**بسم الله الرحمن الرحیم**

**گزارش کار پروژه رباتیک مقدماتی**

**Kuka LWR Robot**

**استاد درس:**

**دکتر خسروی**

**اعضای گروه:**

**امین سرخی لله لو ۹۶۲۳۹۰۱**

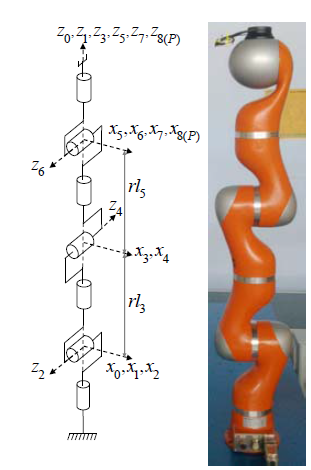
**سپهرقمری ۹۶۲۳۰۹۰**

**محمدمهدی شجاعی فر ۹۶۲۳۰۶۵**

**تیرماه ۱۴۰۰**

۱- از مزایای این ربات دارا بودن هفت درجه ازادی که مانند بازوی انسان عمل میکند و دارای انعطاف پذیری بالایی می باشد و همچنین می توان گشتاور را در تمام مفاصل با استفاده از سنسور های گشتاور اندازه گیری نمود. همچنین این ربات دارای وزن کمی می باشد و در نتیجه باعت استفاده از انرژی کمتر می شود. به دلیل اینکه سه مفصل اخر ربات در موقعیت نهایی تاثیر ندارد، پس می تواند هر جهت گیری ای را اتخاذ کند و هردو فضای کاری ان یکسانند که به دلیل داشتن فضای کاری ماهر بسیار گسترده، بسیار کاربردی است. این ربات در زمینه های مختلفی مانند بازوهای مکانیکی استفاده می شود.

۲- محور های مختصاتی ربات به شکل زیر میباشد:



۳- جدول پارامترهای ربات به شرح زیر است:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| ۱ | ۰ | ۰ | ۰ |  |
| ۲ | ۹۰ | ۰ | ۰ |  |
| ۳ | -۹۰ | ۰ | ۰.۴ |  |
| ۴ | -۹۰ | ۰ | ۰ |  |
| ۵ | ۹۰ | ۰ | ۰.۳۹ |  |
| ۶ | ۹۰ | ۰ | ۰ |  |
| ۷ | -۹۰ | ۰ | ۰ |  |
| ۸ | ۰ | ۰ | ۰ |  |

۴- با استفاده از جدول بالا و فرمول کلی و کد متلب ماتریس های تبدیل زیر به دست می آیند.

۵- برای محاسبه سینماتیک معکوس، با توجه به پیچیده بودن درایه ها، امکان حل پارامتری ان وجود ندارد و به صورت عددی ان را حل کرده ایم. برای اینکار ابتدا ربات را به برنامه شناسانده ایم و موقعیت مجری نهایی نیز به عنوان ورودی به ربات می دهیم. برای محاسبه سینماتیک معکوس از تابع GeneralizedInverseKinematics می توان زاویای مفاصل را با استفاده از موقعیت مجری نهایی به دست اورد. خروجی نهایی این تابع، هفت زاویه برحسب درجه است. برای مثال اگر ورودی تابع ۰.۲، ۰.۵، ۰.۵ باشد، خروجی نهایی که نشان دهند زوایای هرمفصل به درجه است، به صورت زیر است:

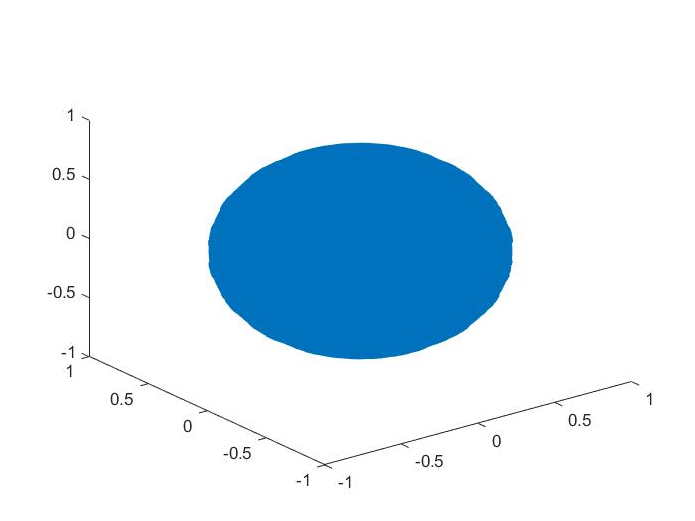
42.0906-, -87.5777, 68.7825, -111.8214, 73.9408, 19.1302, 5.0061

با توجه به اینکه سه مفصل اخر فقط در جهت گیری نهایی تاثیر دارد و در موقعیت مجری نهایی تاثیر ندارد، پس پاسخ های زیادی وجود دارد.

۶- باتوجه به اینکه سه مفصل اخر محورهایشان از یک نقطه عبور میکند در جابجایی مجری نهایی تاثیر ندارد و صرفا جهت گیری ان را مشخص می کند، پس کافی است برای به دست اوردن فضای کاری، صرفا چهار مفصل اول را جابجا کنیم و ابرنقاط را به دست اوریم. با استفاده از اطلاعات موجود در مقالات دیگر که به پیوست اومده است، محدودیت زاویه هر مفصل به شرح زیر است:

|  |  |  |
| --- | --- | --- |
| ماکزیمم درجه | مینیمم درجه | مفصل |
| ۱۷۰ | ۱۷۰- | ۱ |
| ۱۲۰ | ۱۲۰- | ۲ |
| ۱۷۰ | ۱۷۰- | ۳ |
| ۱۲۰ | ۱۲۰- | ۴ |
| ۱۷۰ | ۱۷۰- | ۵ |
| ۸۰ | ۴۵- | ۶ |
| ۱۳۰ | ۴۵- | ۷ |

باتوجه به شکل به دست امده، فضای کاری ربات بیضی گون می باشد.



۷- با توجه به فرمول جزوه، و برای مفصل چرخشی و که ستون سوم ماتریس تبدیل می باشد و المان های ستون چهارم  *است. ماتریس ژاکوبین در زیر اورده شده است و مشتق ان نیز در فایل محاسبه شده است.*

*۸- باتوجه به روش نیوتن-اویلر ابتدا سرعت خطی و شتاب زاویه ای و سپس مشتق سرعت خطی و شتاب زاویه ای را به دست اوردیم :*

سپس نیروها، ممان ها و درنهایت گشتاورهای هرکدام از مفاصل محاسبه شد :

*برای محاسبات این قسمت، مقادیر اولیه زیر درنظر گرفته شد:*

*جرم مفاصل:*

*مقادیر اولیه سرعت ها:*

*مقادیر اولیه مشتق سرعت ها:*

*نیرو و گشتاور وارده به مجری نهایی:*

*ماتریس تنسور اینرسی از مقاله برداشته شده است:*

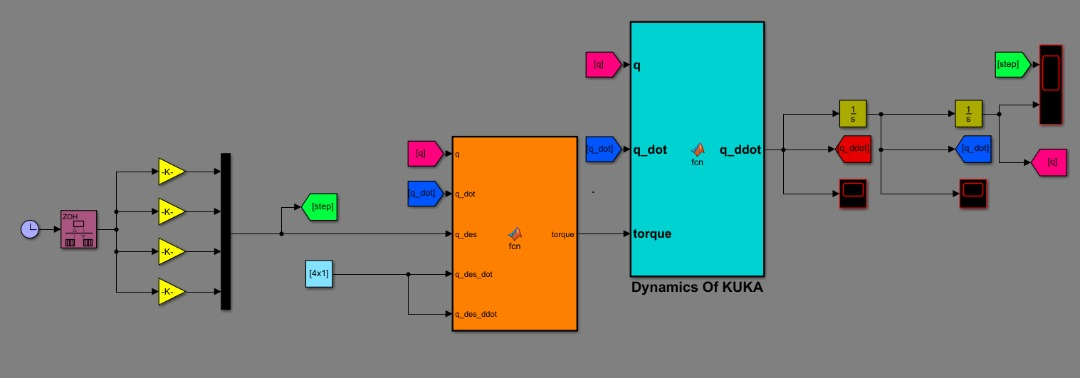
*مختصات مرکز جرم نیز از مقاله برداشته شده است:*

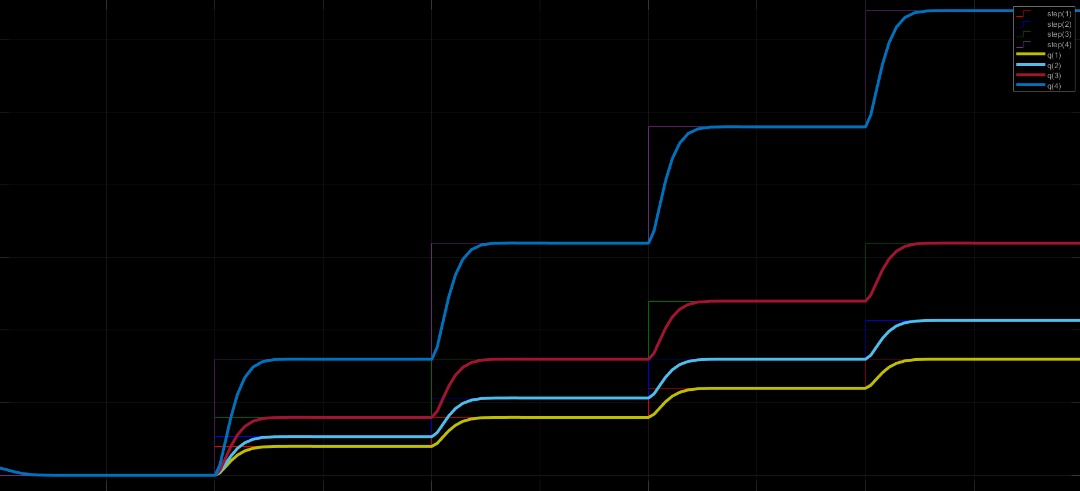
با استفاده از توابع نوشته شده در که در پوشه قرار داده شده است، گشتاور همه مفاصل محاسبه شده که نتایج محاسبات در پوشه قرارداده شده است.

۹- دراین قسمت برای کنترل ربات با استفاده از مدل دینامیکی به دست امده از مرحله قبل، کنترلر مناسب در محیط طراحی شد:

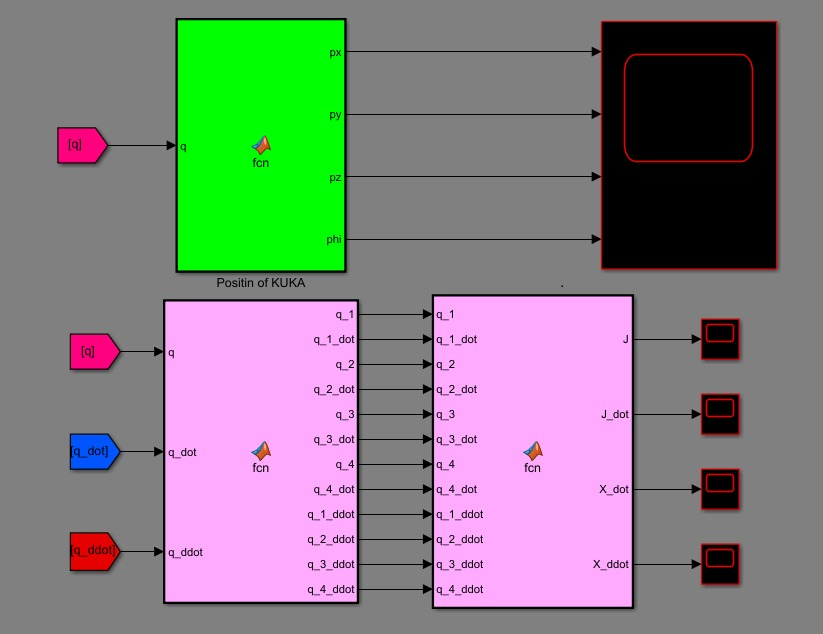
با توجه به محاسبات زیاد برای این قسمت و هماهنگی های صورت گرفته با تدریس یار، کنترلر این قسمت برای چهار مفصل ابتدایی طراحی شده است. کنترلر استفاده شده از نوع است و ضرایب و به ترتیب برابر و انتخاب شد. نتایج شبیه سازی به شرح زیر است:

همانطور که مشاهده می شود،  **و خطای حالت دائم نداریم و نیز حدود یک ثانیه** است. تصاویر این شبیه سازی در زیر موجود است.





۱۰ و ۱۱- برای صحت سنجی سینماتیک مستقیم، شبیه سازی در محیط سیمولینک به صورت زیر انجام گرفته است.



نتایج شبیه سازی به شرح زیر می باشد:



**Phi**



**end effector position**



**x dot**



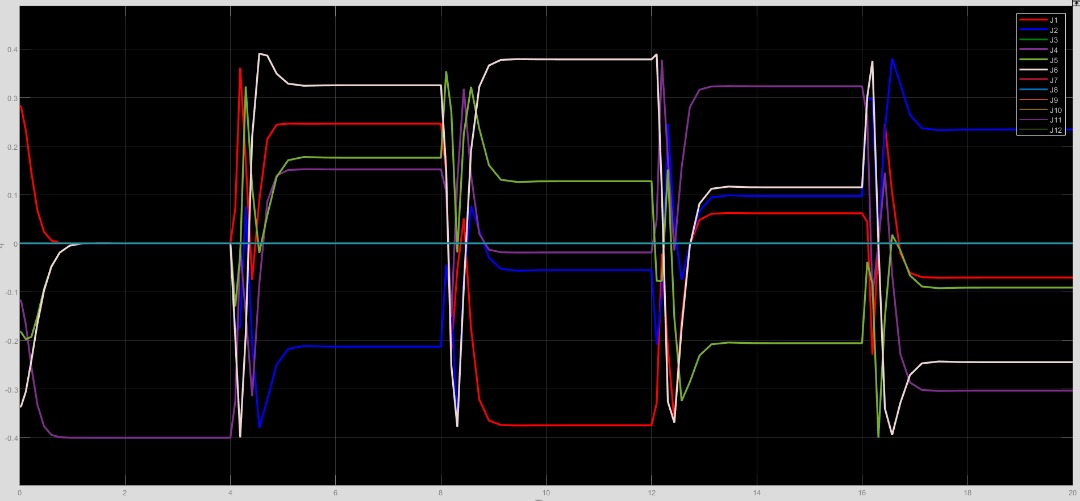
**W**



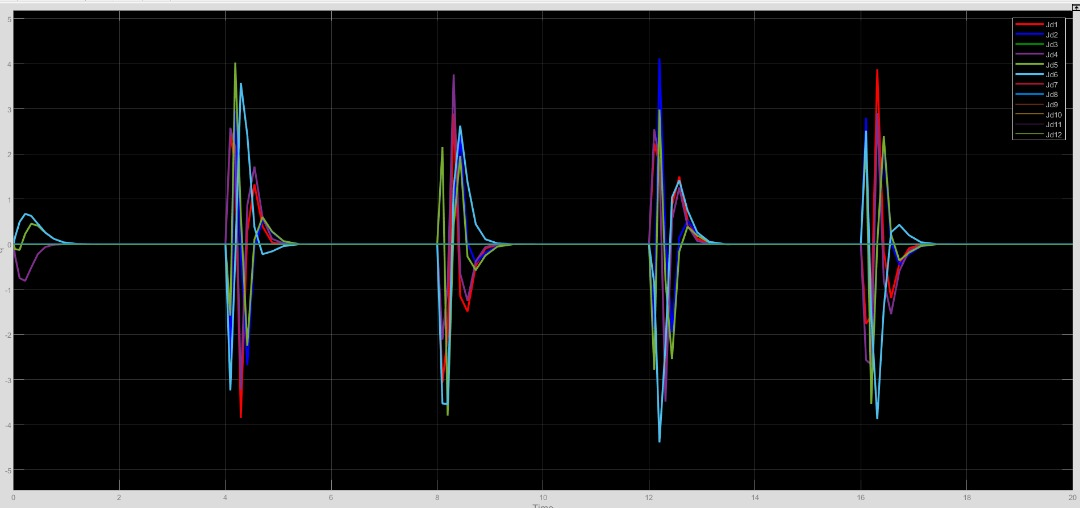
**x ddot**



w dot



**Jaccobian**



**Jaccobian dot**

باتوجه به اینکه با استفاده از کنترلر، هرکدام از زوایا به مقادیر مشخص شده رسیدند و پایدار شده اند، پس مجری نهایی نیز باید طبق سینماتیک مستقیم به موقعیت مشخصی براساس همان زوایا برسد. همانطور که مشاهده می شود، در طی دوره های مختلف برای هرکدام از پله ها، موقعیت مجری نهایی در مقادیر خاصی پایدار شده است. از طرفی سرعت خطی بعد از رسیدن به موقعیت نهایی باید صفر شود که این اتفاق افتاده است. همچنین تمامی درایه های مشتق ژاکوبین هم به صفر میل کرده است.

برای مثال در دوره اول(صفر تا چهار ثانیه) تمامی زوایا و در نتیجه مجموع انها برابر صفر است که باتوجه به طرز قرارگرفتن مفاصل در حالت اولیه، موقعیت نهایی ربات باید طوری باشد که فقط در راستای محور مقدار داشته باشد(ربات باید عمود بر صفحه باشد). همانطور که در نمودار مشاهده می شود، و ربات به صفر میل کرده و ربات به ۰.۴ رسیده که مطابق با انتظار است.

در قسمت ۱۱، موقعیت، سرعت و شتاب خطی و سرعت و شتاب زاویه ای خواست شده است. با توجه به توضیحات بالا، نمودار نشان دهنده موقعیت لحظه ای است.

برای به دست اوردن سرعت خطی و زاویه ای نیز کافی است ماتریس ژاکوبین( ) را در بردار مشتق فضای مفصلی ضرب کنیم. سه درایه اول نشان دهنده سرعت خطی و سه درایه بعد نشان دهنده سرعت زاویه ای است که نمودار های و نشان دهنده ان است. برای به دست اوردن شتاب خطی و زاویه ای کافی است از سرعت خطی مشتق بگیریم که نمودار ان نیز و است.

۱۲- با استفاده از پارامترهای دناویت هارتنبرگ بهبودیافته و نوع هرلینک(چرخشی یا انتقالی) را مشخص میکنیم. سپس با استفاده از دستور این مفاصل را به هم مرتبط و ربات را شبیه سازی میکنیم و با استفاده از دستور ربات را به صورت سه بعدی نمایش می دهیم که به صورت زیر است:

