

به نام خدا



دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

تحلیل و شبیه سازی مکانیزم

ارائه شده برای درس دینامیک ماشین

گردآوری:

مهدی رحمانی

علیرضا ذاکری مطلق

استاد درس:

دکتر علی عظیمی

دانشکده مهندسی مکانیک

۱۳۹۹

چکیده

مکانیزم¹ زنجیره ی سینماتیکی سیستمی از میله ها، یعنی اجام صلب است که یا به هم متصل اند، یا به شکلی با هم تماس دارند که می توانند نسبت به یکدیگر حرکت کنند. ماشین مکانیزمی است که نیرو را انتقال می دهد و اگرچه هر ماشین از مکانیزمی تشکیل شده، همه ی مکانیزم ها ماشین نیستند و کاربرد آنها وسیع تر از این حدود است؛ از این رو تحلیل سینماتیک، یعنی حرکت نسبی اجزای مکانیزم، و دینامیک مکانیزم ها، یعنی تحلیل نیروهای وارده به اجزا و حرکات ناشی از این نیروها، جایگاه ویژه ای دارد. در این گزارش تلاش شده تا حرکت مکانیزمی خاص با استفاده از دو نرم افزار، یکی تحلیلی و دیگری شبیه سازی، مدل شود.

واژگان کلیدی:

برنامه نویسی، شبیه سازی، سینماتیک، دینامیک، سرعت، شتاب

1-mechanism

فهرست مطالب

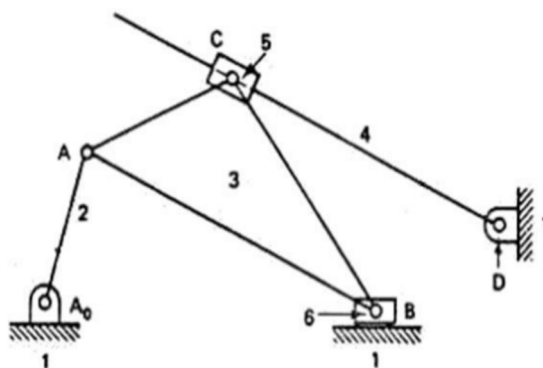
چکیده.....	۱.....
فهرست مطالب.....	ب.....
فهرست اشکال.....	ت.....
فهرست نمودارها.....	ث.....
فصل اول: مقدمه.....	۱.....
معرفی مکانیزم.....	۲.....
فرض های اولیه.....	۳.....
فصل دوم: تحلیل سینماتیکی.....	۴.....
استخراج معادلات.....	۵.....
تحلیل سینماتیکی در متلب.....	۷.....
تحلیل سینماتیکی در شبیه ساز.....	۱۰.....
نتایج.....	۱۱.....
تحلیل خطا.....	۱۸.....
فصل سوم: تحلیل دینامیکی.....	۱۹.....
استخراج معادلات.....	۲۰.....
تحلیل دینامیکی در متلب.....	۲۴.....
تحلیل دینامیکی در شبیه ساز.....	۲۵.....
نتایج.....	۲۷.....
تحلیل خطا.....	۳۳.....

فصل اول:

مقدمه

معرفی مکانیزم

مکانیزم داده شده یک مکانیزم با ۶ لینک می باشد. همان طور که در شکل زیر مشاهده می شود، ۳ میله ای که به صورت مثلثی به یک دیگر متصل شده اند را میتوان یک جسم صلب در نظر گرفت چراکه نقاط روی آن ها نسبت به هم حرکت و سرعتی ندارند؛ دو میله ی به یاتاقان ها متصل شده اند و یک لینک هم زمین می باشد. دو عضو دیگر هم لغزنده می باشند.



شکل ۱: تصویر خام مکانیزم اصلی

هدف پروژه به این صورت است که برای مکانیزم فوق تحلیل های سینماتیکی و دینامیکی لازم انجام شود. در این پروژه تلاش می شود تا سرعت و شتاب در هر لینک و نیروهای وارده در هر اتصال را بررسی کنیم.

ابتدا معادلات سرعت و شتاب را به دست می آوریم و سپس به کمک متلب تحلیل های گفته شده را انجام داده و نمودارهای مربوطه را به دست می آوریم؛ بار دیگر تحلیل را به کمک شبیه ساز انجام می دهیم و در نهایت نتایج به دست آمده را مقایسه می کنیم؛ در مرحله آخر نیز راجع به خطاهای موجود بحث می کنیم.

فرض های اولیه

برای سادگی تحلیل، مکانیزم معرفی شده مکانیزمی با سرعت زاویه ای ورودی ثابت در نظر گرفته شده است؛ بازه حرکتی مکانیزم یک ثانیه و در نتیجه برای سرعت زاویه ای لینک ورودی که لینک شماره دو می باشد $\omega = 2\pi$ حاصل می شود.

در قسمت دینامیکی به منظور منطقی و کاربردی کردن تحلیل، فرض شده نیرویی به اندازه ی 100 N در نقطه B به لغزنده ی شماره ی 6 وارد شود. این نیرو به عنوان نیروی ناشی از گازهای موجود در سیلندری که لغزنده ی 6 پیستون آن است، فرض شده و برای ساده سازی مقدار آن ثابت در نظر گرفته شده است. از طرفی گشتاور اعمالی موتور موجود در یاتاقان لینک دو به عنوان مجهول فرض شده و نمودار آن برای تامین این حرکت با وجود نیروی گاز ها، به عنوان تابعی از زمان، ترسیم شده است.

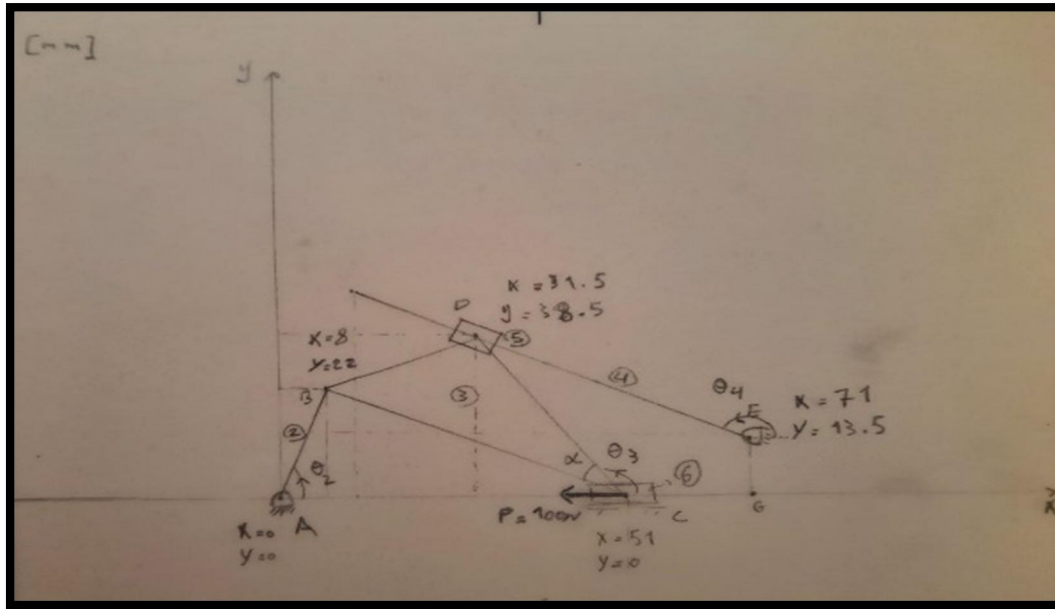
در هر نرم افزار دوره تناوب مکانیزم که مقدار آن یک ثانیه ذکر شد، به 360 المان و بازه زمانی تقسیم شده است تا نمودار ها از وضوح بیشتری برخوردار باشند. باقی فرض ها در ادامه گزارش و به فراخور مطلب بیان می گردد.

فصل دوم

تحليل سينماتيكى

استخراج معادلات

ابتدا مکانیزم را رسم کرده، مختصات گره ها و تکیه گاه ها را مشخص می کنیم. برای این کار مقادیری را فرض می کنیم که در ادامه به آنها می پردازیم:



شکل ۲: مختصات اولیه-پارامترهای سینماتیکی

مشاهده می کنید که نقطه A به عنوان مبدا مختصات در نظر گرفته شده و مختصات سایر نقاط نسبت به آن سنجیده می شود. ادامه ی کار با توجه به نام گذاری ها و مقدار دهی هایی که در شکل آمده است، انجام می شود. در زیر طول لینک ها و داده های ثابت آورده شده است:

$$AB = 23.4094 \text{ mm} \quad BC = 48.30114 \text{ mm} \quad BD = 28.71411 \text{ mm}$$

$$CD = 43.15669 \text{ mm} \quad EG = 13.5 \text{ mm}$$

هم چنین نقطه ی شروع حرکت مکانیزم از زاویه ی $\theta_2 = 1.222 \text{ rad}$ می باشد. سایر مقادیر اولیه در این زاویه ی ورودی در زیر آمده است:

$$AC = 51 \text{ mm}$$

$$ED = 46.74666 \text{ mm}$$

$$\theta_3 = 2.0396 \text{ rad}$$

$$\theta_4 = 2.5773 \text{ rad}$$

در مرحله ی بعد لازم است با نوشتن معادلات **vector loop** و مشتق گرفتن از آن ها معادلات سرعت و شتاب را برای لینک ها بیابیم:

لوپ اول بر روی مسیر بسته ی ABC نوشته می شود و معادلات مربوط به آن در زیر آمده است:

$$1) AB \times \cos(\theta_2) + BC \times \cos(\pi - 0.629 - \theta_3) = AC$$

$$2) AB \times \sin(\theta_2) = BC \times \sin(\pi - 0.629 - \theta_3)$$

لوپ دوم بر روی مسیر بسته ی CDEG نوشته میشود و معادلات مربوط به آن مطابق زیر است:

$$1) CD \times \cos(\theta_3) = ED \times \cos(\theta_4) + CG$$

$$2) CD \times \sin(\theta_3) = ED \times \sin(\theta_4) + EG$$

به این ترتیب چهار معادله و چهار مجهول وجود دارند که عبارتند از θ_3 ، θ_4 ، AC ، ED ؛ با توجه به دانستن حرکت ورودی، در هر لحظه با داشتن زاویه لینک دو و داده های اندازه گیری شده ی مسئله، با این معادلات مقادیر مجهول را مشخص می کنیم.

همچنین معادلات مربوط به مشتقات اول و دوم از معادلات فوق که به ترتیب معادلات سرعت و شتاب را می دهند در پیوست آمده است. در آن معادلات نیز مشتقات این مجهولات به عنوان مجهول وجود دارند که با توجه به دانستن حرکت لینک ورودی و با جاگذاری مقادیر سرعت و شتاب آن و داده های حاصل از معادلات سری اول، این مجهولات را می یابیم.

تحلیل سینماتیکی در متلب:

برای تحلیل سینماتیکی در متلب باید گام به گام به صورت زیر پیش برویم.

ابتدا با کمک معادلات به دست آمده از نوشتن لوپ اول و دوم باید مقادیر AC ، ED ، θ_4 و θ_3 را در هر لحظه به دست آوریم. هم چنین مقدار ω_2 را ثابت و برابر با $2\pi \frac{rad}{sec}$ و کل بازه ی زمانی شبیه سازی برابر با 1sec در نظر گرفته ایم. بنابراین مقدار θ_2 را می توانیم در هر لحظه به کمک رابطه ی $\theta_2 = \omega_2 \times t$ به دست آوریم. لازم به یادآوری است که بازه ی یک ثانیه را به ۳۶۰ قسمت تقسیم کرده ایم.

حال به کمک دستور `fsolve` متلب و با دادن مقادیر اولیه و ۴ معادله ۴ مجهول غیرخطی فوق به آن می توانیم نتایج دلخواه را به دست آوریم. در زیر نمونه کد مربوط به این قسمت آورده شده است.

```
%we concider (AC, ED, teta3 , teta4) as (x(1),x(2),x(3),x(4))
%the initial value of teta2
teta2 = 1.2220;
%the initial values are achieved by solving equation with (teta2 = 70.0169 deg)
firstGuess = [0.051;0.04674666;2.0396;2.5773];
t = linspace(0,1,361);
ans1 = [0;0;0;0];
i = 1;
while teta2 <= 7.5052
    F1 = @(x)[ AB * cos(teta2) + BC * cos(pi - a -x(3)) - x(1);
              AB * sin(teta2) - BC * sin(pi - a -x(3));
              CD * cos(x(3)) - x(2) * cos(x(4)) - 0.071 + x(1);
              CD * sin(x(3)) - EG - x(2) * sin(x(4))];
    x = fsolve(F1,firstGuess);
    ans1(:,i) = x;
    firstGuess = x;
    i = i + 1;
    teta2 = teta2 + 2*pi/360;
end
```

نتایج حاصل از این قسمت در ماتریس `ans1` نگهداری می شود؛ به شکلی که مقادیر AC ، ED ، θ_3 و θ_4 به ترتیب در سطرها ی ۱ تا ۴ آن قرار بگیرند.

به همین ترتیب به کمک نتایج به دست آمده از این قسمت و معادلات مشتق مرتبه اول به دست آمده در مراحل قبل، می توان بار دیگر با جایگذاری آن ها در دستور `fsolve` متلب مقادیر \vec{AC} ، \vec{ED} ، $\dot{\theta}_3$ و $\dot{\theta}_4$ را در هر لحظه به دست آورد. این مقادیر نیز به ترتیب در سطرها ی ۱ تا ۴ ماتریس `ans2` نگهداری می شوند.

سپس به کمک نتایج به دست آمده از دو قسمت قبل و معادلات مشتق مرتبه دوم به دست آمده در مراحل قبل، می توان بار دیگر با جایگذاری آن ها در دستور fsolve متلب مقادیر \ddot{AC} ، \ddot{ED} ، $\ddot{\theta}_3$ و $\ddot{\theta}_4$ را در هر لحظه به دست آورد. این مقادیر نیز به ترتیب در سطرهای ۱ تا ۴ ماتریس ans3 نگهداری می شوند.

نقطه ای روی لینک شناور :

در این قسمت بردار r که بردار مکان لغزنده ۵ هست را تعریف کرده ایم و به کمک مشتق گیری از آن، به معادلات سرعت و شتاب نقطه ی D_5 که روی لینک شناور ۳ نیز می باشد، رسیدیم. بردار r برداری است که مبدأ آن روی A و مقصد آن متصل به همان نقطه ی D_5 می باشد. در زیر معادلات مربوط به $\overrightarrow{r_{x D_5}}$ و $\overrightarrow{r_{y D_5}}$ آمده است.

$$\overrightarrow{r_{x D_5}} = AG + ED \times \cos(\theta_4)$$

$$\overrightarrow{r_{y D_5}} = GE + ED \times \sin(\theta_4)$$

همچنین معادلات حاصل از مشتقات مرتبه اول و مرتبه دوم معادلات فوق در پیوست آمده است.

حال با توجه به اینکه مجهولات لازم از معادلات وکتور لوپ را به دست آوردیم و این مقادیر را در ماتریس های ans2 و ans3 نگهداری کردیم، اکنون می توانیم با جاگذاری آن در مشتقات به دست آمده برای لینک ۵، سرعت و شتاب این نقطه را در لحظات مختلف در راستاهای x و y به دست آوریم. هم چنین در نهایت می توان مقدار مطلق آن ها را نیز به دست آورد.

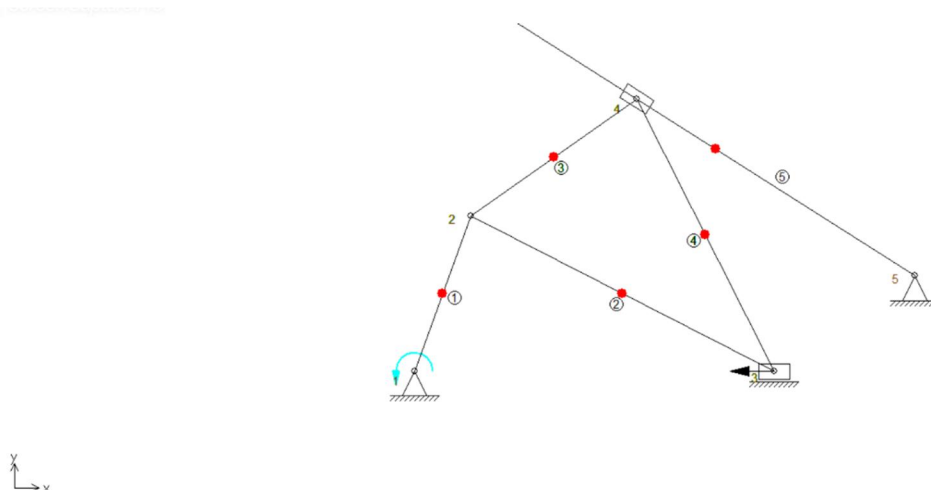
کد مربوط به این قسمت در زیر آمده است:

```
124 -   teta2 = 1.2220;
125 -   VFloatingAns = zeros(1,361);
126 -   AFloatingAns = zeros(1,361);
127 -
128 -   VFloatingAnsX = zeros(1,361);
129 -   VFloatingAnsY = zeros(1,361);
130 -   AFloatingAnsX = zeros(1,361);
131 -   AFloatingAnsY = zeros(1,361);
132 -   k = 1;
133 -   while teta2 <= 7.5052
134 -       Vx = ans2(2,k) * cos(ans1(4,k)) - ans1(2,k) * ans2(4,k) * sin(ans1(4,k));
135 -       Vy = ans2(2,k) * sin(ans1(4,k)) + ans1(2,k) * ans2(4,k) * cos(ans1(4,k));
136 -       VFloatingAnsX(k) = Vx;
137 -       VFloatingAnsY(k) = Vy;
138 -       VFloatingAns(k) = sqrt((Vx)^2 + (Vy)^2);
139 -
140 -       Ax = ans3(2,k) * cos(ans1(4,k)) - 2*ans2(2,k) * ans2(4,k) * sin(ans1(4,k)) - ans1(2,k) * ans3(4,k) * sin(ans1(4,k)) - ans1(2,k) * ((ans2(4,k))^2) * cos(ans1(4,k));
141 -       Ay = ans3(2,k) * sin(ans1(4,k)) + 2*ans2(2,k) * ans2(4,k) * cos(ans1(4,k)) + ans1(2,k) * ans3(4,k) * cos(ans1(4,k)) - ans1(2,k) * ((ans2(4,k))^2) * sin(ans1(4,k));
142 -       AFloatingAnsX(k) = Ax;
143 -       AFloatingAnsY(k) = Ay;
144 -       AFloatingAns(k) = sqrt((Ax)^2 + (Ay)^2);
145 -
146 -       k = k + 1;
147 -       teta2 = teta2 + 2*pi/360;
148 -   end
```

تحلیل سینماتیکی در شبیه ساز:

برای شبیه سازی این مکانیزم از نرم افزار SAM6.1 استفاده کردیم. نرم افزار SAM یکی از نرم افزار های پرکاربرد در مجموعه گروه مکانیک می باشد. با کمک این نرم افزار می توان انواع مکانیزم های صفحه ای مکانیکی و رباتیکی را ترسیم کرد؛ همچنین می توانید به آنالیز، تجزیه و تحلیل و رسم کامل مکانیزم مورد نظرتان یا قسمت خاصی از آن پرداخت.

مدل شبیه سازی شده برای تحلیل در نرم افزار به شکل زیر می باشد:



شکل ۳: مکانیزم شبیه سازی شده

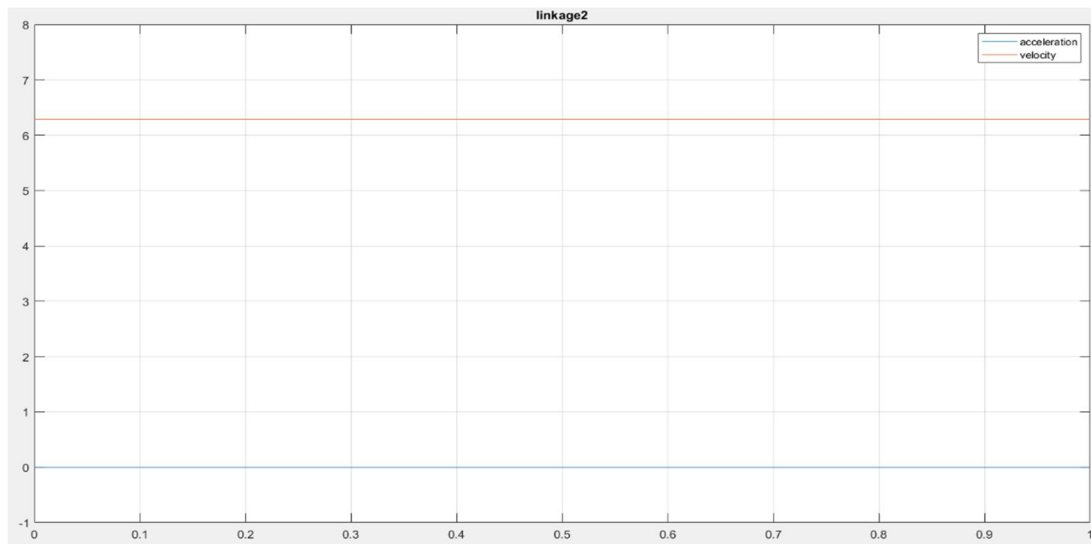
ابتدا مختصات گره ها و تکیه گاه ها مطابق آنچه ذکر شد، داده شده و سپس قید های fix برای نقاط A و B (مرجع به شکل ۲)، قید لغزش در راستای X برای نقطه C و مسیر لغزنده برای نقطه D مشخص شده است. در پیوست فیلمی از شبیه سازی حرکت مکانیزم در این نرم افزار ارسال شده است. همچنین در فایل مربوط به نرم افزار که در پیوست همراه گزارش ارسال می شود می توان تغییرات پیش آمده در نمودار سرعت هنگام گذر مکانیزم از یک شرایط خاص (نقطه مرگ) را مشاهده کرد که بار آموزشی دارد.

نتایج:

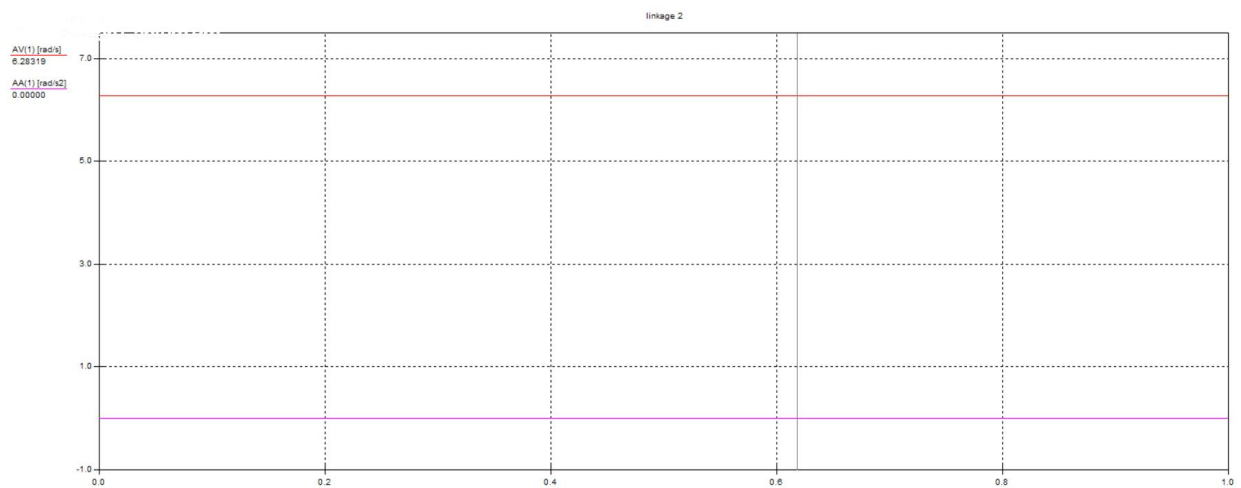
در این قسمت نمودار های مطلوب که با متلب و شبیه ساز ترسیم شد، آورده شده است. در قسمت اول نمودار های مربوط به سرعت و شتاب هرلینک مقایسه شده است. در گام بعدی نیز خطای لحظه ای دلخواه در قیاس دو نمودار محاسبه شده است.

لازم به ذکر است که سرعت و شتاب لینک ۵ با سرعت و شتاب نقطه ی D روی لینک شناور برابر است و توضیح و تحلیل مربوط به آن در قسمت بعدی آورده شده است.

لینک ۲-ورودی:



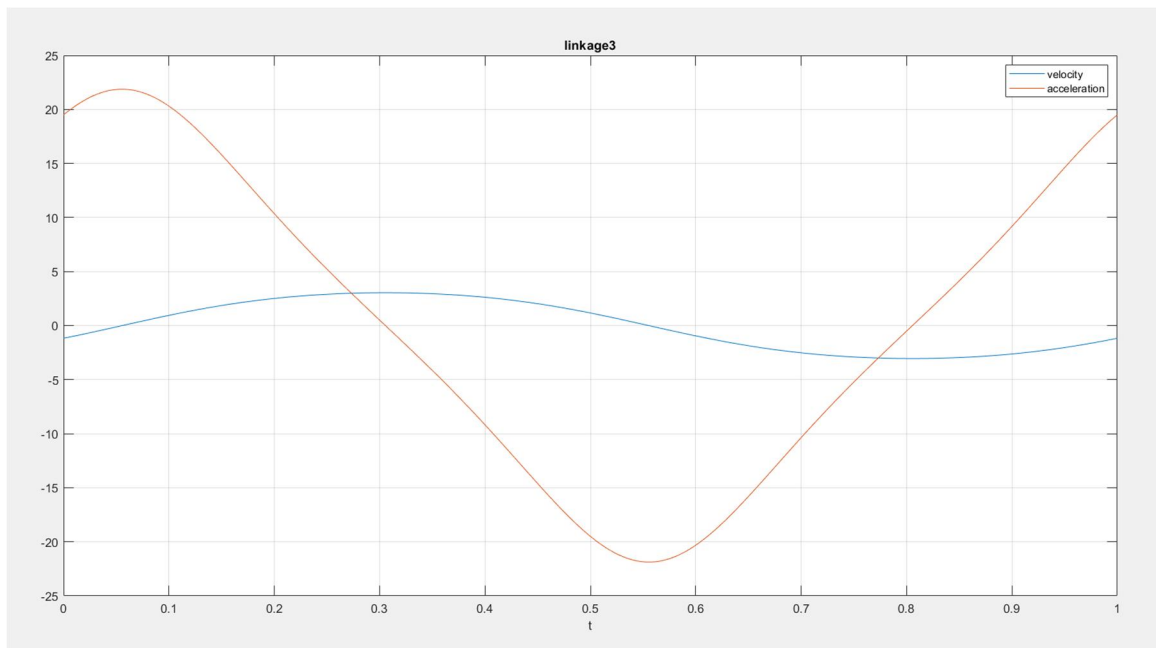
نمودار ۱: سرعت و شتاب لینک ۲ در متلب



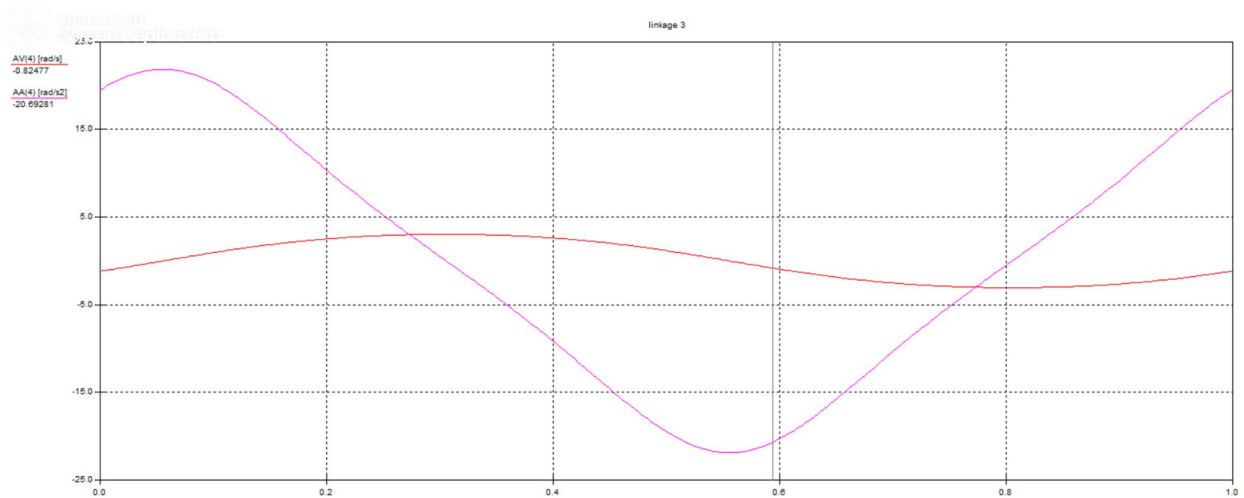
نمودار ۲: سرعت و شتاب لینک ۲ در شبیه ساز

که این لینک ورودی می باشد و در هر دو نرم افزار مقدار سرعت زاویه ای آن چنانچه ذکر شد برابر 2π و مقدار شتاب زاویه ای آن صفر است.

لینک ۳:



نمودار ۳: سرعت و شتاب لینک ۳ در متلب



نمودار ۴: سرعت و شتاب لینک ۳ در شبیه ساز

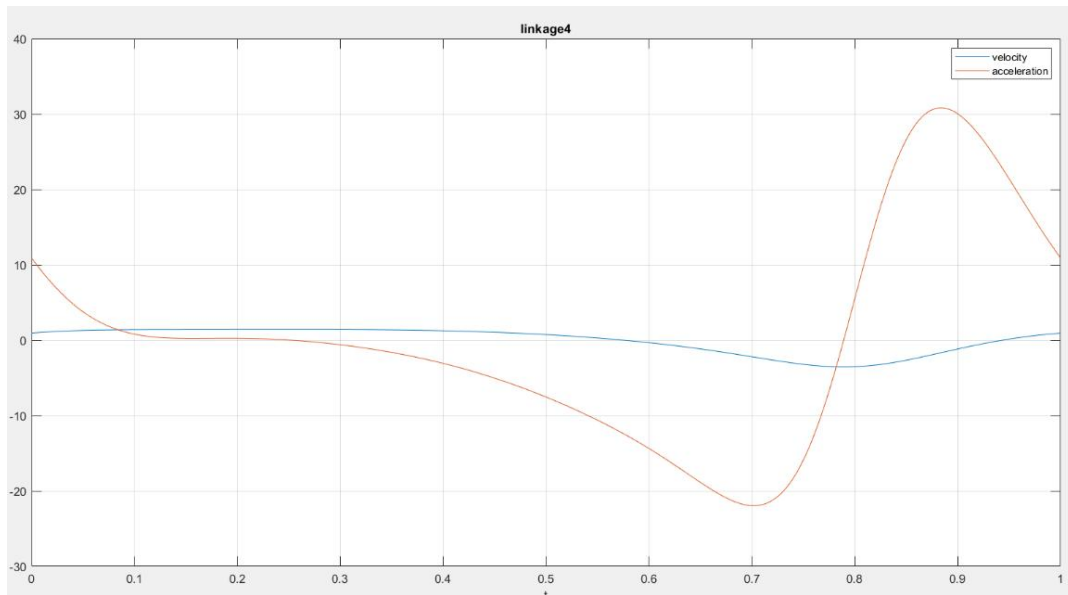
مقایسه و محاسبه خطا در لحظه $t=0.6s$

Matlab: $\omega_3 = -0.947 \text{ rad/s}$ $\alpha_3 = -20.30 \text{ rad/s}^2$

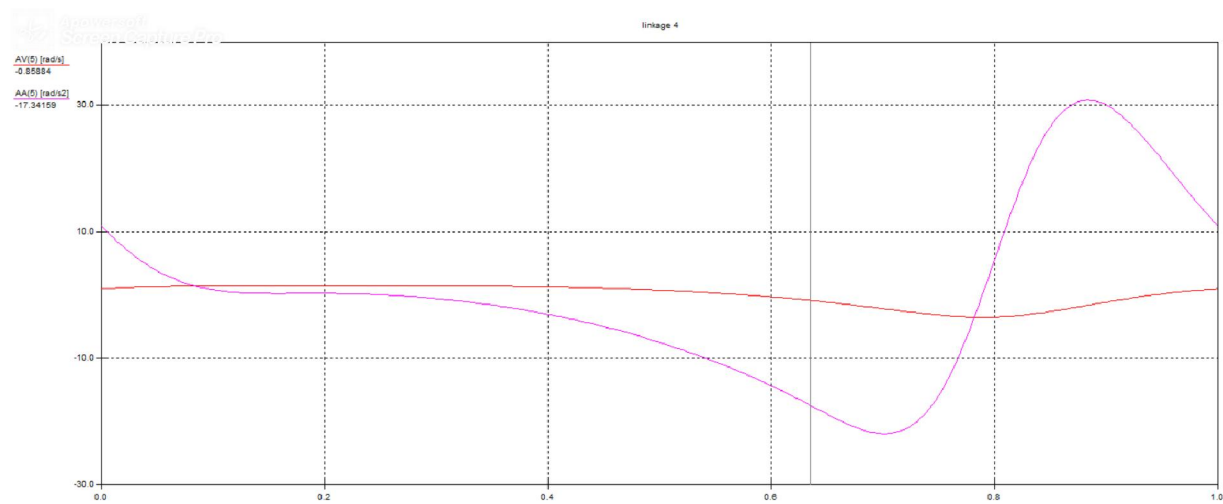
SAM: $\omega_3 = -0.943 \text{ rad/s}$ $\alpha_3 = -20.19 \text{ rad/s}^2$

V.E% = 0.4% A.E% = 0.5%

لینک ۴:



نمودار ۵: سرعت و شتاب لینک ۴ در متلب



نمودار ۶: سرعت و شتاب لینک ۴ در شبیه ساز

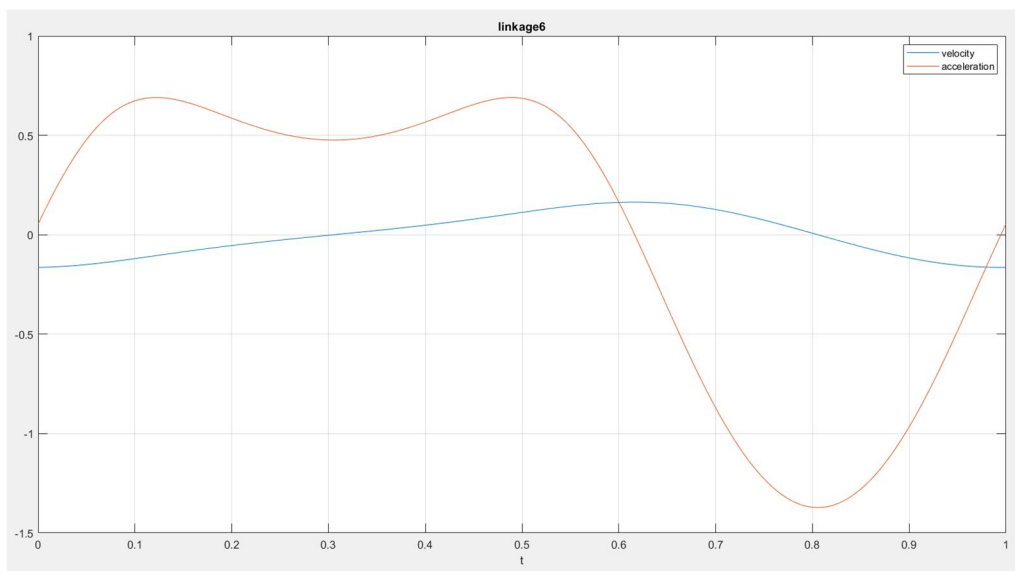
مقایسه و محاسبه خطا در لحظه $t=0.6s$

Matlab: $\omega_3 = -0.3290 \text{ rad/s}$ $\alpha_3 = -14.3672 \text{ rad/s}^2$

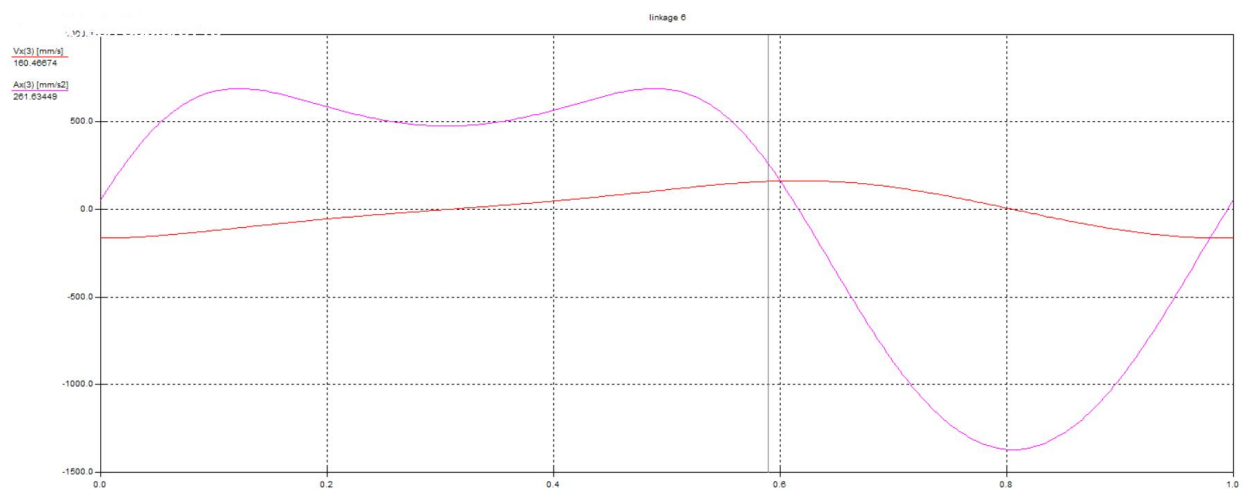
SAM: $\omega_3 = -0.3368 \text{ rad/s}$ $\alpha_3 = -14.4009 \text{ rad/s}^2$

V.E% = 0.8% A.E% = 0.2%

لینک ۶:



نمودار ۷: سرعت و شتاب لینک ۶ در متلب



نمودار ۸: سرعت و شتاب لینک ۶ در شبیه ساز

مقایسه و محاسبه خطا در لحظه $t=0.6s$

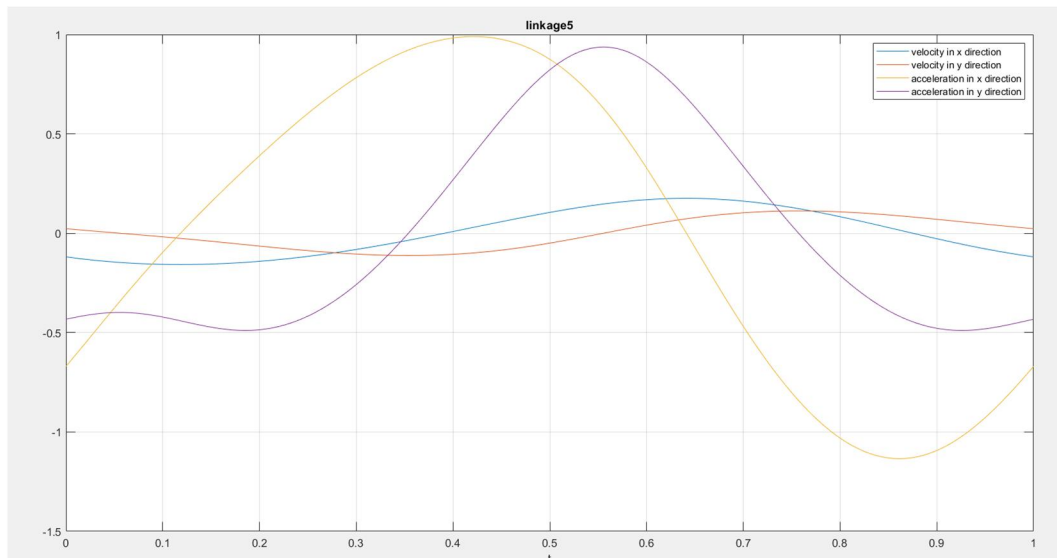
Matlab: $V_x = 0.1635 \text{ m/s}$ $A_x = 0.1627 \text{ m/s}^2$

SAM: $V_x = 0.1627 \text{ m/s}$ $A_x = 0.1634 \text{ m/s}^2$

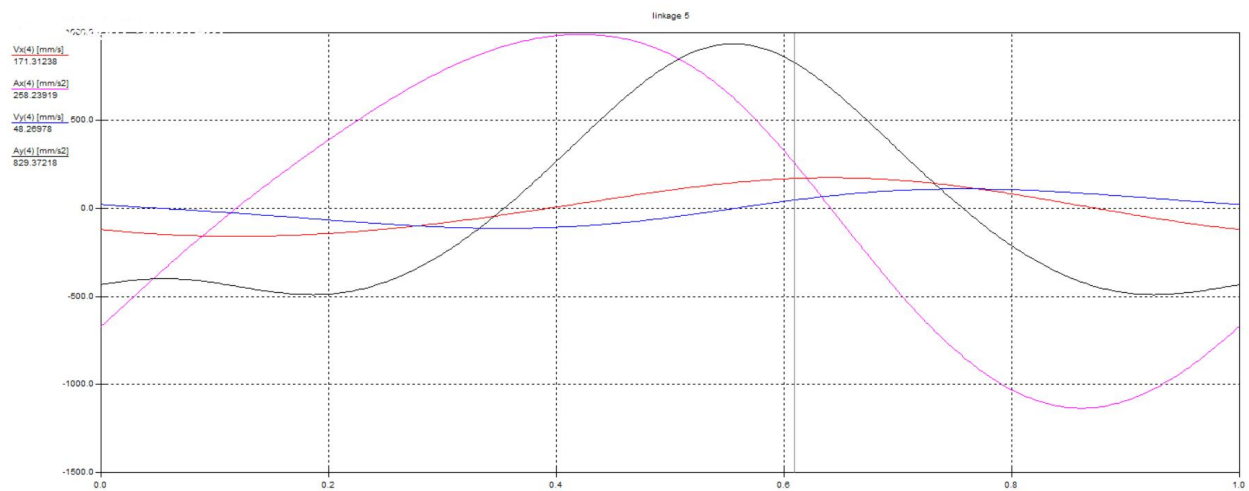
V.E%= 0.5% A.E%= 0.43%

لینک ۵-نقطه روی لینک شناور:

در این قسمت دو مولفه ی سرعت و شتاب آن را ترسیم می کنیم و در قسمت بعد سرعت مطلق و شتاب مطلق را ترسیم کرده و خطای کل را محاسبه می کنیم.

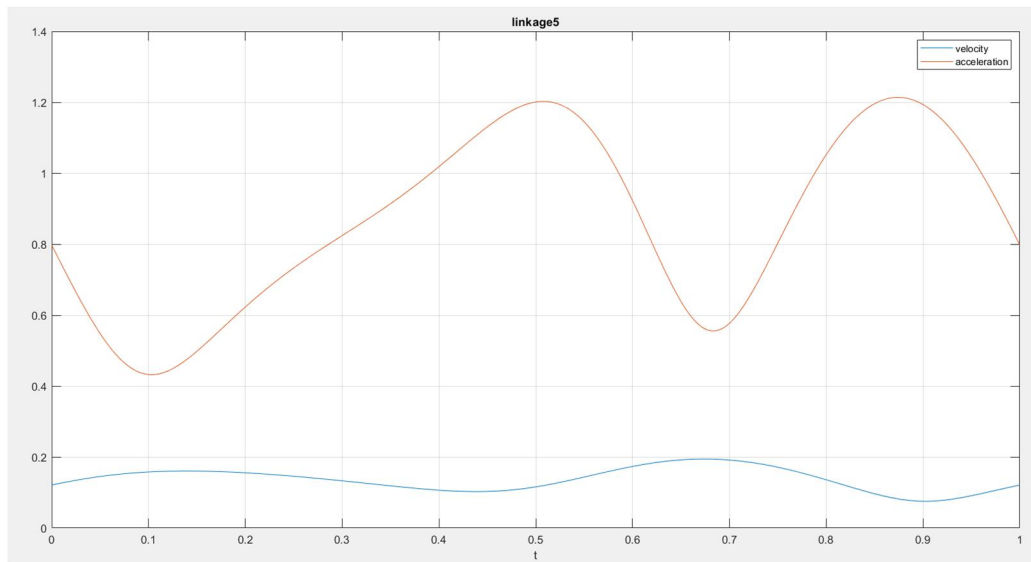


نمودار ۹: سرعت و شتاب لینک ۵ در متلب

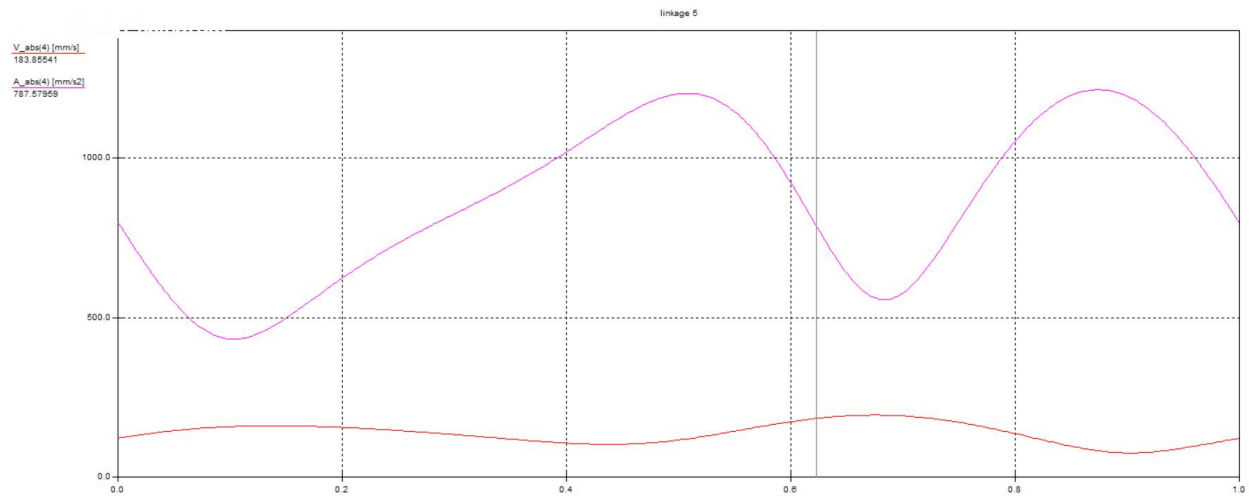


نمودار ۱۰: سرعت و شتاب لینک ۵ در شبیه ساز

لینک ۵-نقطه روی لینک شناور(سرعت و شتاب مطلق):



نمودار ۱۱: سرعت و شتاب مطلق لینک ۵ در متلب



نمودار ۱۲: سرعت و شتاب مطلق لینک ۵ در شبیه ساز

مقایسه و محاسبه خطا در لحظه $t=0.6s$

Matlab: $V_{abs} = 0.1734 \text{ m/s}$

$A_{abs} = 0.9221 \text{ m/s}^2$

SAM: $V_{abs} = 0.1734 \text{ m/s}$

$A_{abs} = 0.9221 \text{ m/s}^2$

$V.E\% = 0.0\%$

(توضیح علت این اتفاق در صفحه بعد)

$A.E\% = 0.0\%$

تحلیل خطا:

خطاهای محاسبه شده به قدری کم هستند که بتوان گفت مقادیر بدست آمده صحیح و روند حل و شبیه سازی درست است اما در این قسمت علت همین خطاهای ناچیز را بررسی می کنیم.

اولین خطای موجود در این فرآیند خطای انسانی در انتقال داده ها می باشد؛ یعنی رقم های اعشاری که در انتقال داده های اولیه از نرم افزار محاسباتی به نرم افزار تحلیلی (و بالعکس) به اشتباه توسط برنامه نویس یا طراح، گرد یا حذف شده اند؛ اگرچه این خطا ها ناچیز هستند و با افزایش دقت و کیفیت پروژه به صفر میل می کنند اما در فرایند مشتق گیری و قرار گیری در معادلات با خطاهای دیگر جمع شده و به مرور زیاد تر می شوند.

خطای دیگر خطای محاسباتی موجود در کد پایه نرم افزار ها (به عنوان مثال خطای موجود در روش نیوتون-رافسون در نرم افزار متلب که از آن برای حل معادله ها استفاده کردیم) هست که مقدار آن نیز بسیار ناچیز است و توضیحات مربوط به آن به فراخور مطلب وجود دارد که در این مقال نمی گنجد.

و اما مهمترین خطا که تقریباً به تنهایی عامل خطاهایی است که مشاهده کردید؛ ذکر شد که در هر دو نرم افزار بازه زمانی یک ثانیه به ۳۶۰ قسمت تقسیم شده است؛ در نرم افزار شبیه ساز، نمودار ترسیمی، نمودار درونیایی این ۳۶۰ لحظه می باشد که با توقف در هر لحظه، می توان مقدار نمودار مربوطه را در آن لحظه خواند؛ در حالی که در متلب، با وجود این که نمودار مانند شبیه ساز درونیایی شده و سپس ترسیم شده، اما نمی توان از آن مقادیر را در لحظه ای خاص خواند؛ به همین علت باید در ماتریس جواب هایی که در نهایت به رسم نمودار منجر شده است، دنبال مقدار معادل بگردیم. تبدیل زمان به آرایه معادلش در ماتریس پاسخ ها منجر به قرار گیری ما بین دو آرایه می شود؛ در محاسبه ی خطاهایی که مشاهده کردید برای سادگی فرایند یکی از دو خانه انتخاب و خوانده شد؛ این موضوع باعث شد که در نمودار سرعت و شتاب لغزنده ۵ در ظاهر خطایی نداشته باشیم اما در واقع و با درونیایی بین دو نقطه مقدار ناچیزی خطا به وجود می آید. خطای بیش از حد در سرعت زاویه ای لینک ۴ هم به همین علت است. با درونیایی خطی بین مقادیر دو آرایه خطا صحیح تر از موارد محاسبه شده و با درونیایی با درجات بالاتر این خطا عملاً صفر می شود و تنها خطاهای انسانی باقی می ماند.

فصل سوم:

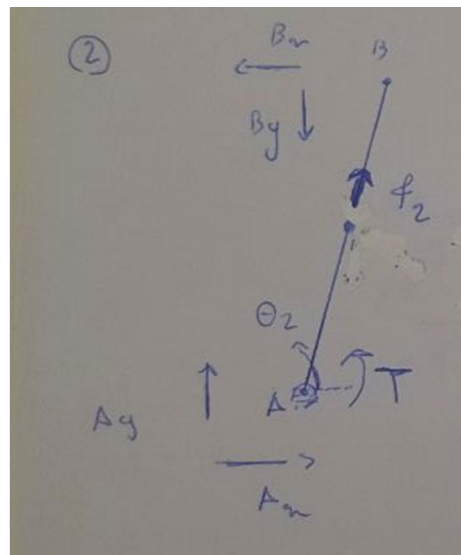
تحليل ديناميكي

استخراج معادلات:

برای تحلیل نیرویی این قسمت ابتدا باید هریک از لینک ها را جداگانه تحلیل کنیم و با کشیدن دیاگرام آزاد فرمول های مربوط به آن را به دست بیاوریم.

لینک ۲:

دیاگرام آزاد مربوط به لینک ۲ به صورت زیر میباشد:



شکل ۴: دیاگرام آزاد لینک ۲

معادلات مربوط به لینک دو را به دست می آوریم و پس از ساده سازی به عبارات زیر میرسیم:

$$1) -B_x + A_x = -f_2 \cos(\theta_2)$$

$$2) -B_y + A_y = -f_2 \sin(\theta_2)$$

$$3) -AB \sin(\theta_2) B_x + AB \cos(\theta_2) B_y - T = 0$$

مجهولات ما T و B_y و B_x و A_x و A_y میباشد.

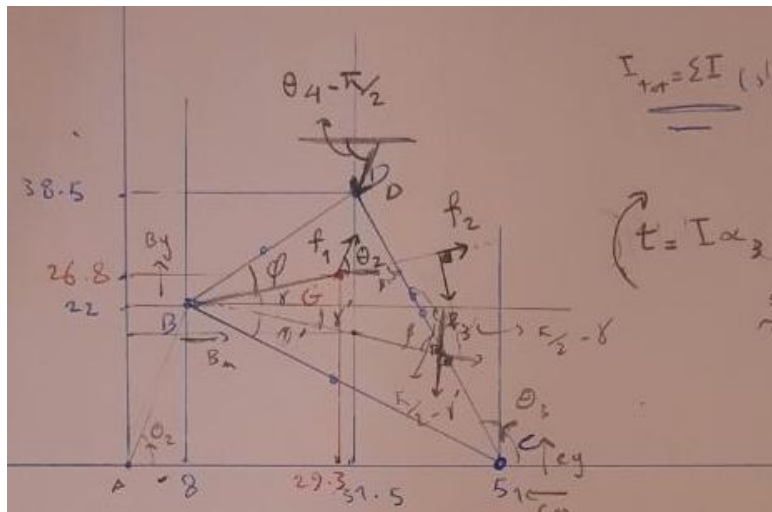
نیروی f_2 نیروی اینرسی لینک دو به جرم 2kg و با مرکز جرمی در وسط آن می باشد:

$$f_2 = 0.5 AB m_2 \omega_2^2$$

گشتاور اینرسی این لینک به علت صفر بودن شتاب زاویه ای آن صفر است. همچنین صفر بودن شتاب زاویه ای سبب صفر شدن شتاب مماسی مرکز جرم شده و در نیروی اینرسی اثر می گذارد.

لینک ۳:

دیاگرام آزاد مربوط به این لینک به صورت زیر می باشد:



شکل ۵: دیاگرام آزاد لینک ۳

لینک ۳ درواقع ۳ میله ی متصل به هم می باشد که ما آن را یک جسم صلب با جرم 8kg، ممان اینرسی 0.02536 kgm^2 و مرکز ثقلی مشخص درنظر گرفته ایم. در ادامه بیشتر راجع به این لینک توضیح می دهیم. با نوشتن روابط و ساده سازی در نهایت به معادلات زیر میرسیم:

$$4) B_x - \cos\left(\theta_4 - \frac{\pi}{2}\right) F_D - C_x = F_3 \cos\left(\frac{\pi}{2} - \gamma\right) - F_2 \cos(\gamma) - F_1 \cos(\theta_2)$$

$$5) B_y - \sin\left(\theta_4 - \frac{\pi}{2}\right) F_D + C_y = F_3 \cos\left(\frac{\pi}{2} - \gamma\right) - F_2 \cos(\gamma) - F_1 \cos(\theta_2)$$

$$6) \left[\cos\left(\theta_4 - \frac{\pi}{2}\right) BD \sin(\varphi - \gamma) - \sin\left(\theta_4 - \frac{\pi}{2}\right) BD \cos(\varphi - \gamma) \right] F_D \\ - BC \sin(\pi - 0.629 - \theta_3) C_x + BC \cos(\pi - 0.629 - \theta_3) C_y \\ = I \alpha_3 - F_3 BG - F_1 \sin(\theta_2) BG$$

به مجهولات F_D و C_x و C_y اضافه میشوند.

شتاب مرکز جرم به صورت نسبی و نسبت به نقطه B محاسبه شده است.

در زیر مقادیر مربوط به F_1 و F_2 و F_3 و I و γ و φ آورده شده است:

$$F_1 = m AB \omega_2^2$$

$$F_2 = m BG \omega_3^2$$

$$F_3 = m BG \alpha_3$$

$$\gamma = \pi - 0.8552 - \theta_3$$

$$\varphi = 0.7854 \text{ rad}$$

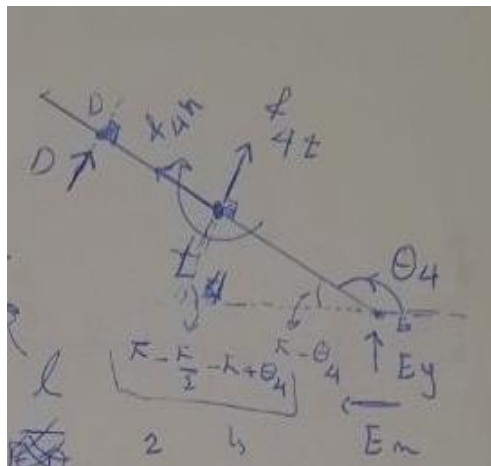
لینک ۴

این لینک دارای یک معادله و یک مجهول است و نتیجه آن در معادلات حساب شده است.

برای این لغزنده جرم و ممانی قائل نشده ایم.

لینک ۵:

دیاگرام آزاد لینک ۵ به صورت زیر می باشد:



شکل ۶: دیاگرام آزاد لینک ۴

این لینک دارای جرم 4kg و ممان اینرسی 0.045kgm^2 می باشد. مرکز ثقل این میله در وسط آن در نظر گرفته شده است.

فرمول های مربوط به آن پس از ساده سازی به صورت زیر می باشد:

$$7) E_x - F_D \cos(\theta_4 - \frac{\pi}{2}) = \left(m_4 \frac{l}{2} \alpha_4\right) \cos(\theta_4 - \frac{\pi}{2}) - \left(m_4 \frac{l}{2} \omega_4^2\right) \cos(\pi - \theta_4)$$

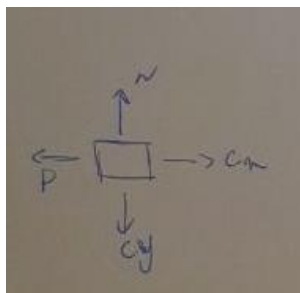
$$8) E_y - F_D \sin(\theta_4 - \frac{\pi}{2}) = -\left(m_4 \frac{l}{2} \alpha_4\right) \sin(\theta_4 - \frac{\pi}{2}) - \left(m_4 \frac{l}{2} \omega_4^2\right) \sin(\pi - \theta_4)$$

$$9) I\alpha + ED F_D = -\frac{l}{2} \left(m_4 \frac{l}{2} \alpha_4\right)$$

مجهولات E_x و E_y اضافه می شوند.

لینک ۶:

دیاگرام آزاد مربوط به آن به صورت زیر میباشد:



شکل ۷: دیاگرام آزاد لینک ۶

فرمول های آن پس از ساده سازی به صورت زیر است:

$$10) N = C_y$$

$$11) C_x = P$$

در این مرحله نیز مجهول N اضافه می شود. لازم به ذکر است که مقدار P معلوم و برابر با $100N$ می باشد.

برای این لغزنده نیز جرم و ممانی قائل نشده ایم.

حال برای حل این دستگاه خطی ۱۱ معادله ۱۱ مجهول به سراغ متلب میرویم.

تحلیل دینامیکی در متلب:

با توجه به معادلات به دست آمده، چون معادلات خطی هستند پس به راحتی با به دست آوردن ماتریس ضرایب و بردار ستونی جواب ها و قرار دادن آن ها در دستور `linesolve` متلب میتوان بردار جواب ها را یافت.

کدی که در متلب زده شده است را میتوانید در زیر مشاهده کنید:

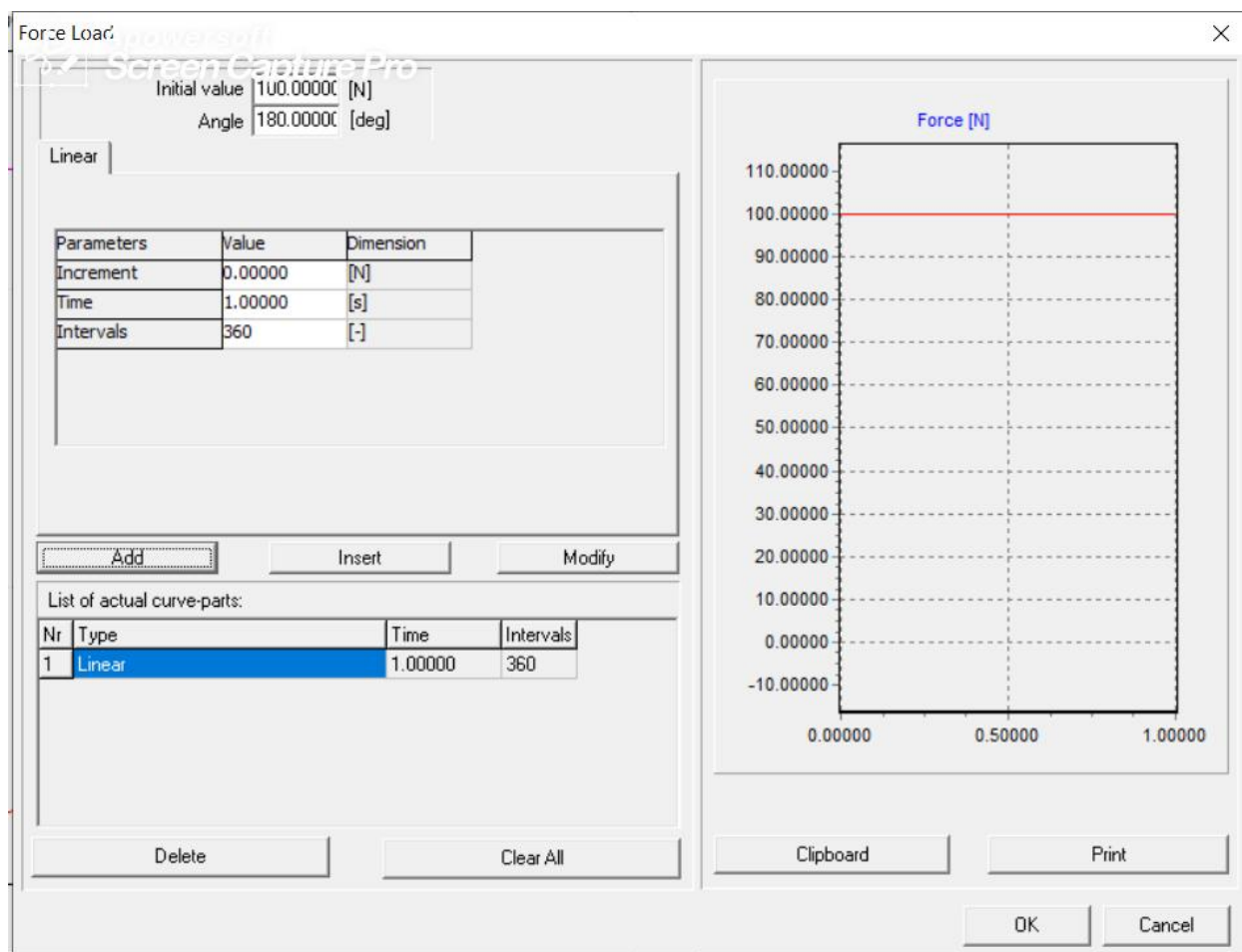
```

184 %now we want to do the force analysis
185 g = 9.81;
186 p = 100;
187 teta2 = 1.2220;
188 k = 1;
189 forceAnalysis = zeros(11,1);
190 %data for second linkage
191 f0 = 2 * (AB/2) * (w2)^2;
192 while teta2 <= 7.5052
193     %data for third linkage
194     m = 8;
195     I3 = 0.02536;
196     BG = 0.029;
197     f1 = m * AB * (w2)^2;
198     f2 = m * BG * (ans2(3,k))^2;
199     f3 = m * BG * (ans3(3,k));
200     gama = pi - ans1(3,k) - 0.8552;%B= 49 deg
201     phy = 0.7854; % 45 deg
202     %data for forth linkage
203     l = 0.04674;
204     I4 = 0.045;
205     m4 = 4;
206     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Ax Ay Bx By T Fd Ex Ey N Cy Cx
207     CoefficientMatrix = [1 ,0 , -1 ,0 , 0 , 0 , 0 , 0 , 0 , 0 , 0;
208                          0 ,1 ,0 , -1 ,0 , 0 , 0 , 0 , 0 , 0 , 0;
209                          0 ,0 , -AB*sin(teta2),AB*cos(teta2),-1,0,0,0,0,0,0;%
210                          0 ,0 ,1 ,0 ,0 , -cos(ans1(4,k)-(pi/2)), 0 , 0 , 0 , -1;
211                          0 ,0 ,0 ,1 ,0 , -sin(ans1(4,k)-(pi/2)),0,0,0,1,0;
212                          0 ,0 ,0 ,0 ,0 , cos(ans1(4,k)-(pi/2)) *BD*sin(pi-gama)-sin(ans1(4,k)-(pi/2)) *BD*cos(phy-gama),0,0,BC*cos(pi-a-ans1(3,k)), -BC*sin(pi-a-ans1(3,k));
213                          0 ,0 ,0 ,0 ,0 , -cos(ans1(4,k)-(pi/2)),1,0,0,0,0;
214                          0 ,0 ,0 ,0 ,0 , sin(ans1(4,k)-(pi/2)),0,1,0,0,0;
215                          0 ,0 ,0 ,0 ,0 , ans1(2,k),0,0,0,0,0;
216                          0 ,0 ,0 ,0 ,0 ,0,0,0,1,-1,0;
217                          0 ,0 ,0 ,0 ,0 ,0,0,0,0,1];
218     AnswerMatrix = [- (f0)*cos(teta2);
219                    - (f0)*sin(teta2);
220                    0;
221                    (f3)*cos((pi/2)-gama) - (f2)*cos(gama)-(f1)*cos(teta2);
222                    (f2)*sin(gama) - (f1)*sin(teta2) + (f3) * sin((pi/2)-gama);
223                    -(f1)*sin(teta2)*BG + (f3)*BG + (I3)*ans3(3,k);
224                    (m4)*(1/2)*ans3(4,k)*cos(ans1(4,k)-(pi/2)) - (m4)*(1/2)*(ans2(4,k)^2)*cos(pi-ans1(4,k));
225                    -(m4)*(1/2)*ans3(4,k)*sin(ans1(4,k)-(pi/2)) - (m4)*(1/2)*(ans2(4,k)^2)*sin(pi-ans1(4,k));
226                    -(1/2)*(m4)*(1/2)*ans3(4,k) - I4 * ans3(4,k);
227                    0;
228                    p];
229     ANS = linsolve(CoefficientMatrix,AnswerMatrix);
230     forceAnalysis(:,k) = ANS;
231     k = k + 1;
232     teta2 = teta2 + 2*pi/360;
233 end

```

تحلیل دینامیکی در شبیه ساز:

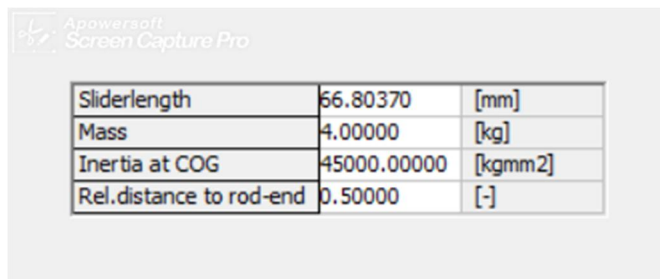
نیروی P به صورت زیر به لغزنده ۶ اضافه شده است:



شکل ۸: نحوه اثر نیروی P در شبیه ساز

این نیرو دارای مقدار ثابت 100N می باشد که به صورت ثابت با زاویه ۱۸۰ درجه نسبت به محور X به لغزنده وارد می شود.

تمام جرم ها و ممان های اینرسی که در قسمت استخراج معادلات به آنها اشاره شد به لینک ها داده شده است. مرکز جرم ها چنانچه گفته شد در وسط طول هر میله قرار گرفته است. به عنوان مثال برای مشخصات لینک چهار داریم:



Sliderlength	66.80370	[mm]
Mass	4.00000	[kg]
Inertia at COG	45000.00000	[kgmm ²]
Rel.distance to rod-end	0.50000	[-]

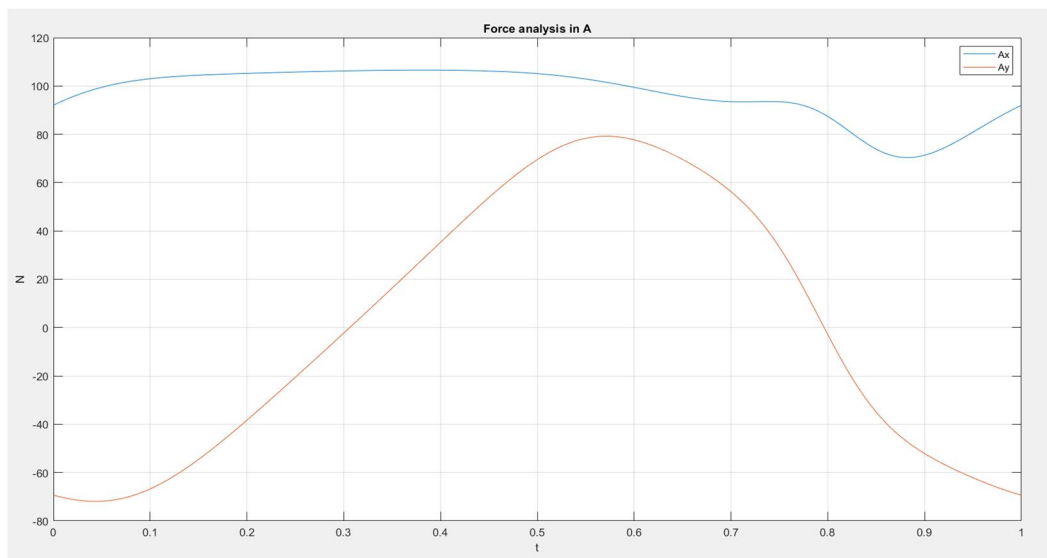
شکل ۹: مشخصات لینک ۴

که مطابق با اطلاعاتی است که پیشتر داده شد.

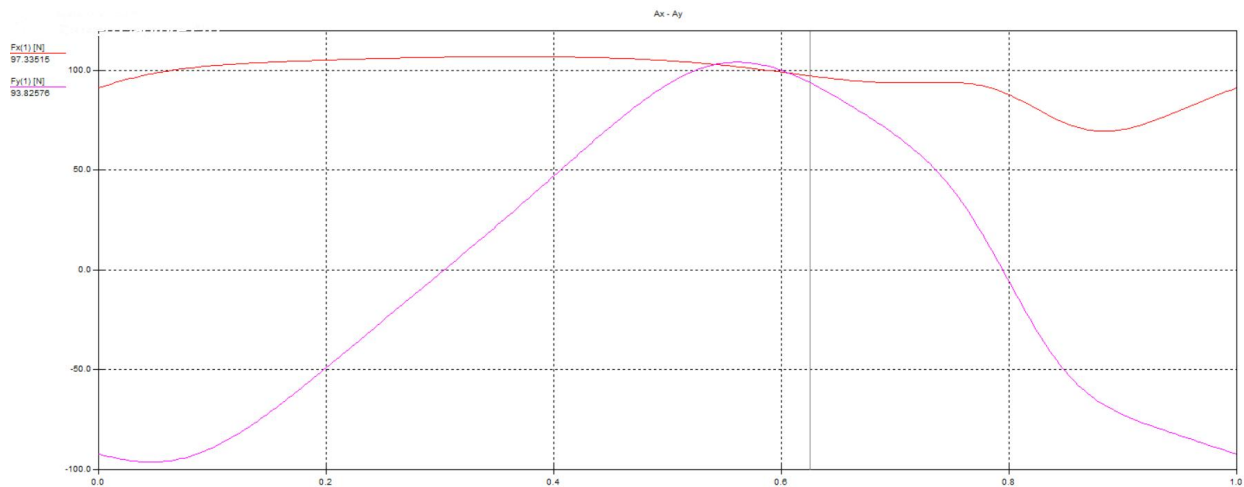
موضوع دیگری که در اینجا وجود دارد این است که در نرم افزار شبیه ساز مورد استفاده، جسم صلب مثلثی با مرکز جرمی در نقطه دلخواه قابل تعریف نیست؛ بلکه سه میله با ممان اینرسی، جرم و مراکز جرم مستقل که از اتصال آنها یک مثلث صلب تو خالی حاصل می شود قابل تعریف است. برای پشت سر گذاشتن این مشکل، جسم صلب مثلثی را (که پیشتر معادلات آن را دیدید) با سه میله ی ذکر شده معادل سازی کردیم؛ یعنی با محاسبه ی مختصات مرکز جرم هر میله و با میانگین گیری وزنی که معادلات آن را در پیوست می بینید، مرکز جرم و ممان اینرسی، و با جمع اجرام سه میله، جرم مثلث صلبی که پیشتر معادلات آن را دیدید محاسبه کردیم. در نهایت با مقادیر حاصل از این عمل معادلات را برای مثلث صلب استخراج و در متلب استفاده کردیم درحالی که در این شبیه ساز، از همان سه میله مستقل که با اتصالشان یک لینک صلب را می سازند استفاده شده است. مشاهده می کنیم که خطای ناشی از این معادل سازی بسیار کم بود و معادل سازی به درستی صورت گرفته است.

نتایج:

$A_x - A_y$



نمودار ۱۳: نیروهای تکیه گاهی در گره A در متلب



نمودار ۱۴: نیروهای تکیه گاهی در گره A در شبیه ساز

مقایسه و محاسبه خطا در لحظه $t=0.6s$

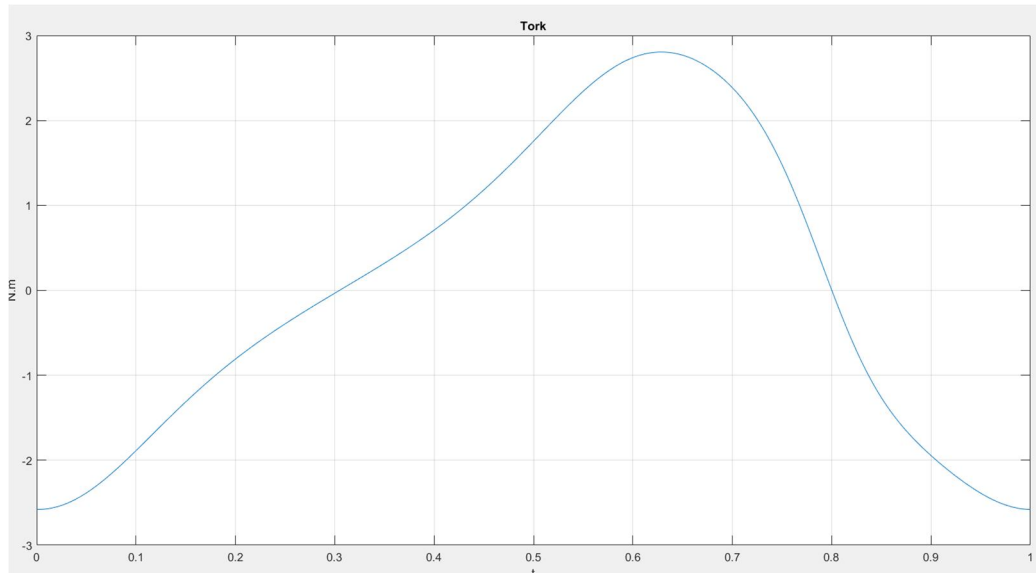
Matlab: $A_x = 99.19 \text{ N}$ $A_y = 100.07 \text{ N}$

SAM: $A_x = 99.44 \text{ N}$ $A_y = 77.73 \text{ N}$

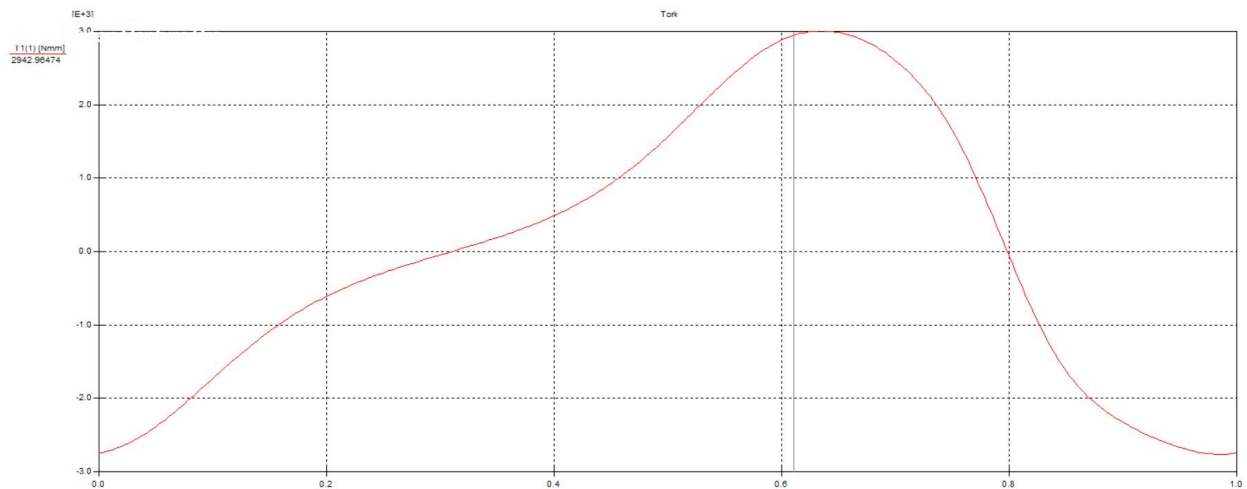
$A_x.E\% = 0.25\%$

$A_y.E\% = 22.3\%$

T



نمودار ۱۵: گشتاور وارده در گره A در متلب



نمودار ۱۶: گشتاور وارده در گره A در شبیه ساز

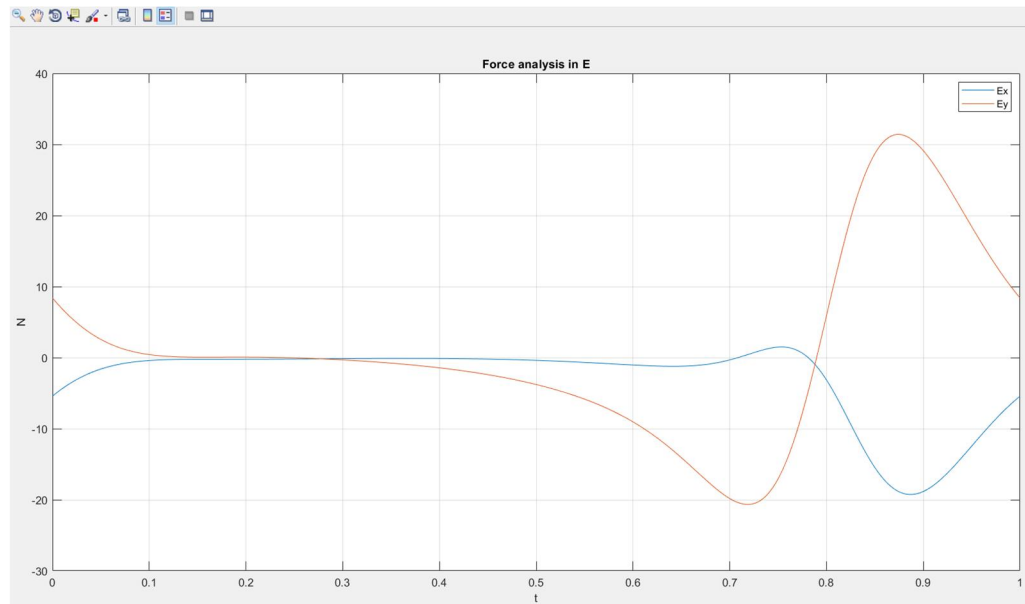
مقایسه و محاسبه خطا در لحظه $t=0.6s$

Matlab: $T=2.73 \text{ Nm}$

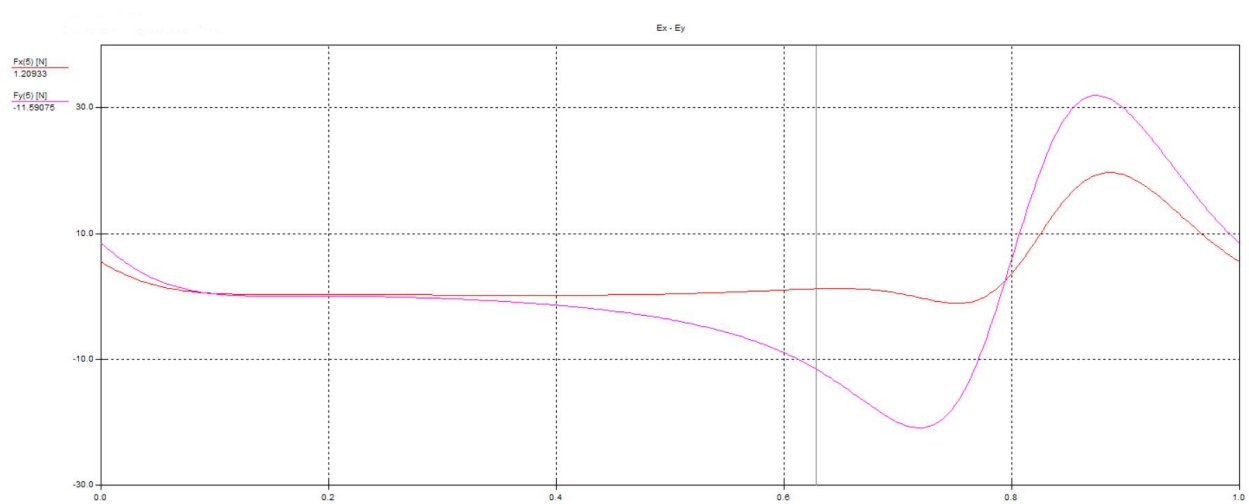
SAM: $T=2.87 \text{ Nm}$

$T.E\% = 5.1\%$

$$E_x - E_y$$



نمودار ۱۷: نیروهای تکیه گاهی در گره E در متلب



نمودار ۱۸: نیروهای تکیه گاهی در گره E در شبیه ساز

لازم به ذکر است جهت نیرو در حل دستی به سمت چپ گرفته شده در حالی که در شبیه ساز جهت مثبت به سمت راست است لذا نمودار E_x در دو نمودار قرینه یک دیگر است و ما برای محاسبه ی خطا قدر مطلق آن را مورد بررسی قرار می دهیم.

مقایسه و محاسبه خطا در لحظه $t=0.6s$

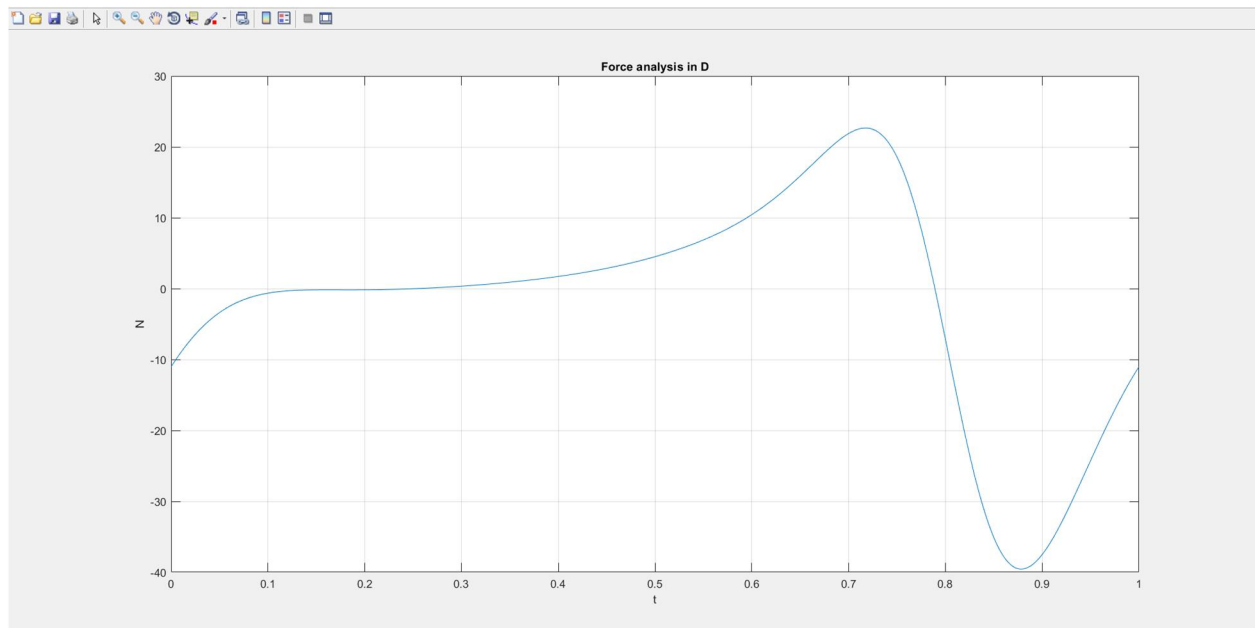
Matlab: $-E_x = 1.0318 \text{ N}$ $E_y = -9.0406 \text{ N}$

SAM: $E_x = 1.0277 \text{ N}$ $E_y = -8.9692 \text{ N}$

$E_x.E\% = 0.4 \%$ $E_y.E\% = 0.79\%$

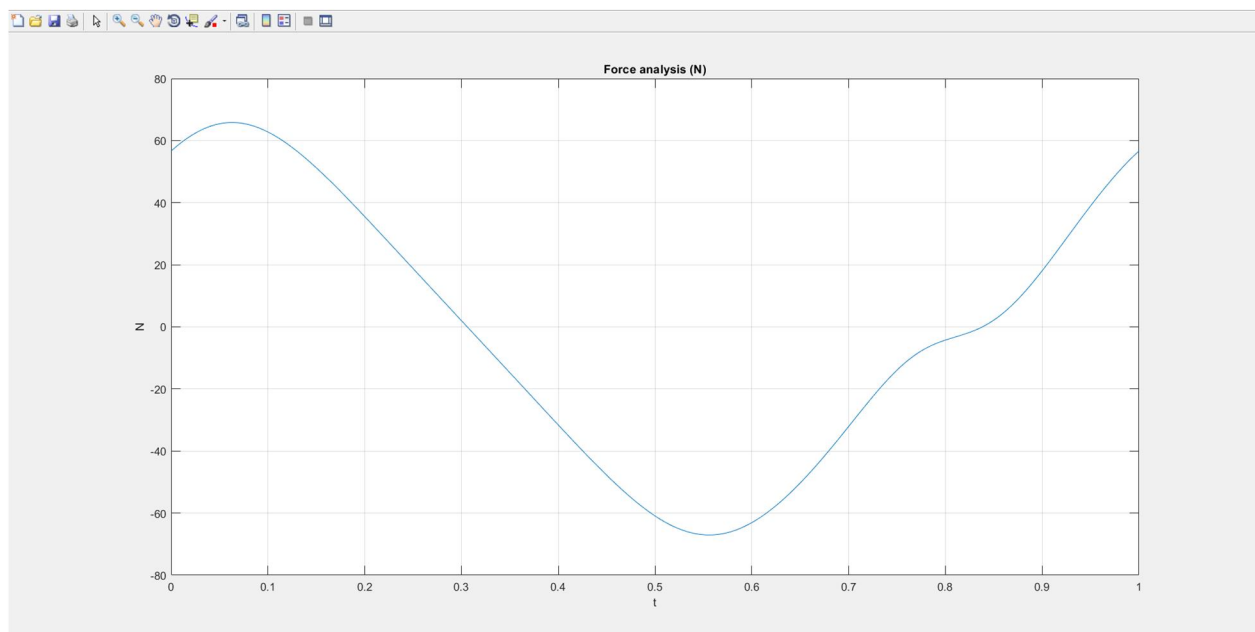
در ادامه باقی نمودار ها را مشاهده می کنید:

F_D



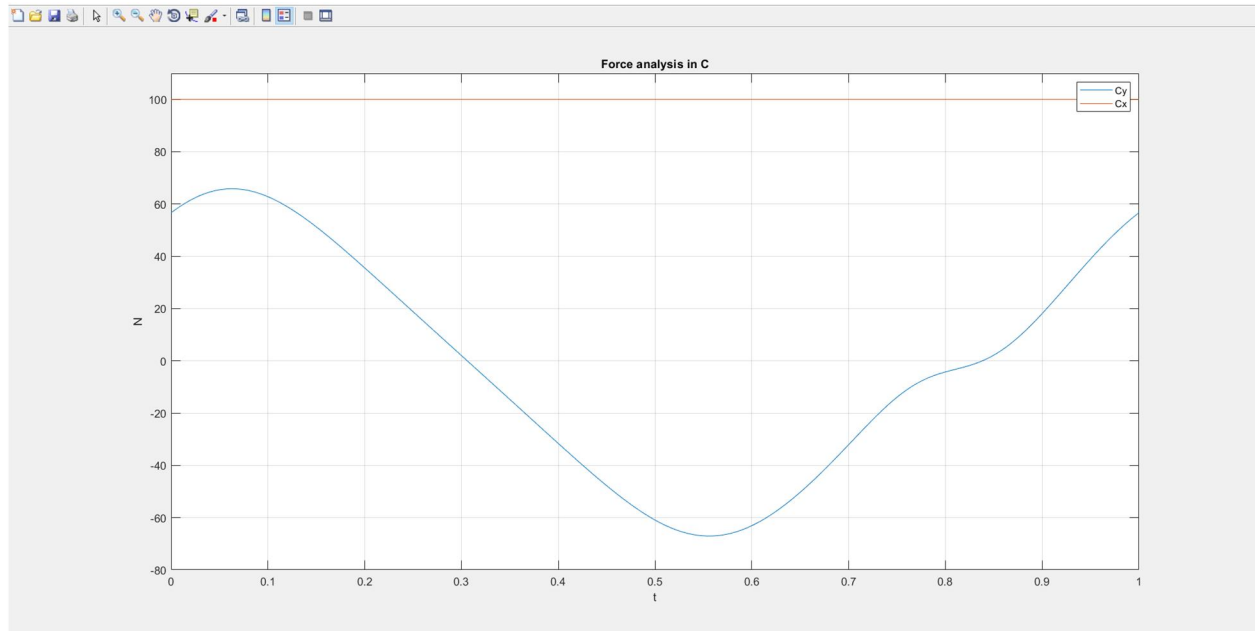
نمودار ۱۹: نیروهای تکیه گاهی در گره D در متلب

N



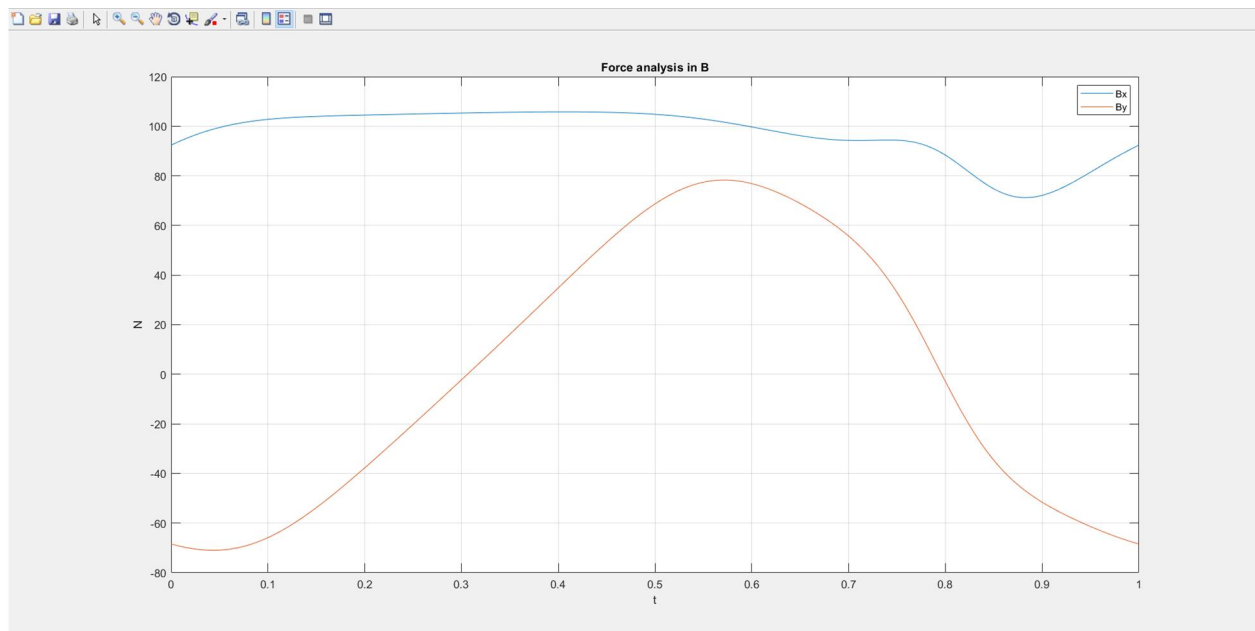
نمودار ۲۰: نیروهای تکیه گاهی در گره N در متلب

$$C_x - C_y$$



نمودار ۲۱: نیروهای تکیه گاهی در گره C در متلب

$$B_x - B_y$$



نمودار ۲۲: نیروهای تکیه گاهی در گره B در متلب

تحلیل خطا:

علاوه بر موارد ذکر شده در قسمت تحلیل سینماتیک، از آنجا که در معادلات دینامیکی به فراخور از زوایا، سرعت ها و شتاب ها استفاده می شود خطای تجمعی آن ها را نیز در اینجا داریم. فارغ از این موارد، خطای دیگر خطای مدل سازی مثلث صلب با سه میله در این قسمت است (به طور معکوس) که اگرچه تلاش شده تا با گرفتن مرکز جرم و میانگین وزنی برای ممان این خطا تا مقدار قابل توجه ای کاهش یابد، همچنان مدل سازی دارای خطا می باشد.