



دانشگاه خواجه نصیرالدین طوسی
دانشکده برق - گروه رباتیک و الکترونیک

گزارش رباتیک دوره کارشناسی ارشد

رشته مهندسی مکاترونیک

عنوان

گزارش بخش دوم پروژه رباتیک

نگارش

۴۰۲۰۲۴۱۴ علیرضا امیری ایمان گندمی ۴۰۲۲۳۷۰۴

آذر ۱۴۰۳

پاسخ پروژه بخش دوم

پاسخ سوال یک، محاسبه ی ماتریس ژاکوبین

در بخش اول از این پروژه، با استفاده از روش های DH و پیچ، سینماتیک مستقیم ربات به دست آمد. در این سینماتیک مستقیم، ماتریس دوران در نقطه ی ابزار ربات نسبت به پایه ربات، به صورت زیر محاسبه شده است:

$$R = \begin{pmatrix} -\cos(\theta_1) \cos(\theta_2) & \cos(\theta_1) \sin(\theta_2) & \sin(\theta_1) \\ -\sin(\theta_2) & -\cos(\theta_2) & 0 \\ \cos(\theta_2) \sin(\theta_1) & -\sin(\theta_1) \sin(\theta_2) & \cos(\theta_1) \end{pmatrix}$$

المان های زاویه ای ماتریس ژاکوبی

محاسبه ی سرعت های زاویه ای ابزار این ربات، بر اساس $110 - 2$ انجام می شود. برای این کار، در ابتدا مشتق ماتریس دوران نسبت به زمان محاسبه می شود.

$$\dot{R} = \begin{pmatrix} \cos(\theta_2(t)) \sin(\theta_1(t)) \frac{\partial}{\partial t} \theta_1(t) + \cos(\theta_1(t)) \sin(\theta_2(t)) \frac{\partial}{\partial t} \theta_2(t) & \cos(\theta_1(t)) \cos(\theta_2(t)) \frac{\partial}{\partial t} \theta_2(t) - \sin(\theta_1(t)) \sin(\theta_2(t)) \frac{\partial}{\partial t} \theta_1(t) & \cos(\theta_1(t)) \frac{\partial}{\partial t} \theta_1(t) \\ -\cos(\theta_2(t)) \frac{\partial}{\partial t} \theta_2(t) & \sin(\theta_2(t)) \frac{\partial}{\partial t} \theta_2(t) & 0 \\ \cos(\theta_1(t)) \cos(\theta_2(t)) \frac{\partial}{\partial t} \theta_1(t) - \sin(\theta_1(t)) \sin(\theta_2(t)) \frac{\partial}{\partial t} \theta_2(t) & -\cos(\theta_1(t)) \sin(\theta_2(t)) \frac{\partial}{\partial t} \theta_1(t) - \cos(\theta_2(t)) \sin(\theta_1(t)) \frac{\partial}{\partial t} \theta_2(t) & -\sin(\theta_1(t)) \frac{\partial}{\partial t} \theta_1(t) \end{pmatrix}$$

برای محاسبه ی ماتریس ژاکوبین، باید ماتریس ω^\times را محاسبه کنیم.

$$\omega^\times = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix}$$

محاسبه ی این ماتریس به صورت زیر انجام می شود.

$$\omega^\times = \dot{\mathbf{R}}\mathbf{R}^\top$$

در نتیجه، با در اختیار داشتن ماتریس دوران و مشتق آن، میتوانیم ماتریس ω^\times را محاسبه کنیم. برای ربات مورد بررسی در این پروژه خواهیم داشت:

$$\omega^\times = \begin{pmatrix} 0 & -\cos(\theta_1(t)) \frac{\partial}{\partial t} \theta_2(t) & \frac{\partial}{\partial t} \theta_1(t) \\ \cos(\theta_1(t)) \frac{\partial}{\partial t} \theta_2(t) & 0 & -\sin(\theta_1(t)) \frac{\partial}{\partial t} \theta_2(t) \\ -\frac{\partial}{\partial t} \theta_1(t) & \sin(\theta_1(t)) \frac{\partial}{\partial t} \theta_2(t) & 0 \end{pmatrix}$$

با توجه به ساختار ماتریس پادمتقارن، می توان المان های ماتریس ژاکوبین را به دست آورد. در ادامه، با جداسازی $\dot{\theta}$ از المان های ω^\times ، آنچه که باقی می ماند بخش زاویه ای ماتریس ژاکوبین خواهد بود. بنابراین، المان های ماتریس ω^\times ، به صورت زیر تعریف می شوند.

$$\omega_x = \sin(\theta_1(t)) \frac{\partial}{\partial t} \theta_2(t)$$

$$\omega_y = \frac{\partial}{\partial t} \theta_1(t)$$

$$\omega_z = \cos(\theta_1(t)) \frac{\partial}{\partial t} \theta_2(t)$$

$$\vec{\omega} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \sin(\theta_1(t)) \frac{\partial}{\partial t} \theta_2(t) \\ \frac{\partial}{\partial t} \theta_1(t) \\ \cos(\theta_1(t)) \frac{\partial}{\partial t} \theta_2(t) \end{bmatrix} = \begin{bmatrix} 0 & \sin(\theta_1) & 0 \\ 1 & 0 & 0 \\ 0 & \cos(\theta_1) & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}$$

در نتیجه ماتریس ژاکوبین در بخش زاویه ای برای این ربات برابر خواهند بود با:

$$J_w = \begin{bmatrix} 0 & \sin(\theta_1) & 0 \\ 1 & 0 & 0 \\ 0 & \cos(\theta_1) & 0 \end{bmatrix}$$

و از آنجا، سرعت های زاویه ای به شکل زیر به دست می آیند.

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \begin{pmatrix} \sin(\theta_1(t)) \frac{\partial \theta_2(t)}{\partial t} \\ \frac{\partial \theta_1(t)}{\partial t} \\ \cos(\theta_1(t)) \frac{\partial \theta_2(t)}{\partial t} \end{pmatrix}$$

المان های خطی ماتریس ژاکوبی

محاسبه ی المان های خطی ماتریس ژاکوبی، به راحتی با محاسبه ی مشتقات جزئی از ماتریس انتقال نسبت به متغیر های فضای مفصلی انجام می شود.

برای این ربات، خواهیم داشت:

$$p = \begin{pmatrix} \cos(\theta_1) (l_1 \cos(\theta_2) - l_5 \cos(\theta_2)) \\ l_1 \sin(\theta_2) - l_5 \sin(\theta_2) \\ -\sin(\theta_1) (l_1 \cos(\theta_2) - l_5 \cos(\theta_2)) \end{pmatrix}$$

حال، با محاسبه ی مشتق های جزئی این ماتریس، خواهیم داشت:

$$J_v = \begin{pmatrix} -\sin(\theta_1) \sigma_1 & l_5 \cos(\theta_1) \sin(\theta_2) & -l_1 \cos(\theta_1) \sin(\theta_2) \\ \sigma_1 & -l_5 \cos(\theta_2) & l_1 \cos(\theta_2) \\ -\cos(\theta_1) \sigma_1 & -l_5 \sin(\theta_1) \sin(\theta_2) & l_1 \sin(\theta_1) \sin(\theta_2) \end{pmatrix}$$

where

$$\sigma_1 = l_1 \cos(\theta_2) - l_5 \cos(\theta_2)$$

در نتیجه، سرعت های خطی این ربات به صورت زیر به دست می آیند.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{pmatrix} l_5 \cos(\theta_1) \sin(\theta_2) \frac{\partial \theta_1(t)}{\partial t} - l_1 \cos(\theta_1) \sin(\theta_2) \frac{\partial \theta_2(t)}{\partial t} - \sin(\theta_1) \frac{\partial \theta_1(t)}{\partial t} \sigma_1 \\ l_1 \cos(\theta_2) \frac{\partial \theta_2(t)}{\partial t} - l_5 \cos(\theta_2) \frac{\partial \theta_1(t)}{\partial t} \\ l_1 \sin(\theta_1) \sin(\theta_2) \frac{\partial \theta_2(t)}{\partial t} - \cos(\theta_1) \frac{\partial \theta_1(t)}{\partial t} \sigma_1 - l_5 \sin(\theta_1) \sin(\theta_2) \frac{\partial \theta_2(t)}{\partial t} \end{pmatrix}$$

where

$$\sigma_1 = l_1 \cos(\theta_2) - l_5 \cos(\theta_2)$$

۱.۰.۰ محاسبه ژاکوبین به روش بازگشتی

در این بخش، تلاش می شود تا با روش بازگشتی، ماتریس ژاکوبین این ربات مجدداً محاسبه شود. برای انجام این پروژه، از معادلات DH برای محاسبه ی ماتریس های تبدیل و ماتریس ژاکوبین استفاده شده است. در

ابتدا، ماتریس DH را به صورت زیر برای ربات تعریف می کنیم:

$$\begin{pmatrix} 0 & -\frac{\pi}{2} & 0 & 0 \\ 0 & \frac{\pi}{2} & 0 & \theta_1 \\ l_1 & 0 & 0 & \theta_2 \\ l_5 & 0 & 0 & \theta_2 - \theta_3 + \pi \end{pmatrix}$$

با محاسبه ی ماتریس های تبدیل به ازای این پارامترها خواهیم داشت:

$$p = \begin{pmatrix} \cos(\theta_1) (l_1 \cos(\theta_2) - l_5 \cos(\theta_2)) \\ l_1 \sin(\theta_2) - l_5 \sin(\theta_2) \\ -\sin(\theta_1) (l_1 \cos(\theta_2) - l_5 \cos(\theta_2)) \end{pmatrix}$$

$$R = \begin{pmatrix} -\cos(\theta_1) \cos(\theta_2) & \cos(\theta_1) \sin(\theta_2) & \sin(\theta_1) \\ -\sin(\theta_2) & -\cos(\theta_2) & 0 \\ \cos(\theta_2) \sin(\theta_1) & -\sin(\theta_1) \sin(\theta_2) & \cos(\theta_1) \end{pmatrix}$$

تا این قسمت مشاهده می شود که سینماتیک مستقیم به دست آمده با روش های پیشین برابر است.

در ادامه ی این بخش، تلاش شد تا با استفاده از الگوریتم بازگشت

پاسخ سوال دو، محاسبه ی سرعت های خطی و زاویه ای در مسیر

در این بخش، با مقداردهی به پارامترهای مفاصل زاویه ای به عنوان ورودی و مشاهده ی پاسخ های به دست

آمده از ماتریس ژاکوبین، سرعت های خطی و زاویه ای ربات را محاسبه می کنیم.

به عنوان ورودی، مقادیر زیر به سرعت های زاویه ای اعمال می شوند:

$$\dot{\theta} = \begin{pmatrix} \frac{\pi \cos(t)}{\phi} \\ \frac{\pi \cos(t)}{\phi} \\ \frac{\pi \cos(t)}{\lambda} \end{pmatrix}$$

با وارد کردن این مقادیر به ماتریس ژاکوبین که در بخش قبل به صورت زیر به دست آمده است، می توانیم سرعت های زاویه ای و خطی ربات را به دست بیاوریم.

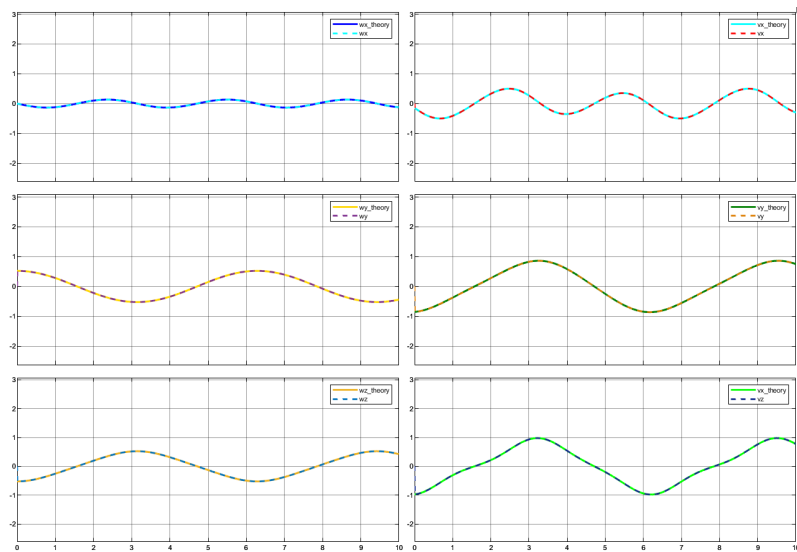
$$J = \begin{pmatrix} -\sin(\theta_1(t))(\sigma_2 - \sigma_1) & l_5 \cos(\theta_1(t)) \sin(\theta_2(t)) & -l_1 \cos(\theta_1(t)) \sin(\theta_3(t)) \\ \circ & -\sigma_1 & \sigma_2 \\ -\cos(\theta_1(t))(\sigma_2 - \sigma_1) & -l_5 \sin(\theta_1(t)) \sin(\theta_2(t)) & l_1 \sin(\theta_1(t)) \sin(\theta_3(t)) \\ \circ & \sin(\theta_1(t)) & \circ \\ \wr & \circ & \circ \\ \circ & \cos(\theta_1(t)) & \circ \end{pmatrix}$$

where

$$\sigma_1 = l_5 \cos(\theta_2(t))$$

$$\sigma_2 = l_1 \cos(\theta_3(t))$$

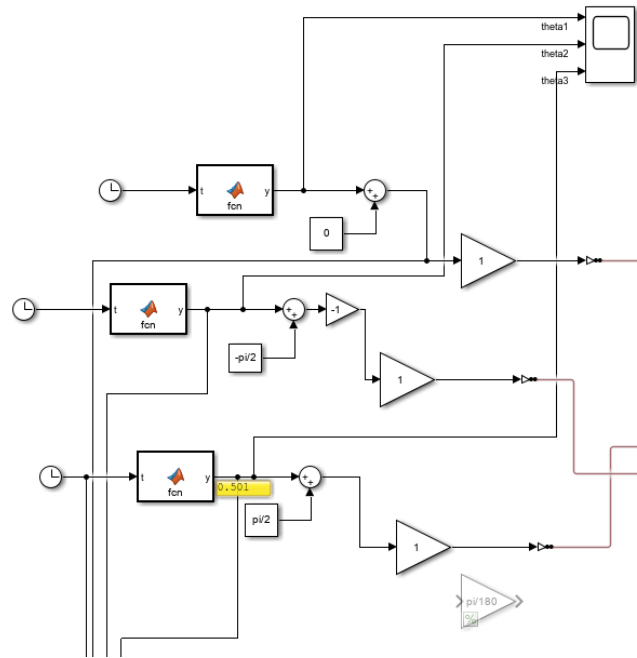
نمودار سرعت های زاویه ای و خطی ابزار این ربات با ورودی های مشخص شده در شکل به همراه مقایسه با نتایج شبیه سازی آورده شده است.



شکل ۱: سرعت های خطی و زاویه ای

پاسخ سوال سه، شبیه سازی سرعت های خطی و زاویه ای

در این بخش، به منظور صحت سنجی محاسبات انجام شده در بخش پیشین، با استفاده از سیستم شبیه سازی شده در محیط، Simscape، مسیر های پیشین به ربات اعمال شده و سرعت های خطی و زاویه ای آن اندازه گیری می شوند. برای این کار، با اضافه کردن بلوک های تابع در محیط سیمولینک، ورودی های مربوطه به صورت متغیر با زمان به هر مفصل اعمال می شوند.



شکل ۲: ورودی های مسیر-زمان

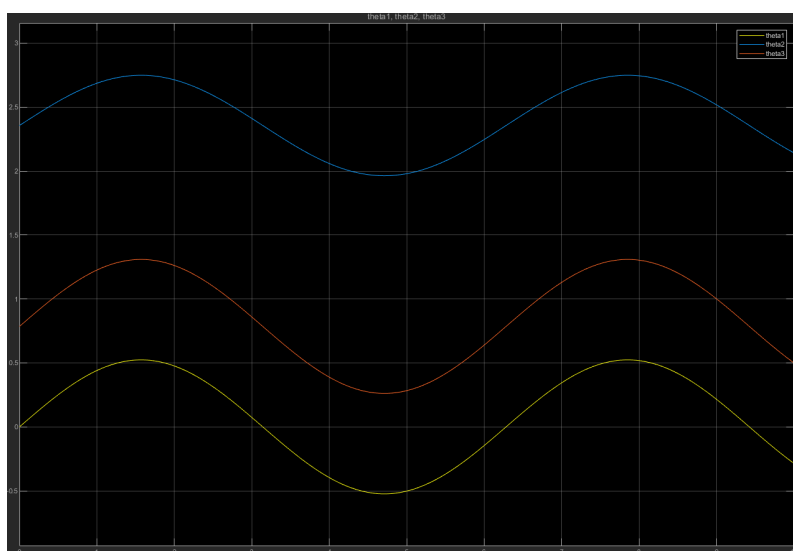
Final_Model ▶ MATLAB Function2

```

1  function y = fcn(t)
2
3  y = (pi/8) * sin(t) + 3*pi/4 ; %theta3
4

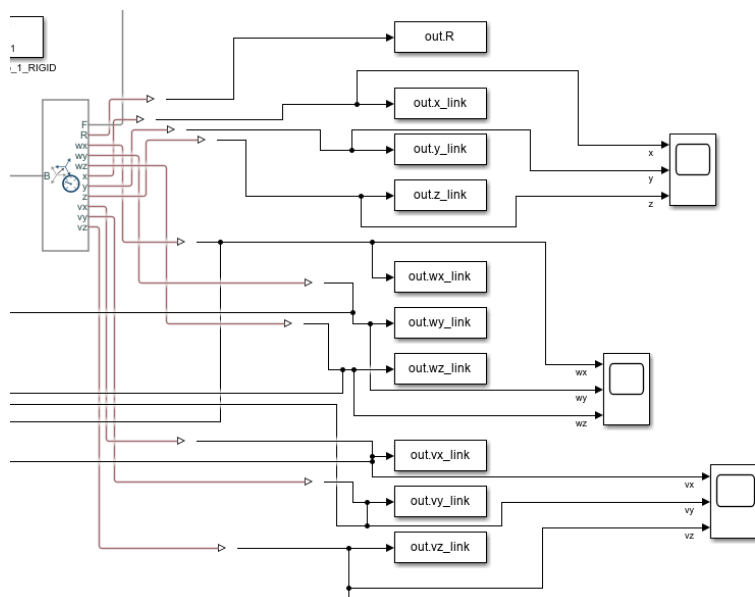
```

شکل ۳: نمونه مسیر-زمان تعریف شده



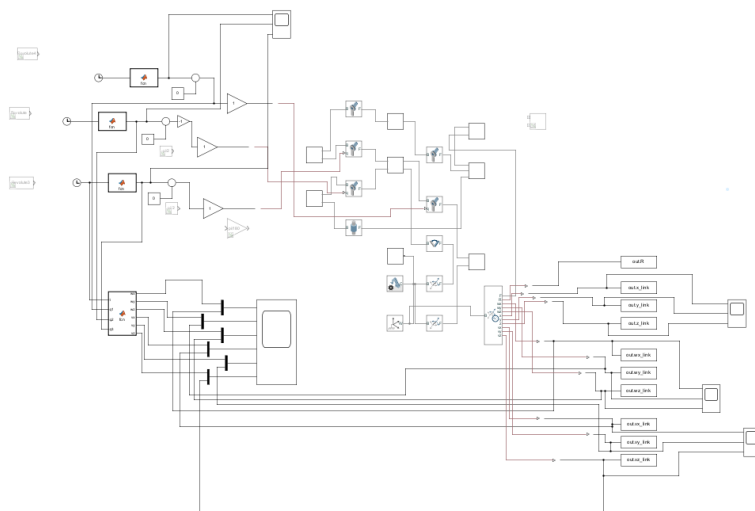
شکل ۴: نمودار ورودی های سیستم

همچنین، با اضافه کردن حسگر به بخش انتهایی مدل، می‌توانیم سرعت‌های خطی و زاویه‌ای به دست آمده از ربات را انتخاب کرده و داده‌های آن را مشاهده کنیم.



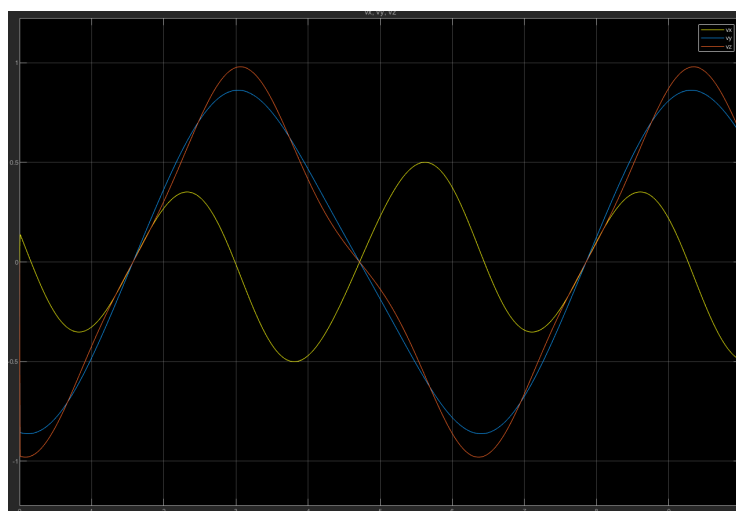
شکل ۵: بلوک خروجی

با افزودن این بلوک‌ها به سیستم، سیستم نهایی به شکل زیر به دست می‌آید.

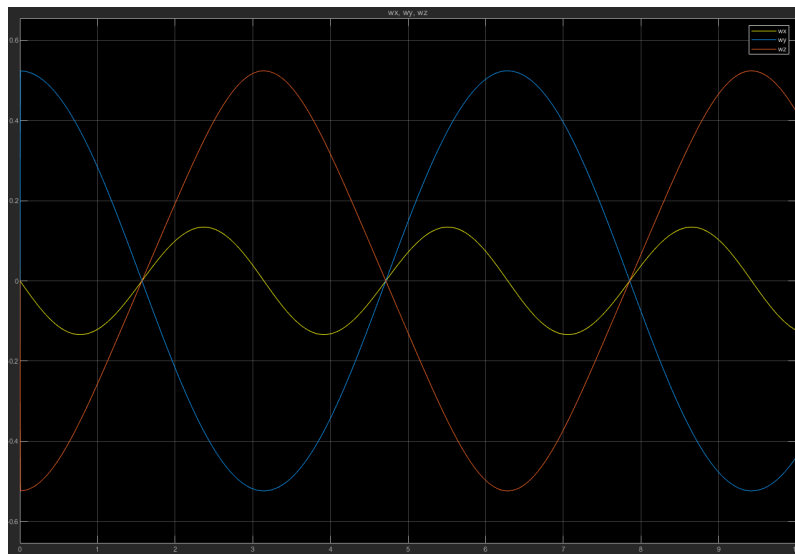


شکل ۶: نمای کلی سیستم

با اجرای شبیه سازی، می توانیم سرعت های خطی و زاویه ای که توسط حسگر اندازه گیری شده اند را نمایش دهیم. در تصاویر زیر، به ترتیب سرعت های خطی و زاویه ای نمایش داده شده اند.



شکل ۷: سرعت خطی



شکل ۸: سرعت زاویه ای

همچنین، مطابق خواسته ی سوال، به منظور اعتبارسنجی نتایج به دست آمده در شبیه سازی با سرعت های به دست آمده از روش های تحلیلی، نتایج دو روش را در کنار یکدیگر رسم و مقایسه می کنیم. برای این کار، ابتدا بلوکی حاوی ماتریس ژاکوبین به دست آمده از روش تحلیلی قرار داده می شود.

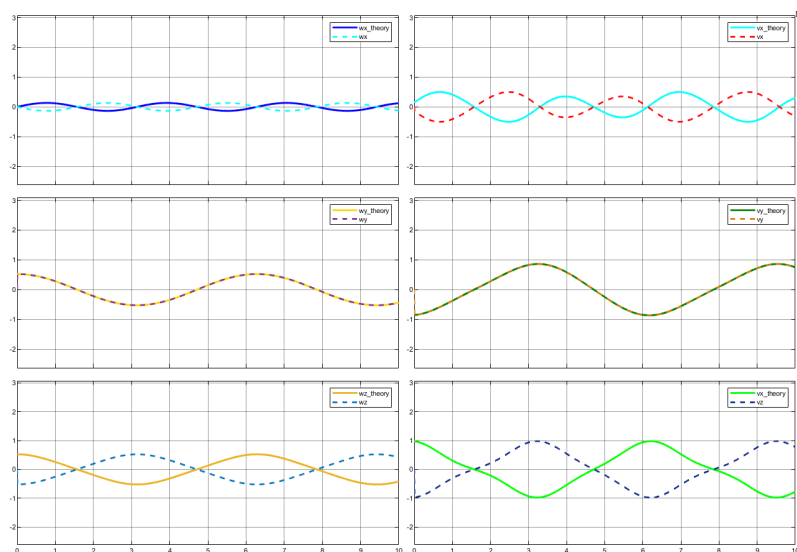
```

Final_Model ▶ MATLAB Function3
1 function [wx, wy, wz, vx, vy, vz] = fcn(t, q1, q2, q3)
2
3
4
5 vx = (((pi/6)*1.35*cos(q1)*sin(q2)*cos(t)) - ((pi/8)*1.28*cos(q1)*sin(q3)*cos(t)) - ((pi/6)*sin(q1)*cos(t)*(1.28*cos(q3) - 1.35*cos(q2))))
6 vy = (((pi/8)*1.28*cos(q3)*cos(t)) - ((pi/6)*1.35*cos(q2)*cos(t)))
7 vz = (((pi/8)*1.28*sin(q1)*sin(q3)*cos(t)) - ((pi/6)*cos(q1)*cos(t)*(1.28*cos(q3) - 1.35*cos(q2))) - ((pi/6)*1.35*sin(q1)*sin(q2)*cos(t)))
8
9 wx = (pi/6) * cos(t) * sin(q1); %theta2
10 wy = (pi/6) * cos(t);
11 wz = (pi/6) * cos(t) * cos(q1);
12

```

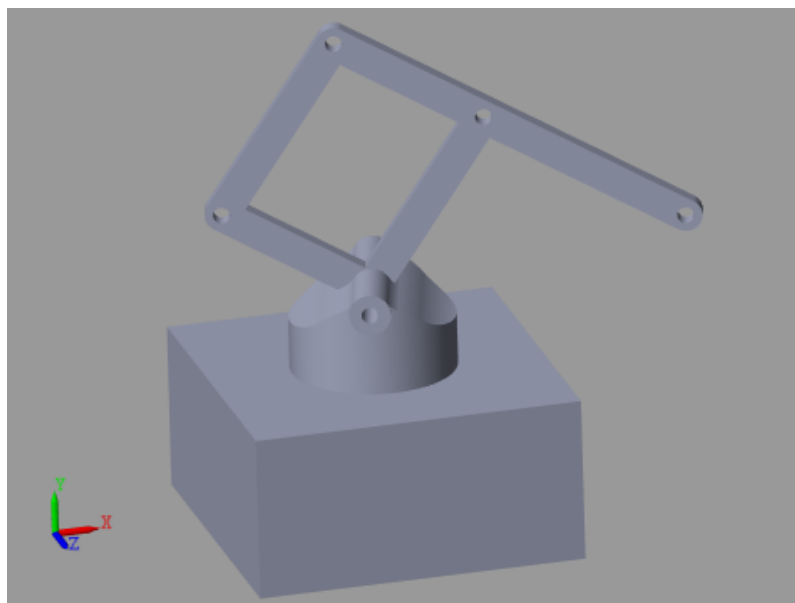
شکل ۹: ماتریس ژاکوبین به دست آمده از روش تحلیلی

با نمایش خروجی های به دست آمده از این دو روش خواهیم داشت:

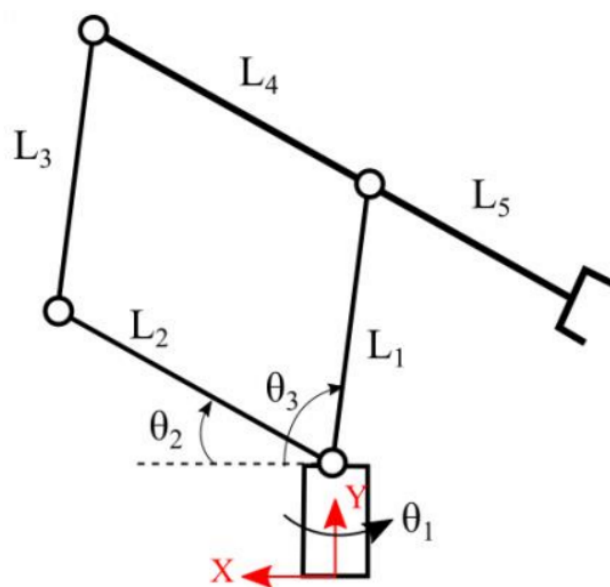


شکل ۱۰: مقایسه نتایج روش تحلیلی و شبیه سازی

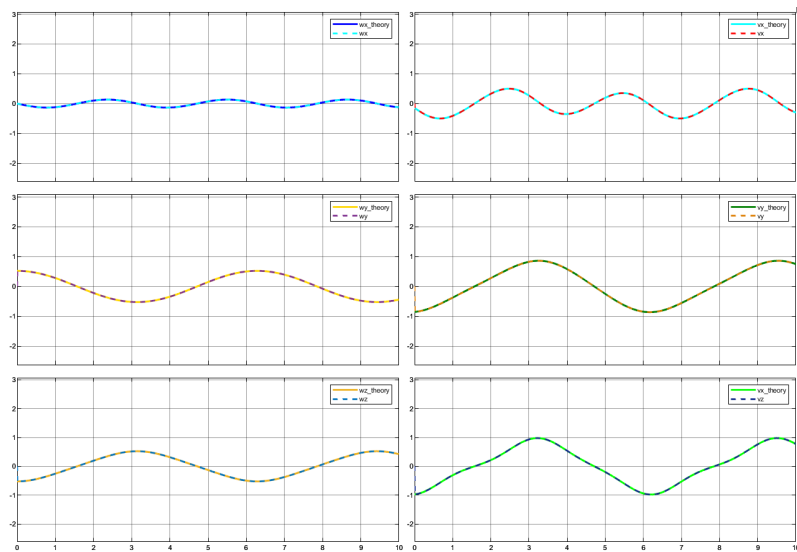
با مشاهده ی نتایج این مقایسه در خواهیم یافت که سرعت های خطی و زاویه ای در راستاهای X و Z دارای علامتی مخالف یکدیگر هستند. علت این اتفاق، تفاوت تعریف محور های مختصات پایه می باشد. در شبیه سازی فعلی، جهت محور ها مطابق با شکل ۱۱ در نظر گرفته شده است، در حالی که در پاسخ ارائه شده برای این سیستم مطابق شکل ۱۲، جهت محور های X و Z در خلاف این جهت ها هستند. از این رو، نتایج به دست آمده از این دو روش، به لحاظ مقداری برابر، اما در علامت با یکدیگر تفاوت دارند. بنابراین، به منظور یکسان سازی خروجی های شبیه سازی با محاسبات، این مقادیر برای محورهای X و Z با علامت منفی در نظر گرفته شده اند.



شکل ۱۱



شکل ۱۲



شکل ۱۳: مقایسه نتایج دو روش پس از یکسان سازی محور ها

پاسخ سوال چهار، محاسبه ی گشتاورهای مفاصل برای وارد کردن نیرو

محاسبه ی گشتاور های لازم برای وارد کردن نیرو به محیط مطابق رابطه ی ۴/۶۱ و با استفاده از ماتریس ترانهاده ی ژاکوبین انجام می شود. این ماتریس، با استفاده از ماتریس ژاکوبین که در مراحل قبل محاسبه شد به صورت زیر به دست می آید.

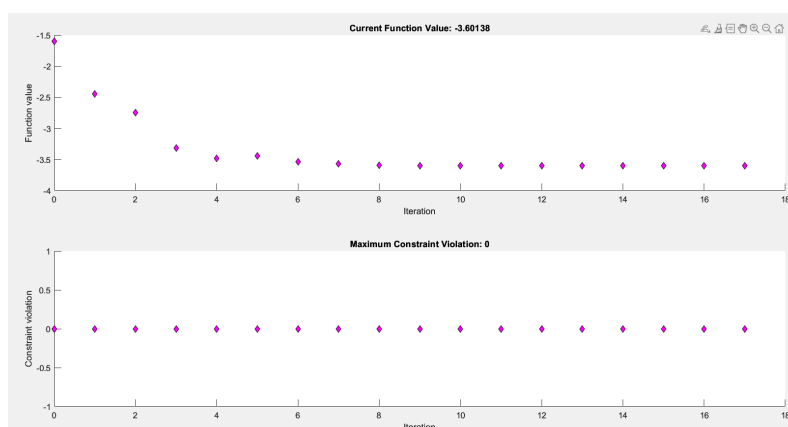
$$\begin{pmatrix} -\sin(\theta_1(t))(\sigma_2 - \sigma_1) & 0 & -\cos(\theta_1(t))(\sigma_2 - \sigma_1) \\ l_5 \cos(\theta_1(t)) \sin(\theta_2(t)) & -\sigma_1 & -l_5 \sin(\theta_1(t)) \sin(\theta_2(t)) \\ -l_1 \cos(\theta_1(t)) \sin(\theta_3(t)) & \sigma_2 & l_1 \sin(\theta_1(t)) \sin(\theta_3(t)) \end{pmatrix}$$

where

$$\sigma_1 = l_5 \cos(\theta_2(t))$$

$$\sigma_2 = l_1 \cos(\theta_3(t))$$

مشاهده می شود که با ضرب این مقادیر در نیروهای وارد بر سیستم از طرف محیط، می توان میزان گشتاور مورد نیاز در هر مفصل را به دست آورد. حال، به منظور پیدا کردن نقطه ای که در آن بیشترین میزان نیرو از طرف محیط به ابزار وارد می شود، لازم است تا یک مسئله بهینه سازی طراحی و حل شود. در این مسئله، هدف پیدا کردن نقطه ای است که مقدار ماتریس ترانزاده ی ژاکوبین بیشینه شود. برای این کار، با تعریف نرم این ماتریس به عنوان تابع هدف، می توانیم مسئله ی بهینه سازی را تعریف کنیم. باید در نظر داشت که حل این مسئله به صورت مقید می باشد و قیدهای ذکر شده در مسئله برای هر مفصل باید در نظر گرفته شوند. برای حل مسئله، از ابزار `fmincon` استفاده شده است و با تنظیم پارامترهای این تابع بهینه سازی، تلاش شده است تا نتایج قابل قبولی دریافت بشود.



شکل ۱۴

با اجرای کد بهینه سازی برای این تابع، مقدار پارامترهایی که تابع را ماکسیمم می کنند به صورت زیر به دست می آید.

$$\theta_1 = 0.6492 \text{ rad } (37.2^\circ)$$

$$\theta_2 = 0.0000 \text{ rad } (0.0^\circ)$$

$$\theta_3 = 2.3562 \text{ rad } (135.0^\circ)$$

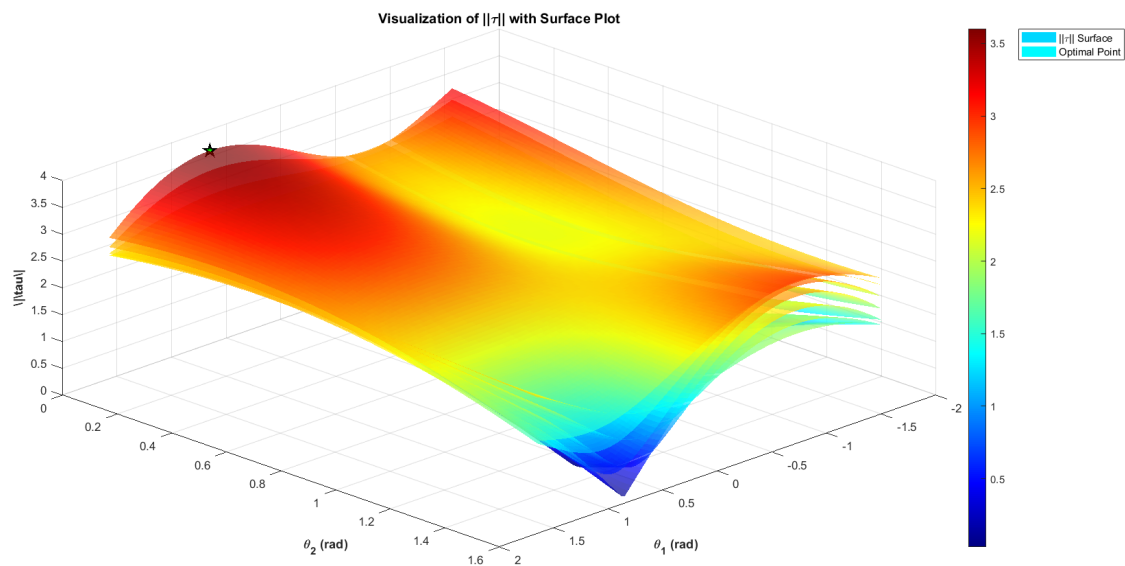
$$\text{Maximum} \|\tau\| = 3.6014$$

همان طور که مشاهده می شود، مانند اکثر مسائل بهینه سازی، در اینجا نیز نقطه ی بهینه در مرز فضای کاری و محل برخورد دو قید به دست آمده است. همچنین در این نقطه، بیشترین گشتاوری که در هر یک از مفاصل ایجاد

می شود به شکل زیر به دست آمده است:

$$\begin{bmatrix} 3,1597 \\ -1,3500 \\ -1,0789 \end{bmatrix}$$

در نهایت، به منظور تصویرسازی بهتر این نتایج، مقدار این تابع در فضای کاری رسم شده و بیشترین مقدار با علامت ستاره نمایش داده می شود.



شکل ۱۵

ساخت ربات

در بخش اول از انجام پروژه، قطعات ربات توسط پرینتر سه بعدی چاپ و آماده شدند. در این بخش، با مونتاژ این قطعات و اتصال لینک ها به یکدیگر، ساختار مکانیکی ربات راه اندازی شده است. توضیحات بیشتر مربوط به این بخش در ویدیوی مربوطه در لینک زیر قابل مشاهده است.

Link File Drive Google