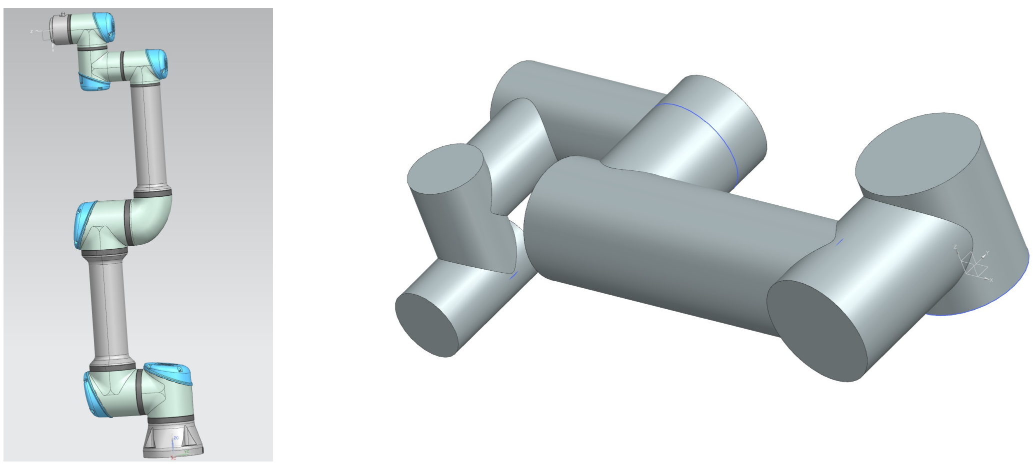
پروژه‌ی درس مقدمه‌ای بر رباتیک

فروغ افخمی اردکانی، علی بهمنیار، سینا ربیعی، سمیرا سلجوقی

خرداد ماه ۱۴۰۱

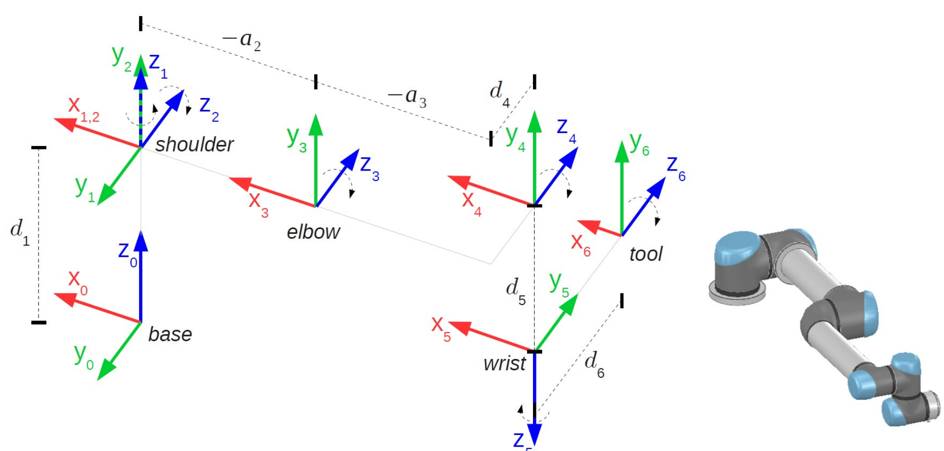
۱ بررسی مشخصات ربات



شکل ۱: طراحی مدل سه بعدی ربات در نرم افزار NX و محاسبه‌ی مشخصات ربات از روی آن

۲ به دست آوردن پارامترهای دناویت-هارتنبرگ

ابتدا با توجه به دناویت هارتنبرگ اصلاح شده (Modified) فریم‌ها را به مفصل‌های ربات اختصاص می دهیم و سپس جدول مربوطه را تکمیل می کنیم:



شکل ۲: قرارگیری محورهای دناویت-هارتنبرگ بر روی ربات

i	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0	d_1	θ_1
2	0	90	0	θ_2
3	a_2	0	0	θ_3
4	a_3	0	d_4	θ_4
5	0	90	d_5	θ_5
6	0	-90	d_6	θ_6

جدول ۱: پارامترهای دناویت-هارتنبرگ

$$Modified\,DH:$$

$$T_1^0=\begin{bmatrix} C_1 & -S_1 & 0 & 0 \\ S_1 & C_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_2^1=\begin{bmatrix} C_2 & -S_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ S_2 & C_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_3^2=\begin{bmatrix} C_3 & -S_3 & 0 & a_2 \\ S_3 & C_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_4^3=\begin{bmatrix} c_4 & -S_4 & 0 & a_3 \\ S_4 & C_4 & 0 & d_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_5^4=\begin{bmatrix} C_5 & -S_5 & 0 & 0 \\ 0 & 0 & -1 & -d_5 \\ S_5 & C_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_6^5=\begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ -S_6 & -C_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

۳ به دست آوردن موقعیت مجری نهایی

در ادامه در متلب تابعی با نام DH تعریف می کنیم که متغیر های مفاصل را با بردار t و پارامتر های ثابت ربات را به عنوان ورودی دریافت می‌کند و در خروجی با توجه به جدول DH ماتریس‌های تبدیل را خروجی می‌دهد.
حال در فایل FK.m این تابع را با تعریف متغیرهای گفته شده صدا می‌زنیم و ماتریس‌های تبدیل را به دست می‌آوریم. در بخش صحت‌سنجی نیز در ابتدا مقادیر پارامترهای ثابت را که از دیتاشیت جمع آوری کرده بودیم، قرار می‌دهیم و حالت zero configuration ربات را در نظر می‌گیریم. سپس با استفاده از توابع Peter Corke ابتدا لینک‌های ربات را با دستور $Link$ با فرمت زیر تعریف می کنیم:

```
L = Link(dh, options)

DH = [THETA D A ALPHA SIGMA OFFSET]
```

Listing 1: Peter Corke's Link

در ادامه با دستور $SerialLink$ لینک‌ها را به هم مرتبط می‌سازیم و در نهایت با دستور $fkine$ و دادن zero configuration در ورودی، ماتریس‌های تبدیل را به دست می آوریم که نتایج با نتایج بخش قبل یکسان است.
برای ماتریس های دوران نیز ابتدا تابعی تعریف می کنیم (RDH.m) که ماتریس‌های تبدیل را در ورودی دریافت کرده و با جدا کردن بخش دوران آن، ماتریس‌های دوران را خروجی می دهد و در نهایت با صدا زدن این تابع در فایل Rotation.m ماتریس‌های دوران را به دست می آوریم:

۱.۳ ماتریس H

$$\begin{pmatrix} \cos(t_6)\,\sigma_5-\sigma_1\,\cos(t_1)\,\sin(t_6) & -\sin(t_6)\,\sigma_5-\sigma_1\,\cos(t_1)\,\cos(t_6) & \sigma_3 & d_6\,\sigma_3+d_4\,\sin(t_1)-a_3\,\cos(t_2+t_3)\,\cos(t_1)-a_2\,\cos(t_1)\,\cos(t_2)+d_5\,\sigma_1\,\cos(t_1) \\ -\cos(t_6)\,\sigma_4-\sigma_1\,\sin(t_1)\,\sin(t_6) & \sin(t_6)\,\sigma_4-\sigma_1\,\cos(t_6)\,\sin(t_1) & -\sigma_2-\sigma_6 & d_5\,\sigma_1\,\sin(t_1)-d_4\,\cos(t_1)-a_3\,\cos(t_2+t_3)\,\sin(t_1)-a_2\,\cos(t_2)\,\sin(t_1)-d_6\,(\sigma_2+\sigma_6) \\ \sigma_7\,\sin(t_6)+\sigma_1\,\cos(t_5)\,\cos(t_6) & \sigma_7\,\cos(t_6)-\sigma_1\,\cos(t_5)\,\sin(t_6) & -\sigma_1\,\sin(t_5) & d_1+d_5\,(\sin(t_2+t_3)\,\sin(t_4)-\cos(t_2+t_3)\,\cos(t_4))-a_3\,\sin(t_2+t_3)-a_2\,\sin(t_2)-d_6\,\sin(t_5)\,(\cos(t_2+t_3)\,\sin(t_4)+\sin(t_2+t_3)\,\cos(t_4)) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$where$

$$\sigma_1=\sin(t_2+t_3+t_4)$$

$$\sigma_2=\cos(t_1)\,\cos(t_5)$$

$$\sigma_3=\cos(t_5)\,\sin(t_1)-\sigma_7\,\cos(t_1)\,\sin(t_5)$$

$$\sigma_4=\cos(t_1)\,\sin(t_5)-\sigma_7\,\cos(t_5)\,\sin(t_1)$$

$$\sigma_5=\sin(t_1)\,\sin(t_5)+\sigma_7\,\cos(t_1)\,\cos(t_5)$$

$$\sigma_6=\sigma_7\,\sin(t_1)\,\sin(t_5)$$

$$\sigma_7=\cos(t_2+t_3+t_4)$$

۲.۳ ماتریس R

$$\begin{pmatrix} \cos(t_6)\ \sigma_3 - \sigma_1\ \cos(t_1)\ \sin(t_6) & -\sin(t_6)\ \sigma_3 - \sigma_1\ \cos(t_1)\ \cos(t_6) & \cos(t_5)\ \sin(t_1) - \sigma_4\ \cos(t_1)\ \sin(t_5) \\ -\cos(t_6)\ \sigma_2 - \sigma_1\ \sin(t_1)\ \sin(t_6) & \sin(t_6)\ \sigma_2 - \sigma_1\ \cos(t_6)\ \sin(t_1) & -\cos(t_1)\ \cos(t_5) - \sigma_4\ \sin(t_1)\ \sin(t_5) \\ \sigma_4\ \sin(t_6) + \sigma_1\ \cos(t_5)\ \cos(t_6) & \sigma_4\ \cos(t_6) - \sigma_1\ \cos(t_5)\ \sin(t_6) & -\sigma_1\ \sin(t_5) \end{pmatrix}$$

$where$

$$\sigma_1 = \sin(t_2 + t_3 + t_4)$$

$$\sigma_2 = \cos(t_1)\ \sin(t_5) - \sigma_4\ \cos(t_5)\ \sin(t_1)$$

$$\sigma_3 = \sin(t_1)\ \sin(t_5) + \sigma_4\ \cos(t_1)\ \cos(t_5)$$

$$\sigma_4 = \cos(t_2 + t_3 + t_4)$$

۴ به دست آوردن زوایای مجری نهایی

در این بخش روابط مربوط به هر کدام از نمایش‌ها را از کتاب های مرجع استخراج می‌کنیم و با داشتن ماتریس‌های تبدیل این نمایش‌ها را برای مجری نهایی ربات بر حسب متغیرهای مفصلی به دست می‌آوریم.

۱.۴ نمایش محور-زاویه(**Equivalent angle—axis representation**):

$$ {}^A_B R_K(\theta) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \qquad R_K(\theta) = \begin{bmatrix} k_x k_x v\theta + c\theta & k_x k_y v\theta - k_z s\theta & k_x k_z v\theta + k_y s\theta \\ k_x k_y v\theta + k_z s\theta & k_y k_y v\theta + c\theta & k_y k_z v\theta - k_x s\theta \\ k_x k_z v\theta - k_y s\theta & k_y k_z v\theta + k_x s\theta & k_z k_z v\theta + c\theta \end{bmatrix}$$

$$\hat{K} = \frac{1}{2\sin\theta} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}. \qquad \theta = \operatorname{Acos}\left(\frac{r_{11} + r_{22} + r_{33} - 1}{2}\right)$$

شکل ۳: روابط نمایش محور-زاویه

۲.۴ نمایش زوایای ثابت(**X-Y-Z fixed angles**):

$$\begin{aligned} {}^A_B R_{XYZ}(\gamma,\beta,\alpha) &= R_Z(\alpha) R_Y(\beta) R_X(\gamma) \\ &= \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\gamma & -s\gamma \\ 0 & s\gamma & c\gamma \end{bmatrix}, \end{aligned}$$

$$ {}^A_B R_{XYZ}(\gamma,\beta,\alpha) = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}.$$

$$\begin{aligned} \beta &= \operatorname{Atan2}(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}), \\ \alpha &= \operatorname{Atan2}(r_{21}/c\beta, r_{11}/c\beta), \\ \gamma &= \operatorname{Atan2}(r_{32}/c\beta, r_{33}/c\beta), \end{aligned} \qquad {}^A_B R_{XYZ}(\gamma,\beta,\alpha) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}.$$

شکل ۴: روابط نمایش زوایای ثابت

۳.۴ نمایش زوایای اویلر(**Z-Y-Z Euler angles**):

$$
{}^{A}_{B}R_{Z'Y'Z'}(\alpha,\beta,\gamma)=\begin{bmatrix} c\alpha c\beta c\gamma - s\alpha s\gamma & -c\alpha c\beta s\gamma - s\alpha c\gamma & c\alpha s\beta \\ s\alpha c\beta c\gamma + c\alpha s\gamma & -s\alpha c\beta s\gamma + c\alpha c\gamma & s\alpha s\beta \\ -s\beta c\gamma & s\beta s\gamma & c\beta \end{bmatrix}.
$$

$$
\begin{aligned}
\beta &= \mathrm{Atan2}\left(\sqrt{r_{31}^2+r_{32}^2},\, r_{33}\right),\\
\alpha &= \mathrm{Atan2}(r_{23}/s\beta,\, r_{13}/s\beta),\\
\gamma &= \mathrm{Atan2}(r_{32}/s\beta,\, -r_{31}/s\beta).
\end{aligned}
\qquad
{}^{A}_{B}R_{Z'Y'Z'}(\alpha,\beta,\gamma)=\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix},
$$

شکل ۵: روابط نمایش زوایای اویلر

۴.۴ نمایش چهارگانه یکه(**Quaternion**):

$$
\begin{gathered}
-1 = i^2 = j^2 = k^2 \\
i = jk = -kj \\
j = ki = -ik \\
k = ij = -ji \\
Q = q_0 + iq_1 + jq_2 + kq_3 \quad q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1 \\
\text{Rotation by } \theta \text{ about the unit vector } n = (n_x, n_y, n_z)^T \\
\downarrow \\
Q = (cos\frac{\theta}{2}, n_x sin\frac{\theta}{2}, n_y sin\frac{\theta}{2}, n_z sin\frac{\theta}{2})
\end{gathered}
$$

۵ به دست آوردن فضای کاری ربات

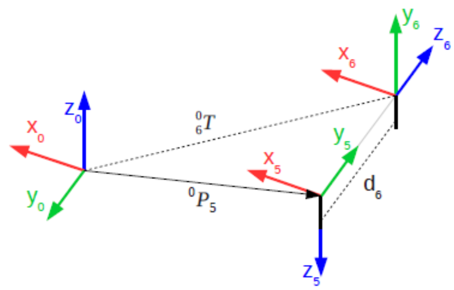
برای به دست آوردن فضای کاری پس از تعیین کردن مقادیر ثابت ربات، یک حلقه تو در تو به تعداد درجات آزادی ربات (۶) ایجاد می کنیم که هر حلقه یکی از متغیرهای مفصلی را از صفر تا 2π با فواصل 0.07π می‌چرخاند (با کاهش فواصل دقت فضای کاری بیشتر و در نتیجه حجم اطلاعات و زمان اجرا شدن برنامه بیشتر خواهد شد).
در ادامه در داخلی ترین حلقه، ماتریس تبدیل مجری نهایی به دستگاه پایه را به دست می‌آوریم و ماتریس‌های x و y و z را که به ترتیب سه سطر اول ستون چهارم ماتریس تبدیل است را مشخص می کنیم. برای پر شدن ماتریس های x و y و z در طی اجرا شدن حلقه، یک شمارنده (iterator) به نام b در خارج از حلقه‌ها تعریف می کنیم که در هر بار ماتریس‌های فوق را یک ستون جلو ببرد.
در نهایت با ترانهاده کردن ماتریس های x و y و z یک ماتریس P تعریف می کنیم که ستون‌های آن مقادیر x و y و z هستند و سپس با داشتن این ماتریس، دستور $trisurf$ مرز و محدوده خارجی این داده ها را رسم می کند که فضای کاری ربات را مشخص می کند. (تعداد همه‌ی نقاط فضای کاری بسیار زیاد است و رسم آن زمان خیلی زیادی می برد، به همین منظور فقط مرز آن را رسم میکنیم) با توجه به اینکه ربات ۶ درجه آزادی دارد و بنابر دیتاشیت آن هیچ کدام از مفاصل دورانی محدودیتی برای گردش ندارند، فضای کاری ربات یک کره خواهد شد:
برای صحت سنجی این بخش نیز مشابه قسمت سینماتیک مستقیم ابتدا لینک‌ها را تعریف کرده و سپس با دستور $plot$ (مربوط به تولباکس Peter Corke) و $pause$ در همان حلقه‌های تو در تو در هر بار فضای کاری را رسم می‌کنیم. آپشن‌های $jvec$ و $noname$ نیز فریم‌ها را بر روی ربات نمایش داده و همچنین اسم روی صفحه را پاک می‌کنند.

۶ سینماتیک معکوس ربات

برای محاسبه سینماتیک معکوس تابعی به نام inverse تعریف می کنیم که ۶ ورودی شامل جهت گیری و موقعیت مجری نهایی ربات دارد و خروجی آن بردار پارامترهای مفصلی است. در این تابع ابتدا پارامترهای ثابت ربات را مقدار دهی می کنیم و x و y و z را concatante میکنیم و در بردار $P60$ (P_6^0) قرار می‌دهیم. سپس با استفاده از قاعده زوایای ثابت و جهت گیری مجری نهایی (داده شده در ورودی) ماتریس های دوران x, y, z را به دست می‌آوریم و با ضرب این سه ماتریس در یک دیگر ماتریس دوران مجری نهایی را نسبت به فریم 0 محاسبه میشود. با کنار هم قرار دادن ماتریس $R60$ و بردار $P60$ و قرار دادن یک سطر dumy، ماتریس $T60$ به دست می‌آوریم.

۱.۶ محاسبه‌ی θ_1

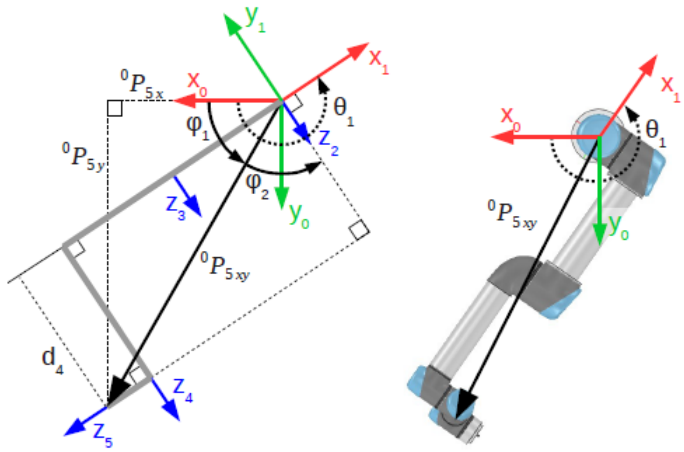
ابتدا یک مثلث تشکیل می‌دهیم که سه زاویه ان همان مراکز محور های فریم های $0, 5, 6$ میباشند. سپس با توجه به شکل میتوان دید که موقعیت مفصل ۵ ام برابر است با موقعیت مفصل ۶ ام منهای d_6 که d_6 در جهت z_6 میباشد.



شکل ۶

$$P_5^0 = P_6^0 - d_6.\hat{Z}_6^0 \tag{۱}$$

برای محاسبه θ_1 ربات را از بالا مشاهده میکنیم:



شکل ۷

با توجه به تصویر میبینیم که دو جوینت $0, 1$ روی یکدیگر قرار دارند. زاویه θ_1 با زاویه بین x_0 به x_1 در راستای z_1 برابر است . حال با توجه به تصویر میبینم که :

$$\theta_1 = \phi_1 + \phi_2 + \frac{\pi}{2} \tag{۲}$$

که در اینجا برای محاسبه ϕ_1 ابتدا بردار $\overrightarrow{P50}$ را تجزیه کرده و $atan2$ میگیریم.سپس برای محاسبه ϕ_2 با توجه به شکل داریم:

$$\cos(\phi_2) = \frac{d_4}{|{}^0P_{5xy}|} \Rightarrow$$
$$\phi_2 = \pm\text{acos}\left(\frac{d_4}{|{}^0P_{5xy}|}\right) \Leftrightarrow$$
$$\phi_2 = \pm\text{acos}\left(\frac{d_4}{\sqrt{{}^0P_{5x}{}^2 + {}^0P_{5y}{}^2}}\right)$$

شکل ۸

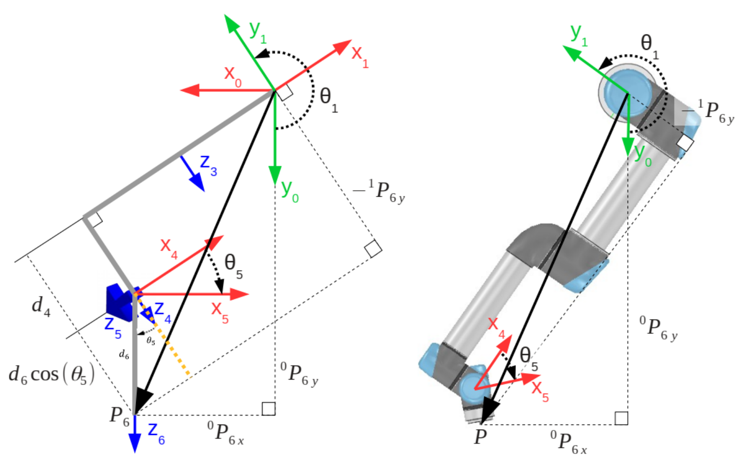
درنهایت θ_1 را با جایگذاری موارد بالا به دست می‌آوریم و داریم:

$$\theta_1 = \operatorname{atan2}\left({}^0P_{5y}, {}^0P_{5x}\right) \pm \operatorname{acos}\left(\frac{d_4}{\sqrt{{}^0P_{5x}{}^2 + {}^0P_{5y}{}^2}}\right) + \frac{\pi}{2}$$

شکل ۹

۲.۶ محاسبه‌ی θ_5

با توجه به شکل:



شکل ۱۰

به دو روش مختلف ${P_6^1}_y$ را محاسبه میکنیم:

۱.۲.۶

با توجه به اینکه θ_5 زاویه بین x_4 به x_5 در جهت z_5 می‌باشد و d_6 برابر فاصله x_5 به x_6 در جهت z_6 میباشد داریم:

$$-P_{6y}^1 = d_4 = d_6 cos\theta_5 \qquad (۳)$$

۲.۲.۶

روش دیگر به این صورت است که میتوان با استفاده از ترانهاده‌ی ماتریس R_1^0 ،P_6^0 را را به P_6^1 تبدیل کرد.(P_6^0 را از قبل داریم که در واقع موقعیت مجری نهایی نسبت به صفر هست که در ورودی داده شده است)

$$
\begin{aligned}
{}^{0}P_6 &= {}^{0}_{1}R \cdot {}^{1}P_6 \Leftrightarrow \\
{}^{1}P_6 &= {}^{0}_{1}R^{\top} \cdot {}^{0}P_6 \Leftrightarrow \\
\begin{bmatrix} {}^{1}P_{6x} \\ {}^{1}P_{6y} \\ {}^{1}P_{6z} \end{bmatrix} &= \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 \\ 0 & 0 & 1 \end{bmatrix}^{\top} \begin{bmatrix} {}^{0}P_{6x} \\ {}^{0}P_{6y} \\ {}^{0}P_{6z} \end{bmatrix} \Leftrightarrow \\
\begin{bmatrix} {}^{1}P_{6x} \\ {}^{1}P_{6y} \\ {}^{1}P_{6z} \end{bmatrix} &= \begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) & 0 \\ -\sin(\theta_1) & \cos(\theta_1) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{0}P_{6x} \\ {}^{0}P_{6y} \\ {}^{0}P_{6z} \end{bmatrix} \Rightarrow \\
{}^{1}P_{6y} &= {}^{0}P_{6x} \cdot (-\sin\theta_1) + {}^{0}P_{6y} \cdot \cos\theta_1
\end{aligned}
$$

شکل ۱۱

پس این دو رابطه باهم برابر قرار میدهیم و $cos\theta_5$ را جدا میکنیم و در اخر $acos$ می‌گیریم.

$$
\begin{aligned}
-d_4 - d_6\cos\theta_5 &= {}^{0}P_{6x}(-\sin\theta_1) + {}^{0}P_{6y}\cos\theta_1 \Leftrightarrow \\
\cos\theta_5 &= \frac{{}^{0}P_{6x}\sin\theta_1 - {}^{0}P_{6y}\cos\theta_1 - d_4}{d_6} \Leftrightarrow \\
\theta_5 &= \pm\mathrm{acos}\left(\frac{{}^{0}P_{6x}\sin\theta_1 - {}^{0}P_{6y}\cos\theta_1 - d_4}{d_6}\right)
\end{aligned}
$$

شکل ۱۲

علت $real$ گرفتن از θ_5 این است که در برخی موارد مقادیر موهومی به دست می‌آیند که مدنظر ما نیستند.

۳.۶ محاسبه‌ی θ_6

به دو روش θ_6 را به دست می‌آوریم:

۱.۳.۶

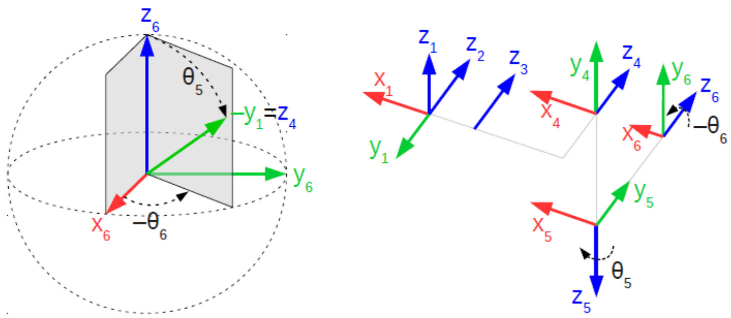
ماتریس R_0^1 و R_6^0 را داریم این دو را در یک دیگر ضرب میکنیم سپس ستون دوم ماتریس دوران را برداشته و به نام Y_1^6 قرار میدهیم که برحسب θ_1 خواهد بود که ما θ_1 را از مرحله قبل به دست اوردیم. داریم:

$$
\begin{aligned}
{}^{6}\hat{Y}_1 &= {}^{6}\hat{X}_0 \cdot (-\sin\theta_1) + {}^{6}\hat{Y}_0 \cdot \cos\theta_1 \Leftrightarrow \\
{}^{6}\hat{Y}_1 &= \begin{bmatrix} -{}^{6}\hat{X}_{0x} \cdot \sin\theta_1 + {}^{6}\hat{Y}_{0x} \cdot \cos\theta_1 \\ -{}^{6}\hat{X}_{0y} \cdot \sin\theta_1 + {}^{6}\hat{Y}_{0y} \cdot \cos\theta_1 \\ -{}^{6}\hat{X}_{0z} \cdot \sin\theta_1 + {}^{6}\hat{Y}_{0z} \cdot \cos\theta_1 \end{bmatrix}
\end{aligned}
$$

شکل ۱۳

۲.۳.۶

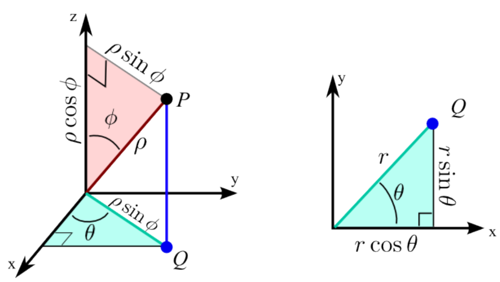
با توجه به شکل زیر:



(a) *Polar angles for* $-{}^{6}\hat{Y}_1$ **(b)** *Reference view of the relevant frames*

شکل ۱۴

و تبدیل مختصات کروی به کارتزین و روابط زیر:



$$
\begin{aligned}
x &= r \sin\theta \cos\phi \\
y &= r \sin\theta \sin\phi \\
z &= r \cos\theta
\end{aligned}
$$

شکل ۱۵

داریم:

$$
-{}^{6}\hat{Y}_1 = \begin{bmatrix} \sin\theta_5 \cos(-\theta_6) \\ \sin\theta_5 \sin(-\theta_6) \\ \cos\theta_5 \end{bmatrix} \Leftrightarrow
$$

$$
{}^{6}\hat{Y}_1 = \begin{bmatrix} -\sin\theta_5 \cos\theta_6 \\ \sin\theta_5 \sin\theta_6 \\ -\cos\theta_5 \end{bmatrix}
$$

شکل ۱۶

این دو رابطه را باهم برابر قرار میدهیم و به نتایج زیر میرسیم:

$$\left.\begin{aligned} -\sin\theta_5\cos\theta_6 &= -{}^6\hat{X}_{0x}\cdot\sin\theta_1 + {}^6\hat{Y}_{0x}\cdot\cos\theta_1 \\ \sin\theta_5\sin\theta_6 &= -{}^6\hat{X}_{0y}\cdot\sin\theta_1 + {}^6\hat{Y}_{0y}\cdot\cos\theta_1 \end{aligned}\right\}\Leftrightarrow$$

$$\left\{\begin{aligned} \cos\theta_6 &= \frac{{}^6\hat{X}_{0x}\cdot\sin\theta_1 - {}^6\hat{Y}_{0x}\cdot\cos\theta_1}{\sin\theta_5} \\ \sin\theta_6 &= \frac{-{}^6\hat{X}_{0y}\cdot\sin\theta_1 + {}^6\hat{Y}_{0y}\cdot\cos\theta_1}{\sin\theta_5} \end{aligned}\right\}\Rightarrow$$

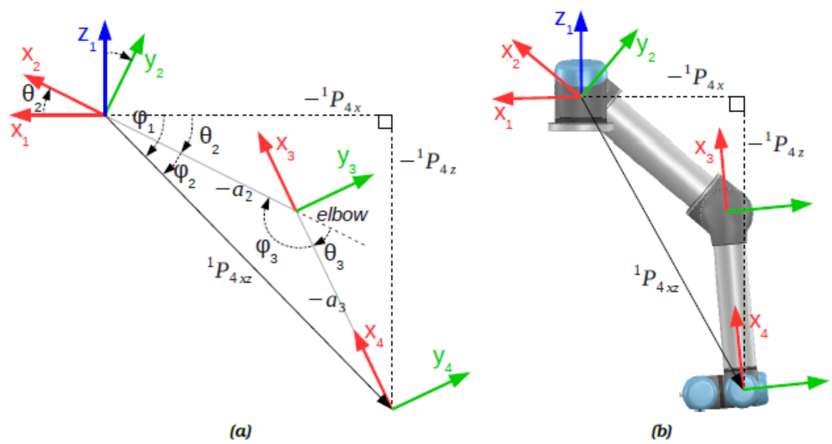
$$\theta_6 = \operatorname{atan2}\left(\frac{-{}^6\hat{X}_{0y}\cdot\sin\theta_1 + {}^6\hat{Y}_{0y}\cdot\cos\theta_1}{\sin\theta_5}, \frac{{}^6\hat{X}_{0x}\cdot\sin\theta_1 - {}^6\hat{Y}_{0x}\cdot\cos\theta_1}{\sin\theta_5}\right)$$

شکل ۱۷

و چون در مخرج $sin\theta_5$ داریم یک شرط میگذاریم که در صورتی که سینوس صفر شود یک مقدار رندم دیگر به θ_6 میدهیم.

۴.۶ محاسبه‌ی θ_3

ابتدا تابع DH را فراخوانی میکنیم.(دقت شود مقادیر به دست امده برای $\theta_1, \theta_5, \theta_6$ را نیز به عنوان ورودی میدهیم) با استفاده از ماتریس های T، T_4^1 را به دست می اوریم سپس ستون اخر آن را که همان P_4^1 است درایه x, z را برداشته و اندازه $P_{4_{xz}}$ را به دست می‌آوریم.



شکل ۱۸

با توجه به شکل میبینیم که θ_3 مکمل ϕ_3 میباشد.میتوان ϕ_3 را با استفاده از قضیه cos ها به دست آورد. داریم:

$$\cos\phi_3 = \frac{(-a_2)^2 + (-a_3)^2 - |{}^1P_{4xz}|^2}{2(-a_2)(-a_3)} = \frac{{a_2}^2 + {a_3}^2 - |{}^1P_{4xz}|^2}{2a_2a_3}$$

$$\cos\theta_3 = -\frac{{a_2}^2 + {a_3}^2 - |{}^1P_{4xz}|^2}{2a_2a_3} \Leftrightarrow$$

$$\theta_3 = \pm\operatorname{acos}\left(\frac{|{}^1P_{4xz}|^2 - {a_2}^2 - {a_3}^2}{2a_2a_3}\right)$$

$$\cos\theta_3 = -\frac{{a_2}^2 + {a_3}^2 - |{}^1P_{4xz}|^2}{2a_2a_3} \Leftrightarrow$$

$$\theta_3 = \pm\operatorname{acos}\left(\frac{|{}^1P_{4xz}|^2 - {a_2}^2 - {a_3}^2}{2a_2a_3}\right)$$

شکل ۱۹

۵.۶ محاسبه‌ی θ_2

با توجه به شکل ۱۸ داریم:

$$\theta_2 = \phi_1 - \phi_2 \tag{۴}$$

که ϕ_1 را با استفاده از ${P_4^1}_x, {P_4^1}_z$ و $atan2$ به دست می اوریم.(P ها نیز از روی ماتریس T_1^4 به دست امده اند) برای محاسبه ϕ_2 نیز از قضیه سینوس ها استفاده میکنیم و ${P_4^1}_{xz}$ را نیز از مرحله قبل داریم. درنتیجه:

$$\phi_1 = \mathrm{atan2}(-\,^1P_{4z}, -\,^1P_{4x})$$
$$\frac{\sin\phi_2}{-a_3} = \frac{\sin\phi_3}{|\,^1P_{4xz}|} \Leftrightarrow$$
$$\phi_2 = \mathrm{asin}\left(\frac{-a_3\sin\phi_3}{|\,^1P_{4xz}|}\right)$$

$$\theta_2 = \phi_1 - \phi_2 = \mathrm{atan2}(-\,^1P_{4z}, -\,^1P_{4x}) - \mathrm{asin}\left(\frac{-a_3\sin\theta_3}{|\,^1P_{4xz}|}\right)$$

شکل ۲۰

۶.۶ محاسبه‌ی θ_4

ابتدا تابع DH را فراخوانی میکنیم.(دقت شود مقادیر به دست امده برای $\theta_{1,2,3,5,6}$ را نیز به عنوان ورودی میدهیم) با استفاده از T ها T_3^4 را محاسبه میکنیم:

```
T4 =

[ cos(t4), -sin(t4), 0, -a3]
[ sin(t4),  cos(t4), 0,   0]
[       0,        0, 1,  d4]
[       0,        0, 0,   1]
```

شکل ۲۱

باتوجه به این ماتریس درایه $2,1$ ستون اول را برداشته و $atan2$ میگیریم تا θ_4 به دست اید. سپس q ها را باهم concatenate میکنیم و به صورت بردار $\overrightarrow{Q}$ خروجی میدهیم. در بخش inverse Kinematic تابع inverse را با ورودی های موقعیت و جهت گیری مجری نهایی فراخوانی میکنیم.

۷.۶ صحت‌سنجی سینماتیک معکوس

برای راستی ازمایی یک path درست کرده ایم که q_1 از -90 تا 90 درجه میرود. یک حلقه از -0.5 تا 0.5 تعریف میکنیم با گام های 0.1 که هر بار t که بردار متغیر های مفصلی ما هست همه را صفر قرار میدهیم ولی برای $i\times\pi$ ،$t(1)$ میگذاریم. سپس تابع DH را با t و بقیه پارامتر های a, d فراخوانی میکنیم و P, R را از دل ماتریس T بیرون میکشیم. با توجه به ماتریس شکل ۲۲ مقادیر α, β, γ را به دست می اوریم و به تابع inverse میدهیم و Q را به دست می اوریم.

$$^A_BR_{XYZ}(\gamma,\beta,\alpha) = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}.$$

شکل ۲۲

سپس Link های ربات را تعریف کرده و با seriaLlink انها را به هم متصل میکنیم. در نهایت $ur.plot$ میکنیم که ورودی را نیز Q میدهیم(Q هر بار که حلقه طی میشود تغییر میکند.) در میبینیم که مفصل اول یک نیم دایره را طی میکند که البته لزوما Q هایی که به ما میدهد ان چیزی نیست که ما فکر میکنیم چون به هر نقطه در فضای کاری میتوان با چند جهت گیری رسید.مهم این است که مسیر مورد نظر مارا طی میکند.

۷ به دست آوردن ماتریس ژاکوبین

ما دونوع ژاکوبین داریم: ۱. velocity propagation ۲. general

۱.۷ ژاکوبین **general**

در ژاکوبین general ما به ردیف ۱ تا ۳ ستون سوم و ستون اخر ماتریس T نیاز داشتیم که ردیف ۱تا ۳ ستون سوم z و ردیف آخر o نامگذاری شده است. با توجه به modified بودن نامگذاری محورها و نوع joint ها که همگی revolute هستند از رابطه‌ی ۵ استفاده می‌کنیم.
یک تابع با نام Jacobain_General تعریف کرده‌ایم که خروجی آن یک ماتریس 6×6 می‌باشد که سه ردیف بالا مربوط به ژاکوبین خطی است و سه ردیف اخر مربوط به ژاکوبین زاویه‌ای می‌باشد و ورودی آن هم t هست تا اگر خواستیم به t ها مقدار دهیم (t در واقع زاویه های هر joint می‌باشد). به پارامتر های a, d مقدار داده‌ایم(اگرچه میتوان این تابع را به صورت سیمبولیک هم اجرا کرد).
در ابتدا یک بار DH و RDH را فراخوانی می‌کنیم سپس خروجی این دو تابع را با دستور cat پشت سر هم قرار می دهیم در واقع ماتریس‌های R و T به دست آمده را به صورت مجزا پشت سر هم قرار می‌دهیم و یک ارایه ۳ بعدی می‌سازیم.
حلقه‌ی مربوط به o هم برای به دست اوردن ماتریس T_{ee} نسبت به $frame_0$ است تابتوانیم از ردیف ۱ تا ۳ ستون اخر آن در محاسبه ژاکوبین‌ها استفاده کنیم. یک حلقه ۶ تایی می‌نویسیم، هر بار طی شدن حلقه مربوط به یک joint می‌باشد. درون این حلقه دو ماتریس T__current و R__current را به صورت همانی تعریف می‌کنیم. حال با نوشتن یک حلقه دیگر از ۱ تا i (شماره‌ی joint مربوطه) T_i و R_i را نسبت به $frame_0$ حساب می‌کنیم. باتوجه به روابط مربوط به جزوه، ستون i ماتریس j_ω و j_v را حساب می‌کنیم. (دقت شود با هر بار طی شدن حلقه‌ی اصلی ماتریس‌های T__current و R__current دوباره تبدیل به ماتریس همانی می‌شوند) در قسمت Jacobian-General، ابتدا t را که ورودی تابع است به صورت $syms$ و یک بردار ردیفی ۶ تایی تعریف می‌کنیم و تابع ژاکوبین را فراخوانی میکنیم سپس j_ω و j_v به دست آمده را simplify کرده و سپس concatanate می‌کنیم.

$$J_{6\times6} = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix} \quad J_v = J_{\omega a} J_{\omega_{3\times6}} = [z_0^0 | R_1^0 Z_1^1 | R_2^0 Z_2^2 | R_3^0 Z_3^3 | R_4^0 Z_4^4 | R_5^0 Z_5^5 | R_6^0 Z_6^6] \tag{۵}$$

۲.۷ ژاکوبین **velocity-omega**

در این روش ابتدا باید v و ω را به دست آوریم در نتیجه یک تابع با نام velocity__omega تعریف کرده ایم که در ابتدا v و ω را برابر یک بردار سه تایی ۰ قرار می‌دهیم. t و q را به صورت متغیر های سمبولیک و یک بردار افقی ۶ تایی تعریف می‌کنیم و به a, d ها هم مقدار داده‌ایم. سپس دوباره توابع DH, RDH را فراخوانی می‌کنیم و Tها و Rها را باهم catمی‌کنیم تا به صورت یک ماتریس سه بعدی در بیایند.
سپس یک حلقه ی ۶ تایی مینویسم و در آن ماتریس P را برابر ماتریس T_{i+1} نسبت به $frame_i$ قرار می‌دهیم زیرا بعدا به ردیف ۱تا ۳ ستون آخر آن نیاز داریم.
با توجه به اینکه محور ها Modified هستند و مفصل‌ها revolute هستند از روابط موجود در جزوه استفاده کرده و در حلقه ی مربوطه v و ω نهایی را حساب میکنیم.
چون v و ω که هر بار به دست می اوریم نسبت به همان frame میباشد درنتیجه v و ω نهایی نسبت به$frame_{ee}$ می‌باشند. در نتیجه در ماتریس RT ضرب میکنیم تا نسبت به $frame_0$ آن را به دست آوریم.(دقت شود هر بار در v و ω از v و ω قبلی استفاده می‌شود و هر بار اپدیت هم می‌شوند)
در روش اصلی، باید اینگونه باشد که یک ماتریس به دست اوریم که باضرب در یک بردار ستونی که شامل مشتق متغیر های مفصل میباشد به v و ω نهایی برسیم که ماتریس های به دست امده j_v و j_ω خواهند بود.
ایده ای که در اینجا پیاده سازی شده بدین گونه است که ما به جای استفاده از مشتق t ها یک بردار سیمبولیک با نام q تعریف کرده ایم و درنتیجه برای مثال اگر از درایه اول v نسبت به $q(1)$ مشتق بگیریم بقیه q ها صفر می‌شوند و ضرایبشان از بین میرود در نتیجه ضریب $q(1)$ باقی می‌ماند و در درایه ۱۱ ماتریس j_v قرار می‌گیرد.
با یک حلقه ۶ تایی این کار انجام شده است و نتایج مربوط به j_ω و نتایج مربوط به j_v را باهم concatenate می‌کنیم تا j_v و j_ω به دست بیایند.
حال در بخش مربوط به Jacobian__Velocity__propagation تابع velocity__omega را فراخوانی کرده و تابع Jacobain__Velocity__propagation را با خروجی تابع قبلی فراخوانی میکنیم. سپس با concatenate کردن j_v و j_ω به دست آمده، به ژاکوبین اصلی می‌رسیم.

۸ به دست آوردن تکینگی‌های ربات

در ابتدا serial Link ها را تعریف کرده ایم. حال یک حلقه ی ۶ تایی داریم که متغیر های مفصلی را با گام‌های$0.2, 0.2$ جلو میبرد و سپس ژاکوبین general را با متغیر های مفصلی حلقه ها فراخوانی میکنیم و j_v و j_ω را concatenate می‌کنیم.
سپس دترمینان ژاکوبین را حساب کرده و اگر از یک حد مشخصی کمتر بود آن حالت را $ur.plot$ می‌کنیم. همه ی حالت های singular را با گام های 0.001 ثانیه ای $ur.plot$ می‌کنیم.
روش دیگر(همان قسمت کامنت شده) در این روش سه مفصل اخر را در یک جهت گیری قفل کرده ایم که این جهت گیری موجب singularity نمی‌شود و سه جوینت اول را مچرخانیم چون در روش قبلی زمان زیادی طول میکشد تا کشیدگی ها را رد کند و به نقاط واضح تر singularity برسد.
در این روش سه لینک اخر را بدون offset در نظر میگیریم و ماتریس ژاکوبین را به صورت بلوکی یعنی ۴ ماتریس 3×3 می‌نویسیم که شامل $J11, J12, J22, J21$ می‌شود. $J11$ مربوط به سه لینک اول و ژاکوبین خطی انهاست و $J21$ مربوط به سه مفصل اول و ژاکوبین زاویه ای ان ها است. $J22$ مربوط به ژاکوبین زاویه سه مفصل اخر میباشد. $J12$ صفر میشود که در واقع ژاکوبین خطی سه مفصل اخر می‌باشد که چون offset ندارند R هم ندارند که R ضرب خارجی اش در ω را بتوان حساب کرد درواقع صفر می‌شود در نتیجه ماتریس ژاکوبین ما یک ماتریس پایین مثلثی میشود که در صورتی دترمینانش صفر است که دترمینان بلوک های روی قطرش صفر شود .دترمینان $J22$ صفر نمیشود چون در حالتی فیکس شده است که باعث singularity نشود. در نتیجه $J11$ را حساب میکنیم و اگر از حدی کوچک تر بود آن حالت را به عنوان singularity وارد ماتریس singular می‌کنیم و همچنین آن ها را هم $ur.plot$ می‌کنیم.

۹ به دست آوردن دینامیک ربات

برای محاسبه دینامیک یک تابع تعریف می‌کنیم که ورودی نمی‌گیرد و ماتریس های D و C و G را خروجی می‌دهد و در نهایت با صدا زدن این تابع ماتریس‌های فوق محاسبه می‌شوند. روند کلی این تابع به این صورت است که در ابتدا پارامتر های ثابت ربات، متغیرهای مفصلی (t)، متغیرهای مربوط به جرم لینک‌ها (m) و متغیرهای اینرسی لینک‌ها را به طور کلی تعریف می‌کنیم و سپس با استفاده از حلقه برای هر کدام از لینک‌ها ماتریس متقارن 3×3 اینرسی و ماتریس سطری جرم (شامل جرم های هر لینک) را به دست می‌آوریم.
سپس ماتریس‌های D و C و G و بردار $\overrightarrow{rc}$ (فواصل مراکز جرم از زمین) و P و همچنین برای ژاکوبین ها، ماتریس‌های j_v و j_ω را با توجه به ابعادشان تعریف کرده و مقدار اولیه صفر می‌دهیم. چون در ادامه همه محاسبات به صورت سمبولیک انجام می شود، این مقداردهی اولیه را نیز با sym تعریف می‌کنیم.
در این بخش علاوه بر توابع DH و RDH که ماتریس‌های تبدیل و دوران را تولید می کردند، به تابع DHC نیز نیاز داریم تا با گرفتن پارامتر های ثابت ربات، ماتریس های تبدیل مرتبط با مرکز جرم هر لینک نسبت به همان لینک را تولید کند (نسبت به پایه نیستند)، به همین ترتیب این ماتریس‌ها دوران ندارند و ماتریس دوران آن‌ها همانی است و فقط آفست‌های آن‌ها در ماتریس $\mathrm{position}$ تعیین شده است. حال پس از فراخوانی این سه تابع و با دستور cat، ماتریس‌ها را در یک آرایه سه بعدی پشت سر هم قرار می‌دهیم.
می دانیم برای به دست آوردن J_{vc_i} برای هر مفصل، در هر مرحله با تغییر مفصل و مرکز جرم، روابط مربوط به مفاصل قبل تغییر می کنند اما مفاصل بعدی آن تأثیری ندارند و ستون های مفاصل بعدی صفر می شوند. برای $J_{\omega c_i}$ نیز مفاصل بعدی بی‌تأثیر خواهند بود.
به عنوان نمونه برای سه مفصل اول به این صورت خواهد بود (همه مفاصل ربات گردشی هستند):

$$J_{vc_1} = [Z^0_{c_1} \times (O^0_{c_1} - O^0_1)|0|0|0|0|0]$$
$$J_{vc_2} = [Z^0_{c_1} \times (O^0_{c_2} - O^0_1)|Z^0_{c_2} \times (O^0_{c_2} - O^0_2)|0|0|0|0]$$
$$J_{vc_2} = [Z^0_{c_1} \times (O^0_{c_2} - O^0_1)|Z^0_{c_2} \times (O^0_{c_2} - O^0_2)|Z^0_{c_3} \times (O^0_{c_3} - O^0_3)|0|0|0]$$

$$J_{\omega c_1} = [Z^0_{c_1}|0|0|0|0|0]$$
$$J_{\omega c_1} = [Z^0_{c_1}|Z^0_{c_2}|0|0|0|0]$$
$$J_{\omega c_1} = [Z^0_{c_1}|Z^0_{c_2}|Z^0_{c_3}|0|0|0]$$

بنابراین در هر مرحله برای محاسبه J_{vc} به ماتریس‌های T^0_c و T^0_i و R^0_i و ماتریس های T^0_c و R^0_j که $j < i$ است نیاز داریم. به همین منظور برای محاسبه T^0_c و T^0_i و R^0_i ابتدا خارج از حلقه دو ماتریس همانی $current$ تعریف می‌کنیم و در هر بار گذر حلقه ماتریس‌های $T^{i-1}_{c_i}$ و T^{i-1}_i و R^{i-1}_i در ماتریس‌های مفاصل قبلی ضرب می‌شوند و در هر حلقه ماتریس‌های $T^0_{c_i}$ و R^0_i را در ماتریس های $stack$ شان ذخیره می کنیم.
حال با داشتن ماتریس‌های فوق و مطابق روابط بالا J_{vc_i} و $J_{\omega c_i}$ برای هر مفصل محاسبه می‌شوند.
در ادامه برای هر مفصل مقدار مولفه‌ی z بردار $\overrightarrow{O^0_{c_i}}$ را برای $\overrightarrow{rc}$ آن مفصل قرار می‌دهیم (فواصل مراکز جرم تا زمین) و ماتریس D را نیز با توجه به رابطه‌ی آن تشکیل می‌دهیم.
روابط ماتریس D و C و G و P به صورت زیر هستند:

$$P = \sum_{i=1}^{n} P_i = \sum_{i=1}^{n} g^T r_{ci} m_i \qquad K = \frac{1}{2}\dot{q}^T D(q)\dot{q} \qquad K = \frac{1}{2}\dot{q}^T \sum_{i=1}^{n} \left[m_i J_{v_i}(q)^T J_{v_i}(q) + J_{\omega_i}(q)^T R_i(q) I_i R_i(q)^T J_{\omega_i}(q) \right] \dot{q}$$

$$G_k = \frac{\partial P}{\partial q_k}$$

شکل ۲۳: روابط ماتریس‌های دینامیکی

برای ماتریس C به ۳ اندیس و در نتیجه ۳ حلقه تو در تو نیاز داریم که در آن با توجه به فرمول c_{ijk} نسبت به عناصر ماتریس D مشتق گرفته و ماتریس C را به صورت زیر پر می‌کند:

$$C_k = [\sum_{j=1}^{n} c_{1jk} \sum_{j=1}^{n} c_{2jk} ... \sum_{j=1}^{n} c_{njk}] \tag{۶}$$

در ادامه برای استفاده بهینه از حلقه با استفاده از اندیس اول (k) ماتریس P را نیز با توجه به رابطه‌اش تشکیل می دهیم که در آن g را برابر 9.8 قرار داده‌ایم. در آخر نیز در یک حلقه جداگانه با مشتق گرفتن از عناصر ماتریس P نسبت به متغیر مفصلی، ماتریس G را محاسبه می‌کنیم.