



مقدمه

تحلیل حرکت یک ربات تحت ورودی‌های مختلف برای ما از اهمیت بالایی برخوردار است. نرم‌افزارهای مختلفی برای این منظور طراحی شده و توسعه یافته‌اند؛ در این پروژه، ابتدا به تحلیل حرکت‌های مختلف یک ربات چهار درج آزادی می‌پردازیم و سپس مراحل تحلیل ربات را در بخش سیمولینک نرم‌افزار متلب پیاده‌سازی می‌کنیم.

هدف این پروژه، استخراج مدل دینامیکی ربات و تحلیل یک ربات چهار درجه آزادی در محیط سیمولینک تحت ورودی‌های گشتاور و سرعت و همچنین استخراج نمودارهای مطلوب در هر بخش می‌باشد.

حتما قبل از شروع انجام پروژه، ویدیوهای آموزشی تهیه شده را مشاهده فرمایید



بخش اول: سینماتیک مستقیم و معکوس

در این بخش قصد داریم قدم به قدم سینماتیک مستقیم و معکوس یک ربات چهار درجه آزادی را بدست آوریم. سپس با معین بودن مختصات نقطه ابتدایی و انتهایی برای یک حرکت خاص، موقعیت مفاصل در ابتدا و انتهای حرکت را محاسبه کرده و به کمک یک مسیریابی مناسب، از نقطه ابتدا به نقطه انتها برویم.

در انتهای همین فایل نقشه‌های قسمت‌های مختلف ربات آورده شده‌است. اطلاعات موجود در این نقشه‌ها برای بدست آوردن پارامترهای $D - H$ ربات مورد نیاز است.

فرض کنید می‌خواهیم ربات از مختصات ابتدایی

$$\begin{bmatrix} x \\ y \\ z \\ \varphi \end{bmatrix} = \begin{bmatrix} \quad \\ \quad \\ \quad \\ \quad \end{bmatrix}$$

به مختصات انتهایی

$$\begin{bmatrix} x \\ y \\ z \\ \varphi \end{bmatrix} = \begin{bmatrix} \quad \\ \quad \\ \quad \\ \quad \end{bmatrix}$$

برود. همچنین می‌خواهیم در ابتدا و انتهای حرکت، شتابی نداشته باشیم تا کمترین تغییرات در نیروی وارده به مجری نهایی اتفاق بیفتد. با توجه به محدودیت‌های موجود در ربات و کاربرد مورد نظر، زمان این حرکت از لحظه شروع تا لحظه پایان باید برابر $t = 1.5 \text{ s}$ باشد.

با انجام مراحل زیر، خواسته‌های این بخش بدست خواهد آمد.

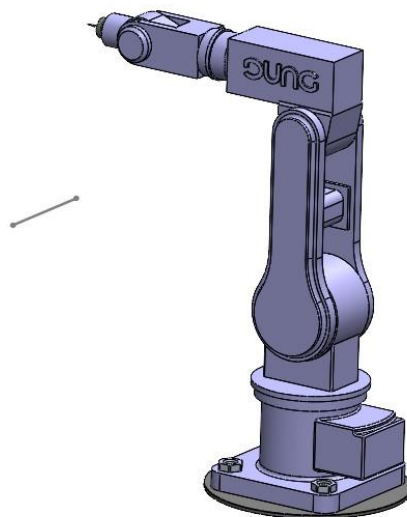


- i. جدول پارامترهای $D - H$ را برای این ربات پیدا کنید.
- ii. ماتریس‌های Q_i و بردارهای a_i را تشکیل دهید.
- iii. معادلات سینماتیک مستقیم را برای این ربات بدست آورده و با دستور *simplify* حاصل را به ساده‌ترین صورت ممکن بنویسید.
- iv. قطعه کدی در محیط متلب بنویسید که با دریافت مختصات مجری نهایی، سینماتیک معکوس را برای ربات حل کرده و پاسخ‌های احتمالی برای موقعیت مفاصل را نشان دهد. این قطعه کد می‌تواند با دریافت ورودی به صورت تابع نیز خواسته‌های مسئله را نمایش دهد.
- v. با توجه به موقعیت‌های بدست آمده به کمک قطعه کد نوشته‌شده برای حالت‌های ابتدایی و انتهایی ربات، به کمک مسیریابی مناسب برای کاربرد خواسته شده، موقعیت هر مفصل در طول زمان را بیابید و در نمودارهایی رسم کنید.

بخش دوم: مسیریابی میانی

فرض کنید از این ربات برای یک عملیات جوشکاری استفاده می‌کنیم. در این عملیات، برای ما اهمیت دارد که ربات از چه مسیری بین نقطه ابتدایی و انتهای حرکت می‌کند. مکانی که قرار است توسط این ربات جوشکاری شود، یک خط موازی با محور x و در فاصله $y = 70\text{ cm}$ و $z = 60\text{ cm}$ بالاتر از زمین می‌باشد. طول محل جوشکاری شده نیز برابر $l = 10\text{ cm}$ می‌باشد.

محل حرکت ربات برای انجام عملیات جوشکاری در شکل (۱) قابل مشاهده است.



شکل ۱ - نمایش محلی که قرار است توسط ربات جوشکاری شود

توجه داشته باشید که مختصات ابتدایی ربات برای شروع حرکت در این بخش، همان مختصات انتهای ربات در قسمت قبل می‌باشد.

با انجام مراحل زیر، خواسته‌های این بخش بدست خواهد آمد.



i. محل جوشکاری که یک خط موازی محور x می باشد را به تعدادی نقطه تقسیم کنید. قرار است چند جمله ای مناسب برای برای حرکت این ربات در این مسیر را بیابیم.
در هر نقطه، وضعیت موقعیت مجری نهایی مشخص است. به طور مثال اگر تعداد نقاط میانی را برابر سه نقطه با فاصله یکسان در نظر گرفته باشید، نقطه اول در مختصات y و z نقطه ابتدایی حرکت و با فاصله ۲.۵ سانتی متر در راستای محور x قرار دارد.

ii. برای هر نقطه، اعم از نقاط میانی و نقطه انتهایی، به کمک فایل تهیه شده سینماتیک معکوس را حل کرده و موقعیت مفاصل ربات را بدست می آوریم.

در این بخش توجه داریم که موقعیت مفاصل در نقطه ابتدایی برای ما مشخص است و از حل سینماتیک معکوس در قسمت قبل بدست می آید. برای باقی نقاط، پس از حل مسئله سینماتیک معکوس به کمک قطعه کد نوشته شده، در صورت وجود چند پاسخ برای سینماتیک معکوس، نزدیک ترین موقعیت به مقادیر بدست آمده در نقطه قبلی را انتخاب می کنیم تا ربات با کمترین حرکت به موقعیت جدید برسد.

iii. پس از مشخص شدن موقعیت مفاصل در هر نقطه، با دانستن مقادیر θ در هر لحظه و تعیین $\dot{\theta}$ در نقاط میانی برابر با $\dot{\theta} = 0$ در نقطه پایانی، می توانیم مسیریابی را برای حرکت مطلوب انجام دهیم.

iv. برای تعیین مسیر حرکت، نیاز است ضرایب معادله زیر را به کمک حل یک دستگاه معادلات پیدا کنیم:

$$S(\tau) = a_n \tau^n + a_{n-1} \tau^{n-1} + \dots + a_1 \tau + a_0$$

که در آن، مقدار $S = \frac{\theta(t) - \theta_I}{\theta_F - \theta_I}$ و $\tau = \frac{t}{T}$ می باشد.

توجه کنید که تعداد مجهولات و مرتبه مسیریابی بسته به انتخاب تعداد نقاطی میانی می تواند متفاوت باشد.

v. در انتخاب تعداد و محل نقاط میانی دقت کنید. مفاصل ربات در تمام نقاط میانی سرعت ثابت دارند. از طرفی به دلیل کاربرد جوشکاری برای این ربات، علاقه مندیم تا حد امکان مجری نهایی با سرعت ثابت حرکت کند. پس بهتر است اولین نقطه میانی که در آن سرعت ربات به حد نهایی می رسد، در نزدیکی نقطه شروع حرکت باشد. همین توضیح برای آخرین نقطه میانی نیز صادق است.



- vi. فاصله نقاط میانی می تواند یکسان نباشد.
- vii. با پیدا شدن چند جمله ای مسیریابی، موقعیت مفاصل ربات در هر زمان به کمک رابطه زیر مشخص می شود. این موقعیت ها را در نمودارهایی بر حسب زمان رسم کنید.
- $$\theta_{(\tau)} = \theta_I + (\theta_F - \theta_I) \times s_{(\tau)}$$
- viii. به کمک سینماتیک مستقیم، موقعیت مجری نهایی ربات را در هر لحظه پیدا کنید و در نمودارهایی رسم کنید.
- ix. در حالت ایده آل این نمودارها باید تغییراتی در راستای و نداشته باشند و تغییرات آنها در راستای محور نیز به صورت یک خط باشد. اگر این چنین نیست، توضیح دهید چرا و چه پیشنهادی برای رفع این مشکل دارید؟



بخش سوم: استخراج مدل دینامیکی

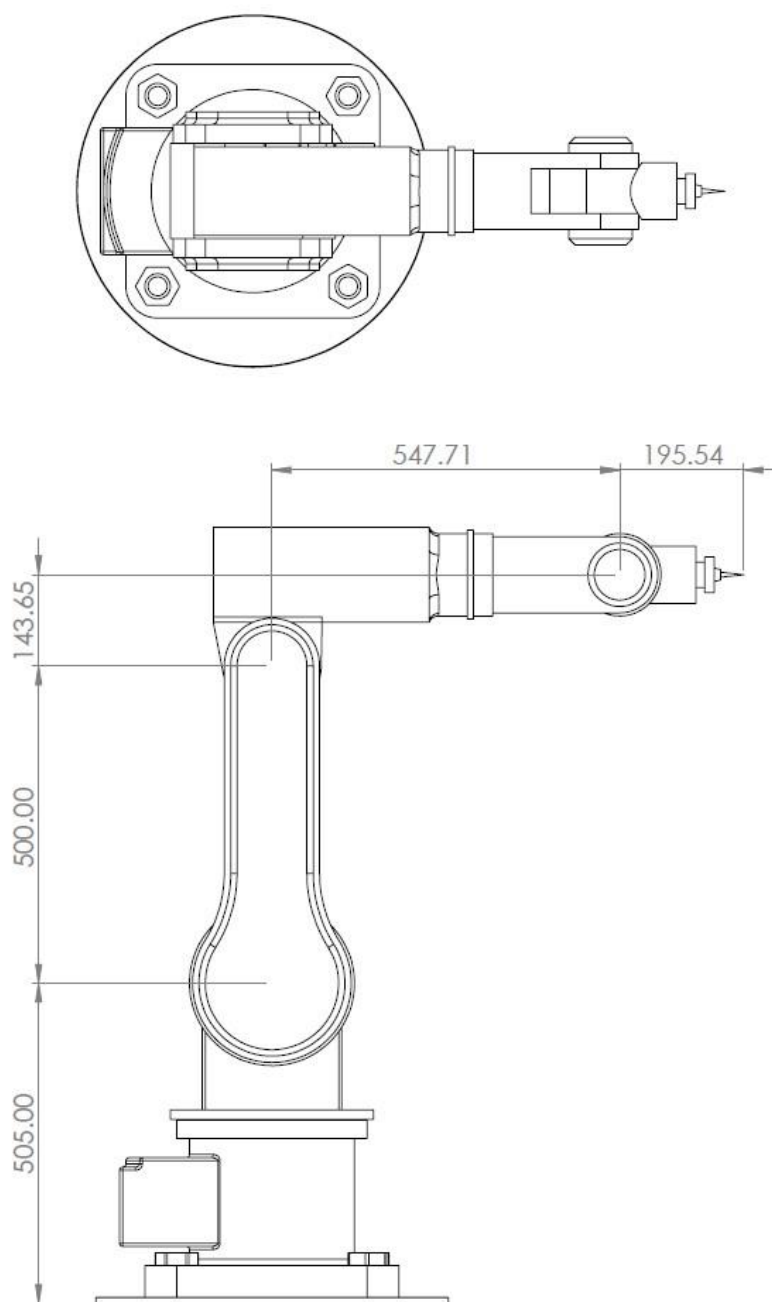
در این بخش، با توجه به نقشه‌های ربات و همچنین داشتن جرم و ممان اینرسی هر مفصل، اقدام به محاسبه مدل دینامیکی ربات کنید. ابتدا ماتریس جرمی $(\mathbf{M}(\theta))$ را به کمک ماتریس‌های \mathbf{N}_i و \mathbf{W}_i محاسبه کنید و سپس به کمک روابط موجود، گشتاور مورد نیاز در هر مفصل را به کمک قطعه کدی در محیط متلب برای حرکتی معین با معلوم بودن $\ddot{\theta}$ و $\dot{\theta}$ و θ بدست آورید.

$$\mathbf{M}\ddot{\theta} + \dot{\mathbf{M}}\dot{\theta} - \frac{1}{2}\dot{\theta}^T \frac{\partial \mathbf{M}}{\partial \theta} \dot{\theta} + \frac{\partial V}{\partial \theta} = \vec{\tau}$$

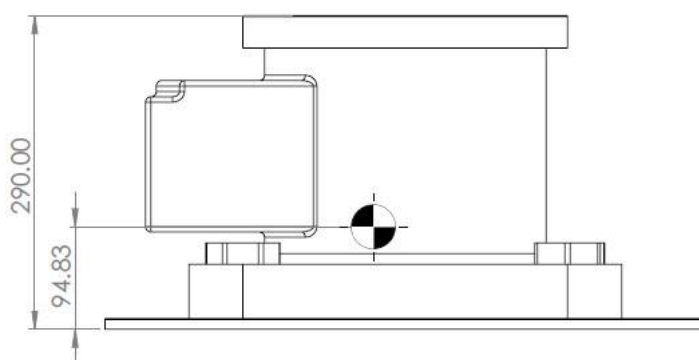
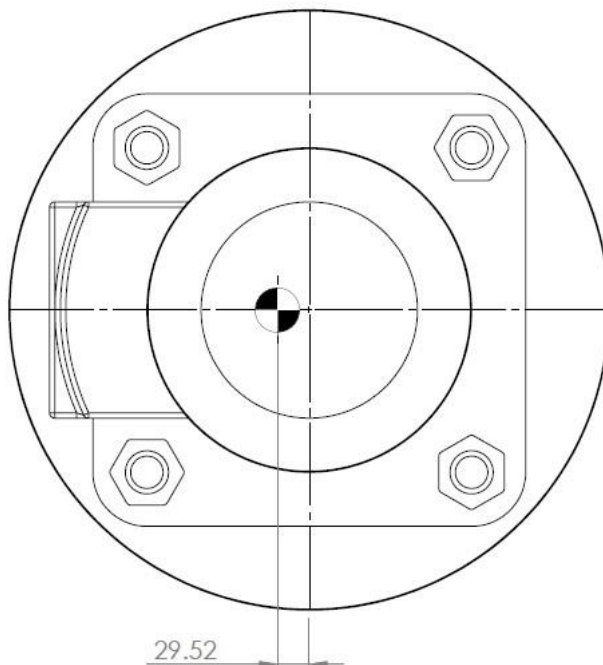
توجه کنید که مقادیر داده شده برای ماتریس‌های ممان اینرسی برای مرکز جرم لینک‌ها می‌باشد.

برای مسیریابی قسمت اول سوال، با معلوم بودن $\ddot{\theta}$ و $\dot{\theta}$ در هر لحظه و پیدا کردن $\ddot{\theta}$ مقادیر گشتاور مورد نیاز هر مفصل در هر لحظه را بیابید و در نمودارهایی رسم کنید.

در ادامه قصد داریم مدل دینامیکی بدست آمده را اعتبار سنجی کنیم. برای این منظور از نرم‌افزار سیمولینک استفاده خواهیم کرد.

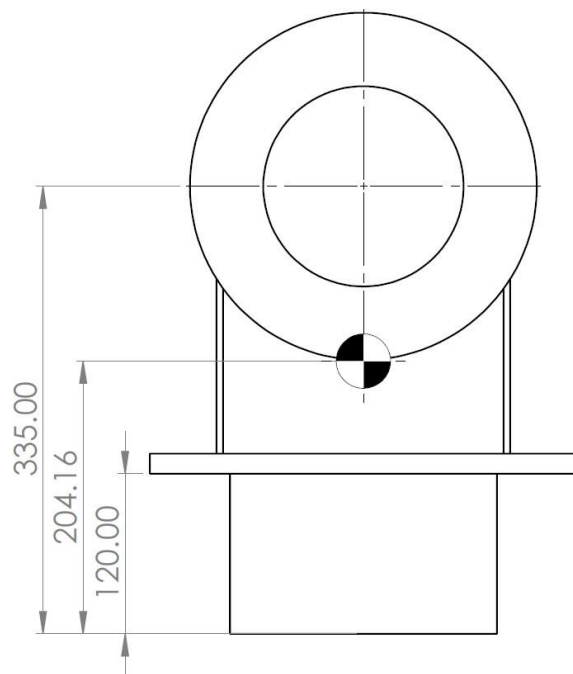


شکل ۲ - نمای کلی ربات چهار درجه آزادی

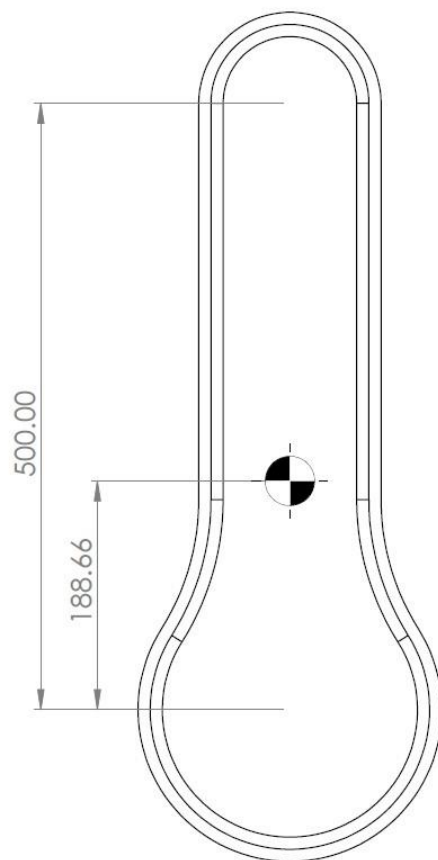


شکل ۳ - پایه ربات (*Base*)

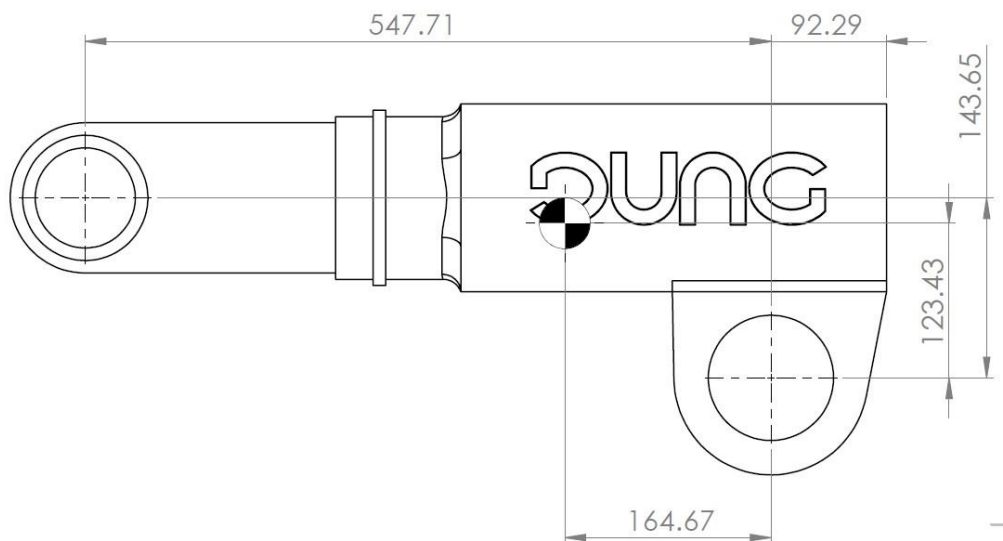
$$M = 10 \text{ kg} \quad I = \begin{bmatrix} \quad \quad \quad \end{bmatrix}$$



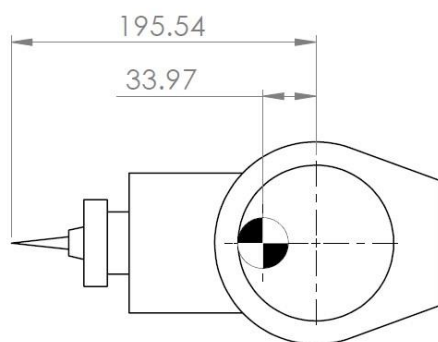
شکل ۴ - مفصل اول



شکل ۵ - مفصل دوم



شکل ۶ - مفصل سوم



شکل ۷ - مفصل چهارم



بخش چهارم: اعتبارسنجی نتایج در سیمولینک

در این قسمت قصد داریم درست بودن مدل دینامیکی بدست آمده در قسمت قبل را تحقیق کنیم. برای این کار لازم است در ابتدا ربات را در محیط سیمولینک مونتاژ کنید. برای این کار، طبق توضیحات ارائه شده در پیوست ویدیویی عمل کنید.

فایل‌های قسمت‌های مختلف ربات در پوشه *Step – Files* با فرمت *Step* در اختیار شما قرار داده شده است.

○ برای عملیات مونتاژ حتماً به مبداء مختصات در هر کدام از لینک‌ها و ابعاد موجود در نقشه‌ها دقت کنید. این نقطه قابل مشاهده در فایل‌های داده شده می‌باشد و لزوماً منطبق بر مرکز جرم قطعه نیست.

در ادامه اطلاعات اینرسی هر لینک را طبق توضیحات بالا برای هر لینک تعیین کنید.

پس از انجام موارد فوق، در این قسمت، ورودی مفاصل را به صورت موقعیت در نظر می‌گیریم. برای این منظور، نتایج بدست آمده از قسمت اول برای موقعیت مفاصل در طول زمان را به صورت *time_series* ذخیره می‌کنیم تا در محیط سیمولینک قابل استفاده باشد. باید مشتقات اول و دوم این پارامتر نیز محاسبه شود و به لینک مربوطه داده شود.

حسگر موجود در هر لینک را روی حالت اندازه‌گیری گشتاور تنظیم کنید و داده‌های بدست آمده را در *workspace* متلب ذخیره کنید. این داده‌ها، همان گشتاور مفاصل در طول زمان می‌باشند. گشتاورهای بدست آمده را در نمودارهایی در طول زمان رسم کنید و نتایج این بخش را با نتایج بدست آمده از مدل دینامیکی ربات در قسمت قبل مقایسه کنید.



می‌دانیم در ربات‌های واقعی، ورودی مفاصل به صورت گشتاور می‌باشد، و نه موقعیت. در انتها قصد داریم این ربات را یک بار دیگر با این فرض شبیه‌سازی کنیم. برای این منظور، اطلاعات بدست آمده از مدل دینامیکی برای گشتاور هر مفصل در طول زمان را به صورت *time_series* ذخیره کنید تا در محیط سیمولینک قابل استفاده باشد. سپس ورودی مفاصل را از نوع گشتاور تعیین کنید و گشتاور بدست آمده برای هر لینک را به لینک مربوطه بدهید. حسگر مفاصل را روی موقعیت قرار داده و داده‌های بدست آمده را در *workspace* متلب ذخیره کنید. این داده‌ها، موقعیت مفاصل در طول زمان برای می‌باشند.

موقعیت هر مفصل در طول زمان را در نمودارهایی رسم کنید و نتایج را با نتایج بدست آمده در قسمت اول مقایسه کنید.

امتیازی

در صورت وجود خطا میان داده‌های بدست آمده از سیمولینک و داده‌های محاسبه‌شده در بخش اول، با داشتن مقدار مورد انتظار در هر لحظه و مقدار بدست آمده در همان لحظه، به کمک بلوک *PID* یک کنترلر در محیط سیمولینک طراحی کنید که خطای ناشی از صحیح نبودن مدل دینامیکی را جبران کند.



نکات مهم:

- ۱- پاسخ تمرین‌ها را با فرمت zip به صورت "zip.شماره دانشجویی_نام" در Elearn آپلود نمایید.
- ۲- برای تاخیرهای کمتر از ۲۴ ساعت نمره ای کسر نخواهد شد. برای تاخیر بین ۱ تا ۲ روز ۱۰ درصد نمره و برای تاخیر بین ۲ تا ۵ روز ۵۰ درصد نمره کسر خواهد شد. تاخیر بیشتر از ۵ روز منجر به صفر شدن تمرین میگردد.
- ۳- در صورت وجود ابهام در انجام هر بخش، با دستیار آموزشی مربوطه از طریق آیدی زیر در تلگرام ارتباط داشته باشید

Telegram: @MJadidii