



دانشگاه صنعتی امیرکبیر - دانشکده برق

پروژه درس کنترل مدرن

پاییز ۹۸

موضوع پروژه:

بررسی عملکرد و کنترل ربات Ball Bot

استاد:

دکتر ایمان شریفی

نگارندگان:

سید محمد حسین طیب زاده

امین رزاقی

سید حمید رضا رودابه

محمد رضا فراهت

فهرست مطالب

بخش اول 3

- 1.1 کلیات سیستم 3
- 1.2 مدل غیر خطی و پارامترها 4
- 1.3 هدف کنترلی و فرضیات سیستم 9
- 1.4 نقطه کار و خطی سازی 9
- 1.5 فرم جردن و ماتریس تبدیل 11
- 1.6 محدوده همگرایی بین سیستم خطی و غیر خطی 12
- 1.7 کنترل پذیری و رویت پذیری 13
- 1.8 پایدار پذیری و رویت پذیری 13
- 1.9 کالمن 14

بخش دوم 15

- 2.1 فیدبک حالت 15
- 2.2 فیدبک حالت استاتیکی 16
- 2.3 فیدبک حالت با اتگرال گیر و دینامیکی 16
- 2.4 رویتگر کامل و کاهش یافته 18
- 2.5 عملکرد فیدلک حالت و رویتگر با هم 21

بخش سوم 23

- 3.1 نقاط تعادل 23
- 3.2 انواع پایداری و استفاده از لیاپانوف 23
- 3.3 بهینه سازی 23
- شیبه سازی 25

بخش اول

1-1 کلیات سیستم

در ساخت ربات ballbot، هدف آن است که بتوانیم در عین حفظ تعادل ربات بر روی یک توپ، آن را در جهات مختلف به حرکت در بیاوریم. بر خلاف ساختار معمول که ربات ها برای صاف ایستادن از مرکز جرم در ارتفاع پایین و سطح مقطع بزرگ بهره می برند، این نوع ربات برای حفظ تعادلش نیاز دارد تا به صورت لحظه ای برای حفظ تعادل خود تلاش کند؛ به عبارت دیگر، نیاز است تا تعادل ربات به صورت دینامیکی حفظ شود.

به طور کلی، دینامیک یک ربات Ball Bot پیچیده و بسیار غیر خطی است، و نیاز است تا در قدم نخست ساده سازی هایی در مورد آن انجام شود. ابتدا، فرض می کنیم که هیچ گونه انعطافی وجود نداشته و ربات از یک جسم صلب که بر روی یک کره ی صلب دوار قرار دارد، تشکیل شده است. همچنین فرض می کنیم که حرکت در دو صفحه قائم، یعنی صفحات $(X-Z)$ و $(Y-Z)$ به ازای انحراف کوچک محور ربات از خط عمود، به صورت جدا و مستقل می باشد، و با توجه با سازه نسبتاً متقارن ربات، می توان معادلات دینامیک در این دو صفحه را مشابه یکدیگر در نظر گرفت.

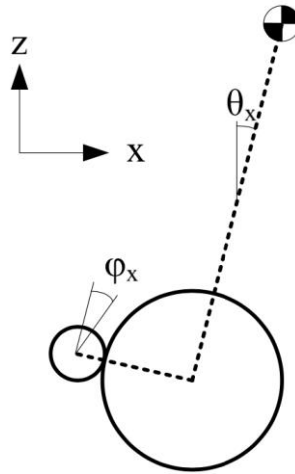
به کمک دو فرض فوق، می توان معادلات حرکت و کنترل برای ربات در فضای سه بعدی را، به دو زیر مساله مستقل تقسیم نموده و هر یک را جداگانه حل نمود. همچنین از میان انواع مدل سازی های موجود برای اصطکاک، تنها اصطکاک ویسکوز را در نظر می گیریم، چرا که اصطکاک کولمبی و اصطکاک استاتیکی می تواند معادلات سیستم را تا حد زیادی پیچیده تر کند.

سخت افزار مورد نیاز برای پیاده سازی Ball Bot عبارت اند از: سروو موتور، یک عدد توپ، مکانیزم انتقال قدرت از موتور به توپ، سنسور شتاب سنج و ژایروسکوپ برای اندازه گیری زاویه و سرعت زاویه ای، سنسور انکودر برای اندازه گیری سرعت شفت موتور، بدنه ربات و میکروکنترلر. مکانیزم انتقال قدرت از دو چرخ omni directional تشکیل شده است که بصورت عمود بر یکدیگر قرار گرفته اند. به کمک دو خاصیت تعامد و چند جهته بودن چرخ ها، می توان کنترل در دو صفحه را بصورت مستقل از هم انجام داد. همچنین برای اندازه گیری زاویه tilt و سرعت زاویه ای محور ربات، می توان از یک سنسور inertial measurement unit یا به اختصار IMU، مانند 6050MPU استفاده کرد.

عملکرد سیستم به این صورت است که چرخش هر یک از دو سروو موتور، از طریق چرخ متصل به شفت موتور به توپ اعمال شده و سبب حرکت ربات در راستای عمود بر محور چرخ می گردد.

1-2 مدل غیر خطی و پارامترها

در این قسمت، معادلات سیستم غیر خطی را به کمک معادلات لاگرانژ بدست می آوریم. معادلات تنها برای صفحه X-Z بدست آمده و سپس با استفاده از مفروضات بیان شده در بخش قبل، به صفحه Y-Z تعمیم می یابند.



شکل - شمای فیزیکی سیستم در صفحه X-Z

نخست، پارامترهای مربوطه را تعریف می کنیم:

R_b - شعاع توپ

L - فاصله مرکز جرم توپ و مرکز جرم بدنه ربات

M_b - جرم توپ

M_B - جرم بدنه

I_b - ممان اینرسی توپ

I_{Bx} - ممان اینرسی بدنه حول محور X

I_{By} - ممان اینرسی بدنه حول محور Y

I_M - ممان اینرسی موتور

n - نسبت گیربکس

$_{Bb}^{??}$ - ضریب اصطکاک بین بدنه و توپ

$_{Bg}^{??}$ - ضریب اصطکاک بین توپ و زمین

K_b - ثابت ولتاژ Back EMF موتور

K_t - ثابت گشتاور موتور

R_m - مقاومت آرمیچر موتور

$tetha$ - زاویه محور ربات با راستای عمود

phi - زاویه شفت موتور

برای استفاده از روش لاگرانژ، نیاز است ابتدا انرژی جنبشی و پتانسیل گرانشی کل مجموعه ربات مشخص شود. برای این منظور، ابتدا روابط سینماتیک برای توپ و بدنه مشخص گردند. روابط سینماتیک توپ به شکل زیر است:

Position		Velocity	
Angle	$\theta_x + n\phi_x$	Angular velocity	$\dot{\theta}_x + n\dot{\phi}_x$
x position, x_b	$R_b(\theta_x + n\phi_x)$	x velocity, \dot{x}_b	$R_b(\dot{\theta}_x + n\dot{\phi}_x)$
z position, z_b	0	z velocity, \dot{z}_b	0

انرژی جنبشی خطی توپ، T_{linb} به شکل زیر،

$$\frac{M_b R_b^2 (\dot{\theta}_x + n\dot{\phi}_x)^2}{2}$$

و انرژی جنبشی دورانی توپ، T_{rotb} نیز به شکل زیر می باشد:

$$\frac{I_b (\dot{\theta}_x + n\dot{\phi}_x)^2}{2}$$

مبدأ پتانسیل را مرکز جرم توپ در نظر می گیریم، در نتیجه انرژی پتانسیل گرانشی آن، V_b برابر صفر فرض می شود.

سپس به سینماتیک بدنه ربات می رسم:

Position		Velocity	
Angle	θ_x	Angular Velocity	$\dot{\theta}_x$
x position, x_B	$x_b + L \sin(\theta_x)$	x velocity, \dot{x}_B	$\dot{x}_b + L \cos(\theta_x) \dot{\theta}_x$
z position, z_B	$L \cos(\theta_x)$	z velocity, \dot{z}_B	$-L \sin(\theta_x) \dot{\theta}_x$

روندی مشابه توپ را، این بار برای بدنه ربات پی می گیریم. ابتدا انرژی جنبشی خطی بدنه، T_{linB} به شکل زیر،

$$\frac{M_B \left(L^2 \dot{\theta}_x^2 + 2 \cos(\theta_x) L n R_b \dot{\phi}_x \dot{\theta}_x + 2 \cos(\theta_x) L R_b \dot{\theta}_x^2 + n^2 R_b^2 \dot{\phi}_x^2 + 2 n R_b^2 \dot{\phi}_x \dot{\theta}_x + R_b^2 \dot{\theta}_x^2 \right)}{2}$$

و انرژی جنبشی دورانی آن، T_{rotB} به شکل زیر بدست می آیند.

$$\frac{I_{Bx} \dot{\theta}_x^2}{2}$$

این بار اما بر خلاف توپ، بدنه ربات می تواند دارای انرژی پتانسیل V_B باشد، و مقدار آن از رابطه زیر بدست می آید:

$$g L M_B \cos(\theta_x)$$

در نهایت، معادلات سینماتیک و انرژی جنبشی دورانی T_{rotm} برای موتور ها به شکل زیر می باشد:

Position		Velocity	
Angle	$\theta_x + \phi_x$	Angular velocity	$\dot{\theta}_x + \dot{\phi}_x$

$$\frac{I_M (\dot{\phi}_x + \dot{\theta}_x)^2}{2}$$

حالا می توانیم لاگرانژین را به شکل زیر بنویسیم:

$$L = T_{linb} + T_{linB} + T_{rotb} + T_{rotB} + T_{rotm} - V_b - V_B$$

معادله اوایلر-لاگرانژ را اعمال می نماییم تا ماتریس نیرو ها بدست بیاید:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = F_i$$

در عبارت فوق، بردار q را به شکل زیر در نظر بگیرید:

$$\mathbf{q} = [\theta_x \ \phi_x]^T$$

ماتریس نیرو به شکل رابطه زیر خواهد بود:

$$\begin{pmatrix} -\mu_{Bg} \dot{\theta}_x \\ \frac{K_t v_x}{R_m} - \mu_{Bb} \dot{\phi}_x - \frac{K_b K_t \phi_x}{R_m} \end{pmatrix}$$

و معادلات دینامیکی غیر خطی ربات را به فرم زیر می نویسیم:

$$M_x(\mathbf{q}, \dot{\mathbf{q}}) \ddot{\mathbf{q}} + R_x(\mathbf{q}, \dot{\mathbf{q}}) = F_x(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{v}_x)$$

ماتریس M_x ، اصطلاحاً ماتریس جرم ربات نامیده می شود، و مقدار آن را از طریق روابطی که در ادامه مشاهده خواهید نمود بدست می آوریم. همانطور که خواهیم دید، مقدار این ماتریس تنها تابعی از q_x می باشد.

ماتریس M_x ابتدا به شکل فشرده زیر نوشته، و در ادامه مقدار باز شده هر یک از چهار بلوک را ذکر می کنیم:

$$\begin{pmatrix} M_x(1,1) & M_x(1,2) \\ M_x(2,1) & M_x(2,2) \end{pmatrix}$$

مقادیر بلوک ها به شکل زیر می باشد:

$$M_x(1,1) = I_{Bx} + I_M + I_b + L^2 M_B + M_B R_b^2 + M_b R_b^2 + 2 L M_B R_b \cos(\theta_x)$$

$$M_x(1,2) = I_M + I_b n + M_B n R_b^2 + M_b n R_b^2 + L M_B n R_b \cos(\theta_x)$$

$$M_x(2,1) = I_M + I_b n + M_B n R_b^2 + M_b n R_b^2 + L M_B n R_b \cos(\theta_x)$$

$$M_x(2,2) = I_M + I_b n^2 + M_B n^2 R_b^2 + M_b n^2 R_b^2$$

برای بدست آوردن معادلات فضای حالت ربات، ابتدا معادلات دینامیکی را به شکل زیر بازنویسی می کنیم:

$$\ddot{\mathbf{q}} = M_x^{-1}(\mathbf{F}_x - \mathbf{R}_x)$$

حال بردار حالت را به شکل

$$\mathbf{x} = [\theta_x \ \phi_x \ \dot{\theta}_x \ \dot{\phi}_x]^T$$

تعریف نموده، معادلات غیر خطی فضای حالت به صورت زیر خواهد بود:

$$\dot{\mathbf{x}} = f(\mathbf{x})$$

1-3 هدف کنترلی و فرضیات سیستم

هدف کنترلی خود را، حفظ تعادل ربات حول نقطه تعادلش در ، یعنی زاویه صفر با محور عمود و صفر شدن تمامی سرعت ها در این نقطه قرار میدهیم.

1-4 نقطه کار و خطی سازی

همانطور که پیشتر اشاره شد، نقطه تعادل ربات، جایی است که تمامی سرعت ها صفر شده و همچنین زاویه محور ربات با راستای عمود نیز برابر صفر شود. صورت کلی چنین نقطه ای به شکل زیر است:

$$\bar{\mathbf{x}} = [0 \ \phi_x \ 0 \ 0]^T$$

یعنی در نقطه تعادل، جز زاویه شفت موتور، سایر متغیرهای حالت مقدار صفر اختیار می کنند. برای خطی سازی حول نقطه تعادل، در همسایگی بسیار کوچک این نقطه رابطه غیر خطی را بصورت زیر تقریب رده،

$$\hat{\dot{\mathbf{x}}} = f(\bar{\mathbf{x}}) + J(\bar{\mathbf{x}})\hat{\mathbf{x}}$$

و مشاهده می کنیم که مقدار مناسب برای ماتریس J ، ژاکوبین رابطه f می باشد. پس از خطی سازی، به رابطه ی آشنای

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$$

می رسیم. صورت فشرده شده ی ماتریس های A و B به شکل زیر است:

$$A_x = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ A_x(3,1) & 0 & A_x(3,3) & A_x(3,4) \\ A_x(4,1) & 0 & A_x(4,3) & A_x(4,4) \end{pmatrix}$$

$$B_x = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ B_x(3,1) & 0 \\ B_x(4,1) & 0 \end{pmatrix}$$

و روابط باز شده ی هریک از بلوک ها را در صفحه بعد مشاهده خواهید نمود:

$$A_x(3,1) =$$

$$\frac{g L M_B (I_M + I_b n^2 + M_B n^2 R_b^2 + M_b n^2 R_b^2)}{D_x}$$

$$A_x(4,1) =$$

$$-\frac{g L M_B (I_M + I_b n + M_B n R_b^2 + M_b n R_b^2 + L M_B n R_b)}{D_x}$$

$$A_x(3,3) =$$

$$- \frac{\mu_{Bg} (I_M + I_b n^2 + M_B n^2 R_b^2 + M_b n^2 R_b^2)}{D_x}$$

$$A_x(4,3) =$$

$$\frac{\mu_{Bg} (I_M + I_b n + M_B n R_b^2 + M_b n R_b^2 + L M_B n R_b)}{D_x}$$

$$A_x(3,4) =$$

$$\frac{\left(\mu_{Bb} + \frac{K_b K_t}{R_m}\right) (I_M + I_b n + M_B n R_b^2 + M_b n R_b^2 + L M_B n R_b)}{D_x}$$

$$A_x(4,4) =$$

$$- \frac{\left(\mu_{Bb} + \frac{K_b K_t}{R_m}\right) (I_{Bx} + I_M + I_b + L^2 M_B + M_B R_b^2 + M_b R_b^2 + 2 L M_B R_b)}{D_x}$$

$$B_x(3,1) =$$

$$- \frac{K_t (I_M + I_b n + M_B n R_b^2 + M_b n R_b^2 + L M_B n R_b)}{D_x R_m}$$

$$B_x(4,1) =$$

$$\frac{K_t (I_{Bx} + I_M + I_b + L^2 M_B + M_B R_b^2 + M_b R_b^2 + 2 L M_B R_b)}{D_x R_m}$$

$$D_x =$$

$$\begin{aligned} & I_{Bx} I_M + I_M I_b - 2 I_M I_b n + I_{Bx} I_b n^2 + I_M I_b n^2 + I_M L^2 M_B + I_M M_B R_b^2 + I_M M_b R_b^2 \\ & + I_b L^2 M_B n^2 + I_{Bx} M_B n^2 R_b^2 + I_M M_B n^2 R_b^2 + I_{Bx} M_b n^2 R_b^2 + I_M M_b n^2 R_b^2 \\ & + 2 I_M L M_B R_b - 2 I_M M_B n R_b^2 - 2 I_M M_b n R_b^2 + L^2 M_B M_b n^2 R_b^2 - 2 I_M L M_B n R_b \end{aligned}$$

5-1 فرم جردن

با جایگذاری پارامترهای سیستم در روابط فوق، همچنین گرفتن x به عنوان خروجی سیستم و v_x که همان ولتاژ موتور است به عنوان ورودی، به نتایج زیر می‌رسیم:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 2272.2 & 0 & 0 & 1029.6 \\ -1.3392 & 0 & 0 & -1220.7 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ -64.3859 \\ 472.8322 \end{bmatrix}$$

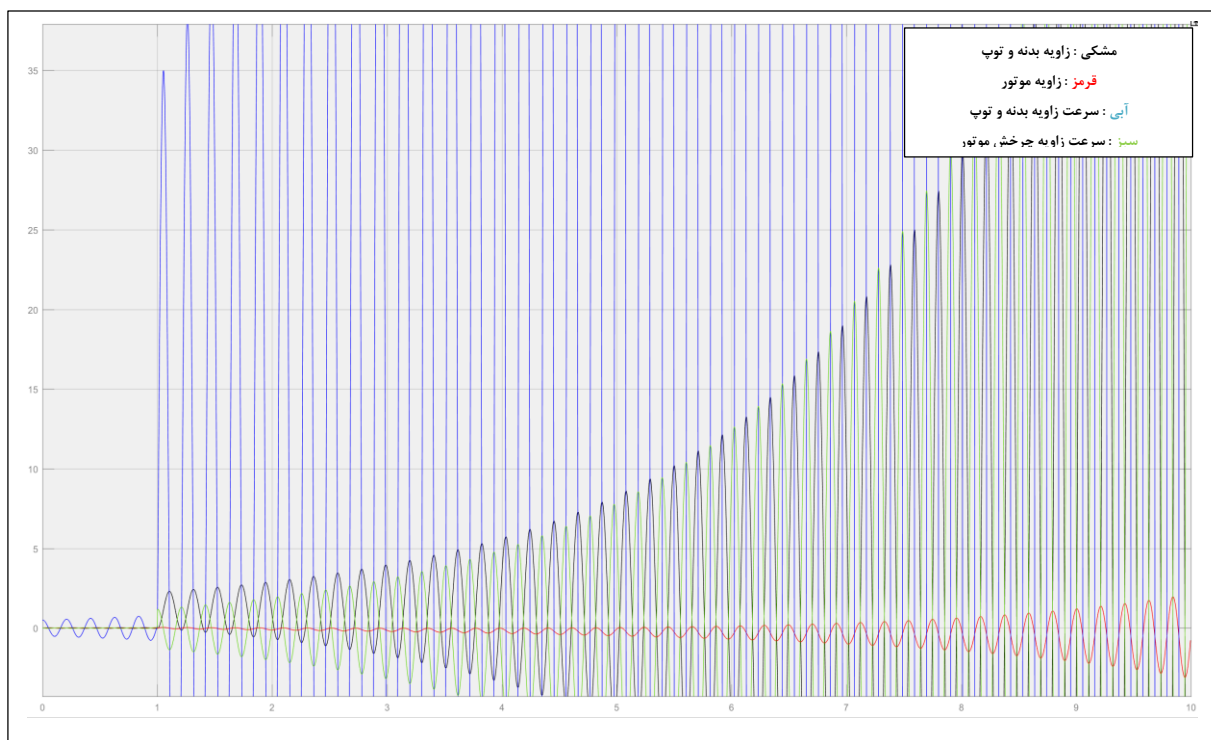
$$C = [0 \quad 1 \quad 0 \quad 0]$$

$$D = 0$$

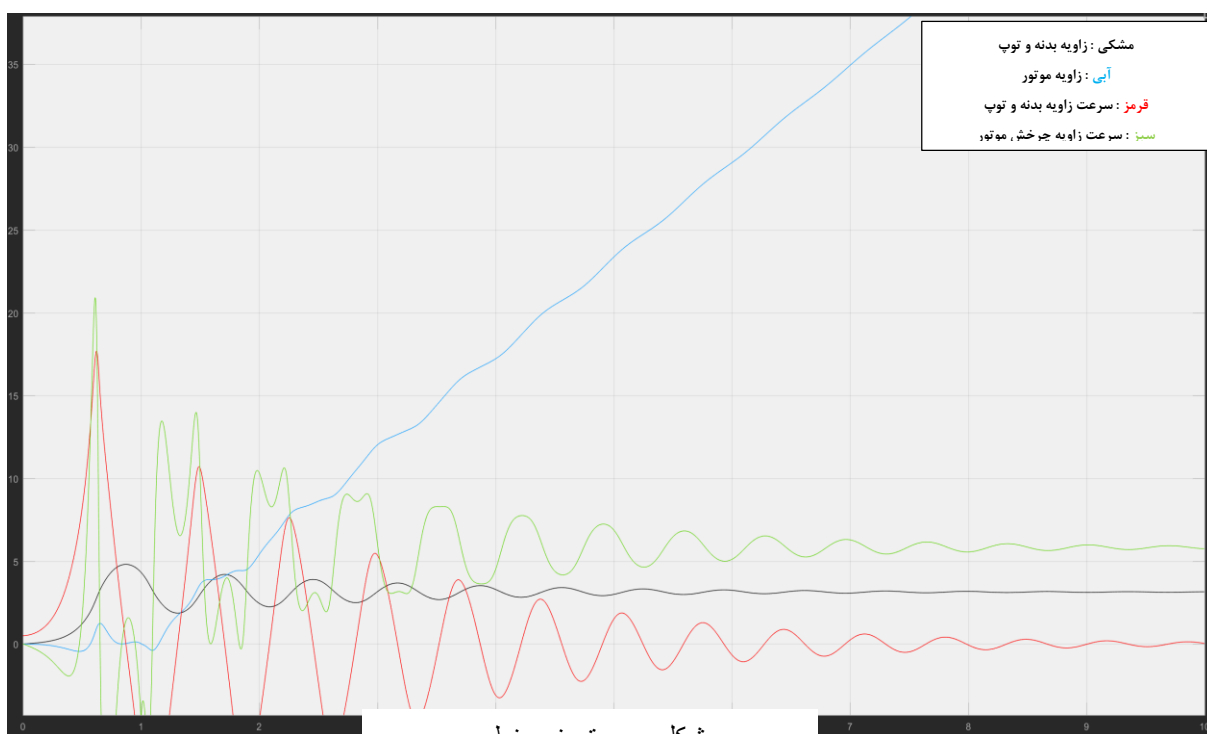
$$T = \begin{pmatrix} 0 & 0.0007 & -0.9119 + j0.0224 & -0.9119 - j0.0224 \\ 1 & -0.0008 & 0.005 + j0.0333 & 0.0005 - j0.0333 \\ 0 & -0.8429 & 0.2518 + j27.3875 & 0.2518 - j27.3875 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

$$J = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -1221.7 & 0 & 0 \\ 0 & 0 & 0.5 - j30 & 0 \\ 0 & 0 & 0 & 0.5 + j30 \end{bmatrix}$$

7- به دلیل اینکه سیستم ما اساساً پایدار نیست و ناپایدار است، مطلبی که از دینامیک سیستم هم نیز به راحتی مشخص است تعادل روی یک جسم کروی مثل توپ به خودی خوب ناپایدار است بنابراین هم در سیستم خطی و هم در سیستم غیر خطی پاسخی ناپایدار خواهیم داشت و ناحیه ای همگرایی برای این 2 سیستم اساساً قابل تعریف نیست، هرچند نوع ناپایداری در این 2 سیستم فرق دارد که در 2 شکل زیر نشان داده شده است.



شکل - سیستم خطی



شکل - سیستم غیر خطی

همان طور که مشاهده می شود هیچ ناحیه همگرایی بین این 2 وجود ندارد و برای تحلیل این 2 سیستم می شود گفت که در سیستم غیر خطی ، به دلیل وجود قیود مثلثاتی ، بیناهیت تعریف نمیشود و محدودیت داریم و اگر مشاهده کنید در این سیستم در حالت STEADY-STATE زاویه بین بدنه و توپ به سمت Pi میل می کند که کاملاً منطقی است هر چند در دنیای واقعی شدن نیست . راجه به سیستم خطی نیز ، به دلیل خطی شدن و استفاده از ماتریس ژاکوبین ، قیود مثلثاتی را نداریم و پاسخ هایمان به بی نهایت میل می کند .

8- راجع به کنترل پذیری و رویت پذیری سیستم هم با توجه به ماتریس های اصلی فضای حالت که در شکل پایین مشاهده می کنید .

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 2272.2 & 0 & 0 & 1029.6 \\ -1.3392 & 0 & 0 & -1220.7 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ -64.3859 \\ 472.8322 \end{bmatrix}$$

$$C = [0 \quad 1 \quad 0 \quad 0]$$

$$D = 0$$

شکل - ماتریس های اصلی فضای حالت سیستم خطی

فقط کافیست رنگ کنترل پذیری و رویت پذیری را چک کنیم :

```
>> rank(ctrb(A,B))
ans =
4
```

```
>> rank(observ(A,C))
ans =
4
```

که همانطور که مشاهده می کنید رنگ هر دو کامل است بنابراین سیستم هم کنترل پذیر و هم رویت پذیر است .

9- تحقق به دلیل آنچه که در قسمت قبلی گفته شد ، هم رویت پذیر و هم کنترل پذیر می باشد بنابراین تحقق ، تحقق مینیمال است .

برای مشخص کردن آشکارپذیری و پایدارپذیری سیستم نیز باز هم به دلیل رویت پذیر بودن و کنترل پذیر بودن ، در صورت داشتن قطب ناپایدار ، هم آشکار پذیر و هم رویت پذیر می باشد .

(توجه شود که سیستم ما ، 1 قطب ناپایدار دارد .)

```
>> eig(A)

ans =

    1.0e+03 *

    0.0000 + 0.0000i
   -1.2217 + 0.0000i
    0.0005 + 0.0300i
    0.0005 - 0.0300i
```

شکل - قطب های سیستم

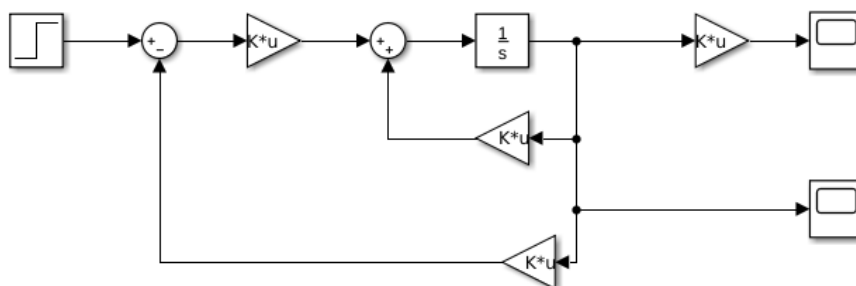
10 - به دلیل مینیمال بودن تحقق ، تجزیه کالمن برابر خود تحقق خواهد بود ، بنابراین نیازی به تجزیه کالمن نیست .

1. برای طراحی فیدبک حالت ، باید این نکته را در نظر داشته باشیم که پاسخ مطلوب ما پاسخی است که متغیر اول ما حول نقطه 0 نوسان کند در واقع متغیر اول همان زاویه محور بدنه با توپ می باشد که در حالت مطلوب باید به صفر میل کند و یا حول آن با دامنه بسیار کم نوسان کند تا نهایتاً حول نقطه تعادل بایستد .
مهم ترین حالت تحت کنترل زاویه انحراف سیستم است، به همین جهت قطب های مطلوب را بگونه ای انتخاب می کنیم که این حالت سریع همگرا شود.
با سعی و خطا قطب های مطلوب در -5 و -6 و -600 و -700 انتخاب شد.

```
k = place(A,B,[-5 -6 -600 -700]);
```

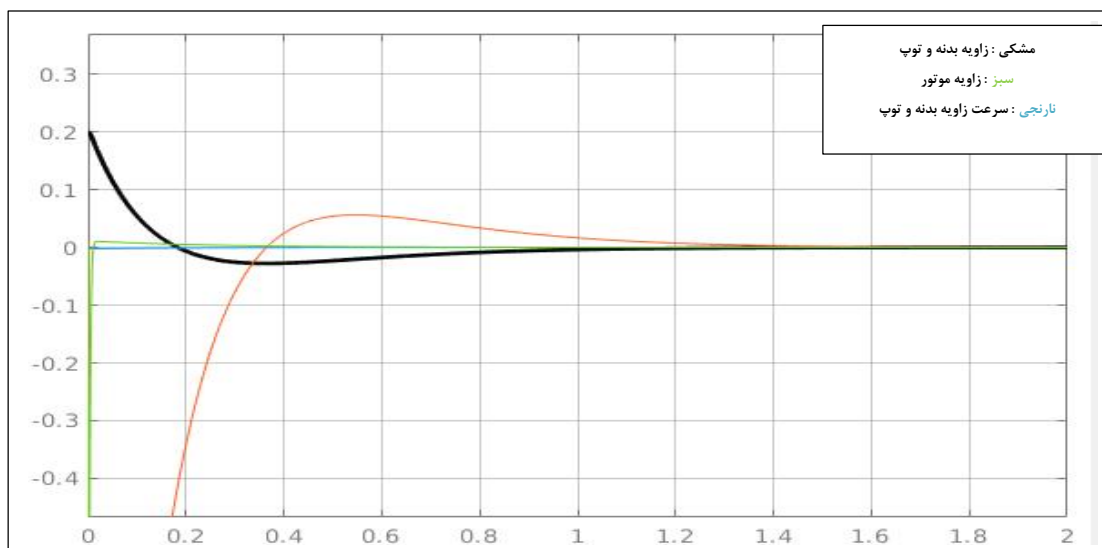
```
k =
```

```
8.7370 -593.8758 1.7538 0.4297
```



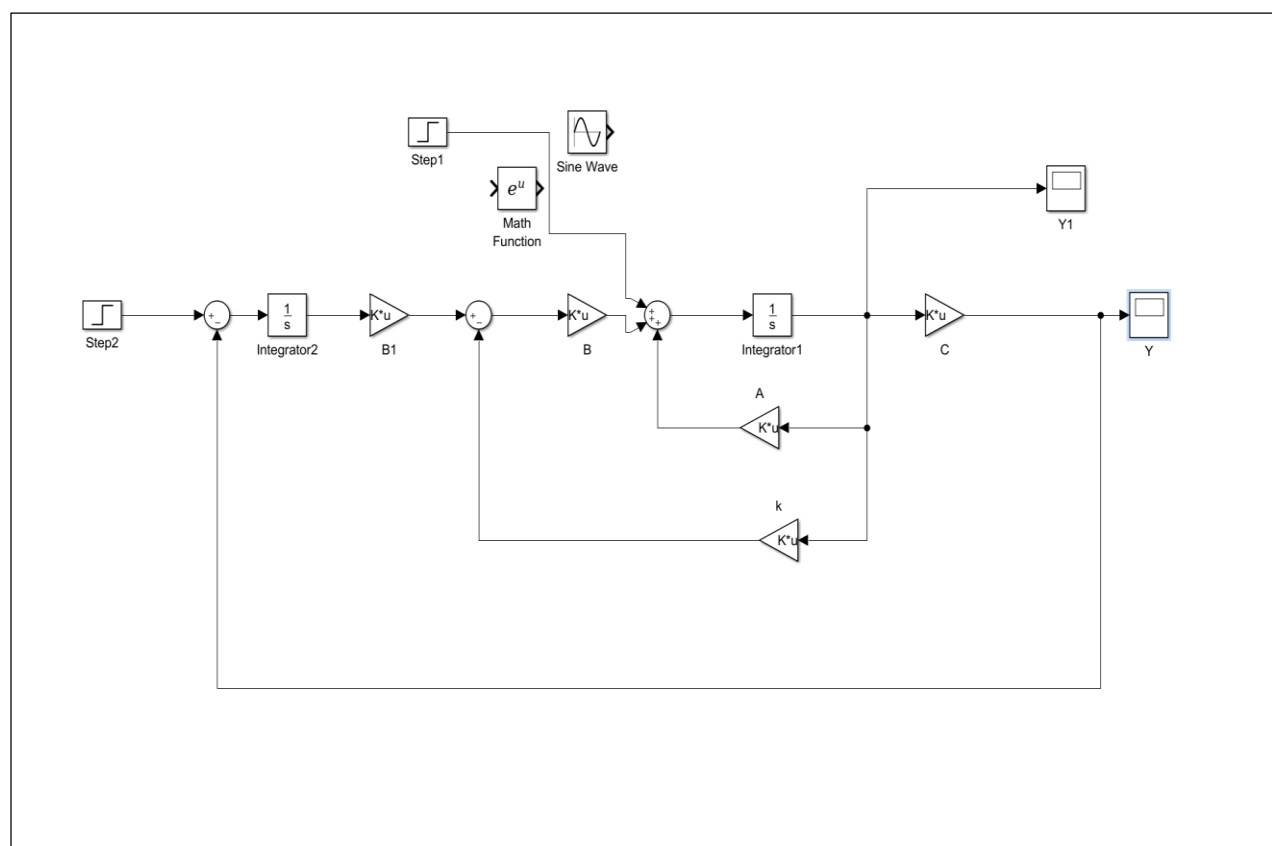
شکل - دیاگرام کلی سیستم با فیدبک حالت

در تصویر زیر همگرایی حالات به نقطه تعادلشان (نقطه صفر) قابل مشاهده است که منحنی مشکی رنگ زاویه انحراف ربات است که از نقطه 11 درجه انحراف شروع شده و به صفر همگرا شده است.



2. در این بخش سوال ، به دلیل اینکه سیستم ما اساسا نباید ورودی را دنبال کند ، چرا که ورودی سیستم ولتاژ موتور و خروجی مورد نظر زاویه محور با توپ می باشد بنابراین به ازای هر ورودی ، خروجی ما باید به صفر میل کند و ورودی مرجع نیز ولتاژ صفر ولت است (نقطه تعادل ربات)، بنابراین اصلا نیازی به جبران ساز استاتیکی نخواهیم داشت.

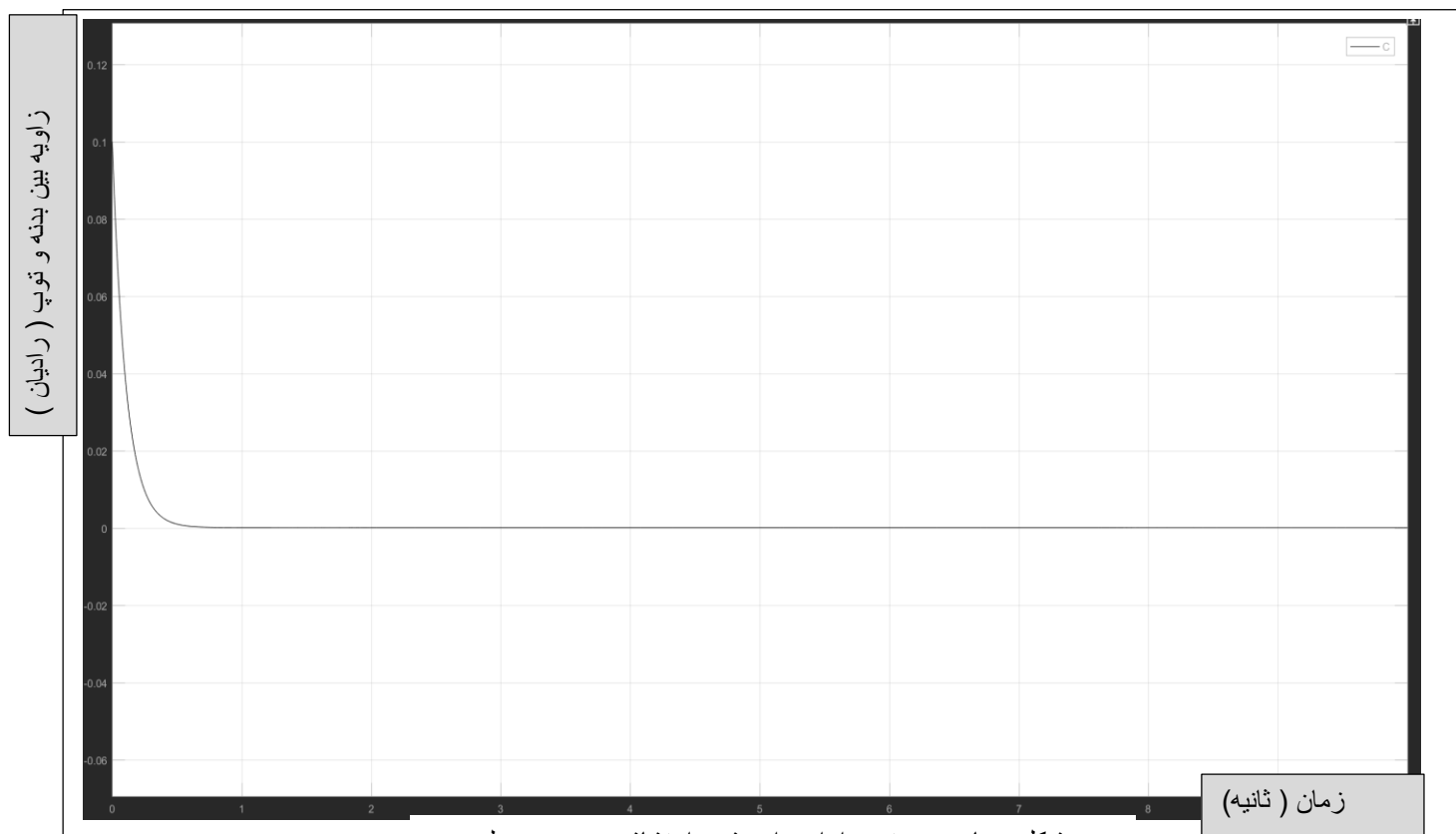
3 – در این بخش برای طراحی فیدبک حالت با انتگرال گیر و به اصطلاح دینامیکی ، مطابق دایاگرامی که در درس داشتیم ابتدا بلوک ها را مطابق شکل زیر طراحی می کنیم .



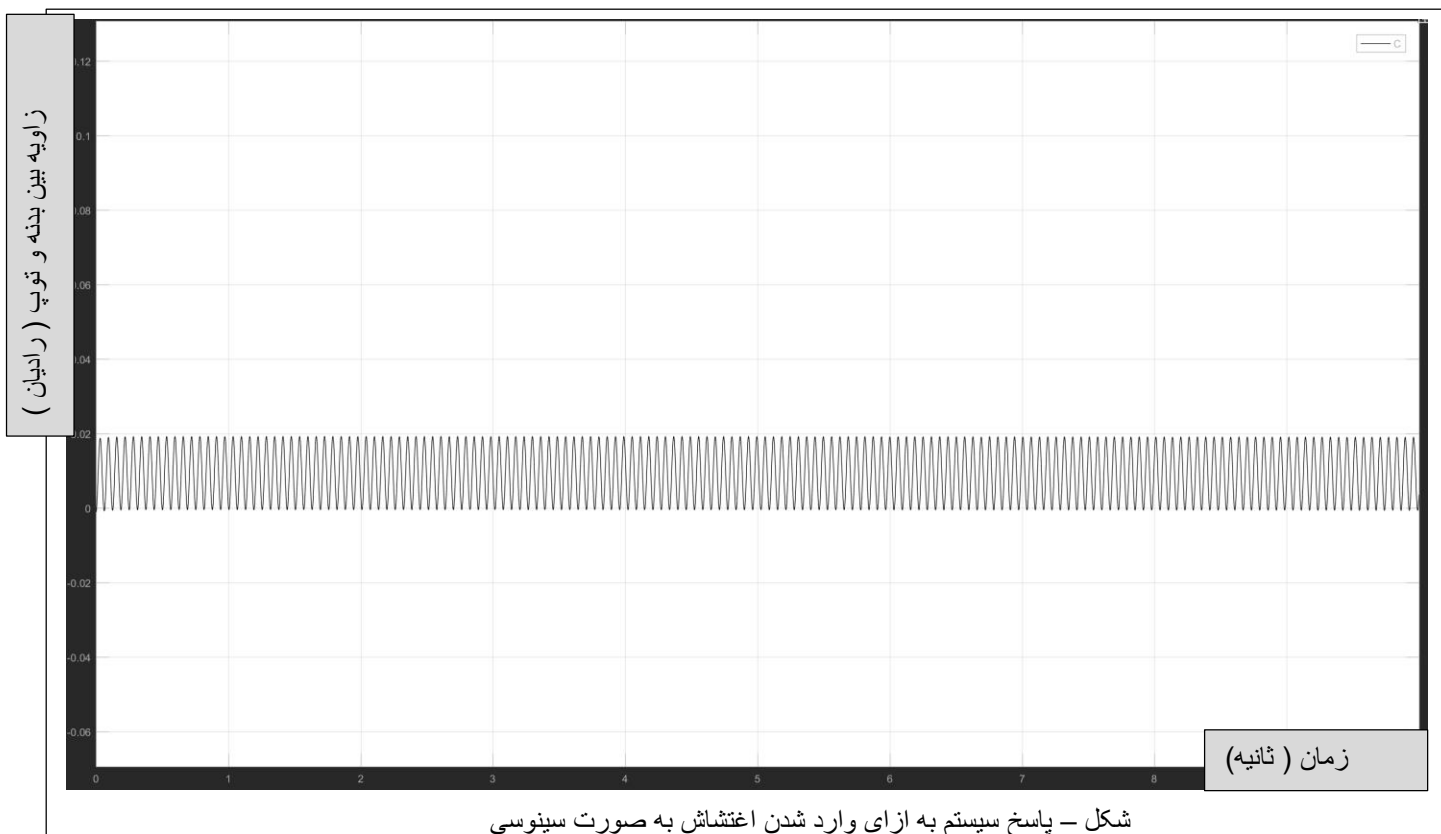
شکل – بلوک دایاگرام فیدبک حالت با انتگرال گیر

به طور کلی روش خاصی برای طراحی گین های فیدبک حالت استفاده نکردیم و بیشتر از سعی و خطا استفاده کردیم تا به پاسخ مطلوب نسبی خود برسیم .
همین طور در این بخش ، طوری گین ها را طراحی کردیم که به انواع اغتشاشات مقاوم باشد ، که در شکل های پایین عملکرد سیستم را مشاهده می کنید که با دقت خوبی و اختلاف اندکی با پاسخ مطلوب توانسته اغتشاشات را دمپ کند .

در شکل های صفحه بعد می توانید مشاهده کنید ، که به خوبی اغتشاش دمپ شده است . در رابطه با اغتشاش سینوسی ، پاسخ نزدیکی 0 و با دامنه بسیار کوچک نوسان می کند که پاسخ مطلوبی است .



شکل – پاسخ سیستم به ازای وارد شدن اغتشاش به صورت پله



شکل – پاسخ سیستم به ازای وارد شدن اغتشاش به صورت سینوسی

4 - طراحی رویتگر مرتبه کامل با دستور place برای قطب های مطلوب -5 و -10 و -80 و -100 :

```
Lo = place(A' , C' , [-5 -10 -80 -100])'
```

Lo =

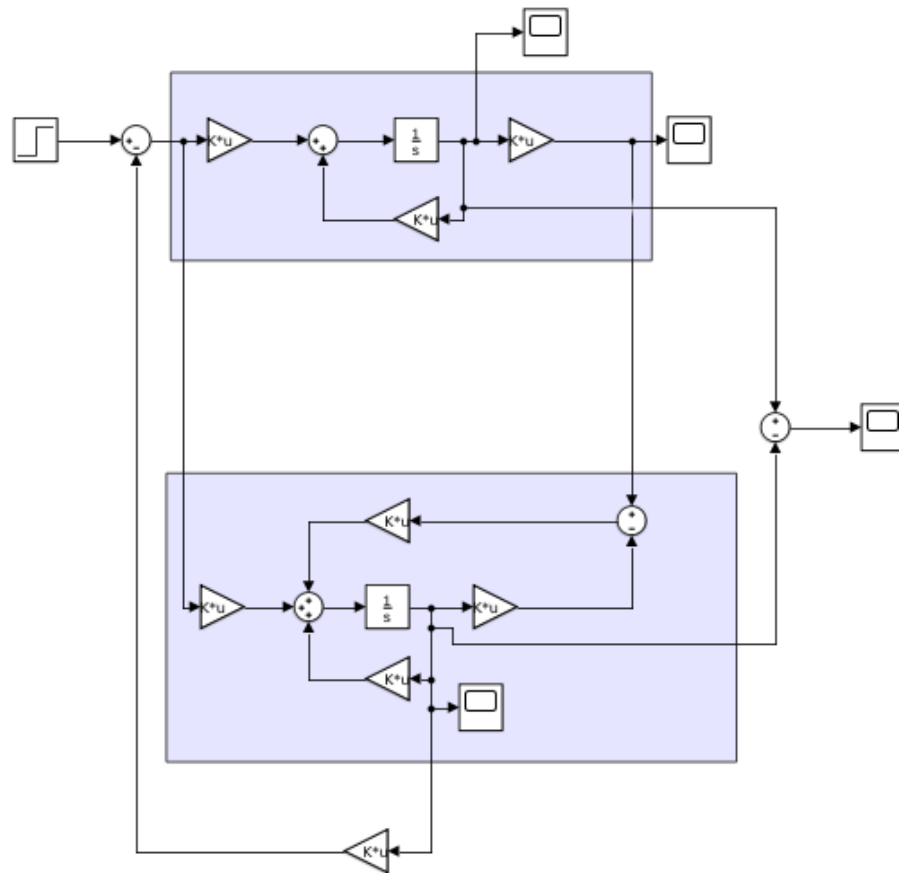
1.0e+06 *

0.0009

-0.0010

-1.0583

1.2631



شکل - با رویتگر

طراحی رویتگر کاهش یافته با قطب های 1- و 1- و 1- :

```
A_thilda = inv(Q)*A*Q
A11 = A_thilda(1:3,1:3)
A21 = A_thild
A12 = A_thild
A22 = A_thilda(4,4)

L_thilda = (D - A11)*A21'*inv(A21*A21');
To = A12 + L_thilda*A22 - D*L_thilda
L2 = [eye(3) L_thilda]*inv(Q)
R = L2*B
aa = inv([C;L2]);
F1 = aa(1:4,1)
F2 = aa(:,2:4)
```

شکل - با رویتگر

To =

```
1.0e+03 *  
0  
-1.0296  
1.2197
```

L2 =

```
1.0e+03 *  
0.0010    0    0    0  
0    -1.0296    0.0010    0  
0    1.2197    0    0.0010
```

R =

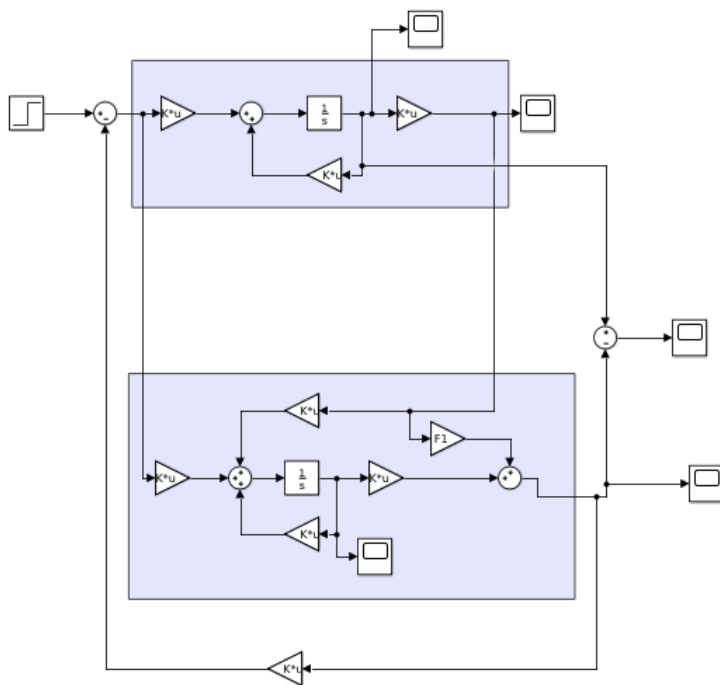
```
0  
-64.3859  
472.8322
```

F1 =

```
1.0e+03 *  
0  
0.0010  
1.0296  
-1.2197
```

F2 =

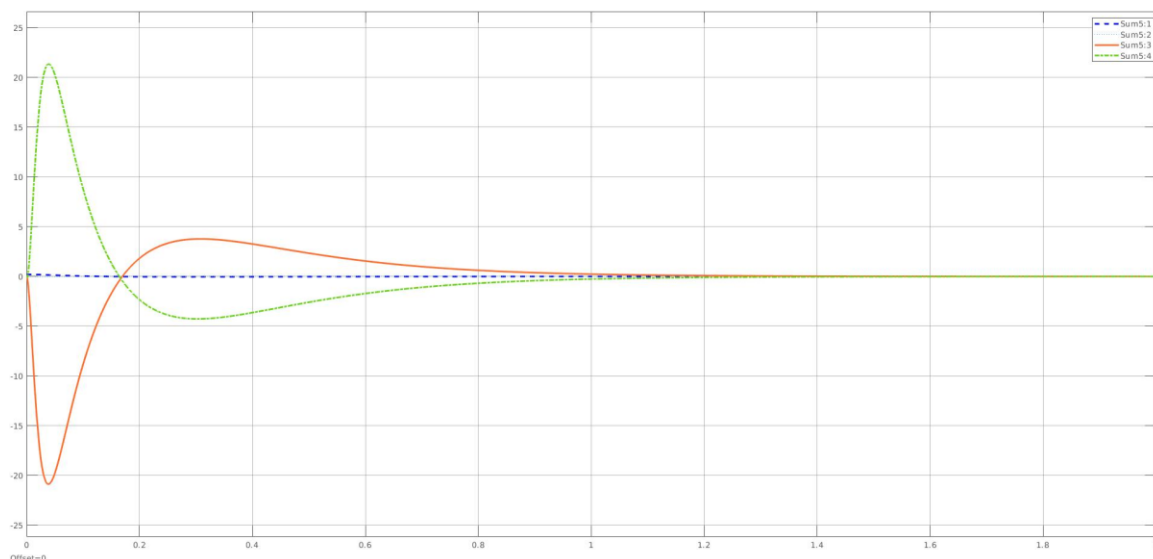
```
1.0000    0    0  
0    0    0  
0    1.0000    0.0000  
0    0    1.0000
```



شکل - با روینگر

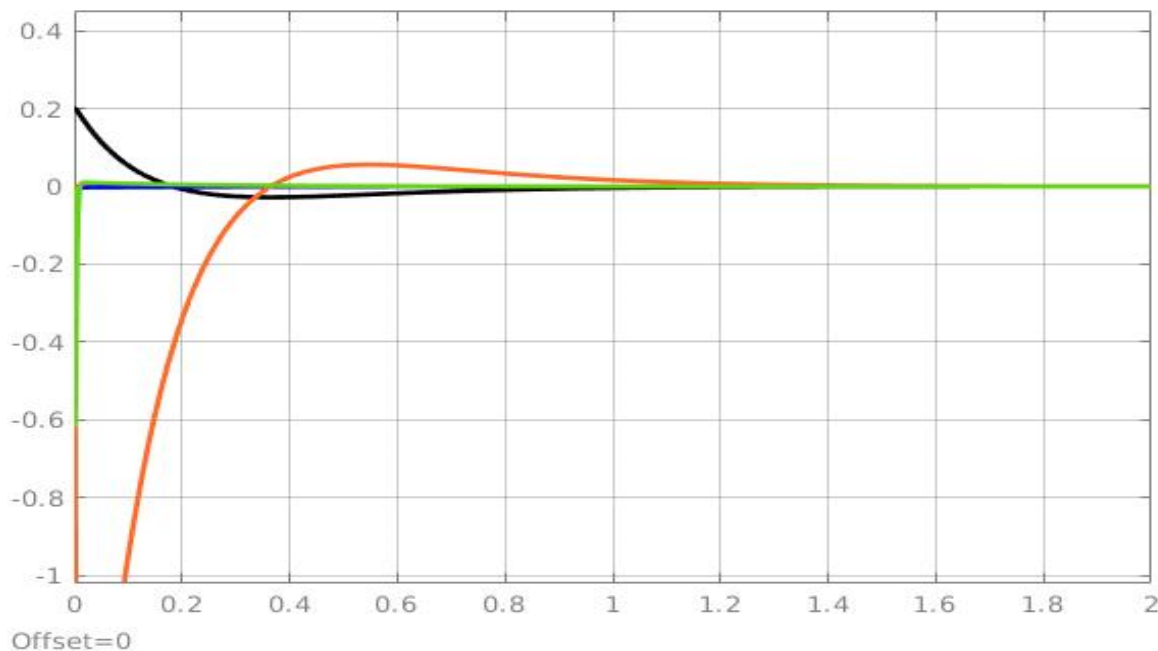
5 - در این قسمت فیدبک حالت طراحی شده را با کمک حالت های تخمین زده شده توسط رویتگر مورد استفاده قرار می دهیم. ابتدا نتایج رویتگر کامل را مورد بررسی قرار می دهیم.

تصویر زیر اختلاف حالت های سیستم با حالت های تخمین زده شده در رویتگر کامل را نشان میدهد که در حضور فیدبک، خطای حالت های تخمین زده شده بعد از یک ثانیه به صفر همگرا می شود



شکل - اختلاف حالت های تخمینی با حالت های اصلی رویتگر کامل

رفتار حالت های سیستم در حضور فیدبک با رویتگر کامل و ورودی صفر و شرایط اولیه ۱۰ درجه انحراف از نقطه تعادل:

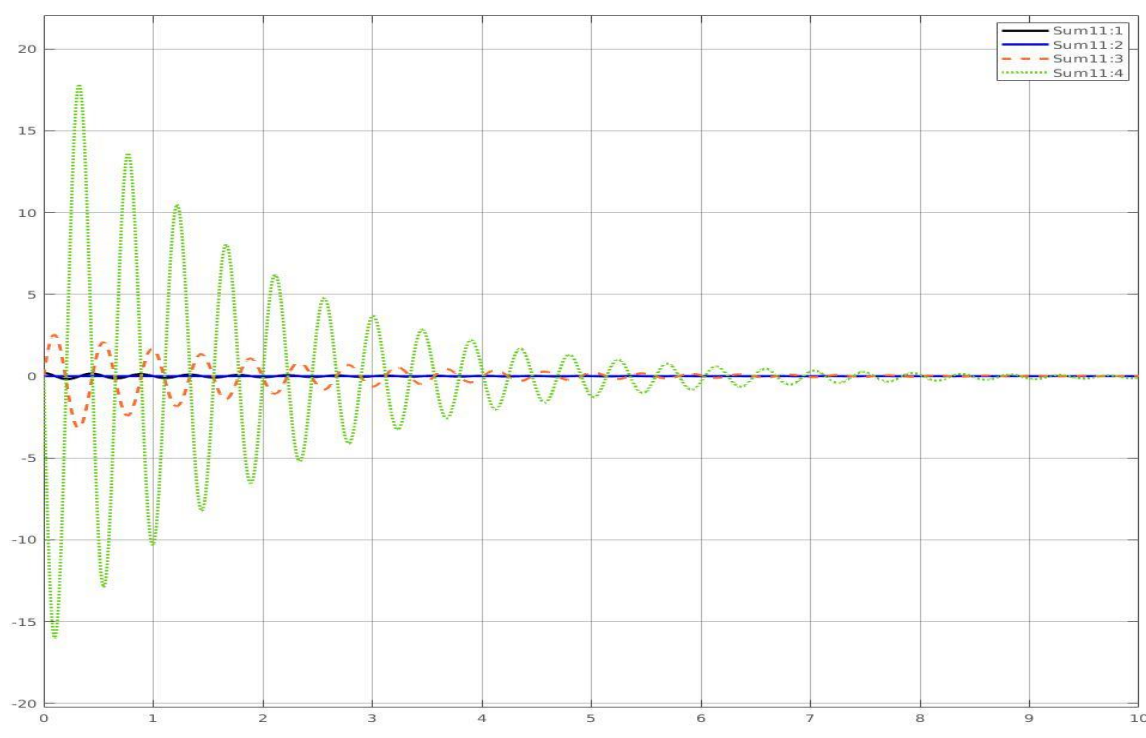


شکل - اختلاف حالت های تخمینی با حالت های اصلی رویتگر کامل و فیدبک حالت

نتایج شبیه سازی برای سیستم غیر خطی عینا مشابه نتایج سیستم خطی سازی شده می باشد.

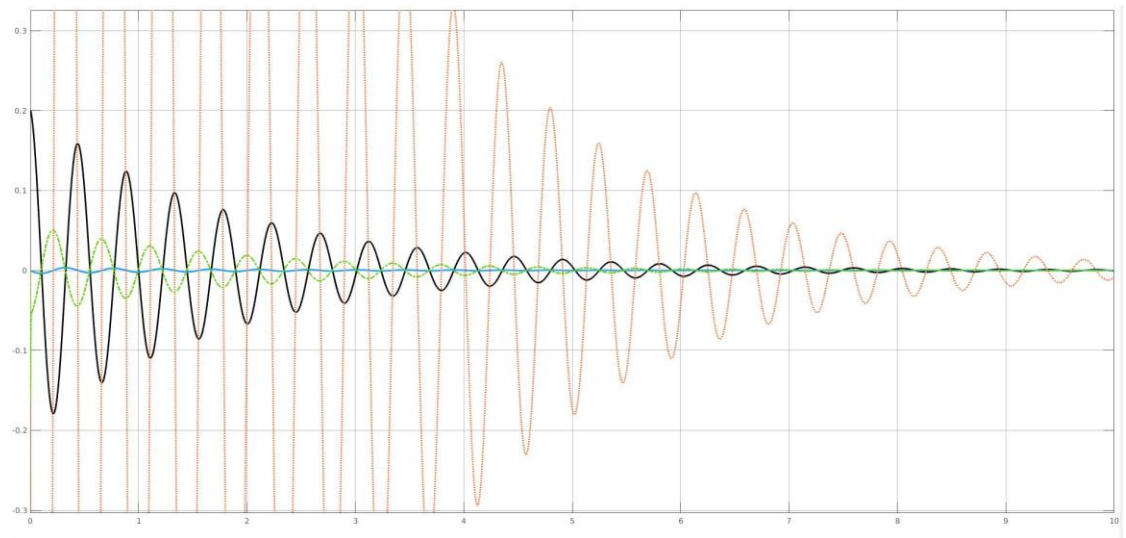
حال به بررسی عملکرد رویتگر کاهش یافته با قطب های $1-$ و $1-$ می پردازیم. لازم به ذکر است که طبق نتایج شبیه سازی با دور تر کردن قطب ها در این رویتگر سیستم به سمت ناپایداری می رود.

تصویر زیر اختلاف حالت های سیستم با حالت های تخمین زده شده در رویتگر کاهش یافته را نشان میدهد که در حضور فیدبک، خطای حالت های تخمین زده شده بعد از ۹ ثانیه به صفر همگرا می شود.



شکل – اختلاف حالت های تخمینی با حالت های اصلی رویتگر کامل و فیدبک حالت در سیستم غیر خطی

رفتار حالت های سیستم در حضور فیدبک با رویتگر کاهش یافته و ورودی صفر و شرایط اولیه 10° درجه انحراف از نقطه تعادل (ربات حدود ۹ ثانیه حول نقطه تعادل حرکت نوسانی می کند):



1. نقطه تعادل در سیستم غیرخطی همان نقطه کار مورد بررسی در قسمت های قبل است. این سیستم مانند پاندول معکوس دو نقطه تعادل دارد، یکی نقطه تعادل پایدار (پایین ترین نقطه) و دیگری نقطه تعادل ناپایدار (در بالاترین نقطه) که مورد بررسی و نقطه کار سیستم ماست.

$$\bar{x} = [0 \ \phi_x \ 0 \ 0]^T$$

- 3 با بهینه سازی تابع هزینه به فرم زیر و تعریف ماتریس های Q و R ماتریس های گین سیستم را محاسبه می کنیم. با در نظر گرفتن تمام حالت های سیستم به فرم زیر، ماتریس R , Q را تعریف می کنیم (سیستم ۸ متغیر حالت و ۲ ورودی دارد) :

$$Q = \begin{bmatrix} 60000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 60000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 400 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 400 \end{bmatrix}$$

$$R = \begin{bmatrix} 1000 & 0 \\ 0 & 1000 \end{bmatrix}$$

انتخاب ماتریس Q وابسته به اهمیت پارامتر هایی است که می خواهیم سریع تر خطای آن ها صفر شود و با توجه به هدف ربات های تعادلی، زاویه انحراف اصلی ترین متغیر است که می خواهیم آن را صفر کنیم و به همین جهت بیشترین وزن را به آن می دهیم که در اینجا مقدار ۶۰۰۰۰ برای آن انتخاب شده است. انتخاب ماتریس R وابسته به اهمیت محدود کردن خروجی کنترلی است که به پیشنهاد خود مقاله ۱۰۰۰ انتخاب شده است.

با دستورات زیر در متلب گین بهینه را پیدا می کنیم:

```
sys = ss(A,B,C,[0]);  
Q = eye(4);  
Q(1,1) = 60000;Q(4,4) = 400;  
R = [1000];  
[K_lqr ,S ,e] = lqr(sys , Q ,R , 0)
```

K_lqr =

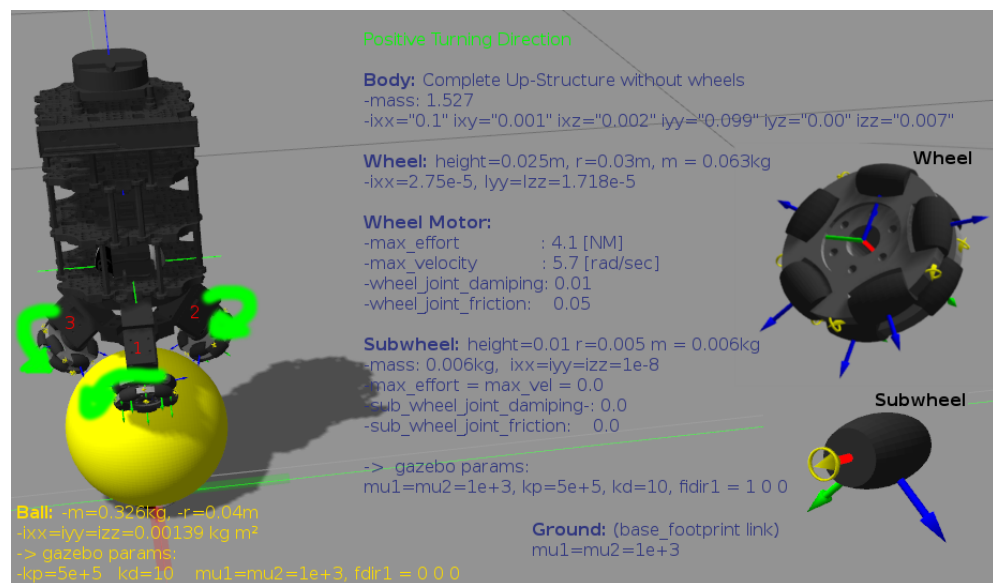
5.5165	-0.0316	0.1962	0.2340
--------	---------	--------	--------

شبیه سازی در محیط گزبو:

شبیه ساز گزبو (Gazebo) یک محیط شبیه سازی سه بعدی ربات است که در محیط لینوکس و در تعامل با ROS، فعالیت های ربات در تعامل با محیط را تحلیل می کند و بصورت گرافیکی در محیط سه بعدی نمایش می دهد.

برای استفاده از گزبو نیاز به مدل ربات مورد نظر داریم که در این پروژه از مدل ربات بال بات که توسط توسعه دهندگان ربات تر تل بات (turtlebot) طراحی شده است استفاده می کنیم.

<https://github.com/CesMak/bb>



شکل - شبیه سازی

در سایت گیت هاب فوق، علاوه بر مدل ربات کنترلری نیز به زبان ++C برای آن طراحی شده است که این ربات را پایدار می کند. اما به جهت عدم تطابق مدل دینامیکی مورد بررسی در این پروژه با ربات فوق، امکان پیاده سازی کنترلر طراحی شده روی این ربات ممکن نبود.