

## آزمایش هفتم شناسایی به کمک شبکه عصبی

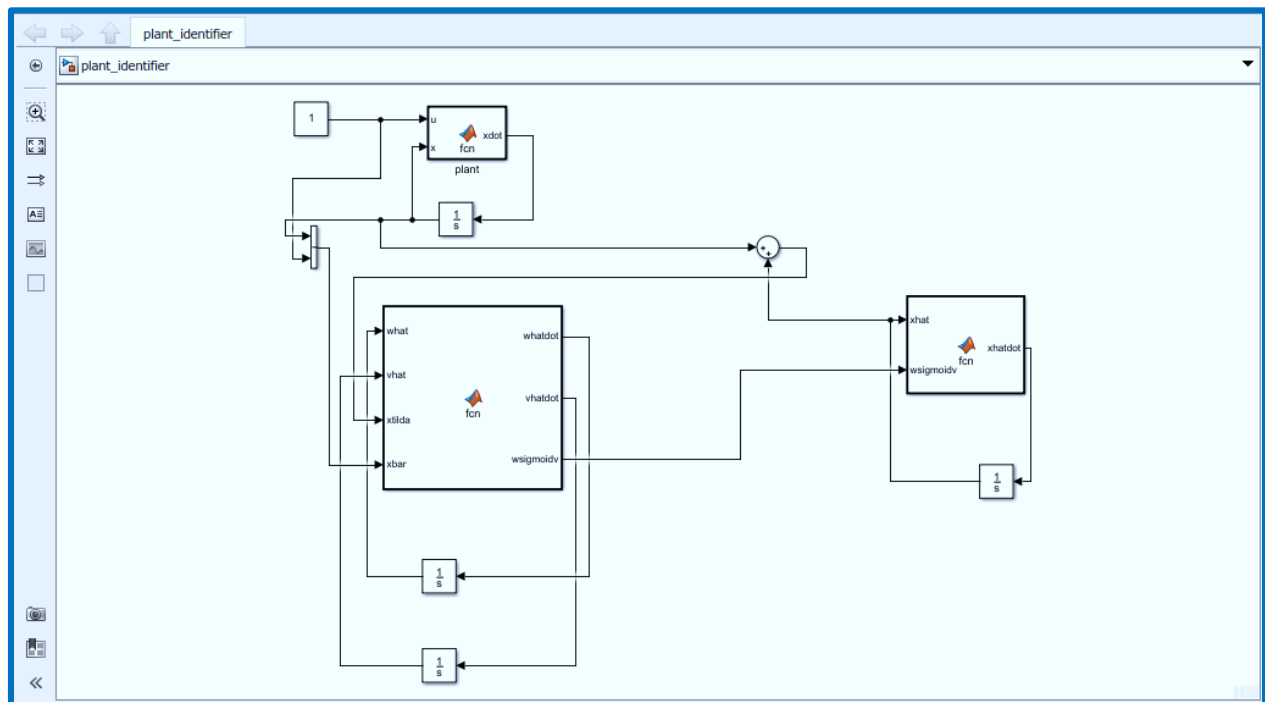
اعضای گروه:

ملیکا صالحیان ، محدثه قاسم مهرابی، مریم عیدری

❖ مراحل پیاده سازی شناساگر پیوسته

با استفاده از تابع تبدیل مشخص شده در گزارشکار ابتدا فضای حالت را بدست آورده و شناساگر پیوسته را به شرح زیر با استفاده از سه بلوک تابع در متلب پیاده سازی می کنیم:

$$H(s) = \frac{40}{0.2 * s^2 + s}$$

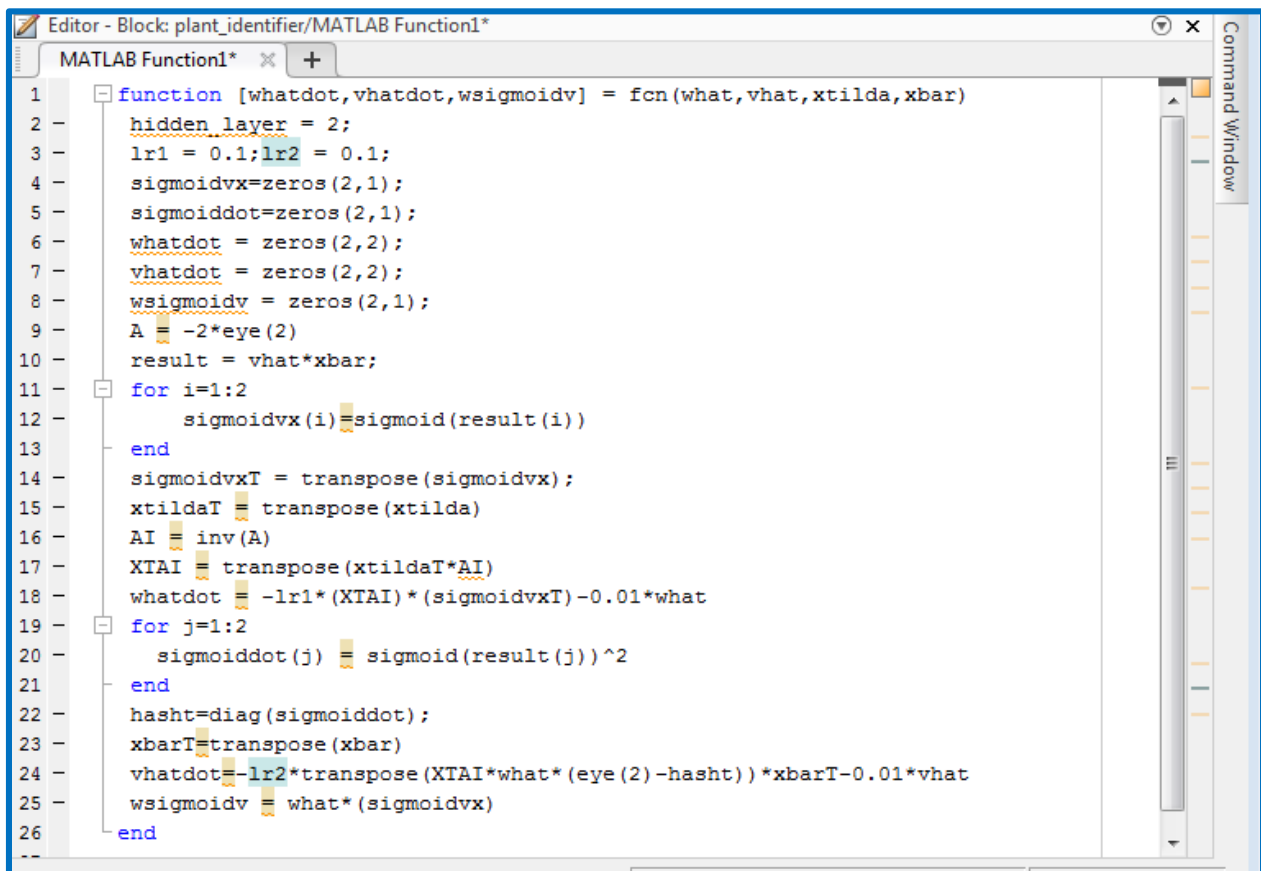


# آزمایشگاه هوش مناسباتی

که دستورات موجود در هر یک از بلوک های توابع به شرح زیر می باشد:

Plant:

```
1 function xdot = fcn(u,x)
2 - xdot=zeros(2,1)
3 - A = [-50,0;1,0];
4 - B = [1;0];
5 - xdot = A*x+B*u;
6 end
```



```
Editor - Block: plant_identifier/MATLAB Function1*
MATLAB Function1* x +
1 function [whatdot,vhatdot,wsigmoidv] = fcn(what,vhat,xtilda,xbar)
2 - hidden_layer = 2;
3 - lr1 = 0.1;lr2 = 0.1;
4 - sigmoidvx=zeros(2,1);
5 - sigmoiddot=zeros(2,1);
6 - whatdot = zeros(2,2);
7 - vhatdot = zeros(2,2);
8 - wsigmoidv = zeros(2,1);
9 - A = -2*eye(2)
10 - result = vhat*xbar;
11 - for i=1:2
12 -     sigmoidvx(i)=sigmoid(result(i))
13 - end
14 - sigmoidvxT = transpose(sigmoidvx);
15 - xtildaT = transpose(xtilda)
16 - AI = inv(A)
17 - XTAI = transpose(xtildaT*AI)
18 - whatdot = -lr1*(XTAI)*(sigmoidvxT)-0.01*what
19 - for j=1:2
20 -     sigmoiddot(j) = sigmoid(result(j))^2
21 - end
22 - hasht=diag(sigmoiddot);
23 - xbarT=transpose(xbar)
24 - vhatdot=-lr2*transpose(XTAI*what*(eye(2)-hasht))*xbarT-0.01*vhat
25 - wsigmoidv = what*(sigmoidvx)
26 end
```

```
1 function xhatdot = fcn(xhat,wsigmoidv)
2 - A=-2*eye(2)
3 - xhatdot=A*xhat+wsigmoidv ;
4 end
```

بنابراین حالت های سیستم با استفاده از شناساگر فوق پیش بینی می شود.

# آزمایشگاه هوش مناسباتی

در ادامه مراحل طراحی شناساگر گسسته در تمرین دوم مورد بررسی قرار گرفته است.

## تمرین

۱- فرض کنید در بازوی ربات با مدل زیر (M) شناخته شده نیست، این سیستم را شناسایی کنید. می‌خواهیم انتهای بازوی ربات مسیر  $\sin 10t$  را دنبال کند.

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = M^{-1}(x)[u - C(x_1, x_2)x_2 - Dx_2 - g(x_1)]$$

که  $x_1 = [q_1 \ q_2]$  موقعیت انتهای هر لینک،  $\dot{x}_1 = x_2$  سرعت انتهای هر لینک،  $u$  گشتاور،  $M$  ماتریس اینرسی،  $C$  نیروهای کریولیس و گریز از مرکز،  $D$  دمپینگ و ویسکوز و  $g$  نیروی گرانش است و به صورت زیر تعریف شده است:

$$M = \begin{bmatrix} a_1 + 2a_4 \cos q_2 & a_2 + a_4 \cos q_2 \\ a_2 + a_4 \cos q_2 & a_3 \end{bmatrix}, \quad D = 0$$

$$C = a_4 \sin q_2 \begin{bmatrix} -\dot{q}_2 & -(\dot{q}_1 + \dot{q}_2) \\ \dot{q}_1 & 0 \end{bmatrix}, \quad g = \begin{bmatrix} b_1 \cos q_1 + b_2 \cos(q_1 + q_2) \\ b_2 \cos(q_1 + q_2) \end{bmatrix}$$

$$a_1 = 200.01, \quad a_2 = 23.5, \quad a_3 = 122.5, \quad a_4 = 25, \quad b_1 = 784.8, \quad b_2 = 245.25$$

## آزمایشگاه هوش مناسباتی

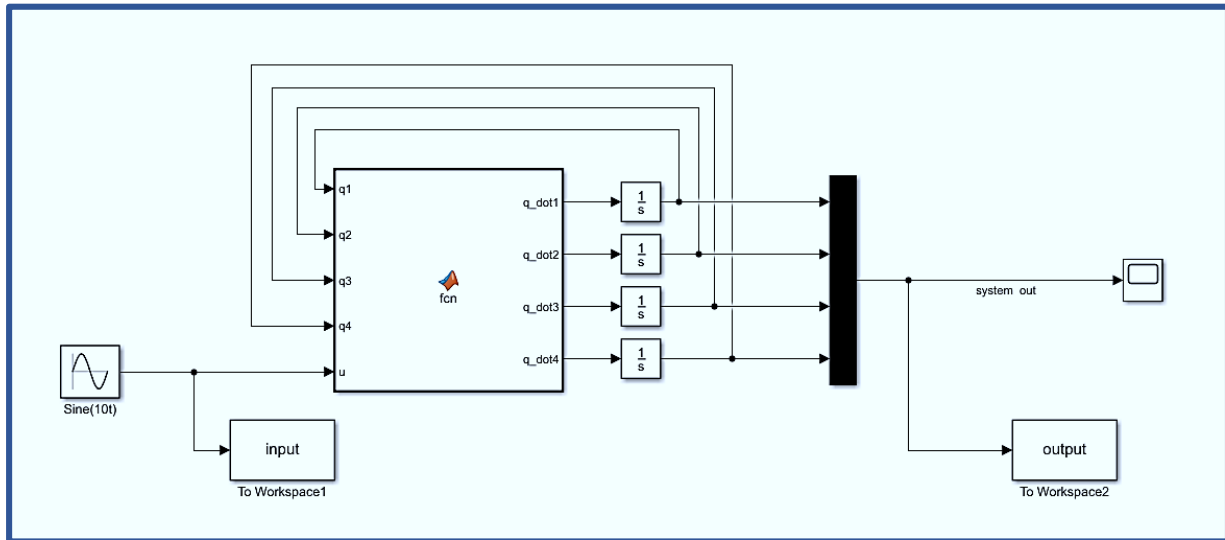
ابتدا متغیرهای سیستم را در matlab function به صورت زیر تعریف می کنیم:

```
function [q_dot1,q_dot2,q_dot3,q_dot4] = fcn(q1,q2,q3,q4,u)
a1 = 200.01;
a2 = 23.5;
a3 = 122.5;
a4 = 25;
b1 = 784.8;
b2 = 245.25;
M = [a1 + 2*a4*cos(q2) a2 + a4*cos(q2);a2 + a4*cos(q2) a3];
D = 0;
C = a4*sin(q2)*[-q4 -(q3+q4);q3 0];
g = [b1*cos(q1) + b2*cos(q1+q2);b2*cos(q1+q2)]
x2 = [q3;q4]
x_dot2 = inv(M)*(u-C*x2-D*2-g);
q_dot1 = q3;
q_dot2 = q4;
q_dot3 = x_dot2(1);
q_dot4 = x_dot2(2);
```

برای اینکه خود  $x_1$  و  $x_2$  دومتغیره هستند، تعداد متغیرهای سیستم را 4 در نظر می گیریم. در ادامه با استفاده از **nnntool** به جای طراحی بلوکی به حل مساله پرداخته شده است.

# آزمایشگاه هوش مناسباتی

سیستم را به صورت زیر در سیمولینک طراحی می کنیم:



داده های ورودی و خروجی را از طریق بلوک مربوطه به work space ارسال می کنیم.

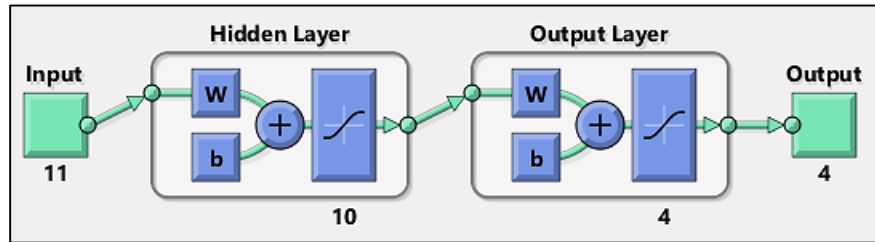
در pace works این داده ها را ذخیره و با ایجاد ورودی و خروجی های تاخیردار ورودی شبکه را می سازیم:

```
Input = input.Data;
Output = output.Data;
Input_1_delay = [0;Input(1:end-1)];
Input_2_delay = [0;0;Input(1:end-2)];
Output_1_delay = [[0 0 0 0];Output(1:end-1,:)];
Output_2_delay = [[0 0 0 0;0 0 0 0];Output(1:end-2,:)];
X = [Input, Input_1_delay, Input_2_delay, Output_1_delay, Output_2_delay]';
Output = Output';
nntool;
```

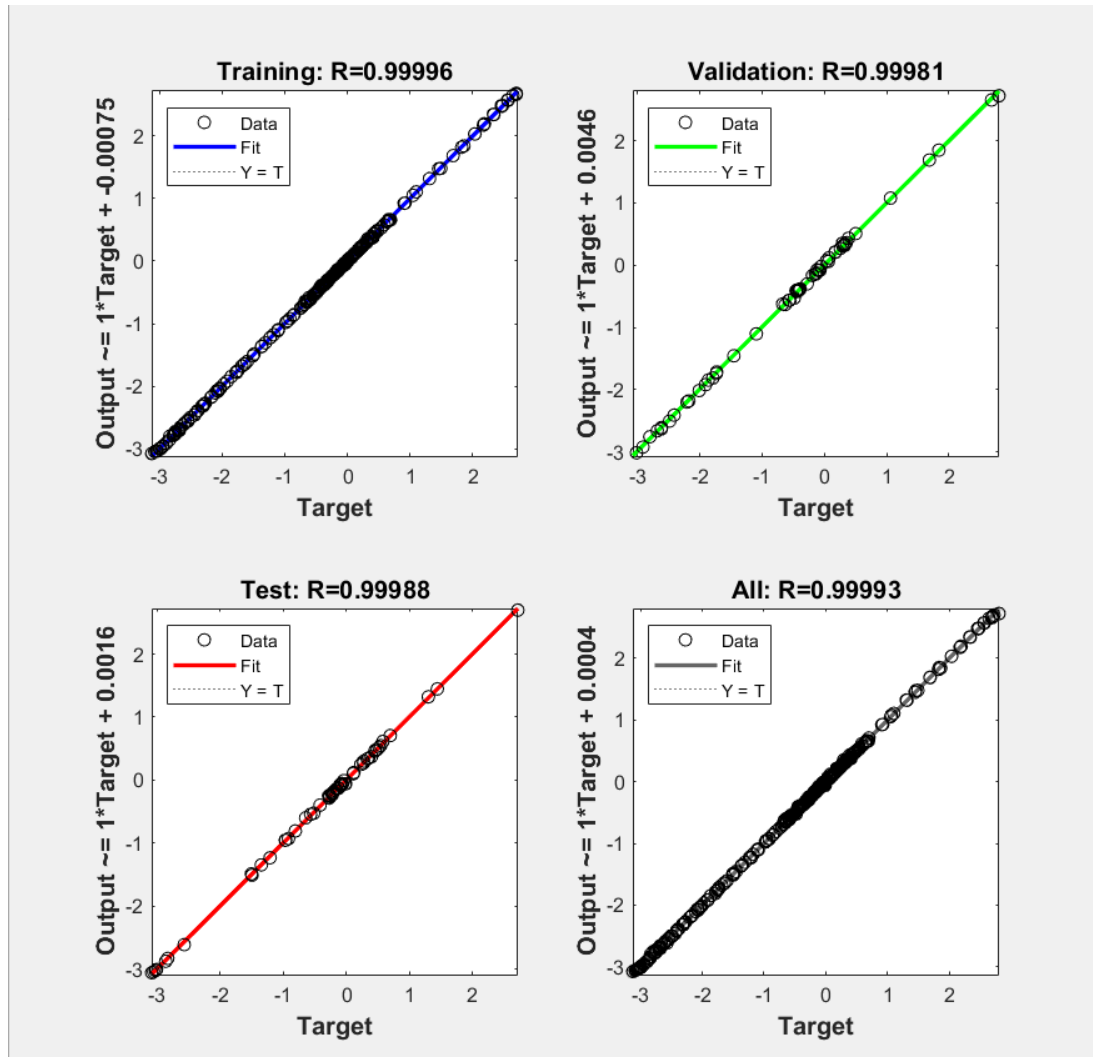
ابعاد داده های ورودی و خروجی دریافت شده از سیمولینک به ترتیب  $101 \times 1$  و  $101 \times 4$  است که در نهایت ابعاد ورودی و خروجی شبکه به ترتیب برابر  $4 \times 101$  و  $11 \times 101$  می گردد. برای ترین شبکه از nntool استفاده می کنیم.

پس از انتخاب ورودی و خروجی، شبکه را با 10 نرون برای لایه میانی ایجاد می کنیم.

# آزمایشگاه هوش محاسباتی

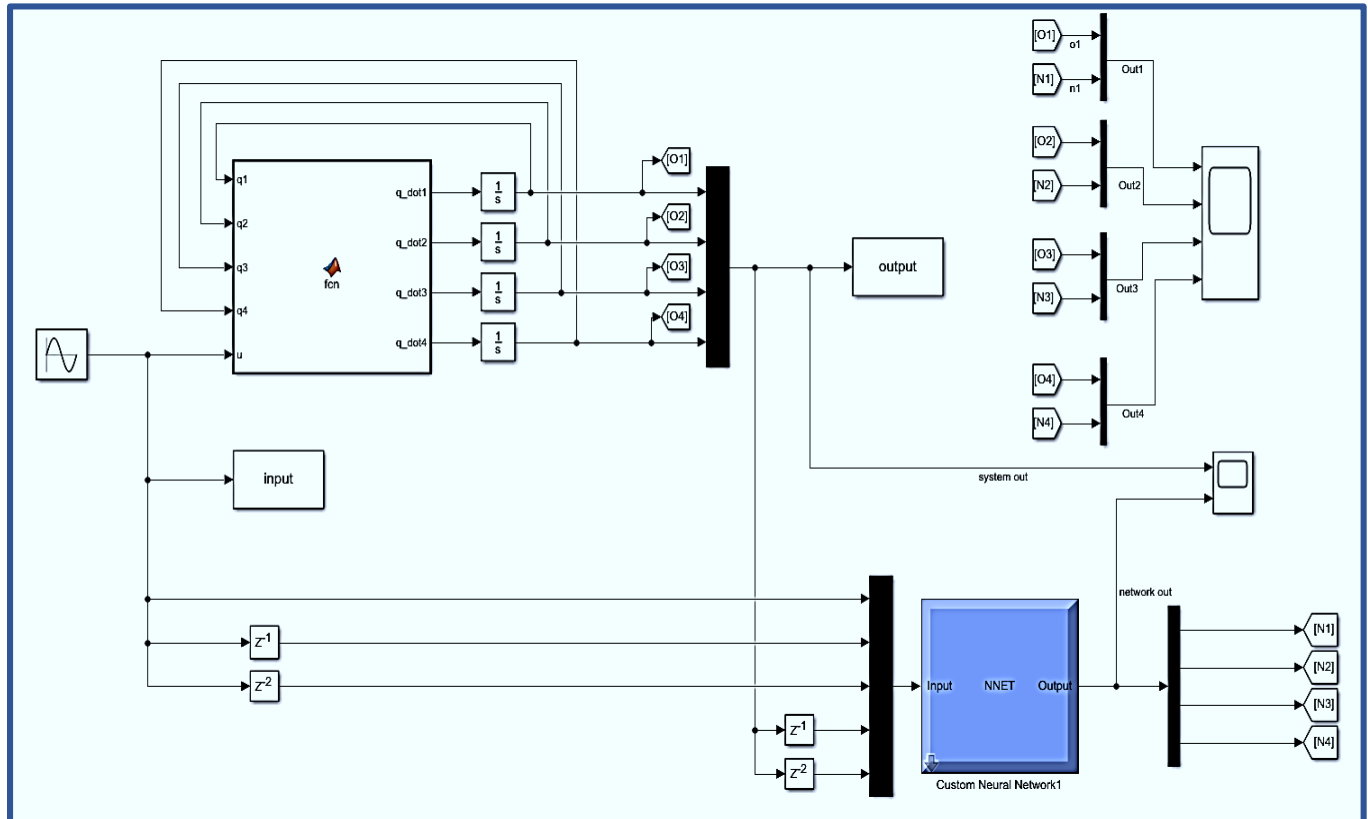


پس از ترین شبکه داریم:

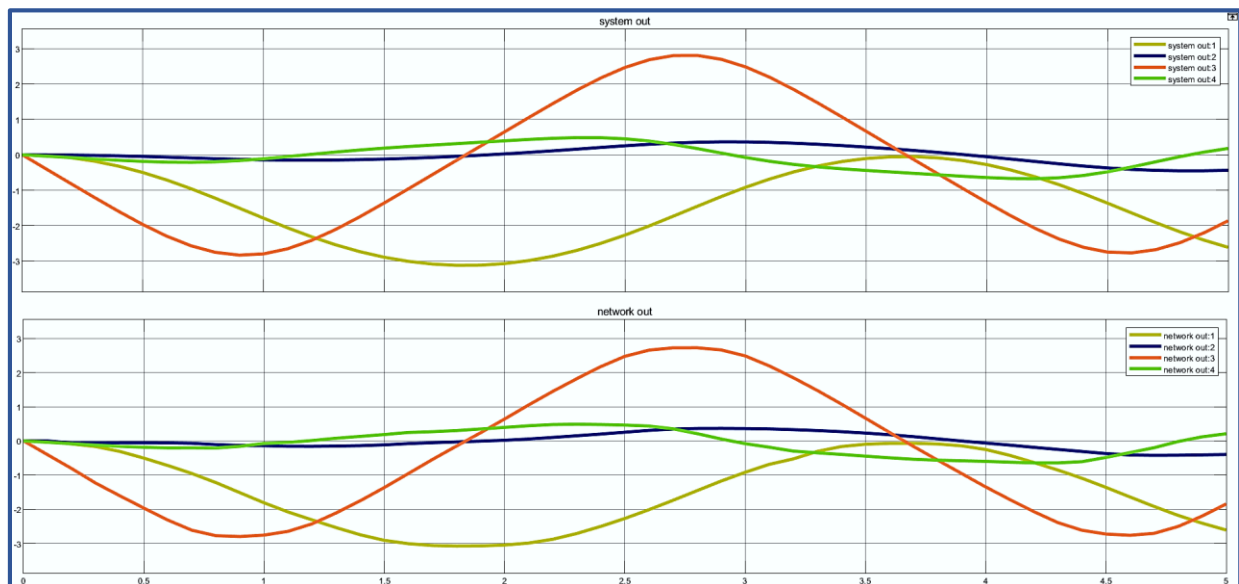


پس از ترین شبکه را export می کنیم و از طریق دستور gensim به سیمولینک ارسال می کنیم. سپس از بلوک Custom Neural Network در سیستم طراحی شده قبلی استفاده می کنیم:

# آزمایشگاه هوش محاسباتی



خروجی شبکه به صورت زیر است:



## آزمایشگاه هوش مناسباتی

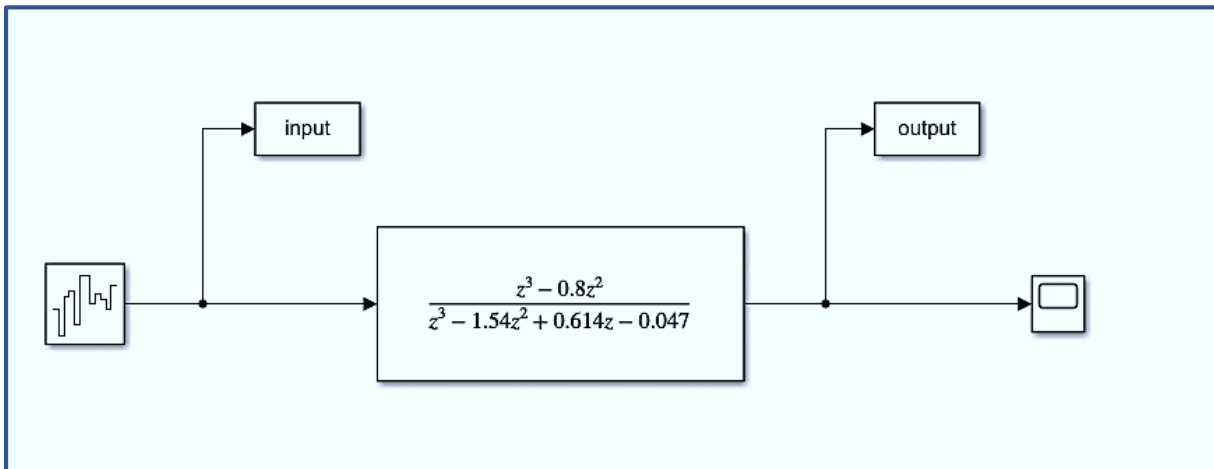
همان‌طور که مشاهده می‌شود خروجی شناسایی‌شده توسط شبکه شباهت زیادی به خروجی اصلی سیستم دارد و تقریباً یکسان‌اند.

۲- ابتدا در سیمولینک سیستمی با تابع تبدیل زیر را پیاده‌سازی کنید

$$G(z^{-1}) = \frac{(1 - 0.8z^{-1})}{(1 - 0.1z^{-1})(1 - 0.5z^{-1})(1 - 0.94z^{-1})}$$

- ابتدا چند ورودی مختلف به این سیستم اعمال کنید و داده‌های مربوطه را ذخیره کنید.

سیستم در سیمولینک طراحی شده و داده‌های ورودی و خروجی به workspace ارسال می‌شود.



- با فرض اینکه تنها داده‌ها را در اختیار دارید و از تابع تبدیل واقعی سیستم اطلاعی ندارید، بهترین سیستم مرتبه ۳ را برای این داده‌ها تخمین بزنید. از چه روشی (روش‌هایی) استفاده کردید؟ توضیح دهید. به ازای همان ورودی‌های بخش قبل، خروجی سیستمی که تخمین زده‌اید را با خروجی سیستم اصلی در یک نمودار رسم کنید.

با توجه به رابطه تابع تبدیل سیستم داریم :

$$G(z^{-1}) = \frac{(1 - 0.8z^{-1})}{(1 - 0.1z^{-1})(1 - 0.5z^{-1})(1 - 0.94z^{-1})} = \frac{Y(z^{-1})}{U(z^{-1})}$$



## آزمایشگاه هوش مناسباتی

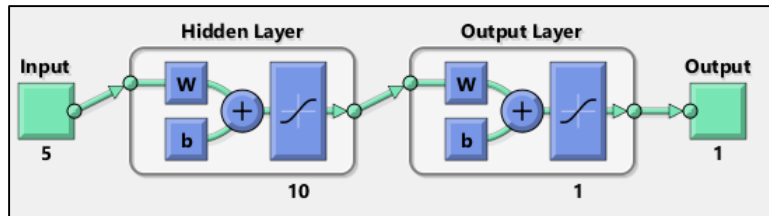
$$\rightarrow y(n) = 1.54y(n-1) - 0.614y(n-2) + 0.047y(n-3) + u(n) - 0.8u(n-1)$$

لذا برای طراحی شبکه مرتبه ۳، ورودی و تاخیر یافته آن و خروجی و ۳ تاخیر یافته آن را در نظر می‌گیریم.

```
Input = input.Data;  
Output = output.Data;  
Input_1_delay = [0;Input(1:end-1)];  
Output_1_delay = [0;Output(1:end-1)];  
Output_2_delay = [0;0;Output(1:end-2)];  
Output_3_delay = [0;0;0;Output(1:end-3)];  
X = [Input, Input_1_delay, Output_1_delay, Output_2_delay, Output_3_delay]';  
Output = Output';  
nntool;
```

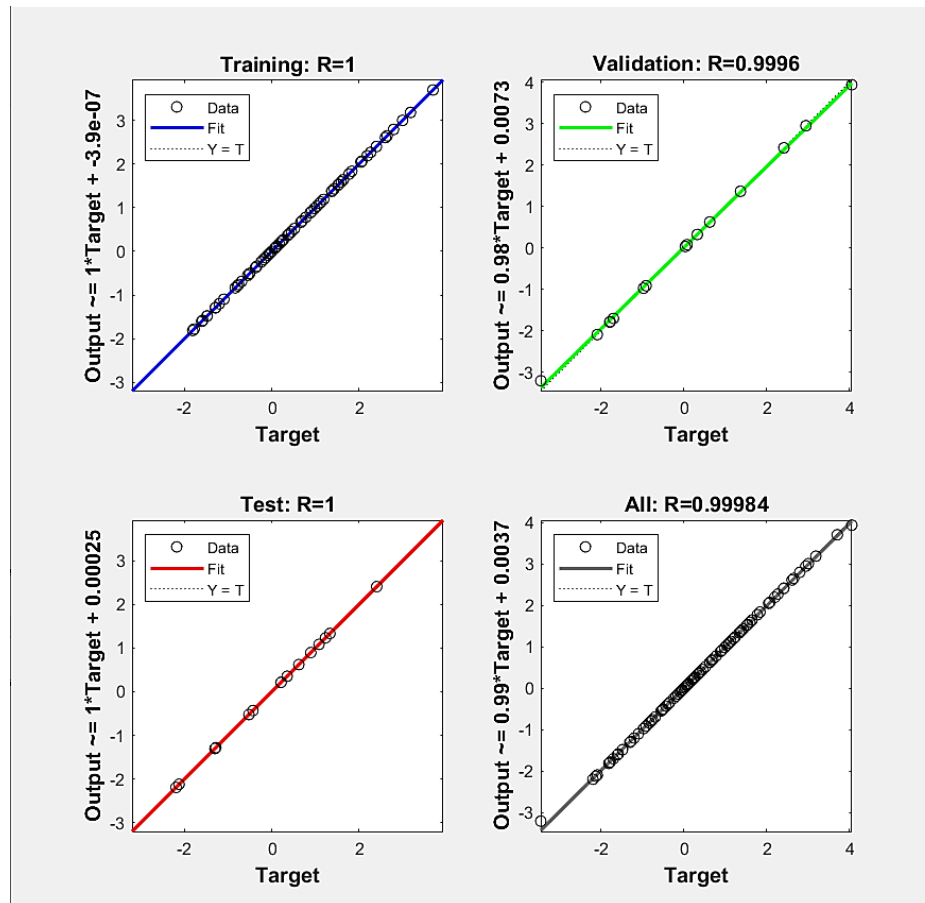
پس از ذخیره داده‌های دریافتی از سیمولینک، تاخیر یافته‌های ورودی و خروجی ایجاد و به شبکه اعمال می‌شود

و شبکه به شکل زیر طراحی می‌شود:



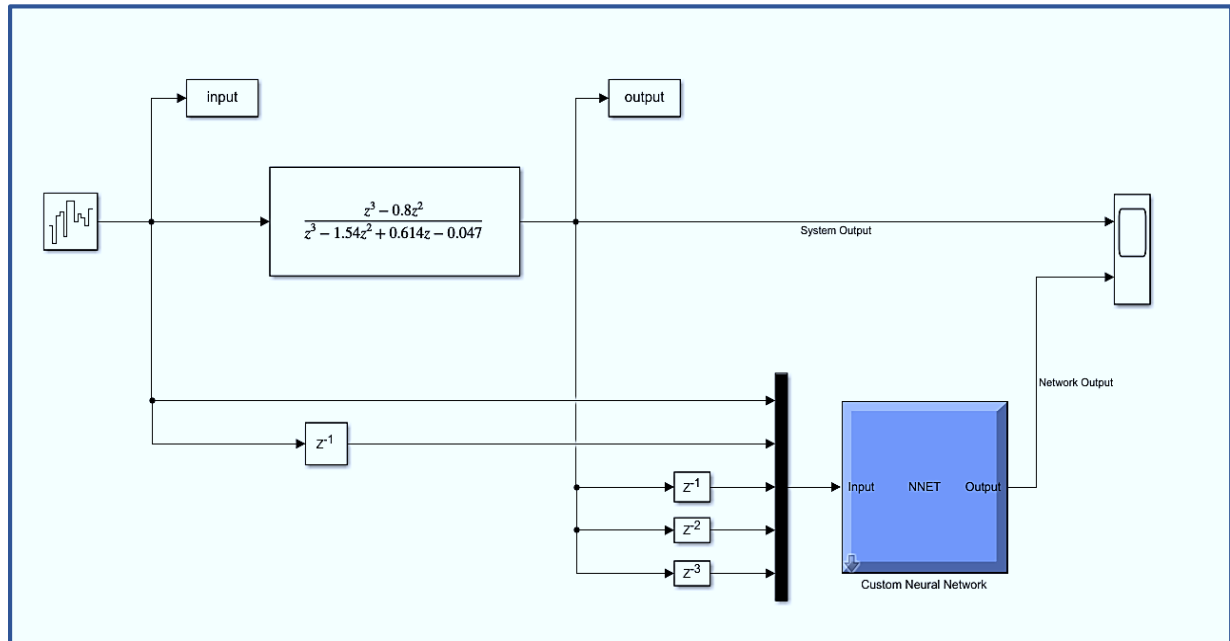
شکل ۱- شبکه مرتبه ۳ ترین شده برای شنا

# آزمایشگاه هوش مناسباتی

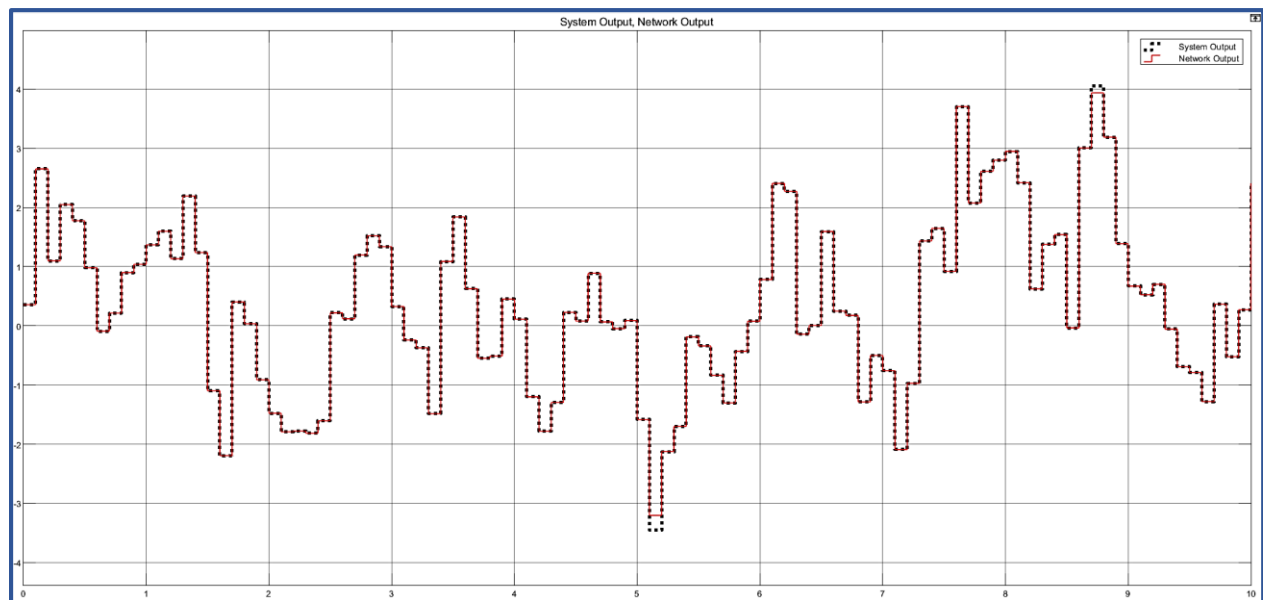


پس از ترین شبکه و export شبکه ایجاد شده، این شبکه به سیمولینک ارسال می شود و مداری مطابق شکل زیر طراحی می گردد تا خروجی سیستم و خروجی شناسایی شده توسط شبکه را مقایسه کنیم:

# آزمایشگاه هوش محاسباتی



خروجی سیستم و شبکه به صورت زیر است:



همان طور که مشاهده می شود، شبکه خروجی سیستم را به خوبی شناسایی کرده و تفاوت قابل توجهی با آن ندارد.

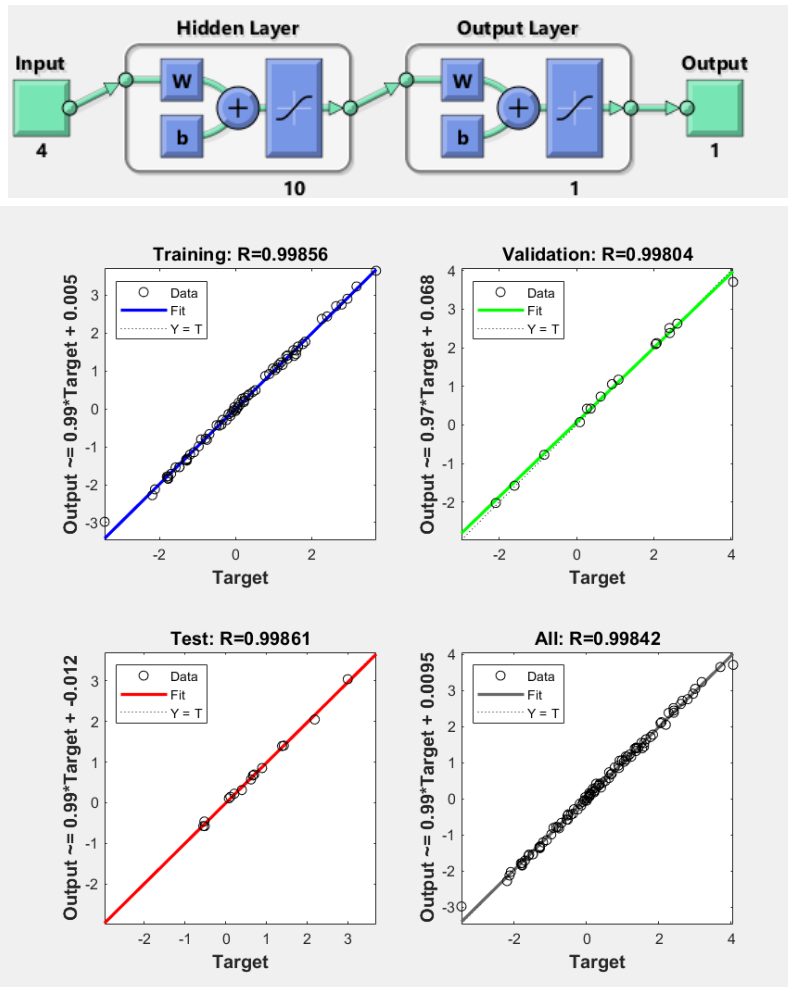
# آزمایشگاه هوش مناسباتی

- با استفاده از داده‌ها یک سیستم مرتبه ۲ تخمین بزنید و مراحل سؤال ۱ را تکرار کنید.

برای تخمین مرتبه ۲، به جای ۳ تاخیر یافته خروجی، ۲ تاخیر آن را در نظر می‌گیریم:

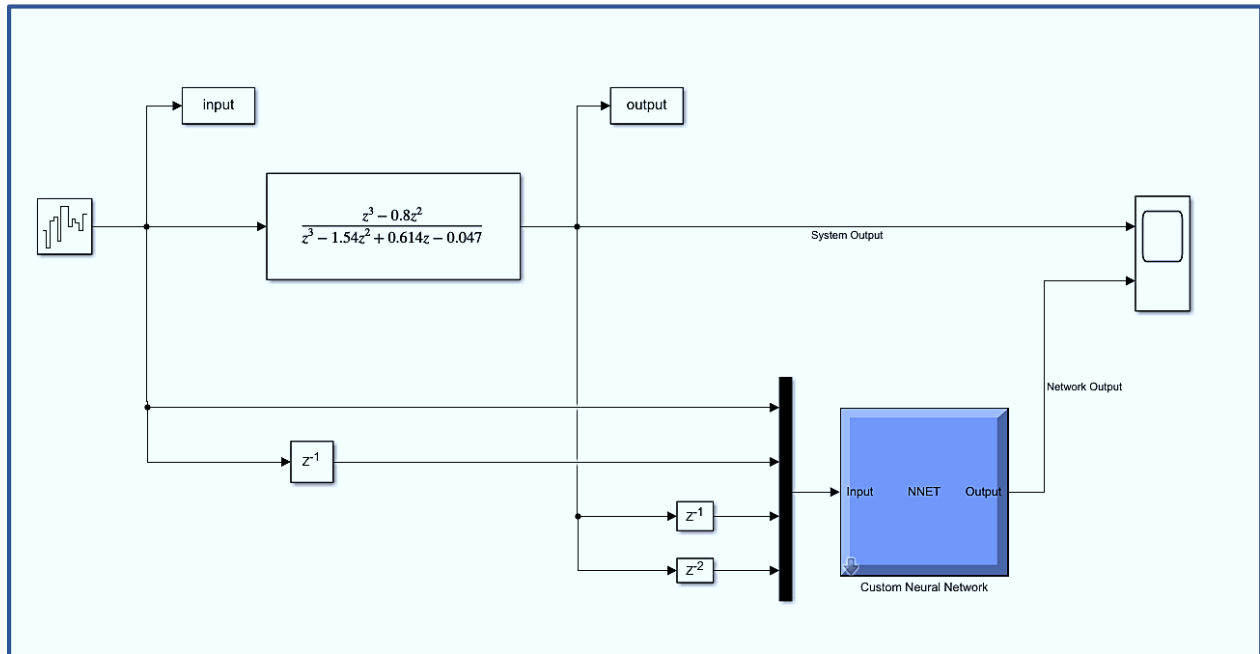
```
Input = input.Data;  
Output = output.Data;  
Input_1_delay = [0;Input(1:end-1)];  
Output_1_delay = [0;Output(1:end-1)];  
Output_2_delay = [0;0;Output(1:end-2)];  
X = [Input, Input_1_delay, Output_1_delay, Output_2_delay]';  
Output = Output';  
nntool;
```

پس از ذخیره داده‌های دریافتی از سیمولینک، تاخیر یافته‌های ورودی و خروجی ایجاد و به شبکه اعمال می‌شود و شبکه به شکل زیر طراحی می‌شود:

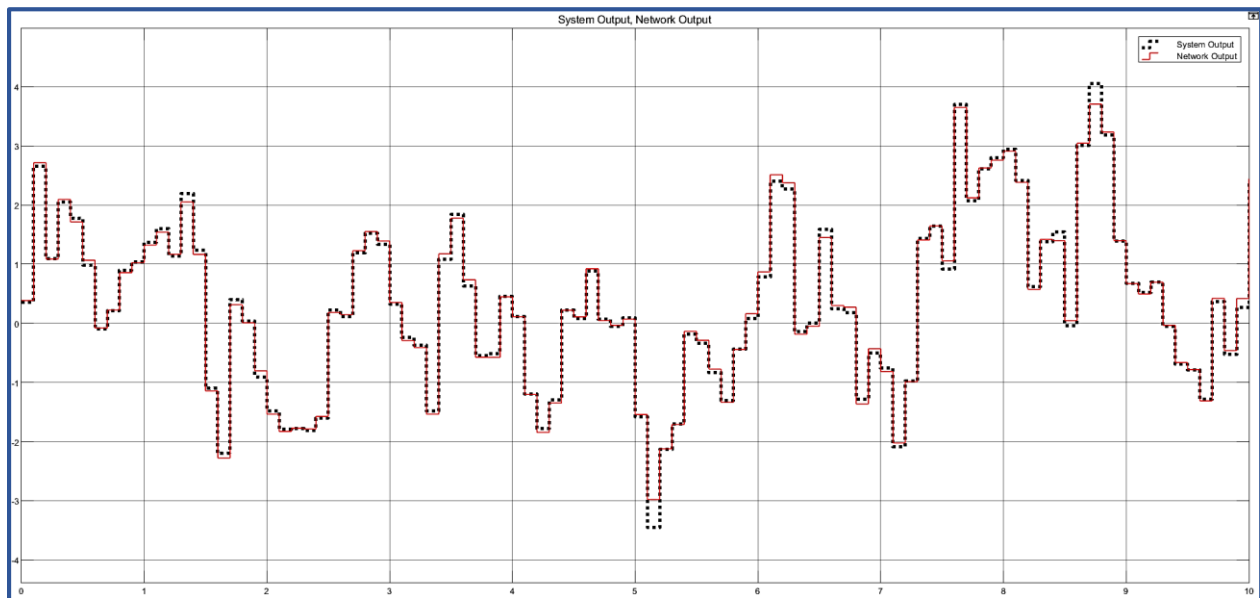


# آزمایشگاه هوش محاسباتی

پس از ترین شبکه و export و ارسال به سیمولینک در مدار سیستم قرار می گیرد:



خروجی سیستم و شبکه به صورت زیر است:



همان طور که مشاهده می شود خروجی شبکه مرتبه ۲، نسبت به شبکه مرتبه ۳، خطای بیشتری با خروجی اصلی سیستم دارد اما با این وجود شناسایی قابل قبولی برای سیستم است.

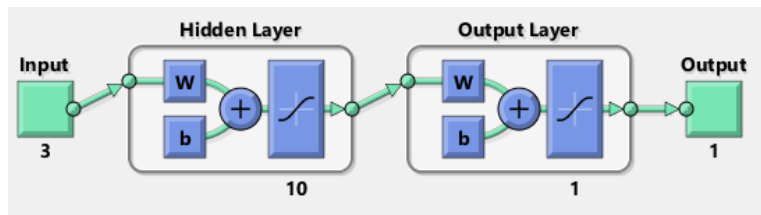
# آزمایشگاه هوش محاسباتی

– با استفاده از داده‌ها یک سیستم مرتبه ۱ تخمین بزنید و مراحل سؤال ۱ را تکرار کنید.

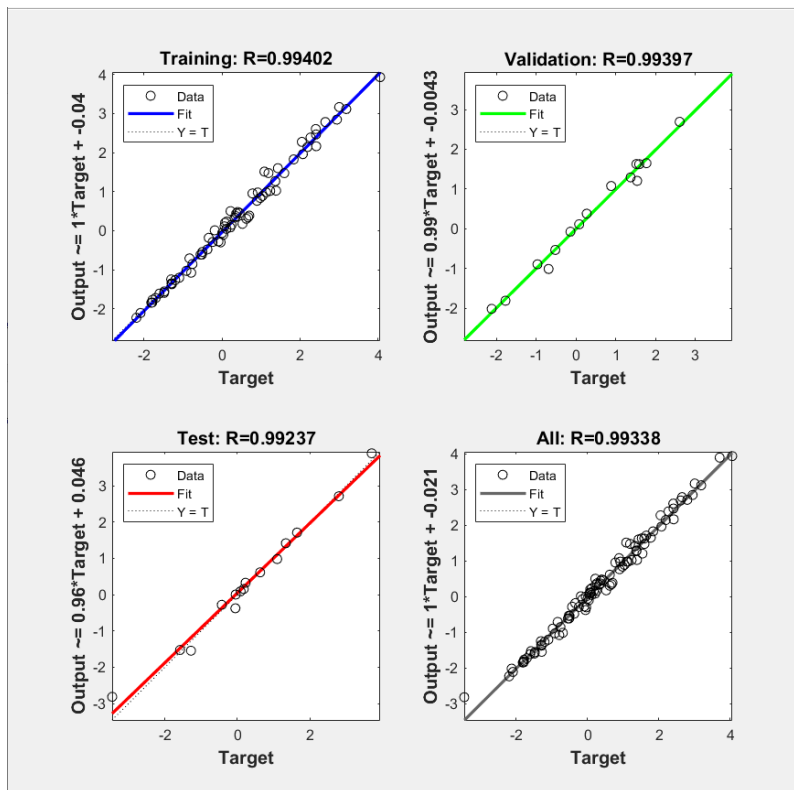
برای تخمین مرتبه ۱، فقط یک تاخیر یافته خروجی را در نظر می‌گیریم:

```
Input = input.Data;  
Output = output.Data;  
Input_1_delay = [0;Input(1:end-1)];  
Output_1_delay = [0;Output(1:end-1)];  
X = [Input, Input_1_delay, Output_1_delay]';  
Output = Output';  
nntool;
```

بنابراین شبکه دارای ورودی با ابعاد ۳ است:

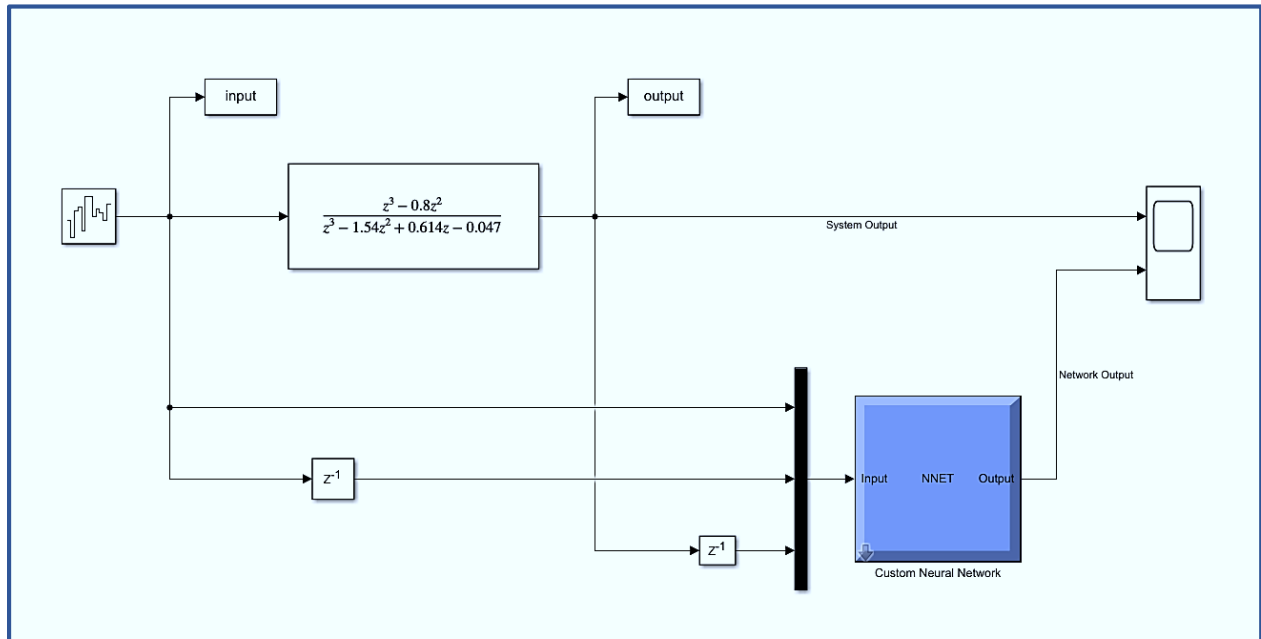


شکل ۲- شبکه مرتبه ۱ ترین شده برای شناسایی

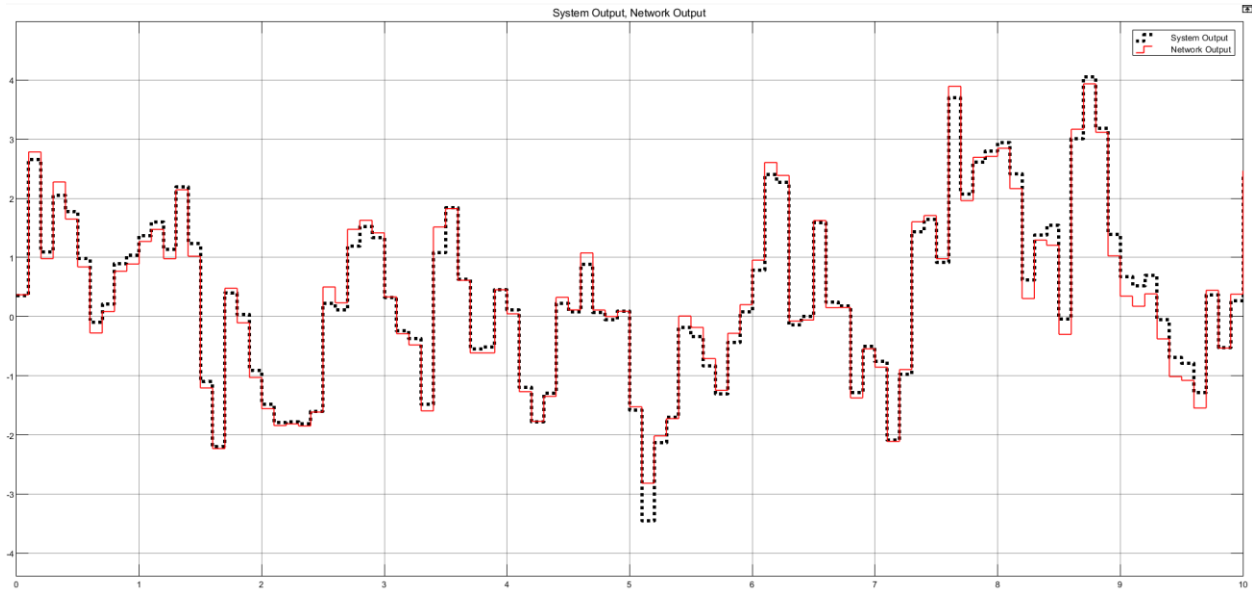


# آزمایشگاه هوش محاسباتی

پس از ترین شبکه و انتقال به سیمولینک داریم:



خروجی سیستم و شبکه به صورت زیر است:



همان طور که مشاهده می شود خروجی شناسایی شده توسط شبکه تفاوت زیادی با سیستم اصلی دارد و نسبت به حالت قبل شناسایی بدتری انجام شده است.

## آزمایشگاه هوش مناسباتی

- اگر قرار باشد در یک پروژه واقعی چنین کاری انجام دهید، کدام یک از تخمین‌های فوق را برای مدل‌سازی سیستم و داده‌ها انتخاب می‌کنید. چرا؟

سیستم مرتبه ۳ - با توجه به خروجی‌های شبکه که در هر قسمت تشریح شد، هر چه مرتبه سیستم بالاتر بود، شناسایی بهتری توسط شبکه انجام می‌شد.