

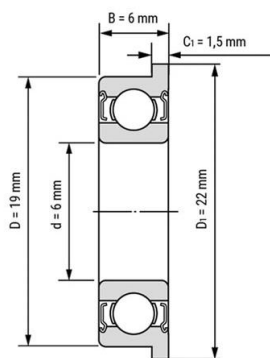
پیاده سازی و کنترل ربات سه درجه آزادی با الگوبرداری از ربات Viper 650

مصطفی حمیدی فرد، مهشاد علیان، فاطمه کرار، سینا رشیدی زاده، محمدصادق نجفی، مریم حیدری

دانشکده مهندسی برق، دانشگاه صنعتی امیرکبیر، تهران، ایران

با استفاده از آن لینک ها و نحوه قرار گیری موتورها در نرم افزار مشخص شده است که در نهایت با استفاده از فایل STL خروجی گرفته شده از سالیدورکس و بهره گیری از نرم افزار کورا^۵ قطعات ربات را پرینت کردیم.

در این ربات از سه سروو موتور^۶ استفاده شده است. در ادامه به منظور اتصال اجزای ربات از سه بلبرینگ شیار عمیق مدل ZZ۶۲۶، کوپلینگ سخت 6mm x 6mm و شفت استفاده کردیم.



شکل ۲- شماتیک بلبرینگ شیار عمیق

شفت‌های استفاده شده در ربات ۶ میلی‌متر ضخامت و ۸ سانتی‌متر طول دارند.

شکل نهایی ربات ساخته شده به صورت زیر می‌باشد:



شکل ۳- ربات در پیکربندی صفر

چکیده— در این مقاله، یک ربات سه درجه آزادی طراحی و پیاده سازی شده است. در ادامه مراحل ساخت و مونتاژ ربات توضیح داده شده و کنترل ربات با استفاده از آردوینو^۱ و متلب^۲ انجام می‌شود. در نهایت راهکارهای مناسب برای کنترل و راه حل‌های پیش رو در جهت رفع چالش‌ها ارائه شده است.

کلمات کلیدی— ربات، کنترل، آردوینو، متلب، پیاده سازی.

I. مقدمه

Viper 650 یک ربات مفصلی با ۶ درجه آزادی است که در ماشین کاری، مونتاژ و جابجایی مواد استفاده می‌شود. وضوح بالا، استفاده از انگرد^۳ مطلق و دقت بالا، اینرسی کم و بازوی سبک جهت ایجاد حداکثر شتاب، از جمله مزایای این ربات می‌باشد.

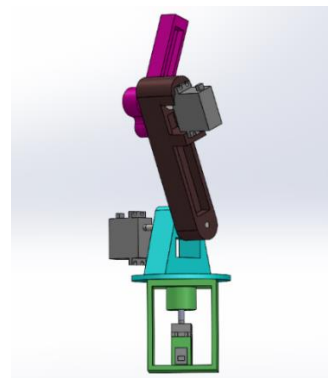
در این مقاله یک ربات سه درجه آزادی با الهام از Viper پیاده سازی کرده ایم که در آن موتور قرار گرفته شده در محل هر یک از مفاصل، نقش چرخاندن لینک‌ها را برعهده دارد.

در ادامه به منظور تحقق هدف ردیابی مسیر دلخواه توسط ربات، به نحوه کنترل موتورها با استفاده از نرم افزار و سایر المان‌های مورد استفاده می‌پردازیم.

در نهایت مشاهده می‌شود که موتورها این قابلیت را دارند که یک مسیر سینوسی را با خطای کم دنبال کنند.

II. ساختار ربات

ما ابتدا یک ربات سه درجه آزادی را در نرم افزار سالیدورکس^۴ طراحی کردیم.



شکل ۱- ربات طراحی شده نهایی در محیط سالیدورکس

⁴ SolidWorks
⁵ Cora
⁶ Servo Motor

¹¹ Arduino
² MATLAB
³ Encoder

$$I_{Link_3} = \begin{bmatrix} 179937.28 & 34003.18 & -94339.01 \\ 34003.18 & 207856.24 & -34453.66 \\ -94339.01 & -34453.66 & 157860.07 \end{bmatrix}$$

شماره لینک	جرم (گرم)	مرکز جرم
۱	۱۶۸.۱۳	$\begin{bmatrix} -2.47 \\ 32.05 \\ -9.31 \end{bmatrix}$
۲	۱۶۷.۸۷	$\begin{bmatrix} 16.87 \\ 116.62 \\ 144.30 \end{bmatrix}$
۳	۴۵.۵۴	$\begin{bmatrix} 43.73 \\ 21.06 \\ -45.00 \end{bmatrix}$

جدول ۱- داده های مورد استفاده در محاسبه دینامیک ربات

با استفاده از ماتریس اینرسی، دینامیک ربات به روش لاگرانژ و از طریق محاسبه ماتریس های C ، G و M در متلب به دست می آید.

این ماتریس ها به عنوان دینامیک ربات ساخته شده در نظر گرفته می شوند.

مطابق شکل ۱۰ زوایای مورد نظر به عنوان ورودی در سیمولینک متلب وارد می شوند. پس از عبور از بلوک دینامیک معکوس، گشتاور ربات از طریق معادله ۱ به دست می آید:

$$\tau = M a_q + C \dot{q} + G$$

معادله ۱

a_q مطابق معادله ۲ به دست می آید.

$$a_q = \ddot{q}_d - K_0(q - q_d) - K_1(\dot{q} - \dot{q}_d)$$

معادله ۲

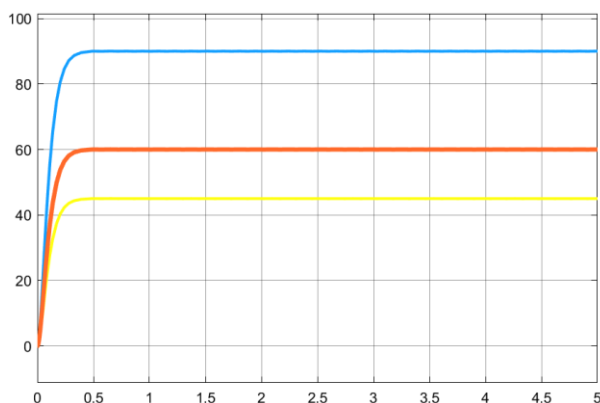
q زاویه مفصل و q_d زاویه مورد نظر ما می باشد. با تغییر ضرایب K_0 و K_1 خروجی کنترل می شود.

در نهایت با استفاده از رابطه ۳ زاویه هر مفصل در خروجی نمایش داده می شود.

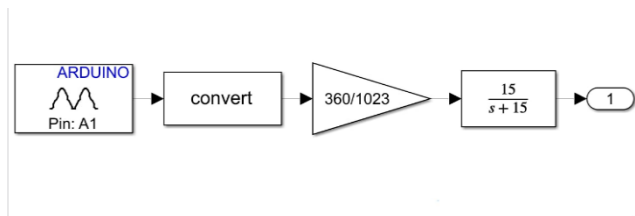
$$\ddot{q} = M^{-1}(\tau - C \dot{q} - G)$$

معادله ۳

خروجی نهایی به صورت زیر می باشد:



شکل ۱۱- نتیجه شبیه سازی



شکل ۸- زیرسیستم فیدبک، مورد استفاده در سیمولینک متلب

در نتیجه، زاویه موتور از سیگنال فیدبک به دست می آید.

در ابتدا سیگنال مورد نظر را که می خواهیم توسط ربات دنبال شود (مانند سیگنال سینوسی) به عنوان ورودی به حلقه کنترل می دهیم.

با استفاده از سیگنال فیدبک خطای زاویه به دست می آید که عدد بدست آمده بعد از عبور از بلوک کنترلر PID به زیرسیستم PWM وارد می شود.

اختلاف زاویه مورد نظر و زاویه موتور جهت چرخش موتور را تعیین می کند.

در نهایت مشاهده می شود که سیگنال ورودی سینوسی ۱ رادیان بر ثانیه به خوبی توسط هر یک از موتورهای دنبال می شود.

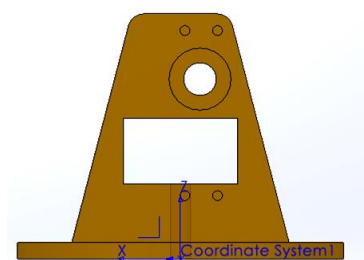
V. کنترل ربات با استفاده از سینماتیک معکوس

با استفاده از سینماتیک معکوس ربات، می توان End Effector را در مسیر دلخواه حرکت داد.

در متلب کد سینماتیک معکوس، زاویه هر مفصل را در هر زمان محاسبه می کند. زوایای محاسبه شده به عنوان ورودی سیمولینک در قسمت قبل در نظر گرفته می شوند.

VI. شبیه سازی و کنترل به روش دینامیک معکوس

در ابتدا در نرم افزار سالدورکس محورهای هریک از لینک ها به صورت جداگانه تعیین می شوند.

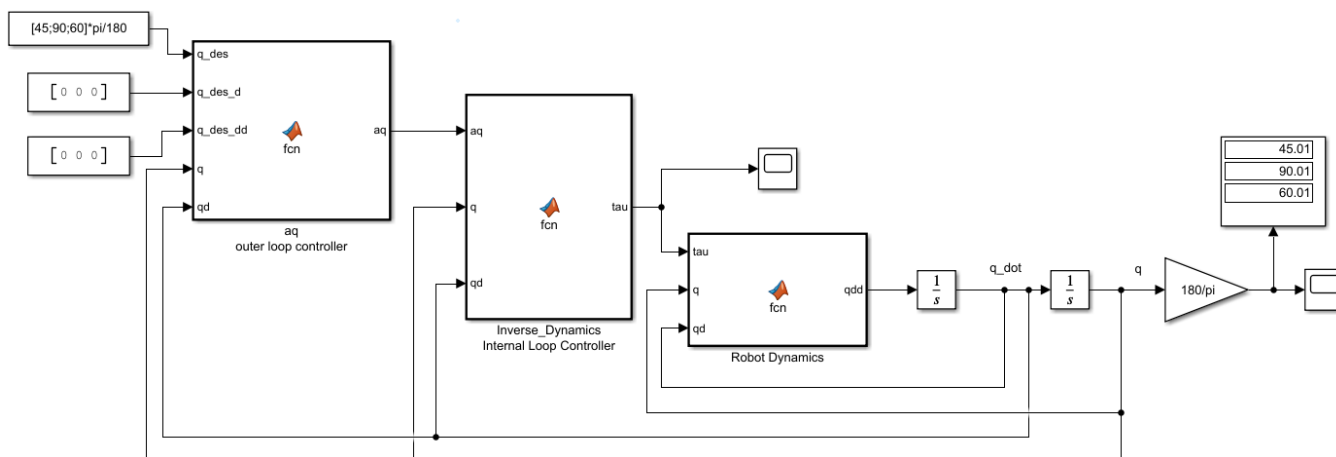


شکل ۹- نمونه تعیین محورهای ربات در سالدورکس

پس از تعیین محورها، ماتریس اینرسی به دست می آید.

$$I_{Link_1} = \begin{bmatrix} 384957.06 & -2691.73 & 43626.55 \\ -2691.73 & 180695.89 & -19024.35 \\ 43626.55 & -19024.35 & 309313.62 \end{bmatrix}$$

$$I_{Link_2} = \begin{bmatrix} 5886595.04 & 330234.57 & 368561.17 \\ 330234.57 & 4128626.67 & 2824947.69 \\ 368561.17 & 2824947.69 & 2826241.18 \end{bmatrix}$$



شکل ۱۰- شبیه سازی ربات در سیمولینک

می گیرد، قبل از شکستن خم می شود و تغییر شکل می دهد، در حالی که PLA به سرعت می شکند.

یکی از چالش های پروژه، وجود سیم ها و نويز پتانسیومتر بود که تاثیر نامطلوبی بر سیگنال فیدبک موتور می گذاشت. جهت رفع این مشکل در سیمولینک متلب، سیگنال فیدبک قبل از ورود به حلقه کنترل از یک فیلتر عبور داده می شود. اگرچه وجود فیلتر باعث تاخیر می شود اما عملکرد کلی ربات را بهبود می بخشد.

منابع

- [1] Mark W. Spong, Seth Hutchinson and M. Vidyasagar, Robot Modeling and Control, 1st ed, Dec 12, 2022.
- [2] John J. Craig, Introduction to Robotics Mechanics and Control, Third Edition, Pearson Education International.

VII. نتیجه گیری

در این پروژه یک ربات را از ابتدا پیاده سازی کردیم و موقعیت موتورهای استفاده شده در ربات را کنترل کردیم تا با دقت بالا و حداقل تاخیر مسیر مورد نظر را ردیابی کنیم.

در هنگام پرینت قطعات، ربات را از جنس فیلامنت ABS چاپ کردیم. از مزایای فیلامنت ABS این است که این ماده قابلیت ذوب شدن و سرد شدن را بدون تغییر زیاد خواص شیمیایی دارا می باشد. بنابراین، این جنس فیلامنت ایده آل برای چاپگرهای سه بعدی در نظر گرفته می شود. این فیلامنت به استحکام، سختی و دوام بالا شهرت دارد. همچنین مقاومت قابل قبولی در برابر خراش های سطحی، حرارت و خوردنده های شیمیایی رایج دارد؛ بنابراین فیلامنت ABS برای چاپ اجسام در معرض استهلاک یا فشار مناسب است. وقتی ABS تحت فشار قرار