دانشگاه صنعتی شریف

دانشکده مهندسی مکانیک

عنوان:

تمرینات سری پنجم

طراحی تراژکتوری

نگارش

محمدسعید صافی زاده

استاد راهنما

دکتر بهزادی پور

آذر ماه ۱۴۰۳

فهرست مطالب

۱ صورت سوالات

برای ربات تمرین ۳ باید طراحی حرکت صورت گیرد. برای این منظور مبدا عملگر نهایی باید از مختصات ابتدایی به مختصات نهایی حرکت کند.

$X_i(mm)$	284.829	$X_f(mm)$	91.203
$Y_i(mm)$	-164.446	$Y_f(mm)$	340.373
$Z_i(mm)$	190.293	$Z_f(mm)$	605.682

جدول ۱: دیتاهای داده شده برای موقعیت ابتدایی وانتهایی ربات end-effector

۱.۱ سوال اول

مسیر حرکت فاقد هر نوع اهمیتی است و فقط کافی است که مفاصل حرکت نرم داشته باشند و حرکت در یک و نیم ثانیه انجام شود. با در نظرگرفتن تراژکتوری چند جمله ای برای مفاصل و شرط پیوستگی سرعت و شتاب در طول حرکت، تراژکتوری مفاصل را طراحی کنید و سپس حرکت طراحی شده را به مدل ربات در محیط سیمولینک اعمال کنید. منحنی حرکت مفاصل و مختصات عملگر را بر حسب زمان رسم کنید.

۲.۱ سوال دوم

با فرض حرکت روی خط راست و حدود مقادیر شتاب و سرعت عملگرنهایی مطابق زیر، تراژکتوری مفاصل را بر اساس پروفیل ذوزنقه ای برای سرعت عملگر نهایی، طراحی نمایید. سپس بکمک بلوک signal builder تراژکتوریهای طراحی شده را به مدل اعمال و منحنی حرکت مفاصل و مختصات عملگر را بر حسب زمان رسم کنید:

$$V_{max} = 20\frac{Cm}{s}, a_{max} = \pm\frac{g}{4}$$

برای ایجاد حرکت از joint actuatorاستفاده نمایید. این بلوک موقعیت، سرعت و شتاب را بعنوان ورودی نیاز دارد. مدل سیمولینک و گزارش کاملی از نتایج، تمام مراحل حل شامل استفاده از سیمولینک و مطالب تئوری را بارگذاری کنید.

۲ پاسخ سوال اول

در ابتدا ذکر این نکته الزامی است که برای بخش اول مدلی با نام PartA و برای بخش دوم(سوال دوم) مدلی با نام PartB در نظر گرفته شده است، تا نحوه اعمال ورودی ها به مفاصل کاملاً مشخص باشد.

۱.۲ الگوریتم حل سوال اول

برای حل این مسئله به دنبال یک جندجمله برای حرکت مفاصل هستیم. با توجه به اینکه مقادیر شتاب و سرعت باید پیوسته باشند به نظر جندجمله ای مرتبه سوم برای ما مناسب باسد چرا که در حین حرکت هم شتاب و هم سرعت پیوسته باقی می مانند. در این چند جمله ای ها ضرایب مجهول هستند. بنابراین برای یافتن ضرایب از مسئله Inverse Kinematics استفاده می کنیم و موقعیت مفاصل را در ابتدا و انتهای حرکت End Effector می یابیم. سپس با این دو شرط و دو شرط دیگر که عبارتند از صفر بودن سرعت در ابتدا و انتهای حرکت این ضرایب را می یابیم و چندجمله ای یافته شده را به مفاصل می دهیم.

۲.۲ حل مسئله Inverse Kinematics

در تمرین سوم به حل کامل Inverse Kinematics این مدل ربات به صورت مفصل پرداختیم. در اینجا از نتایج بدست آمده در آن تمرین برای یافتن مقادیر ابتدایی وانتهایی هر یک از مفاصل در حرکت می پردازیم.

۱.۲.۲ مفصل اول

برای زاویه مفصل اول داریم :

$$\sin(\theta_1) = \frac{Y}{\sqrt{X^2+Y^2}}$$
$$\cos(\theta_1) = \frac{X}{\sqrt{X^2+Y^2}}$$
$$\theta_1 = atan2(\cos(\theta_1), \sin(\theta_1))$$

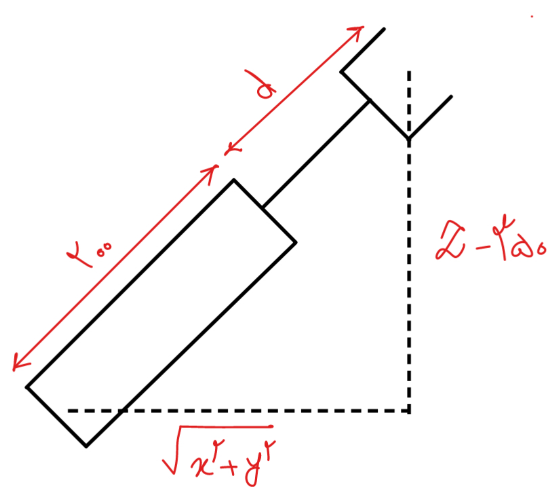
با جایگذاری مقادیر داده شده در جدول ۱ خواهیم داشت:

$$\theta_{1i} = -30\,\text{deg}, \theta_{1f} = 75\,\text{deg}$$

هم چنین دو شرط $\dot{\theta_{1i}} = 0$ و $\dot{\theta_{1f}} = 0$ را نیز داریم.

۲.۲.۲ مفصل دوم

برای مفصل دوم طبق شکل ۱ داریم:



شکل ۱: شکل ربات برای درک راحت تر حل مسئله Inverse Kinematic

$$\theta_2 = atan2(\sqrt{X^2+Y^2}, Z-350)$$

که با جایگذاری مقادیر جدول ۱ برای ابتدا و انتهای حرکت خواهیم داشت:

$$\theta_{2i} = atan2(328.8921, -159.707) = -25.9008\,\text{deg}$$

$$\theta_{2f} = atan2(352.3801, 255.682) = 35.9641\,\text{deg}$$

هم چنین مشابه قسمت قبل دو شرط سرعت ابتدا و انتهای حرکت برابر صفر را داریم.

$$\dot{\theta_{2i}} = 0, \dot{\theta_{2f}} = 0$$

۳.۲.۲ مفصل سوم

برای مفصل سوم نیز مطابق با شکل ۱ داریم:

$$d = \sqrt{X^2 + Y^2 + (Z - 350)^2} - 200$$

با جایگذاری مقادیر داده شده در جدول ۱ خواهیم داشت:

$$\theta_{3i} = 165.6178mm, \theta_{3f} = 235.3677mm$$

۳.۲ یافتن چند جمله ای ها برای مفاصل

چندجمله ای را همانطور که در بالا توضیح داده شد درجه سه و به شکل

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

در نظر می گیریم و با جایگذاری چهار شرطی که در قسمت قبل برای هر مفصل بدست آوردیم، ضرایب را به کمک کد زیر محاسبه می کنیم.

```
% polynomial definition
syms a0 a1 a2 a3 t
theta = a0 + a1 * t + a2 * t^2 + a3 * t^3;

% first joint
eq1 = -30 == subs(theta, t, 0);
eq2 = 75 == subs(theta, t, 1.5);
eq3 = 0 == subs(diff(theta,t), t, 0);
eq4 = 0 == subs(diff(theta,t), t, 1.5);
s1 = solve([eq1,eq2,eq3,eq4]);

%second joint
eq1 = -25.9008 == subs(theta, t, 0);
eq2 = 35.9641 == subs(theta, t, 1.5);
eq3 = 0 == subs(diff(theta,t), t, 0);
eq4 = 0 == subs(diff(theta,t), t, 1.5);
s2 = solve([eq1,eq2,eq3,eq4]);

%third joint
eq1 = 165.6178 == subs(theta, t, 0);
eq2 = 235.3677 == subs(theta, t, 1.5);
eq3 = 0 == subs(diff(theta,t), t, 0);
eq4 = 0 == subs(diff(theta,t), t, 1.5);
s3 = solve([eq1,eq2,eq3,eq4]);
```

که s1,s2 و s3 به ما ضرایب چندجمله ای های مربوط به مفصل اول و دوم و سوم را می دهند. بنابراین چندجمله ای های یافته شده به فرم زیر خواهند بود.

$$\theta_1(t) = -30 + 140t^2 - 62.2222t^3$$

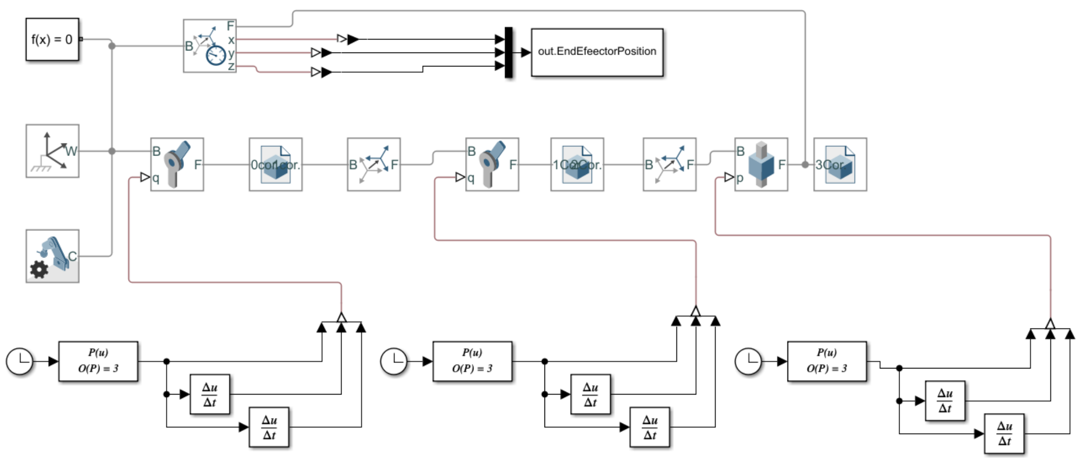
$$\theta_2(t) = -25.9008 + 82.4865t^2 - 36.6607t^3$$

$$\theta_3(t) = 165.6178 + 93t^2 - 41.3333t^3$$

دقت ضرایب تا چهار رقم اعشار محاسبه شده است.

۴.۲ دادن ورودی به مفاصل و شبیه سازی حرکت در سیمولینک

برای ورودی دادن به مفاصل، ابتدا بلوک polynomial را وارود می کنیم و ضرایب چند جمله ای را در آ» وارد می کنیم. (من چندجمله ای مربوط به جا به جایی را وارد کردم). سپس با استفاده از بلوک deravitive در دوشاخه مجزا چندجمله ای مربوط به سرعت و شتاب را محاسبه می کنیم و با استفاده از بلوک PS-Simulink Convertor ورودی خودمان را به مفصل می دهیم. همین کار را برای دومفصل دیگر تکرار می کنیم. شکل بلوک دیاگرام به صورت شکل ۲ در خواهد آمد.



شکل ۲: بلوک دیاگرام پس از اعمال ورودی های مفاصل

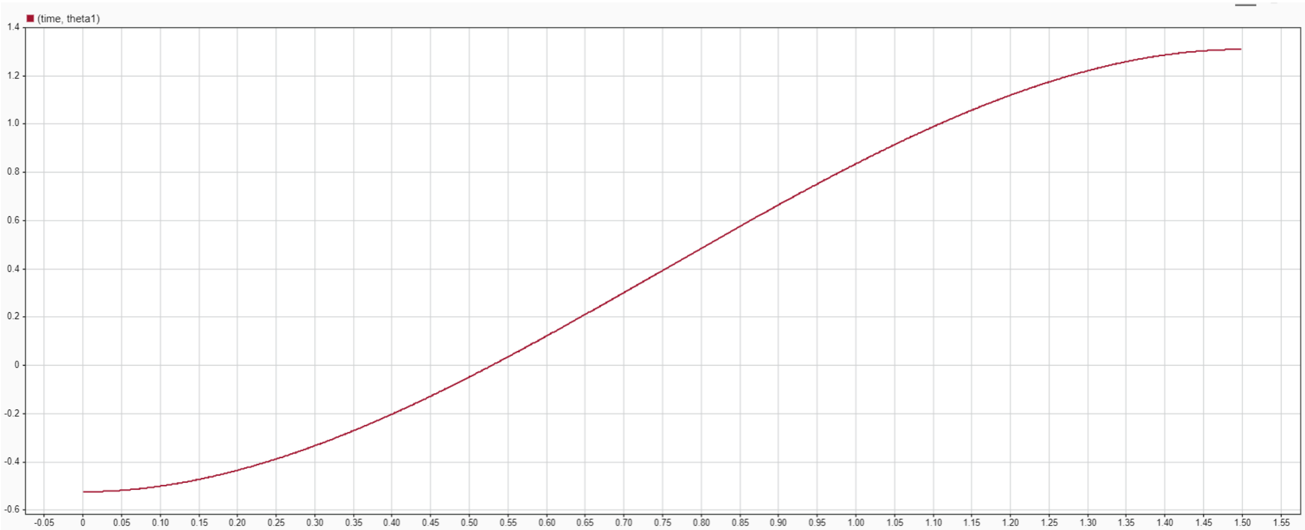
۵.۲ رسم منحنی حرکت مفاصل و عملگر نهایی

برای رسم منحنی مفاصل و عملگر نهایی از بلوک XY Graph استفاده می کنیم. به این صورت که در تنظیمات مفصل از بخش sensing گزینه position را فعال می کنیم و سپس این قسمت را به همراه یک بلوک Clock به بلوک XY Graph می دهیم. برای عملگر نهایی نیز از transform sensor استفاده کردم به این صورت که مولفه های موقعیت عملگر نهایی را نسبت به زمین پیدا می کنم و با تابعی که نوشتم اندازه آن را می یابم و سپس به همراه بلوکClock آن را به بلوک XY Graph برای رسم می دهم. کد تابع متلب به صورت زیر می باشد و منحنی های مفاصل اول، دوم، سوم را می توانید به ترتیب در شکل های ۳، ۴، ۵ و منحنی مولف های عملگر نهایی را در شکل های ۶، ۷ و ۸ ببینید.

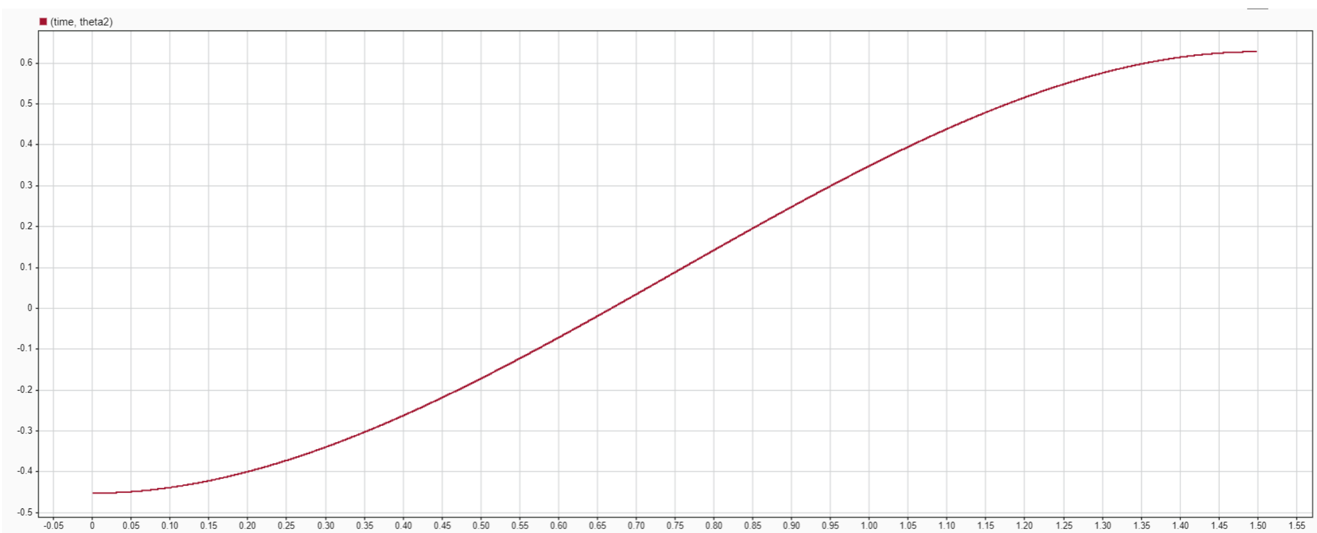
```
function d= fcn(x,y,z)
d = (x^2 + y^2 + z^2)^0.5;
```

هم چنین با استفاده از کد زیر می توان منحنی سه بعدی حرکت عمگر نهایی که در شکل ۹ آمده است را رسم کرد.

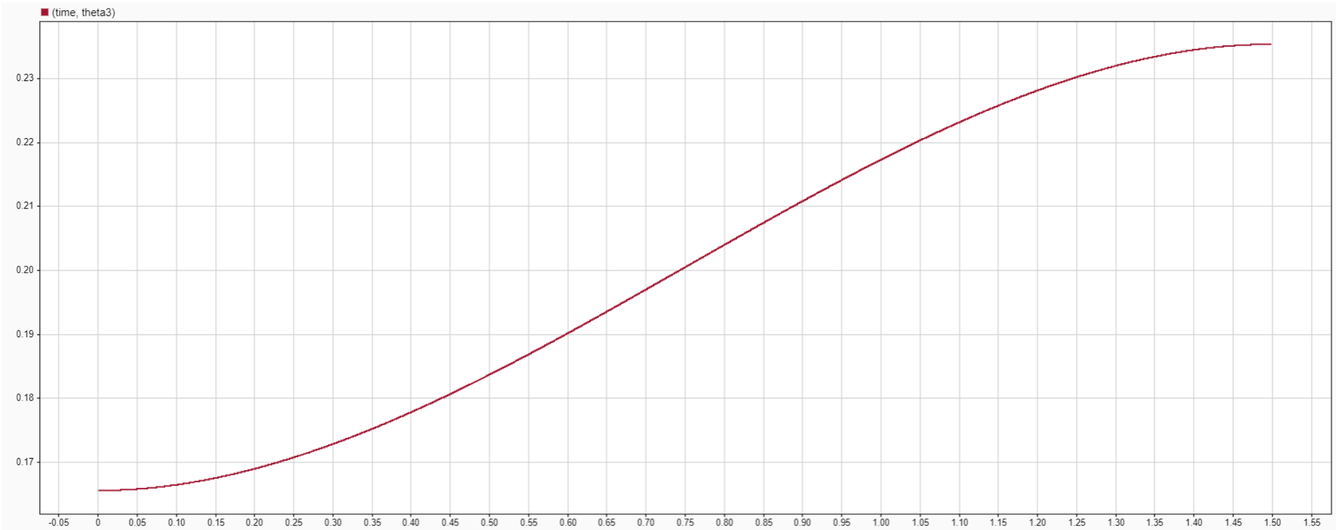
```
plot3(out.EndEfeectorPosition(:,1),out.EndEfeectorPosition(:,2),
... out.EndEfeectorPosition(:,3))
```



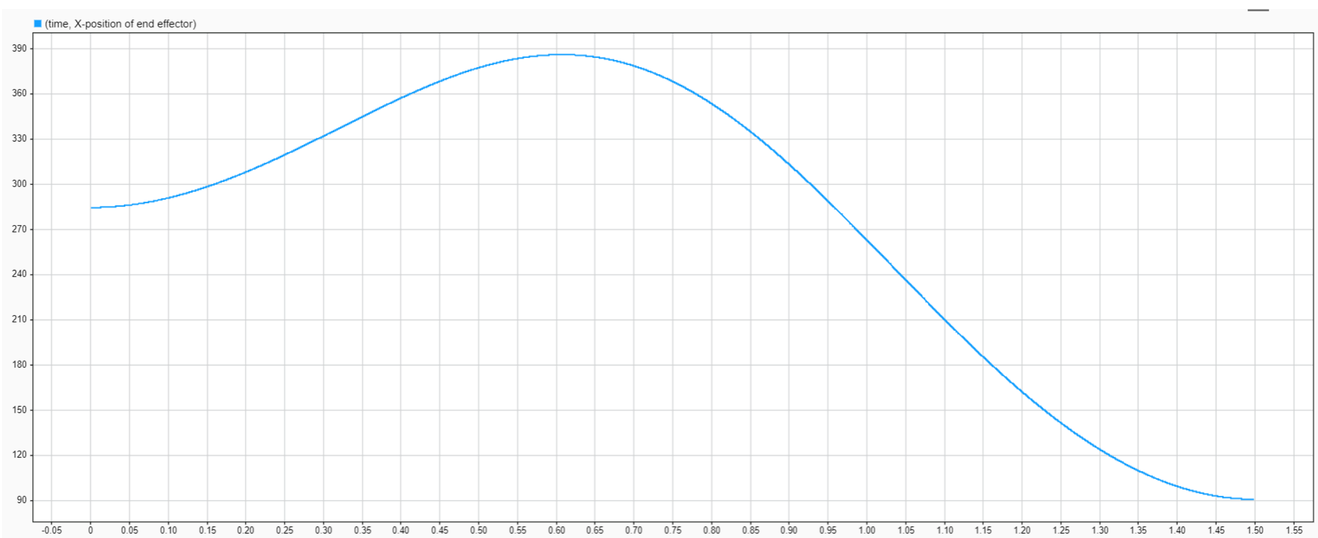
شکل ۳: منحنی حرکت مفصل اول



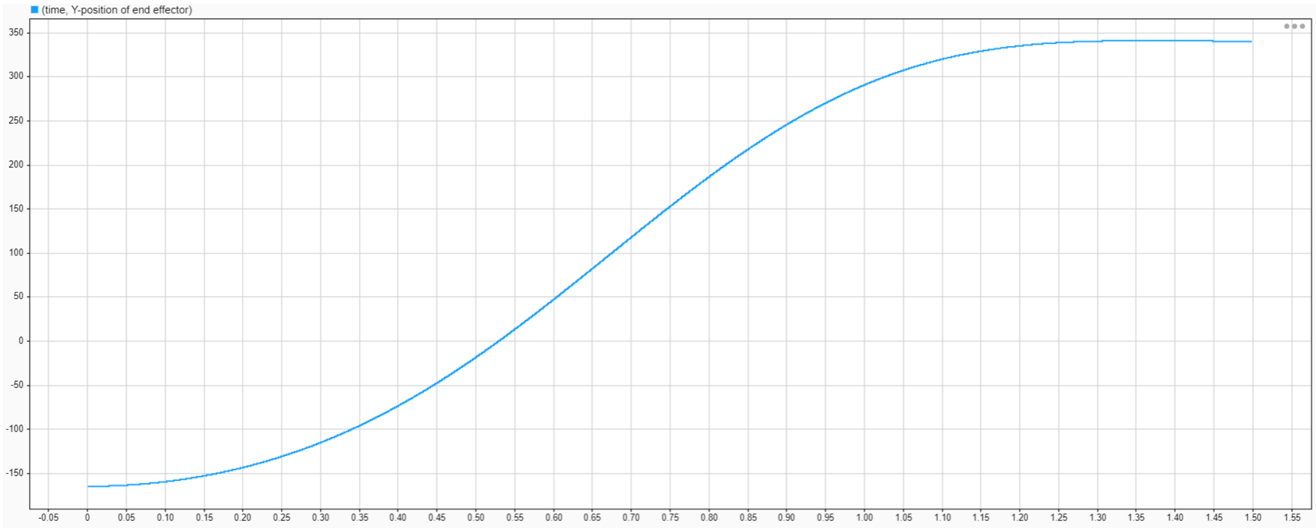
شکل ۴: منحنی حرکت مفصل دوم



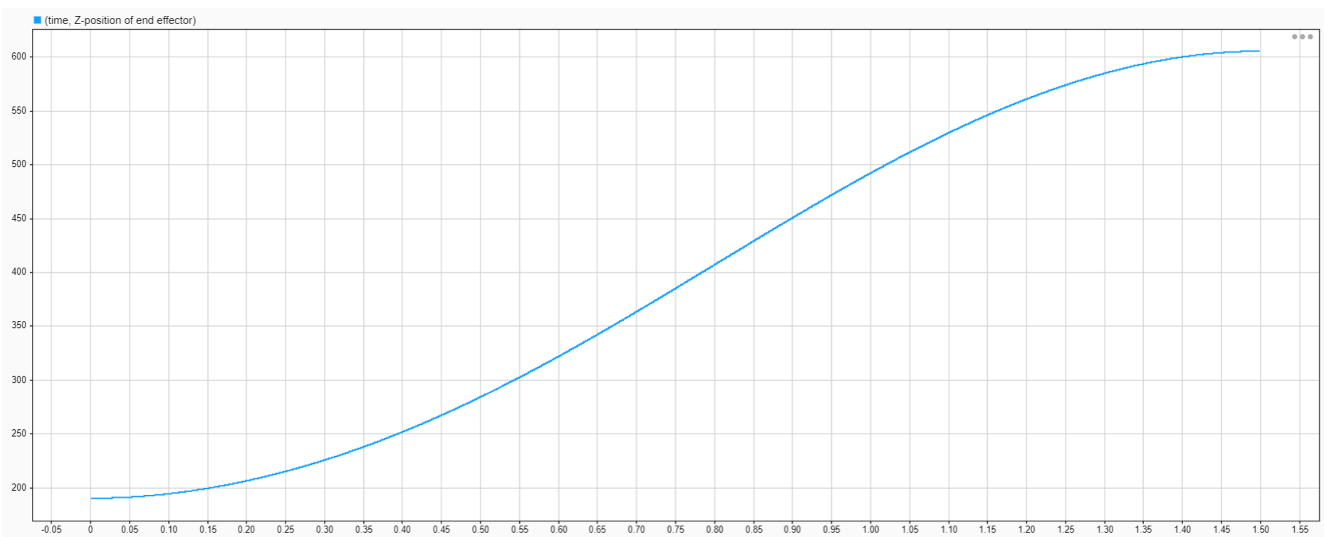
شکل ۵: منحنی حرکت مفصل سوم



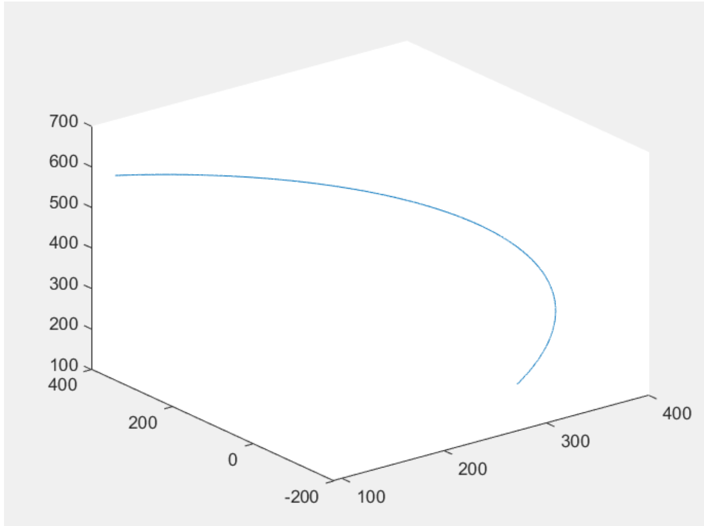
شکل ۶: مولفه x حرکت عملگر نهایی



شکل ۷: مولفه y حرکت عملگر نهایی



شکل ۸: مولفه z حرکت عملگر نهایی



شکل ۹: منحنی سه بعدی حرکت عملگر نهایی

۳ پاسخ سوال دوم

۱.۳ روند و الگوریتم حل

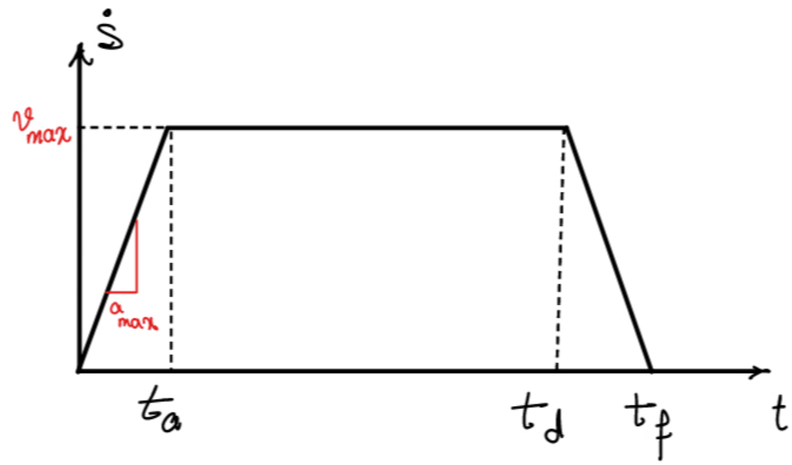
برای این سوال ابتدا با توجه به محدودیت های سرعت و شتابی که برای عملگر نهایی داریم، پروفیل سرعت آن را به صورت ذوزنقه ای رسم می کنیم. سپس از آن انتگرال گرفته تا پرمایش عملگر نهایی S(t) را بدست آوریم نهایتاً با استفاده از فرمول

$$[x(t), y(t), z(t)] = [x1, y1, z1] + \frac{[x2-x1, y2-y1, z2-z1]}{d}.S$$

مولفه های مکان عملگر نهایی را نسبت به زمان به دست می آوریم. سپس این مولفه ها را در مسئله Inverse Kinematics قرار می دهیم و متغیر های مفصلی را بدست می آوریم. در انتها این متغیر های مفصلی را ذخیره و به مدل می دهیم.

۲.۳ طراحی تراژکتوری سرعت عملگر نهایی

برای عملگر نهایی دو محدودیت $V_{max} = 20\frac{cm}{s}$ و $a_{max} = \pm\frac{g}{4}$ را داریم. منحنی سرعت را به صورت شکل ۱۰ در نظر می گیریم. و متغیر های t_a، t_d و t_f را با استفاده از کد زیر می یابیم.

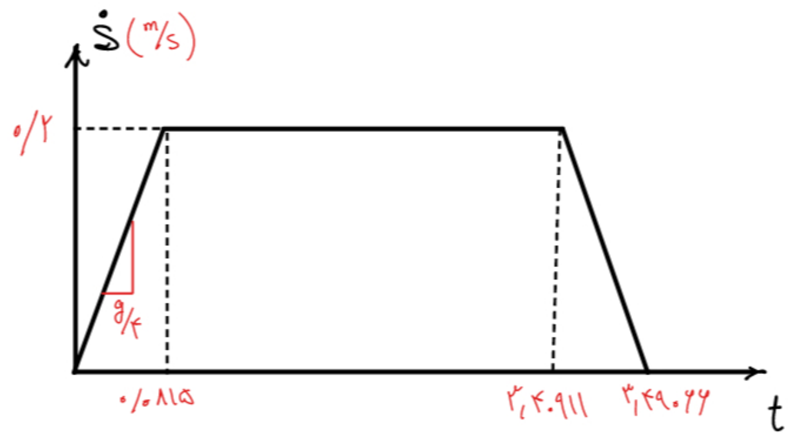


شکل ۱۰: پروفیل ذوزنقه ای مد نظر

```
%finding the Trapezoid LSPB for end effector
% total displacement
Si = [284.829, -164.446, 190.293];
Sf = [91.203, 340.373, 605.682];
d = sqrt((Si(1)-Sf(1))^2 + (Si(2)-Sf(2))^2 + (Si(3)-Sf(3))^2)/ 1000; % in m

% finding trapezoid parameters ta td tf
g = 9.81;
ta = 4 * 0.2 / g;
syms td tf
eqn1 = ((td - ta) + tf) * 0.2 /2 == d;
eqn2 = 0.2 / (tf - td) == g/4;
s = solve([eqn1, eqn2]);
td = vpa(s.td);
tf = vpa(s.tf);
```

در این کد ابتدا مختصات ابتدا و انتهای حرکت داده شده اند و فاصله بین این دو در متغیر d محاسبه شده است. با توجه به شیب خط در قسمت اول نمودار که همان شتاب $\frac{g}{4}$ می باشد، مقدار $t_a = 0.0815s$ بدست می آید. سپس با استفاده از دو معادله دیگر که اولی عبارت است از جا به جایی کلی عملگر که همان d است و دومین معادله عبارت است از محاسبه شیب در قسمت انتهایی نمودار مقادیر $t_d = 3.409s$ و $t_f = 3.4907s$ بدست می آیند. بنابراین پروفیل ذوزنقه ای نهایی به صورت شکل ۱۱ در خواهد آمد.
در این مرحله معادلات هر سه قسمت از شکل ۱۱ را می نویسیم که به صورت معادلات زیر در خواهند آمد:(اندیس ها نمایش دهنده ناحیه هستند.)

$$v_1 = \frac{g}{4}t$$
$$v_2 = 0.2$$
$$v_3 = -\frac{g}{4}t + \frac{0.2t_f}{t_f - t_d}$$



شکل ۱۱: پروفیل طراحی شده برای عملگر نهایی

که با انتگرال گیری از این معادلات مقدار s برای هر ناحیه بدست می آید.

$$s_1 = \frac{g}{8}t^2$$

$$s_2 = 0.2(t - t_a) + s_1(t_a)$$

$$s_3 = -\frac{g}{8}(t - t_d)^2 + 0.2(t - t_d) + s_2(td)$$

کد این قسمت به صورت زیر می باشد:

```
% finding the end effector posion of each point
syms t
% 0 < t < ta
v1 = g/4*t;
% ta < t < td
v2 = 0.2;
% td < t < tf
v3 = -g/4*t + 0.2*tf/(tf -td);
% integerating from above equations
x1 = g/8 * t^2 ;
x2 = 0.2*(t - ta) + subs(x1, t, ta);
x3 = -(g/8) *(t - td)^2 + 0.2*(t - td) + subs(x2, t, td);
```

حالا با استفاده از فرمول پرمایش خط، رابطه مولفه های مکان عملگر نهایی را بر حسب زمان می یابیم:

$$[x(t), y(t), z(t)] = [x1, y1, z1] + \frac{[x2 - x1, y2 - y1, z2 - z1]}{d}.S$$

با جایگذاری مقادیر داریم :

$$[x(t), y(t), z(t)] = [284.829, -164.446, 190.293] +$$

$$\frac{[91.203 - 284.829, 340.373 - (-164.446), 605.682 - 190.293]}{d}.S$$

که مقادیر $x(t)$، $y(t)$ و $z(t)$ به دست می آیند. کد این قسمت به صورت زیر می باشد. (توجه شود که عدد یک در x1t در کد به معنای ناحیه ابتدایی پروفیل می باشد.)

```
x1t = Si(1) + (Sf(1)-Si(1))*x1/d; %in mm
x2t = Si(1) + (Sf(1)-Si(1))*x2/d;
x3t = Si(1) + (Sf(1)-Si(1))*x3/d;

y1t = Si(2) + (Sf(2)-Si(2))*x1/d;
y2t = Si(2) + (Sf(2)-Si(2))*x2/d;
y3t = Si(2) + (Sf(2)-Si(2))*x3/d;

z1t = Si(3) + (Sf(3)-Si(3))*x1/d;
z2t = Si(3) + (Sf(3)-Si(3))*x2/d;
z3t = Si(3) + (Sf(3)-Si(3))*x3/d;
```

۳.۳ حل مسئله Inverse Kinematic

در این مرحله با استفاده از مقادیر مولفه های عملگر نهایی بر حسب زمان که در بخش قبل بدست آورده ایم به حل مسئله معکوس سینماتیک می پردازیم تا متغیر های مفصلی را بیابیم. به این صورت که با استفاده از یک حلقه، زمان را با گام ها یک صدم ثانیه در روابط مولفه های عملگر نهایی قرار می دهیم و اعداد آن را به دست می آوریم و سپس با قرار دادن این اعداد، در تابع Inverse Kinematic، مقادیر متغیر های مفصلی را بدست می آوریم. کد این روند را می توانید در ادامه مشاهده کنید:

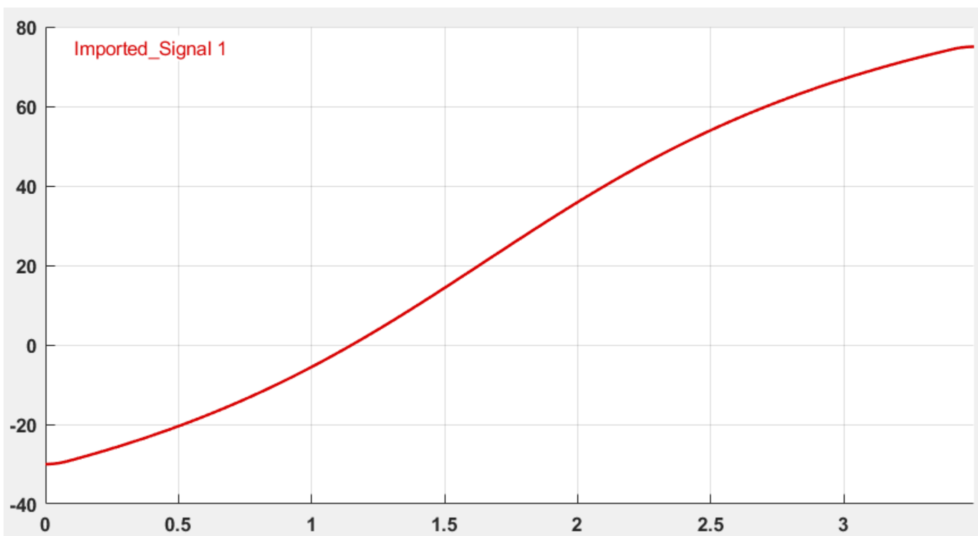
```
theta1 = [];
theta2 = [];
d = []; % in mm
time = [];

for i=0:0.01:ta
[theta1(end+1), theta2(end+1), d(end+1)] = inverse_kinematic(subs(x1t, t, i),
... subs(y1t, t, i), subs(z1t, t, i));
time(end+1) = i;
end

for i=ta:0.01:td
[theta1(end+1), theta2(end+1), d(end+1)] = inverse_kinematic(subs(x2t, t, i),
... subs(y2t, t, i), subs(z2t, t, i));
time(end+1) = i;
end

for i=td:0.01:tf
[theta1(end+1), theta2(end+1), d(end+1)] = inverse_kinematic(subs(x3t, t, i),
... subs(y3t, t, i), subs(z3t, t, i));
time(end+1) = i;
end
```

اکنون مقادیر متغیر های مفصلی یعنی لیست theta1, theta2 و theta3 را به همراه زمان در یک فایل اکسل ذخیره می کنیم تا نهایتاً در محیط مدلسازی، این فایل را به بلوک signal builder، import کنیم.
کد ذخیره سازی متغیر های مفصلی به همراه زمان به صورت زیر می باشد:

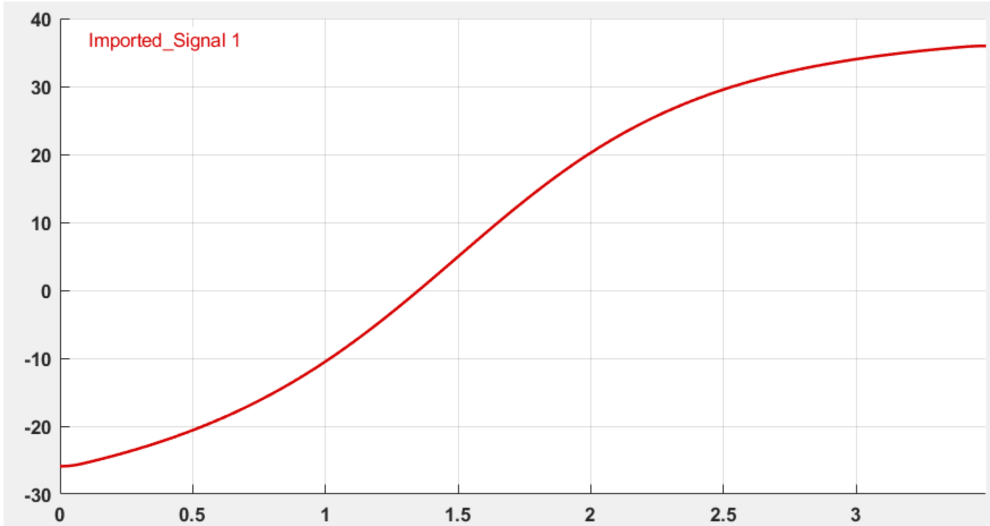
```
theta1_data = [time.',theta1.'];
xlswrite('theta1.xlsx',theta1_data);

theta2_data = [time.',theta2.'];
xlswrite('theta2.xlsx',theta2_data);

theta3_data = [time.',d.'];
xlswrite('theta3.xlsx',theta3_data);
```

۴.۳ اعمال متغیر های مفصلی به مدل

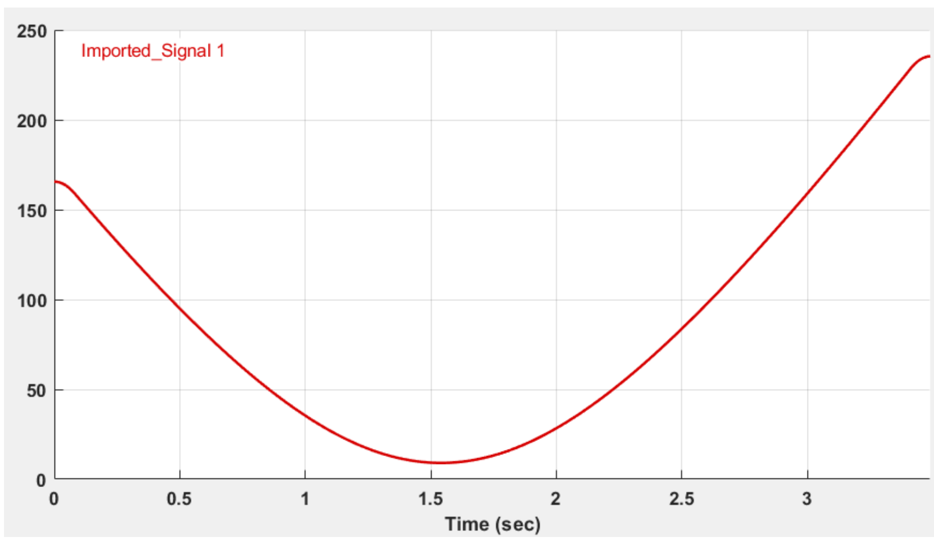
در این بخش، در محیط سیمولینک، بلوک signal builder را وارد می کنیم. در منوی فایل گزینه import from file را برمی گزینیم و برای هر مفصل فایل اکسل مربوطه را وارد می کنیم. این فایل شامل موقعیت متغیر های مفصلی بر حسب زمان می باشد که یکی از خواسته های مسئله است. بنابراین نمودار متغیر های مفصلی بر حسب زمان به ترتیب برای مفاصل اول، دوم و سوم مطابق با شکل های ۱۲، ۱۳ و ۱۴ خواهند شد.



شکل ۱۲: منحنی حرکت مفصل اول بر حسب زمان



شکل ۱۳: منحنی حرکت مفصل دوم بر حسب زمان



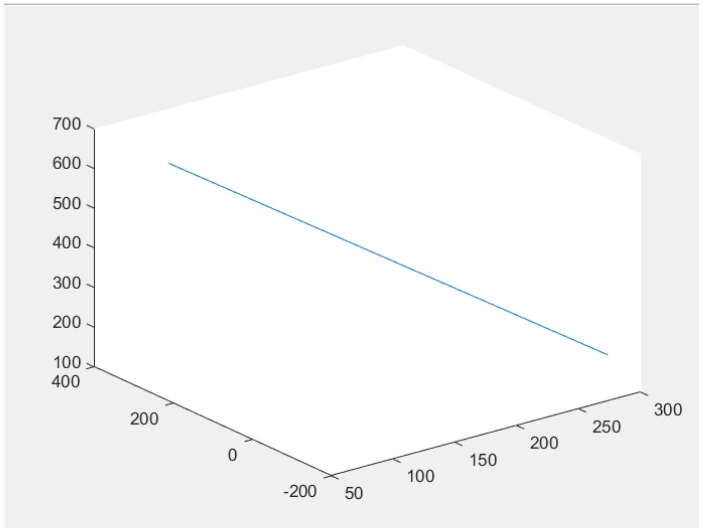
شکل ۱۴: منحنی حرکت مفصل سوم بر حسب زمان

پس از وارد کردن سیگنال به بلوک signal builder با استفاده از یک بلوک Simulink 2 Ps convertor این سیگنال را به همراه مشتق اول و مشتق دوم آن به مفاصل می دهیم. انیمیشن کارکرد این ربات در پوشه Animation با نام AnimationB ضمیمه شده است.

۵.۳ رسم منحنی سه بعدی عملگر نهایی

مشابه با بخش اول سوال، می توانیم با استفاده از کد زیر منحنی سه بعدی حرکت عملگر نهایی را رسم کنیم. همانطور که توقع داشتیم یه خط راست شده است. این منحنی را می توانید در شکل ۱۵ مشاهده کنید.

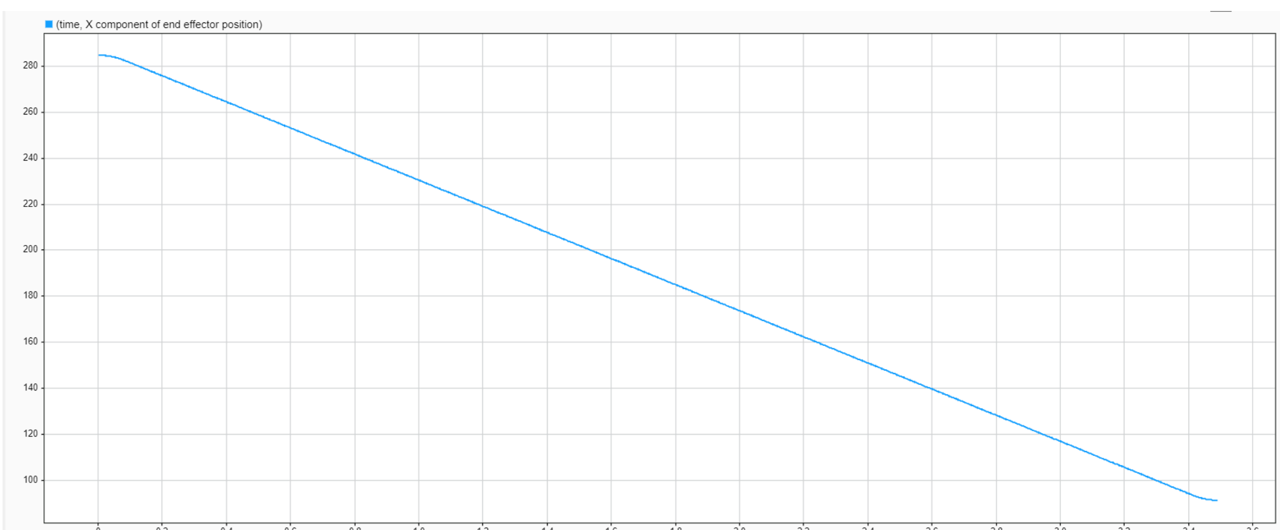
```
>> plot3(out.EndEfeectorPosition(:,1),out.EndEfeectorPosition(:,2),
...out.EndEfeectorPosition(:,3))
```



شکل ۱۵: منحنی سه بعدی عملگر نهایی

۶.۳ صحه سنجی

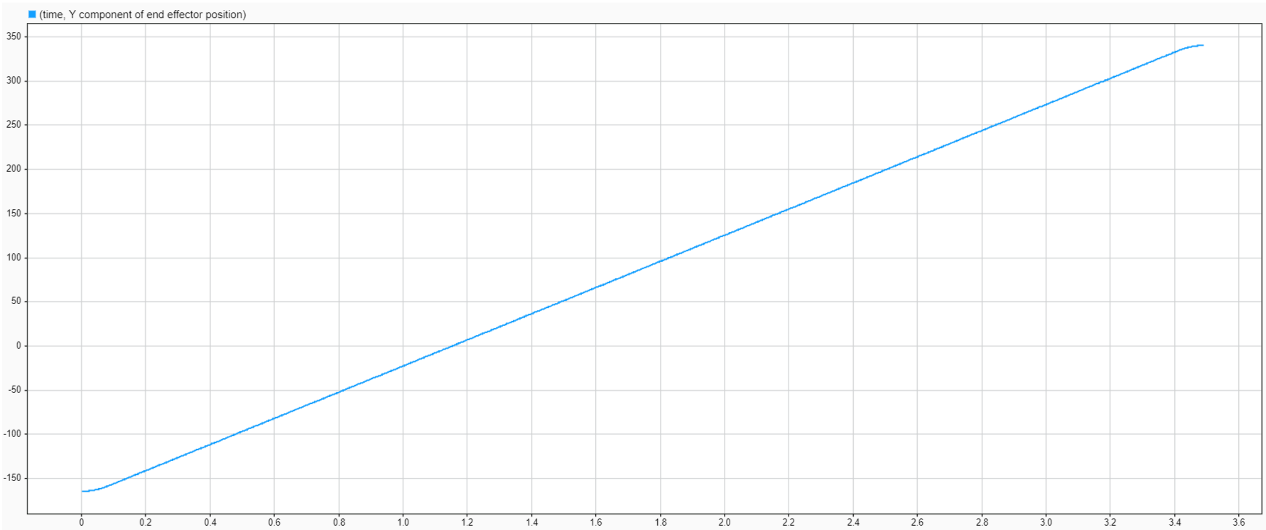
برای اطمینان از صحت جواب ها و قرار گرفتن در نقاط دقیق شروع و پایان با استفاده از بلوک transform sensor موقعیت نهایی عملگر نهایی را به دست آوردم که در محیط workspace و در متغیر EndEffectorPosion ذخیره شده است. با کمک همین بلوک می توانیم موقعیت عملگر نهایی را بر حسب زمان نیز رسم کنیم که مولفه های راستای x، y و z موقعیت عملگر نهایی به ترتیب مطابق شکل های ۱۶ ، ۱۷ و ۱۸ خواهند شد.



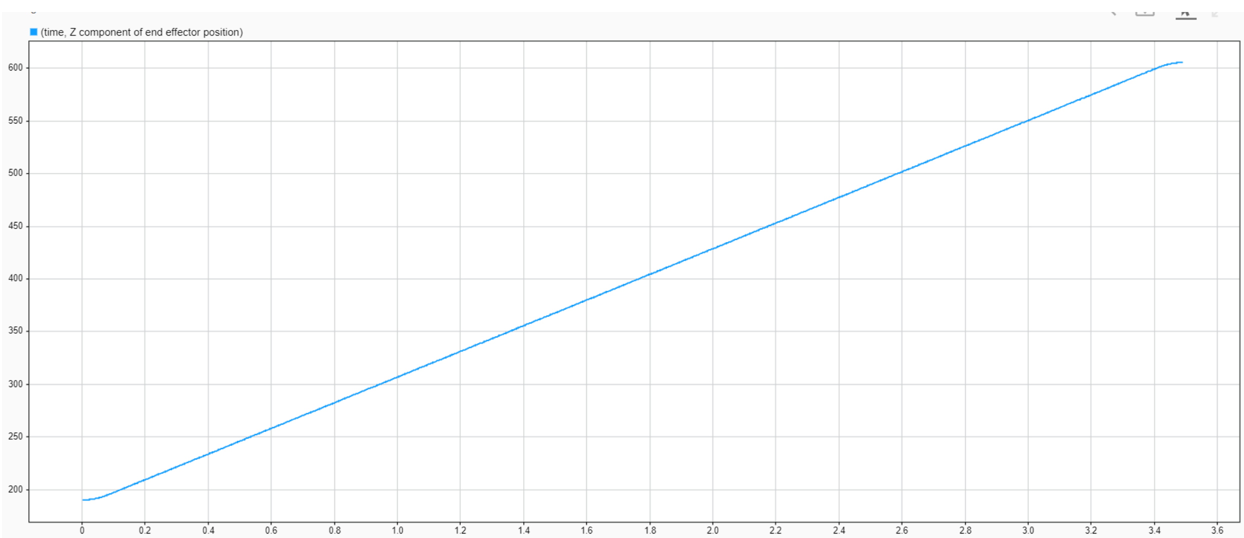
شکل ۱۶: منحنی حرکت مولفه راستای x عملگر نهایی بر حسب زمان

هم چنین منحنی اندازه حرکت عملگر نهایی را، به کمک یه بلوک matlab function که کد آن به صورت زیر است رسم کردم و می توانید آن را در شکل ۱۹ مشاهده نمایید.

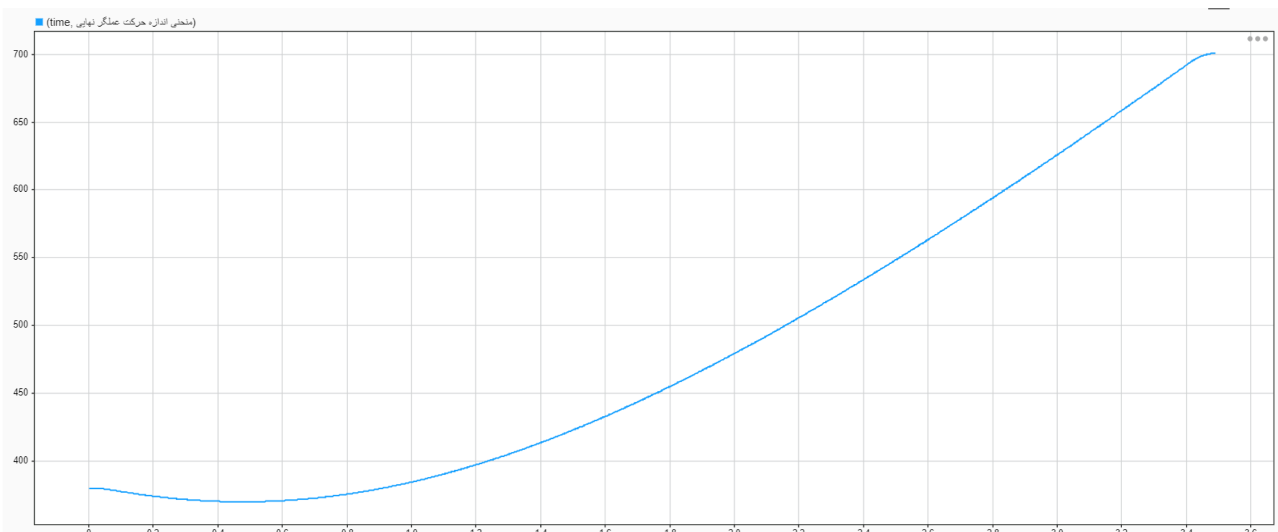
```
function d= fcn(x,y,z)
d = (x^2 + y^2 + z^2)^0.5;
```



شکل ۱۷: منحنی حرکت مولفه راستای y عملگر نهایی بر حسب زمان



شکل ۱۸: منحنی حرکت مولفه راستای z عملگر نهایی بر حسب زمان



شکل ۱۹: منحنی اندازه حرکت عملگر نهایی بر حسب زمان