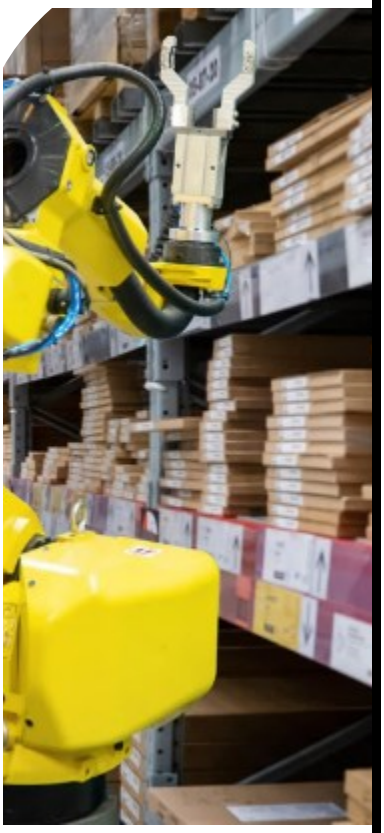
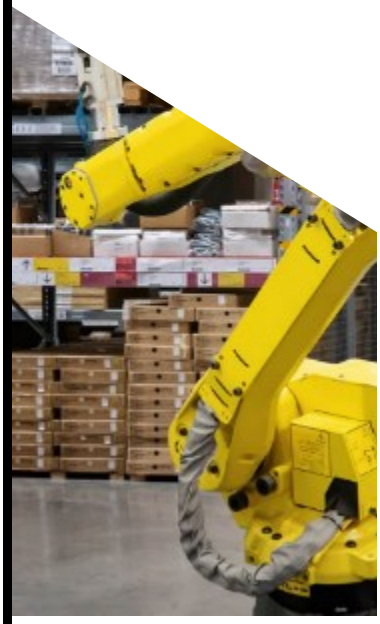
به نام خدا



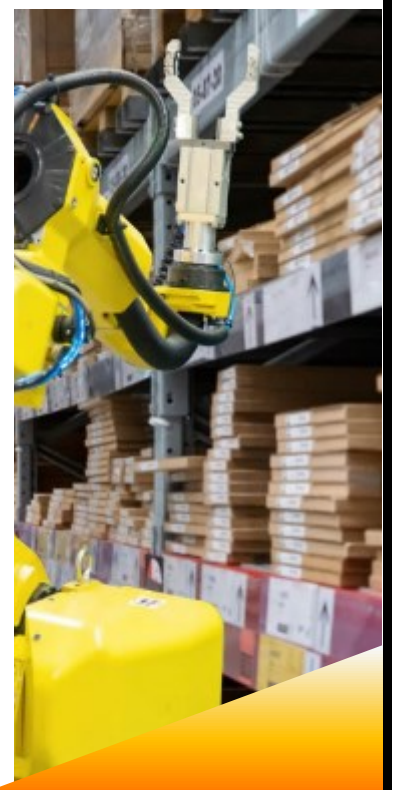


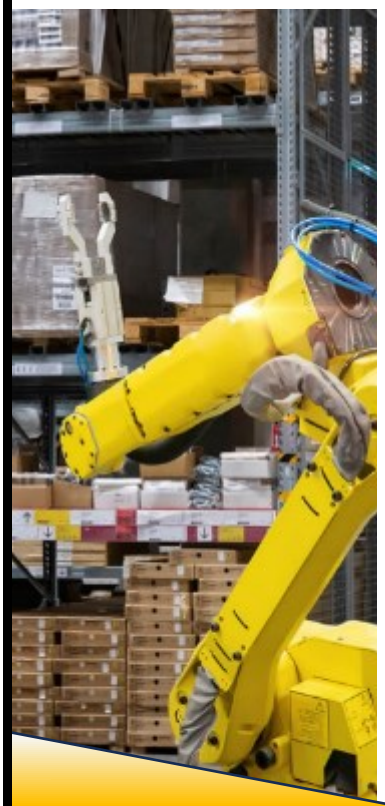
گزارش تمرین شماره 8 رباتیک

نام دانشجو : **عرفان رادفر**

شماره دانشجویی : **99109603**

استاد درس : **دکتر سعید بهزادی پور**





پاییز 1402

Erfan Radfar
99109603

Robotics

Dr. Saeed Behzadipour

فهرست

سوال 1

مطلوب است ساخت مدلی از ربات تمرین پنجم شامل مفاصل اول تا سوم به همراه عملگرهای الکتریکی مربوطه. برای این منظور:

۱. یک مدل از مکانیک ربات را با درنظرگرفتن سه لینک به صورت میله یکنواخت(با استفاده از قطعات پیشفرض) در Simscape بسازید. جرم میله‌ها را به ترتیب برای لینک‌های ۱ ، ۲ و ۳ برابر چهار، چهار و دو کیلوگرم در نظر بگیرید. طول لینک سه را برابر ۳۰ سانتی متر در نظر بگیرید. (باقی ابعاد و مختصات صفر را مطابق تمرین پنجم درنظر بگیرید.)

با توجه به خواسته مسئله ، ربات و لینک های آن را در جعبه افزار Simulink ، بخش simscape طراحی می‌کنیم. این مدل را در model_1.slx مشاهده می‌کنید. البته قبل اجرا کردن مدل، این بخش از کد تابع main را اجرا کنید تا پارامترهای مدل در workspace ذخیره شوند.(در تصویر مدل ربات سمت چپ، اجزا نسبت به ورودی صفر جابه‌جا شده اند و در تصویر راست، تمام مفاصل در اندازه صفر قرار دارند.)

```
Ra=50;
Jm=0.1;
Jg=0.05;
r=50;
Bm=0.01;
Km=7.5; Kb=Km;
J=Jg+Jm;
B=Bm+Km*Kb/Ra;
radius=0.010 ; % in meters
max_v=1e6; %volt
```

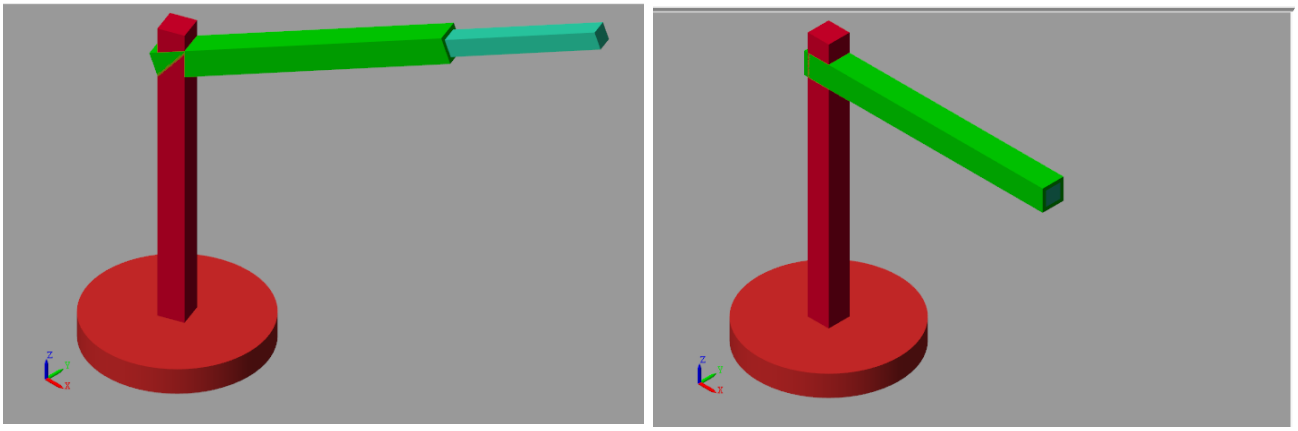


Figure 1 :model_1

سوال 2

۲. به کمک روابط ارائه شده برای موتور DC و جدول زیر مدلی از موتور مربوطه ساخته و به مفاصل ربات متصل نمائید. ربات را در موقعیت اولیه معادل: $\theta_1 = 0 \ \ ,\theta_2 = 0 \ , \ d = 0$

قراردهید و ولتاژ ۴۸ ولت را به هرسه موتور به مدت ۳ ثانیه اعمال کنید. زاویه هرمفصل را برحسب زمان رسم کنید . (در همرفتگی لینک‌ها اهمیتی ندارد.)

با توجه به روابط زیر، مدل های مورد نیاز موتور را طراحی می‌کنیم.

$$\frac{\Theta_m(s)}{V(s)} = \frac{K_m/R}{s(J_m s + B_m + K_b K_m/R)}.$$

$$\frac{\Theta_m(s)}{\tau_\ell(s)} = -\frac{1}{s(J_m(s) + B_m + K_b K_m/R)}$$

تابع تبدیل سیستم موتورها که شامل موتور خطی (موتور دورانی ترکیب شده با چرخدنده و شانه) و موتور دورانی(با افزایش گشتاور از طریق گیربکس) بر روی مدل 1 پیاده می‌کنیم و شکل نهایی model_1 به فرمت زیر خواهد بود.

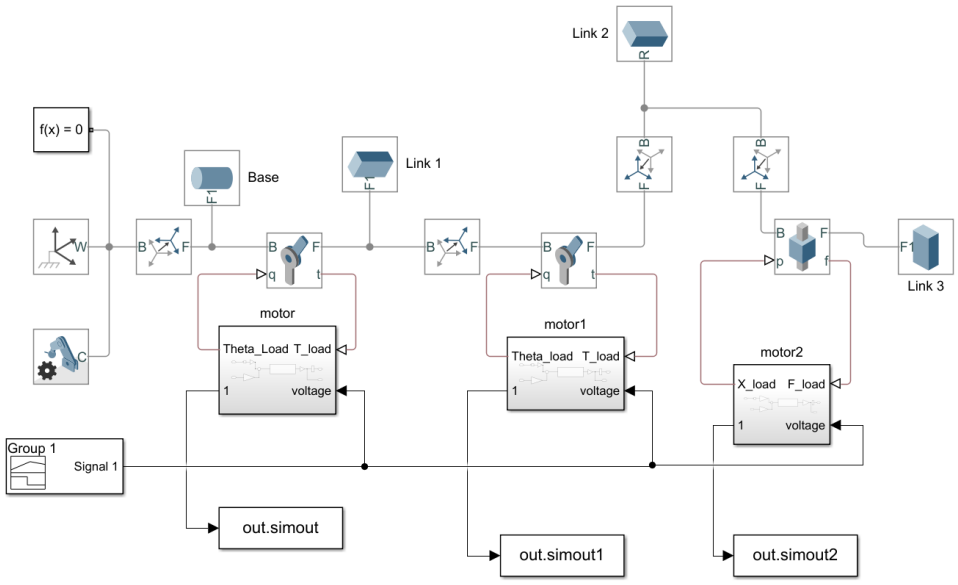


Figure2 : model_1 Simulink diagram

بلوک های motor و motor1 که موتورهای دورانی هستند دارای سیستم داخلی زیر هستند که از معادلات ذکر شده در بالا پیروی می‌کنند.

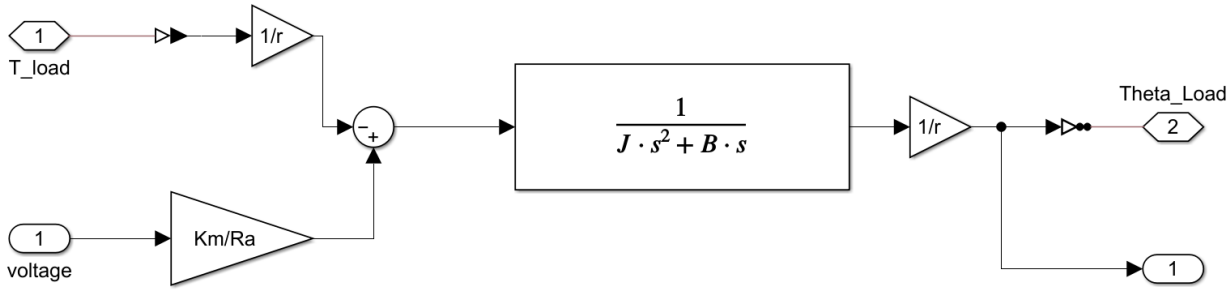


Figure3 : motor & motor1 block

همچنین برای motor2 که موتوری خطی می‌باشد، به جای استفاده از نسبت تبدیل گیربکس، از نسبت تبدیل دوران به جابه‌جایی چرخدنده و شانه (همان شعاع چرخدنده که در اینجا با توجه به ابعاد ربات و توان موتور با radius مقدار 1 سانتی‌متر ارزش گذاری شده است) استفاده می‌کنیم.

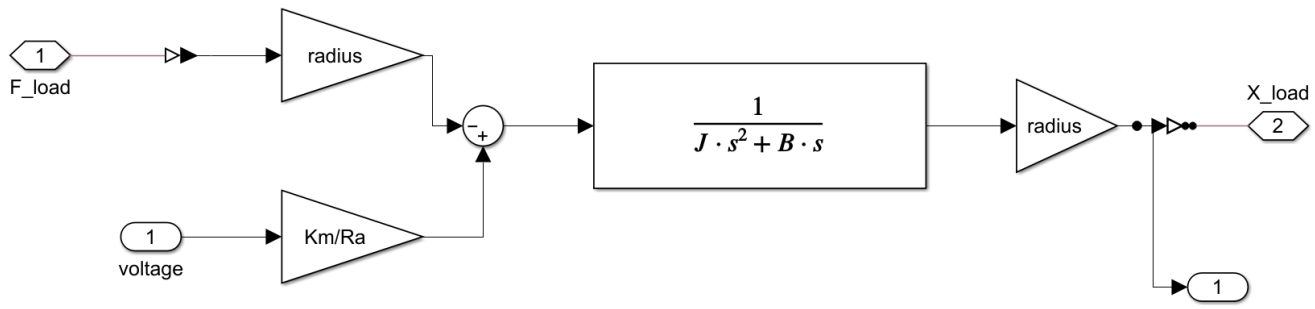


Figure4 : motor2 block

سپس از signal builder استفاده می‌کنیم تا سیگنال ثابت 48V را تا زمان 3 ثانیه اعمال کنیم و پس از آن سیگنال صفر می‌شود.

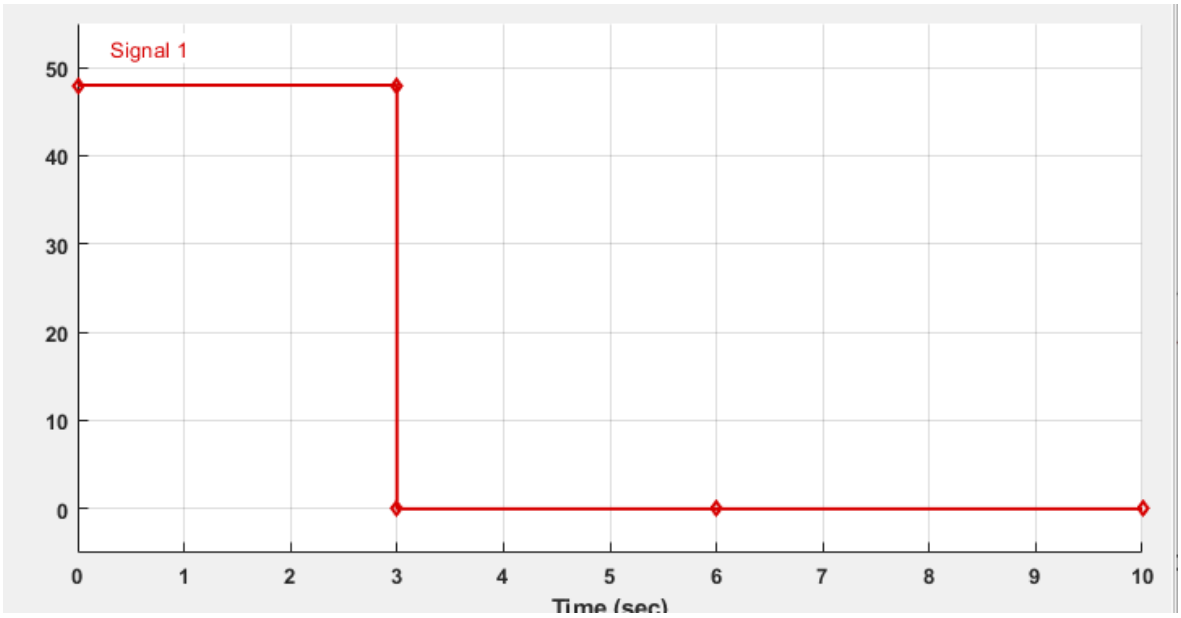


Figure5 : signal builder properties

در نهایت مدل را اجرا کرده و پس از آن با تابع "plott" زاویه هر مفصل را رسم می‌کنیم. تابع plott صرفا خروجی های گرفته شده مفاصل از مدل را ترسیم می‌کند.

```
function plott(out)
t=out.simout.Time;
theta1=out.simout.Data*180/pi;
theta2=out.simout1.Data*180/pi;
d=out.simout2.Data*100;

plot(t,theta1);
xlabel("time_{(S)}");
ylabel("\Theta_1_{(deg)}");
title("\Theta_1_{(time)}");
figure

plot(t,theta2);
xlabel("time_{(S)}");
ylabel("\Theta_2_{(deg)}");
title("\Theta_2_{(time)}");
figure

plot(t,d);
xlabel("time_{(S)}");
ylabel("d_{(cm)}");
title("d_{(time)}");
end
```

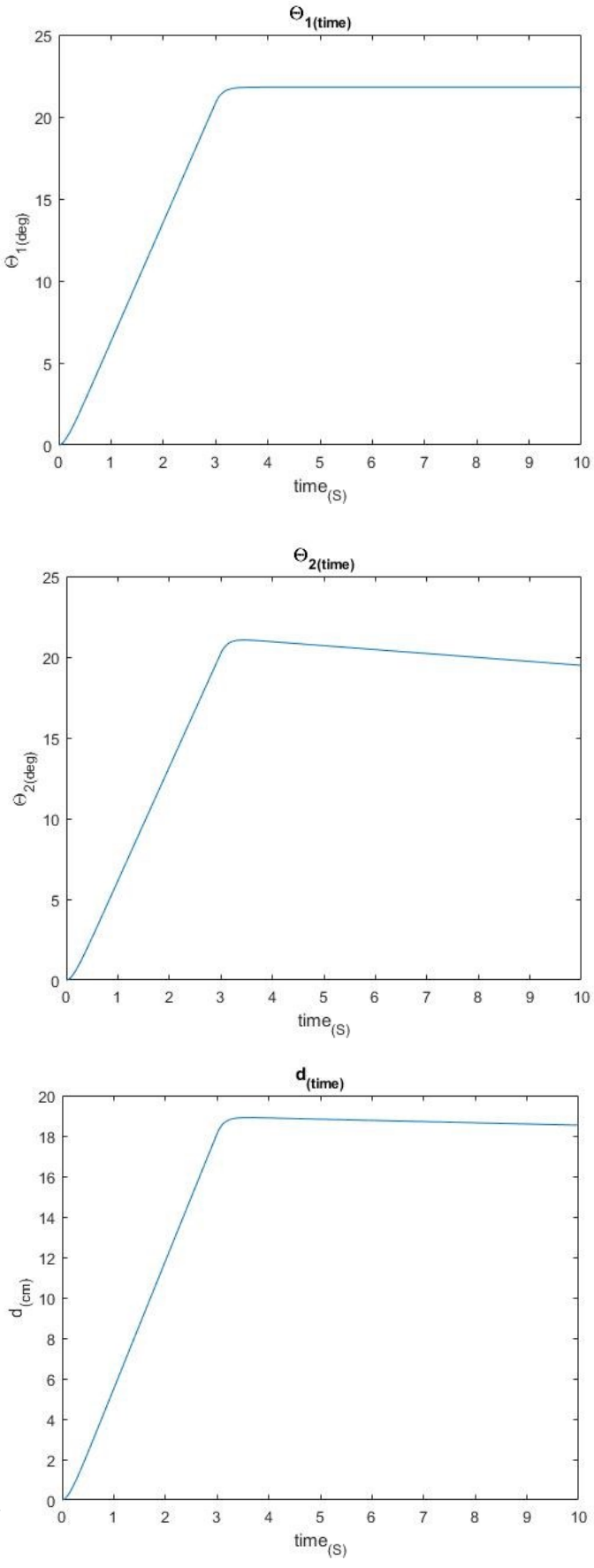


Figure 6 :joints parameters

همانطور که در تصویر بالا مشاهده می‌شود ، تا زمان 3 ثانیه ، هر سه موتور مقادیر مفاصل را افزایش می‌دهند تا زمانی که در انتهای 3 ثانیه، سیگنال ورودی دوباره صفر شود. در این زمان موتورها قطع می‌شوند و مفاصل تحت بار، به آرامی تغییر می‌کنند تا پتانسیل گرانشی به مینیمم برسد.

سوال 3

۳. برای هریک از مفاصل سه گانه ربات، یک کنترلر PID طراحی کنید که زمان نشست ربات حدود ۱ ثانیه و فراجهش آن کمتر از ۵٪ باشد. برای این منظور تابع تبدیل حلقه بسته هر مفصل را بدست آورید و از روابط زیر برای تخمین ضرایب استفاده کنید. (البته نهایتا ممکن است مجبور به تنظیم نهایی ضرایب با سعی و خطا باشید).

$$M_p = e^{-\pi\xi/\sqrt{1-\xi^2}} \quad t_s = \frac{4.6}{\xi\omega_n}$$

کنترلر خود را برای بردن از موقعیت اولیه به

$$\theta_1 = 50 \ , \theta_2 = 35 \ , d = 15\ cm$$

تست کنید. منحنی‌های زاویه مفاصل برحسب زمان را رسم کنید. (طراحی تراژکتوری لازم نیست. مقادیر زاویه‌ها را به صورت ورودی پله به کنترل کننده هر مفصل اعمال کنید).

با توجه به رابطه اوورشوت برای مقدار M=0.05، پارامتر ξ برابر با 0.690 آمده و با توجه به رابطه زمان نشست به مدت 1 ثانیه، ω_n برابر با 6.67 به دست می‌آید. حال ضریب K_i را فعلا صرف در نظر می‌گیریم و کنترلر PD طراحی می‌کنیم. برای این هدف از رابطه زیر استفاده می‌کنیم:

$$\omega = \left(\frac{K_p}{J}\right)^{\frac{1}{2}} \qquad \xi = \frac{1}{2}\frac{B+K_d}{J}\left(\frac{J}{K_p}\right)^{\frac{1}{2}}$$

با توجه به این مقادیر، به K_p=6.67 و K_d=0.246 می‌شود. (با توجه به داده های مسئله، J=0.15 و B=1.135 می‌باشد)

با توجه به اینکه این روابط برای کنترل زاویه موتور است و نه مفصل، باید ضریب r=50 در آن ضرب شود. سپس تیونینگ را با نرم افزار PID Tuner انجام می‌دهیم و مقدار K_i را هم در نظر می‌گیریم و در نهایت به مقادر زیر برای هر کدام از ضرایب PID می‌رسیم(اندیس 1). این فرایند برای کنترلر موتور خطی نیز تکرار می‌شود (اندیس 2).

```
Kd1=145; Ki1=10; Kp1=1963;
Kd2=97; Ki2=10; Kp2=2377;
```

مقادیر نهایی

حال کنترلر ها به مفاصل متصل می‌کنیم (در model_2)

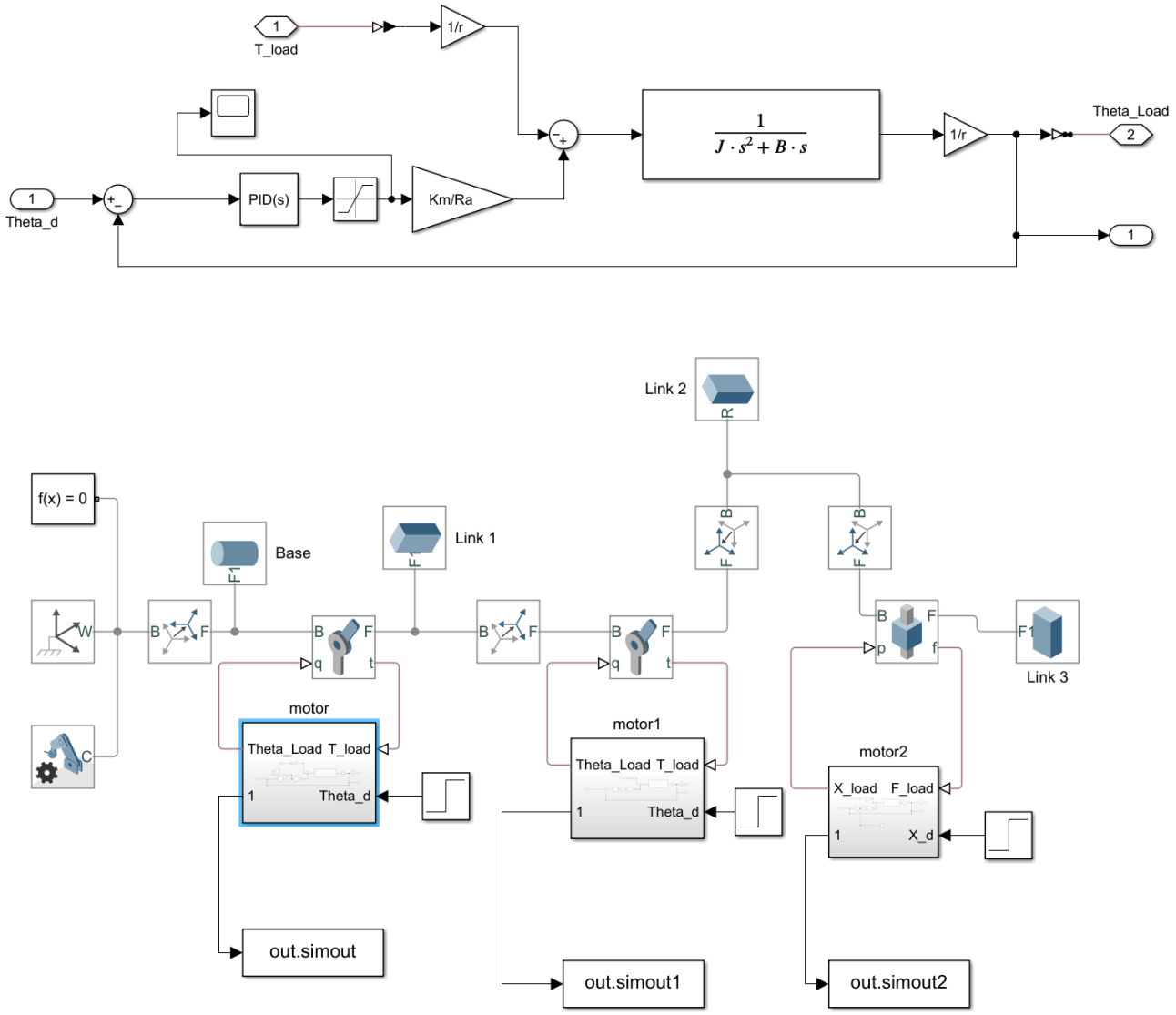


Figure 7 : model_2 diagram & motor block

ورودی های دلخواه به صورت پله، به بخش ورودی desired بلوک موتورها داده شده است. سپس کنترلر ولتاژ را طوری تنظیم می‌کند تا مفاصل ربات به موقعیت های خواسته شده برسند.

در نهایت کد این بخش را اجرا کرده و بعد از آن مدل را ران می‌کنیم. در نهایت با دستور plott ، نمودارهای مفاصل را رسم می‌کنیم.

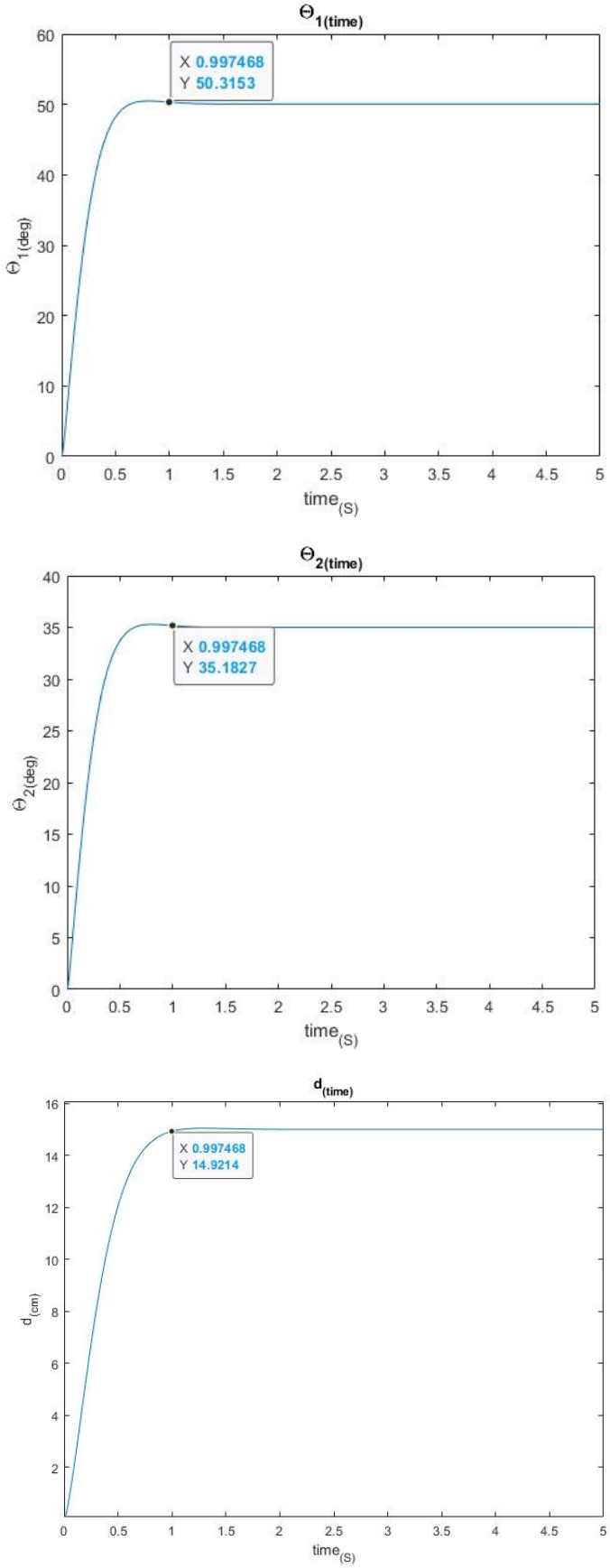


Figure 8 : Joints diagram & settling times

همانطور که در نمودارها مشاهده می‌شود، زمان نشست نزدیک به 1 ثانیه بوده و اووشوت کمتر از 5٪ است.

سوال 4 (امتیازی)

۴. تراژکتوری طراحی شده در قسمت الف تکلیف۵ را روی ربات پیاده کنید و خطای آر ام اس مسیر، نسبت به سوال ۳ را گزارش کنید. (موقعیت اولیه و پایانی مسیر را مختصات تکلیف۵ در نظر بگیرید.) (امتیازی ۲۵٪)

برای این قسمت باید طی یک مسیر نرم با درجه جمله 3 نسبت به زمان، تراژکتوری خواسته شده بین دو نقطه زیر را ایجاد کنیم.

X_i (mm)	Y_i (mm)	Z_i (mm)
284.829	-164.446	190.293

X_f (mm)	Y_f (mm)	Z_f (mm)
91.203	340.373	605.682

برای ایجاد این مسیر از کد تمرین 5 استفاده می‌کنیم تا ورودی desired هر مفصل بر حسب زمان به دست آید.

```
%% Problem 4
% First run this
Kd1=145; Ki1=10; Kp1=1963;
Kd2=97; Ki2=10; Kp2=2377;
L1=310; %mm
L2=300; %mm
position=[284.829 -164.446 190.293 ;
          91.203 340.373 605.682 ];
parameters=zeros(3,2);
a=zeros(3,4);
s=1; % in seconds
for k=1:2
    x=position(k,1);
    y=position(k,2);
    z=position(k,3);
    t1=atan2(y,x);
    if abs(x)<1e-3
       t2=atan( (z-L1)*sin(t1)/y );
       d=y/(sin(t1)*cos(t2))-L2;
    else
       t2=atan( (z-L1)*cos(t1)/x );
       d=x/(cos(t1)*cos(t2))-L2;
    end
    parameters(:,k)=[t1;t2;d];
end

%Now we calculate trajectory coefficient
a(:,1)=parameters(:,1);
%a(:,2) remain zero
a(:,4)=(-parameters(:,2)+a(:,1))/(0.5*s^3);
a(:,3)= -1.5*a(:,4)*s;
%Now we generate trajactory inputs
```

```
t=0:0.01:1;
f=@(x) a(:,1)+a(:,2)*x+a(:,3)*x.^2+a(:,4)*x.^3;
output=f(t); % outputs are in mm and rad
joints_traj=[t',output(1,:)',t',output(2,:)',t',output(3,:)'];
```

سپس این ورودی ها را، در model_3 در بلوک Ideal Trajectory به مفاصل می‌دهیم تا مسیر ایده‌آل را برای محاسبه خطای rms داشته باشیم.

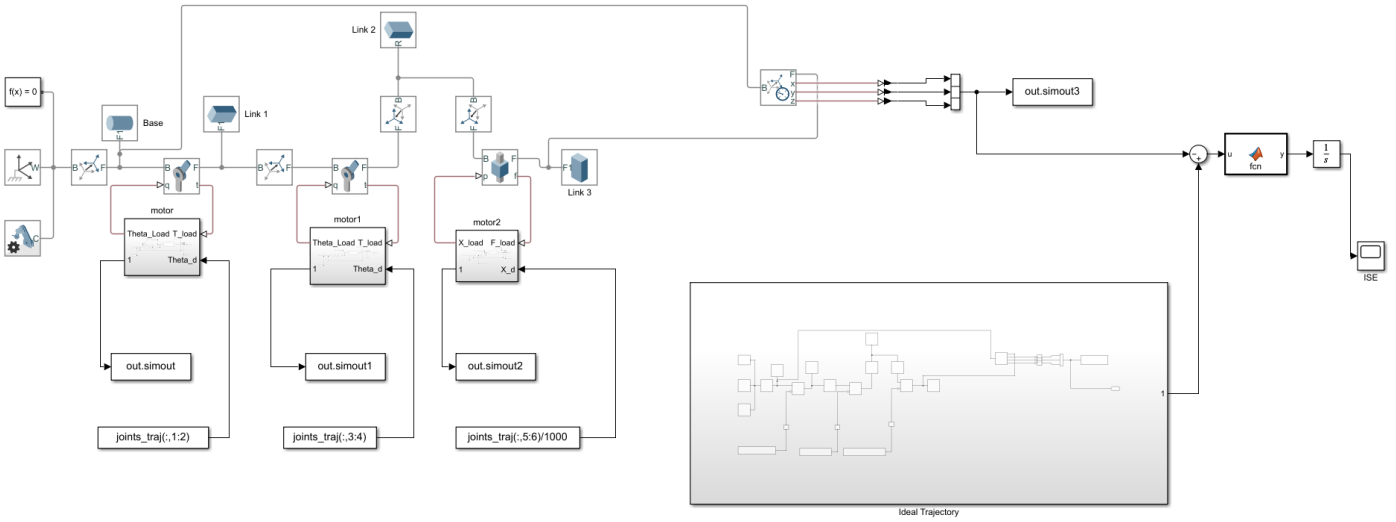


Figure 9 :model_3

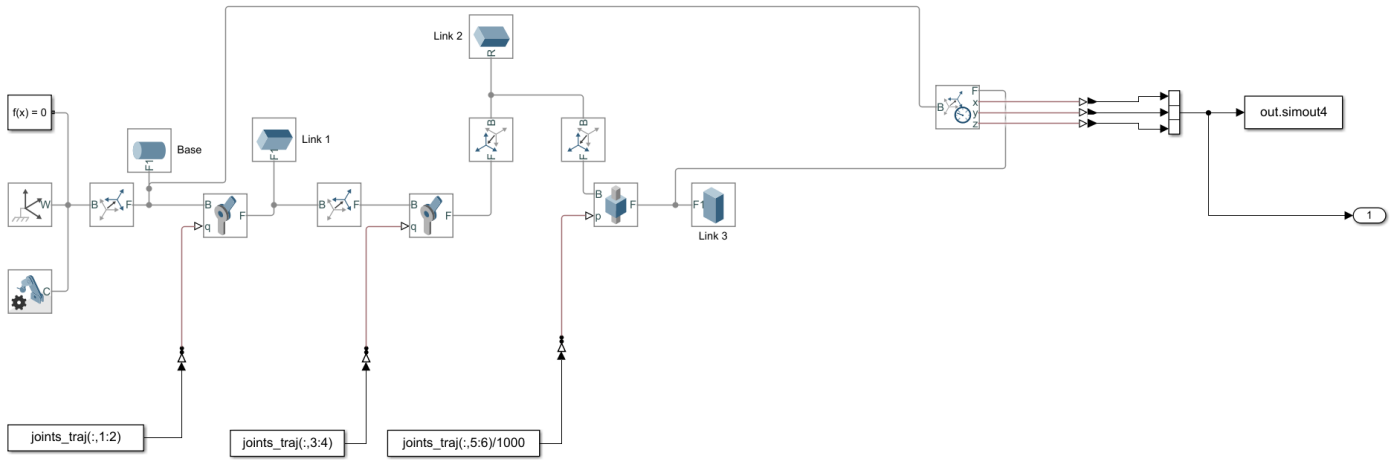


Figure 10 : Ideal Trajectory block

برای اجرا کردن model_3 از همان کنترلرهایی استفاده می‌کنیم که در سوال 3 استفاده کردیم اما باید دقت کرد که موقعیت اولیه مفاصل صفر نخواهد بود، درحالی که کنترلر با فرض ورودی صفر طراحی می‌شود. برای رفع این مشکل، در بلوک موتورهای کنترل شده به فرم زیر تغییراتی ایجاد می‌کنیم.

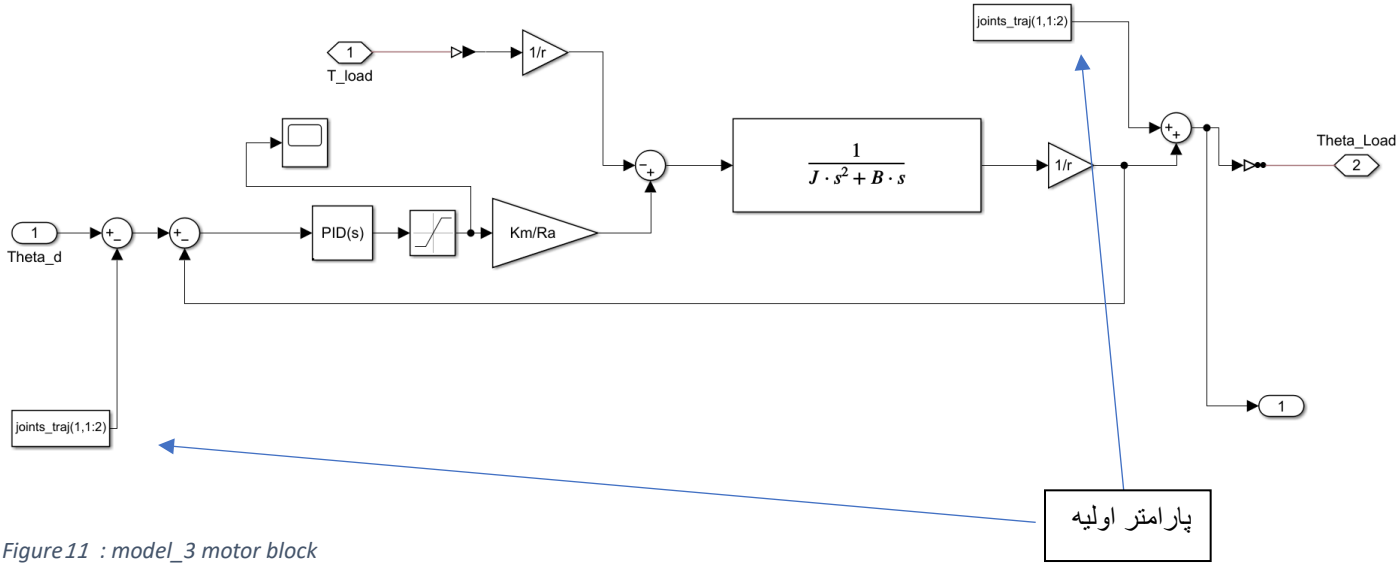


Figure 11 : model_3 motor block

در ابتدا فرض می‌کنیم که موتور واقعا در موقعیت گفته شده طبق trajectory قرار دارد. در نتیجه زمانی که موتور، ورودی دریافت می‌کند، از آن پارامتر موقعیت اولیه را کم می‌کنیم. به این شکل اگر کاربر بخواهد ربات در موقعیت خود بماند، هیچ خطایی مشاهده نمی‌شود. اما باید در انتها مقدار اولیه به خروجی سیستم کنترلری اضافه شود تا موقعیت واقعی به عنوان ورودی به مفصل داده شود. در نهایت نیز خروجی بلوک تراژکتوری ایده‌آل و سیستم کنترل شده از یکدیگر کم می‌شوند و اندازه بردار اختلاف بر حسب زمان، انتگرال گرفته می‌شود.(تابع matlab تعریف شده در Simulink کار تابع ()norm به توان 2 را انجام می‌دهد)

در نتیجه نمودار انتگرال مجذور خطا، به شکل زیر خواهد بود:

می‌توان مشاهده کرد که اندازه انتگرال حداکثر تا 0.036 پیش خواهد رفت. در نتیجه خطای rms برابر می شود با :

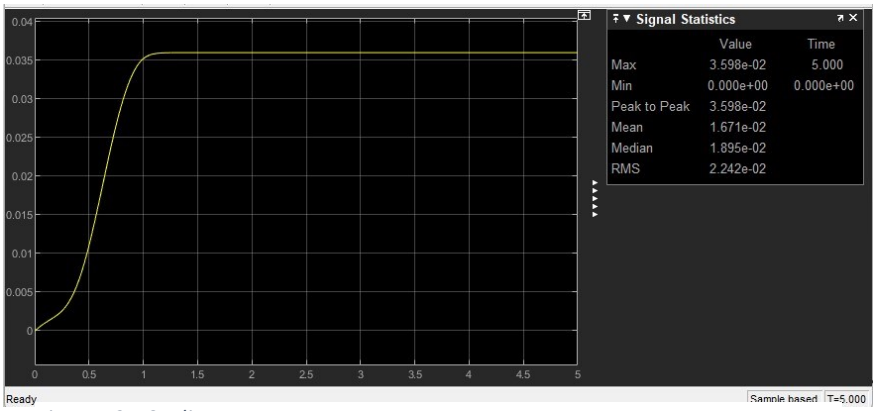
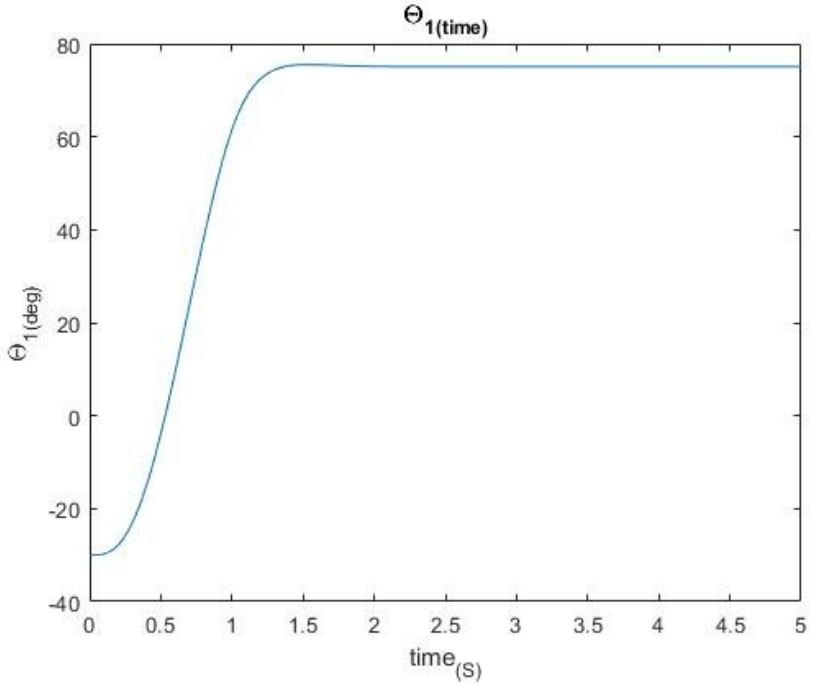


Figure 12 : ISE diagram

$$y_{RMS} = \sqrt{\frac{1}{T}\int_0^T y^2(t)dt}$$

$$E_{rms} = \left(\frac{ISE}{T}\right)^{\frac{1}{2}}$$

چون زمان نشست T برابر با 1.5 ثانیه است، rms برابر با 0.155 متر خواهد بود. این به این معناست که کنترلر به طور متوسط در بازه نشست می‌تواند به صورت محدودی تراژکتوری را دنبال کند اما با تاخیر در مرتبه زمان نشست، به آن می‌رسد. در نهایت با اجرا plottt که مانند plott عمل می‌کند ولی خطای تراژکتوری و مقدار واقعی موقعیت اندفکتور را نیز می‌دهد، به نمودار های زیر می‌رسیم.



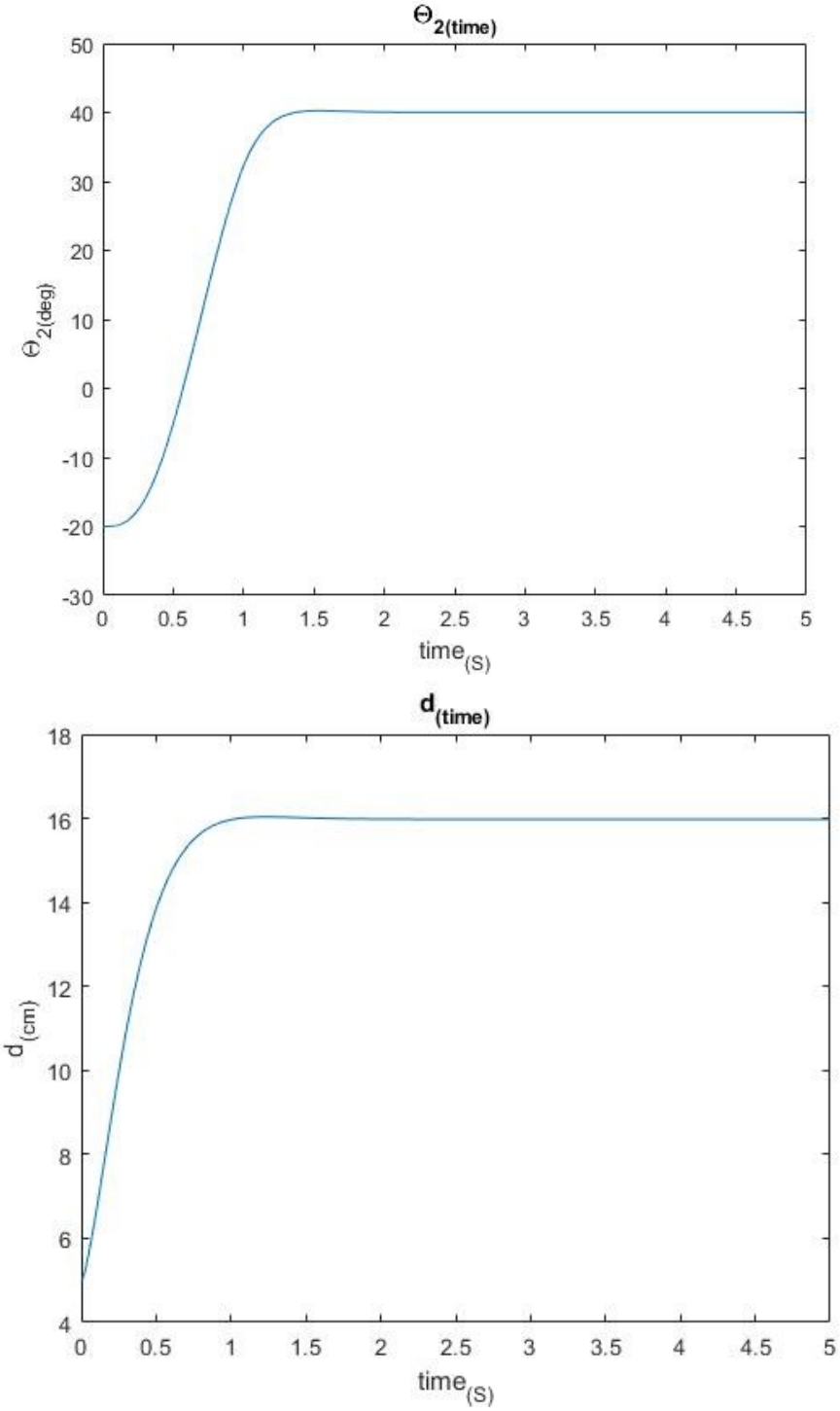


Figure 13 : Joints parameters in model_3

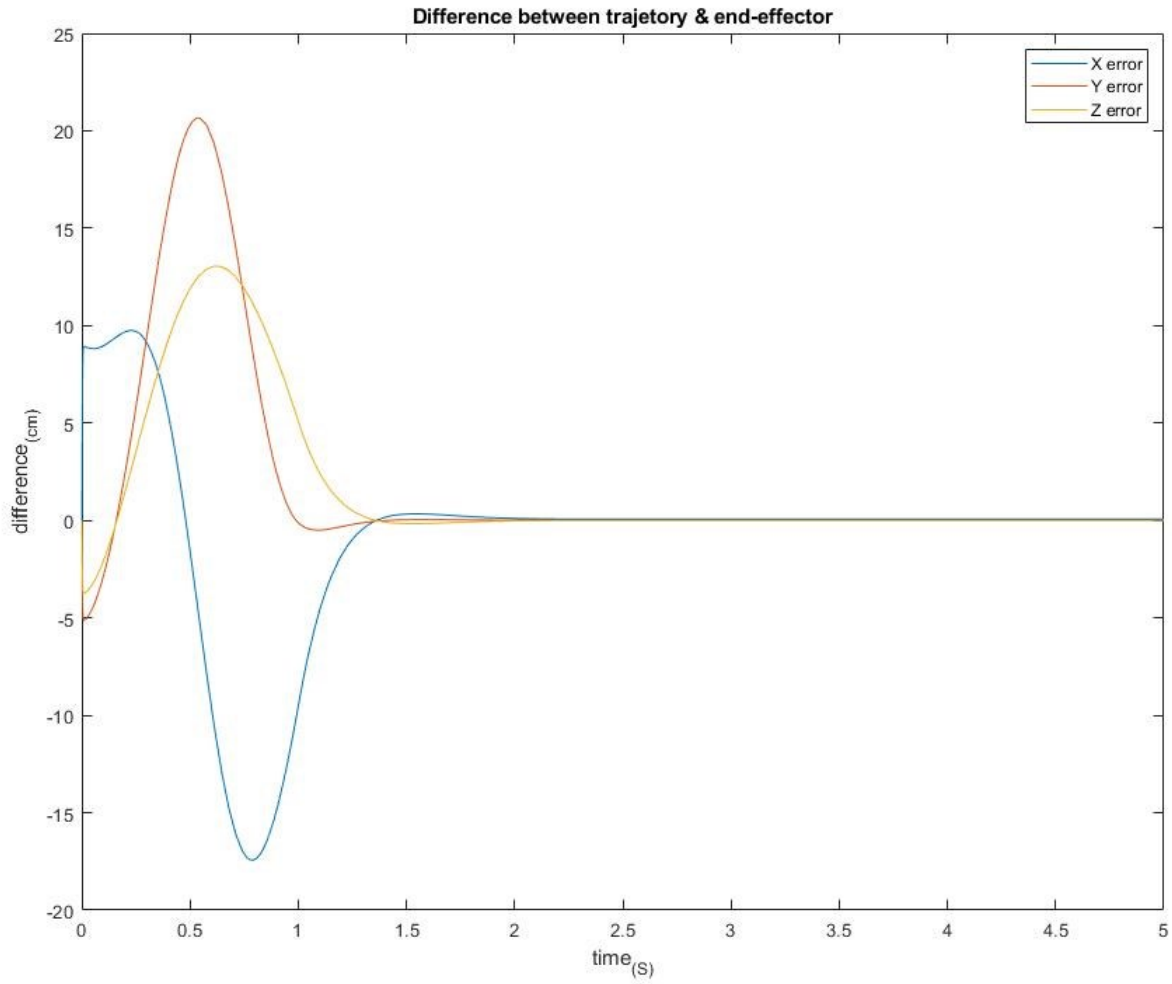


Figure 14 : Trajectory & end-effector difference

همچنین باید توجه کرد که پس از 1 ثانیه، موقعیت desired همان موقعیت نهایی بماند. برای این کار، در بلوک from workspace باید این تنظیم را انجام دهیم.