



دانشگاه خواجه نصیرالدین طوسی  
دانشکده برق - گروه رباتیک و الکترونیک

گزارش رباتیک دوره کارشناسی ارشد

رشته مهندسی مکاترونیک

عنوان

گزارش بخش سوم پروژه رباتیک

نگارش

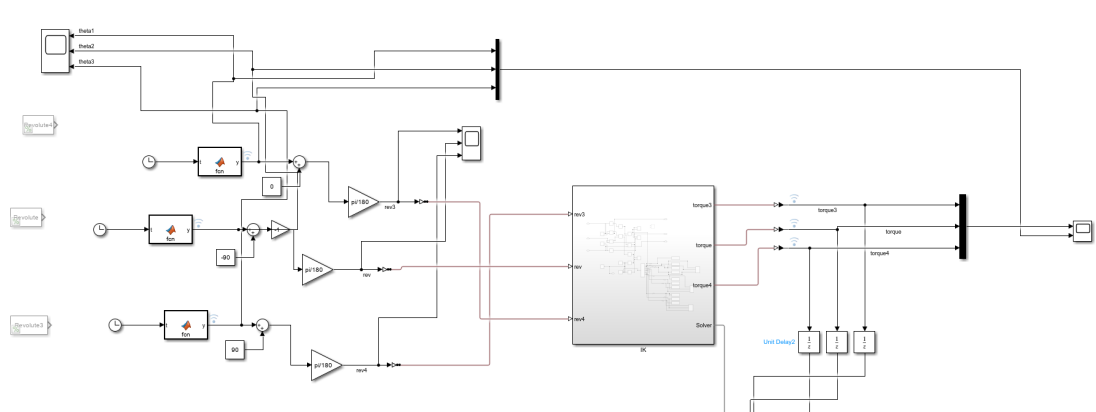
۴۰۲۰۲۴۱۴ علیرضا امیری ایمان گندمی ۴۰۲۲۳۷۰۴

آذر ۱۴۰۳

# پاسخ پروژه بخش سوم

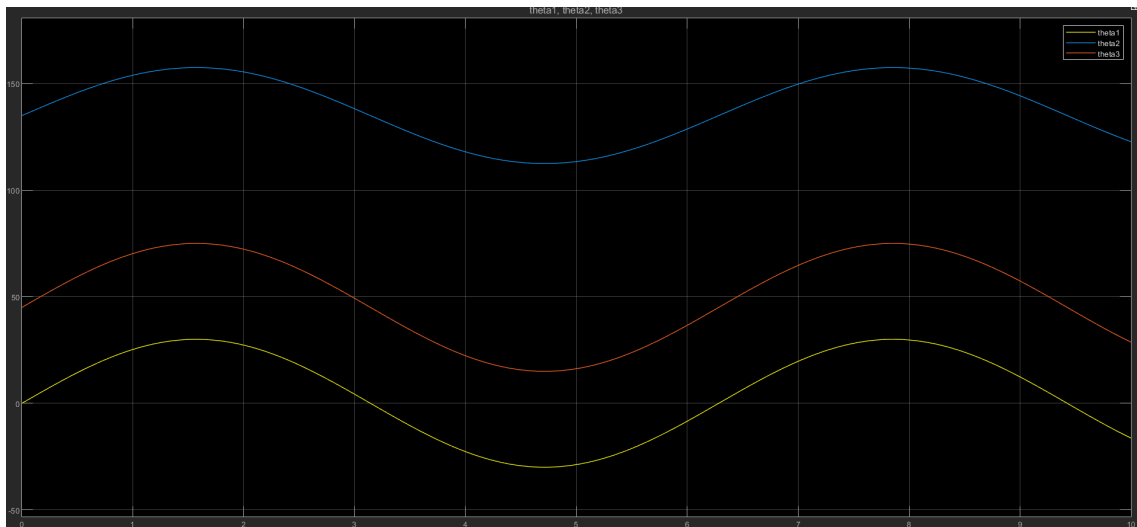
## پاسخ سوال یک، محاسبه دینامیک وارون در شبیه ساز

در این بخش، برای شبیه سازی دینامیک وارون سیستم باید مقادیر فضای کاری به سیستم داده شده و مقادیر گشتاور مفاصل اندازه گیری شود. برای این منظور، با اعمال ورودی به مفاصل در قالب زاویه و اندازه گیری گشتاور تولید شده در هر مفصل، می توانیم دینامیک وارون سیستم را شبیه سازی کنیم. دیاگرام سیستم در این قسمت به صورت صورت زیر طراحی می شود.



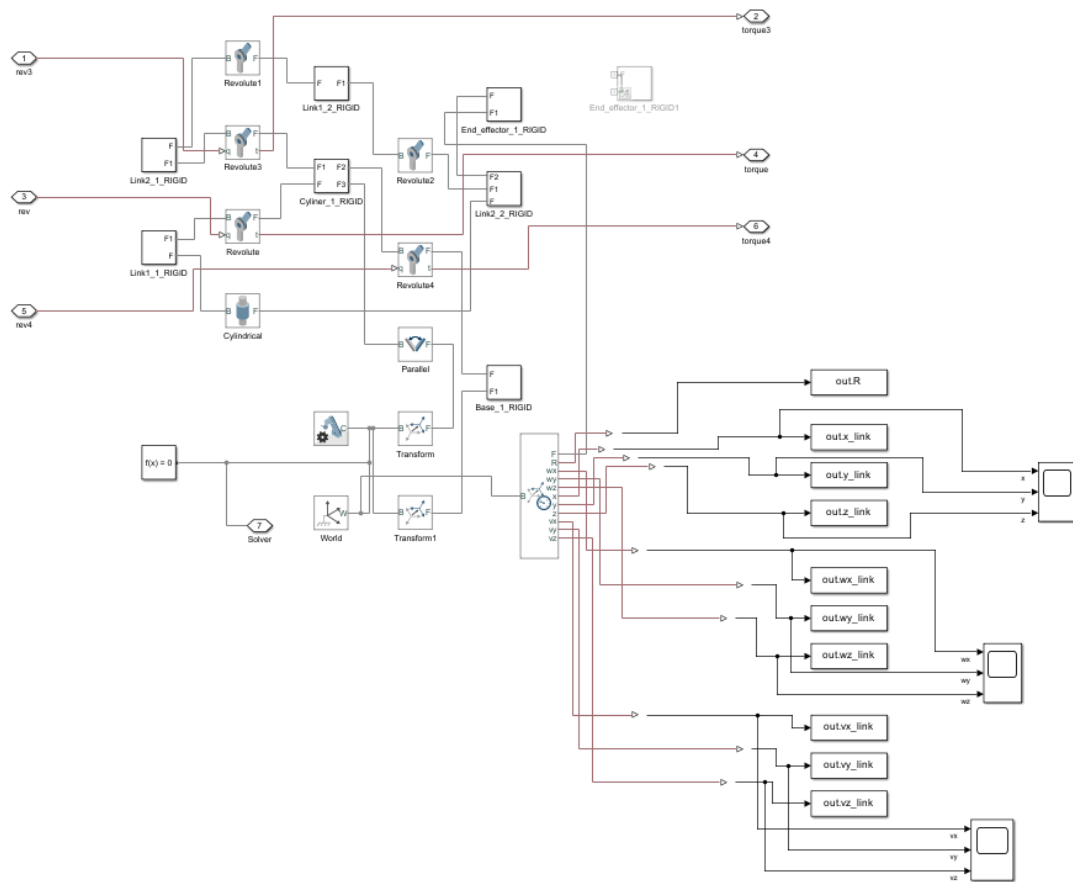
شکل ۱: دیاگرام دینامیک وارون

در این دیاگرام، زوایای سینوسی ورودی بر حسب زمان در ورودی به سیستم وارد می شود. این سیگنال های ورودی در تصویر زیر نمایش داده شده است.



شکل ۲: ورودی های سینوسی به سیستم

در ادامه، برای تبدیل زوایای ورودی به سیستم به رادیان و همچنین تطابق زوایای ابتدایی سیستم، با استفاده از دو بلوک بهره و بایاس این مقادیر مشخص می شود. این مقادیر ورودی در بخش بعد به بلوک دینامیک وارون سیستم وارد می شود که در تصویر زیر نمایش داده شده است.



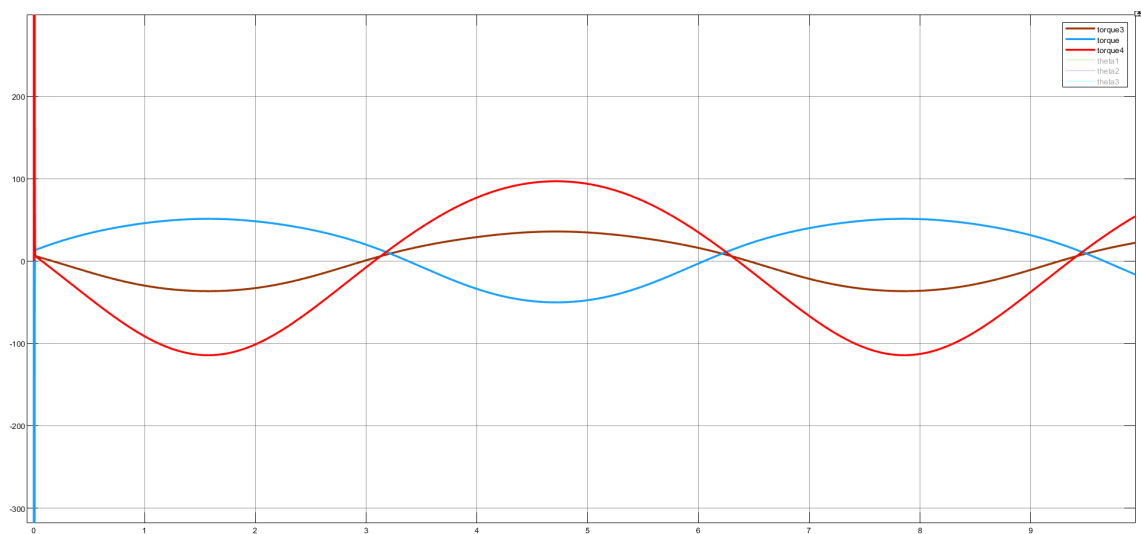
شکل ۳: دیاگرام دینامیک وارون

در این بلوک، مقادیر ورودی به بلوک های revolute به عنوان ورودی های مفاصل  $q$  وارد می کنیم. برای مشاهده ی گشتاور تولید شده در هر مفصل، سنسور  $ActuatorTorque$  را فعال می کنیم.

Revolute Joint		<input checked="" type="checkbox"/> Auto Apply	?
Settings	Description		
NAME	VALUE		
<b>▼ Z Revolute Primitive (Rz)</b>			
▼ State Targets			
<input checked="" type="checkbox"/> Specify Position Target			
Priority	Low (approximate) ▼		
Value	smiData.RevoluteJoint(4).Rz.Pos	deg ▼	Compile-time ▼
<input type="checkbox"/> Specify Velocity Target			
> Internal Mechanics			
> Limits			
> Actuation			
▼ Sensing			
<input type="checkbox"/> Position			
<input type="checkbox"/> Velocity			
<input type="checkbox"/> Acceleration			
<input checked="" type="checkbox"/> Actuator Torque			
<input type="checkbox"/> Lower-Limit Torque			
<input type="checkbox"/> Upper-Limit Torque			
> Mode Configuration			
> Composite Force/Torque Sensing			

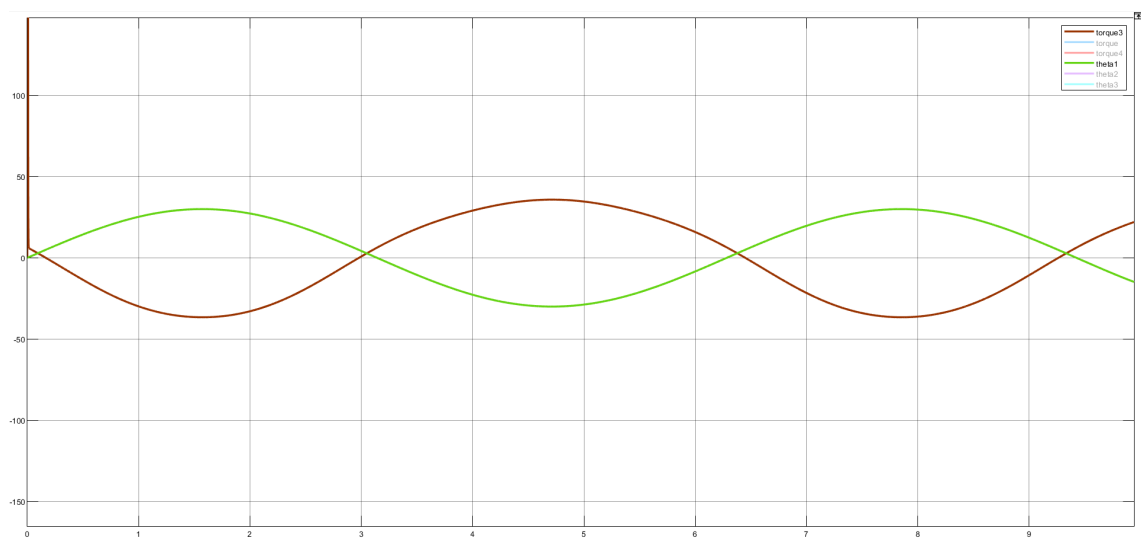
شکل ۴: تنظیمات مفاصل

در پایان این بخش، با اعمال ورودی های مفاصل و بررسی گشتاور تولیدی در هر یک، به بررسی خروجی های به دست آمده برای آنها می پردازیم.

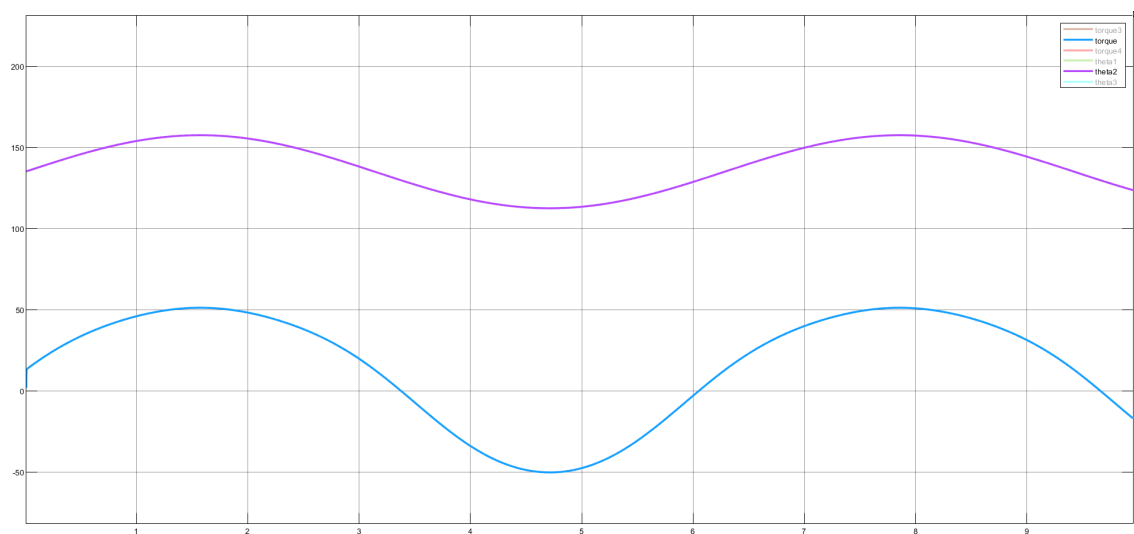


شکل ۵: نمودار گشتاور تولید شده در مفاصل

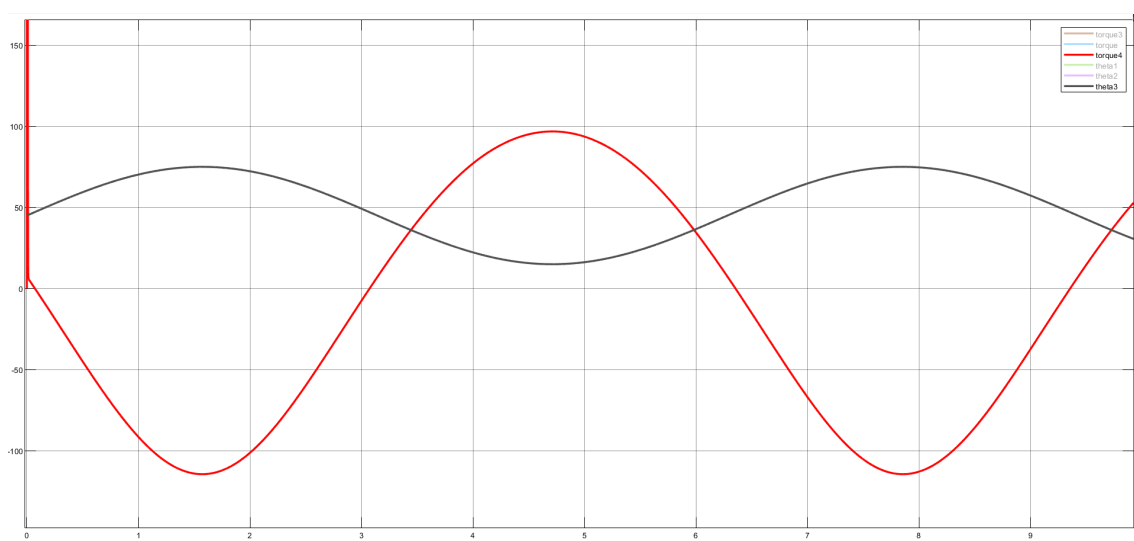
مشاهده می شود که در لحظات ابتدایی از شبیه سازی، اورشوت زیادی به سیستم اعمال می شود. اما رفتار آن پس از زمان راه اندازی، پایدار است و مطابق با مسیر های داده شده، خروجی های سینوسی ایجاد می کند. برای بررسی بیشتر و مقایسه ی گشتاور ایجاد شده در هر مفصل با زاویه ی ورودی به آن، می توانیم این سیگنال ها را رسم کنیم.



شکل ۶:  $\theta$



شکل ۷:  $\theta_2$

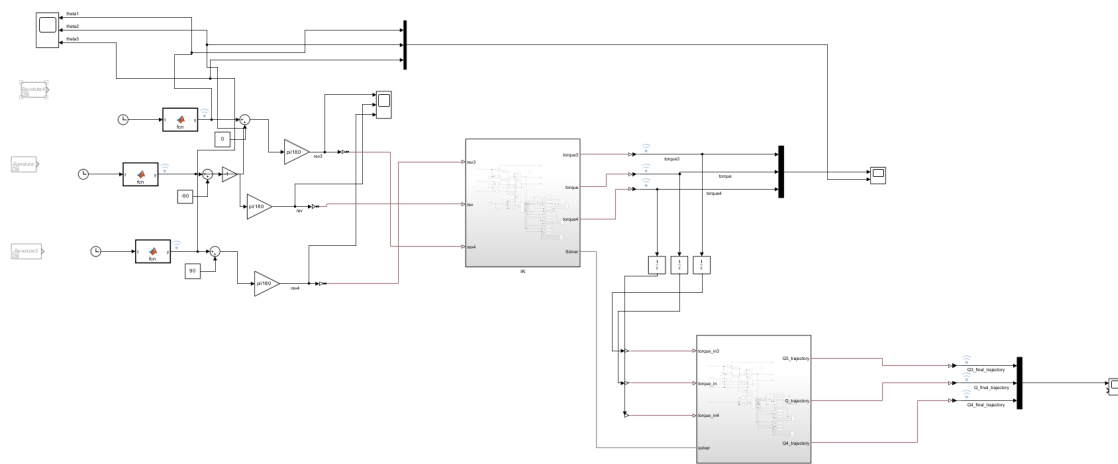


شکل ۸:  $\theta_3$

با در نظر داشتن این نمودارها، می توان نتیجه گرفت که بیشترین گشتاور ایجاد شده در هر مفصل در شرایطی است که سرعت زاویه ای آن  $\dot{q}$  برابر صفر شود و مفصل قصد تغییر جهت حرکت داشته باشد.

پاسخ سوال دو، محاسبه دینامیک مستقیم در شبیه ساز





شکل ۱۰: دیاگرام سیستم با دینامیک مستقیم و وارون

در این سیستم، مقادیر زوایای ورودی در گام اول به بلوک دینامیک وارون شده و با انجام شبیه سازی، گشتاورهای هر مفصل ایجاد می شود. در ادامه، با اعمال این گشتاور ها به بلوک دینامیک مستقیم، مجدداً زوایای هر مفصل ایجاد می شود و می توان انتظار داشت که در صورت صحت سیستم های دینامیک مستقیم، نتایج یکسانی در خروجی به دست بیاید.

## پاسخ سوال سه، طراحی مسیر

برای ایجاد بلوک طراحی مسیر، مطابق روابط ب.۵ و ب.۱۰ کتاب کد تابع متلب نوشته شده و با وارد کردن مقادیر اولیه و نهایی موقعیت، سرعت و شتاب و همچنین زمان شروع و پایان، مسیر مورد نظر برای هر مفصل ایجاد می شود. کد این برنامه در بخش زیر آورده شده است:

```

function [pos3, pos5] = fcn(t, theta0, thetáf, thetad0, thetadf, thetadd0, thetaddf, tstart, tfinal)
% Input parameters:
% t - current time from Simulink clock
% theta0 - initial position
% thetáf - final position
% thetad0 - initial velocity
% thetadf - final velocity
% thetadd0 - initial acceleration (for 5th order only)
% thetaddf - final acceleration (for 5th order only)
% tstart - start time
% tfinal - final time

% Outputs:
% pos3 - position from 3rd order trajectory
% pos5 - position from 5th order trajectory

% Calculate time duration
T = tfinal - tstart;

% 3rd order trajectory coefficients
a0_3 = theta0;
a1_3 = thetad0;
a2_3 = (-3 * (theta0 - thetáf) - (2 * thetad0 + thetadf) * T) / T^2;
a3_3 = (2 * (theta0 - thetáf) + (thetad0 + thetadf) * T) / T^3;

% 5th order trajectory coefficients
a0_5 = theta0;
a1_5 = thetad0;
a2_5 = 0.5 * thetadd0;
a3_5 = (1/(2*T^3)) * (20 * (thetáf - theta0) - (8 * thetadf + 12*thetad0) * T - (3 * thetaddf - thetadd0) * T^2);
a4_5 = (1/(2*T^4)) * (30 * (theta0 - thetáf) + (14 * thetadf + 16*thetad0) * T + (3 * thetaddf - 2*thetadd0) * T^2);
a5_5 = (1/(2*T^5)) * (12 * (thetáf - theta0) - 6*(thetadf + thetad0) * T - (thetaddf - thetadd0) * T^2);

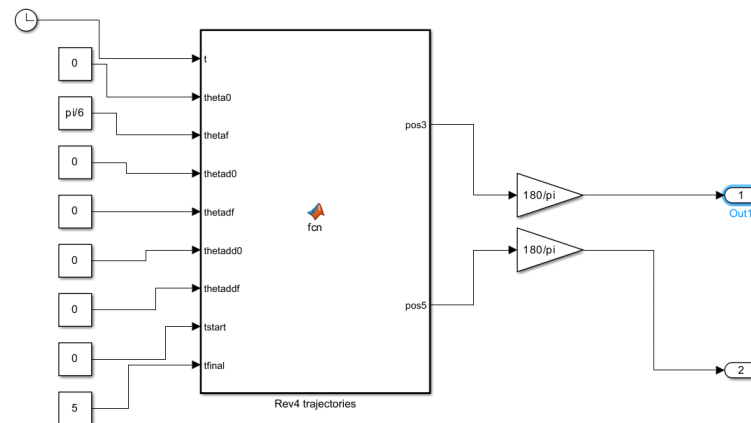
% Calculate relative time
t_rel = t - tstart;

% Ensure trajectory stays at final position after tfinal
if t < tstart
    pos3 = theta0;
    pos5 = theta0;
elseif t > tfinal
    pos3 = thetáf;
    pos5 = thetáf;
else
    % Calculate positions using polynomial equations
    pos3 = a0_3 + a1_3*t_rel + a2_3*t_rel^2 + a3_3*t_rel^3;
    pos5 = a0_5 + a1_5*t_rel + a2_5*t_rel^2 + a3_5*t_rel^3 + a4_5*t_rel^4 + a5_5*t_rel^5;
end
end

```

شکل ۱۱: کد طراحی مسیر

با قرار دادن کد ارائه شده در بلوک *MatlabFunction* و ارائه ی ورودی های مناسب به صورت زیر، می توان مسیر های طراحی شده درجه ۳ و ۵ را به دست آورد. در تصویر زیر، دیاگرام بلوک طراحی شده نمایش داده شده است.



شکل ۱۲: بلوک طراحی مسیر

لازم به ذکر است که مسیر های طراحی شده توسط این بلوک بر حسب رادیان محاسبه می شوند و برای استفاده از آن در ربات، با بلوک های بهره مقادیر به فضای درجه تبدیل شده اند.

## پاسخ سوال چهار، طراحی مسیر برای ربات

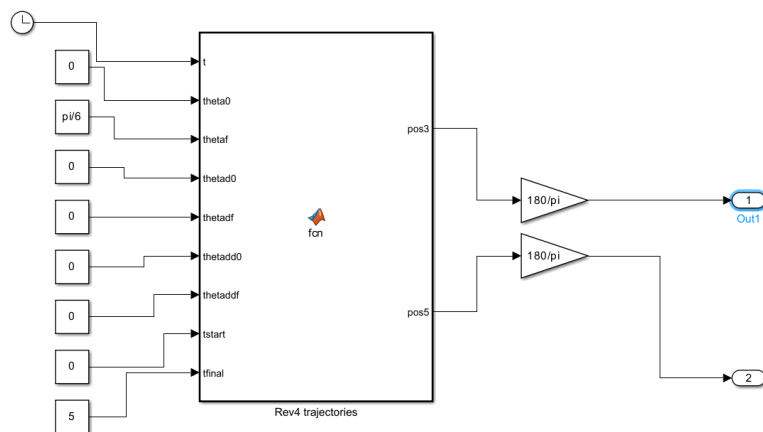
با استفاده از بلوک طراحی مسیر در بخش قبل و اعمال ورودی های مشخص شده در تمرین این بخش به شرح زیر، مسیر های خواسته شده را طراحی می کنیم.

$$t = 0 \text{ s} \rightarrow t = 5 \text{ s}, \quad \mathbf{q}_0 = \begin{bmatrix} 0 \\ 45 \\ 135 \end{bmatrix} \rightarrow \mathbf{q}_f = \begin{bmatrix} 30 \\ 75 \\ 157.5 \end{bmatrix},$$

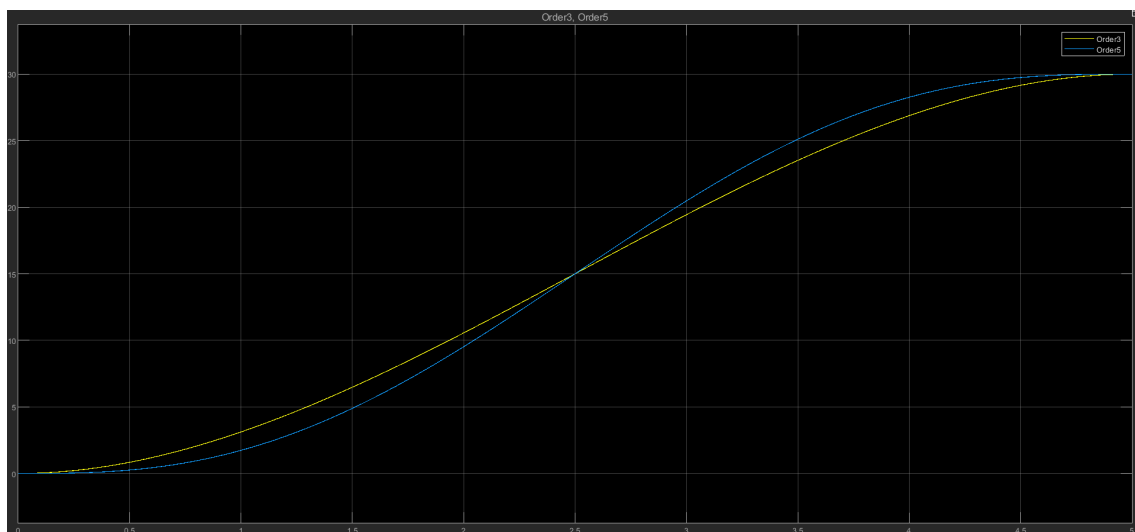
$$\dot{\mathbf{q}}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \rightarrow \dot{\mathbf{q}}_f = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},$$

$$\ddot{\mathbf{q}}_e = \begin{bmatrix} \circ \\ \circ \\ \circ \end{bmatrix} \rightarrow \ddot{\mathbf{q}}_f = \begin{bmatrix} \circ \\ \circ \\ \circ \end{bmatrix}.$$

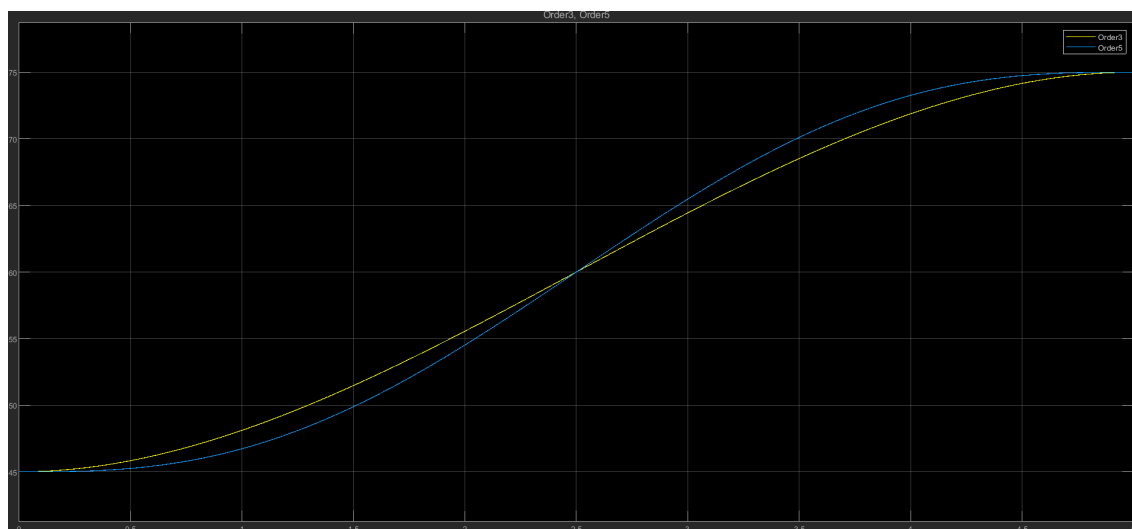
با اعمال ورودی ها به صورت زیر برای بلوک ها خواهیم داشت:



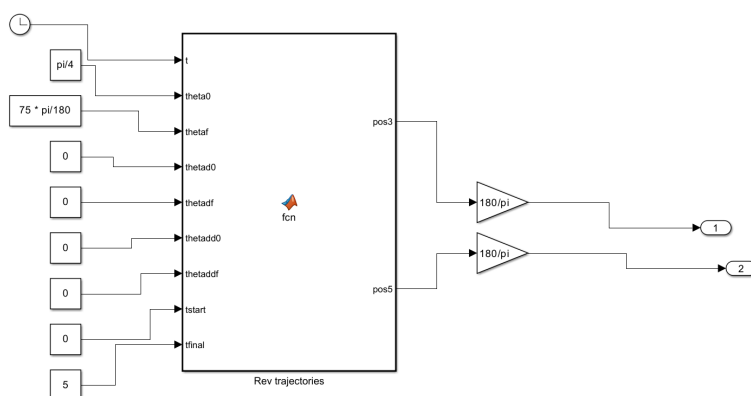
شکل ۱۳: دیاگرام طراحی مسیر مفصل تتا ۴



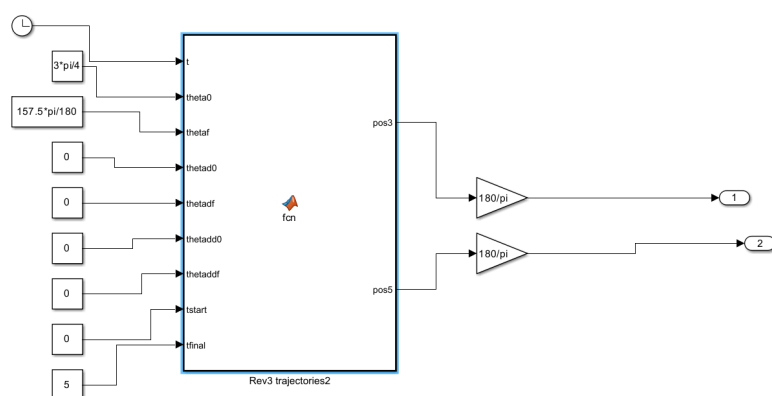
شکل ۱۴: مسیر مرتبه ۳ و ۵ برای مفصل تتا ۴



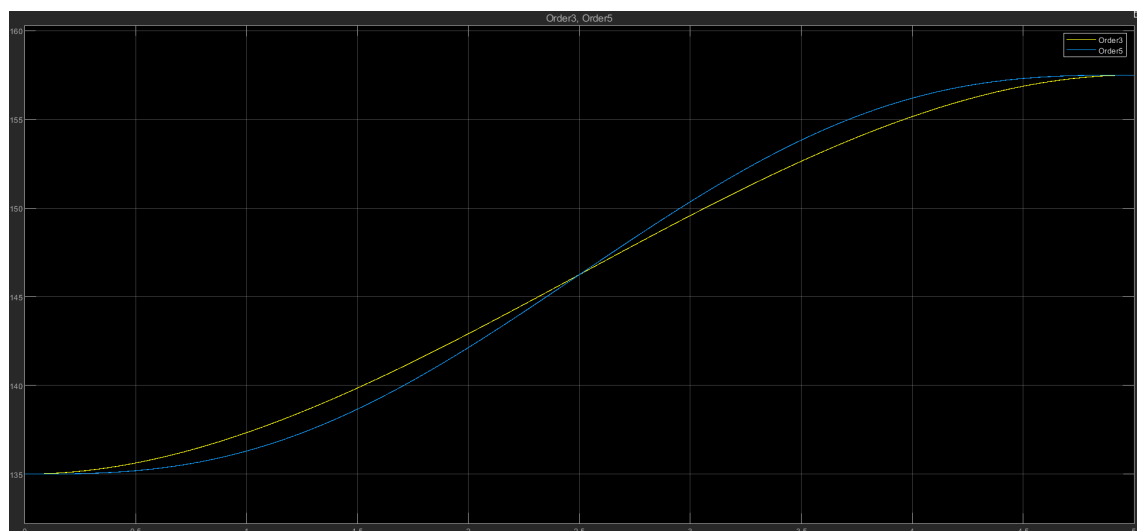
شکل ۱۶: مسیر مرتبه ۳ و ۵ برای مفصل تتا



شکل ۱۵: دیاگرام طراحی مسیر مفصل تتا

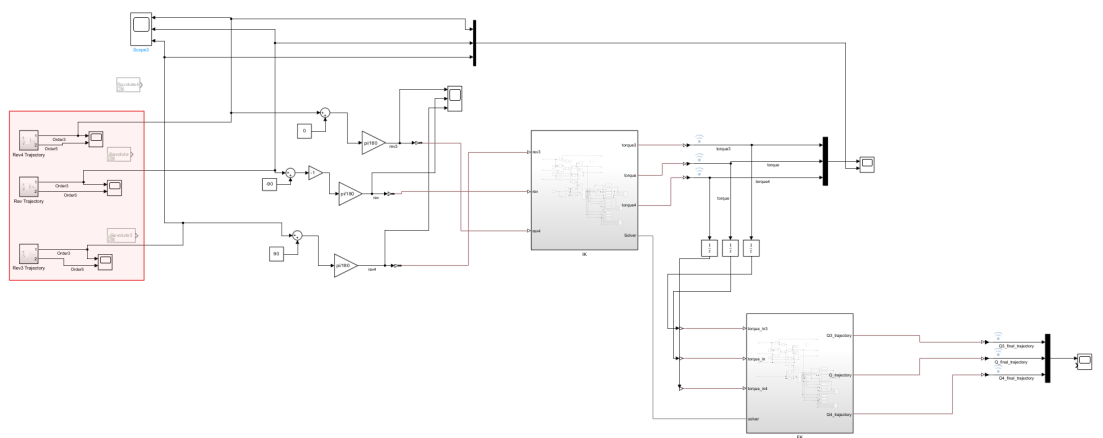


شکل ۱۷: دیاگرام طراحی مسیر مفصل تتا ۳



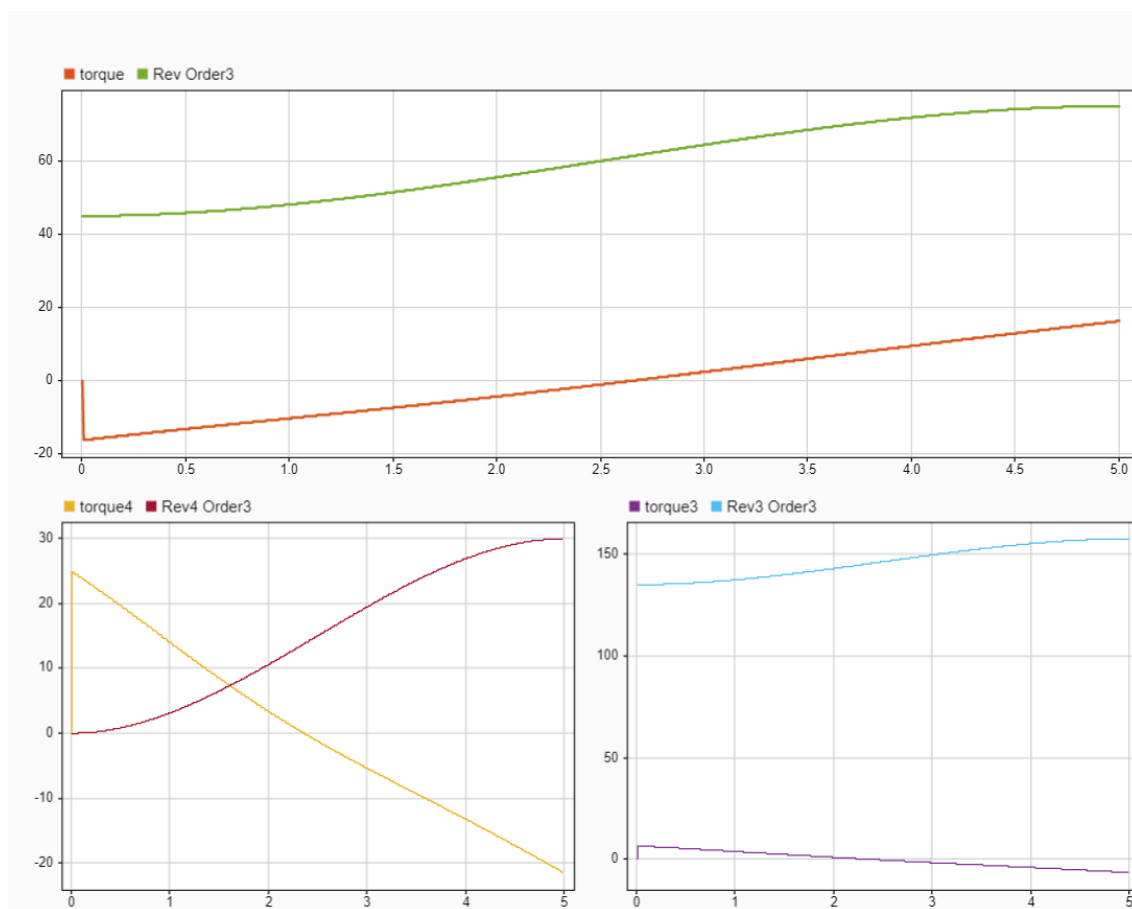
شکل ۱۸: مسیر مرتبه ۳ و ۵ برای مفصل تتا ۳

حال با جایگذاری مسیر مرتبه سوم ایجاد شده توسط این بلوک به عنوان ورودی ربات در شبیه ساز، دیاگرام سیستم به صورت زیر تغییر می کند.



شکل ۱۹: دیاگرام به همراه بلوک طراحی مسیر

با اعمال این ورودی به بلوک دینامیک وارون سیستم و اندازه گیری مقادیر گشتاور تولید شده، می توانیم نحوه عملکرد موتورهای سیستم را در بخش های مختلف مسیر مشاهده کنیم. در نمودار زیر، مسیر حرکتی هر مفصل و گشتاور تولید شده برای آن رسم شده است.



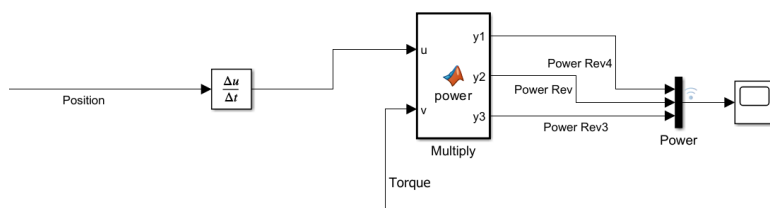
شکل ۲۰: گشتاور مفصل ها و مسیر ورودی به هر مفصل

مشاهده می شود که در نقطه ی عطف مسیر هر مفصل، مقدار گشتاور برابر صفر شده و پس از آن، با علامت عکس عمل می کند تا سرعت و شتاب را تا نقطه ی انتهایی به صفر برساند.

برای محاسبه ی توان هر موتور، لازم است مطابق رابطه ی زیر از حاصل ضرب سرعت موتور در گشتاور آن مطابق رابطه ی زیر استفاده کرد.

$$P = \tau \cdot \omega$$

برای اندازه گیری این مقدار در شبیه ساز، ابتدا با مشتق گیری از مسیر داده شده به سیستم، سرعت آن در هر لحظه محاسبه شده و سپس، با ضرب این مقدار در گشتاور اندازه گیری شده، مقدار توان گزارش می شود. در نتیجه، دیاگرام این قسمت به صورت زیر به دست می آید.



شکل ۲۱: محاسبه توان

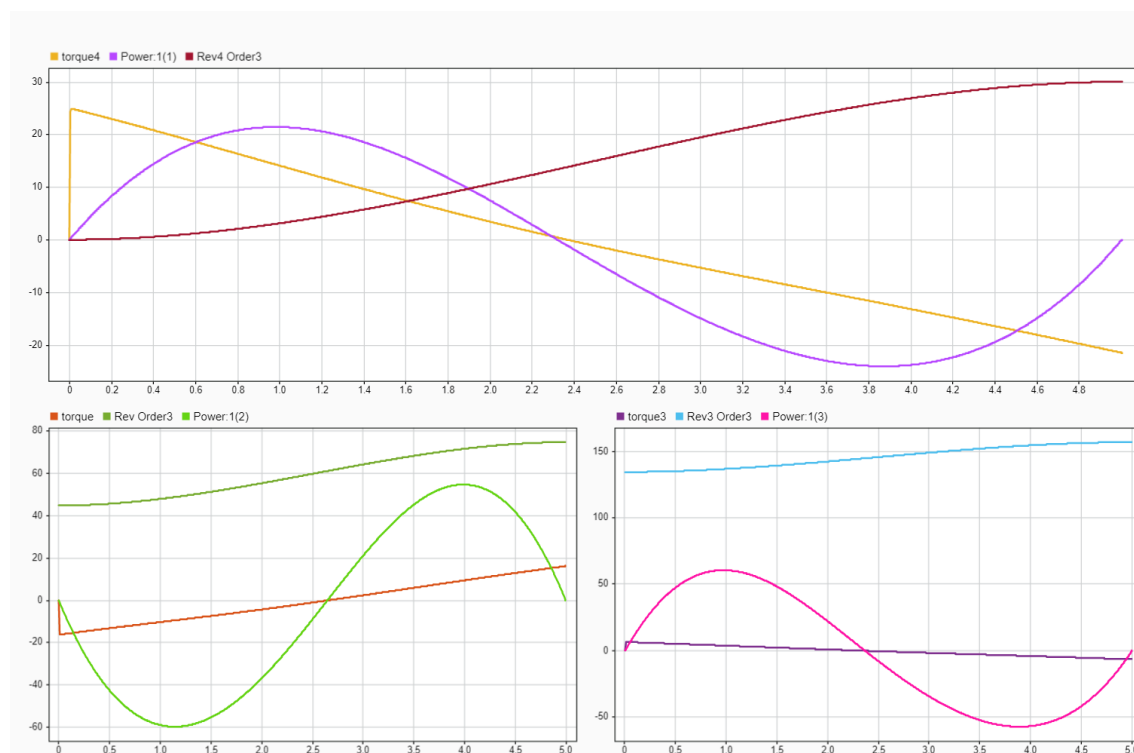
با اجرای شبیه سازی، نمودار های توان موتور ها به صورت زیر به دست می آید.



شکل ۲۲: نمودار توان موتور ها

همچنین، برای مقایسه ی توان و گشتاور ها، می توانیم این دو نمودار را در کنار یکدیگر رسم کنیم.





شکل ۲۳: نمودار توان، گشتاور و مسیر حرکت هر مفصل