



دانشگاه صنعتی شروز
دانشکده مهندسی برق

پروژه کارشناسی
در رشته مهندسی برق گرایش کنترل

کنترل سیستم تعلیق گوی مغناطیسی با استفاده از مکانیزم تنظیم عصبی

نگارش:

شاهد مومن زاده

استاد راهنما:

دکتر زارعی

آبان ماه ۱۴۰۲

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

چکیده

کنترل سیستم تعلیق گوی مغناطیسی با استفاده از مکانیزم تنظیم عصبی

نگارش:

شاهد مومن زاده

سیستم تعلیق گوی مغناطیسی یک سیستم الکترومکانیکی است. به دلیل ماهیت و ساختار آن سیستمی غیرخطی و بسیار ناپایدار به شمار می‌رود. همچنین تابع تبدیل خطی سازی شده این سیستم نیز بسیار به مقادیر اولیه سیستم وابسته است. از این رو در ابتدا سعی می‌شود با استفاده از کنترل کننده خطی PID سیستم را کنترل نمود سپس به کمک فیدبک حالت سیستم پایدار می‌شود و در نهایت برای کنترل سیستم در شرایط اولیه مختلف به کمک شبکه عصبی مصنوعی، مکانیزمی طراحی می‌شود که به سبب آن سیستم توانایی پاسخگویی در مقادیر اولیه متفاوت را داشته باشد. شبیه سازی و مدلسازی این سیستم در محیط MATLAB و Simulink انجام می‌پذیرد.

کلمات کلیدی : تعلیق مغناطیسی، PID، فیدبک حالت، LQR، شبکه عصبی مصنوعی

فهرست مطالب

معرفی سیستم تعلیق گوی مغناطیسی	۱۰
مدلسازی ریاضی سیستم	۱۲
خطی سازی سیستم	۱۴
پیاده سازی در Simulink	۱۵
طراحی کنترل کننده PID	۲۰
طراحی کنترل کننده LQR	۲۵
فیدبک حالت	۲۵
کنترل کننده LQR	۲۶
پیاده سازی تنظیم مکانیزم عصبی	۳۴
شبکه عصبی مصنوعی	۳۴
پیاده سازی در محیط MATLAB	۳۵
مراجع	۴۳
پیوست	۴۴
رسم نمودار شبیه سازی	۴۴
شبیه سازی سه بعدی	۴۴

فهرست شکل ها

- تصویر ۱- نمونه آزمایشگاهی سیستم تعلیق گوی مغناطیسی..... ۸
- تصویر ۲- یک نمونه از قطار مغناطیسی..... ۹
- تصویر ۳- نمونه ای از وسایل تزئینی و سرگرمی..... ۹
- تصویر ۴- مدل فیزیکی و الکتریکی سیستم تعلیق مغناطیسی..... ۱۰
- تصویر ۵- مقادیر اولیه بلوک انتگرال گیر..... ۱۴
- تصویر ۶- بلوک دیاگرام کنترل کننده PID..... ۱۷
- تصویر ۷- محیط جعبه ابزار pidtool..... ۲۰
- تصویر ۹- محیط tuning tools..... ۲۰
- تصویر ۱۰- بلوک دیاگرام فیدبک حالت..... ۲۲
- تصویر ۱۱- مقادیر بلوک سیگنال سینوسی و پله..... ۲۹
- تصویر ۱۲- تغییر ورودی..... ۲۹
- تصویر ۱۳- فرآیند آموزش شبکه عصبی..... ۳۴
- تصویر ۱۴- بلوک تولید شده شبکه عصبی..... ۳۶

فهرست نمودارها

نمودار ۱ خروجی سیستم و ولتاژ ورودی.....	۱۸
نمودار ۲-خروجی سیستم حلقه باز در حضور اغتشاش.....	۱۹
نمودار ۳- پاسخ سیستم حلقه بسته با کنترل کننده PID.....	۲۴
نمودار ۴- خروجی به ازای ورودی ۱۰mm.....	۲۹
نمودار ۵- خروجی به ازای ورودی ۱۰mm و اغتشاش بیشتر.....	۳۰
نمودار ۶- خروجی به ازای ورودی ۲۰mm.....	۳۱
نمودار ۷- ورودی ۲۰mm و باز طراحی کنترل کننده.....	۳۱
نمودار ۸ - خروجی سیستم به ازای ورودی سینوسی.....	۳۳
نمودار ۹- خروجی به ازای ورودی پله ای.....	۳۴
نمودار ۱۰- تغییرات k بر حسب XSS.....	۳۷
نمودار ۱۱- نمودار عملکرد شبکه عصبی.....	۳۹
نمودار ۱۲- مقایسه خروجی شبکه عصبی با مقدار واقعی.....	۳۹
نمودار ۱۳- خروجی سیستم کنترل عصبی به ازای ورودی سینوسی.....	۴۲
نمودار ۱۴- تغییرات K.....	۴۲

فهرست سیستم ها

- سیستم ۱- نمای کلی سیستم حلقه باز..... ۱۶
- سیستم ۲- سیستم با ورودی زمان و اغتشاش..... ۱۸
- سیستم ۳- سیستم کنترل فیدبک حلقه بسته..... ۲۸
- سیستم ۴- اضافه کردن شبکه عصبی به سیستم..... ۴۰
- سیستم ۵- تکمیل سیستم حلقه بسته..... ۴۱

فهرست جدول ها

- جدول ۱- پارامترهای فیزیکی و الکتریکی سیستم..... ۱۵
- جدول ۲- ضرایب کنترل کننده PID..... ۲۴

فهرست نشانه های اختصاری

v	ولتاژ
i	جریان
m	جرم
t	زمان
R	مقاومت
K_f	ثابت نیرو
g	شتاب گرانش
k	بهره حالت

معرفی سیستم تعلیق گوی مغناطیسی

سیستم تعلیق گوی مغناطیسی یک سیستم الکترومکانیکی است که از نیروی مغناطیسی برای معلق نگه داشتن یک گوی در فضای خالی استفاده می‌کند. این سیستم ها معمولاً از یک یا چند سیم پیچ مغناطیسی تشکیل شده‌اند که در اطراف گوی قرار گرفته‌اند. هنگامی که جریان الکتریکی از طریق سیم پیچ ها عبور می‌کند، یک میدان مغناطیسی ایجاد می‌شود که گوی را به سمت خود جذب می‌کند. با تنظیم جریان الکتریکی، می‌توان موقعیت گوی را در فضای خالی کنترل کرد. (تصویر ۱)



تصویر ۱ - نمونه آزمایشگاهی سیستم تعلیق گوی مغناطیسی

یکی از مزایای اصلی سیستم های تعلیق گوی مغناطیسی این است که هیچ تماس فیزیکی بین گوی و سیم پیچ ها وجود ندارد. این امر باعث کاهش اصطکاک و افزایش بهره‌وری می‌شود. علاوه بر این، سیستم های تعلیق گوی مغناطیسی بسیار دقیق هستند و می‌توانند گوی را در فضای خالی با دقت بسیار بالایی کنترل کنند.

از سیستم های تعلیق گوی مغناطیسی در زمینه های مختلفی استفاده می‌شود که می‌توان به قطارهای سریع‌السیر (تصویر ۲) اشاره کرد. به دلیل عدم وجود اصطکاک این نوع از قطارها توانایی حرکت با سرعت های بالا را دارا هستند.



تصویر ۲- یک نمونه از قطار مغناطیسی

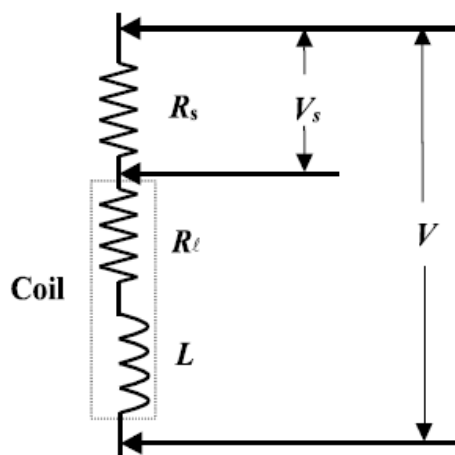
همچنین از این سیستم به دلیل وجود جذابیت بصری بالا، در وسایل سرگرمی و دکوراسیون نیز استفاده می‌شود. (تصویر ۳)



تصویر ۳- نمونه ای از وسایل تزیینی و سرگرمی

مدلسازی ریاضی سیستم

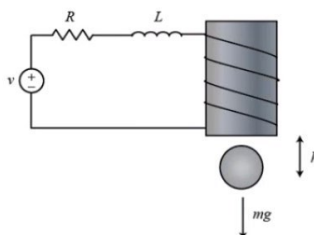
سیستم تعلیق گوی مغناطیسی را می‌توان به دو زیر سیستم تقسیم بندی کرد : ۱- قسمت الکتریکی ۲- قسمت مکانیکی



تصویر ۴ مدل الکتریکی سیستم تعلیق مغناطیسی

تصویر ۴ نمایی از نسخه مدل شده زیر سیستم الکتریکی تعلیق گوی مغناطیسی است که L اندوکتانس و R_l مقاومت سیم پیچ هستند. با اعمال ولتاژ V در مدار جریان i جاری می‌شود که در سیستم واقعی با اندازه گیری ولتاژ V_s ، این جریان اندازه گیری می‌شود. معادله الکتریکی مطابق معادله ۱ است.

$$V = i(R_l + R_s) + L \frac{di}{dt} \quad (1)$$



تصویر ۵- نمای کلی سیستم (بخش مکانیکی و الکتریکی)

همان طور که در تصویر ۵ مشخص است، نیروی گرانش mg به گوی وارد می شود. y ارتفاع گوی از مبدا مختصات (سطح زمین) است. از قانون دوم نیوتن معادله ۲ نوشته می شود.

$$m \frac{d^2 y}{dt^2} = mg - K_f \left(\frac{i}{y} \right)^2 \quad (2)$$

پس از طی دوره گذرا و رسیدن به حالت ماندگار (steady state) در سیستم داریم :

$$V_{ss} = i(R_l + R_s) + 0 \quad (3)$$

$$0 = mg - K_f \left(\frac{i_{ss}}{y_{ss}} \right)^2 \quad (4)$$

از معادلات ۳ و ۴ مقادیر ولتاژ و جریان حالت ماندگار سیستم بدست می آید.

$$I_{ss} = y_{ss} \sqrt{\frac{mg}{K_f}} \quad (5)$$

$$V_{ss} = I_{ss}(R_l + R_s) \quad (6)$$

در این معادلات V_{ss} و I_{ss} ولتاژ و جریان نهایی و y_{ss} موقعیت نهایی عمودی گوی است .

واضح است که معادلات سیستم غیر خطی هستند، از آنجایی که برای طراحی کنترل کننده خطی باید یک سیستم خطی تغییر ناپذیر با زمان در اختیار داشته باشیم، نیاز است که سیستم را حول نقطه کار آن خطی سازی کرده و معادلات فضای حالت یا تابع تبدیل سیستم را بدست بیاوریم.

خطی سازی سیستم

برای خطی سازی، از معادلات ۱ و ۲ بدست آمده از قسمت مدل سازی استفاده می کنیم . با در نظر گرفتن بردار حالت X به صورت زیر معادلات را به صورت ماتریسی بازنویسی می کنیم.

$$X = [i(t), y(t), \dot{y}(t)]^T \quad (7)$$

معادلات غیرخطی سیستم به صورت زیر بازنویسی می شود (معادله ۸) :

$$\dot{X}(t) = \begin{bmatrix} -\frac{R_l + R_s}{L} X_1 \\ X_3 \\ g - \frac{K_f}{m} \left(\frac{X_1}{X_2}\right)^2 \end{bmatrix} + \begin{bmatrix} 1 \\ \frac{1}{L} \\ 0 \\ 0 \end{bmatrix} V \quad (8)$$

$$\dot{X}(t) = f(x) + g(x)v$$

اگر i, v, y تغییرات کوچک حول نقطه کار سیستم تعریف کنیم، داریم :

$$\begin{cases} I(t) = I_{ss} + i(t) \\ X(t) = X_{ss} + x(t) \\ V(t) = V_{ss} + v(t) \end{cases} \quad (9)$$

با استفاده از معادله ۹ سیستم را حول نقطه کار خطی سازی می کنیم :

$$\frac{d}{dx}(x + X_{ss})(t) = \frac{\partial}{\partial x} f(x)(X - X_{ss}) + \frac{\partial}{\partial V} g(v)(V - V_{ss}) \quad (10)$$

در نهایت معادلات فضای حالت سیستم حول نقطه کار بدست می آید (معادله ۱۱) :

$$\dot{X} = \begin{bmatrix} -\frac{R_l + R_s}{L} & 0 & 0 \\ 0 & 0 & 1 \\ -\frac{2K_f}{m} \frac{I_{ss}}{X_{ss}^2} & \frac{2K_f}{m} \frac{I_{ss}^2}{X_{ss}^3} & 0 \end{bmatrix} X + \begin{bmatrix} \frac{1}{L} \\ 0 \\ 0 \end{bmatrix} v \quad (11)$$

پارامترهای سیستم در جدول ۱ آورده شده است :

Physical quantity	Symbol	Numerical value	units
Coil inductance	L	0.4125	Henry
Coil resistance	R_l	10	Ohm
Current sensor resistance	R_s	1	Ohm
Force constant	K_f	32654	$\frac{mN - mm^2}{Amp^2}$
Gravitational constant	g	9810	$\frac{mm}{s^2}$
Ball mass	m	0.068	Kg

جدول ۱- پارامترهای فیزیکی و الکتریکی سیستم

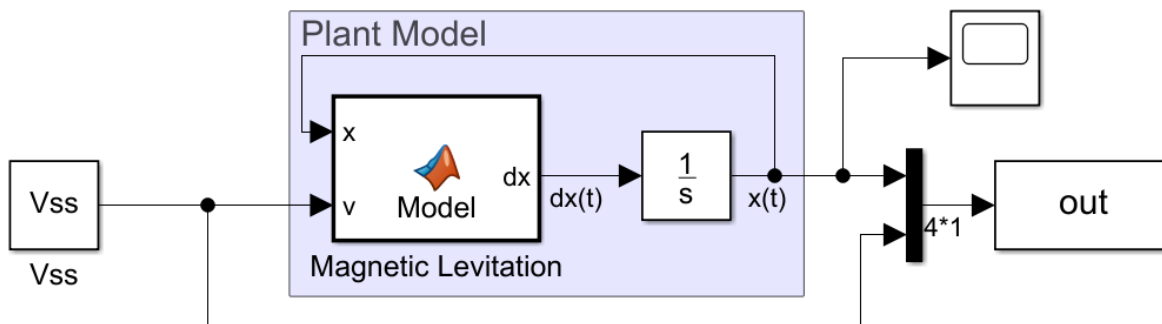
پیاده سازی در Simulink

جهت پیاده سازی سیستم، از معادلات غیرخطی سیستم (معادله ۸) استفاده می کنیم :

$$\dot{X}(t) = \begin{bmatrix} -\frac{R_l + R_s}{L} X_1 \\ X_3 \\ g - \frac{K_f}{m} \left(\frac{X_1}{X_2}\right)^2 \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \\ 0 \end{bmatrix} V \quad (12)$$

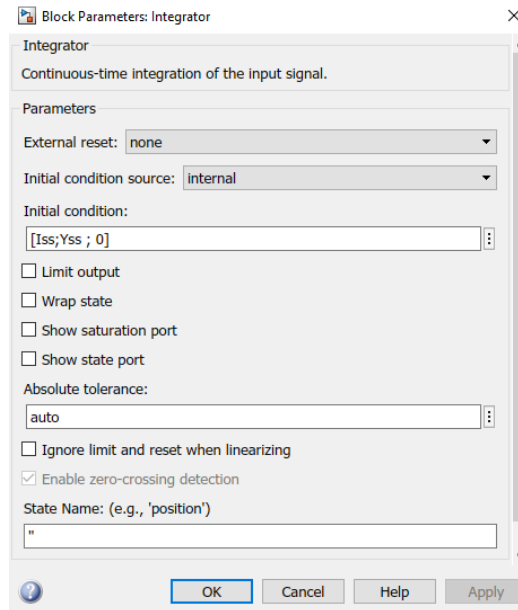
$$\dot{X}(t) = f(x) + g(x)v$$

برای این منظور از بلوک MATLAB Function استفاده می کنیم که خروجی این بلوک مشتق متغیرهای حالت می باشد. سپس با استفاده از یک انتگرال گیر که دارای شرایط اولیه ای که همان نقطه کار سیستم است متغیرهای حالت تولید می شود. مدل سیستم بدین صورت است :



سیستم ۱ - نمای کلی سیستم حلقه باز

و در بلوک انتگرال گیر بردار شرایط اولیه سیستم که شامل جریان و موقعیت و مشتق موقعیت (سرعت) در حالت ماندگار سیستم است وارد می شود.



تصویر ۶- مقادیر اولیه بلوک انتگرال گیر

بلوک MATLAB function مطابق کد زیر برنامه نویسی می شود :

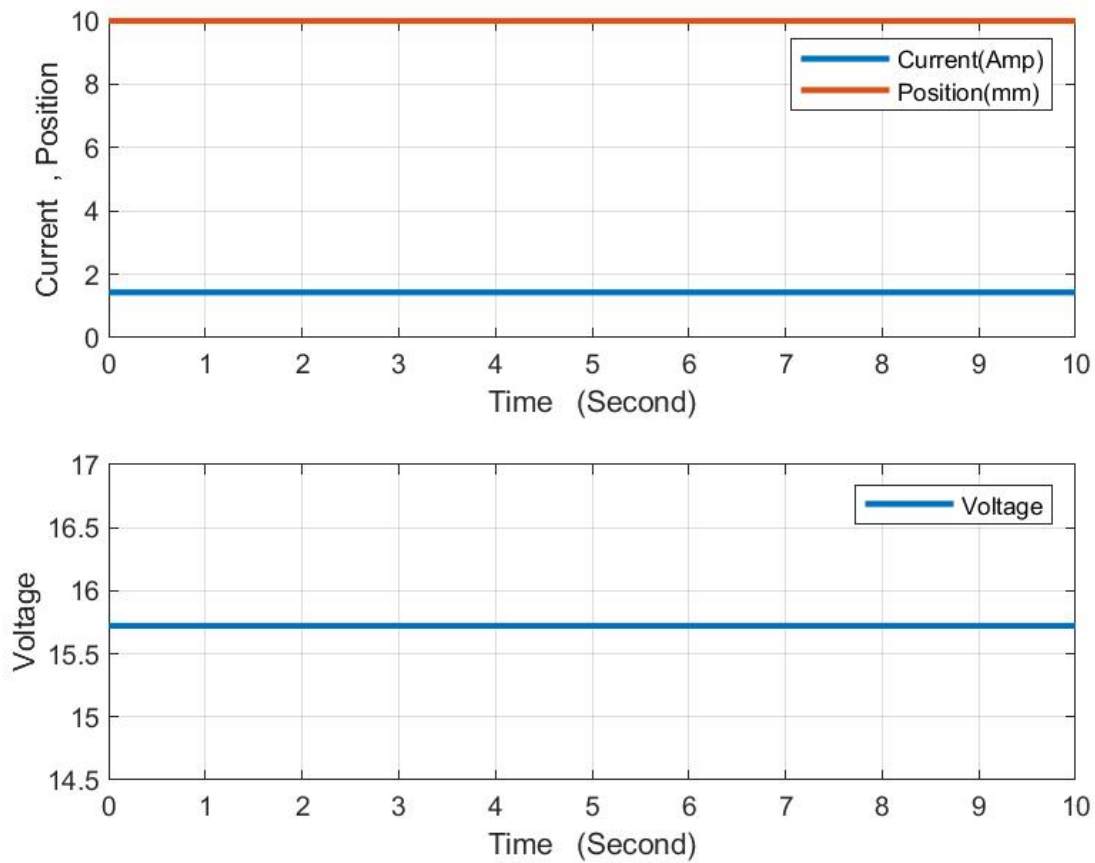
```
function dx = Model(x , v)

L = 0.4125 ; % Henry
RI = 10 ; % ohm
Rs = 1 ; % ohm
Kf = 32654e-9 ; % N*m^2/Amp^2
g = 9810e-3 ; % m/sec^2 ;
m = 0.068 ; % kg

dx = zeros(3,1) ;
dx(1) = -(RI + Rs)/L * x(1) + 1/L*v ;
dx(2) = x(3) ;
dx(3) = g - Kf/m*(x(1)/x(2))^2;
```

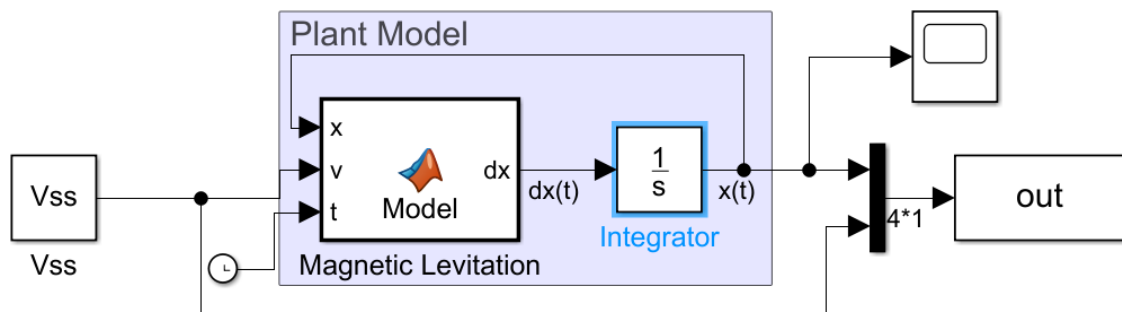
در ابتدا مقادیر سیستم وارد می شود و سپس توسط معادلات سیستم، مشتق متغیرهای حالت تولید می شود.

در ادامه خروجی بلوک وارد انتگرال گیر شده و متغیرهای حالت سیستم ساخته می شود.



نمودار ۱ خروجی سیستم و ولتاژ ورودی

با اضافه کردن ورودی زمان به MATLAB function یک اغتشاش وارد می کنیم :



سیستم ۲- سیستم با ورودی زمان و اغتشاش

`function dx = Model(x , v, t)`

`L = 0.4125 ; % Henry`

`RI = 10 ; % ohm`

$R_s = 1$; % ohm

$K_f = 32654e-9$; % $N \cdot m^2 / Amp^2$

$g = 9810e-3$; % m/sec^2 ;

$m = 0.068$; % kg

$F = 0.01 \cdot (t \geq 5) \cdot (t \leq 5.001)$; %Disturbance at 5th second

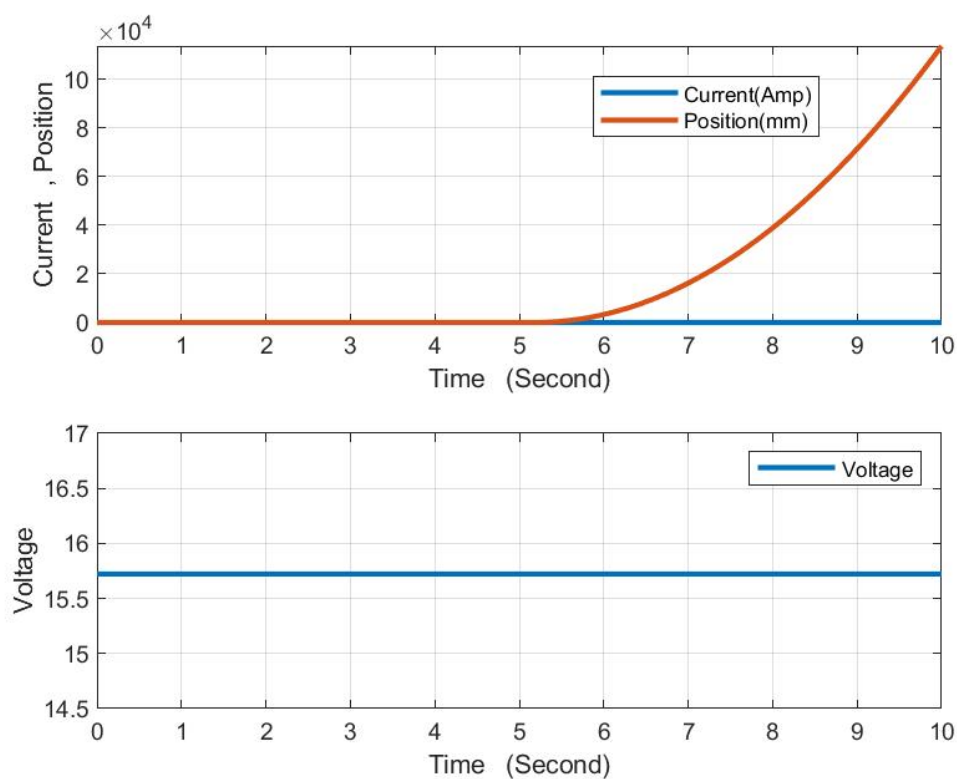
$dx = \text{zeros}(3,1)$;

$dx(1) = -(R_l + R_s)/L \cdot x(1) + 1/L \cdot v$;

$dx(2) = x(3)$;

$dx(3) = g + F/m - K_f/m \cdot (x(1)/x(2))^2$;

در لحظه $t=5s$ یک اغتشاش به سیستم وارد می شود و خروجی سیستم مطابق نمودار ۲ می شود.



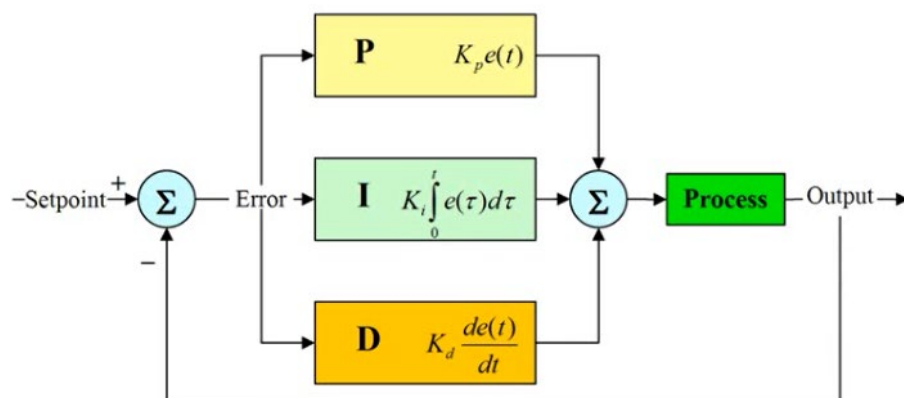
نمودار ۲-خروجی سیستم حلقه باز در حضور اغتشاش

مشخص است که سیستم بسیار ناپایدار بوده و با وارد شدن اغتشاش، خروجی آن بیقران می شود.

طراحی کنترل کننده PID

کنترل کننده PID یک کنترل کننده خطی است و تابع تبدیل کلی آن مطابق زیر است :

$$C(s) = K_p + \frac{K_i}{s} + K_d s \quad (13)$$



تصویر ۷ - بلوک دیاگرام کنترل کننده PID

برای طراحی کنترل کننده PID به معادلات فضای حالت سیستم یا تابع تبدیل آن نیاز داریم. در قسمت قبل پس از خطی سازی سیستم ، فرم فضای حالت آن را بدست آوردیم .

$$\dot{X} = \begin{bmatrix} -\frac{R_l + R_s}{L} & 0 & 0 \\ 0 & 0 & 1 \\ -\frac{2K_f}{m} \frac{I_{ss}}{X_{ss}^2} & \frac{2K_f}{m} \frac{I_{ss}^2}{X_{ss}^3} & 0 \end{bmatrix} X + \begin{bmatrix} \frac{1}{L} \\ 0 \\ 0 \end{bmatrix} v = AX + BU \quad (14)$$

$$Y = [0 \ 1 \ 0]X$$

خروجی مورد نظر سیستم موقعیت عمودی سیستم یا همان $y(t)$ است که دومین درایه بردار X است. به همین دلیل بردار خروجی C برابر $[0 \ 1 \ 0]$ است.

یک فایل جدید در MATLAB ایجاد کرده و نام آن را DesignController قرار می‌دهیم.
سپس سیستم را در آن تعریف می‌کنیم:

```
clc;
clear;
close all;
%%
L = 0.4125 ; % Henry
RI = 10 ; % ohm
Rs = 1 ; % ohm
Kf = 32654e-9 ; % N*m^2/Amp^2
g = 9810e-3 ; % m/sec^2 ;
m = 0.068 ; % kg
Xss = 20e-3 ;
lss = Xss * sqrt(m*g/Kf) ;
Vss = (RI + Rs) * lss ;
a = 2*Kf/m * lss/Xss^2 ;
b = 2*Kf/m * lss^2/Xss^3 ;
A = [-(RI + Rs)/L    0    0
      0             0    1
      -a            b    0];
B = [1/L  0  0]';
C = [0  1  0];
D = 0 ;
state_space = ss(A, B, C, D) ;
```

بعد از ایجاد فرم فضای حالت سیستم، قطب های سیستم را بدست می‌آوریم :

```
>> pole(state_space)
```

```
ans =
```

```
31.3209
-31.3209
-26.6667
```

سیستم مرتبه ۳ است و دارای ۱ قطب در نیم صفحه راست است که باعث ناپایداری سیستم حلقه باز می‌شود.

با استفاده از تابع $tf()$ تابع تبدیل سیستم بدست می‌آید :

```
system = tf(ss(A , B , C , D)) ;
```

```
>> system
```

```
system =
```

$$\frac{-16.64}{s^3 + 26.67 s^2 - 981 s - 2.616e04}$$

همچنین فرم صفر و قطب تابع تبدیل به وسیله تابع `zpk()` بدست می آید :

```
zero_pole_gain = zpk(state_space) ;
```

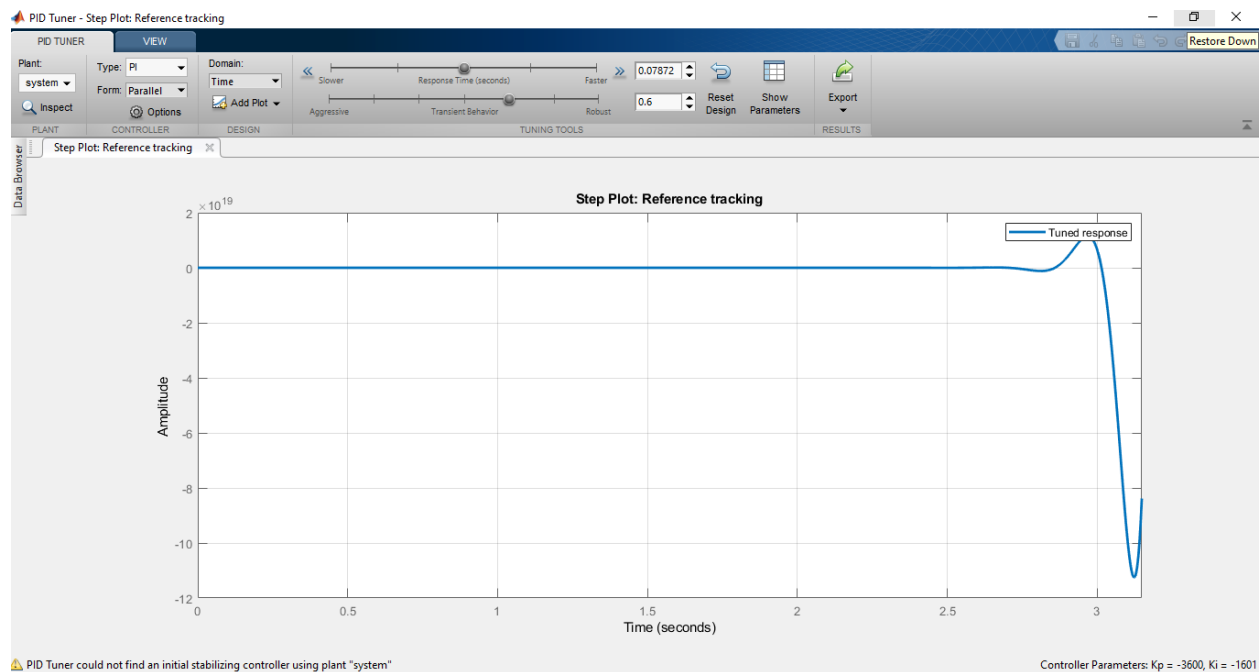
```
>> zero_pole_gain
```

```
zero_pole_gain =
```

$$\frac{-16.639}{(s-31.32)(s+31.32)(s+26.67)}$$

Continuous-time zero/pole/gain model.

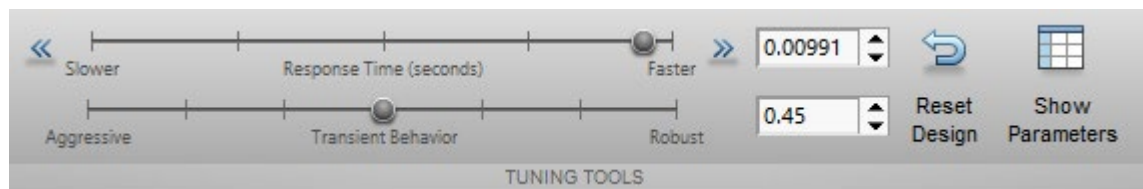
برای طراحی کنترل کننده از `toolbox` طراحی PID استفاده می کنیم. جهت فراخوانی این تولباکس دستور `pidtool(system)` را وارد می کنیم.



تصویر ۸- محیط جعبه ابزار *pidtool*

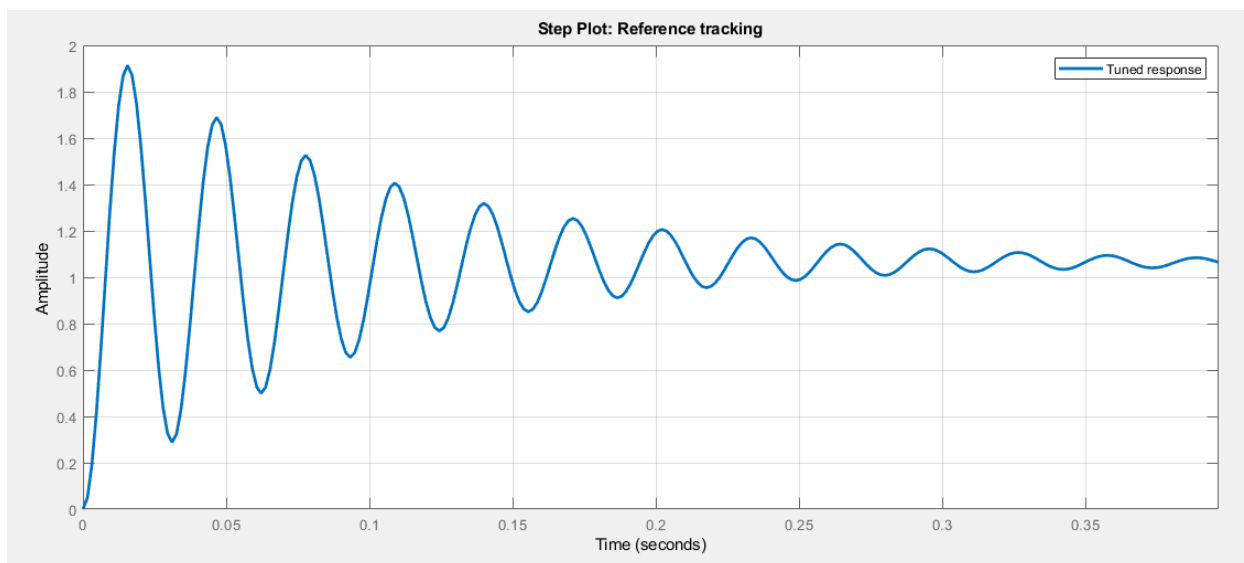
از قسمت بالا، میتوان نوع کنترل کننده مورد نظر را مشخص کرد، که با انتخاب نوع کنترل کننده *pid*، به طراحی می پردازد.

از قسمت *tuning tools* می توان تنظیمات دلخواه برای دریافت پاسخ مورد نظر را انجام داد .
(تصویر ۹)



تصویر ۹ محیط *tuning tools*

با تغییر مقدار *Response Time* سیستم تقریباً به حالتی پایدار می رسد که پاسخ سیستم حلقه بسته در نمودار ۳ آورده شده است .



نمودار ۳ پاسخ سیستم حلقه بسته با کنترل کننده PID

پاسخ پله سیستم حلقه بسته پایدار است ولی مشکلی وجود دارد : نوسانات سیستم زیاد است و همچنین مقادیر کنترل کننده بسیار بزرگ هستند و در واقعیت طراحی و ساخت چنین کنترل کننده ای ناممکن است .

برای دسترسی به ضرایب کنترل کننده در بخش tuning tools گزینه show parameters را انتخاب کرده تا مقادیر نمایش داده شوند.

Controller Parameters	
	Tuned
Kp	-17858.9858
Ki	-31541.2172
Kd	-2527.9888
Tf	n/a

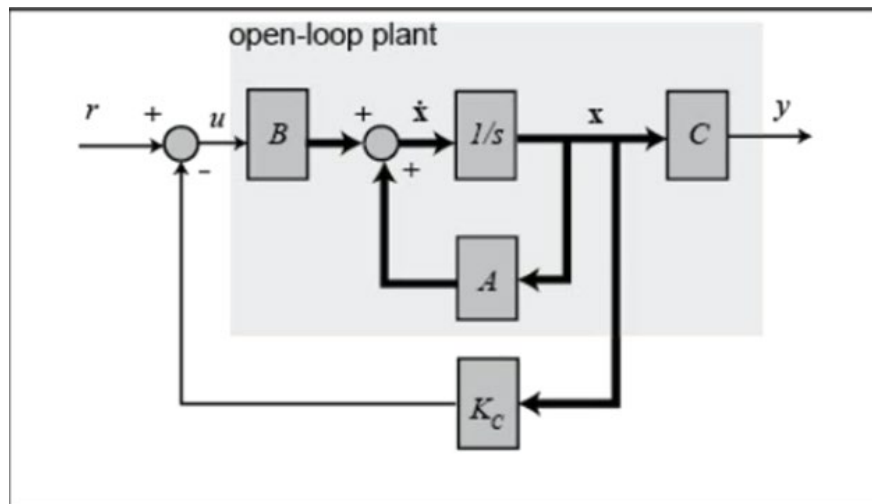
جدول ۲- ضرایب کنترل کننده PID

کاملاً مشخص است که ضرایب کنترل کننده PID ، حتی در بهینه ترین حالت بسیار بزرگ هستند و استفاده از کنترل کننده PID منطقی نیست.

طراحی کنترل کننده LQR

فیدبک حالت

سیستم مورد نظر بررسی دارای قطب در نیم صفحه راست است و ناپایدار است. از این رو نیاز به روشی داریم که قطب های سیستم حلقه بسته را به نیم صفحه چپ (LHP) منتقل کند تا سیستم پایدار شود. فیدبک حالت یکی از این روش ها است. با فیدبک متغیرهای حالت به جای فیدبک خروجی، می توان به شرط کنترل پذیری حالت سیستم، آن را پایدار کرده و به هر معادله مشخصه مطلوب حلقه بسته دست یافت. تصویر ۸ نمای کلی سیستم کنترل فیدبک حالت را نشان می دهد.



تصویر ۱۰ بلوک دیاگرام فیدبک حالت

بردار بهره فیدبک حالت (K) را باید به گونه ای انتخاب کنیم که معادله مشخصه جدید سیستم معادل با معادله مشخصه مطلوب بدست آمده از قطب های مورد نظر باشد.

فرم کلی تحقق فضای حالت یک سیستم خطی LTI به صورت زیر است :

$$\dot{X} = AX(t) + bu(t) \quad (15)$$

$$y(t) = cX(t)$$

همچنین معادله مشخصه این سیستم $a(s) = \det(sI - A)$ است.

با تعریف بردار بهره فیدبک $K = [k_1, k_2, \dots, k_n]$ کنترل کننده فیدبک حالت عبارت است از :

$$u(t) = -kX(t) \quad (16)$$

با جایگذاری $u(t)$ در معادله فضای حالت سیستم ، معادله سیستم حلقه بسته بدست می آید :

$$\dot{x}(t) = (A - bK)x(t) \quad (17)$$

و معادله مشخصه جدید آن $a_k = \det(sI - A + bK)$ است . اگر $\alpha(s)$ معادله مشخصه مطلوب باشد، برای تحقق طراحی و جاییابی تمامی قطب های حلقه بسته در مکان های مطلوب، باید K را به گونه ای انتخاب کنیم که $a_k = \alpha$ باشد.

کنترل کننده LQR

سیستم خطی LTI زیر را در نظر بگیرید:

$$\dot{x} = Ax + B \quad (18)$$

می خواهیم کنترل فیدبک حالت $u = -Kx$ را به گونه ای طراحی کنیم که معیار عملکرد زیر (معادله ۱۹) را کمینه سازد.

$$J = \int_0^{\infty} [x^T Q x + u^T R u] dt \quad (19)$$

که در آن ماتریس های Q و R متقارن بوده و $Q \geq 0, R > 0$ می باشد.

معادله سیستم حلقه بسته و معیار عملکردی آن به صورت زیر (معادله ۲۰ و ۲۱) خواهد بود:

$$\dot{x} = (A - Bk)x \quad (20)$$

$$J = \int_0^{\infty} x^T(t)(Qx + k^T R k)x(t) dt \quad (21)$$

با حل مسئله فوق، ضریب بهره k بهینه متناسب با ماتریس ضرایب Q, R بدست می‌آید.

برای طراحی کنترل کننده LQR از تابع `lqr()` استفاده می‌کنیم. این تابع با دریافت مقادیر ماتریس های A و B و Q و R مقدار بهینه بردار حالت K را بدست می‌آورد.

```
n = size(A , 1) ;  
Q = eye(n) ;  
R = 1 ;  
N = 0 ;  
K = lqr(A , B , Q , R , N) ;
```

مقدار بهره K برابر است با :

```
>> K  
  
K =  
  
1.0e+03 *  
  
0.0366 -8.3805 -0.1892
```

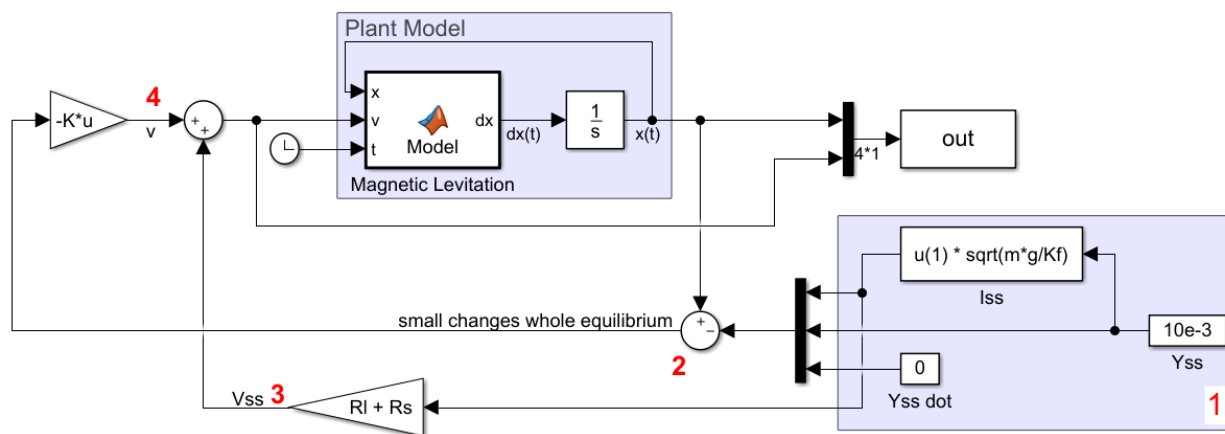
سیستم حلقه بسته به فرم $A-B*K$ بدست می‌آید. برای بدست آوردن قطب های سیستم حلقه بسته مقادیر ویژه ماتریس $A-B*K$ را محاسبه می‌کنیم :

```
>> eig(A-B*K)  
  
ans =  
  
-44.2999 + 0.4713i  
-44.2999 - 0.4713i  
-26.7671 + 0.0000i
```

سیستم حلقه بسته بدست آمده دارای دو قطب مختلط و یک قطب حقیقی در سمت چپ محور حقیقی می‌باشد و سیستم دارای شرط کافی پایداری می‌باشد.

در ادامه جهت شبیه سازی به Simulink رفته و به طراحی سیستم حلقه بسته می‌پردازیم.

نمای کلی سیستم حلقه بسته شبیه سازی شده در ادامه (سیستم ۳) آورده شده است.



سیستم ۳- سیستم کنترل فیدبک حلقه بسته

- بخش های شماره گذاری شده است در سیستم ۳ در ادامه توضیح داده خواهد شد:
- ۱- در این قسمت مقادیر steady state یا set point وارد شده است . مقدار ورودی Yss است و از روی آن مقدار I_{ss} محاسبه می شود.
- ۲- از بخش خطی سازی داشتیم :

$$\begin{cases} I(t) = I_{ss} + i(t) \\ X(t) = X_{ss} + x(t) \\ V(t) = V_{ss} + v(t) \end{cases}$$

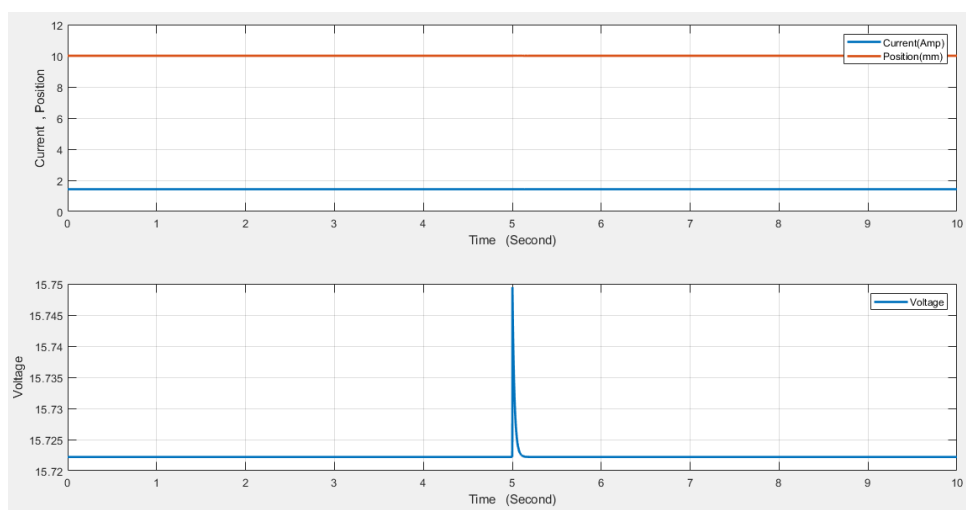
مقادیر $i(t)$, $x(t)$, $v(t)$ تغییرات کوچک حول نقطه تعادل سیستم هستند و حاصل جمع آن ها با مقادیر steady state خروجی سیستم را تشکیل می دهند. ورودی کنترل کننده همان مقادیر تغییرات کوچک حول نقطه تعادل هستند. در بخش شماره ۲ با کم کردن مقادیر SS از خروجی ، این تغییرات کوچک تولید می شود و وارد بلوک ضرب کننده با ضریب $-K$ می شود. ضریب منفی به دلیل فیدبک منفی بودن سیستم حلقه بسته است.

- ۳- در این قسمت مقدار $V_{ss} = I_{ss}(R_l + R_s)$ محاسبه می شود.

۴- خروجی ضرب کننده با ضریب $-K$ ، تغییرات کوچک ولتاژ اعمالی حول V_{SS} است. برای اینکه ورودی $V(t)$ سیستم تولید شود، خروجی با V_{SS} جمع می‌شود.

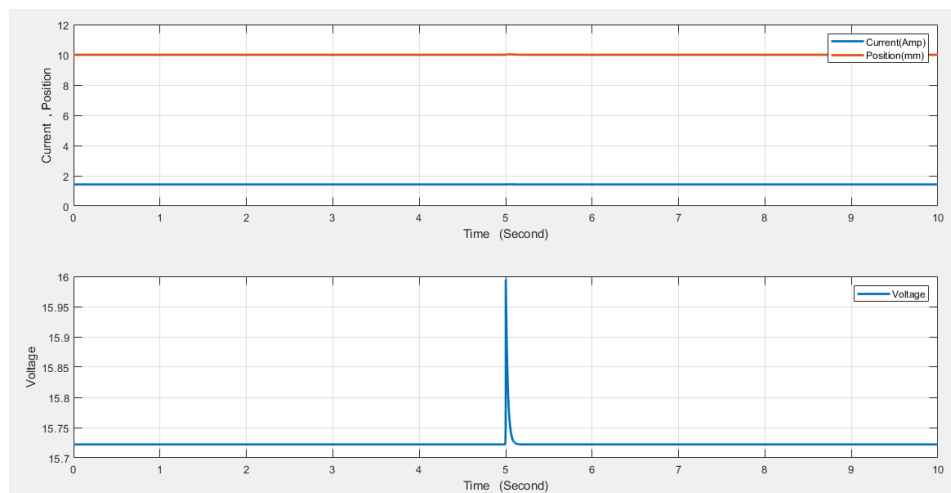
در ادامه به بررسی عملکرد سیستم می‌پردازیم:

در حالت اول نقطه تعادل مورد نظر را 10mm قرار می‌دهیم خروجی سیستم مطابق با نمودار ۴ است. مشاهده می‌شود در ثانیه ۵ اغتشاش به سیستم وارد می‌شود و برخلاف حالت حلقه باز که سیستم ناپایدار می‌شد، سیستم حلقه بسته به خوبی اغتشاش را حذف می‌کند.



نمودار ۴- خروجی به ازای ورودی 10mm

در ادامه دامنه اغتشاش را ده برابر می‌کنیم :

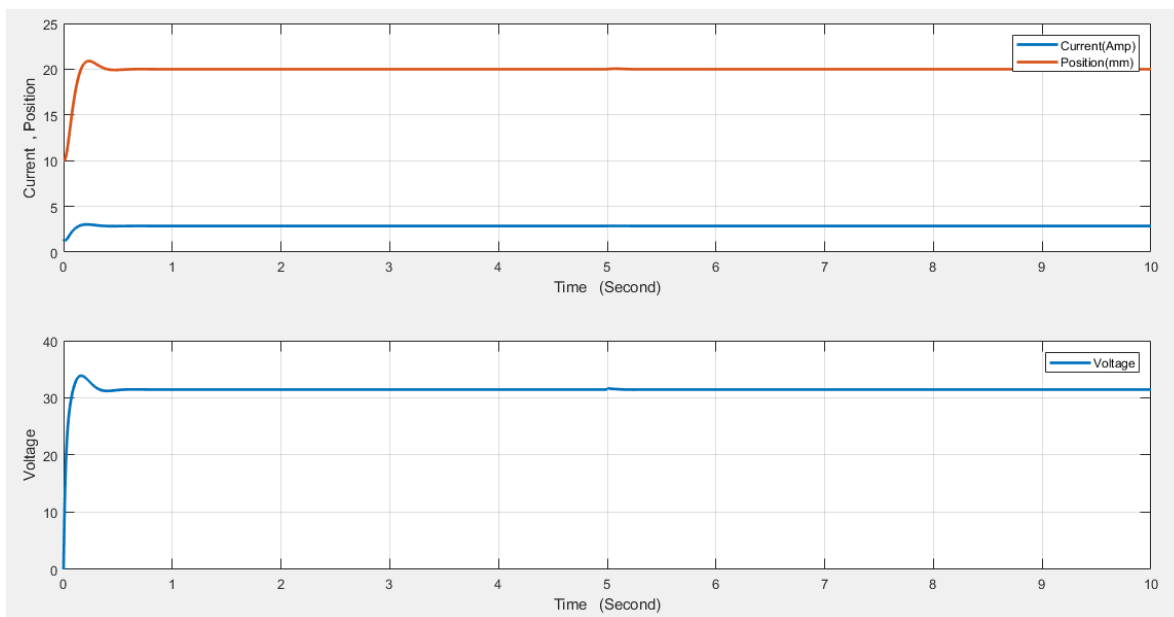


نمودار ۵- خروجی به ازای ورودی 10mm و اغتشاش بیشتر

کنترل کننده به ازای اعمال مقدار کمی ولتاژ بیشتر توانسته است اغتشاش را به خوبی حذف کند.(نمودار ۵)

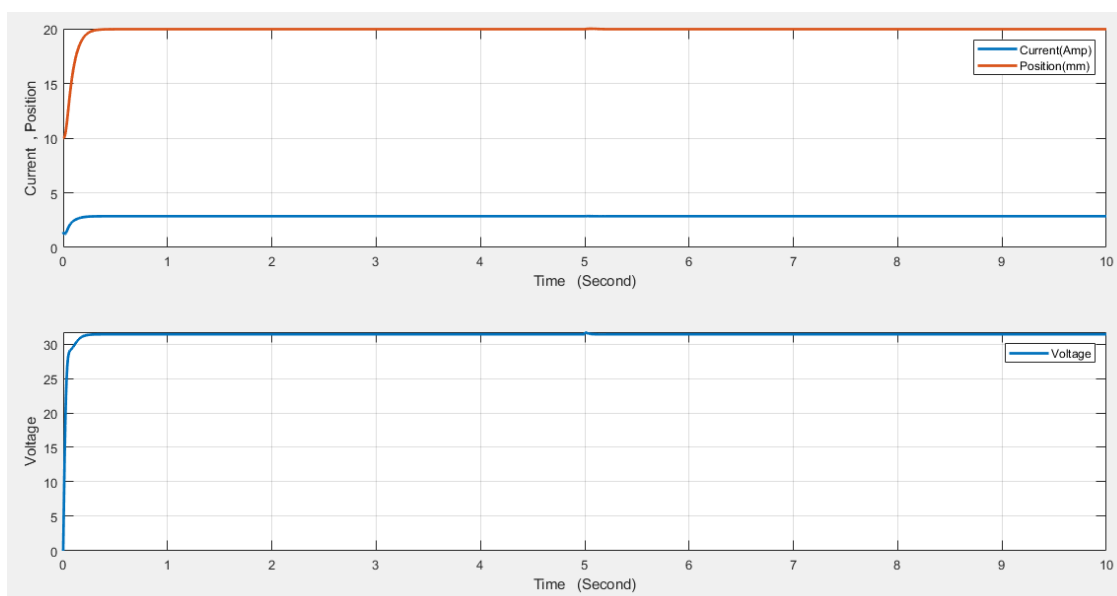
نکته ای که بسیار حائز اهمیت است این است که ماتریس A سیستم تعلیق گوی، به شدت به مقادیر نقطه کار سیستم (XSS) وابسته است و بردار کنترل کننده حالت K نیز بر اساس ماتریس A بدست آمده در این نقطه کار طراحی شده است و با تغییر نقطه کار سیستم ممکن است سیستم ناپایدار شود یا خروجی کنترل کننده سیگنال بزرگی شود که در عمل ساخت آن مشکل باشد.

برای مثال با همان کنترل کننده ای که برای نقطه تعادل 10mm طراحی شده است، اقدام به کنترل سیستم با ورودی 20mm می کنیم. مشاهده می شود که سیستم دارای بالازدگی و نوسان می شود.(نمودار ۶)



نمودار ۶- خروجی به ازای ورودی 20mm

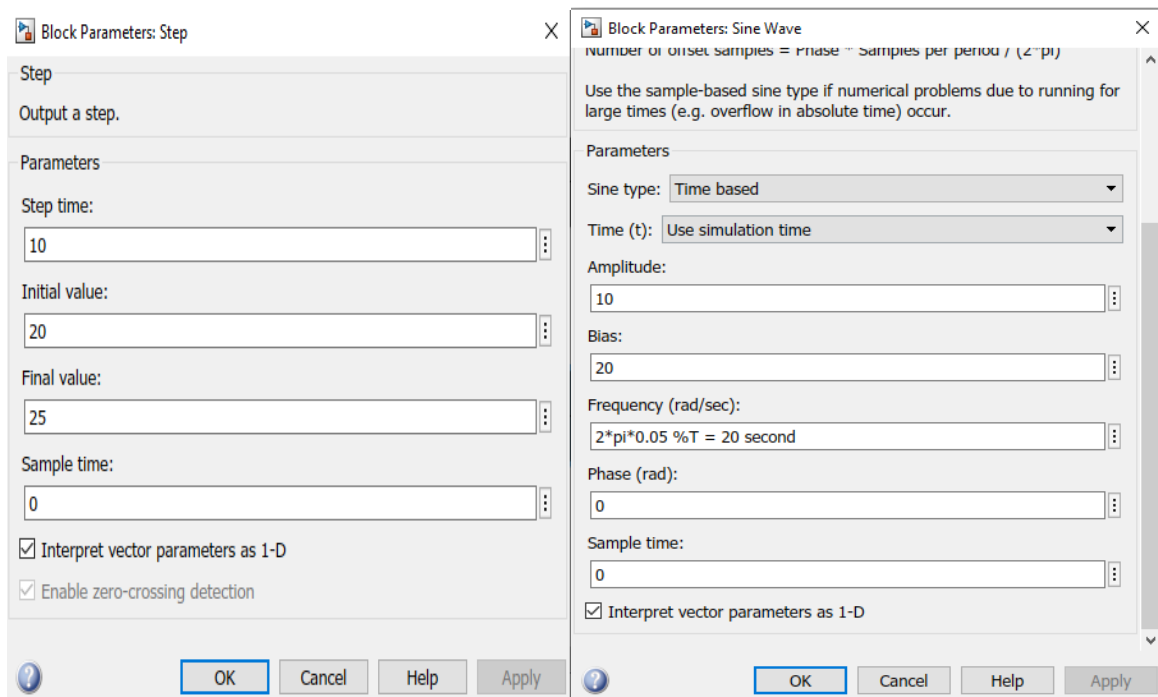
سپس با تغییر مقدار Xss در کد طراحی کنترل کننده از 10mm به 20mm اقدام به باز طراحی کنترل کننده می کنیم.



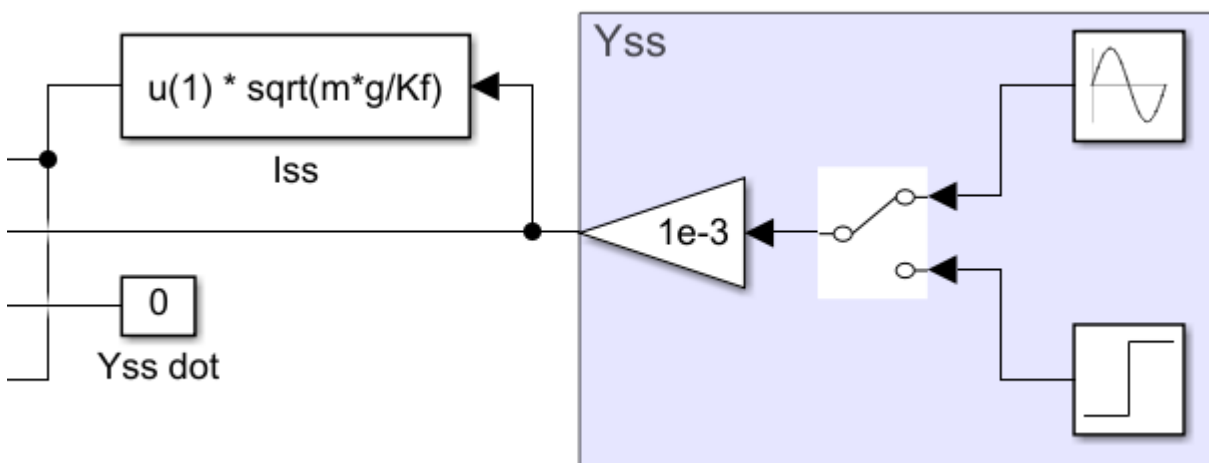
نمودار ۷- ورودی 20mm و باز طراحی کنترل کننده

از مقایسه نمودار ۷ و ۶ مشخص است موقعیت گوی و ولتاژ اعمالی به آن دارای بالازدگی نیست و پاسخ بهتری را شاهد هستیم.

البته می‌توان نشان داد پاسخ سیستم حوالی نقطه کار تقریباً پاسخ خوبی است و کنترل کننده می‌تواند پاسخگوی نیاز سیستم جهت پایداری باشد. برای این منظور دو ورودی جدید تعریف می‌کنیم: ۱- ورودی پله ای با مقدار اولیه 20mm که در دهمین ثانیه به 25mm می‌رود و ۲- یک سیگنال سینوسی که حول مقدار 20mm با دامنه‌ی 10mm با دوره تناوب ۲۰ ثانیه نوسان می‌کند. (تصویر ۱۱)

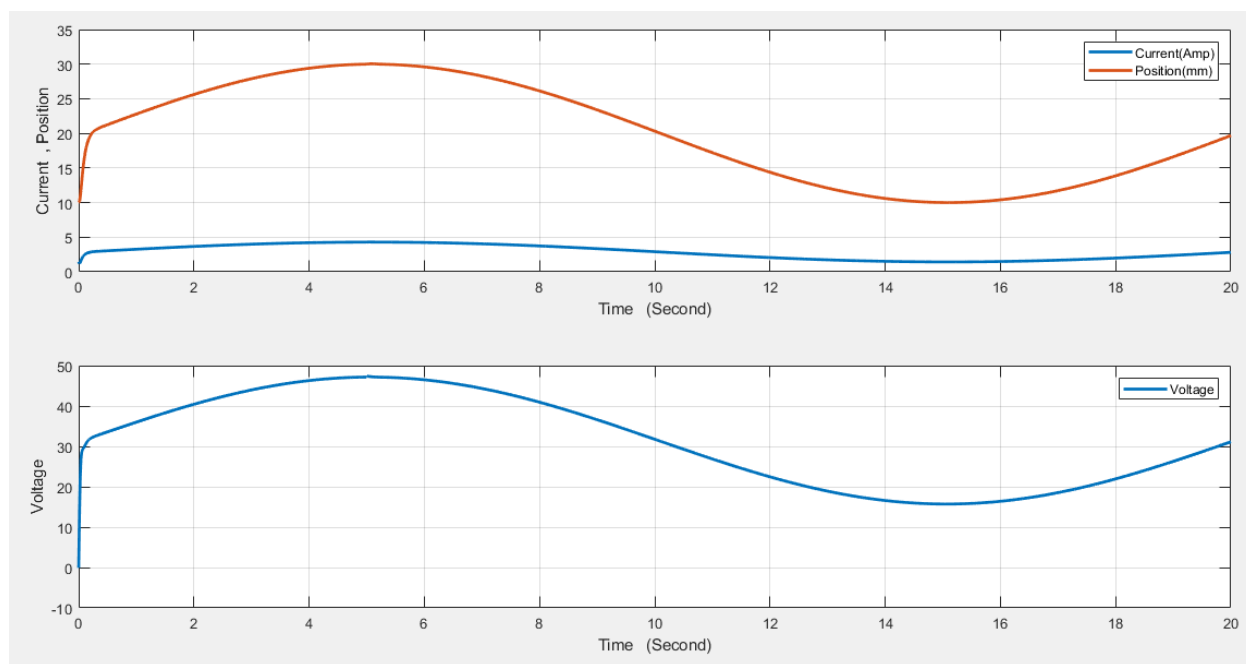


تصویر ۱۱- مقادیر بلوک سیگنال سینوسی و پله

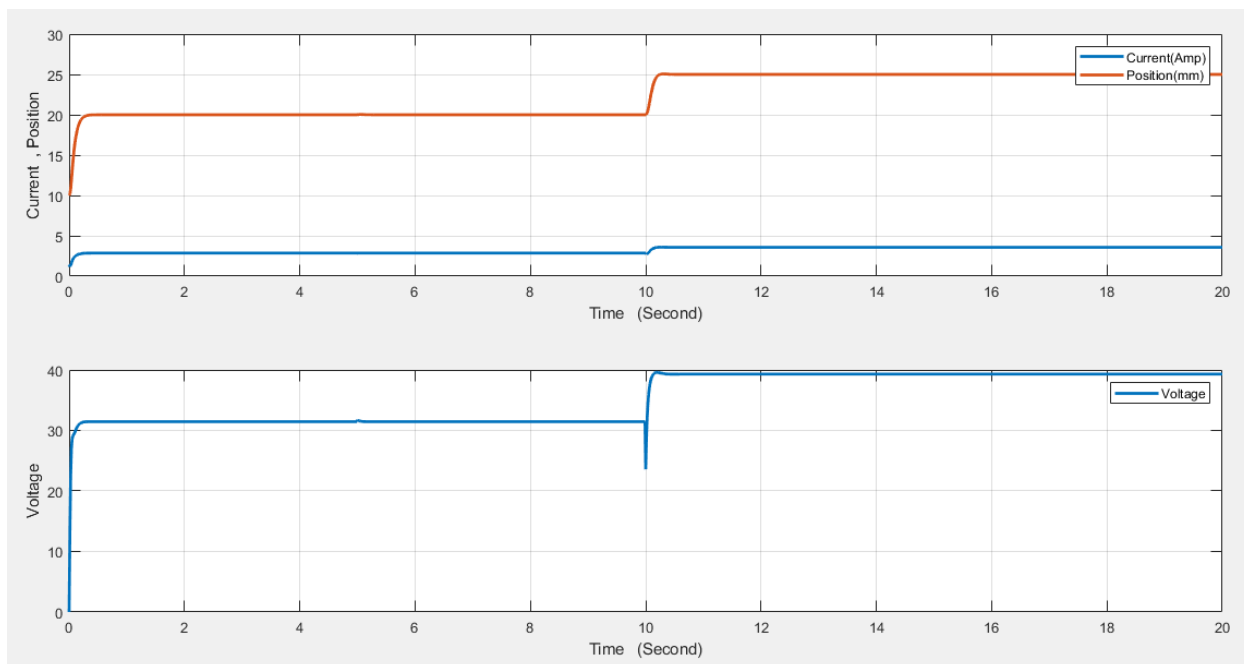


تصویر ۱۲ - تغییر ورودی

خروجی سیستم به ازای ورودی سینوسی و پله در نمودارهای ۸ و ۹ آورده شده است .



نمودار ۸ - خروجی سیستم به ازای ورودی سینوسی



نمودار ۹- خروجی به ازای ورودی پله ای

پیاده سازی تنظیم مکانیزم عصبی

شبکه عصبی مصنوعی

هدف شبکه‌های عصبی تشخیص الگوها و دسته‌بندی اطلاعات جدید بر پایه دانش قبلی خود است. یادگیری آنی الگوها در قالب بردارهای عددی انجام می‌شود. به عبارتی، تمامی داده‌های دنیای واقعی نظیر تصاویر، صوت و متن باید به بردارهای عددی تبدیل شوند و به عنوان ورودی، در اختیار شبکه عصبی قرار می‌گیرد تا مدل هوش مصنوعی بتواند آن‌ها را درک کند.

در حالت کلی، کاربردهای شبکه عصبی را می‌توان به سه گروه «دسته‌بندی» (Classify) داده‌ها، «خوشه‌بندی» (Clustering) داده‌ها و مسائل «رگرسیون» (Regression) تقسیم‌بندی کرد. در این بخش مسئله، تخمین یک تابع غیر خطی است که از نوع رگرسیون به شمار می‌رود.

پیاده سازی در محیط MATLAB

یک فایل جدید ایجاد و با نام TrainNN ذخیره می کنیم. در ابتدا پارامترهای فیزیکی سیستم وارد می شود.

```
L = 0.4125 ; % Henry  
RI = 10 ; % ohm  
Rs = 1 ; % ohm  
Kf = 32654e-9 ; % N*m^2/Amp^2  
g = 9810e-3 ; % m/sec^2 ;  
m = 0.068 ; % kg
```

اولین قدم پیاده سازی شبکه عصبی تهیه دیتاست شامل مجموعه ای از ورودی ها و خروجی ها است. پس در اولین مرحله ورودی مورد نظر ایجاد می شود. ورودی یا رفرنس سیستم XSS است، بازه مورد نظر از ۵ تا ۵۰ میلی متر است. در این بازه به کمک تابع linspace() یک بردار ورودی با ۱۰۰۰ نمونه ایجاد می کنیم.

```
N = 1000 ;  
Xss = linspace(5, 50, N)/1000 ; %divided to 1000 to convert mm to m
```

در مرحله بعد یک بردار به طول ۱۰۰۰ با مقدار صفر به اسم K تعریف کرده تا در ادامه مقادیر بهره حالت متناظر با هر ورودی در آن ذخیره شود.

```
K = zeros(N, 3) ;
```

از آنجایی که نمایش فضای حالت سیستم یا تابع تبدیل آن به پارامتر XSS وابسته است ، با تعریف یک حلقه به ازای تمام مقادیر XSS فضای حالت سیستم مجدد محاسبه می شود و یک به یک بهره ی فیدبک حالت بهینه متناظر با هر سیستم در بردار K جایگذاری می شود.

```
for i = 1:N  
lss = Xss(i) * sqrt(m*g/Kf) ;  
Vss = (RI + Rs) * lss ;  
  
a = 2*Kf/m * lss/Xss(i)^2 ;
```

```

b = 2*Kf/m * lss^2/Xss(i)^3 ;
A = [-(Rl + Rs)/L  0  0
      0            0  1
      -a           b  0];
B = [1/L  0  0]';
n = size(A , 1) ;
Q = eye(n) ;
R = 1 ;
N = 0 ;
K(i, :) = lqr(A , B , Q , R , N) ;
end

```

ماتریس K ، یک ماتریس به ابعاد 1000×3 است که هر سطر آن مربوط به یک Xss خاص است . برای نمونه ۵ سطر اول K در زیر آورده شده است:

```
>> K(1:5, :)
```

```
ans =
```

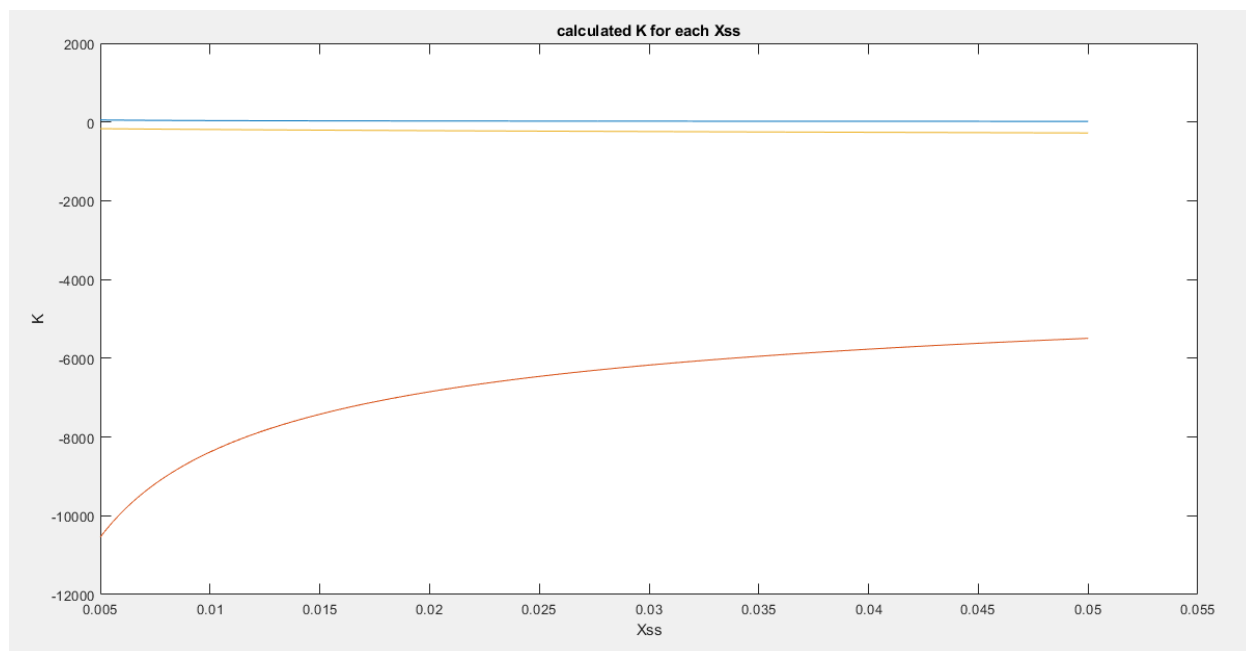
```
1.0e+04 *
```

```

0.0052 -1.0544 -0.0168
0.0051 -1.0511 -0.0169
0.0051 -1.0478 -0.0169
0.0051 -1.0446 -0.0169
0.0051 -1.0414 -0.0169

```

همچنین با رسم نمودار K بر حسب Xss تغییرات K را نشان داد.



نمودار ۱۰ - تغییرات k بر حسب X_{ss}

تا اینجا مشخص شد که نیاز است یک تابع غیر خطی را به وسیله یک شبکه عصبی تقریب زد.

یک شبکه عصبی با ۵ لایه پنهان (Hidden layer) به فرمت feedforward ایجاد می کنیم.

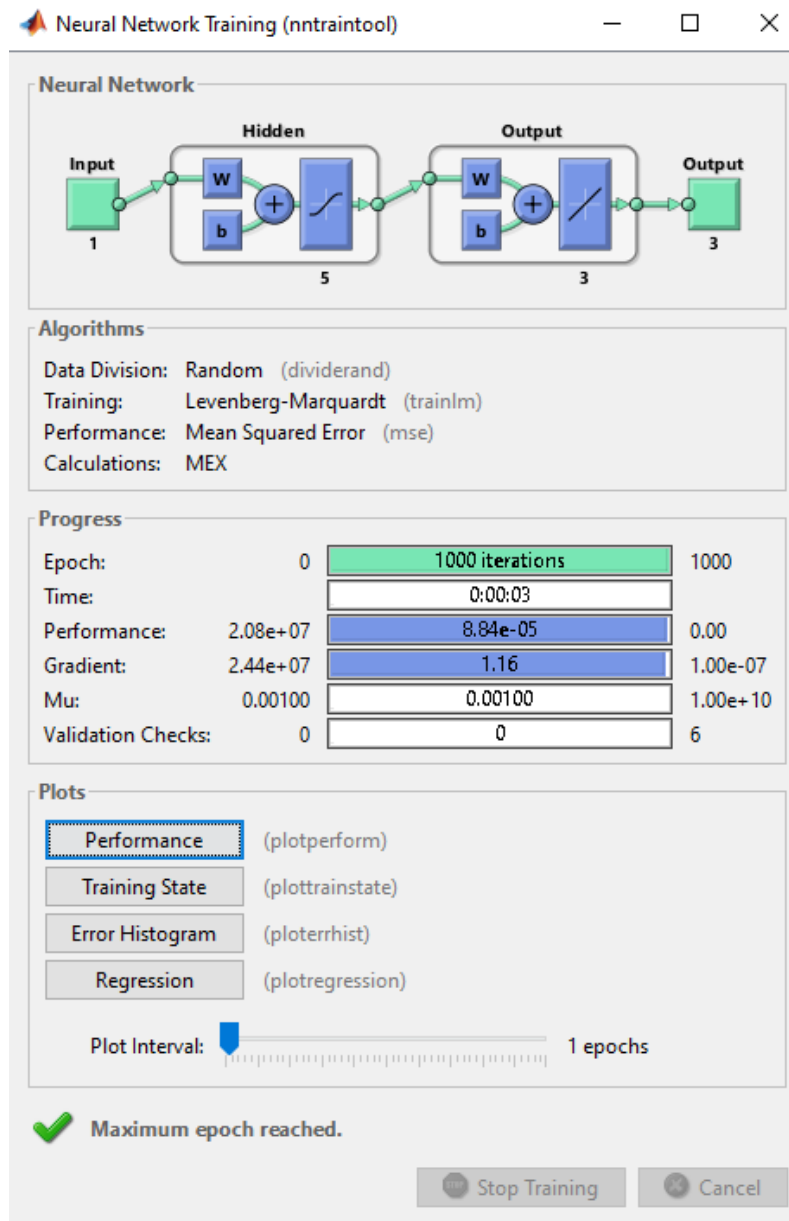
```
net = feedforwardnet(5);
```

سپس در ادامه به وسیله تابع $\text{train}(\text{net}, X, T)$ که در آن پارامتر اول شبکه عصبی تعریف

شده ، پارامتر دوم و سوم به ترتیب ورودی و خروجی مدنظر هستند.

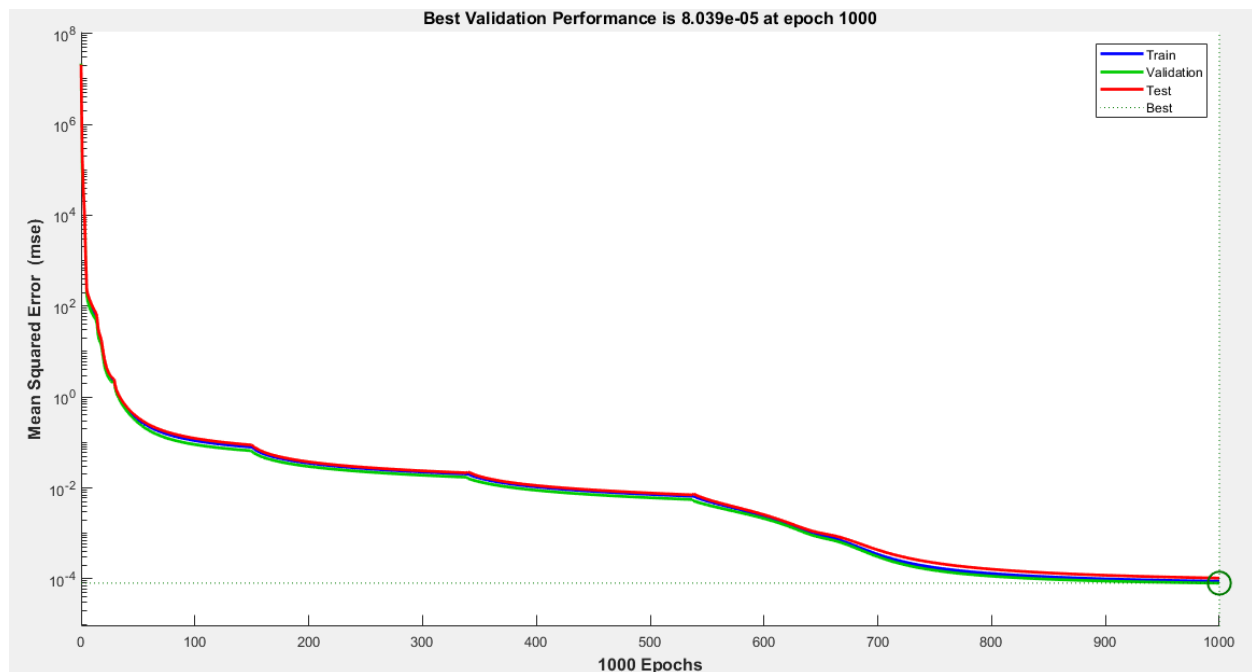
```
net = train(net, Xss, K');
```

توجه شود که نیاز است K ترانهادده شود تا ساختار ابعاد شبکه عصبی با مشکل مواجه نشود.



تصویر ۱۳- فرآیند آموزش شبکه عصبی

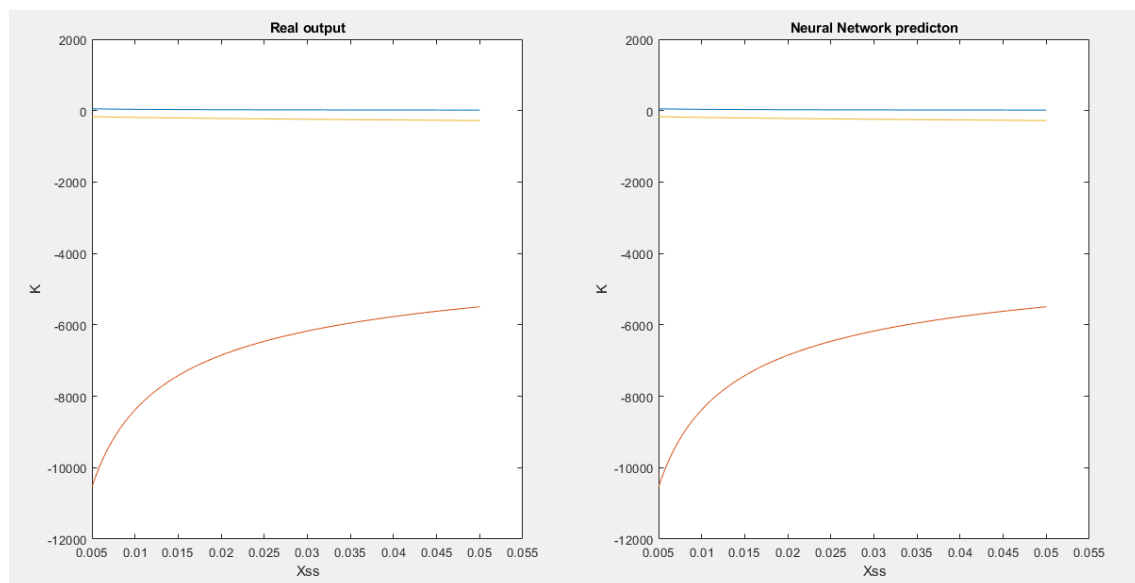
از قسمت performance می‌توان نمودار عملکرد شبکه عصبی را مشاهده کرد. این نمودار به وسیله معیار خطای Mean Squared Error یا به اختصار MSE عملکرد سیستم را در هر پیمایش نمایش می‌دهد.



نمودار ۱۱- نمودار عملکرد شبکه عصبی

با اعمال ورودی Xss به شبکه ، ماتریس Knet که خروجی شبکه است را بدست آورده و با رسم نمودار با K واقعی مقایسه می کنیم.

Knet = net(Xss);

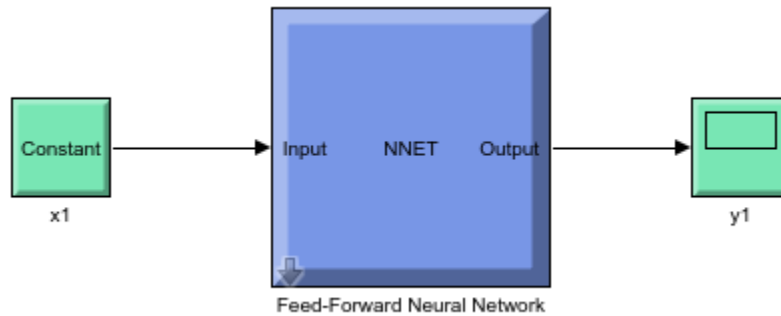


نمودار ۱۲- مقایسه خروجی شبکه عصبی با مقدار واقعی

در ادامه نیاز است که تابع net را به صورت بلوک Simulink درآورده و از آن در سیستم اصلی به جای فیدبک حالت استفاده کنیم. برای این منظور از دستور `gensim()` استفاده می‌کنیم.

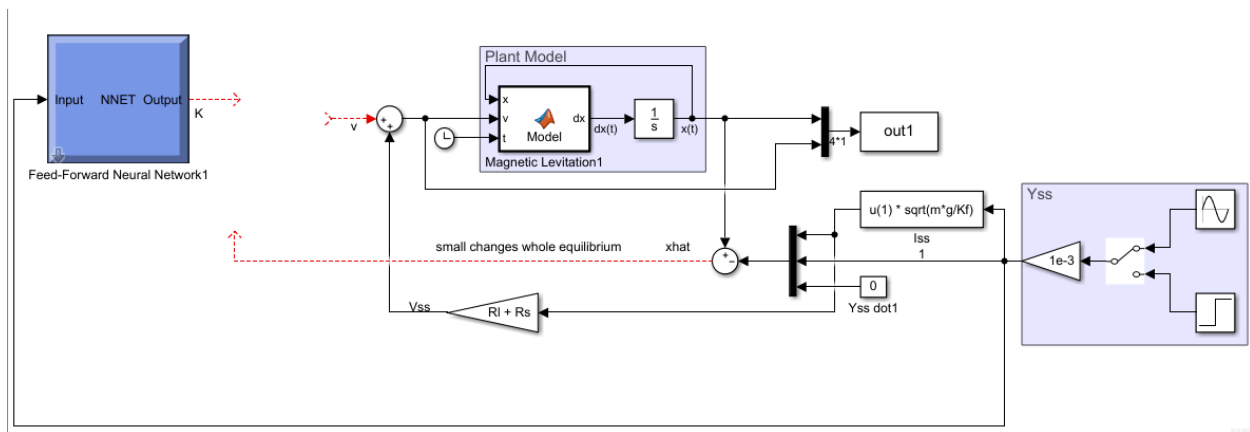
`gensim(net)`

با فراخوانی این دستور یک پنجره Simulink باز شده که شامل بلوک شبکه عصبی است.



تصویر ۱۴ - بلوک تولید شده شبکه عصبی

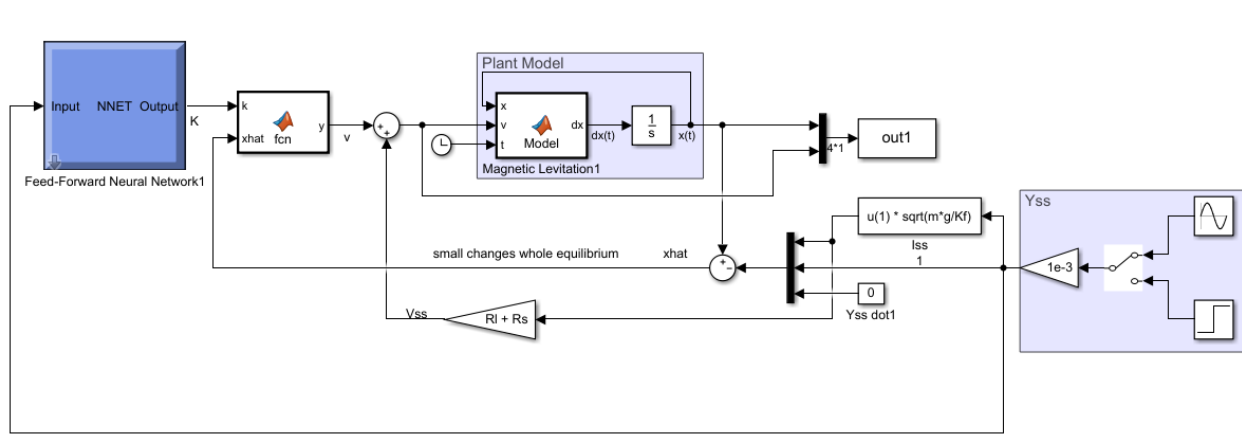
قسمت اصلی را کپی کرده و به فایل مدل منتقل می‌کنیم.



سیستم ۴ - اضافه کردن شبکه عصبی به سیستم

ورودی شبکه عصبی مقدار Y_{ss} است و خروجی آن K متناظر با آن را تولید می‌کند.

در ادامه بردار K با ضریب منفی در \hat{x} (تغییرات کوچک حول نقطه تعادل) ضرب شده تا سیستم حلقه بسته تشکیل شود. با اضافه کردن یک بلوک MATLAB function این عملیات را انجام می‌دهیم.



سیستم ۵- تکمیل سیستم حلقه بسته

بلوک MATLAB function به صورت زیر برنامه نویسی می‌شود.

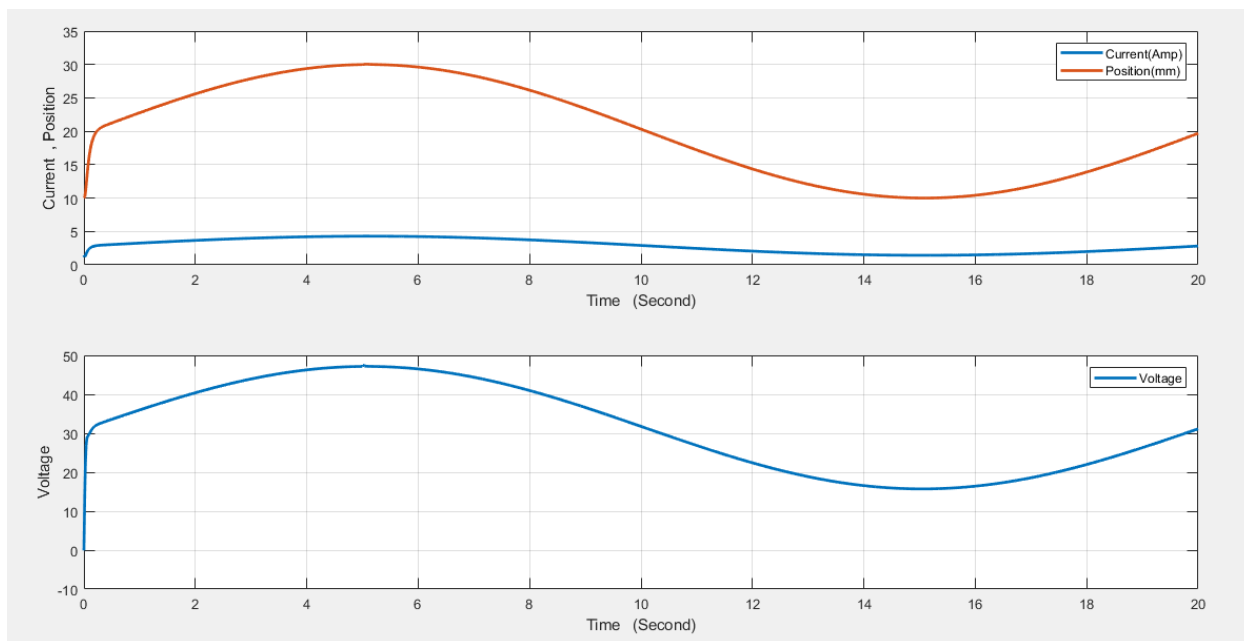
`function y = fcx(k, xhat)`

`y = -k' * xhat;`

*برای اینکه ضرب بردارها به درستی انجام شود نیاز است که بردار K ترانهاده شود.

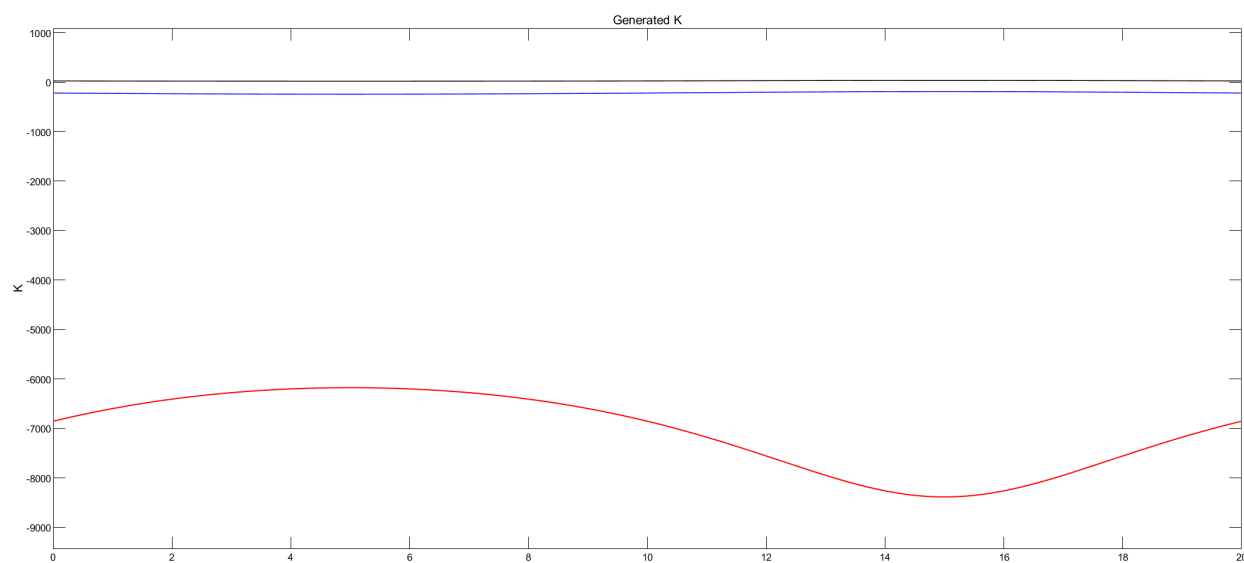
خروجی این بلوک تغییرات ولتاژ حول V_{ss} است، پس خروجی تابع با V_{ss} جمع شده و به سیستم گوی اعمال می‌شود.

با اعمال ورودی سینوسی با دامنه ۱۰، مقدار بهره DC ۲۰ و دوره تناوب ۲۰ ثانیه به سیستم خروجی زیر بدست می‌آید (نمودار ۱۳):



نمودار ۱۳- خروجی سیستم کنترل عصبی به ازای ورودی سینوسی

مشاهده می شود که با تغییر موقعیت ورودی، سیستم به خوبی عمل کرده و نتیجه مطلوب حاصل شده است. همچنین می توان نمودار تغییرات K را نیز مشاهده کرد:



نمودار ۱۴- تغییرات K

مراجع

[1] R. C. Dorf and R. H. Bishop, *Modern Control Systems*, Addison Wesley, Menlo Park, CA, 1998, 8th Ed., pp. 108.

[۲] حمیدرضا تقی راد، "مقدمه ای بر کنترل مدرن"، دانشگاه صنعتی خواجه نصیرالدین طوسی، ۱۳۹۶

پیوست

رسم نمودار شبیه سازی

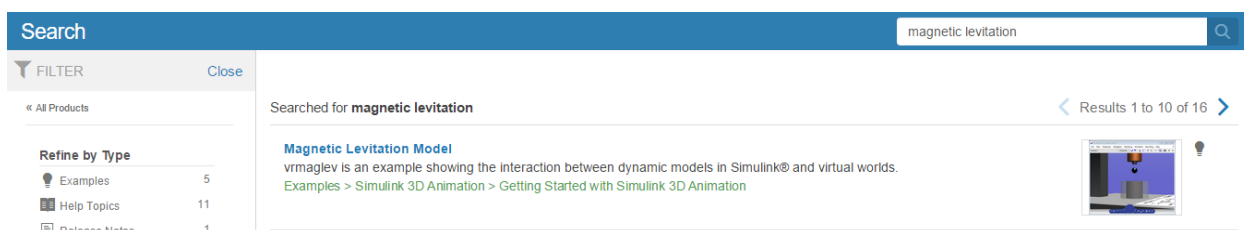
برای تولید نمودار خروجی فایل شبیه سازی از کد زیر استفاده شده است . از بخش `model` properties قسمت `callbacks` در قسمت `stop fcn` کد زیر را وارد می کنیم:

```
figure(1) ;  
subplot(211) ;  
out.signals.values(:, 2) = 1e3*out.signals.values(:, 2) ;  
plot(out.time , out.signals.values(:, [1,2]) , 'LineWidth' , 2) ;  
grid on  
xlabel('Time (Second)') ;  
ylabel('Current , Position') ;  
legend('Current(Amp)','Position(mm)') ;
```

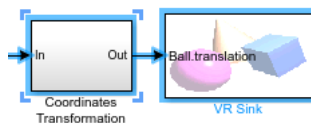
```
subplot(212) ;  
plot(out.time , out.signals.values(:, 4) , 'LineWidth' , 2) ;  
grid on  
xlabel('Time (Second)') ;  
ylabel('Voltage') ;  
legend('Voltage') ;
```

شبیه سازی سه بعدی

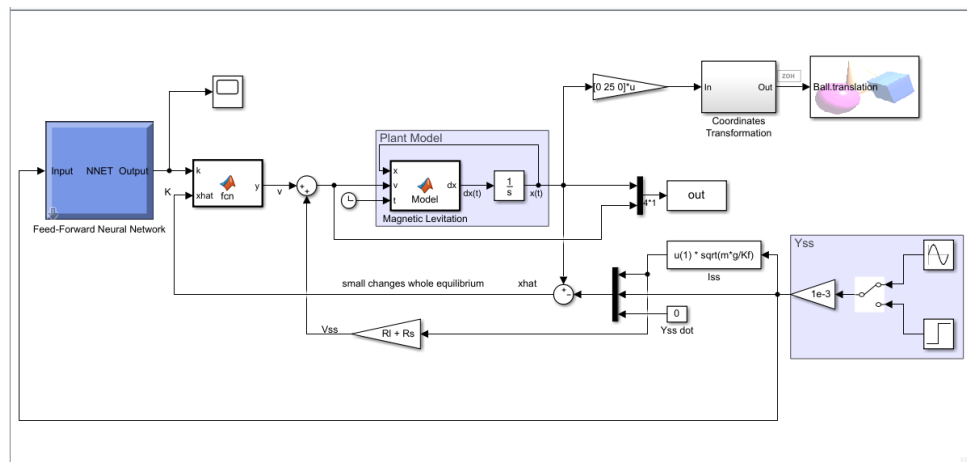
با جستجوی `magnetic levitation` در قسمت `Search` ، تعدادی شبیه سازی سیستم تعلیق نمایش داده می شود با باز کردن پروژه زیر، می توان از امکانات شبیه سازی سه بعدی موجود در این فایل استفاده کرد



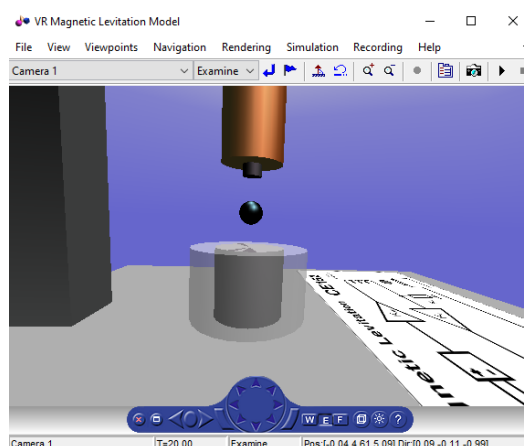
فایل پروژه را در سیمولینک باز کرده و قسمت زیر را کپی کرده و در فایل خود قرار می دهیم.



ورودی موقعیت گوی است که برابر با دومین مقدار خروجی بردار حالت است، از آنجایی که محدوده کاری این سیستم با سیستم ما متفاوت است برای نمایش بهتر خروجی سیستم را با یک بهره به نمایشگر گرافیکی وارد می‌کنیم.



این نمایشگر سه بعدی به محض اجرای شبیه سازی باز شده و خروجی سیستم را به صورت سه بعدی نشان می‌دهد.



با زدن start شبیه سازی شروع شده و موقعیت گوی تغییر می‌کند.

Abstract

Magnetic levitation systems are electromechanical systems. Due to their nature and structure, they are considered to be nonlinear and very unstable systems. The linearized transfer function of a magnetic levitation system is very dependent on the system's initial conditions. Therefore, a three-step approach to controlling a magnetic levitation system is proposed:

1. Use a linear PID controller to control the system.
2. Stabilize the system using state feedback and LQR controller.
3. Design a mechanism using artificial neural networks to control the system in different initial conditions.

Using neural network as a regulation mechanism will make the system capable of responding to different initial values. The simulation and modeling of the system is performed in MATLAB and Simulink environments.

Keywords: maglev, PID, LQR, state feedback, neural network



دانشگاه صنعتی شیراز
SHIRAZ UNIVERSITY
OF TECHNOLOGY

Department of Electrical Engineering

BSc project in:

Control engineering

**Control of magnetic levitation system using neural network
regulation mechanism**

By:

Shahed Momenzadeh

Supervisor:

Dr. Jafar Zarei

November 2023