دانشگاه خواجه نصیرالدین طوسی

دانشکده برق- گروه مکاترونیک

تمرین درس رباتیک دوره کارشناسی ارشد

رشته مهندسی مکاترونیک

عنوان

تمرین رباتیک

نگارش

علیرضا امیری

آذر ۱۴۰۳

پاسخ سوالات سری چهارم

پاسخ سوال یک

در این بخش، ابتدا کدی برای محاسبه ی ماتریس های تبدیل با در اختیار داشتن پارامتر های DH نوشته شده است. محاسبه ی ماتریس های تبدیل مربوط به هر یک از لینک ها با ساتفاده از تابع DH از پیش تعریف شده انجام گرفته است و برای سادگی کار، به جای فراخوانی مکرر این دستور با پارامتر های متفاوت برای هر لینک، ابتدا پارامترها در یک ماتریس تعریف شده و سپس در یک حلقه ی for، هر یک از ماتریس های تبدیل محاسبه می شوند. برای ربات $۲R$ داریم:

$$DHparameters = \begin{pmatrix} ۰ & ۰ & a_۱ & \theta_۱ \\ ۰ & ۰ & a_۲ & \theta_۲ \end{pmatrix}$$

لازم به ذکر است که پیش از این، تعداد لینک ها برای برنامه مشخص شده است و همچنین، طول بازوهای ربات نیز در یک آرایه برای استفاده های بعدی تعریف شده است. ماتریس های تبدیل ربات $۲R$ به صورت زیر محاسبه خواهند شد:

$$T_{۱۰} = \begin{pmatrix} \cos(\theta_۱) & -\sin(\theta_۱) & ۰ & ۰ \\ \sin(\theta_۱) & \cos(\theta_۱) & ۰ & ۰ \\ ۰ & ۰ & ۱ & a_۱ \\ ۰ & ۰ & ۰ & ۱ \end{pmatrix}$$

$$T_{20} = \begin{pmatrix} \sigma_1 & -\cos(\theta_1)\,\sin(\theta_2) - \cos(\theta_2)\,\sin(\theta_1) & 0 & 0 \\ \cos(\theta_1)\,\sin(\theta_2) + \cos(\theta_2)\,\sin(\theta_1) & \sigma_1 & 0 & 0 \\ 0 & 0 & 1 & a_1 + a_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{where} \quad \sigma_1 = \cos(\theta_1)\,\cos(\theta_2) - \sin(\theta_1)\,\sin(\theta_2)$$

پس از محاسبه ی ماتریس های تبدیل برای هر لینک، می توان ماتریس های دوران و انتقال را در هر مفصل نسبت به دستگاه مختصات مرجع به سادگی به دست آورد. بنابراین، آخرین ماتریس دوران و انتقال به دست آمده، مربوط به پنجه ی ربات و یا نقطه ی $EndEffector$ خواهد بود.

$$P_{endeffector} = \begin{pmatrix} 0 \\ 0 \\ a_1 + a_2 \end{pmatrix}$$

$$R_{endeffector} = \begin{pmatrix} \cos(\theta_1)\,\cos(\theta_2) - \sin(\theta_1)\,\sin(\theta_2) & -\cos(\theta_1)\,\sin(\theta_2) - \cos(\theta_2)\,\sin(\theta_1) & 0 \\ \cos(\theta_1)\,\sin(\theta_2) + \cos(\theta_2)\,\sin(\theta_1) & \cos(\theta_1)\,\cos(\theta_2) - \sin(\theta_1)\,\sin(\theta_2) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

در گام بعد، ماتریس ژاکوبین محاسبه خواهد شد. اما نکته ای که باید در این قسمت به آن توجه شود این است که در این سوال، ماتریس های ژاکوبین مربوط به مرکز جرم هر لینک خواسته شده است و بنابراین، با توجه به محاسبات موجود در هر کتاب، برای لینک آخر باید نیمی از طول آن را در محاسبات لحاظ کرد و در ترم های قبلی، طول کامل بازوها را در نظر گرفت.

با توجه به این مورد، ابتدا در یک حلقه ی for، ماتریس های دوران و انتقال تجمعی برای مفاصل ربات به

صورت کامل و بدون لحاظ کردن مراکز جرم محاسبه شده و در تنسورهایی ذخیره می شوند.

حال، برای محاسبه ی ماتریس های ژاکوبین خطی مطابق روابط ۵٫۴۹، ۵٫۵۳ و ۵٫۵۴ محاسبه می شوند. چنان که پیش از این بیان شد، در هر مرحله از محاسبه، برای لینک فعلی نیمی از طول آن در نظر گرفته خواهد شد و برای لینک های قبلی، با استفاده از نتایج ذخیره شده، ماتریس ها به صورت کامل محاسبه می شوند. نکته ای باید در این محاسبه در نظر گرفته شود آن است که ماتریس های ژاکوبین برای هر لینک باید ذخیره شوند تا در ادامه در محاسبه ی ماتریس جرمی و .. مورد استفاده قرار می گیرند. در نهایت، ماتریس ژاکوبین خطی با در اختیار داشتن این ماتریس های موقعیت محاسبه می شود.

$$J_v = \begin{pmatrix} -\frac{a_۲\ \sin(\theta_۱+\theta_۲)}{۲} - a_۱\ \sin(\theta_۱) & -\frac{a_۲\ \sin(\theta_۱+\theta_۲)}{۲} \\ \frac{a_۲\ \cos(\theta_۱+\theta_۲)}{۲} + a_۱\ \cos(\theta_۱) & \frac{a_۲\ \cos(\theta_۱+\theta_۲)}{۲} \\ ۰ & ۰ \end{pmatrix}$$

$$J_w = \begin{pmatrix} ۰ & ۰ \\ ۰ & ۰ \\ ۱ & ۱ \end{pmatrix}$$

از مقایسه ی نتایج به دست آمده در این روش با پاسخ های کتاب، مشاهده می شود که به صورت کامل همخوانی دارند.

پاسخ سوال دو

برای پاسخ به این سوال، مطابق رابطه ی ۵٫۶۲ برای محاسبه ی معادلات دینامیک سیستم به فرم بسته لازم است ابتدا ماتریس های جرمی، گرانش و کوریولیس ربات محاسبه شوند.

ماتریس جرمی

محاسبه ی ماتریس جرمی با استفاده از رابطه ی ۵/۹۵ انجام می شود که پیش از آن لازم است ممان های اینرسی ربات برای هر مرکز لینک در ماتریس $\mathbf{I}$ محاسبه شود. برای این محاسبه خواهیم داشت:

$$I_i = \frac{۱}{۱۲} m_i L_i^{۲} \quad \text{for} i = ۱, ۲, \ldots, n$$

در نتیجه با محاسبه ی این مقادیر برای ربات $۳R$ خواهیم داشت:

$$I = \begin{pmatrix} \frac{a_۱ \, {m_۱}^{۲}}{۱۲} \\ \frac{a_۲ \, {m_۲}^{۲}}{۱۲} \\ \frac{a_۳ \, {m_۳}^{۲}}{۱۲} \end{pmatrix}$$

ممان اینرسی

برای محاسبه ی ماتریس ممان اینرسی، از تابع $InertiaFunction(Lengths, Masses)$ که به همین منظور نوشته شده است استفاده می شود. حال با در اختیار داشتن این روابط، برای محاسبه ماتریس جرمی مطابق رابطه ی زیر خواهیم داشت:

$$M = \sum_{i=۱}^{n} \left(m_i \mathbf{Jv}_i^T \mathbf{Jv}_i + \mathbf{Jw}_i^T \mathbf{I}_i \mathbf{Jw}_i \right)$$

با تعریف تابع $MassMatrix$ برای محاسبه ی مقدار ماتریس جرمی، می توانیم مقدار آن را به دست آوریم.

Mass Matrix (M):

$$\begin{pmatrix} \frac{m_3\left(12a_1^2+24\cos(\theta_2)a_1a_2+12\cos(\theta_2+\theta_3)a_1a_3+12a_2^2+\sigma_2+3a_3^2+m_3a_3\right)}{12}+\frac{m_2\left(12a_1^2+12\cos(\theta_2)a_1a_2+3a_2^2+m_2a_2\right)}{12}+\frac{a_1m_1(3a_1+m_1)}{12} & \sigma_1 & \sigma_3 \\ \sigma_1 & \frac{m_3\left(12a_2^2+\sigma_2+3a_3^2+m_3a_3\right)}{12}+\frac{a_2m_2(3a_2+m_2)}{12} & \sigma_4 \\ \sigma_3 & \sigma_4 & \frac{a_3m_3(3a_3+m_3)}{12} \end{pmatrix}$$

where

$$\sigma_1=\frac{m_3\left(a_3m_3+12a_2^2+3a_3^2+6a_1a_3\cos(\theta_2+\theta_3)+12a_1a_2\cos(\theta_2)+12a_2a_3\cos(\theta_3)\right)}{12}+\frac{a_2m_2(3a_2+m_2+6a_1\cos(\theta_2))}{12}$$

$$\sigma_2=12\cos(\theta_3)a_2a_3$$

$$\sigma_3=\frac{a_3m_3(3a_3+m_3+6a_1\cos(\theta_2+\theta_3)+\sigma_5)}{12}$$

$$\sigma_4=\frac{a_3m_3(3a_3+m_3+\sigma_5)}{12}$$

$$\sigma_5=6a_2\cos(\theta_3)$$

شکل ۱

ماتریس گرانش

در گام بعد برای محاسبه ی ماتریس گرانش مطابق رابطه ی ۵٫۶۵ خواهیم داشت:

$$\mathbf{g}=\begin{bmatrix}\text{۰}\\-g\\\text{۰}\end{bmatrix}$$

$$G_i=-m_i\mathbf{J}_{v_i}^T\mathbf{g}$$

$$G=\sum_{i=\text{۱}}^{N}G_i=\sum_{i=\text{۱}}^{N}\left(-m_i\mathbf{J}_{v_i}^T\mathbf{g}\right)$$

با فراخوانی تابع $GravitationalVector$ که بدین منظور نوشته شده است برای ربات $۳R$ خواهیم داشت:

$$\begin{pmatrix} g\,m_۳\left(\sigma_۱ + a_۱\,\cos\left(\theta_۱\right) + \sigma_۲\right) + g\,m_۲\left(\frac{\sigma_۱}{۲} + a_۱\,\cos\left(\theta_۱\right)\right) + \frac{a_۱\,g\,m_۱\,\cos(\theta_۱)}{۲} \\ g\,m_۳\left(\sigma_۱ + \sigma_۲\right) + \frac{a_۲\,g\,m_۲\,\cos(\theta_۱+\theta_۲)}{۲} \\ \frac{a_۳\,g\,m_۳\,\cos(\theta_۱+\theta_۲+\theta_۳)}{۲} \end{pmatrix}$$

where

$$\sigma_۱ = a_۲\,\cos\left(\theta_۱ + \theta_۲\right)$$

$$\sigma_۲ = \frac{a_۳\,\cos(\theta_۱+\theta_۲+\theta_۳)}{۲}$$

ماتریس کوریولیس

در ادامه، ماتریس کوریولیس مطابق روابط ۵/۳۴ و ۵/۳۷ محاسبه می شود. در این روابط داریم:

$$\dot{M}(i,j) = \sum_{k=۱}^{n} \frac{\partial M(i,j)}{\partial \theta_k} \cdot \dot{\theta}_k$$

Time Derivative of the Mass Matrix (M_dot):

$$\begin{pmatrix} -\mathrm{dtheta}_2\left(\frac{m_3\,(24\,a_1\,a_2\sin(\theta_2) + 12\,a_1\,a_3\sin(\theta_2+\theta_3))}{12} + a_1\,a_2\,m_2\sin(\theta_2)\right) - \frac{\mathrm{dtheta}_3\,m_3\,(12\,a_2\,a_3\sin(\theta_3) + 12\,a_1\,a_3\sin(\theta_2+\theta_3))}{12} & \sigma_1 & \sigma_2 \\ \sigma_1 & -a_2\,a_3\,\mathrm{dtheta}_3\,m_3\sin(\theta_3) & \sigma_3 \\ \sigma_2 & \sigma_3 & 0 \end{pmatrix}$$

where

$$\sigma_1 = -\mathrm{dtheta}_2\left(\frac{m_3\,(12\,a_1\,a_2\sin(\theta_2) + 6\,a_1\,a_3\sin(\theta_2+\theta_3))}{12} + \frac{a_1\,a_2\,m_2\sin(\theta_2)}{2}\right) - \frac{\mathrm{dtheta}_3\,m_3\,(12\,a_2\,a_3\sin(\theta_3) + 6\,a_1\,a_3\sin(\theta_2+\theta_3))}{12}$$

$$\sigma_2 = -\frac{a_3\,\mathrm{dtheta}_3\,m_3\,(6\,a_1\sin(\theta_2+\theta_3) + 6\,a_2\sin(\theta_3))}{12} - \frac{a_1\,a_3\,\mathrm{dtheta}_2\,m_3\sin(\theta_2+\theta_3)}{2}$$

$$\sigma_3 = -\frac{a_2\,a_3\,\mathrm{dtheta}_3\,m_3\sin(\theta_3)}{2}$$

شکل ۲

$$\mathbf{B} = \dot{M} \cdot \dot{\theta} - \frac{\partial K}{\partial \theta}$$

```
Coriolis Force Vector (B):
```

$$\begin{pmatrix} -\mathrm{dtheta}_2 \left(\mathrm{dtheta}_2 \left(\frac{m_3\,(12\,a_1\,a_2\sin(\theta_2)+\sigma_2)}{12}+\frac{\sigma_3}{2}\right)+\frac{\mathrm{dtheta}_3\,m_3\,(\sigma_4+\sigma_2)}{12}\right)-\mathrm{dtheta}_1\left(\mathrm{dtheta}_2\left(\frac{m_3\,(24\,a_1\,a_2\sin(\theta_2)+\sigma_1)}{12}+\sigma_3\right)+\frac{\mathrm{dtheta}_3\,m_3\,(\sigma_4+\sigma_1)}{12}\right)-\mathrm{dtheta}_3\left(\frac{a_3\,\mathrm{dtheta}_3\,m_3\,(6\,a_1\sin(\theta_2+\theta_3)+6\,a_2\sin(\theta_3))}{12}+\frac{a_1\,a_3\,\mathrm{dtheta}_2\,m_3\sin(\theta_2+\theta_3)}{2}\right) \\ \frac{a_1\,a_3\,\mathrm{dtheta}_1{}^2\,m_3\sin(\theta_2+\theta_3)}{2}+\frac{a_1\,a_2\,\mathrm{dtheta}_1{}^2\,m_2\sin(\theta_2)}{2}+a_1\,a_2\,\mathrm{dtheta}_1{}^2\,m_3\sin(\theta_2)-\frac{a_2\,a_3\,\mathrm{dtheta}_3{}^2\,m_3\sin(\theta_3)}{2}-a_2\,a_3\,\mathrm{dtheta}_1\,\mathrm{dtheta}_3\,m_3\sin(\theta_3)-a_2\,a_3\,\mathrm{dtheta}_2\,\mathrm{dtheta}_3\,m_3\sin(\theta_3) \\ \frac{a_3\,m_3\,(a_1\,\mathrm{dtheta}_1{}^2\sin(\theta_2+\theta_3)+a_2\,\mathrm{dtheta}_1{}^2\sin(\theta_3)+a_2\,\mathrm{dtheta}_2{}^2\sin(\theta_3)+2\,a_2\,\mathrm{dtheta}_1\,\mathrm{dtheta}_2\sin(\theta_3))}{2} \end{pmatrix}$$

where

$$\sigma_1 = 12\,a_1\,a_3\sin(\theta_2+\theta_3)$$

$$\sigma_2 = 6\,a_1\,a_3\sin(\theta_2+\theta_3)$$

$$\sigma_3 = a_1\,a_2\,m_2\sin(\theta_2)$$

$$\sigma_4 = 12\,a_2\,a_3\sin(\theta_3)$$

شکل ۳

$$K = \frac{۱}{۲}\dot{\theta}^T M \dot{\theta}$$

پاسخ سوال سه

با استفاده از این روابط و توابع نوشته شده برای هر یک که در این برنامه با نام های $MassMatrixTimeDerivative$، $CoriolisForceVector$ و $KineticEnergy$ تعریف شده اند، می توانیم ماتریس های مورد نظر را به دست بیاوریم که در اینجا از ذکر کردن نتایج تمامی مقادیر بالا به دلیل فضای ناکافی اجتناب شده است.

در نهایت، پس از تعریف هر یک از این ماتریس ها، می توانیم معادلات دینامیکی سیستم را طبق رابطه ی زیر به دست بیاوریم.

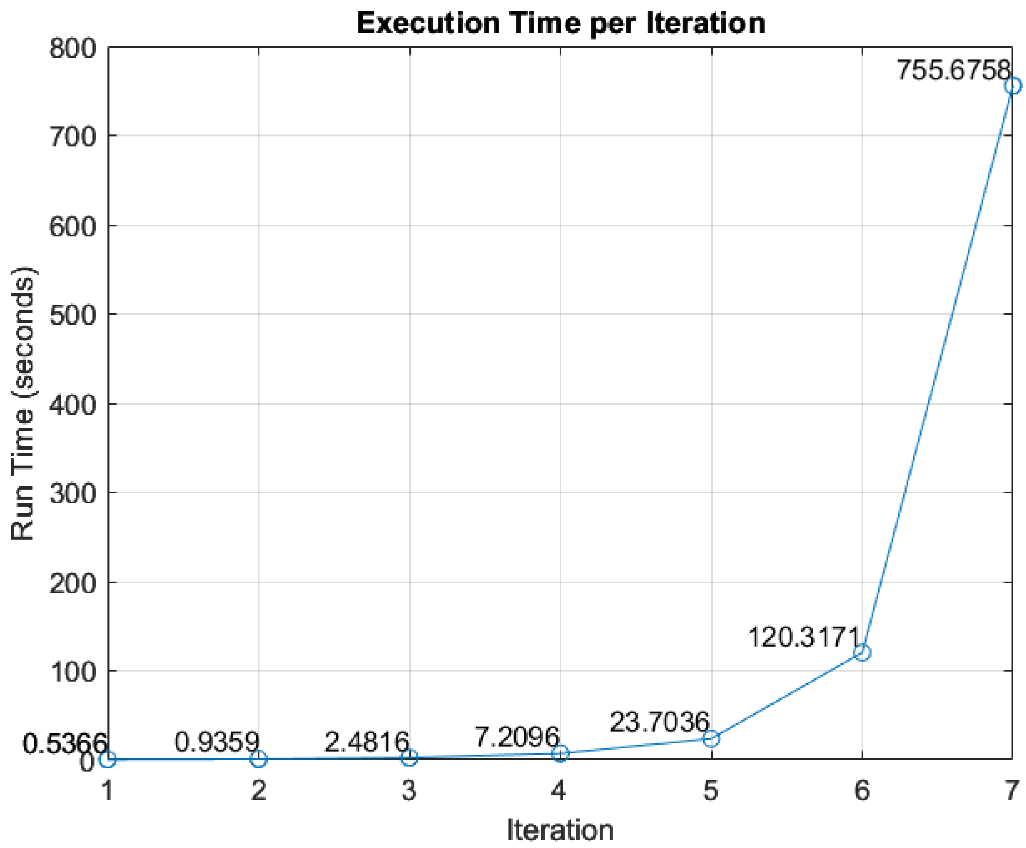
$$Q = B(q, \dot{q}) + G(q) + M(q) \cdot \ddot{\theta}$$

نتیجه ی این ماتریس در فایل متلب نمایش داده شده است.

پاسخ سوال چهار

با اجرای برنامه به ازای تعداد درجات آزادی مختلف از ۱ تا ۷، تلاش می کنیم تا زمان اجرای برنامه را مقایسه کنیم. انتظار می رود که با توجه به حجم بالای محاسبات ماتریسی، با افزایش درجات آزادی زمان اجرای برنامه

به صورت صعودی و نمایی افزایش یابد. برای این کار، از دستور $tictoc$ متلب استفاده شده است. مطابق آنچه که در تصویر زیر نمایش داده شده است، زمان های اجرای برنامه مشخص شده است. لازم به ذکر است که این فرایند بدون استفاده از حلقه ی for صورت گرفته و در نتیجه اعداد گزارش شده هر بار مجزا اندازه گیری شده است. علت این امر، نیاز به تغییر پارامترهای ورودی سیستم از جمله پارامتر های DH بوده که به دلیل پیچیدگی زیاد، از آن صرف نظر شده است



شکل ۴

پاسخ سوال پنج

کسب اطمینان از صحت پاسخ های به دست آمده، همانطور که در متن سوال ذکر شده است به دو روش Verification به معنای تایید و Validation به معنای اعتبارسنجی انجام می گیرد که در ادامه به توضیح هر یک

خواهیم پرداخت.

Verification

منظور از تایید، آن است که مطمئن شویم محاسبات پیاده سازی شده با تئوری های موجود در این زمینه یکسان باشند. بنابراین، روش پیشنهادی برای تایید فرایند آن است که به صورت جز به جز محاسبات، فرمول ها و روابط مورد بررسی قرار بگیرند و در صورت بروز اشتباهات منطقی و یا محاسباتی رفع شوند. همچنین، در هر مرحله با استفاده از نمونه های ساده تر که قبلا درستی آنها اثبات شده است، می توان پاسخ های به دست آمده را با مقادیرر مرجع مقایسه کرد.

Validation

اعتبارسنجی نتایح به دست آمده پس از تایید فرایند انجام می شود. در این تست، هدف آن است که مشخص شود پاسخ های به دست آمده از برنامه که صحت اجرا و محاسبات آنان به درستی انجام شده است، آیا در مثال های واقعی و کاربردی نیز صدق می کند یا خیر. برای این منظور، معمولا نتایج شبیه سازی و محاسبات با نسخه های واقعی از سیستم، نتایج گزارش شده در منابع معتبر نظیر مقالات و یا داده های معتبر مقایسه می شوند و در صورتی که مدل توانسته باشد به خوبی عمل کند، می توان انتظار داشت که نتایج به دست آمده از آن با منابع همخوانی داشته باشد

پاسخ سوال شش

برای محاسبه دینامیک سیستم‌های رباتیکی، روش‌های متعددی وجود دارند که هر یک از نظر کارایی و هزینه محاسباتی متفاوت عمل می‌کنند. یکی از این روش‌ها، روش کلاسیک **اولر-لاگرانژ** است که از توابع انرژی برای استخراج معادلات حرکت استفاده می‌کند. اما با افزایش تعداد لینک‌های ربات، این روش به طور فزاینده‌ای محاسباتی و زمان‌بر می‌شود. به عنوان مثال، زمان اجرای این روش برای تعداد لینک‌های ۱ تا ۷ به ترتیب۰٫۵۳۳، ۰٫۹۳، ۲٫۴۸، ۲٫۷، ۲۳، ۱۲۰ و ۷۵۰ به دست آمده است.

روش‌های جایگزین دیگری نیز وجود دارند که از نظر هزینه محاسباتی عملکرد بهتری دارند:

- **روش نیوتن-اولر:** این روش از قوانین نیوتن و اولر برای محاسبه نیروها و گشتاورها در هر لینک استفاده می‌کند و محاسبات را به صورت بازگشتی انجام می‌دهد. این ساختار بازگشتی باعث افزایش کارایی محاسبات می‌شود و نیاز به محاسبه و وارون‌سازی ماتریس‌های بزرگ که در روش اولر-لاگرانژ معمول است را حذف می‌کند.

- **الگوریتم‌های بازگشتی:** الگوریتم‌هایی مانند **الگوریتم بدنه سخت ترکیبی (CRBA)** و **الگوریتم بدنه مفصل‌دار (ABA)** از روش‌های بازگشتی برای محاسبه دینامیک استفاده می‌کنند. این الگوریتم‌ها دارای پیچیدگی محاسباتی خطی هستند ($O(n)$) و این ویژگی باعث می‌شود که حتی با افزایش تعداد لینک‌ها، زمان محاسبات به صورت خطی افزایش یابد. این روش‌ها برای کاربردهای زمان واقعی بسیار مناسب هستند.

- **انتگرال‌گیرهای واریاسیونی:** این روش‌ها از روش‌های عددی برای حفظ ویژگی‌های هندسی دینامیک سیستم، مانند انرژی و تکانه، استفاده می‌کنند. در سال‌های اخیر، پیشرفت‌هایی در این حوزه صورت گرفته است که منجر به توسعه انتگرال‌گیرهای واریاسیونی با پیچیدگی خطی برای سیستم‌های چندجسمی شده است. این روش‌ها دقت بالا و کارایی محاسباتی را ترکیب می‌کنند و برای شبیه‌سازی‌ها و بهینه‌سازی‌های مسیر بسیار مفید هستند.

با مقایسه این روش‌ها، می‌توان گفت:

- روش اولر-لاگرانژ با وجود ساختار منظم، در سیستم‌های پیچیده و با درجات آزادی بالا، هزینه محاسباتی بالایی دارد.

- روش نیوتن-اولر به دلیل ساختار بازگشتی خود، محاسبات را کارآمدتر انجام می‌دهد و برای کنترل زمان واقعی مناسب‌تر است.

- الگوریتم‌های بازگشتی مانند CRBA و ABA به دلیل پیچیدگی خطی، بهترین عملکرد را در سیستم‌های با تعداد لینک‌های زیاد ارائه می‌دهند.

- انتگرال‌گیرهای واریاسیونی ترکیبی از دقت و کارایی را ارائه می‌دهند و برای کاربردهایی که نیاز به حفظ ویژگی‌های هندسی دینامیک دارند، مناسب هستند.