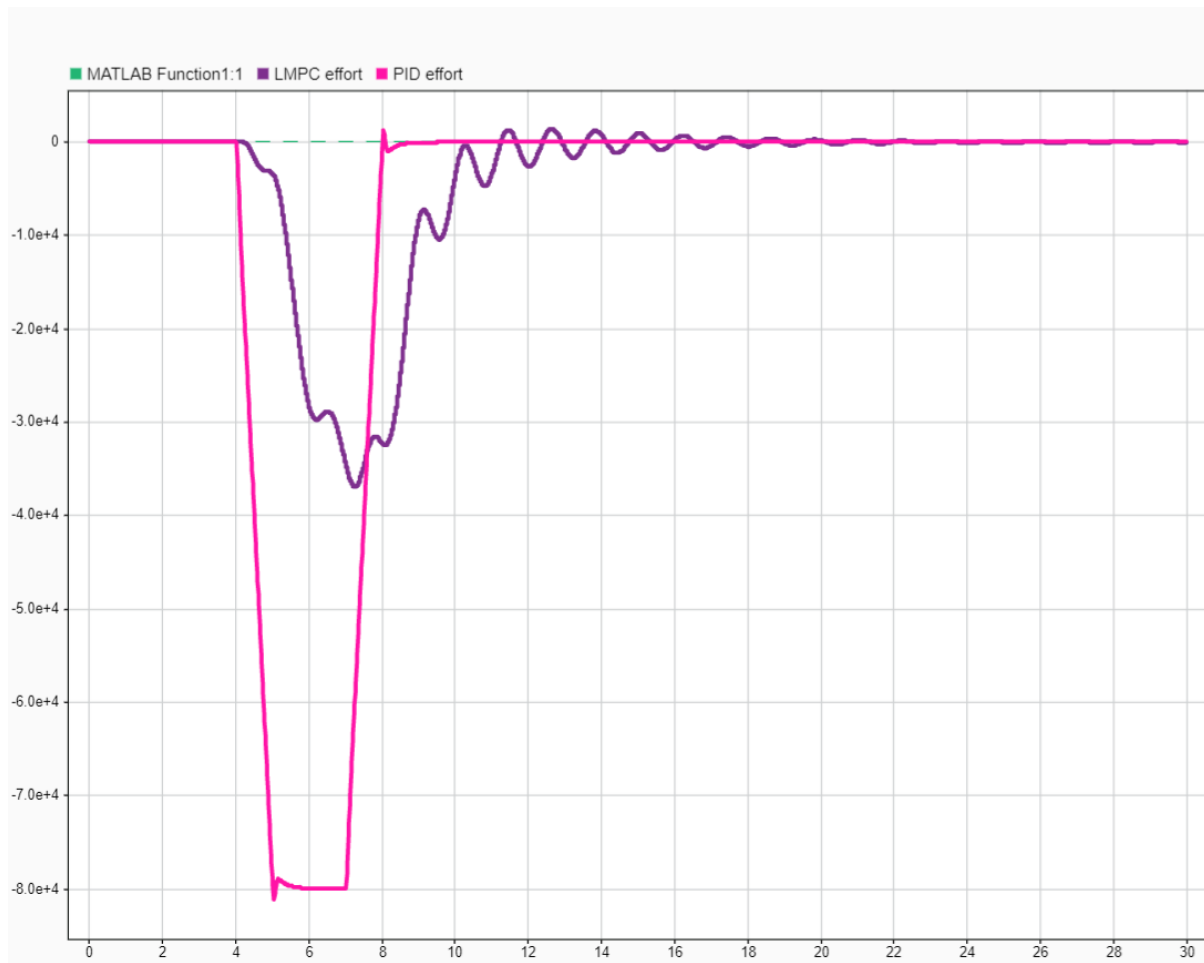


شکل ۱۱: output \state PID

در اینجا با در نظر گرفتن نتایج به دست آمده از MPC و PID می توانیم مقایسه ای در عملکرد آنها داشته باشیم. نتایج به دست آمده از خروجی کنترلی حالت اول با این دو کنترلر و تلاش کنترلی برای رفع آن به صورت زیر خواهد بود.

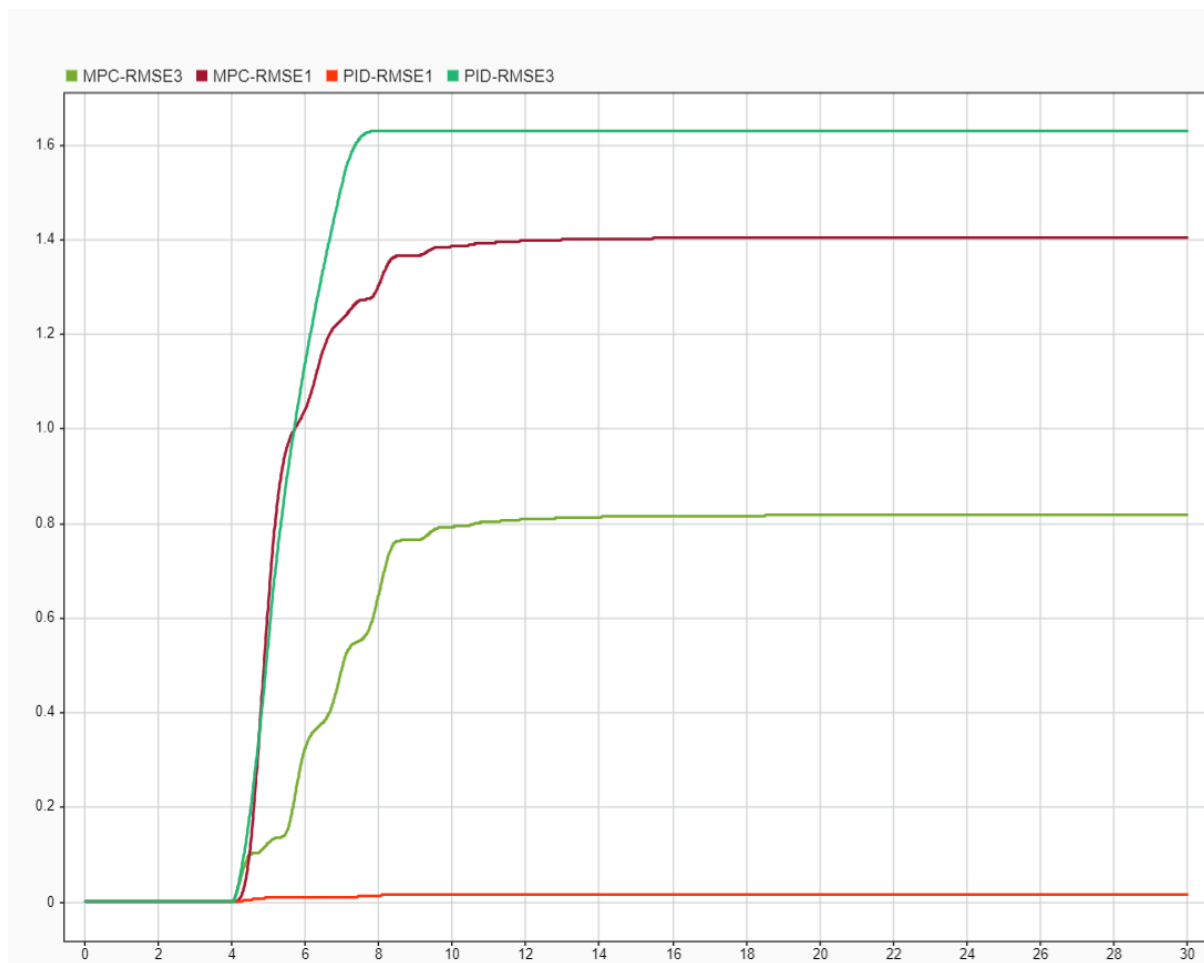


شکل ۱۲: output \state LMPC



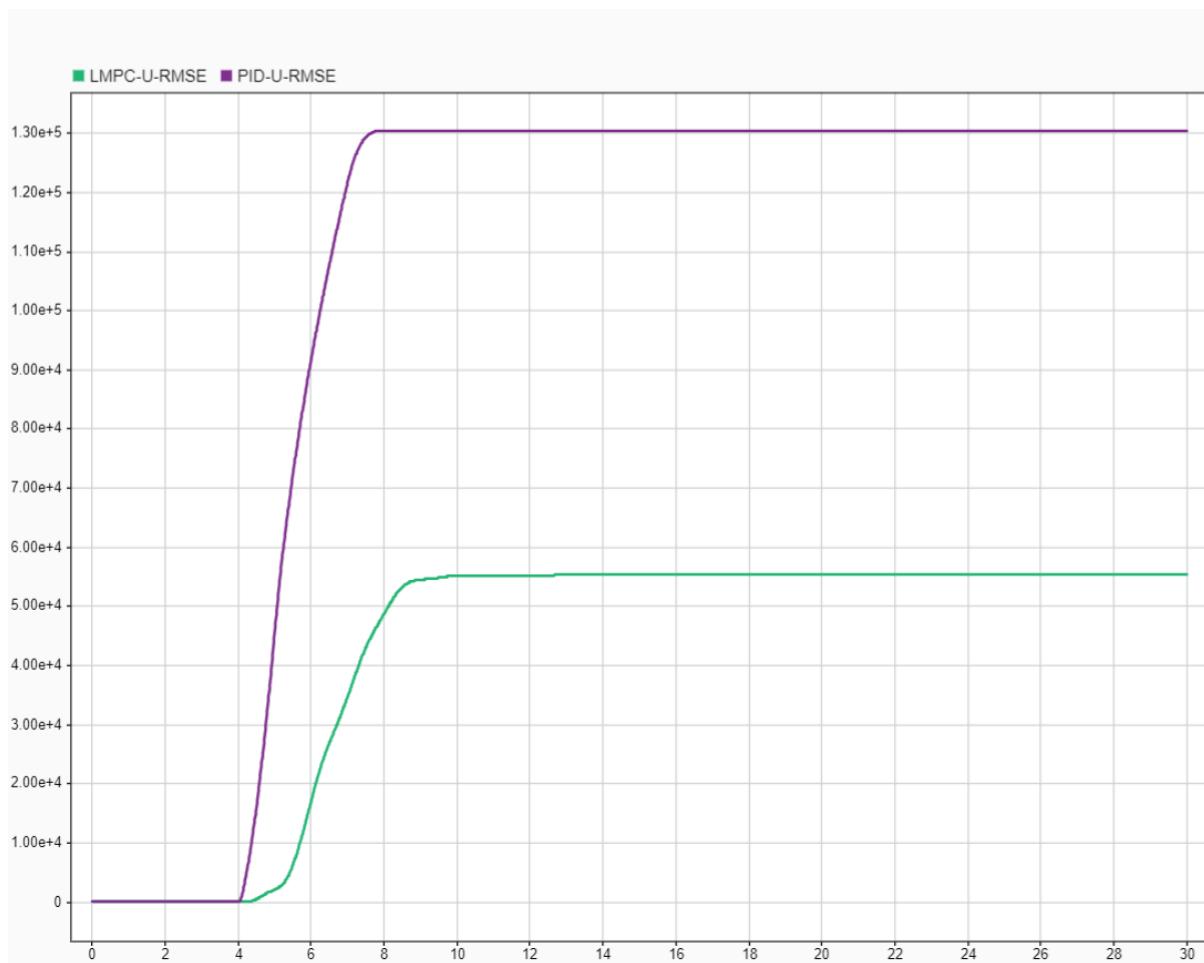
شکل ۱۳: effort control LMPC vs PID

همچنین مقادیر RMSE برای این دو کنترلر در دو حالت مورد نظر به صورت زیر به دست می آید.



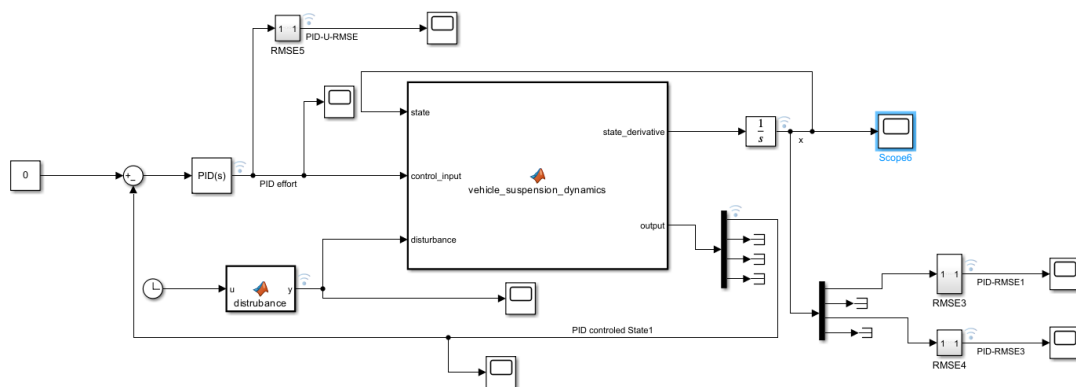
شکل ۱۴: ۳ and ۱ states of RMSE LMPC vs PID

نتایج به دست آمده نشان می دهد که مقادیر RMSE برای کنترلر MPC بیشتر از کنترلر PID است. همچنین، مقدار RMSE برای ورودی کنترلرها به صورت زیر خواهد بود:

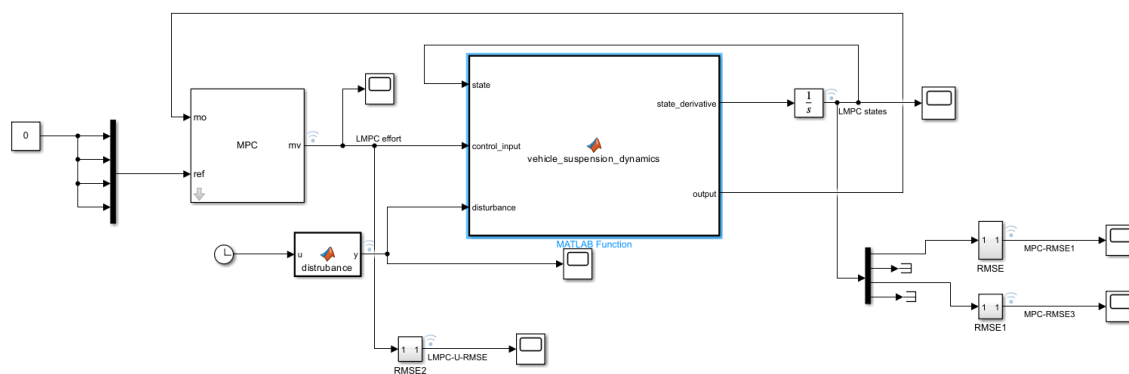


شکل ۱۵: RMSE effort control LMPC vs PID

در نتیجه دیگرام های سیستم ها با توجه به تغییرات اعمال شده به صورت زیر برای دو کنترلر خواهد بود.



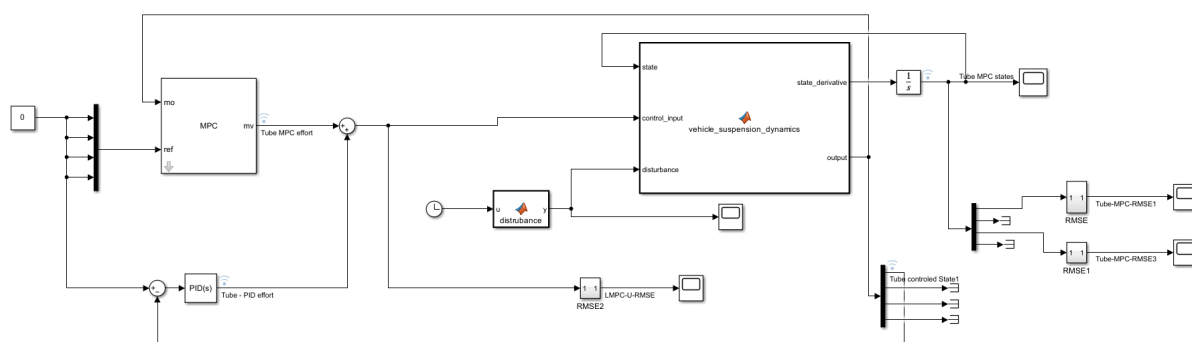
شکل ۱۶: کنترلر Tube



شکل ۱۷: کنترلر LMPC

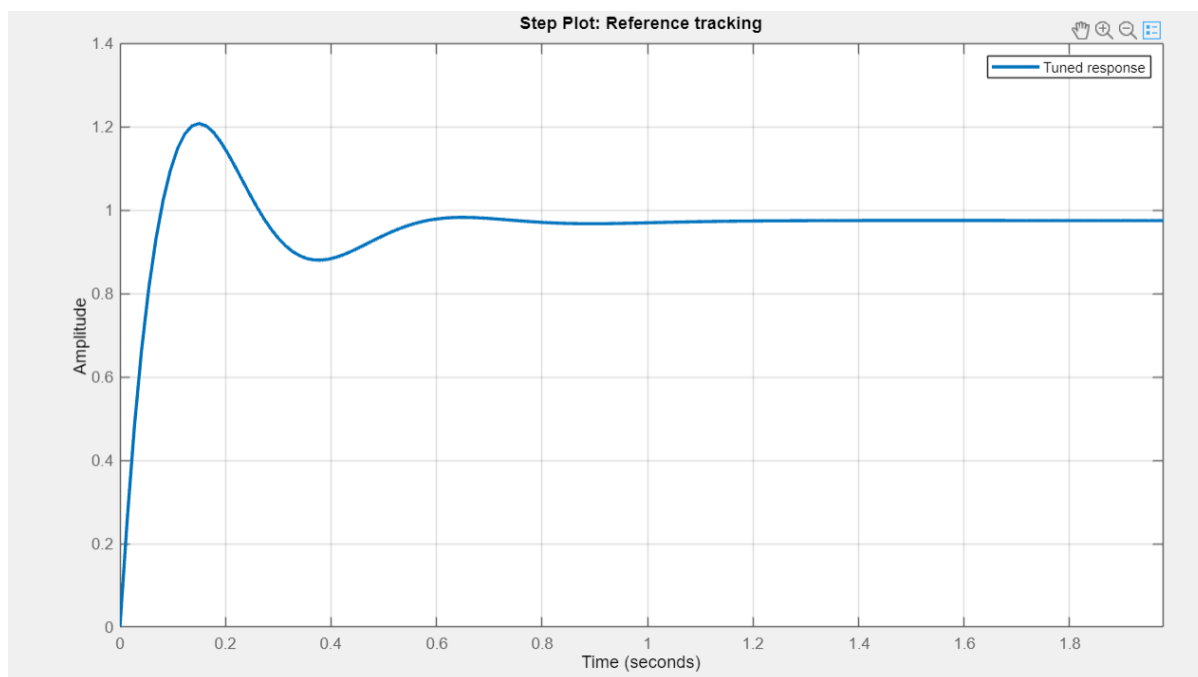
بخش دوم

در این بخش، با افزودن یک کنترلر PID به MPC و تشکیل یک کنترلر $TubeMPC$ ، مجدداً نتایج را بررسی می‌کنیم. بنابراین، ساختار سیستم طراحی شده به صورت زیر خواهد بود:



شکل ۱۸: کنترلر LMPC

در ادامه، با تنظیم مجدد کنترلر MPC و PID، کنترلر ها را مجدداً طراحی می‌کنیم. تنظیم های کنترلر PID به صورت زیر خواهد بود.



شکل ۱۹

Proportional (P): 673792.542179079 6.7379e+05

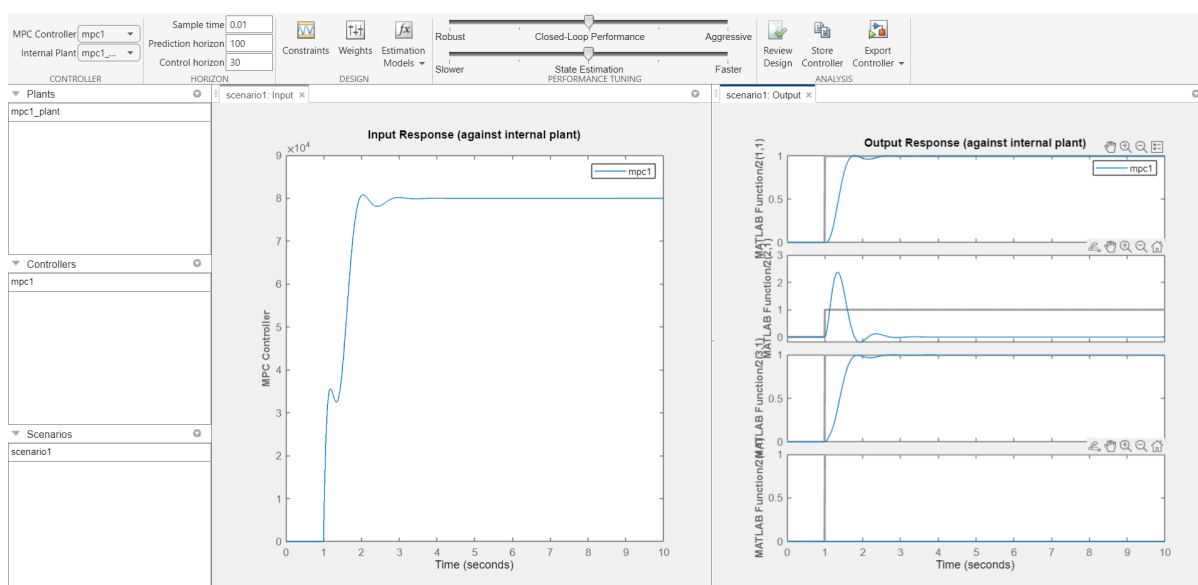
Integral (I): 2194376.82779204 2.1944e+06 ☐ Use I*Ts (optimal for codegen)

Derivative (D): 50799.688193398 50800

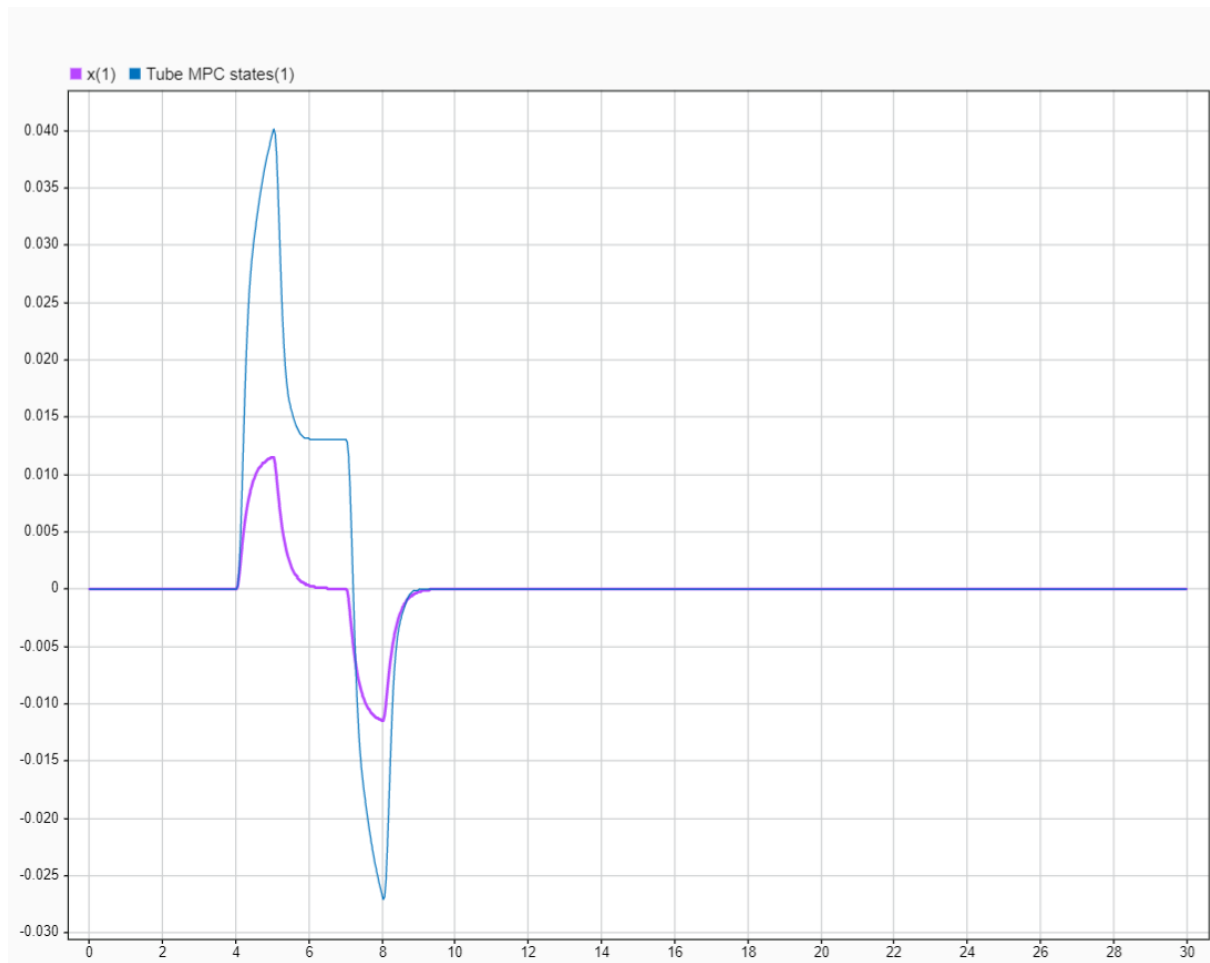
Filter coefficient (N): 2414.22496661449 2414.2 ☒ Use filtered derivative

شکل ۲۰

تنظیمات MPC به صورت زیر خواهد بود:

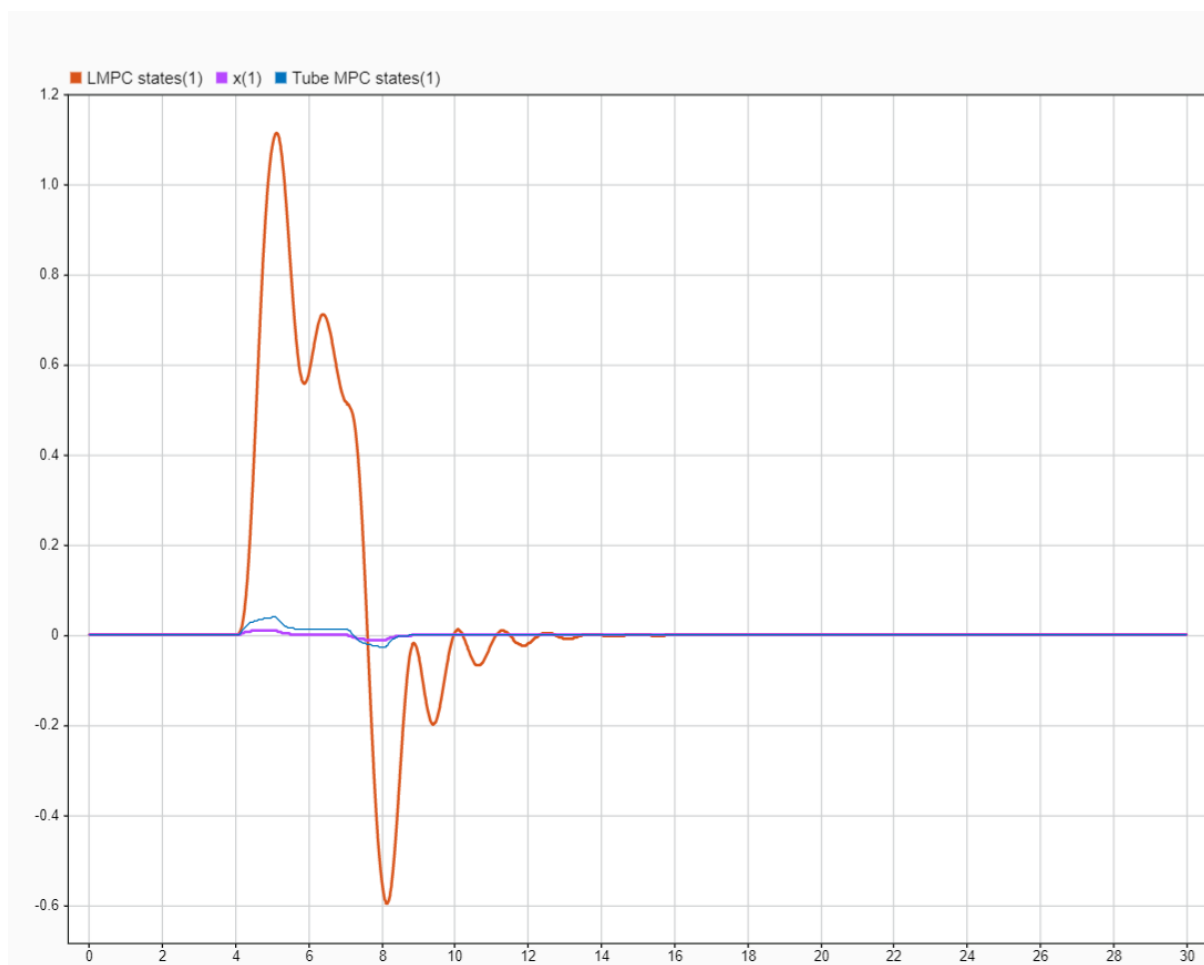


شکل ۲۱



شکل ۲۲





شکل ۲۴

بخش سوم

در قسمت بعد، سیستم را با پیاده سازی کنترلر *ExplicitMPC* به جای کنترلر LMPC کنترل می کنیم. برای این کار با استفاده از دستورات *generateExplicitRange*، *generateExplicitOptions* و *generateExplicitMPC* کنترلر MPC طراحی شده را با شرایط زیر به کنترلر EMPC تبدیل می کنیم.

```
% Create a copy of the original MPC for explicit version
empc_setup = mpc1;

% Significantly reduce horizons for explicit MPC
empc_setup.PredictionHorizon = 30; % Reduced from 100
empc_setup.ControlHorizon = 3;    % Reduced from 30

% Generate range for explicit MPC
range = generateExplicitRange(empc_setup);

% Set combined state and output constraints
range.State.Min = [-1; -1; -1; -1; -1; -1; -1; -1];
range.State.Max = [1; 1; 1; 1; 1; 1; 1; 1];

% Set reference constraints
range.Reference.Min = [-1; -1; -1; -1];
range.Reference.Max = [1; 1; 1; 1];

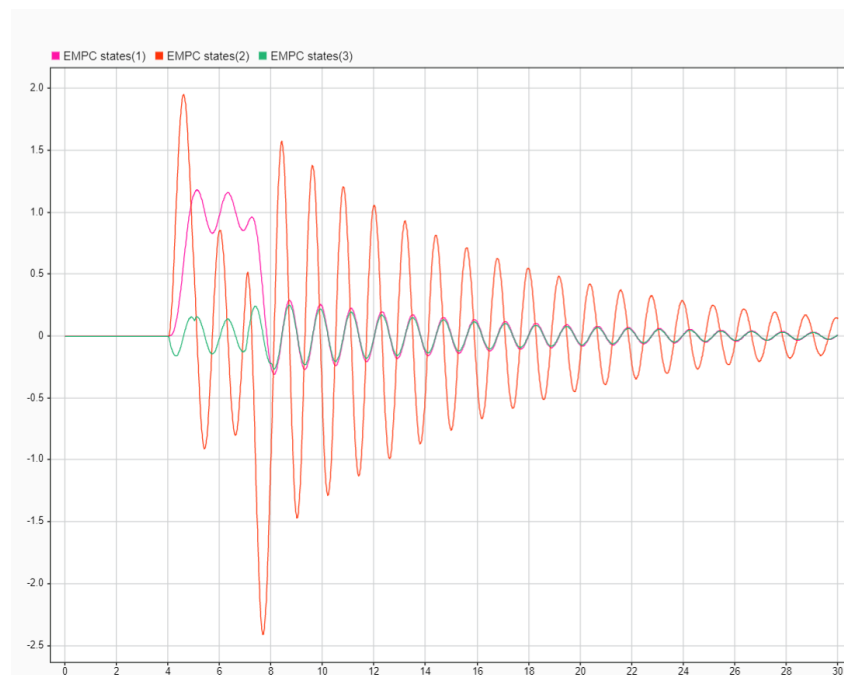
% Set MV constraints
range.ManipulatedVariable.Min = -1;
range.ManipulatedVariable.Max = 1;

% Configure explicit MPC generation options
opt = generateExplicitOptions(empc_setup);
opt.polyreduction = 1;
opt.dimensions = 'all';
opt.verbose = 2;

% Generate the explicit MPC controller
empc1 = generateExplicitMPC(empc_setup, range, opt);
```

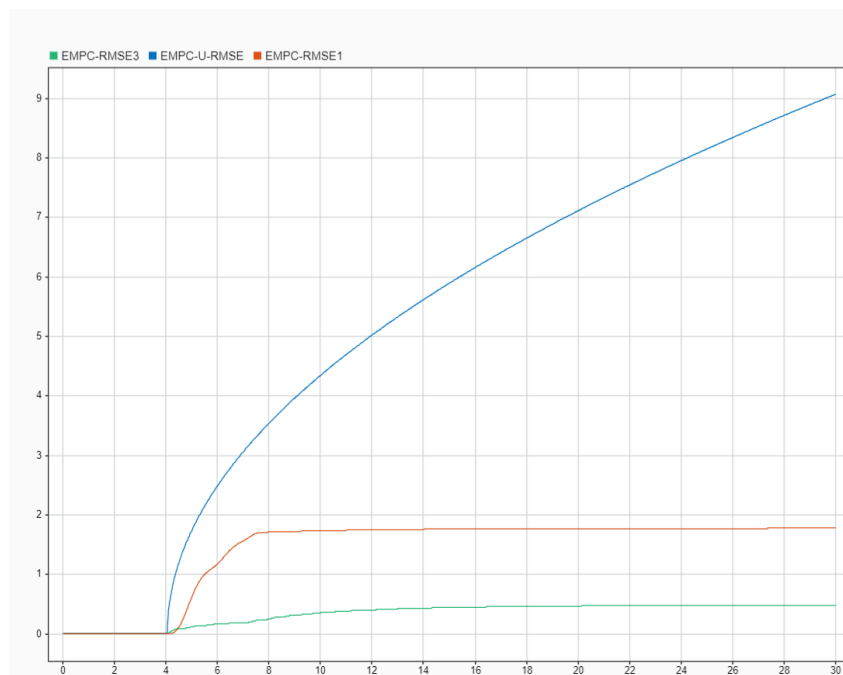
در اینجا لازم به ذکر است که استفاده از افق های پیش بینی و کنترل مورد استفاده در LMPC (افق پیش بین: ۱۰۰، افق کنترل ۳۰) برای کنترلر EMPC بسیار زیاد بوده و تعداد نواحی بسیار زیادی را ایجاد میکند که برای پیاده سازی، امکان پذیر نیست. بنابراین، پارامترهای این کنترلر به مقادیر زیر تغییر کرده اند. افق پیش بین: ۳۰ افق کنترل: ۳

با اجرای برنامه، خروجی کنترلر به صورت زیر به دست می آید و مشاهده می شود که عملکردی مشابه با کنترلر LMPC دارد.



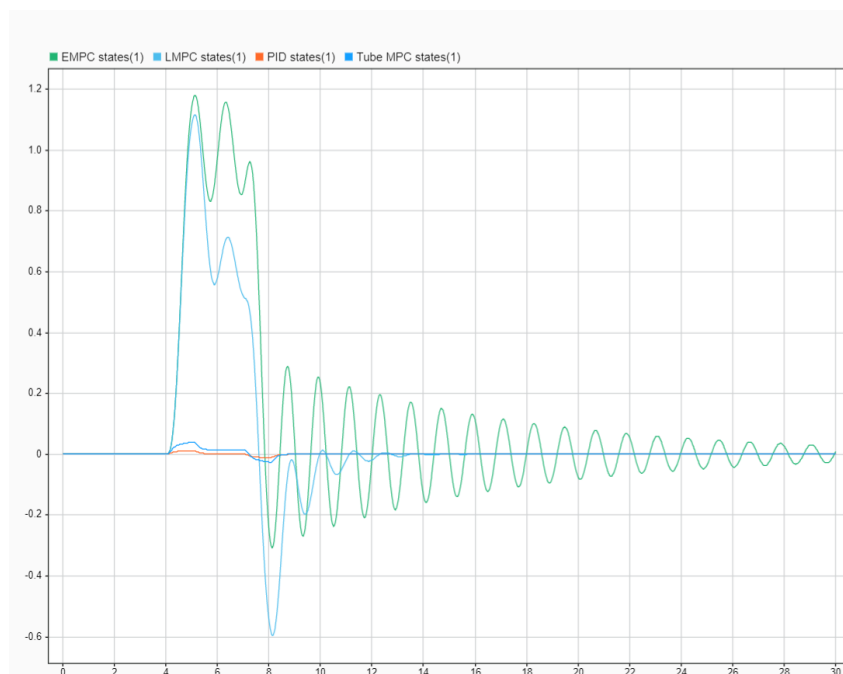
شکل ۲۶

مقادیر RMSE محاسبه شده برای این مدل به صورت زیر خواهد بود.



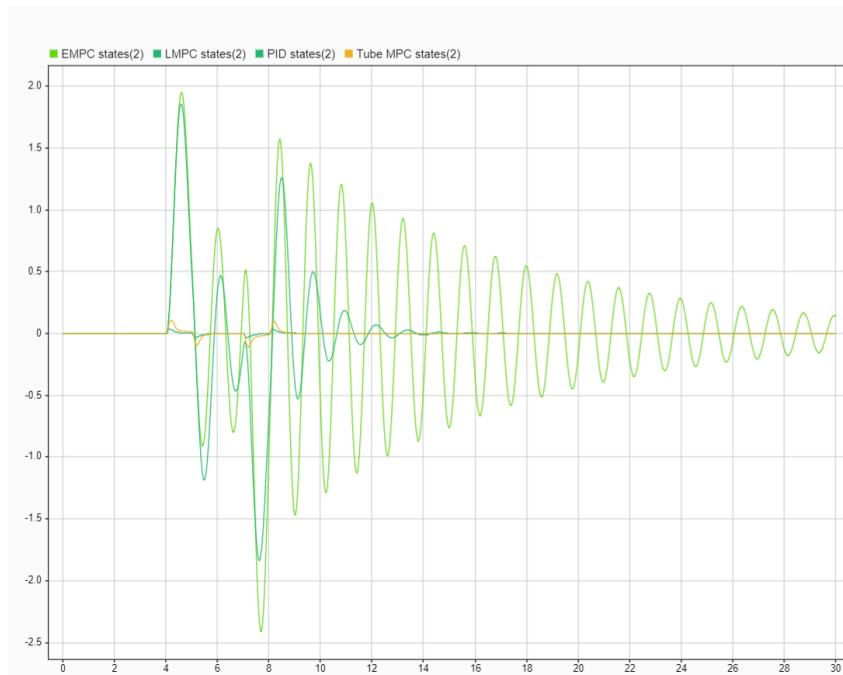
شکل ۲۷: ۷

در نهایت، به مقایسه ی نتایج به دست آمده از کنترلر های ارائه شده برای این سیستم خواهیم پرداخت.
مقایسه حالت اول:



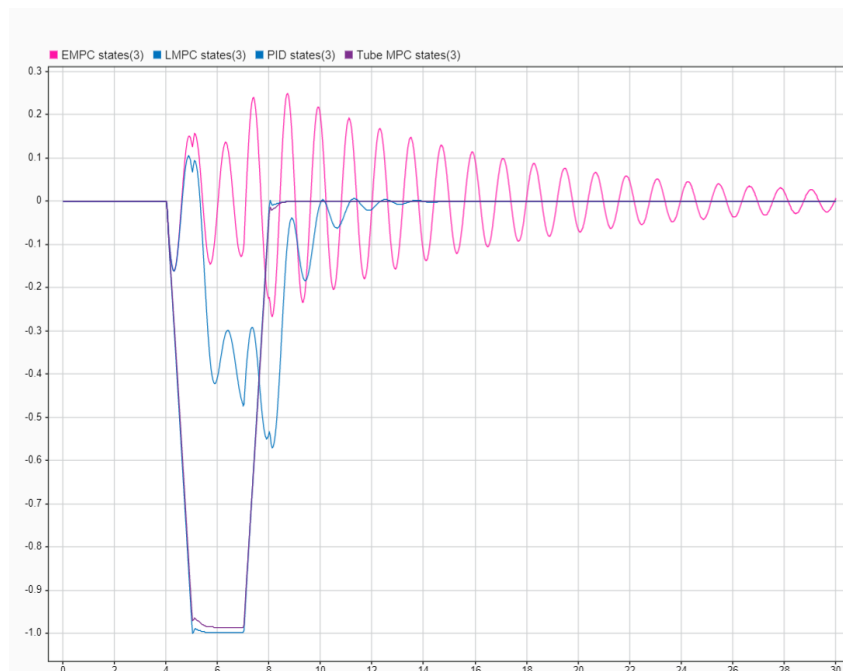
شکل ۲۸

مقایسه حالت دوم:



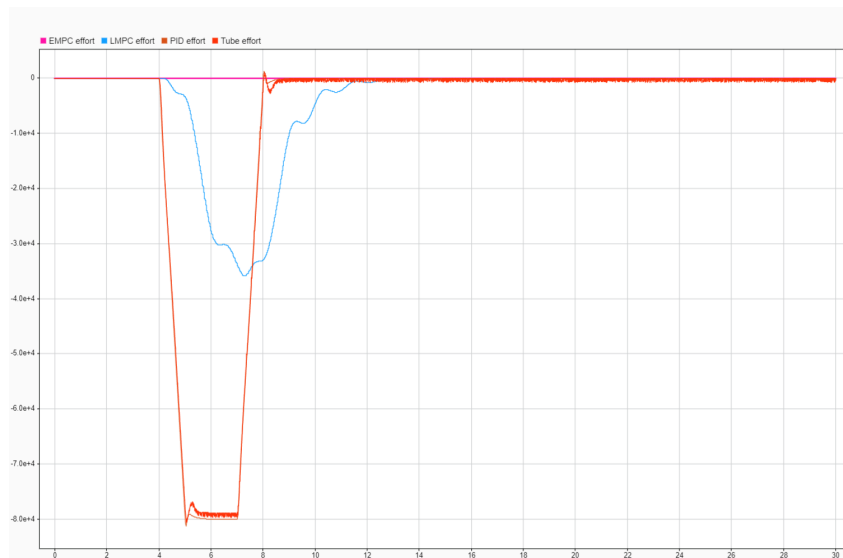
شکل ۲۹

مقایسه حالت سوم:



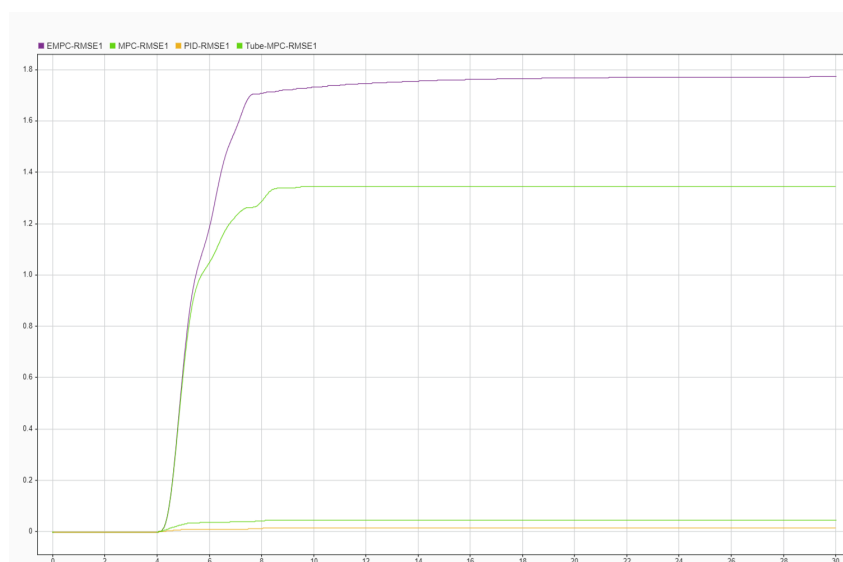
شکل ۳۰

مقایسه تلاش کنترلی:



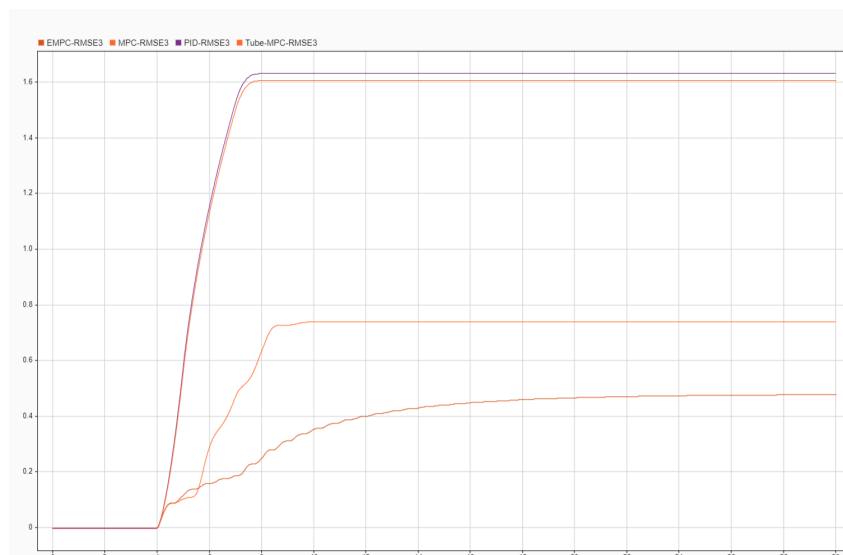
شکل ۳۱

مقایسه RMSE حالت ۱:



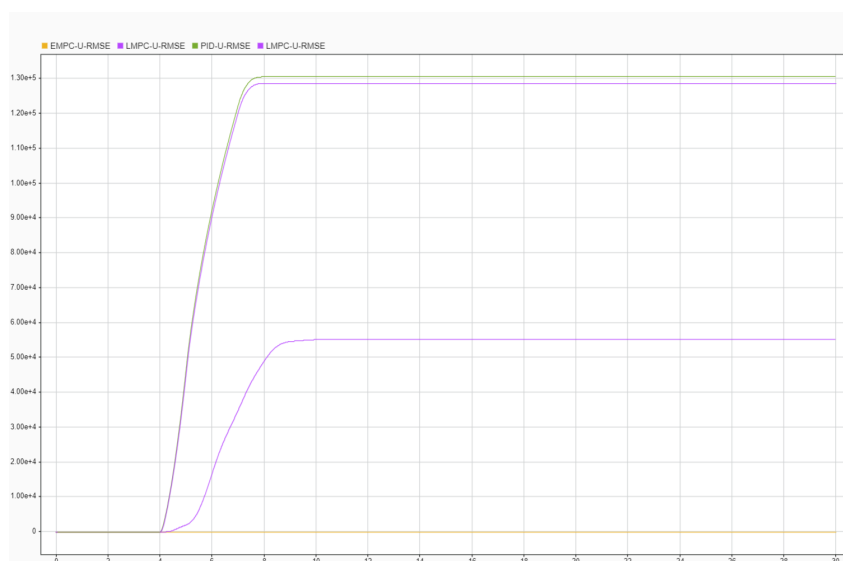
شکل ۳۲

مقایسه RMSE حالت ۳:



شکل ۳۳

مقایسه RMSE کنترلر:

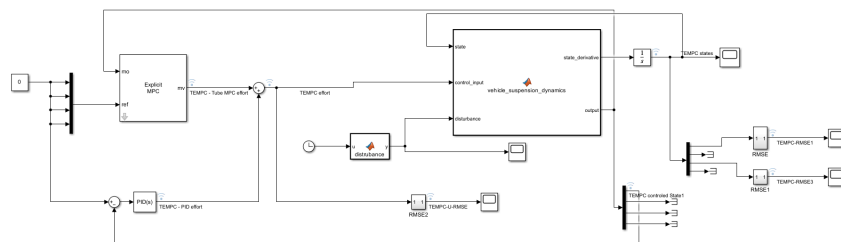


شکل ۳۴

بخش چهارم

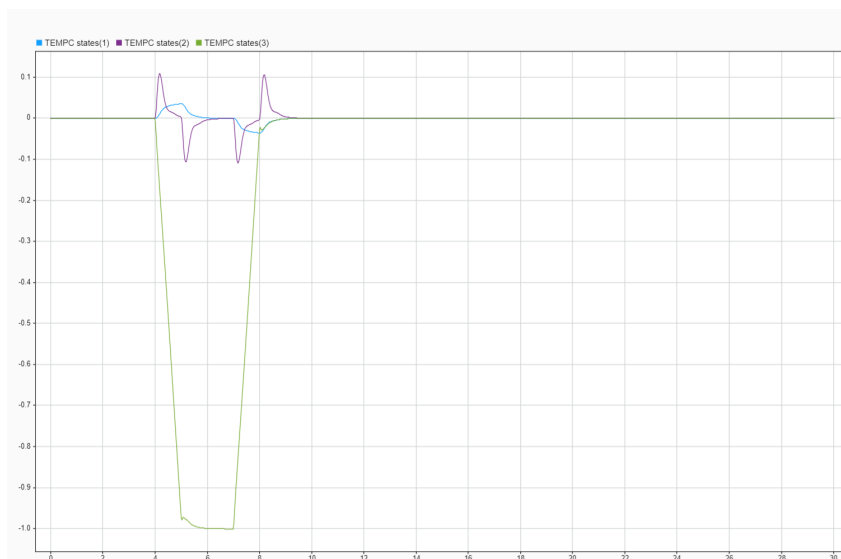
در این بخش، یک کنترلر $TubeexplicitMPC$ با اضافه کردن کنترلر PID به سیستم طراحی خواهد شد. ساختار

دیاگرام این سیستم به صورت زیر است:



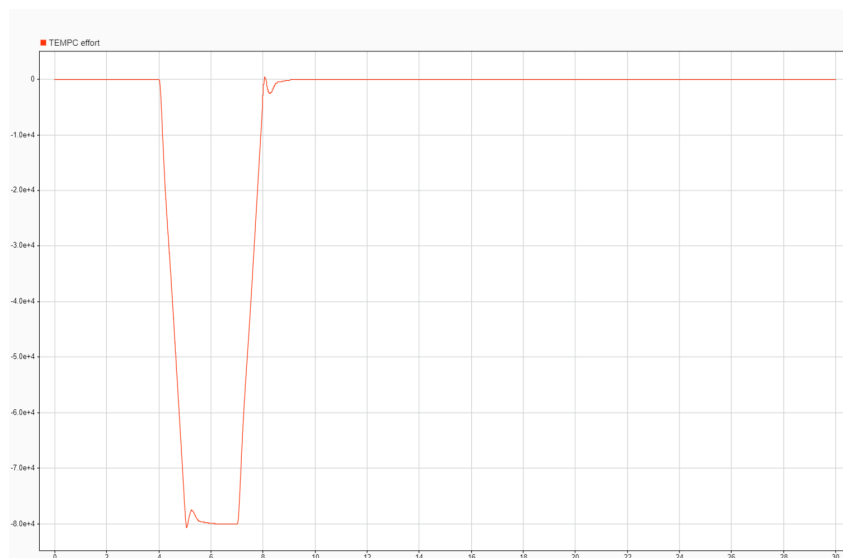
شکل ۳۵

ضرایب PID مانند کنترلر *TubeMPC* طراحی شده است و بلوک EMPC نیز با استفاده از تنظیمات نمایش داده شده در بخش قبل در دیاگرام قرار گرفته است. پاسخ های به دست آمده از این سیستم در شکل زیر نمایش داده شده است.



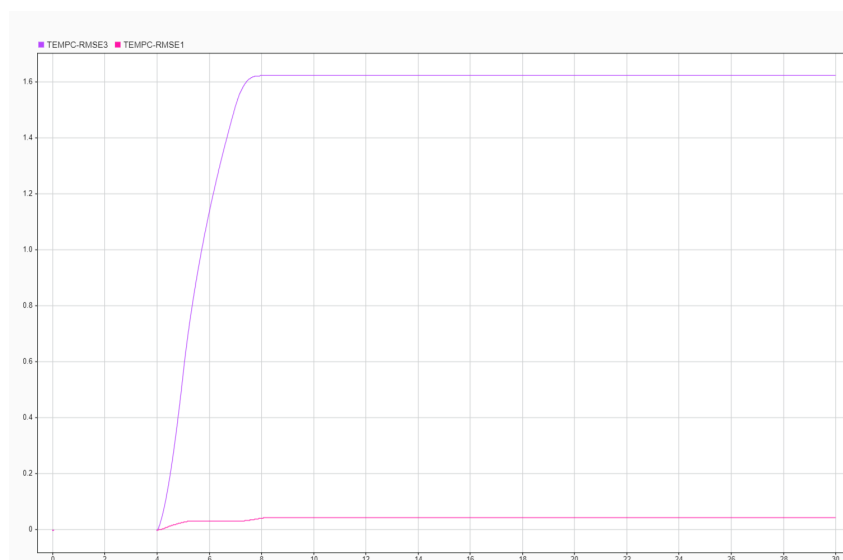
شکل ۳۶

همچنین نمودار تلاش کنترلی این سیستم به صورت زیر است:



شکل ۳۷

و مقادیر RMSE به دست آمده برای این سیستم در شکل زیر نمایش داده شده است:

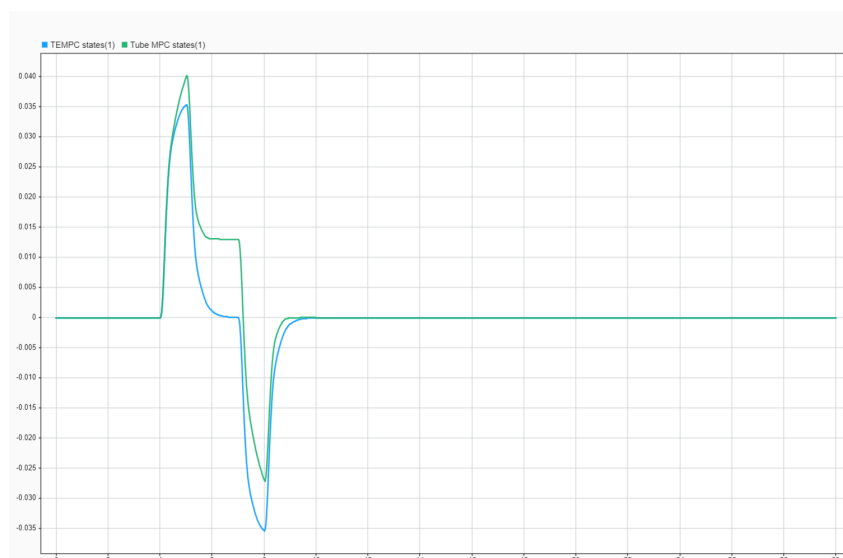


شکل ۳۸



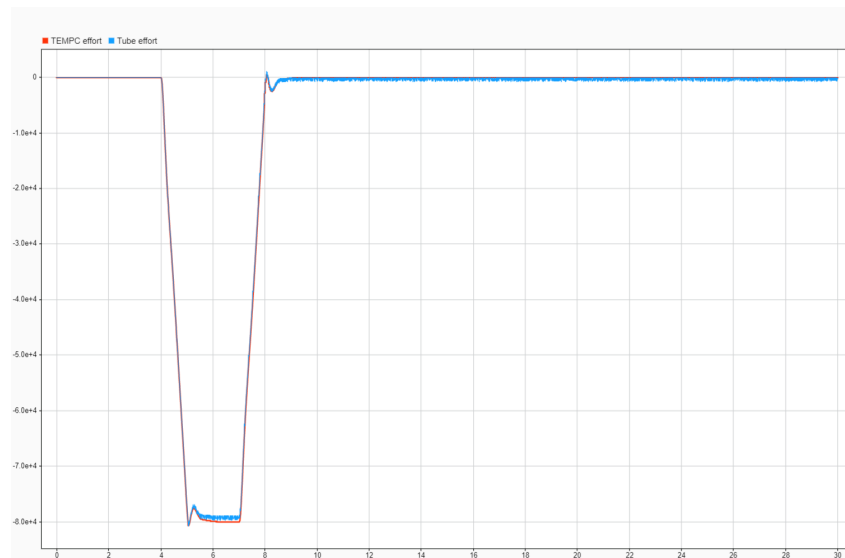
شکل ۳۹

در پایان این بخش، به مقایسه‌ی نتایج به دست آمده از کنترلر TEMPC با کنترلر TMPC خواهیم پرداخت. نمودار خروجی حالت اول سیستم با استفاده از این دو کنترلر به صورت زیر است:



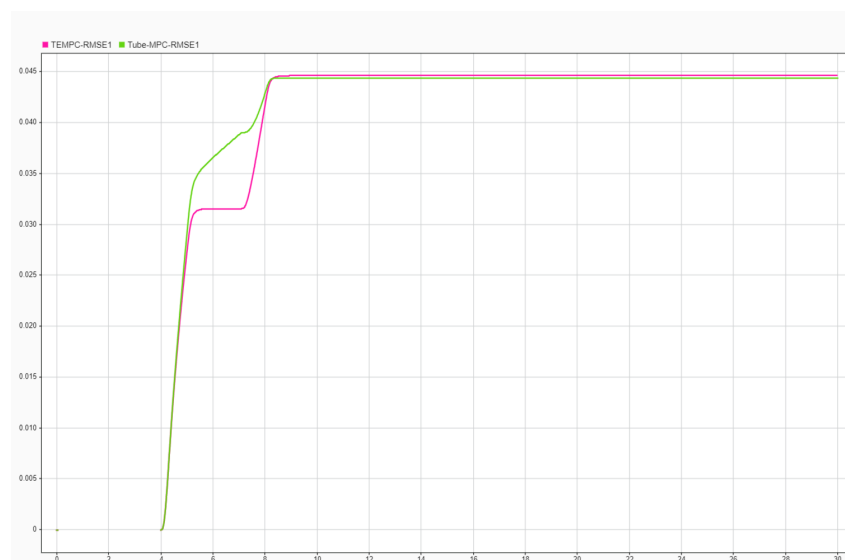
شکل ۴۰

همچنین، نمودار تلاش کنترلی برای این دو در شکل زیر نمایش داده شده است:

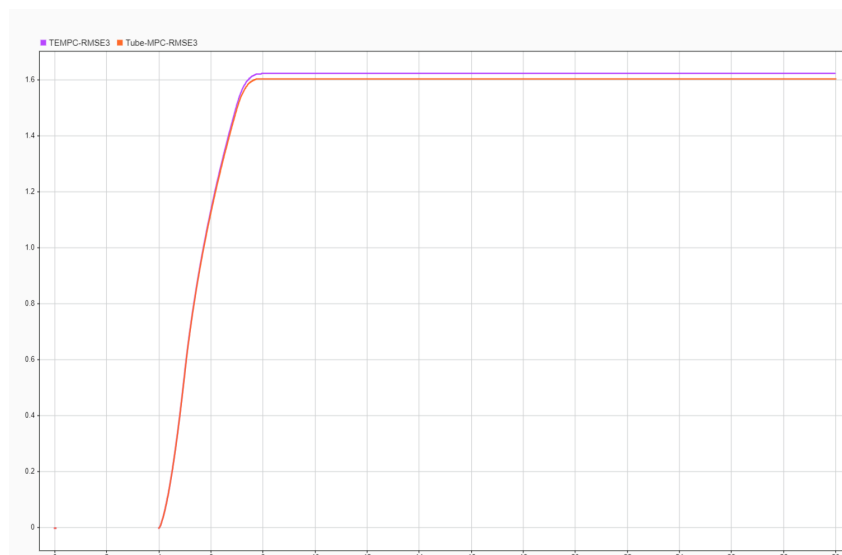


شکل ۴۱

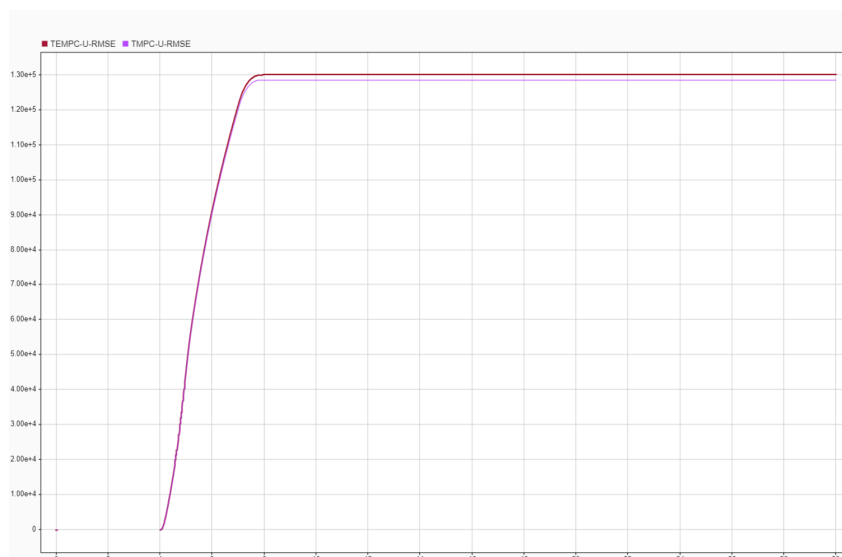
در نهایت، مقادیر RMSE به دست آمده برای این دو کنترلر به صورت زیر است:



شکل ۴۲



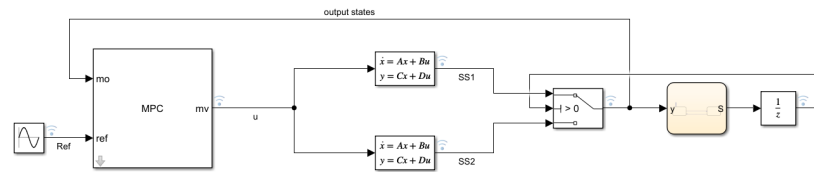
شکل ۴۳



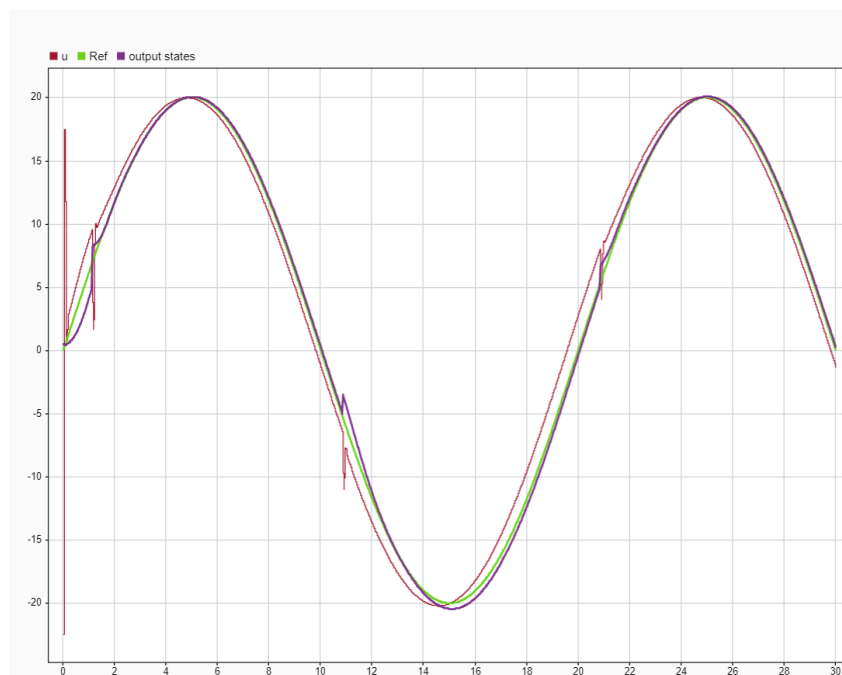
شکل ۴۴

۱.۰ پاسخ سوال دو

برای طراحی کنترلر $HybridMPC$ ، برای سیستمم ارائه شده، لازم است تا با استفاده از یک سویچ، تو بلوک حالت سیستم را به صورت موازی در سیستم قرار داده و سپس با طراحی یک ماشین حالت، جابه جایی میان آن دورا ممکن ساخت. بنابراین، دیاگرام سیستم به صورت زیر تشکیل می شود. با قرار دادن شرط سویچ برای ماشین حالت به صورتی که اگر مقدار خروجی کمتر از -۵ باشد، در مد اول و اگر بیشتر از ۵ باشد در مد کاری دوم باشد، معین می شود. سپس، با طراحی کنترلر MPC با افق پیش بین ۱۱۰ و افق کنترلی ۸ سیستم کنترل می شود. لازم به ذکر است که در



شکل ۴۵: دیاگرام کنترلر هیبرید



شکل ۴۶

این بخش، قید های خواسته شده توسط سوال برای مقادیر ورودی بازه ی -30 و 30 در نظر گرفته شده است. با اجرای سیستم، خروجی های کنترلر، مقدار رفرنس و خروجی سیستم به صورت زیر به دست می آیند و مشاهده می شود که کنترلر توانسته است به خوبی خروجی را کنترل کند. اگرچه، در نقطه ی تغییر مد سیستم، نوسانی مشاهده می شود که با تلاش کنترلی، به راحتی حذف شده است.