به نام خدا

گزارش‌کار پروژه درس مقدمه ای بر رباتیک

استاد درس: دکتر خسروی

اعضای گروه:

سلمان عامی مطلق

شیما سادات ناصری

سید محمد صالح میرزا طباطبایی

بهار ۱۴۰۰

فهرست

مقدمه:

امروزه استفاده از ربات ها در صنایع مختلف بسیار متداول شده و روز به روز در حال پیشرفت و توسعه می‌باشد. طراحی این ربات ها عموما به صورت سری یا موازی انجام می‌شود که هر کدام از این طراحی ها کاربرد ها و استفاده های خود را دارد. در این میان ربات های هیبرید که عموما دارای طراحی ترکیبی سری و موازی هستند نیز دارای استفاده های خاص خود است. مزیت مهم این دسته از ربات ها این است که می‌توانند ویژگی های هر دو مدل سری و موازی را داشته باشند از جمله کاهش حرکات ناخواسته کوچک و صلب بالا در ربات های موازی و فضای کاری بزرگ در ربات های سری.

شکل 1- نمایی از ربات مورد بررسی

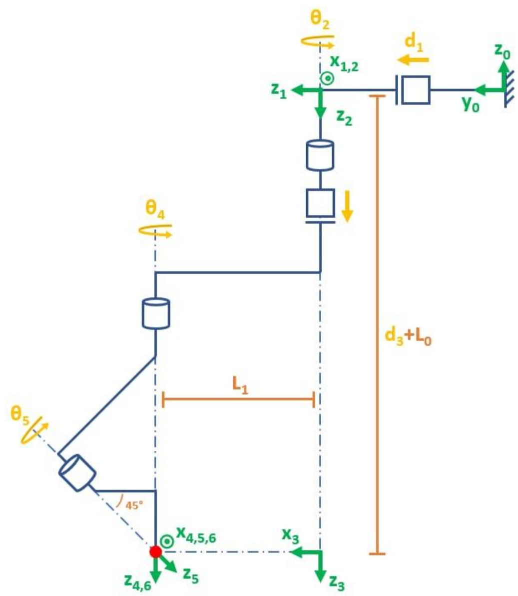
رباتی که ما در این پروژه از آن استفاده کردیم یک ربات ۵ درجه آزادی است که ۲ درجه‌ی آن موازی و ۳ درجه‌ی آن سری می‌باشد. کاربرد اصلی این ربات و طراحی آن به دلیل امکان تغییر زاویه مجری نهایی بدون تغییر مکان آن، روی برای ماشینکاری سطح و کاربردهای ساختمانی متمرکز شده است. هر چند این ربات دارای مزیت های گفته شده در قبل است اما دارای معایبی نیز می‌باشد که می‌توان به پیچیدگی ربات که به دلیل نیازمند بودن این دسته از ربات ها به یک صفحه‌ی انتقالی یا چرخشی اشاره کرد. کنترل این دسته از ربات ها، در صورتی که دارای افزونگی در مفاصل فعال باشند، بسیار سخت می‌شود. علاوه بر این، با اینکه سیستم دارای قسمت سری است که فضای کاری بزرگی می‌تواند داشته باشد، اما همان بخش موازی سیستم را محدود می‌کند و از بزرگی فضای کاری ربات می‌کاهد.

ما در این پروژه ابدا به بررسی سینماتیک مستقیم و معکوس سیستم، فضای کاری و ژاکوبین سیستم، دینامیک سیستم و در نهایت تعیین مسیر مرجع، کنترل و شبیه سازی پرداخته‌ایم. از نکات حائز اهمیت این پروژه، تعریف live script ها و کد هایی جدید برای محاسبه‌ی سینماتیک مستقیم و معکوس این ربات، فضای کاری، ژاکوبین و دینامیک و مدل استاندارد دینامیکی ربات است که تماما توسط اعضای گروه نوشته شده‌اند که می‌توان خروجی latex آن را نیز از خود متلب دریافت کرد.(این کد ها در نسخه های متلب 2020 به بالا یه دلیل دستورات جدید قادر به پاسخگویی کامل هستند).

محاسبات سینماتیکی:

۱. شماتیک ربات و جدول DH بهبود یافته مربوط به آن

همانطور که در قسمت قبل گفته شد، ربات شامل دو درجه آزادی موازی و سه درجه آزادی سری می‌باشد. در این پروژه سعی شد تا بر اساس کاری که قسمت موازی ربات انجام می‌دهد، آن را به دو مفصل سری مدل کرده و ادامه مسیر را با آن پیش ببریم. بعد از بررسی های انجام شده روی نحوه کارکرد ربات به شماتیک ساده‌ی زیر رسیدیم:



شکل 2- شماتیک ربات

دو مفصل اول حاصل سیستم مدل شده ی موازی است. نکته بسیار مهم در رویه کاری ربات، مجری نهایی آن است. مجری نهایی در این ربات، در صورت عدم جابجایی d_1 و θ_2 و d_3 ، در یک نقطه ثابت قرار دارد و می‌تواند فقط جهت گیری خود را (با استفاده از θ_4 و θ_5) تغییر دهد. تغییرات مکانی مجری نهایی نیز بر عهده d_1 و θ_2 و d_3 می‌باشد. جدول DH بهبود یافته مربوط به این شماتیک در زیر آمده است.

جدول ۱-۲- مقادیر DH بهبود یافته

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	90	0	d_1	0
2	90	0	0	θ_2
3	0	0	L_0+d_3	90
4	0	L_1	0	θ_4
5	45	0	0	θ_5
6	-45	0	0	0

در ادامه گزارش تمامی d ها و θ را توسط φ_i ها (i=1,...,6) که پارامتری وابسته به موتور های هر لینک است نمایش خواهیم داد به طوری که:

$$\varphi_1 = d_1 \quad , \quad \varphi_2 = \theta_2 \quad , \quad \varphi_3 = d_3 \quad , \quad \varphi_4 = \theta_4 \quad , \quad \varphi_5 = \theta_5$$

۲. سینماتیک مستقیم

بعد از بدست آوردن مقادیر DH ربات، ماتریس تبدیل هر مفصل به صورت جدا توسط فرمول زیر بدست می‌آیند:

$$ {}^{i-1}T_i = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1} d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (1)$$

ماتریس تبدیل هر مفصل توسط کد متلب بدست آمده که به شرح زیر می‌باشد(به دلیل محدودت های نام گذاری در متلب L_1 بجای L_0 و L_2 بجای L_1 قرار داده شده اند):

$$T_0^1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -\phi_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_1^2 = \begin{bmatrix} \cos(\phi_2) & -\sin(\phi_2) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin(\phi_2) & \cos(\phi_2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^3 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & \phi_3 + L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3^4 = \begin{bmatrix} \cos(\phi_4) & -\sin(\phi_4) & 0 & L_2 \\ \sin(\phi_4) & \cos(\phi_4) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_4^5 = \begin{bmatrix} \cos(\phi_5) & -\sin(\phi_5) & 0 & 0 \\ \frac{\sqrt{2}\ \sin(\phi_5)}{2} & \frac{\sqrt{2}\ \cos(\phi_5)}{2} & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}\ \sin(\phi_5)}{2} & \frac{\sqrt{2}\ \cos(\phi_5)}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_5^6 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

می‌دانیم که از ضرب کردن این ماتریس ها می‌توانیم به ماتریس نهایی ربات که base را به مجری نهایی مرتبط می-کند، برسیم. با استفاده از این موضوع به این ماتریس رسیده ایم.

$$T_0^6 = \begin{bmatrix} \frac{\sqrt{2}\ \sin(\phi_2)\sin(\phi_4)\sin(\phi_5)}{2} - \cos(\phi_4)\cos(\phi_5)\sin(\phi_2) - \frac{\sqrt{2}\ \cos(\phi_2)\cos(\phi_4)\sin(\phi_5)}{2} - \cos(\phi_2)\cos(\phi_5)\sin(\phi_4) & \sigma_5 - \frac{\sigma_9}{4} - \sigma_2 - \frac{\sigma_{10}}{4} - \sigma_6 & \sigma_2 - \frac{\sigma_9}{4} - \frac{\sigma_{10}}{4} + \sigma_5 - \sigma_6 & -L_2\sin(\phi_2) \\ \cos(\phi_5)\sin(\phi_2)\sin(\phi_4) - \cos(\phi_2)\cos(\phi_4)\cos(\phi_5) + \frac{\sqrt{2}\ \cos(\phi_2)\sin(\phi_4)\sin(\phi_5)}{2} + \frac{\sqrt{2}\ \cos(\phi_4)\sin(\phi_2)\sin(\phi_5)}{2} & \frac{\sigma_8}{4} + \frac{\sigma_7}{4} + \sigma_1 - \sigma_3 + \sigma_4 & \frac{\sigma_8}{4} + \frac{\sigma_7}{4} - \sigma_1 - \sigma_3 + \sigma_4 & -\phi_1 - L_2\cos(\phi_2) \\ -\frac{\sqrt{2}\ \sin(\phi_5)}{2} & \frac{1}{2} - \frac{\cos(\phi_5)}{2} & -\frac{\cos(\phi_5)}{2} - \frac{1}{2} & -\phi_3 - L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where

$$\sigma_1 = \frac{\sin(\phi_2 + \phi_4)}{2}$$

$$\sigma_2 = \frac{\cos(\phi_2 + \phi_4)}{2}$$

$$\sigma_3 = \frac{\sqrt{2}\ \sigma_7}{4}$$

$$\sigma_4 = \frac{\sqrt{2}\ \sigma_8}{4}$$

$$\sigma_5 = \frac{\sqrt{2}\ \sigma_9}{4}$$

$$\sigma_6 = \frac{\sqrt{2}\ \sigma_{10}}{4}$$

$$\sigma_7 = \sin(\phi_2 + \phi_4 - \phi_5)$$

$$\sigma_8 = \sin(\phi_2 + \phi_4 + \phi_5)$$

$$\sigma_9 = \cos(\phi_2 + \phi_4 - \phi_5)$$

$$\sigma_{10} = \cos(\phi_2 + \phi_4 + \phi_5)$$

با کمی دقت در ماتریس نهایی می‌توان دریافت که طبق شماتیک، φ_1 و φ_2 و φ_3 در تعیین موقعیت و φ_4 و φ_5 به همراه φ_2 در جهت گیری مجری نهایی نقش دارند.

۳. سینماتیک معکوس سیستم

اگر فرض شود که ماتریسی به صورت زیر و با داده های معلوم به ما داده شود، با استفاده از تناظر بخش موقعیت و جهت گیری دو ماتریس می‌توان به معادله ی هر φ دست یافت.

$$T_0^6 = \begin{bmatrix} R_{11} & R_{12} & R_{13} & P_x \\ R_{21} & R_{22} & R_{23} & P_y \\ R_{31} & R_{32} & R_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

با متناظر قرار دادن این ماتریس و ماتریس سینماتیک مستقیم بدست آمده در بخش قبل، می‌توان به معادلات زیر رسید.

$$P_z = -\phi_3 - L_1$$

$$P_x = -L_2 \sin(\phi_2)$$

$$P_y = -\phi_1 - L_2 \cos(\phi_2)$$

$$R_{31} = -\frac{\sqrt{2}\ \sin(\phi_5)}{2}$$

$$R_{11} = \frac{\sqrt{2}\ \sin(\phi_2)\sin(\phi_4)\sin(\phi_5)}{2} - \cos(\phi_4)\cos(\phi_5)\sin(\phi_2) - \frac{\sqrt{2}\ \cos(\phi_2)\cos(\phi_4)\sin(\phi_5)}{2} - \cos(\phi_2)\cos(\phi_5)\sin(\phi_4)$$

با حل این معادلات، مقادیر(و شرایط وجود جواب) φ های سیستم بر اساس موقعیت و جهت گیری مدنظر بدست می‌آیند:

$$\phi_3 = -L_1 - P_z \tag{2}$$

$$\phi_5 = \begin{bmatrix} -\mathrm{asin}\left(\sqrt{2}\ R_{31}\right) \\ \pi + \mathrm{asin}\left(\sqrt{2}\ R_{31}\right) \end{bmatrix} \tag{3}$$

$$\text{When} \quad \begin{bmatrix} k \in \mathbb{Z} \wedge -1 \le 1.4142\, R_{31} \wedge 1.4142\, R_{31} \le 1 \\ k \in \mathbb{Z} \wedge -1 \le 1.4142\, R_{31} \wedge 1.4142\, R_{31} \le 1 \end{bmatrix}$$

$$\phi_2 = \begin{bmatrix} -\mathrm{asin}\left(\frac{P_x}{L_2}\right) \\ \pi + \mathrm{asin}\left(\frac{P_x}{L_2}\right) \end{bmatrix} \tag{4}$$

When $\begin{bmatrix} k \in \mathbb{Z} \wedge -1 \leq \frac{P_x}{L_2} \wedge \frac{P_x}{L_2} \leq 1 \\ k \in \mathbb{Z} \wedge -1 \leq \frac{P_x}{L_2} \wedge \frac{P_x}{L_2} \leq 1 \end{bmatrix}$

$$\phi_1 = -P_y - L_2 \cos(\phi_2) \tag{5}$$

When $P_y \in \mathbb{R}$

$$\phi_4 = \begin{bmatrix} -2\,\mathrm{atan}\left(\frac{\sigma_1 - 2\cos(\phi_2)\cos(\phi_5) + \sqrt{2}\,\sin(\phi_2)\sin(\phi_5)}{\sigma_2}\right) \\ 2\,\mathrm{atan}\left(\frac{2\cos(\phi_2)\cos(\phi_5) + \sigma_1 - \sqrt{2}\,\sin(\phi_2)\sin(\phi_5)}{\sigma_2}\right) \end{bmatrix} \tag{6}$$

where

$$\sigma_1 = \sqrt{2}\ \sqrt{-2\,{R_{11}}^2 + 2\cos(\phi_2)^2\cos(\phi_5)^2 + \cos(\phi_2)^2\sin(\phi_5)^2 + 2\cos(\phi_5)^2\sin(\phi_2)^2 + \sin(\phi_2)^2\sin(\phi_5)^2}$$

$$\sigma_2 = 2\cos(\phi_5)\sin(\phi_2) - 2\,R_{11} + \sqrt{2}\,\cos(\phi_2)\sin(\phi_5)$$

When

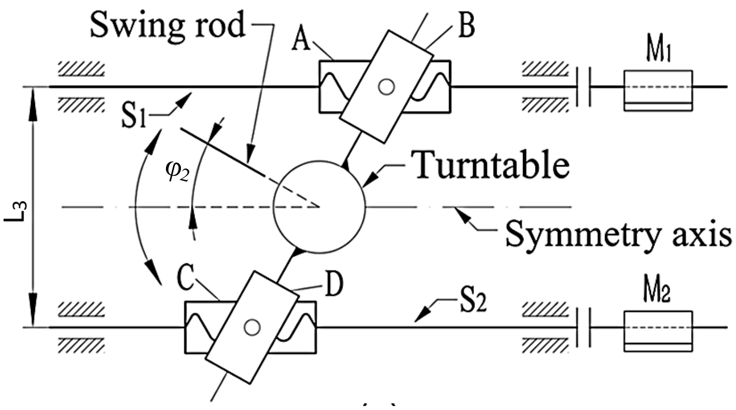
$$\begin{bmatrix} 2\,{R_{11}}^2 \leq \sigma_1 \wedge k \in \mathbb{Z} \wedge 2\,R_{11} \neq 1.4142\cos(\phi_2)\sin(\phi_5) + 2\cos(\phi_5)\sin(\phi_2) \\ 2\,{R_{11}}^2 \leq \sigma_1 \wedge k \in \mathbb{Z} \wedge 2\,R_{11} \neq 1.4142\cos(\phi_2)\sin(\phi_5) + 2\cos(\phi_5)\sin(\phi_2) \end{bmatrix}$$

where

$$\sigma_1 = 2\cos(\phi_2)^2\cos(\phi_5)^2 + \cos(\phi_2)^2\sin(\phi_5)^2 + 2\cos(\phi_5)^2\sin(\phi_2)^2 + \sin(\phi_2)^2\sin(\phi_5)^2$$

جواب اصلی زاویه φ_4 شامل قسمت موهومی نیز می‌شود که این بخش به دلیل وجود محدودیت در جهت گیری برای مجری نهایی است.مجری نهایی دارای 2 درجه آزادی برای حرکت چرحشی است و به دلیل نبود درجه سوم این حالت ایجاد می‌شود.

می‌دانیم که φ_1 و φ_2 به عنوان حالت مدل شده ی سیستم موازی ماست. شماتیک بخش موازی را در زیر می‌بینید.



شکل 3- نمایی شماتیک از بخش موازی سیستم

حال اگر حرکت در راستای S_1 را با X_1 و حرکت در راستای S_2 را با X_2 نمایش دهیم؛ خواهیم داشت.

$$cos^2\varphi_2 = \frac{L_3^2}{(X_2 - X_1)^2 + L_3^2}$$

$$\Rightarrow |X_2 - X_1| = L_3 \tan\varphi_2 \qquad (7)$$

از طرفی برای حرکت انتقالی آن خواهیم داشت:

$$\varphi_1 = \frac{X_1 + X_2}{2}$$

$$\Rightarrow X_1 + X_2 = 2\varphi_1 \qquad (8)$$

از حل دو معادله (7) و (8) به دو جواب زیر خواهیم رسید.

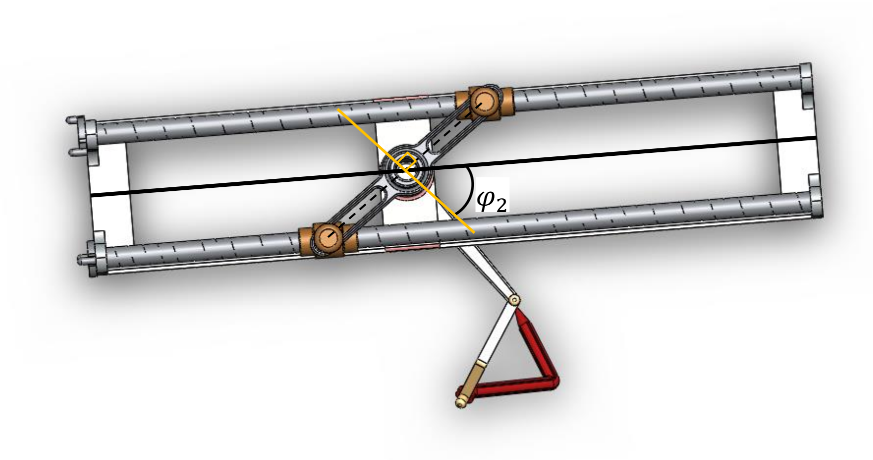
$$X_1 = \varphi_1 \pm \frac{L_3 \tan\varphi_2}{2}$$

$$X_2 = \varphi_1 \mp \frac{L_3 \tan\varphi_2}{2}$$

که شرط وجود جواب برای این دو مثبت بودن مقدار تانژانت φ_2 است.

تعداد جواب های سینماتیک معکوس

۱. برای مقدار زاویه φ_2:



شکل 4- نمای فیزیکی φ_2

همانطور که دیده می‌شود، بازه حقیقی φ_2 بین ۵۳- تا ۵۳ درجه است و بیشتر از آن قابلیت تغییر ندارد؛ بنابراین جواب دوم سیستم که ۱۸۰ درجه با آن اختلاف فازی دارد قابل قبول نیست.

۲. برای مقدار زاویه φ_4 و φ_5:

این دو زاویه بدون محدودیت هستند و فقط وابسته به ورودی ماتریس جهت گیری هستند که شرایط وجود جواب برای هر کدام در بخش قبل گفته شد.

محاسبات دینامیکی:

۱. ژاکوبین ربات

در این بخش طبق فرمول های اشاعه سرعت خطی و زاویه ای پیش می‌رویم.

برای مفصل چرخشی داریم:

$$^{i+1}\omega_{i+1} = {}^{i+1}_{i}R\, {}^{i}\omega_i + \dot{\theta}_{i+1}\, {}^{i+1}\hat{Z}_{i+1}.$$

$$^{i+1}v_{i+1} = {}^{i+1}_{i}R({}^{i}v_i + {}^{i}\omega_i \times {}^{i}P_{i+1}).$$

و برای مفصل انتقالی داریم:

$$^{i+1}\omega_{i+1} = {}^{i+1}_{i}R\, {}^{i}\omega_i,$$
$$^{i+1}v_{i+1} = {}^{i+1}_{i}R({}^{i}v_i + {}^{i}\omega_i \times {}^{i}P_{i+1}) + \dot{d}_{i+1}\, {}^{i+1}\hat{Z}_{i+1}.$$

در نهایت روی سرعت های بدست آمده در مجری نهایی دستور ژاکوبین را اجرا می‌کنیم. سپس برای بدست آوردن ژاکوبین روی base از تبدیل ماتریس دورانی استفاده می‌کنیم.

ماتریس های سرعت خطی و دورانی مجری نهایی در حالتی که سرعت اولیه صفر باشد، برابر است با:

$$\omega_6 = \begin{bmatrix} \dfrac{\sqrt{2}\,\sin(\phi_5)\,(\dot{\phi}_2+\dot{\phi}_4)}{2} \\ \dfrac{\dot{\phi}_2\cos(\phi_5)}{2} - \dfrac{\dot{\phi}_4}{2} - \dfrac{\sqrt{2}\,\dot{\phi}_5}{2} - \dfrac{\dot{\phi}_2}{2} + \dfrac{\dot{\phi}_4\cos(\phi_5)}{2} \\ \dfrac{\dot{\phi}_2}{2} + \dfrac{\dot{\phi}_4}{2} + \dfrac{\sqrt{2}\,\dot{\phi}_5}{2} + \dfrac{\dot{\phi}_2\cos(\phi_5)}{2} + \dfrac{\dot{\phi}_4\cos(\phi_5)}{2} \end{bmatrix}$$

$$v_6 = \begin{bmatrix} \frac{\sqrt{2}\ \dot{\phi}_3 \sin(\phi_5)}{2} + \frac{9\,\dot{\phi}_2 \cos(\phi_5)\sin(\phi_4)}{20} + \dot{\phi}_1 \cos(\phi_2)\cos(\phi_4)\cos(\phi_5) - \dot{\phi}_1 \cos(\phi_5)\sin(\phi_2)\sin(\phi_4) + \frac{9\ \sqrt{2}\ \dot{\phi}_2 \cos(\phi_4)\sin(\phi_5)}{40} - \frac{\sqrt{2}\ \dot{\phi}_1 \cos(\phi_2)\sin(\phi_4)\sin(\phi_5)}{2} - \frac{\sqrt{2}\ \dot{\phi}_1 \cos(\phi_4)\sin(\phi_2)\sin(\phi_5)}{2} \\ \sigma_9 - \frac{\dot{\phi}_3}{2} + \sigma_{10} + \sigma_6 - \sigma_8 - \sigma_7 - \sigma_5 - \sigma_4 - \sigma_3 - \sigma_2 + \sigma_1 \\ \frac{\dot{\phi}_3}{2} - \sigma_9 + \sigma_{10} + \sigma_6 + \sigma_8 + \sigma_7 - \sigma_5 - \sigma_4 - \sigma_3 - \sigma_2 + \sigma_1 \end{bmatrix}$$

where

$$\sigma_1 = \frac{\sqrt{2}\ \dot{\phi}_1 \sin(\phi_2)\sin(\phi_4)\sin(\phi_5)}{2}$$

$$\sigma_2 = \frac{\sqrt{2}\ \dot{\phi}_1 \cos(\phi_2)\cos(\phi_4)\sin(\phi_5)}{2}$$

$$\sigma_3 = \frac{9\ \sqrt{2}\ \dot{\phi}_2 \sin(\phi_4)\sin(\phi_5)}{40}$$

$$\sigma_4 = \frac{\dot{\phi}_1 \cos(\phi_4)\cos(\phi_5)\sin(\phi_2)}{2}$$

$$\sigma_5 = \frac{\dot{\phi}_1 \cos(\phi_2)\cos(\phi_5)\sin(\phi_4)}{2}$$

$$\sigma_6 = \frac{9\,\dot{\phi}_2 \cos(\phi_4)\cos(\phi_5)}{40}$$

$$\sigma_7 = \frac{\dot{\phi}_1 \cos(\phi_4)\sin(\phi_2)}{2}$$

$$\sigma_8 = \frac{\dot{\phi}_1 \cos(\phi_2)\sin(\phi_4)}{2}$$

$$\sigma_9 = \frac{9\,\dot{\phi}_2 \cos(\phi_4)}{40}$$

$$\sigma_{10} = \frac{\dot{\phi}_3 \cos(\phi_5)}{2}$$

در نهایت ژاکوبین مجری نهایی برابر است با:

$$J^{ee} = \begin{bmatrix} \cos(\phi_2)\cos(\phi_4)\cos(\phi_5) - \cos(\phi_5)\sin(\phi_2)\sin(\phi_4) - \frac{\sqrt{2}\,\cos(\phi_2)\sin(\phi_4)\sin(\phi_5)}{2} - \frac{\sqrt{2}\,\cos(\phi_4)\sin(\phi_2)\sin(\phi_5)}{2} & \frac{9\cos(\phi_5)\sin(\phi_4)}{20} + \frac{9\,\sqrt{2}\,\cos(\phi_4)\sin(\phi_5)}{40} & \sigma_3 & 0 & 0 \\ \sigma_8 - \frac{\sigma_{10}}{4} - \sigma_7 - \frac{\sigma_{11}}{4} - \sigma_9 & \sigma_6 + \sigma_5 - \sigma_4 & \sigma_1 & 0 & 0 \\ \sigma_7 - \frac{\sigma_{10}}{4} - \frac{\sigma_{11}}{4} + \sigma_8 - \sigma_9 & \sigma_5 - \sigma_6 - \sigma_4 & \sigma_2 & 0 & 0 \\ 0 & \sigma_3 & 0 & \sigma_3 & 0 \\ 0 & \sigma_1 & 0 & \sigma_1 & -\frac{\sqrt{2}}{2} \\ 0 & \sigma_2 & 0 & \sigma_2 & \frac{\sqrt{2}}{2} \end{bmatrix}$$

where

$$\sigma_1 = \frac{\cos(\phi_5)}{2} - \frac{1}{2}$$

$$\sigma_2 = \frac{\cos(\phi_5)}{2} + \frac{1}{2}$$

$$\sigma_3 = \frac{\sqrt{2}\,\sin(\phi_5)}{2}$$

$$\sigma_4 = \frac{9\,\sqrt{2}\,\sin(\phi_4)\sin(\phi_5)}{40}$$

$$\sigma_5 = \frac{9\cos(\phi_4)\cos(\phi_5)}{40}$$

$$\sigma_6 = \frac{9\cos(\phi_4)}{40}$$

$$\sigma_7 = \frac{\sin(\phi_2 + \phi_4)}{2}$$

$$\sigma_8 = \frac{\sqrt{2}\,\sigma_{10}}{4}$$

$$\sigma_9 = \frac{\sqrt{2}\,\sigma_{11}}{4}$$

$$\sigma_{10} = \sin(\phi_2 + \phi_4 - \phi_5)$$

$$\sigma_{11} = \sin(\phi_2 + \phi_4 + \phi_5)$$

ژاکوبین نسبت به مبدا به دلیل طویل بودن ذکر نخواهد شد اما تمامی مقادیر اشاعه سرعت و ژاکوبین و معکوس آنها در پیوست V_W.mlx آورده شده است.

۲. فضای کاری ربات

برای رسم فضای کاری چندین روش وجود دارد. یک روش این است که فضا را به صورت ابر نقاط در نظر گرفته و به کمک سینماتیک معکوس بررسی کنیم که آیا این نقطه توسط مجری نهایی قابل دستیابی هست یا خیر. اما روش دیگری که ما استفاده کردیم، این است که ابر نقاط را در فضای مفصلی تعریف کنیم و به کمک سینماتیک مستقیم، موقعیت مجری نهایی را ترسیم کنیم. نکته ی حائز اهمیت آن است که ابر نقاط فضای مفصلی نباید از حدود مجاز مفصل ها تجاوز کند.

در این جا محدوده تغییرات مفاصل به شرح زیر است.

```
q1 = linspace(0, 2000, 100);
q2 = linspace(-0.92, 0.92, 30);
q3 = linspace(0, 500, 25);
q4 = linspace(0, 2*pi, 10);
q5 = linspace(0, 2*pi, 10);
```

شکل 5- محدوده تغییرات متغیر های مفصلی

و کد مربوط به استفاده از سینماتیک مستقیم به صورت زیر می‌باشد.

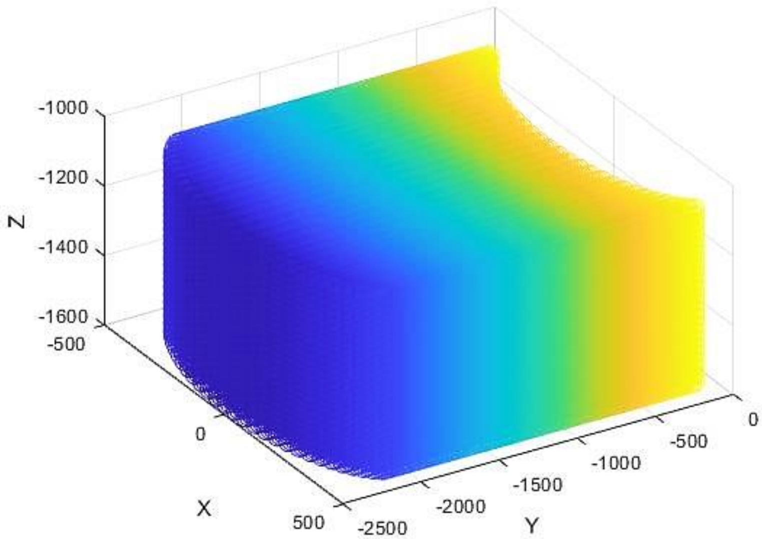
```
X = [];
Y = [];
Z = [];

index = 0;

for i_q1 = 1:size(q1,2)
    for i_q2 = 1:size(q2,2)
        for i_q3 = 1:size(q3,2)
            for i_q4 = 1:size(q4,2)
                for i_q5 = 1:size(q5,2)
                    index = index + 1;
                    [X(index), Y(index), Z(index)] = forward_kin(q1(i_q1),q2(i_q2),q3(i_q3),q4(i_q4),q5(i_q5));
                end
            end
        end
    end
end
```

شکل 6- نحوه استفاده از سینماتیک مستقیم

در نهایت خروجی بدست آمده برای ربات ما به صورت زیر می‌باشد.



شکل 7- فضای کاری رسم شده با متلب

۳. مدل دینامیکی ربات

یکی از مهم ترین پیش نیاز ها برای بدست آوردن مدل دینامیکی سیستم، ترم های P، Pc و I هر لینک است. بردار های P براساس ماتریس های بدست آمده در بخش سینماتیک محاسبه شده‌اند. با توجه به تفاوت جایگیری مختصات هر لینک نسبت به جایگیری مورد استفاده در مقاله، ابتدا نیاز بود تا مقادیر داده شده‌ی مقاله متناسب با داده های ما شوند. در نهایت مقادیر بدست آمده‌ی متناسب با تعریف شماتیک ما به صورت زیر بدست آمده اند(نحوه بدست آمدن هر کدام از این ها در pre_dynamics.mlx که پیوست این گزارش شده اس، قابل مشاهده می‌باشد)

Ic0 =

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Ic1 =

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

```
Ic2 =
```

$$\begin{bmatrix} 61.0080 & -3.2917e+06 & 15.8580 \\ 5.0952e+06 & -61.0080 & 1.0191e+03 \\ 1.0191e+03 & -15.8580 & 2.3677e+06 \end{bmatrix}$$

```
Ic3 =
```

$$\begin{bmatrix} 1.4760e+06 & -3.9700e+03 & -7.4820e+05 \\ -3.9700e+03 & 2.3665e+06 & -8.3272e+03 \\ -7.4820e+05 & -8.3272e+03 & 9.1197e+05 \end{bmatrix}$$

```
Ic4 =
```

$$\begin{bmatrix} 1.4227e+05 & -0.1140 & 0.0020 \\ -0.0820 & -697.1769 & -4.4825e+04 \\ -0.0792 & 6.4360e+04 & 1.0988e+05 \end{bmatrix}$$

```
Ic5 =
```

$$\begin{bmatrix} 27.3420 & 3.6160e+04 & -1.8452e+04 \\ 8.4551e+04 & -1.3029e+04 & 6.6522e+04 \\ -8.4552e+04 & -1.3067e+04 & 6.6523e+04 \end{bmatrix}$$

```
Pc0 =
```

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

```
Pc1 =
```

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

```
Pc2 =
```

$$\begin{bmatrix} 0 \\ 0 \\ -44.1820 \end{bmatrix}$$

```
Pc3 =
```

$$\begin{bmatrix} 145.4370 \\ -1.3060 \\ -848.9250 \end{bmatrix}$$

Pc4 =

$$\begin{bmatrix} 0 \\ 285.9908 \\ -376.0564 \end{bmatrix}$$

Pc5 =

$$\begin{bmatrix} -52.9090 \\ -184.7253 \\ -184.7253 \end{bmatrix}$$

مقادیر جرمی نیز بر اساس مقاله و شماتیک ربات به صورت زیر است:

$M = [0 \quad 83.0250 \quad 21.3330 \quad 8.0220 \quad 9.1200]$

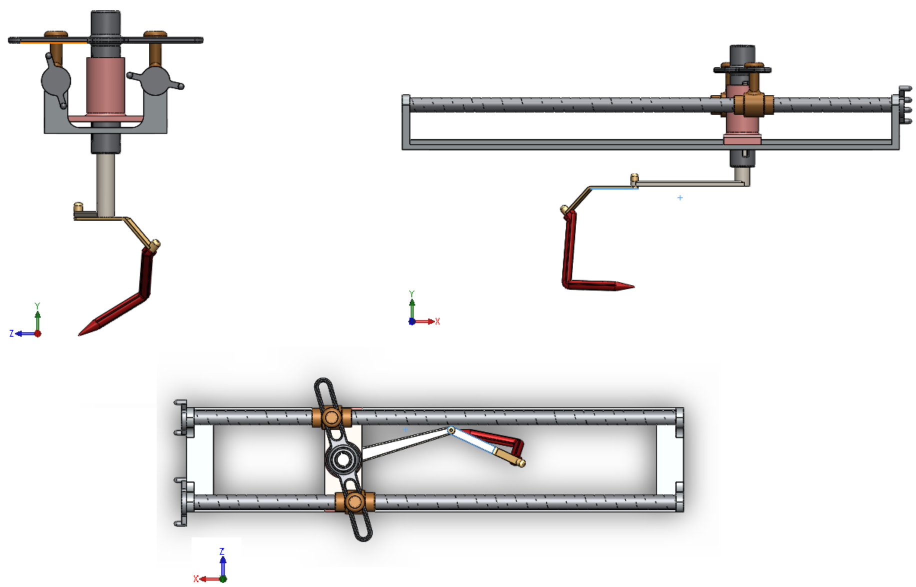
معادلات دینامیک سیستم ابتدا به صورت outward محاسبه شده و همه این ها در ماتریس های v، ω و مشتق های اول و دوم هر کدام و F و N هر لینک محاسبه شد. سپس به صورت inward نیز معادلات گشتاور و نیروی هر مفصل نیز محاسبه گردید و در نهایت با استفاده از اینها توانستیم به ماتریس τ سیستم برسیم. برای یافتن ماتریس های M، G و V و C نیز از دستور horner(که در live script متلب 2020 اضافه شده و کار آن فاکتور گیری است) استفاده کردیم. ژاکوبین مشتقات دوم که با استفاده از همین دستور بدست آمده‌اند برابر با ماتریس M می‌باشد. برای محاسبه ماتریس G بقیه مشتقات اول و دوم صفر شدند و در نهایت ماتریس V از کم کردن ماتریس های قبلی از ماتریس اصلی τ بدست آمده‌اند.

تمامی معادلات ذکر شده‌ی بالا در فایل پیوست Dynamic.mlx محاسبه شده‌اند.(به دلیل طویل بودن معادلات، امکان نمایش خروجی های بدست آمده در گزارش وجود ندارد)

شبیه سازی سیستم و طراحی کنترل کننده:

۱. مدل مکانیکی

با استفاده از برنامه Solid Works، در این بخش ما یک نمای مکانیکی کاملا ساده از ربات طراحی کردیم. ابتدا هر قطعه بصورت جداگانه در فرمت part ایجاد شد و سپس همه به هم اسمبل گردید. شکل زیر نمایی از سیستم اسمبل شده می‌باشد.(تمامی فایل های مربوط به این در فولدر SLD پیوست گزارش شده‌اند)



شکل8.1- شمای مکانیکی شبیه سازی شده‌ی ربات در Solid Works

شکل 8.2- شمای مکانیکی شبیه سازی شده‌ی ربات در Solid Works

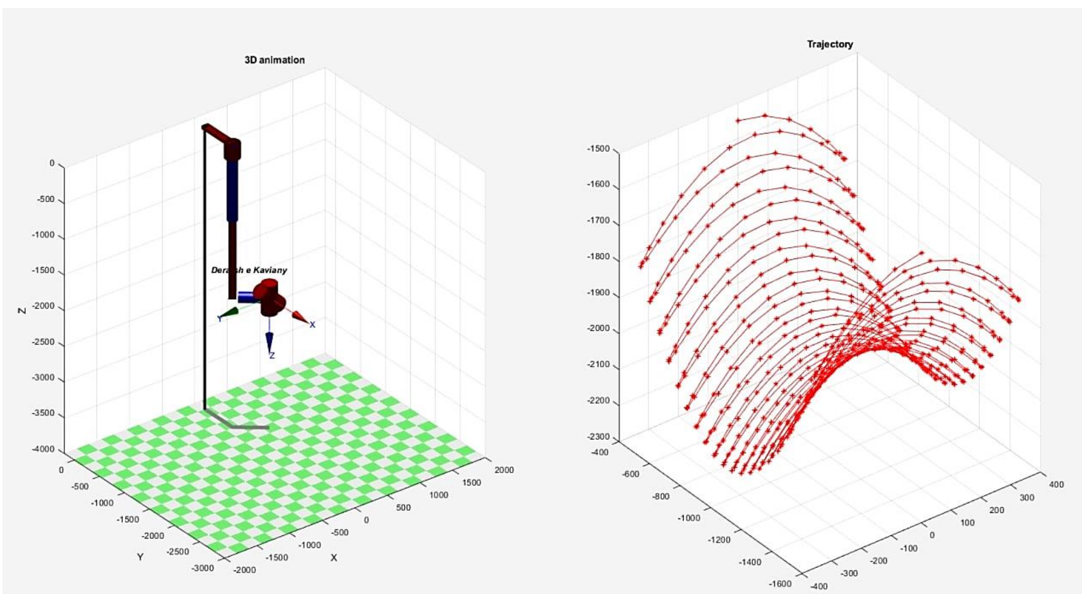
۲. جعبه ابزار RVC

ابتدا با استفاده از ابزار RVC، نمایی ظاهری از ربات را با استفاده از پارامتر های DH بهبود یافته ایجاد کردیم.

```
l(1) = Link('prismatic', 'theta', 0, 'a', 0, 'alpha', pi/2,'modified');
l(2) = Link('revolute', 'd', 0, 'a', 0, 'alpha', pi/2,'modified');
l(3) = Link('prismatic', 'theta', pi/2, 'a', a(3), 'alpha', 0,'offset',465+50+2*285,'modified');
l(4) = Link('revolute', 'd', 0, 'a', a(4), 'alpha', alpha(4),'modified');
l(5) = Link('revolute', 'd', 0, 'a', a(5), 'alpha', alpha(5),'modified');
l(6) = Link([0,0,0,-pi/4],'modified');
R = SerialLink(l,'name', 'Derafsh e Kaviany')
```

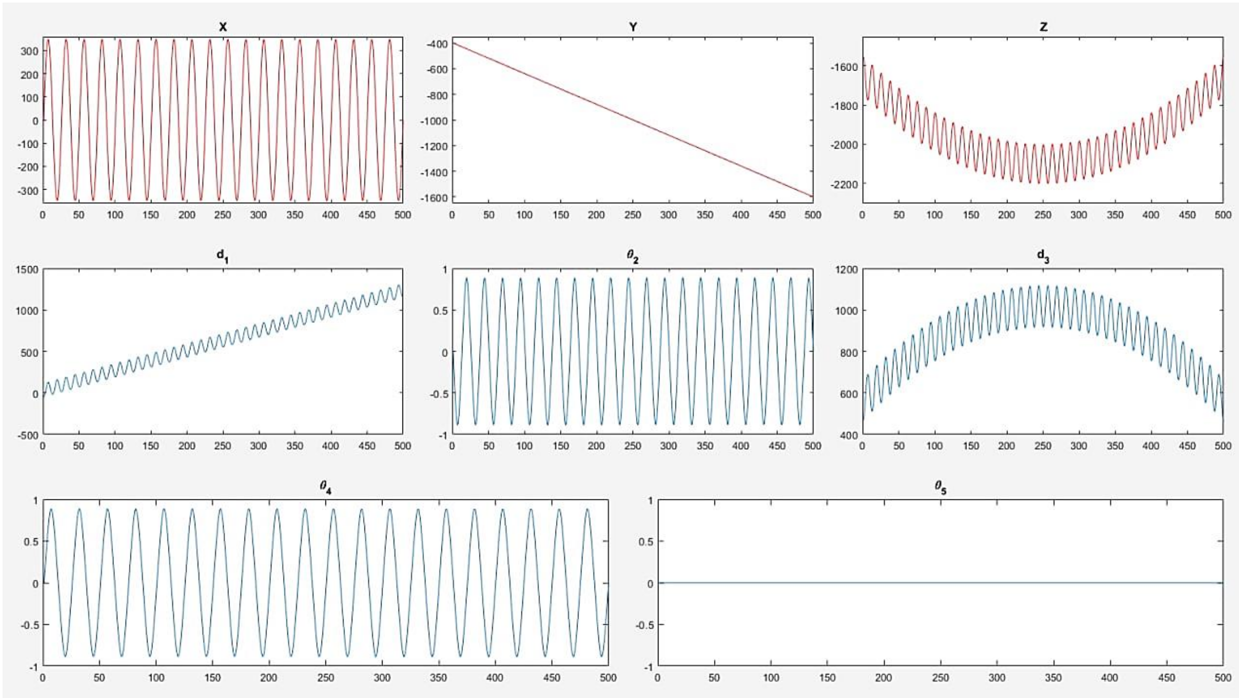
شکل 9- کد تعریف ربات در RVC

حال به کمک سینماتیک معکوسی بدست آمده، نقاط مسیر مرجع را به نقاط مطلوب مفاصل تبدیل کردیم. سپس این مجموعه نقاط به کمک rvc به ربات اعمال گردید. سینماتیک معکوسی که در ابزار RVC موجود است نتوانست مقدار مفاصل ما را محاسبه کند. از این رو از سینماتیک معکوس محاسبه شده در قسمت‌های قبل استفاده کردیم.



شکل 10- شبیه سازی سه بعدی با RVC

علاوه بر نمایش سه بعدی ربات، موقعیت مجری نهایی و مقادیر هر مفصل را به صورت جداگانه رسم کردیم:



شکل 11- نمایی از موقعیت مجری نهایی و زوایای هر مفصل

۳. مسیر مرجع مناسب

دو ترجکتوری برای ربات در نظر گرفته شده است. ترجکتوری اول یک شکل لیساژوی سه بعدی و ترجکتوری دوم یک رویه‌ی زین اسبی است و زمان طی مسیر نیز ده ثانیه در نظر گرفته شده است.

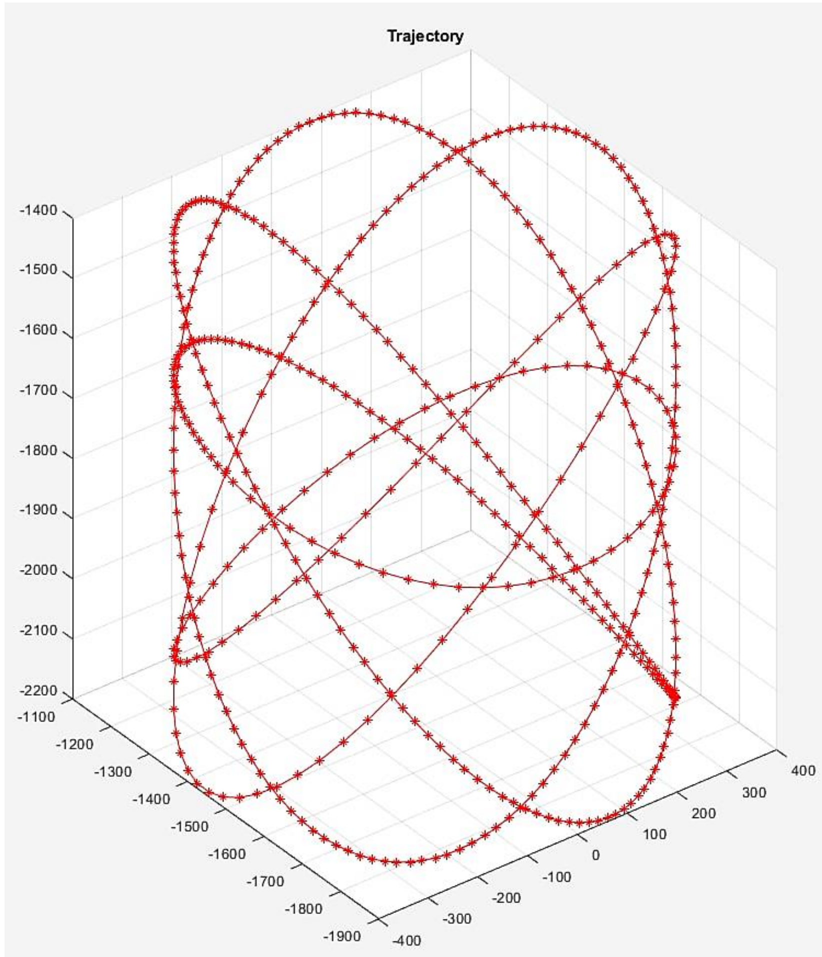
۳.۱. لیساژو:

معادله ی پارامتری مسیر بر حسب زمان به صورت زیر است:

$x = 400sin(5time)$

$y = 400cos(5time) - 1500$

$z = 400sin(6time) - 1800$



شکل 12– مسیر مرجع لیساژو

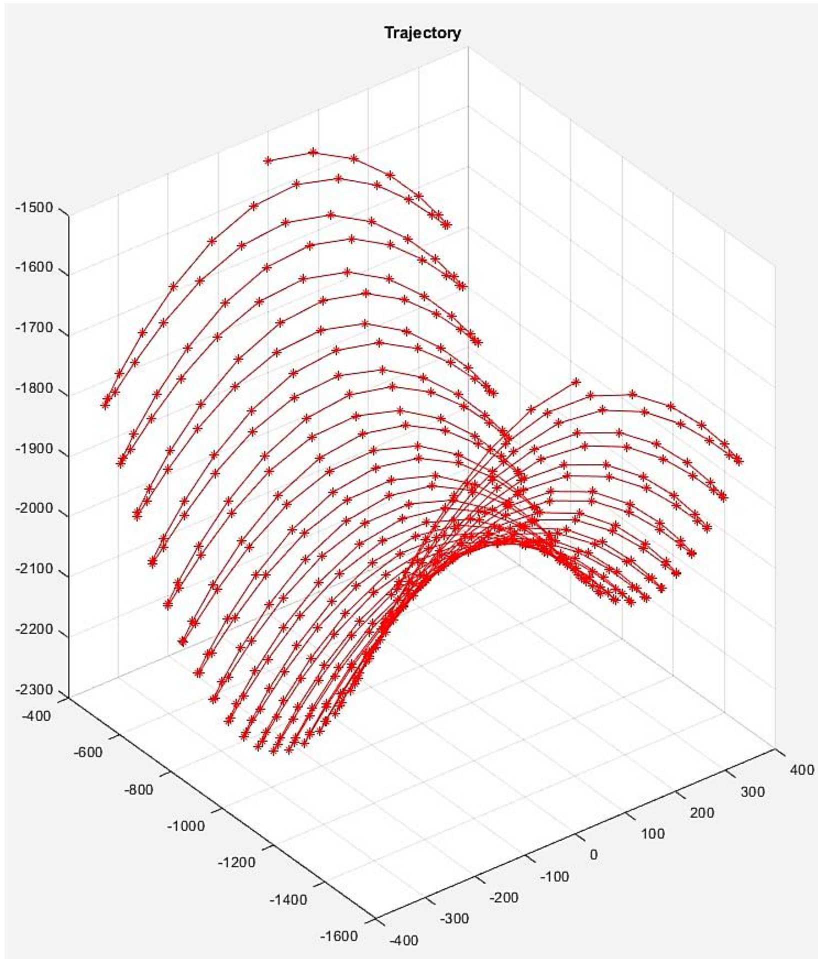
۳.۲. رویه‌ی زین اسبی:

معادله ی پارامتری مسیر بر حسب زمان به صورت زیر است:

$$x = 350sin\left(10\,{}^{\pi}/_{3}\,time\right)$$

$$y = -100time - 400$$

$$z = -\frac{\frac{x^2}{150} - (y+1000)^2}{4} - 2000$$



شکل 13- مسیر مرجع زین اسبی

۴. کنترل کننده

۴.۱. کنترل کننده خطی

می‌دانیم که عملگرهای ربات (موتورهای مفاصل) دارای جعبه دنده هستند تا گشتاور را تقویت و سرعت را کم کنند. در رابطه‌ی دینامیکی موتور و گشتاور بار، اثر لختی بار به کمک جعبه دنده کاهش می‌یابد به طوری که می‌توانیم از اثر گشتاور بار صرف نظر و تغییرات آن را به صورت اغتشاش مدل کنیم.

در این حالت فقط موقعیت موتور کنترل می‌شود. از آنجا که رابطه‌ی موتور یک رابطه‌ی خطی با دو قطب است، یک کنترلر PID می‌توانیم سرعت و خطای مناسب را فراهم کند.

J = ۰.۰۱

b = ۰.۱

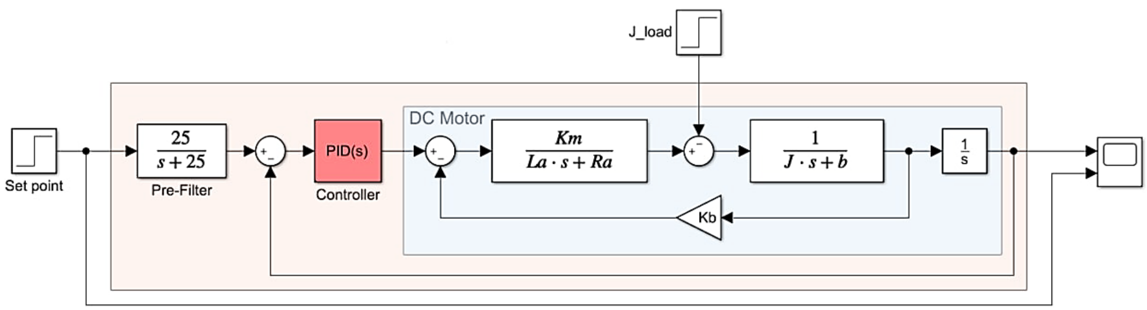
K_m = ۰.۰۱

K_b = ۰.۰۱

R_a = ۱

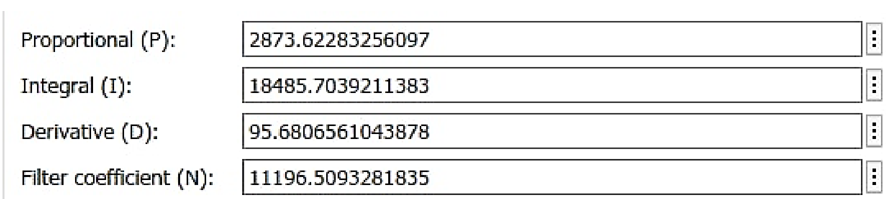
L_a = ۰

بلوک دیاگرام موتور DC و کنترلر آن به همراه یک پیش فیلتر به صورت زیر است:



شکل 14- بلوک دیاگرام موتور DC فرض شده برای هر مفصل

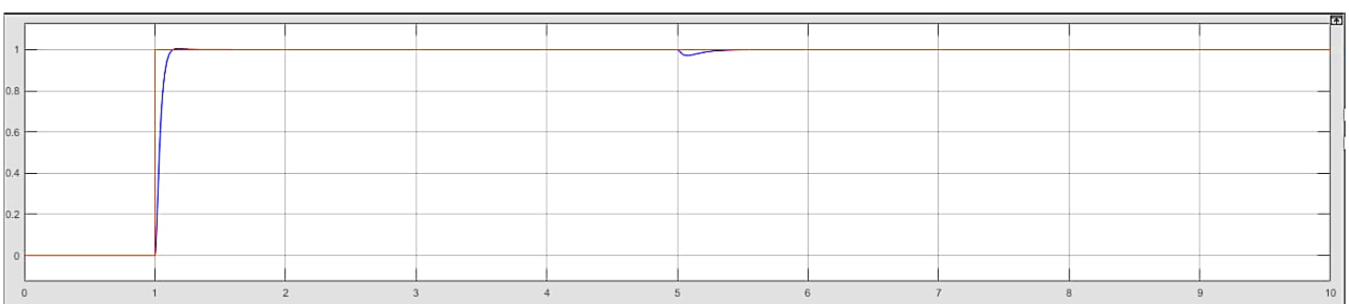
ضرایب PID به صورت زیر محاسبه شده اند:



$$P + I\frac{1}{s} + D\frac{N}{1+N\frac{1}{s}}$$

شکل 15- ضرایب بلوک PID

نتیجه‌ی شبیه‌سازی این کنترلر در حضور یک اغتشاش پله با دامنه‌ی ۱ نیوتون متر در لحظه‌ی t=۵ به صورت زیر است:



شکل 16- خروجی سیستم همراه با کنترل کننده خطی

از آنجا که تمام موتورهای ربات از یک نوع هستند، این کنترلر برای کنترل ربات کافی است.

۴.۲. کنترل کننده غیر خطی

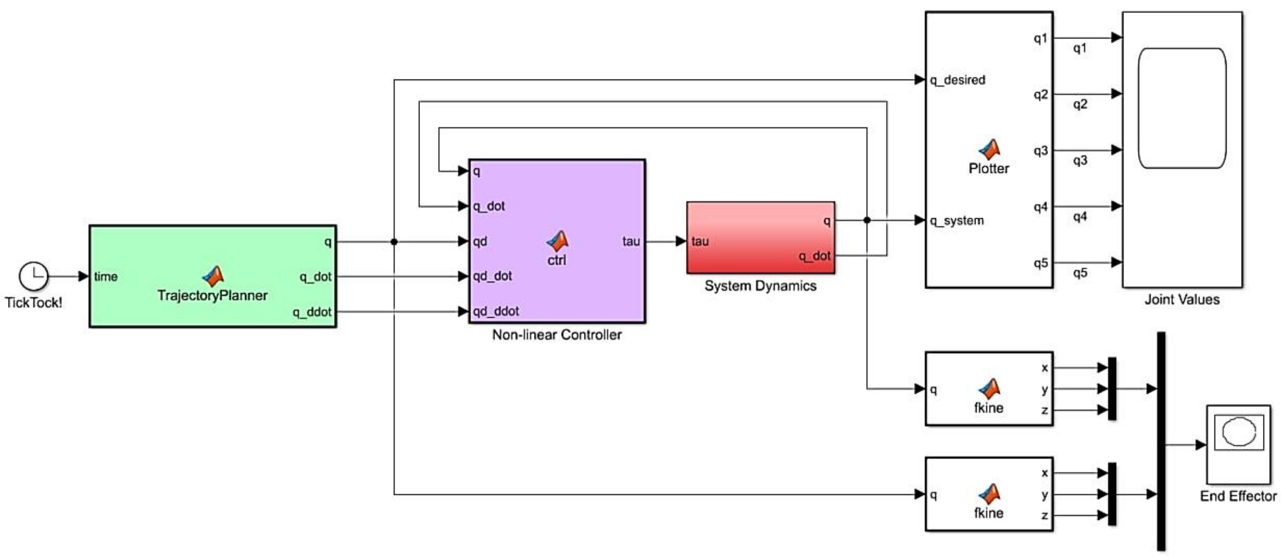
مدل دینامیکی ربات، یک رابطه‌ی غیرخطی است. برای کنترل این سیستم از یک کنترلر غیرخطی با روش دینامیک معکوس بهره بردیم. روش دینامیک معکوس خطای حالت دائم را صفر می‌کند اما نیاز به مدل بسیار دقیق سیستم دارد. از آنجا که در عمل شناسایی دقیق سیستم ممکن نیست و همواره خطای شناسایی و نایقینی وجود دارد، بهتر از روش های تطبیقی یا اسلایدینگ مود استفاده شود. اما در این پروژه، دینامیک معکوس نیاز ما را پاسخ می‌دهد.

رابطه‌ی خروجی کنترلر که گشتاور موتور مفاصل است، از روابط زیر به دست می‌آید:

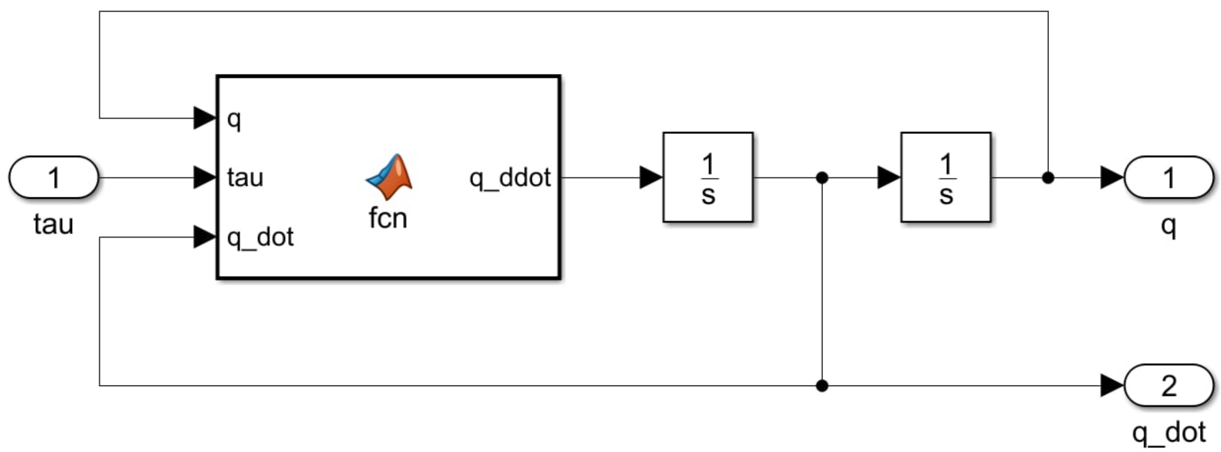
$$a = \ddot{q}_d + 10(\dot{q}_d - \dot{q}) + 25(q_d - q)$$

$$\tau = Ma + C\dot{q} + G$$

شبیه‌سازی این کنترلر در سیمولینک به صورت زیر است:

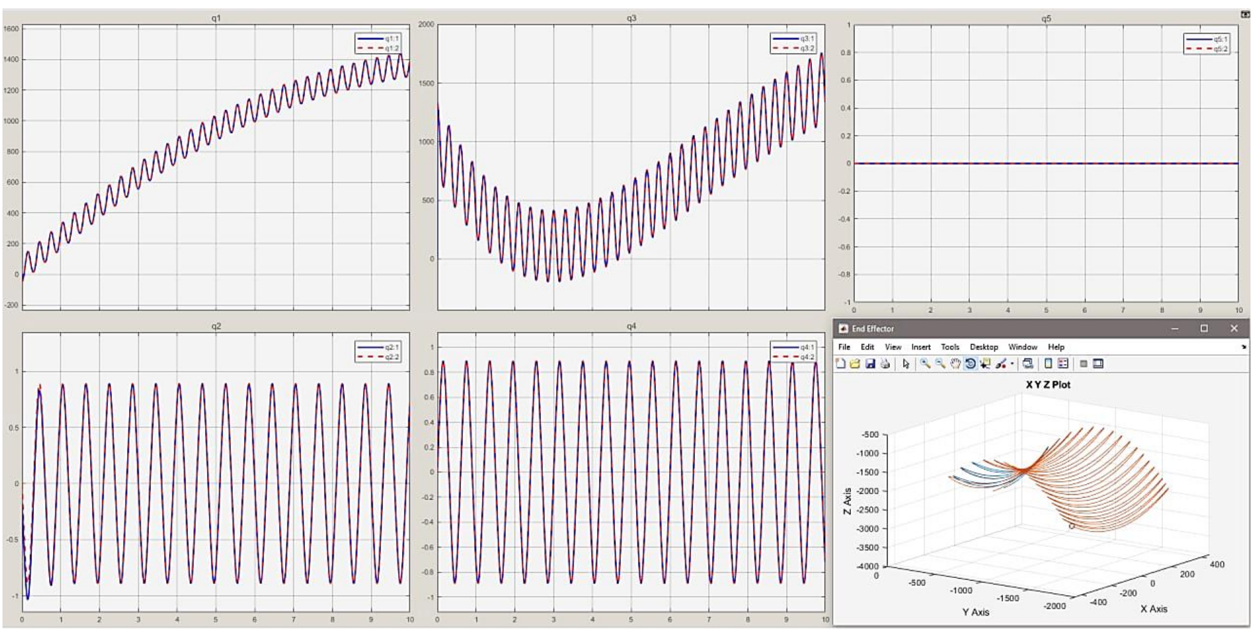


شکل 17- بلوک دیاگرام کلی سیستم



شکل 18- زیر سیستم دینامیک ربات

بلوک برنامه‌ریز مسیر، وظیفه‌ی تولید ست پوینت در لحظه‌های مختلف را دارد. مسیری که در این شبیه‌سازی به ربات داده شده، یک رویه‌ی زین اسبی است. نتیجه کنترلر به صورت شکل زیر است:



شکل 19- نتایج بدست آمده از کنترلر غیر خطی

خطوط آبی رنگ، مقادیر مرجع هستند که باید توسط مفاصل تعقیب شوند. خطوط قرمز رنگ مقدار واقعی هر مفصل است. همانطور که مشاده می‌شود این کنترلر با سرعت بسیار زیاد خطا را به صفر کاهش می‌دهد و سرعت و دقت بسیار بالایی دارد.

منحنی پایین سمت راست، نمای سه بعدی مجری نهایی است که در آن منحنی آبی رنگ محل واقعی مجری نهایی و منحنی قرمز رنگ ست پوینت مجری نهایی است. در این شکل نیز انطباق این دو منحنی پس از گذشت زمان بسیار کوتاهی قابل مشاهده است.

جمع بندی و نتیجه گیری

رباتی که در این گزارش بررسی شد، دارای دو قسمت موازی و سری بود که قسمت موازی با قوانین هندسی بصورت سری مدل شد. در ادامه سینماتیک مستقیم و معکوس ربات با توجه به شماتیک ساده شده توسط live script بدست آمد و درباره جواب‌های محتمل بحث گردید. فای کاری سیستم بدست آمد و پس از آن با استفاده از کد های livescript ، ژاکوبین و مدل دینامیکی سیستم تولید گردید. با استفاده از مدل موتور های مورد استفاده در ربات، کنترل خطی سیستم تقریب زده شد و با استفاده از مدل دینامیکی بدست آمده در قبل، یک کنترلر غیر خطی نیز پیاده گردید.

از مشکلات ما در این پروژه، استفاده از جعبه ابزاری نسبتا قدیمی اما با امکانات بالا (RVC) بود که هنگام پیاده سازی آن، بشدت سردرگم شدیم. علاوه بر آن، متفاوت بودن محل قرار گیری مختصات های الصاق شده‌ی ما نسبت به مقاله بود که باعث شد تا تمام معادلات ما متفاوت از مقاله باشد و امکان بررسی صحت آنها وجود نداشت و علاوه بر آن نیاز شد تا ماتریس های P و Pc و I هر لینک براساس متخصات های جدید بازنویسی شوند. در کنار همه این ها مزیتی که الصاق مختصات ما نسبت به مقاله داشت، ساده تر شدن معادلات بدست آمده بود؛ بطوریکه در حالت اول امکان اجرای کد دینامیک و ژاکوبین روی آنها وجود نداشت اما بعد از اینکه به این شماتیک رسیدیم، سیستم خیلی ساده تر شد و توانستیم خروجی های دینامیکی را بدست آوریم.

از اهداف آینده‌ی ما در این بخش، تولید یک جعبه ابزار واحد برای متلب(MATLAB Toolbox) توسط کد های تولید شده در طی این پروژه است تا بتوان به آسانی تمامی ویژگی های ربات های سری را مورد بررسی قرار داد.

منابع

1. Kinematics, dynamics, and controlsystem of a new 5-degree-of-freedomhybrid robot manipulator, By: Wanjin Guo , Ruifeng Li , Chuqing Cao , and Yunfeng Gao, DOI: 10.1177/1687814016680309
2. https://www.mdpi.com/2076-3417/9/10/2033/htm
3. https://petercorke.com/toolboxes/robotics-toolbox/
4. https://www.youtube.com/watch?v=440-3LCZ_ow
5. https://youtu.be/MIDNF7EvHWs
6. https://www.youtube.com/watch?v=XiHigWht7-E
7. https://youtu.be/5DnKot3mMSc