سمینار درس شبکه های عصبی

موضوع:

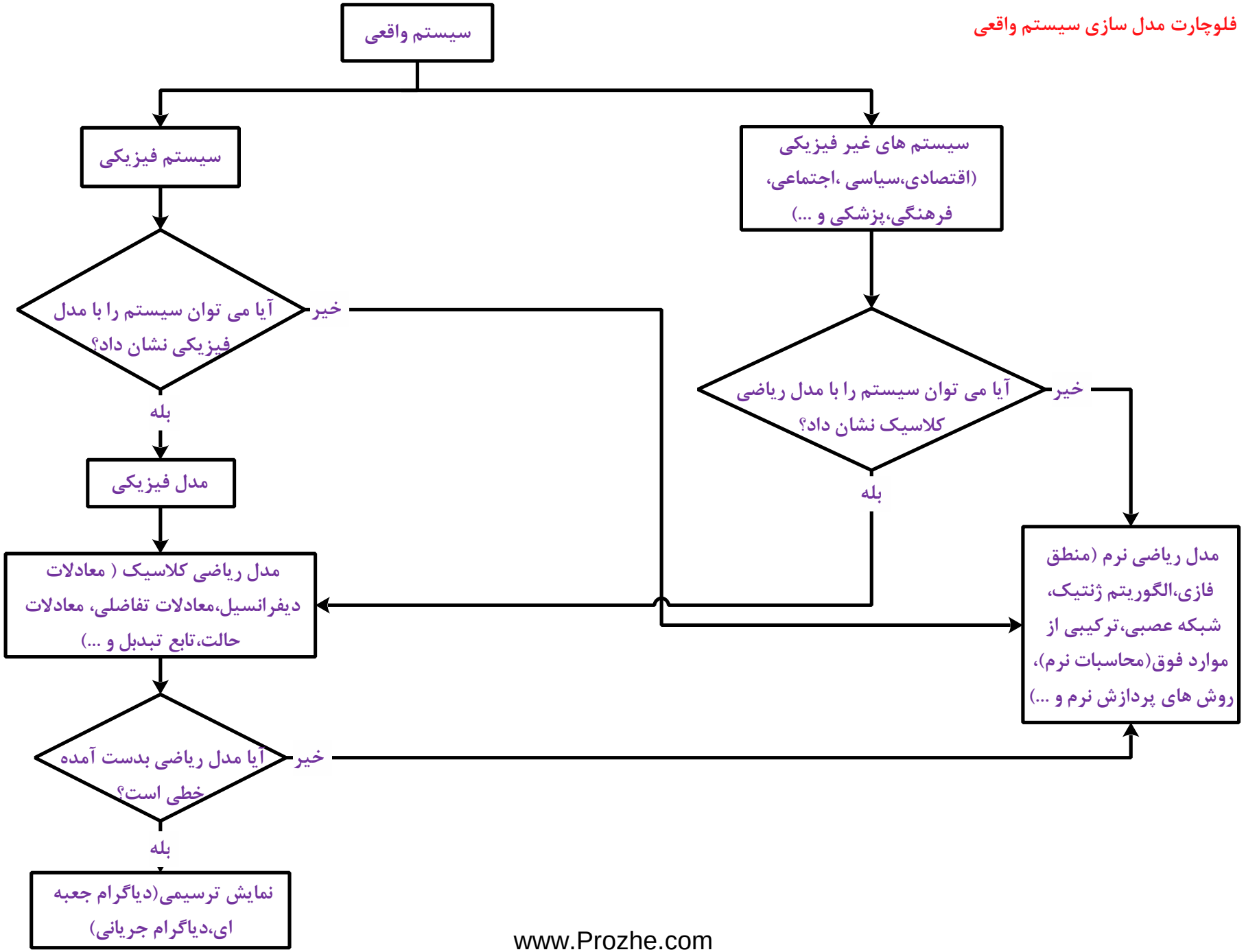
کاربرد شبکه های عصبی در کنترل

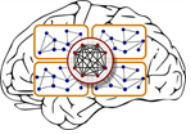
هادی بهین
ارشد مکاترونیک
hadi.behin@gmail.com
دانشگاه آزاد تهران جنوب
تیر ۹۴

سوال؟

با وجود کنترل کلاسیک وکنترل مدرن چه لزومی به استفاده از شبکه عصبی یا روش های محاسبات نرم در کنترل وجود دارد؟

فلوچارت مدل سازی سیستم واقعی





پس در حالت های زیر در کنترل یک سیستم باید به سمت روش های محاسبات نرم برویم:

- اگر برای یک سیستم فیزیکی نتوان مدل مناسب فیزیکی طراحی کرد.
- اگر برای سیستم های فیزیکی و غیر فیزیکی نتوان با روش های ریاضی کلاسیک ،مدل ریاضی آنها را استخراج کرد.
- رابطه بین ورودی و خروجی غیر خطی باشد.(مدل ریاضی بدست آمده غیر خطی باشد).

در واقع همیشه این امکان وجود ندارد که رابطه بین ورودی و خروجی با ریاضی کلاسیک بیان شود و همچنین همیشه این وجود ندارد که یک مدل خطی بدست آورد.

مدل های غیر خطی به خوبی مدل های سیستم های غیر خطی درک نشده اند.روش هایی برای بعضی از سیستم های غیر خطی ساده تر وجود دارد.

شبکه های عصبی روش مفیدی برای کنترل سیستم های غیر خطی پیچیده ارائه می دهد ،زیرا نیازی به مدل سیستم ندارد.

مراحل پیشرفت روش های کنترل سیستم ها

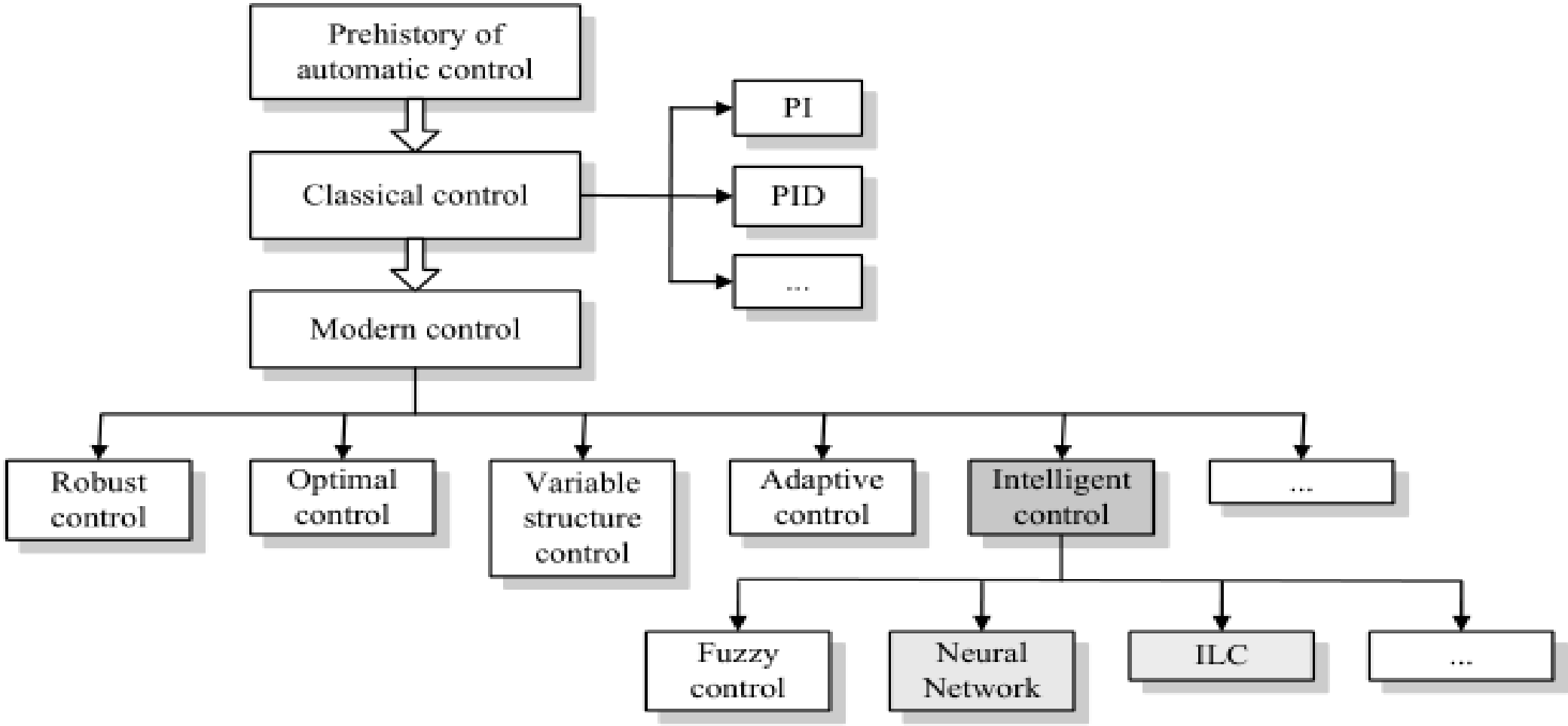
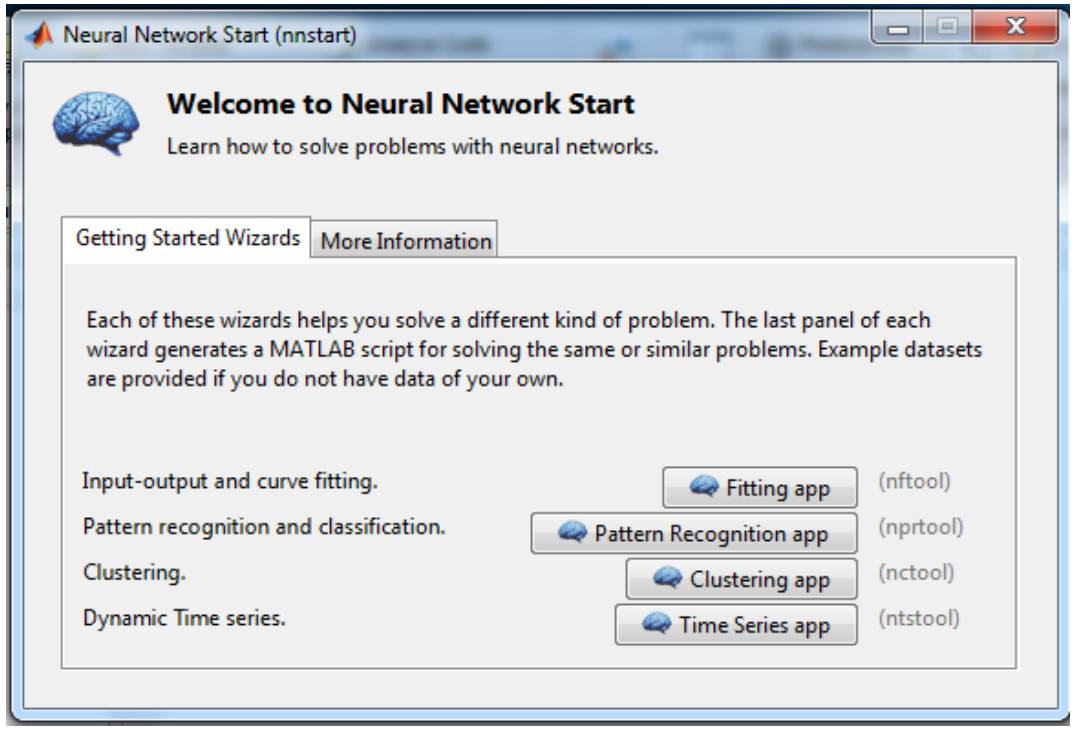


Figure 1.3.1 Development of control technology

کدامیک از کاربرد های شبکه عصبی در کنترل سیستم ها استفاده می شود؟

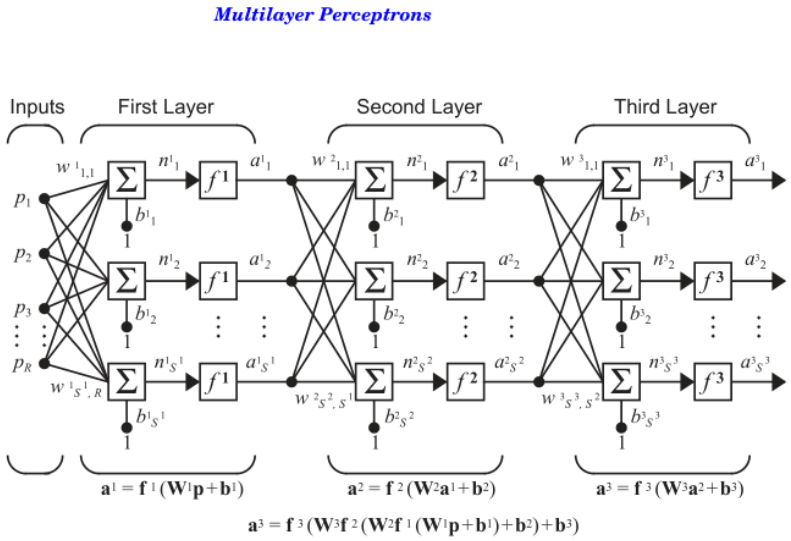


- با توجه به مرجع بودن برنامه **MATLAB** در بسیاری از مباحث مهندسی از جمله شبکه های عصبی، اگر **nnstart** را در خط فرمان برنامه **MATLAB** نوشته شود، در صفحه باز شده چهار کاربرد را برای استفاده از شبکه عصبی در نظر گرفته است:

➢ Input-Output and Curve fitting
- Pattern recognition and classification
- Clustring
- Dynamic Time series

- چون در سیستم های کنترلی معمولا ورودی و خروجی به plant را داریم ولی خود plant ممکن است ناشناخته یا در فرم پیچیده غیر خطی باشد، بنابراین از اولین کاربرد شبکه های عصبی یعنی تخمین توابع با استفاده از ورودی و خروجی به Plant، استفاده می شود.

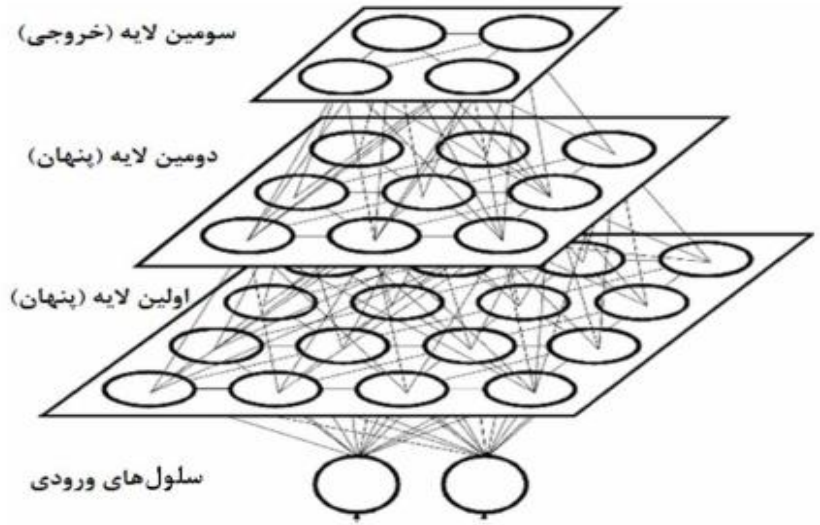
مهمترین نوع شبکه عصبی که در کنترل استفاده می شود چیست؟



شبکه عصبی پرسپترون چند لایه (MultiLayer Perceptron)

پرسپترون چند لایه مزیتی بر شبکه ادلاین(خطی) دارد و آن این است که توانایی تقلید هر رابطه ورودی و خروجی را دارد.

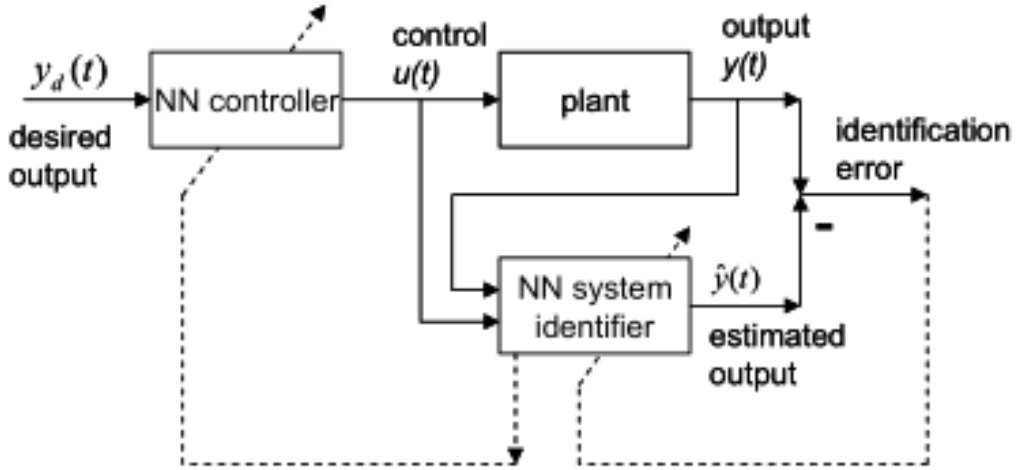
بنابراین از MLP می توان برای رقابت با هر کنترل کننده موجودی استفاده کرد.



نمایش یک شبکه عصبی پرسپترون سه لایه

کاربرد شبکه عصبی پرسپترون چند لایه(MLP) در کنترل

- **کاربرد اول : شناسایی سیستم با استفاده ازمدل مستقیم(Forward Models)**



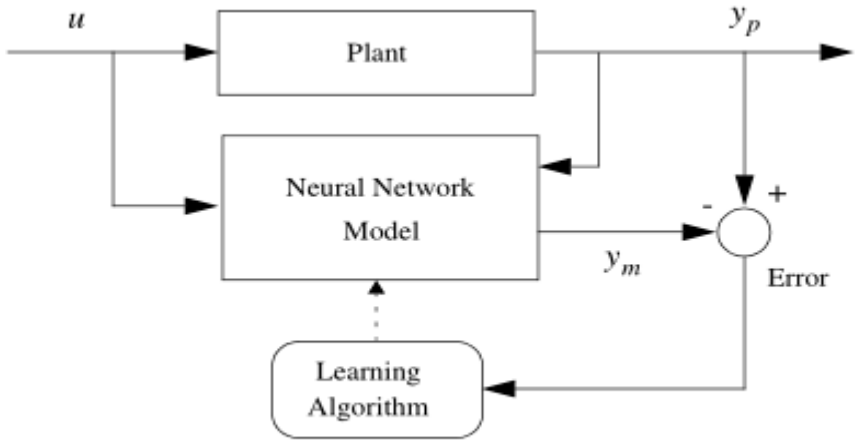


Figure 13 Plant Identification

جالب ترین خاصیت یک شبکه پرسپترون چند لایه قابلیش در تقلید کردن هر رابطه ورودی و خروجی است(تقریب توابع). در یک سیستم خطی،این رابطه تابع تبدیل،G ، است وهدف از شناسایی سیستم این است که تقریبی از تابع تبدیل سیستم ، پیدا کنیم .از این امر می توان در مدل سازی یا تقلید Plant ،یا گاهی در کنترل تطبیقی،در جایی که تاثیر خروجی های کنترل کننده باید قبل از اعمال آن به دستگاه معلوم باشند, استفاده کرد.

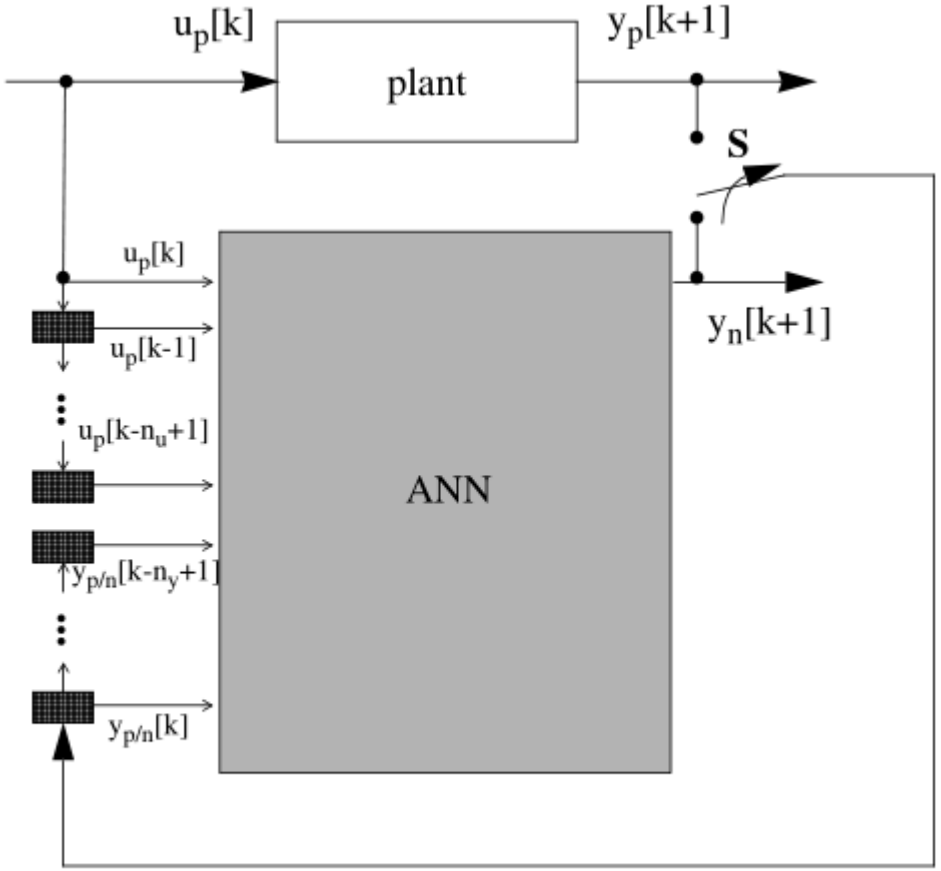
درواقع اینجا شبکه عصبی به عنوان یک تخمین گر(Estiminator) عمل می کند ،که ورودی و هدف ها را به عنوان ورودی دریافت می کند و خروجی تخمین زده را با هدف مقایسه کرده و از خطای حاصل را برای تنظیم وزن های شبکه عصبی پس انتشار می کند.

اگر سیستم زمان گسسته غیر خطی با معادله زیر توصیف شود:

$$y_p[k+1] = f(y_p[k], \ldots, y_p[k-n_y+1], u[k], \ldots, u[k-n_u+1])$$

n_y and n_u تاخیر زمانی ورودی و خروجی

این رابطه یک نگاشت بین یک فضای n_u+n_y بعدی ورودی و فضای یک بعدی خروجی بیان میکند.



Forward plant modelling with ANNs

یک شبکه عصبی پرسپترون چند لایه می تواند رابطه بین این ورودی ها و خروجی ها را تقریب بزند.
در واقع تابع نامعلوم f را تقریب می زند.

کاربرد شبکه عصبی پرسپترون چند لایه(MLP) در کنترل

کاربرد دوم: کنترل حلقه باز با استفاده از مدل معکوس(Inveres Modelling)

تابع تبدیل مطلوب برای یک سیستم ۱ است.

اگر تابع تبدیل یک plant ، G باشد اگر بتوانیم معکوس تابع تبدیل(1/G) را پیدا کنیم ،تابع بسیار مفیدی خواهد بود.

اگر یک کنترل کننده به شکل 1/G باشد و این کنترل کننده در ابتدای plant، بدون فیدبک، قرار داده شود ،آنگاه تابع تبدیل کلی عبارت خواهد بود از $\frac{1}{G} \times G = 1$

بنابراین به عنوان یک قاعده کلی معکوس G را بتوانیم تقریب بزنیم، $\frac{1}{G}$، آنگاه از آن می توان به عنوان یک کنترل کننده حلقه باز استفاده کرد و به کنترل خوبی برای plant دست یافت.از آنجایی که یک شبکه عصبی MLP می تواند هر رابطه ای را تقریب بزند باید بتواند معکوس تابع را هم تقریب بزند.

در واقع target ها را به عنوان ورودی به شبکه عصبی داده ایم و خطای بین خروجی شبکه عصبی و ورودی های plant را به پس انتشار میکنیم.

در این حالت plant و شبکه عصبی به عنوان کنترلر به صورت سری هستند.

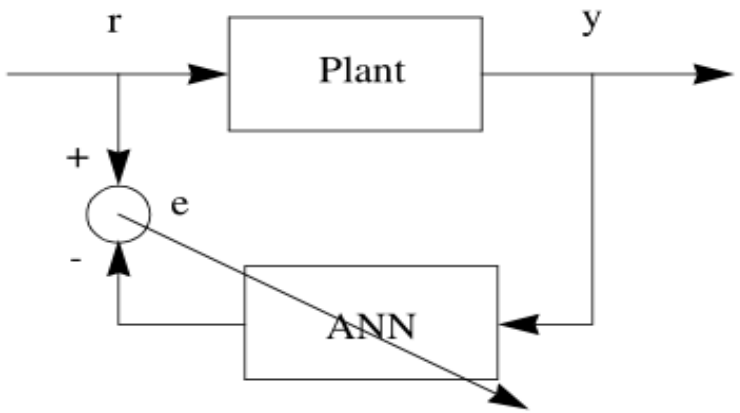
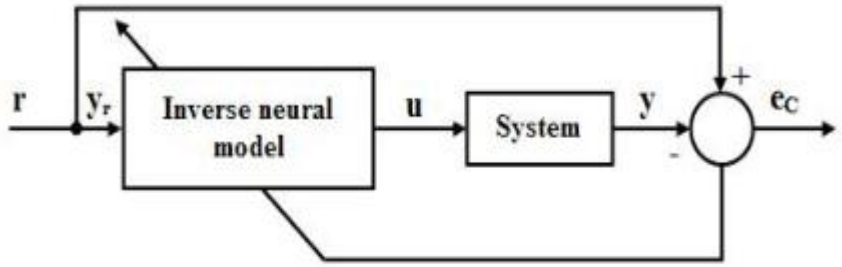
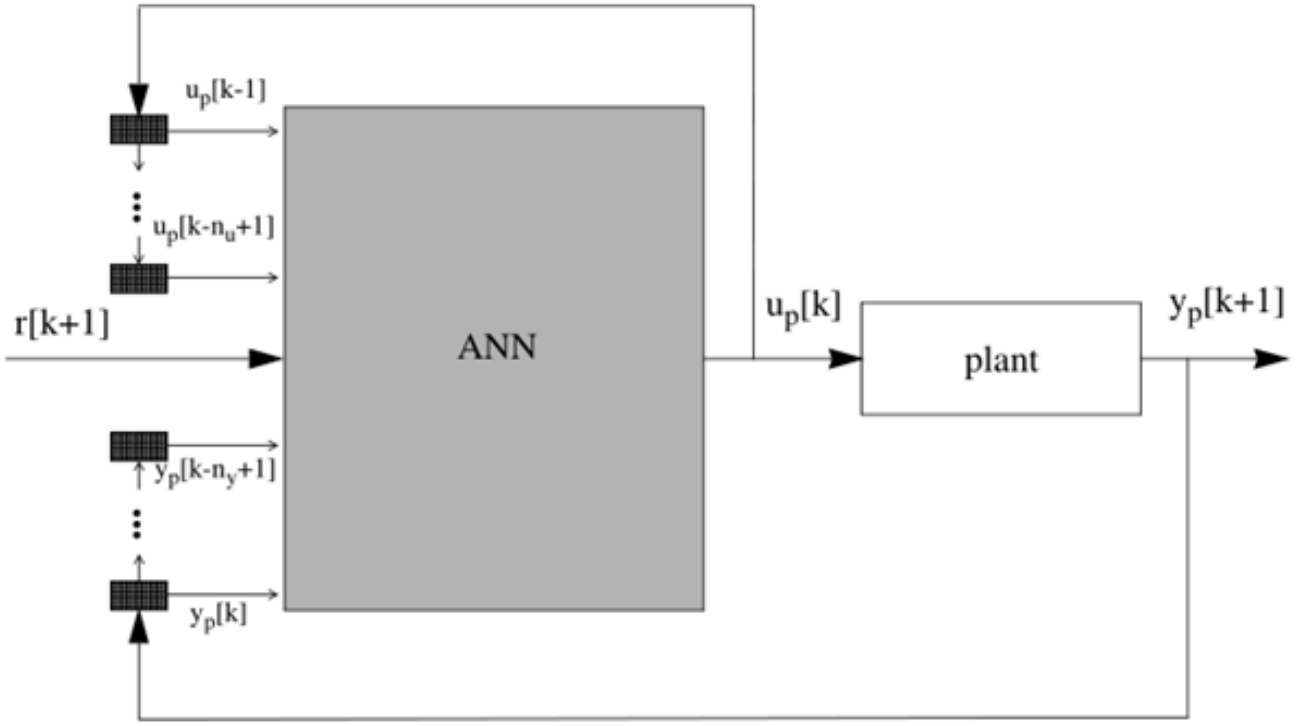


Fig 2.6 - Generalised learning architecture



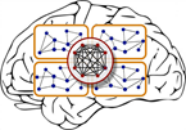
اگر سیستم زمان گسسته غیر خطی با معادله زیر توصیف شود:

$$u[k] = f^{-1}(r[k+1], y_p[k], \ldots, y_p[k-n_y+1], u[k-1], \ldots, u[k-n_u+1])$$



Inverse modelling with ANNs

در واقع شبکه عصبی معکوس تابع بین ورودی ها و خروجی ها را تقریب می زند و از آن به عنوان کنترل کننده حلقه باز استفاده میکند بصورت سری با دستگاه با تابع نامعلوم .



مهترین معماری های شبکه عصبی برای کنترل سیستم ها

- Fixed Stabilizing Controllers
- Adaptive Inverse Control
- Model predictive Control
- Nonlinear Autoregressive-Moving Average(NARMA-L2) control
- Model Reference Control

1. Fixed Stabilizing Controllers

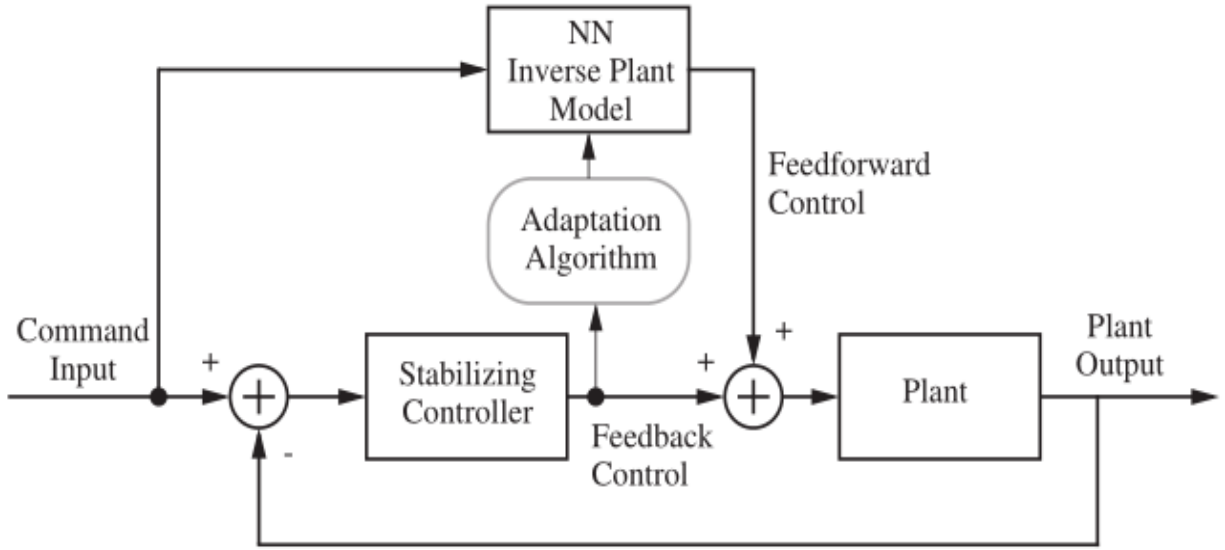
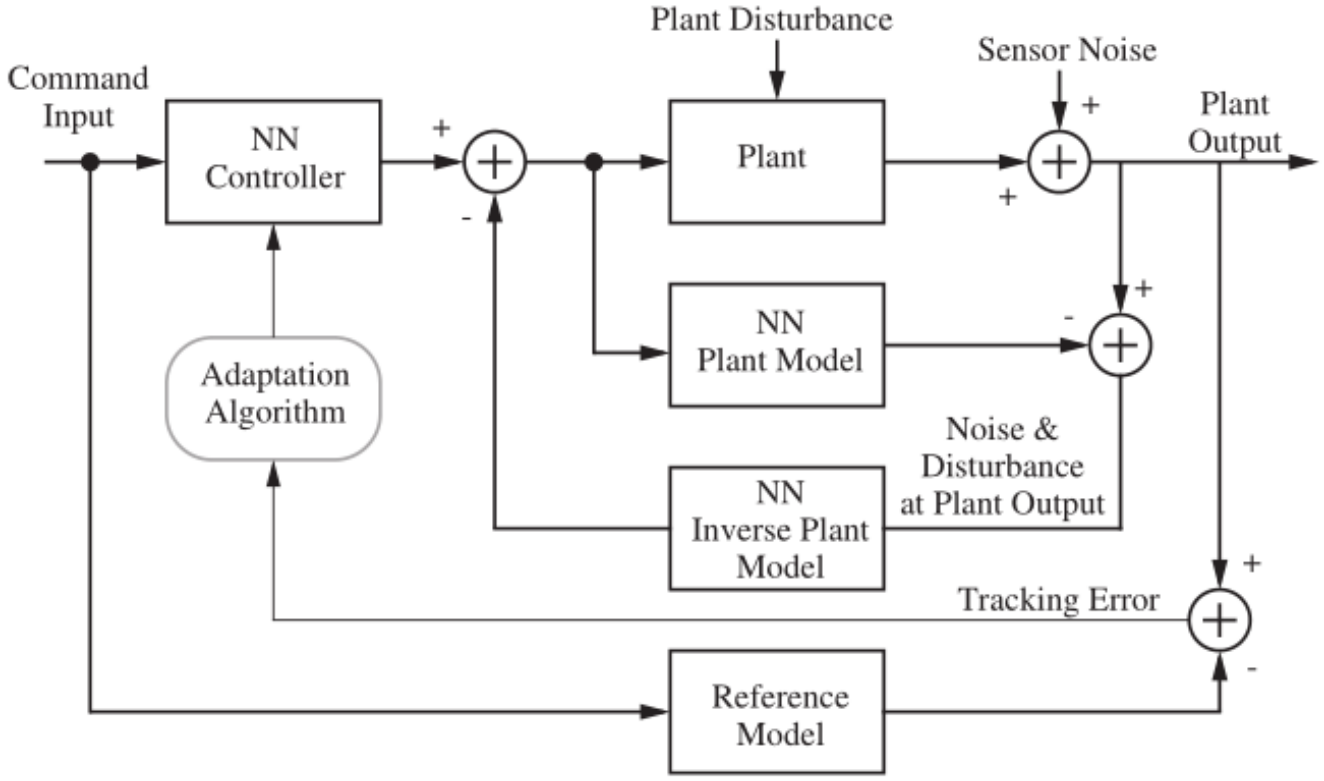


Figure 14 Stabilizing Controller

- این بلوک دیاگرام برای کنترل مسیر حرکت یک بازوی ربات است.
- دارای یک کنترلر تناسبی با بهره مناسب استفاده شده در یک فیدبک کنترلی که پایدار کننده سیستم است .
- همانطور که می بینیم ورودی کنترلی به Plant مجموع سیگنال حلقه فیدبک و سیگنال Feedforward که مدلی است تقریب زده شده بصورت معکوس از Plant با استفاده از شبکه عصبی. که شبکه عصبی دو ورودی دارد یکی ورودی مرجع و دیگری سیگنال error فیدبک در واقع با این ورودی و این خروجی شبکه عصبی فیدبک کنترل با کنترلر تناسبی را شبیه سازی می کند . با این کار همیشه سیستم پایدار باقی خواهد ماند.

2. Adaptive Inverse Control



Adaptive Inverse Control System

- در این معماری خطای بین target که خروجی سیستم مرجع است با خروجی plant به عنوان خطای پس انتشار برای آپدیت وزن های شبکه عصبی کنترلر،برای مینیم کردن تابع هزینه خطا.
- در نوع کنترل سیستم تحت تاثیر نویز های فرایندی و نویز های اندازه گیری قرار دارد. استفاده از یک شبکه عصبی بصورت موازی.
- تفاوت بین خروجی نویز دار plant و خروجی تخمینی شبکه عصبی به عنوان ورودی شبکه عصبی که عکس شبکه عصبی plant می باشد وارد می شود و خروجی این شبکه سیگنال فیلتر شده نویز می باشد که از ورودی plant کم می شود.

❖ این یک ایده برای حذف کردن تاثیرات نویز فرآیندی و آندازه گیری است.

3. Model predictive Control

- کنترل کننده ی کنترل مدل پیش گویانه از شبکه های عصبی برای پیش گویی رفتار آینده plant به سیگنال های کنترلی استفاده می کند.
- سپس یک الگوریتم بهینه سازی سیگنال های کنترلی را محاسبه کرده و کارایی را بهینه سازی می کند.

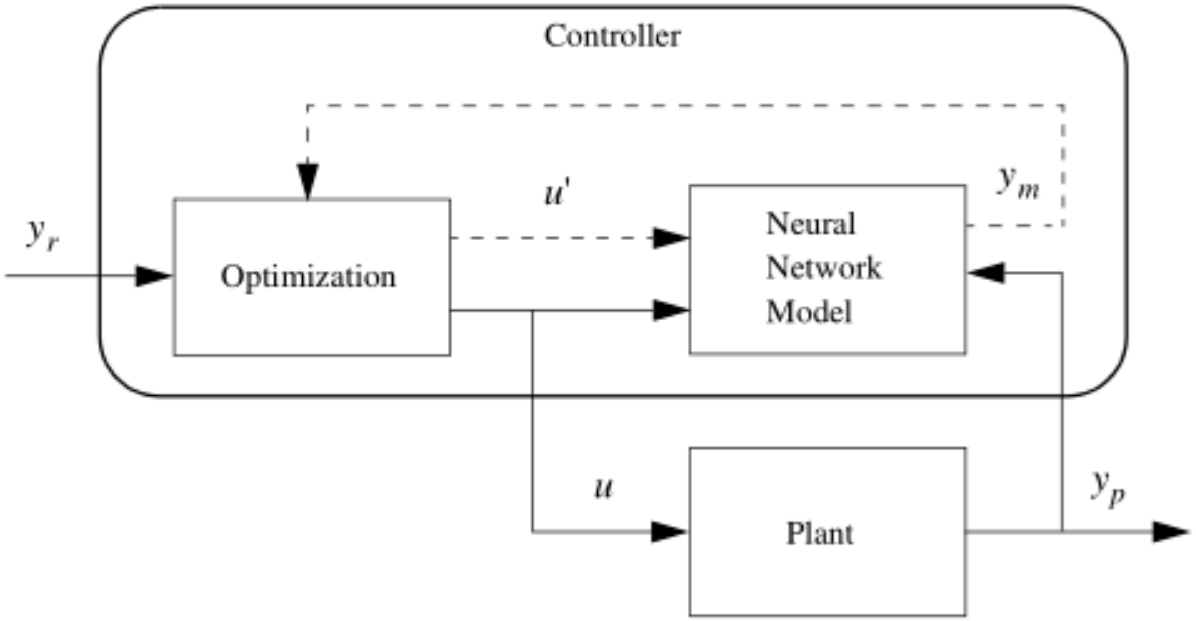


Figure 15 Neural Network Predictive Control

تابع هزینه بهینه سازی

$$J = \sum_{j=N_1}^{N_2} (y_r(k+j) - y_m(k+j))^2 + \rho \sum_{j=1}^{N_u} (u'(k+j-1) - u'(k+j-2))^2$$

N ها محدوده های(افق ها)

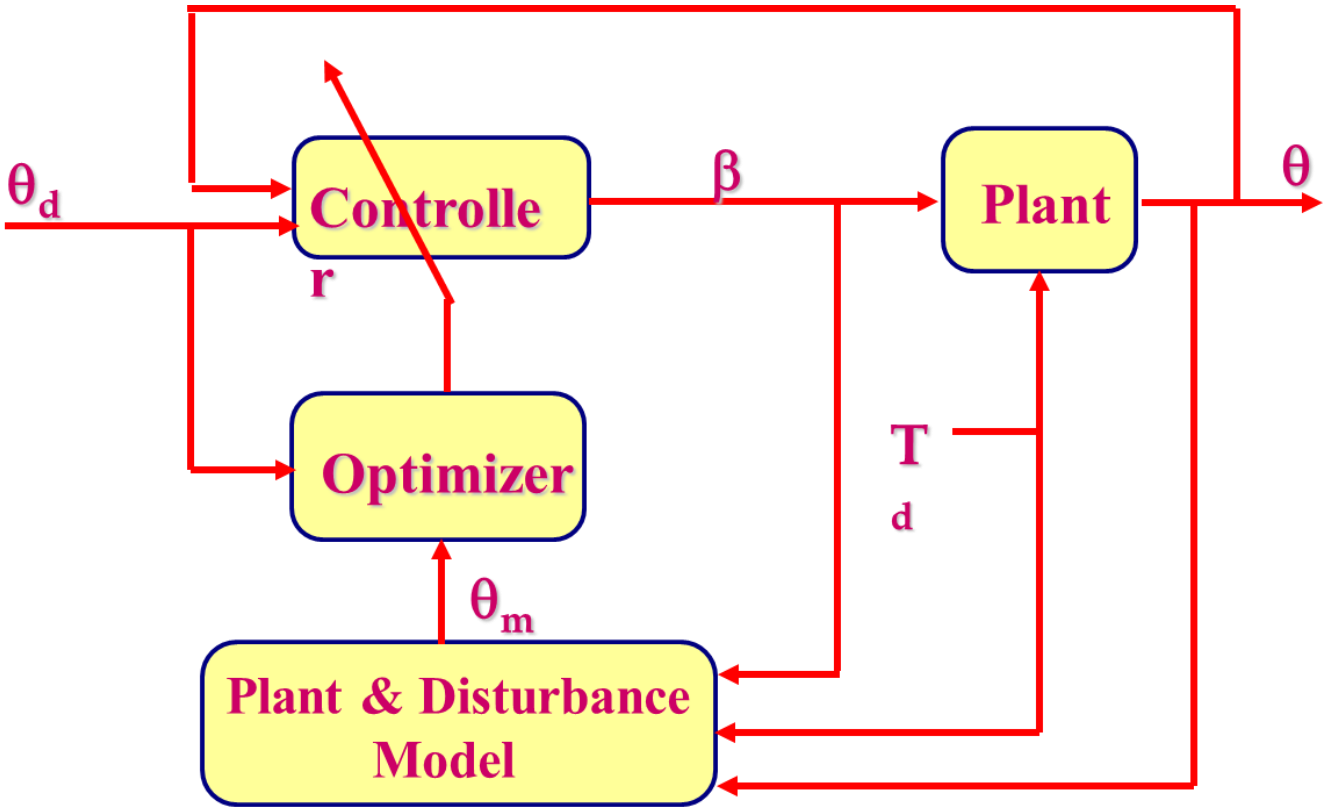
متغیر u سیگنال کنترلی آزمایش که توسط بهینه ساز کمینه می شود.

y_r عکس العمل مطلوب

y_m عکس العمل شبکه و

ρ هم مجموع مربعات گام های کنترلی روی تابع هزینه می باشد.

یک کنترل کننده مدل پیش بین از چند قسمت تشکیل شده است:

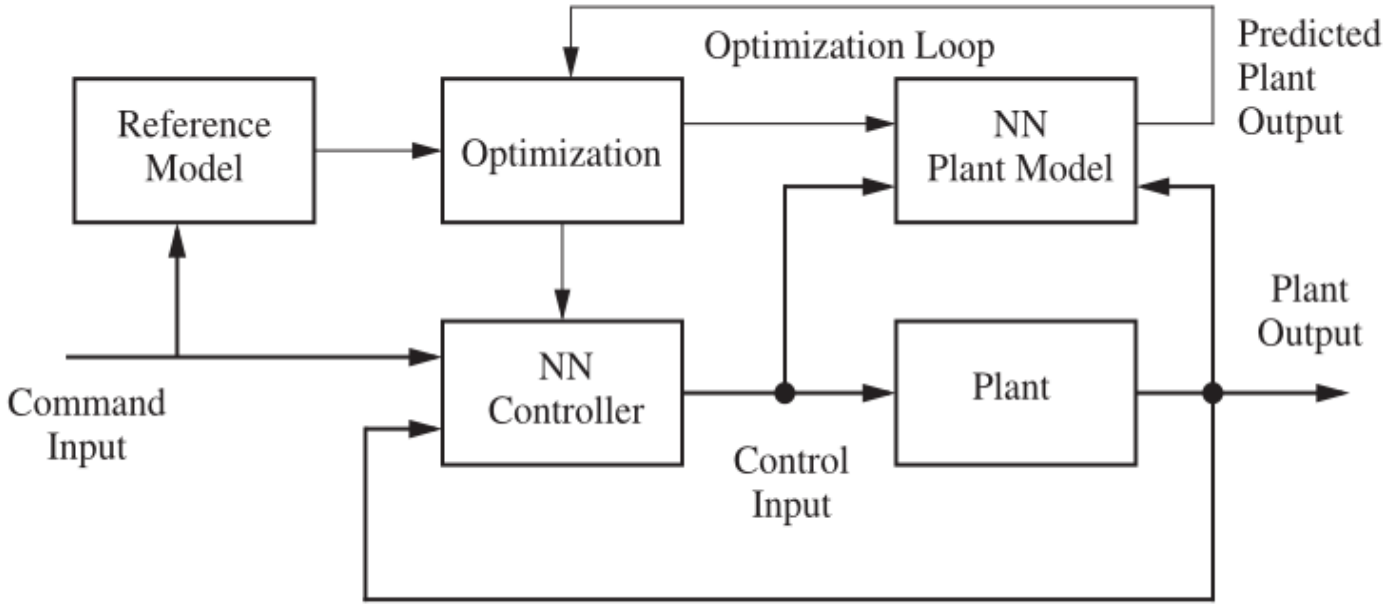


- مدل پیش بین
- بهینه سازی افق لرزان
- تصحیح فیدبک

نمونه کلی از کاربرد شبکه های عصبی در MPC

اجرای سیستم کنترلی :

- مدل شبکه عصبی سیستم تحت کنترل که به صورت feedForward نسبت به plant قرار می گیرد.
- کنترل کننده شبکه عصبی
- تابع عملکرد جهت ارزیابی پاسخ های سیستم
- روند بهینه سازی برای انتخاب بهترین ورودی کنترلی



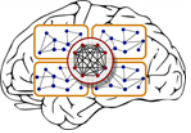
Model Predictive Control

❖ در واقع کنترل کننده شبکه عصبی یاد می گیرد که ورودی انتخاب شده توسط فرآیند بهینه سازی را تولید کند.

❖ وقتی آموزش کامل شود مرحله بهینه سازی می تواند کاملا توسط کنترل کننده شبکه عصبی جایگزین شود.

NMPC

- MPC کلاسیک که بر اساس مدل های غیر خطی پیش بین می باشد فاقد شرایط لازم برای سیستم های غیر خطی است.
- برای سیستم های غیر خطی باید کنترل پیش بین غیر خطی استفاده شود.(NMPC)
- یک مدل از سیستم های غیر خطی پایه و اساس NMPC است ،در ابتدا باید ایجاد شود.



مدل های غیر خطی

- تعدادی از مدل های غیر خطی جهت توصیف سیستم های غیر خطی ،ارائه شده اند.

- در بین این متد ها،مدل زیر که یک مدل ورودی –خروجی عالی است استفاده شده، که تقریبا همه مدل های غیر خطی دیگر را پوشش می دهد:

- **NONLINEAR AUTOREGRESSIVE MOVING AVERAGE WITH EXOGENOUS INPUTS**

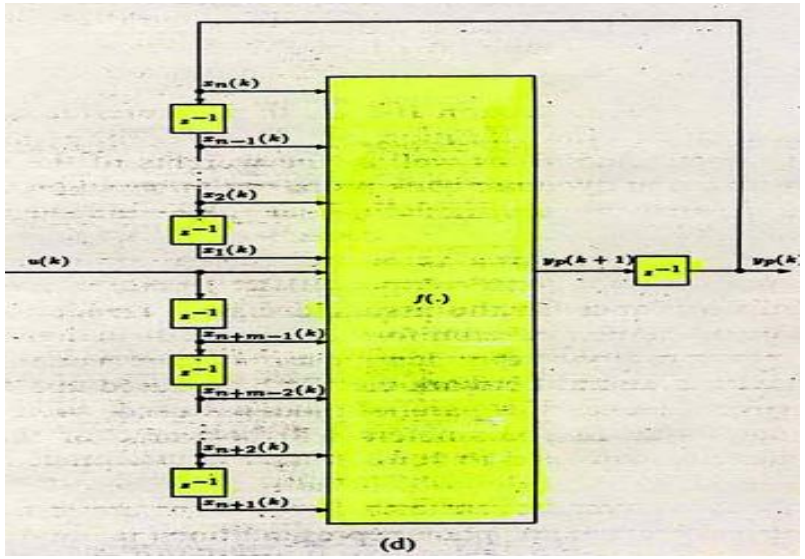
(NARMAX)

NARMAX

- فرض کنید مدل NARMAX یک سیستم غیرخطی تک ورودی- تک خروجی SISO به صورت زیر باشد :

$$y(k+1) = f\big[y(k), y(k-1), \cdots, y(k-n_y+1), u(k), u(k-1), \cdots, u(k-n_u+1), e(k), e(k-1), \cdots, e(k-n_e)\big] + e(k)$$

$$y(k) = u^3(k-1) + 2u(k-2) + u(k-1)u(k-2) + \sin\big[y(k-1)\big]$$



- u : ورودی سیستم غیرخطی
- y : خروجی سیستم غیرخطی
- nu : مرتبه ورودی مدل
- e : نویز سفید گوسی با میانگین صفر
- f : رابطه نگاشت ورودی – خروجی
- ny : مرتبه خروجی مدل

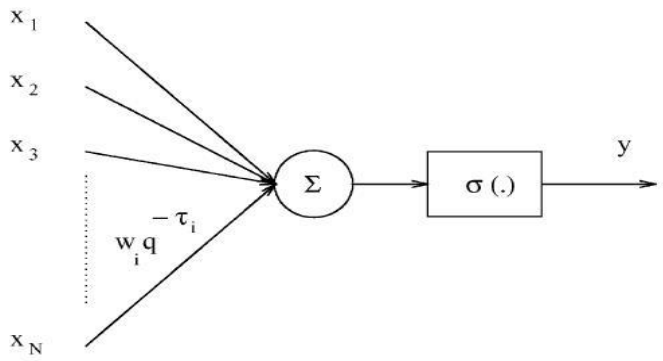
NARMAX

- اگر از اثر نویز سفید گوسی صرفنظر شود. فرمول مربوطه به صورت زیر ساده می شود :

$$y(k+1) = f\Big[y(k), y(k-1), \cdots, y(k-n_y+1),$$
$$u(k), u(k-1), \cdots, u(k-n_u+1)\Big]$$

- این فرمول یک مدل پیش بین یک مرحله ای سیستم غیرخطی تک ورودی- تک خروجی است.

شباهت سیستم های عصبی با مدل های غیر خطی



Dynamic neuron structure, $q^{-\tau}$ is the shift operator.

معادلات حالت سیستم های غیر خطی

$$y_p(k) = \Psi[x(k)]$$

$$y_p(k+1) = \Psi[\Phi[x(k), u(k)]]$$

$$y_p(k+n-1) = \Psi[\Phi[\Phi[\Phi[x(k), u(k)], u(k+1)], \cdots, u(k+n-2)]]$$

روابط شبکه های عصبی

$$y = \Gamma[W^3\Gamma[W^2\Gamma[W^1 u]]]$$

$$x(k+1) = N_1[x(k)]$$

وزنهای شبکه عصبی=پارامترهای سیستم

one-step predictive model based on BP neural network

اگر اطلاعات گذشته و رخ داده ورودی u(k),u(k-1),…,u(k-nu+1)

و اطلاعات گذشته و رخ داده خروجی y(k),y(k-1),…,y(k- ny+1)

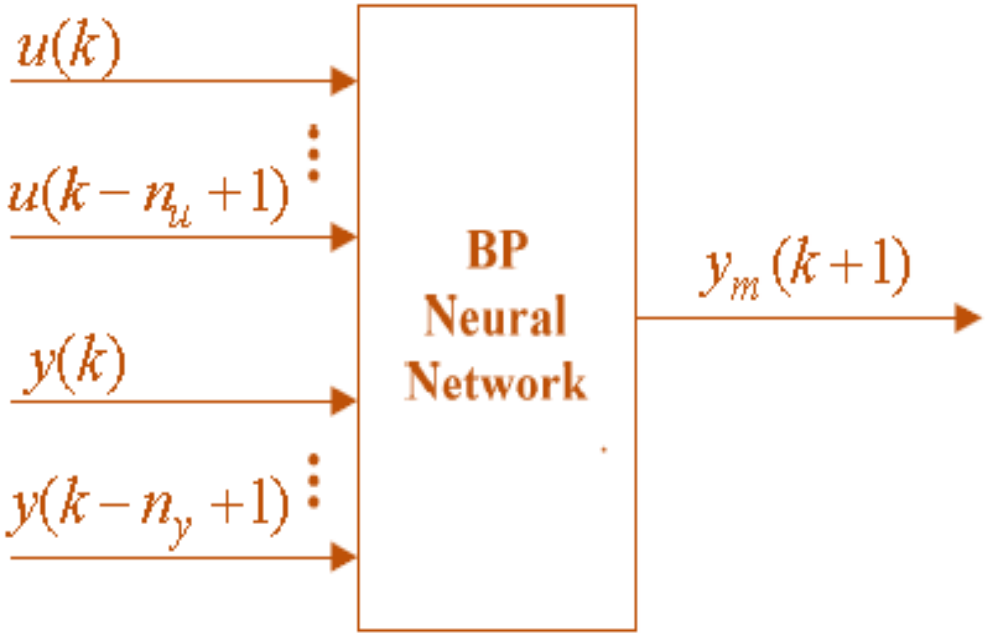
به عنوان بردار ورودی یک شبکه عصبی چند لایه با الگوریتم BP

و آخرین اطلاعات خروجی y(k+1) به عنوان بردار خروجی شبکه عصبی استفاده شود

آنگاه یک مدل پیش بین تک گامی براساس شبکه عصبی BP می تواند تشکیل شود.

طرح اولیه مدل پیش بین گام اول براساس شبکه عصبی BP

ym(k+1) خروجی پیش بین یک سیستم غیرخطی در k+1 است.



- فرمول زیر بیان ریاضی مدل پیش بین گام اول براساس شبکه عصبی BP است :

$$y_m(k+1) = F_{NN}\left[\mathbf{P}(k)\right]$$

- FNN رابطه نگاشت غیرخطی و P(k) بردار ورودی مدل پیش بین است.

$$\mathbf{P}(k) = \left[\tilde{\mathbf{u}}^T(k), \tilde{\mathbf{y}}^T(k)\right]^T = \left[p_1, p_2, \cdots, p_R\right]^T$$

$$\tilde{\mathbf{u}}(k) = \left[u(k), u(k-1), \cdots, u(k-n_u+1)\right]^T$$

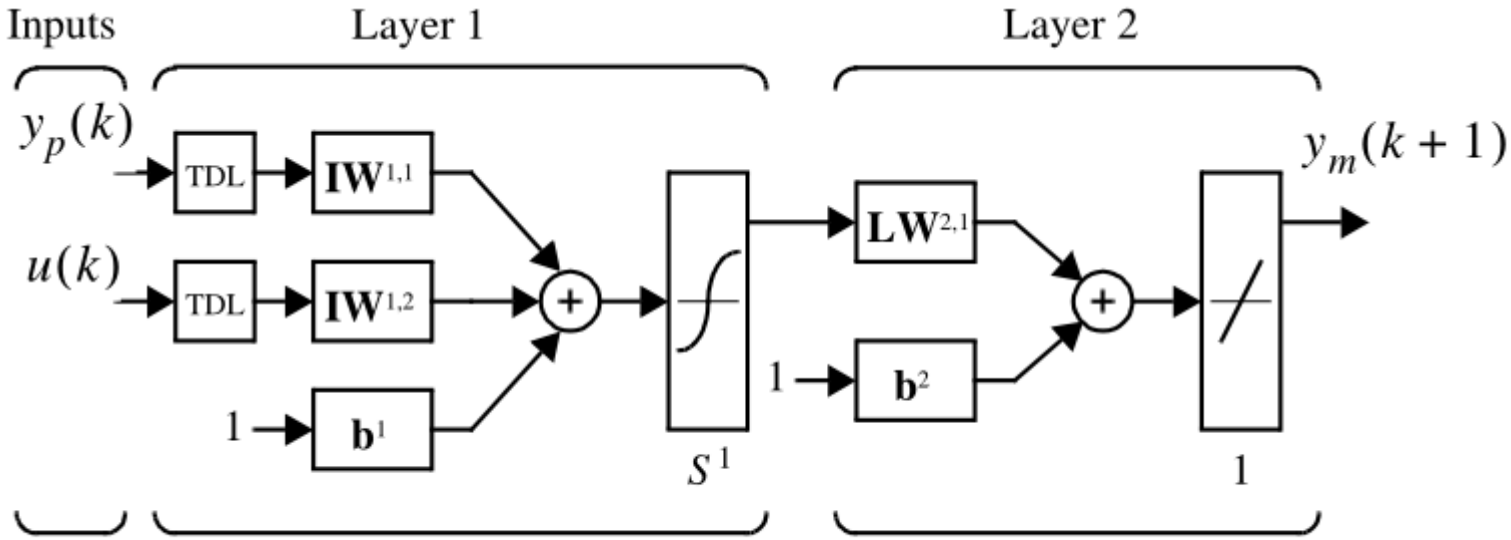
$$\tilde{\mathbf{y}}(k) = \left[y(k), y(k-1), \cdots, y(k-n_y+1)\right]^T$$

$$R = n_u + n_y$$

معماری شبکه عصبی مورد استفاده

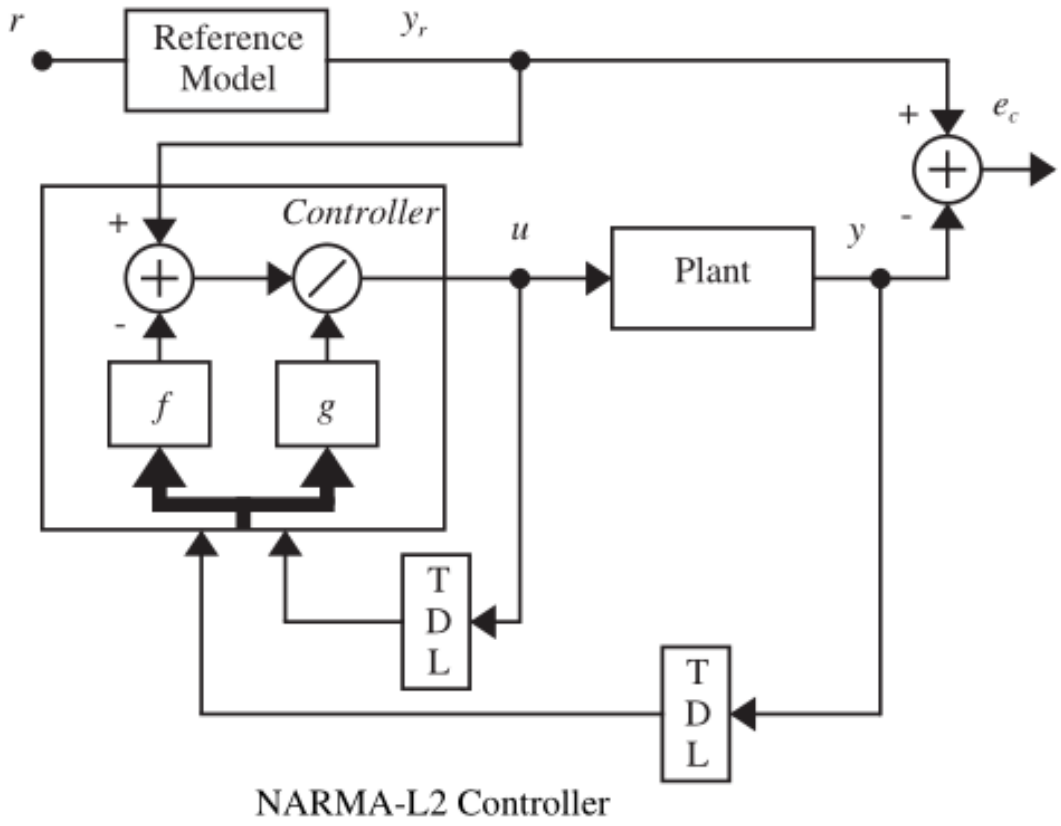
- ثابت می شود که یک شبکه عصبی دو لایه پس انتشار که نرونهای لایه پنهانش از تابع انتقال tansig و نرونهای لایه خروجی اش از تابع انتقال pureline استفاده می کنند , می تواند تقریباً برای همه سیستمهای غیرخطی مناسب باشد به شرط اینکه تعداد نرونهای لایه پنهان به اندازه کافی بالا باشند.

- **این در یک واقع یک شبکه عصبی پس انتشار دو لایه ای است که می تواند جهت تشکیل مدل پیش بین تک گامی غیرخطی انتخاب شود.**



Neural Network Plant Model

4. Nonlinear Autoregressive-Moving Average (NARMA-L2) control



NARMA-L2 Controller

- به این کنترلر به نام خطی سازی پسخوردی(Feedback Linearization) هم شناخته می شود.زمانی که مدل دستگاه دارای فرم خاصی باشد از آن تحت عنوان خطی سازی پس خوردی و زمانی که مدل دستگاه قابل تخمین به همان فرم باشد از آن تحت عنوان NARMA-L2 یاد می شود.
- ایده اصلی در این نوع کنترل،انتقال غیر خطی از دینامیک سیستم به یک دینامیک خطی با از بین بردن داده های غیر خطی می باشد.
- اگر بتوانیم با استفاده از شبکه عصبی MLP از روی مدلplant یک مدل برای کنترلر تخمین بزنیم مشکل حل می شود.

تعریف مدلNARMA

- گام اول همانند مدل پیشگو ،تعریف سیستمی است که باید کنترل شود.بنابراین ابتدا باید ساختار مدل مورد استفاده را مشخص کرد
- یک مدل استاندارد برای توصیف سیستم های غیر خطی مدل NARM می باشد بصورت زیر:

NARMA discrete-time model:

$$y(k+d) = N[y(k), y(k-1), \ldots, y(k-n+1), u(k), u(k-1), \ldots, u(k-n+1)]$$

Non Linear *Auto-Regressive* *Moving Average*

- که u(k) ورودی سیستم و y(k) خروجی سیستم می باشند. می توانیم با این ورودی و خروجی شبکه عصبی را آموزش دهیم تا تابع N را تقریب بزند این همان مرحله شناسایی سیستم است ولی ما قصد داریم کنترل کننده را مدل کنیم بنابراین باید تغییراتی در آن اعمال کنیم.

مدل کنترلر

- اگر بخواهیم خروجی سیستم با خروجی مدل مرجع یکی باشد ($y(k+d) = y_r(k+d)$)
باید در یک کنترل کننده را با توجه به مدل سیستم توسعه دهیم:

$$u(k) = G[y(k), y(k-1), \ldots, y(k-n+1), y_r(k+d), u(k-1), \ldots, u(k-m+1)]$$

- مشکل استفاده از این نوع کنترلر غیر خطی این است که اگر خواسته شود که شبکه عصبی را برای تقریب تابع G که خطای مجموع مربعات(mse) را حداقل می کندوآموزش داد،باید از پس انتشار پویا استفاده شود که این کار بسیار کند انجام می شود.یک راه حل برای این مشکل تخمین مدل می باشد.

تخمین مدل کنترلر

- کنترل کننده ای که در این بخش مورد استفاده قرار می گیرد بر مبنای مدل تخمینی NARMA-L2 عمل می کند،این مدل که توسط دو محقق پیشنهاد شده است به شکل زیر است:

Narendra e Mukhopahyay, 1997: NARMA L2 Norm Model

$$y(k+d) = f[y(k), y(k-1),..., y(k-n+1), u(k-1),..., u(k-m+1)] + g[y(k), y(k-1),..., y(k-n+1), u(k-1), ..., u(k-m+1)] \cdot u(k)$$

- مزیت این فرم در این است که برای ورودی هایی که خروجی هایشان دارای سیگنال مرجع هستند.

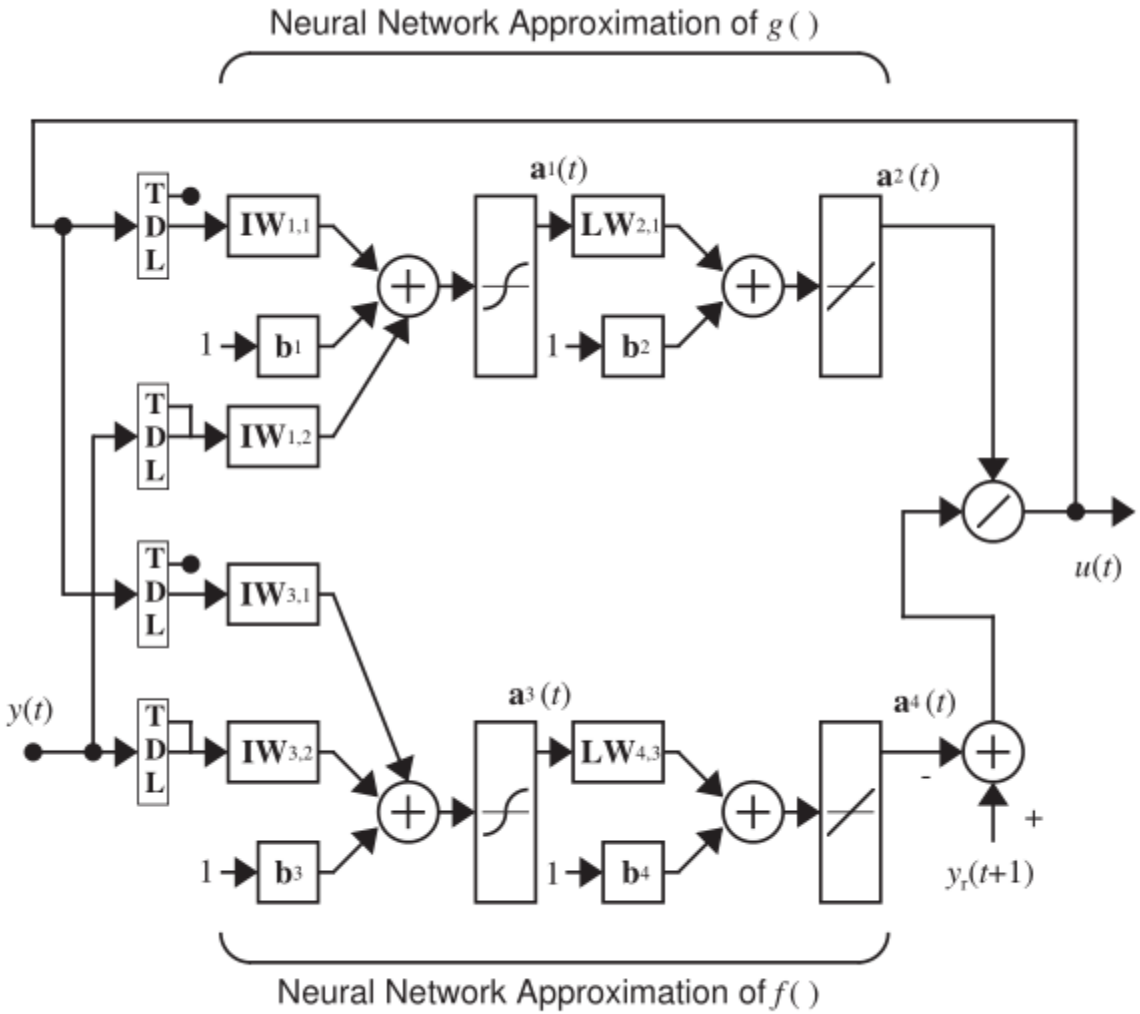
$$y_r(k+d) \equiv y(k+d)$$

کارائی خوبی دارد.کنترل کننده ی حاصل دارای فرم زیر است:

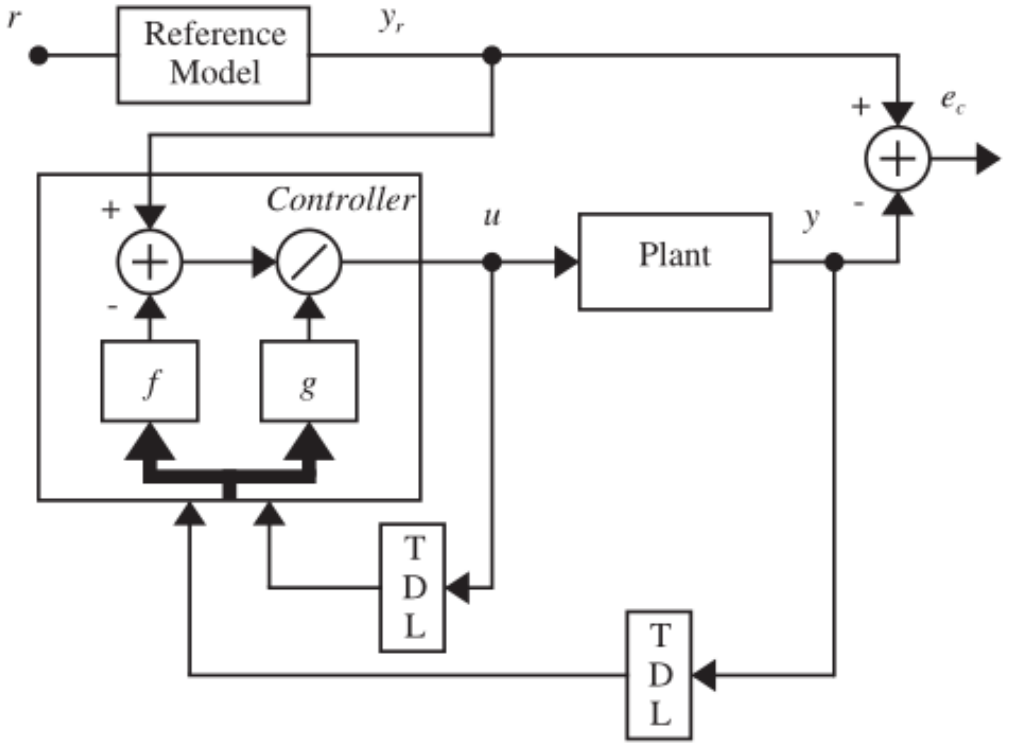
$$u(k) = \frac{y_r(k+d) - f[y(k), y(k-1),..., y(k-n+1), u(k-1),..., u(k-n+1)]}{g[y(k), y(k-1),..., y(k-n+1), u(k-1),..., u(k-n+1)]}$$

که در آن d>=1 در نظر گرفته می شود

ساختار شبکه عصبی شبیه سازی شده برای نروکنترلر NARMA-L2



Implementation of NARMA-L2 Controller



NARMA-L2 Controller

تفاوت کنترل با NARMA-L2 با کنترل PID برای کنترل پایداری در موتور DC تحریک جداگانه

- **معادلات دیفرانسیل حاکم بر این موتور:**

$$v_t = L_a \frac{di_a}{dt} + R_a i_a + E_a \qquad (۱)$$

$$E_a = K\,\omega_a \qquad (۲)$$

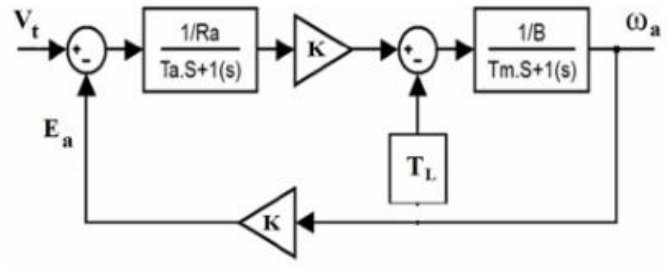
$$J \frac{d^2\theta}{dt^2} + B \frac{d\theta}{dt} - T_L = K i_a \qquad (۳)$$

$$\omega_a = \frac{d\theta}{dt} \qquad (۴)$$

که در آن‌ها R_a و L_a و i_a و V_t به ترتیب مقاومت، اندوکتانس، جریان و ولتاژ آرمیچر، E_a نیروی ضد محرکه موتور، ω_a سرعت زاویه‌ای، θ وضعیت، T_L گشتاور بار، J و B نیز لختی دورانی و ضریب اصطکاک معادل در محور موتور هستند.

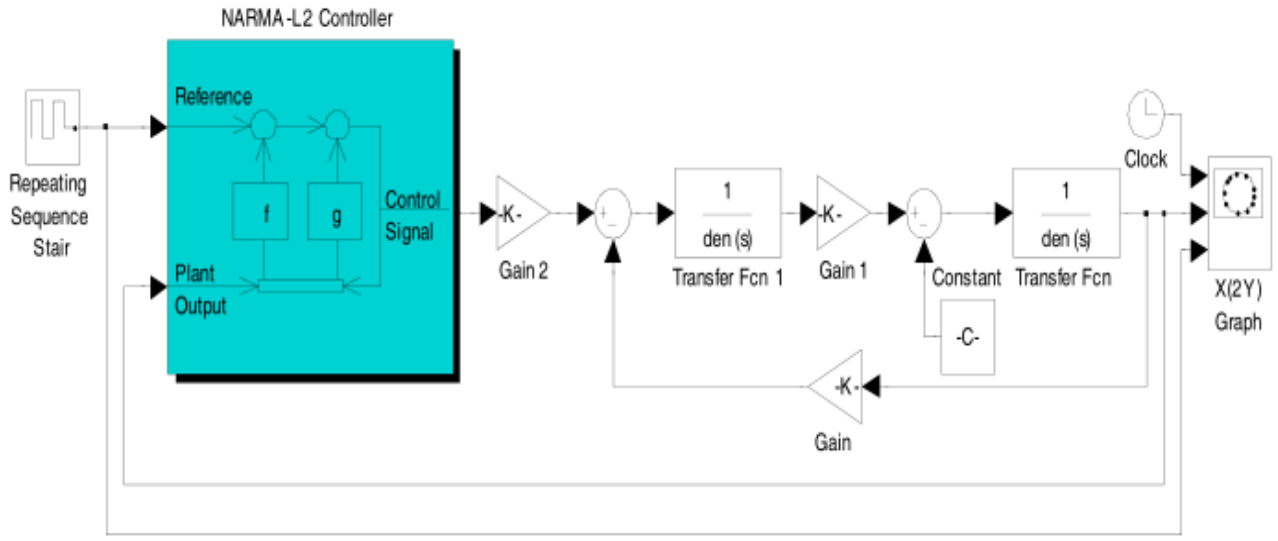
T_m ثابت زمانی مکانیکی موتور ($T_m=J/B$ (s)) و T_a ثابت زمانی جریان آرمیچر موتور ($T_a = L_a/R_a$ (s)) است

بلوک دیاگرام سسیتم:



مدل موتور **dc** تحریک جداگانه

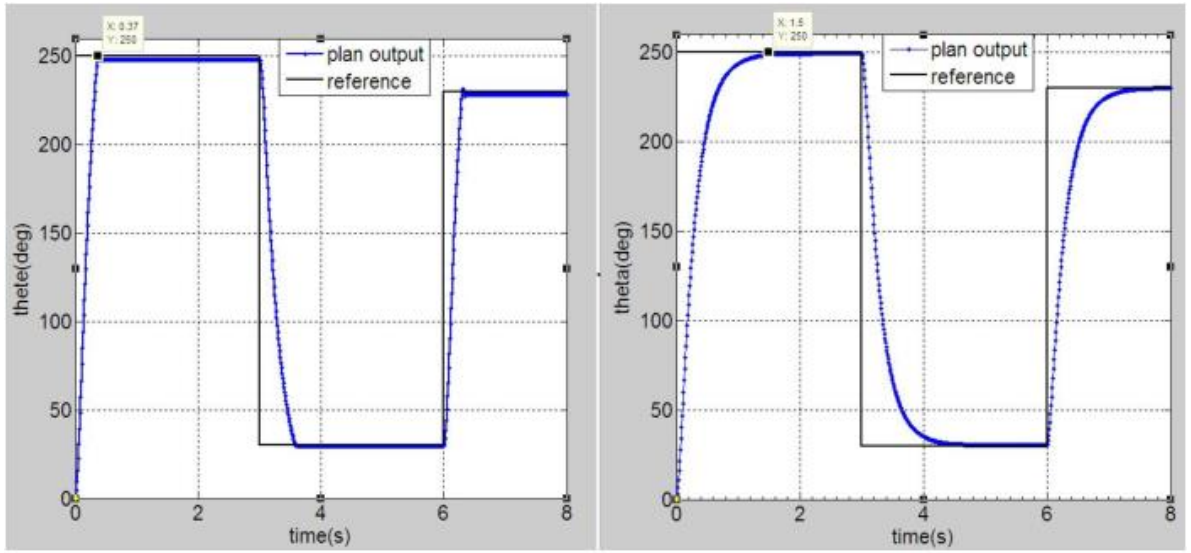
حلقه کنترلی سیستم با کنترلر NARMA-L2



نحوه استفاده از کنترلر NARMA-L2 برای کنترل وضعیت موتور DC

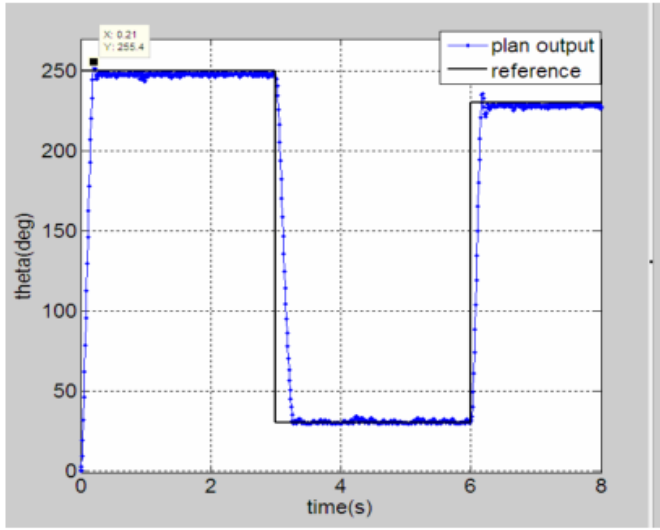
نتیجه به کار بردن کنترل NARMA-L2 و کنترلرPID برای پایدار سازی موتور DC

- در یک مقاله کنترلی طراح شده است برای افزایش پایداری و کاهش زمان رسیدن به حالت مطلوب در برابر اعمال اغتشاش در یک پروسه .که این پروسه را با کنترلرکلاسیک PID نیز انجام داده است و به نتایج زیر رسیده است:
- استفاده از شبکه عصبی باعث شده است سیستم نسبت به تغییر پارامترها (لختی دورانی J و اصطکاک B) مقاوم باشد.و پارامترها در صورت تغییر بصورت خودکار تنظیم می شوند و همچنین این کنترلر در زمان کمتری به حالت پایدار می رسد.
- ولی در کنترلر PID در صورت تغییر یکی از پارامترهای سیستم،کارایی بهینه خود را از دست می دهد و نیاز به تنظیم مجدد پارامتر های کنترلر دارد.
- طبق شکل زیر در شرایطی که ثابت زمانی مکانیکی موتور ۳۰٪ ثابت زمانی اولیه است در این حالت کنترلرNARMA-L2 با حدود ۲٪ overshot و در مدت زمان 0.21 ثانیه به مقدار نهایی خودش می رسد در حالی که کنترلرPID با حدود overshott 11% و پس از حدود 0.56 ثانیه به مقدار نهایی خودش می رسد.

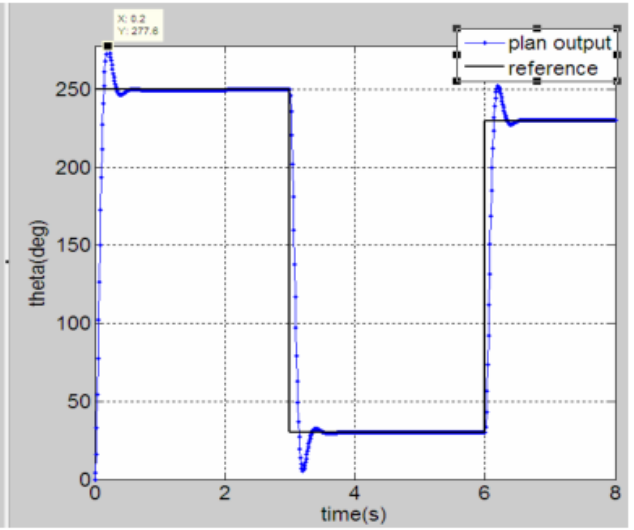


(الف) کنترلر NARMA-L2

(ب) کنترلر PID



(الف) کنترلر NARMA-L2



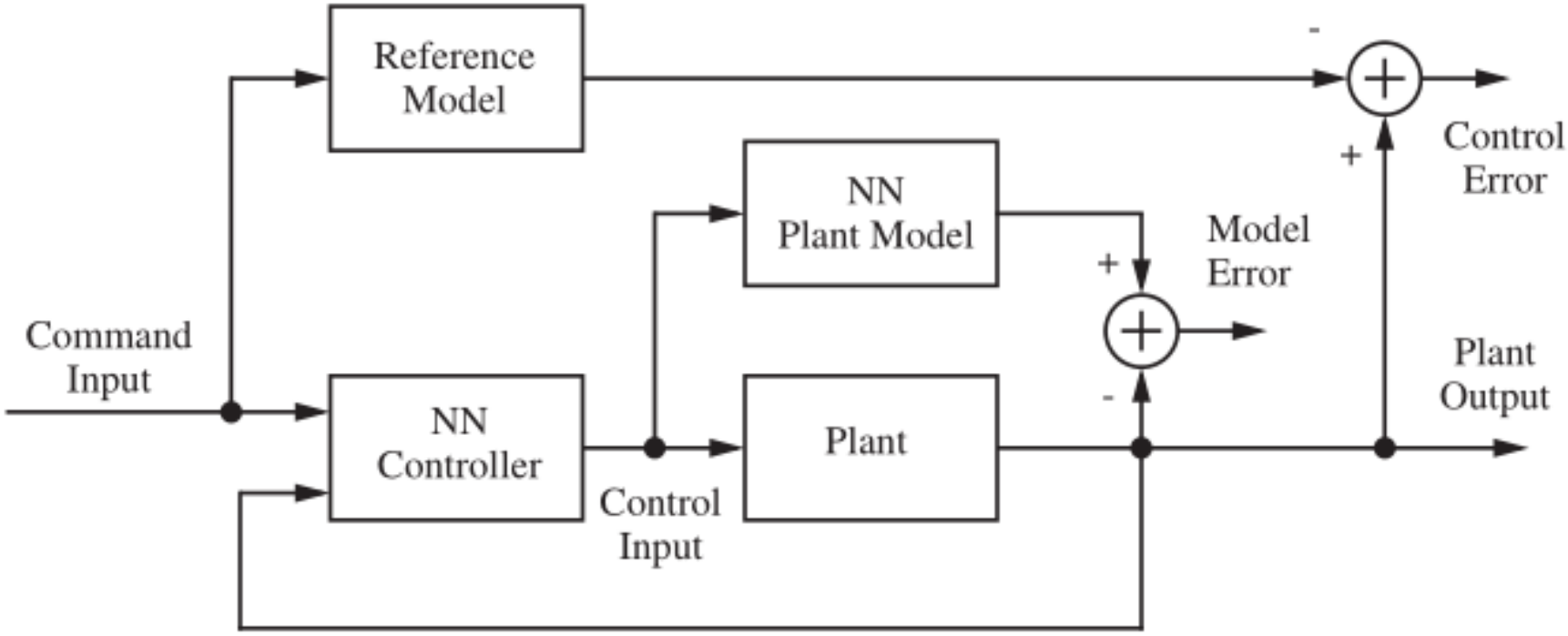
(ب) کنترلر PID

نتایج کنترل وضعیت موتور DC در حالت Tm3 = 30% Tm

نتایج کنترل وضعیت در شرایط ثابت بودن پارامترهای موتور DC

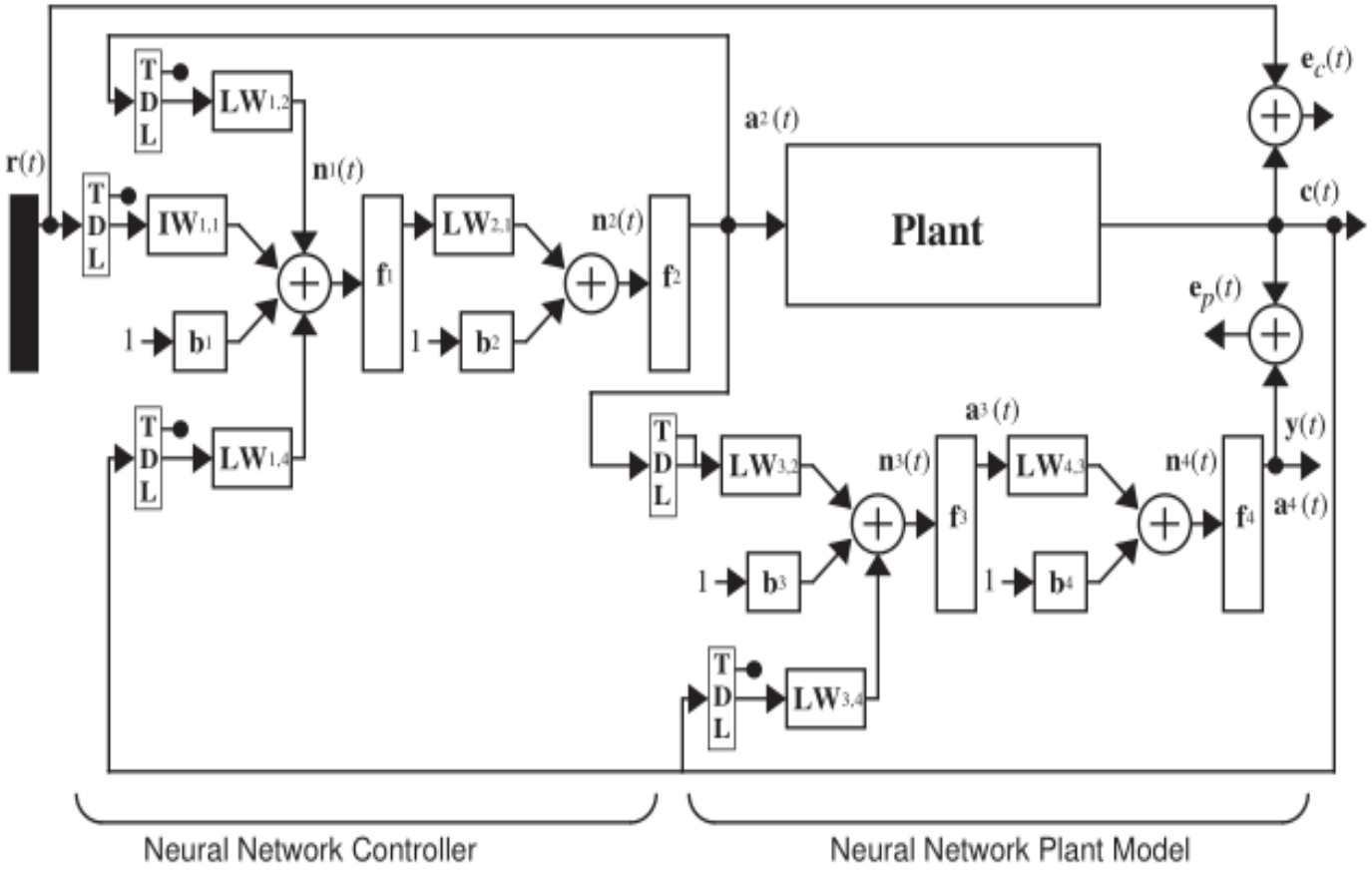
5. Model Reference Control

- معماری شبکه عصبی کنترلر مدل مرجع از دو شبکه عصبی تشکیل شده است.**یک شبکه برای کنترل کننده** و **یک شبکه برای شناسایی plant**.
- ابتدا مدل تخمینی plant با استفاده از ورودی و خروجی plant تعیین می شود و سپس کنترلر تا جایی که خروجی plant با خروجی مدل مرجع یکسان شود آموزش میبینند.



Model Reference Control Architecture

معماری شبکه عصبی مورد استفاده در نروکنترلر مدل مرجع



Detailed Model Reference Control Structure

- هر شبکه دارای دو لایه می باشد .تعداد نورون های مجود در لایه مخفی قابل تعیین توسط کاربر است.

- ورودی های کنترلر به سه دسته تقسیم می شوند:
  - ورودی های تاخیری مرجع
  - خروجی های تاخیری کنترل کننده
  - خروجی های تاخیری دستگاه

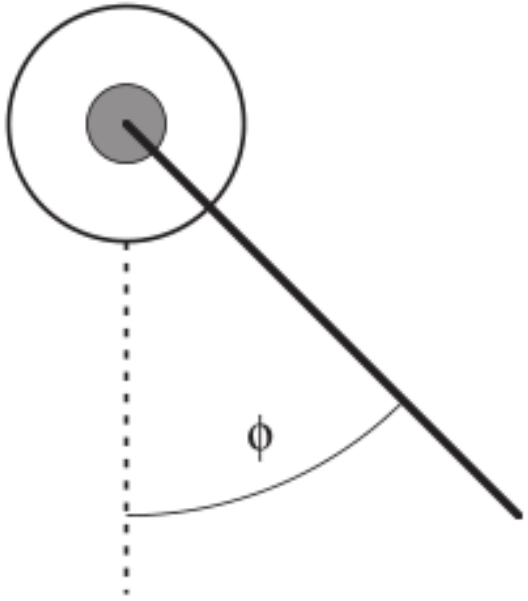
- برای هر یک از این ورودی ها،تعداد تاخیرها قابل تعیین می باشد
عموما تعداد تاخیرها با درجه دستگاه افزایش می یابند.

- ورودی های شبکه عصبی تخمین زننده plant
  - خروجی های تاخیری کنترل کننده
  - خروجی های تاخیری plant

مثالی از کنترل کاربرد نروکنترلر مدل مرجع درSIMULINK

در این قسمت با ارائه یک مثال نحوه آموزش شبکه عصبی برای این نوع کنترل کننده را مورد بررسی قرار میدهیم.

هدف : کنترل حرکت یک بازوی ساده ربات می باشد.همانطوری که در شکل مشخص است این بازو تنها دارای یک نقطه اتصال است:



Single-link robot arm

معادله دینامیکی حرکت که به صورت غیر خطی است:

$$\frac{d^2\phi}{dt^2} = -10\sin\phi - 2\frac{d\phi}{dt} + u$$

کهϕزاویه بازو وuگشتاور بازو می باشد که توسط یک موتور DC تامین می گردد

هدف از کنترل

- هدف از کنترل آموزش کنترلر برای تعقیب مدل مرجع زیر می باشد:

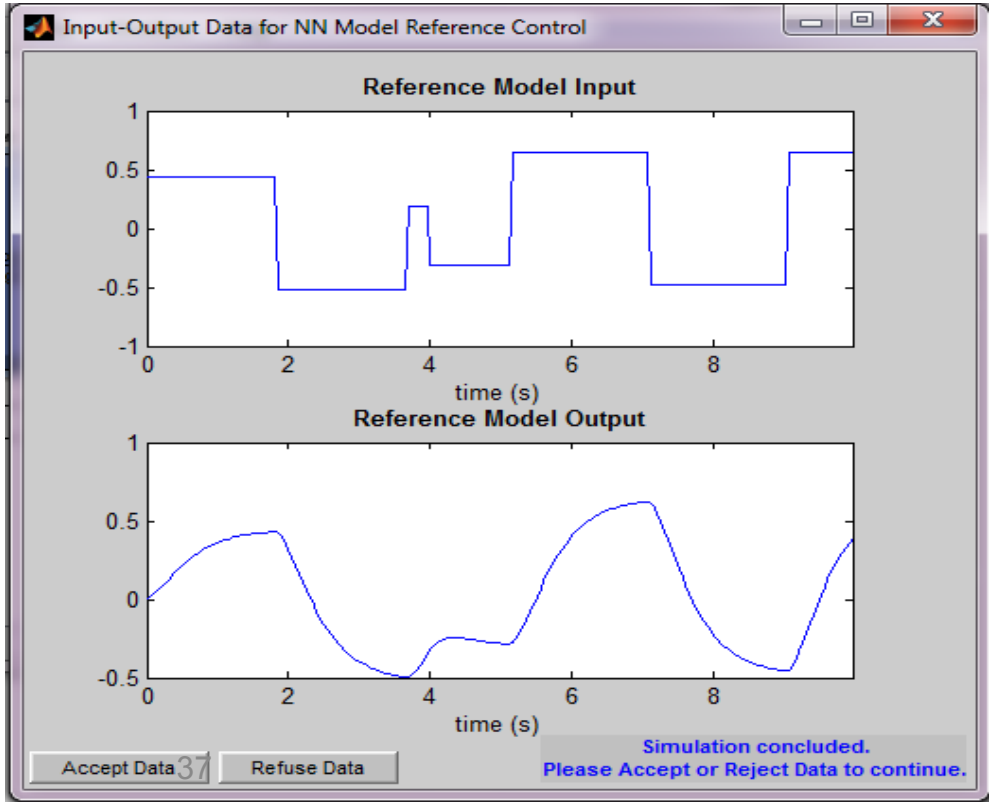
$$\frac{d^2 y_r}{dt^2} = -9y_r - 6\frac{dy_r}{dt} + 9r$$

- که yr خروجی مدل مرجع و r سیگنال مرجع ورودی می باشد.

که اگر ورودی تصادفی به این سیستم بدهیم چنین خروجی خواهد داد

در واقع شبکه عصبی منحنی خروجی با این ورودی

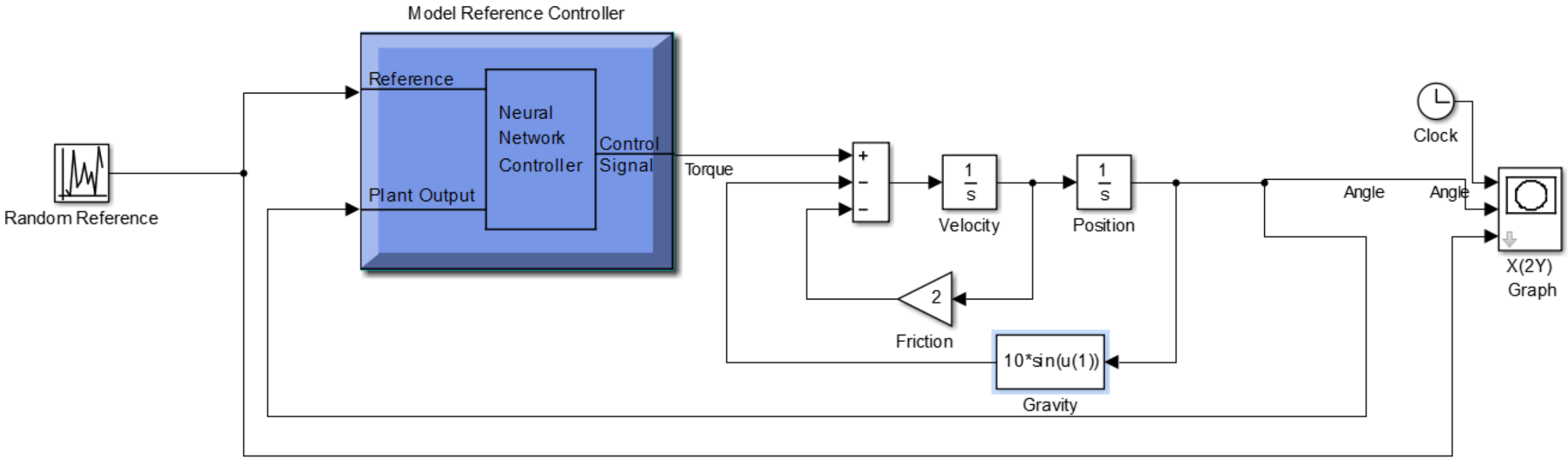
را تقریب می زند.



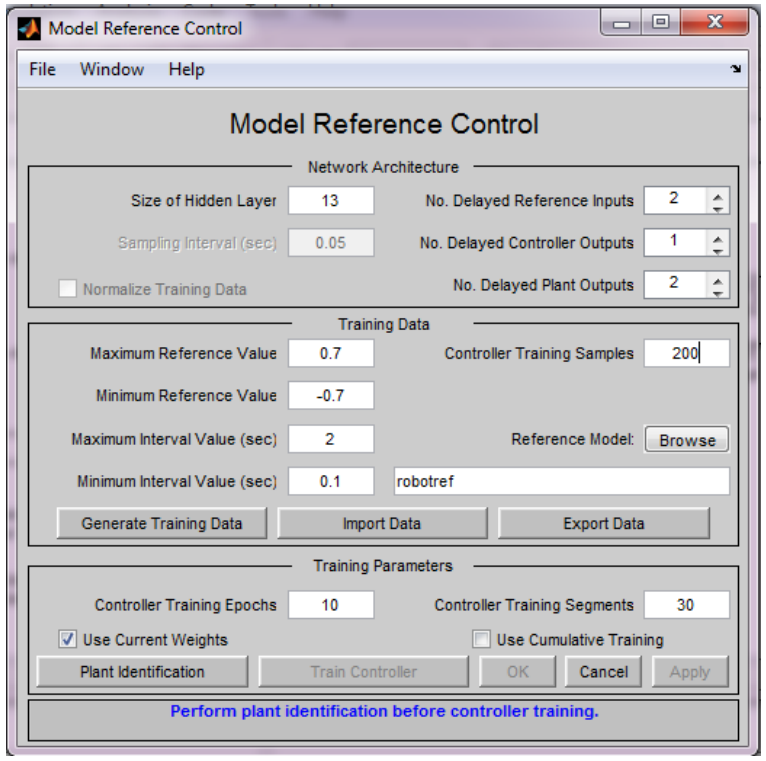

```
T =

          9
  -------------
  s^2 + 6 s + 9
```

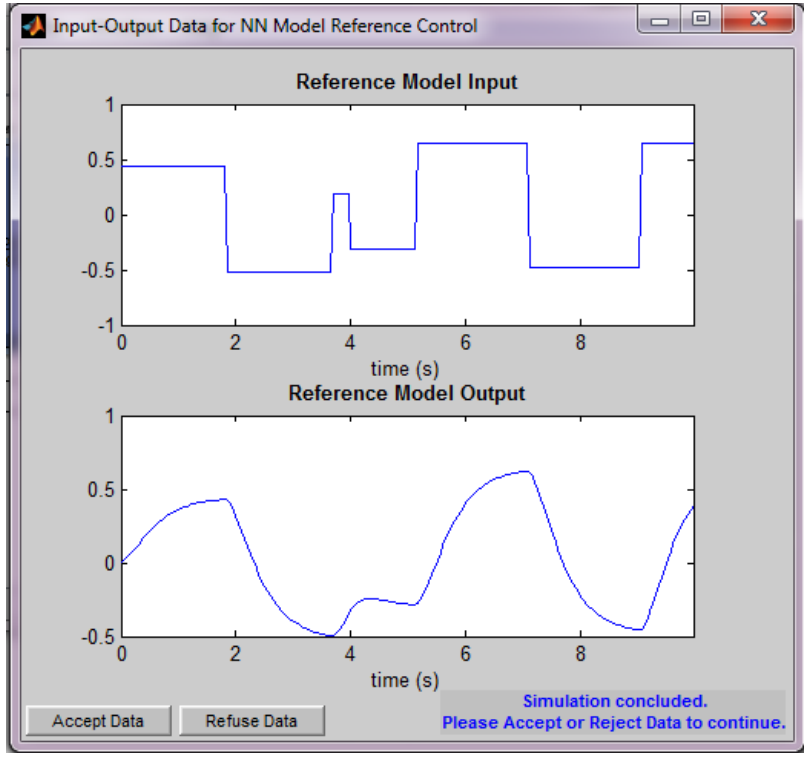
ورودی ها و خروجی ها و معماری شبکه عصبی

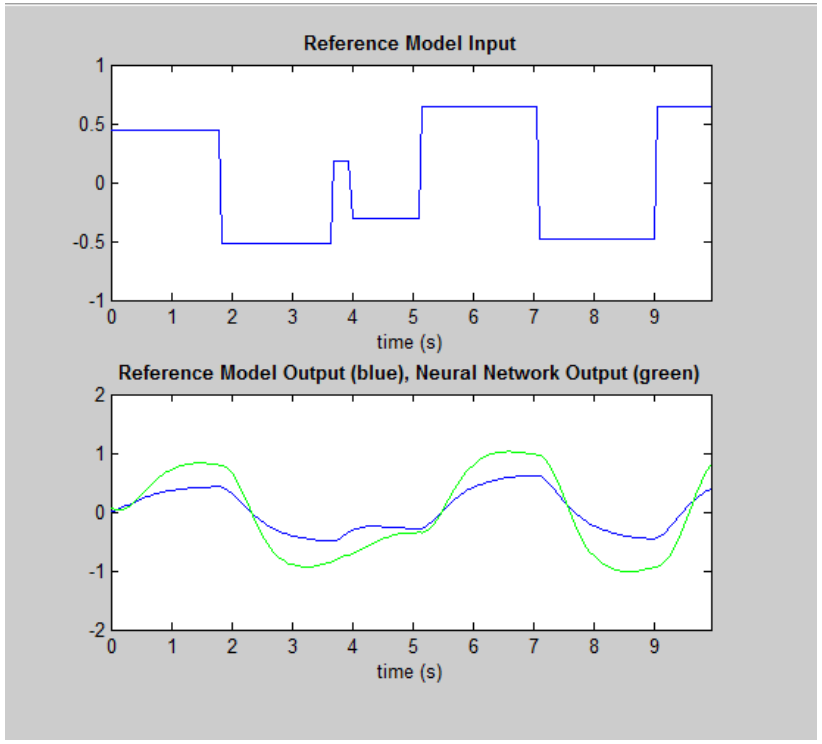
- در این مثال از یک شبکه عصبی با معماری 1-13-5 استفاده می شود.
- ورودی های کنترل کننده شامل دو ورودی تاخیری مرجع،دو خروجی تاخیری plant و یک خروجی تاخیری کنترل کننده هستند .
- فاصله زمانی نمونه برداری (sample time) 0.05 در نظر گرفته می شود.

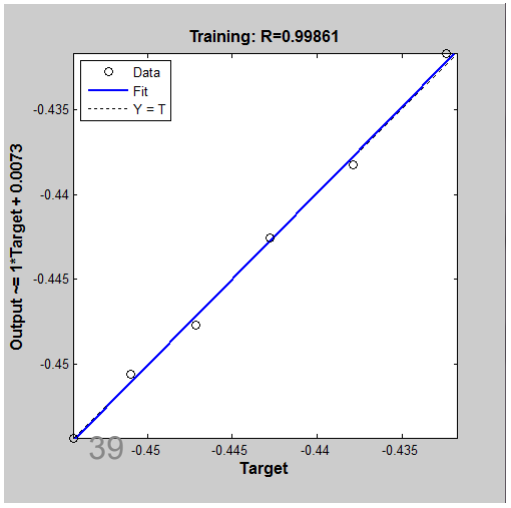


آموزش شبکه عصبی کنترلر



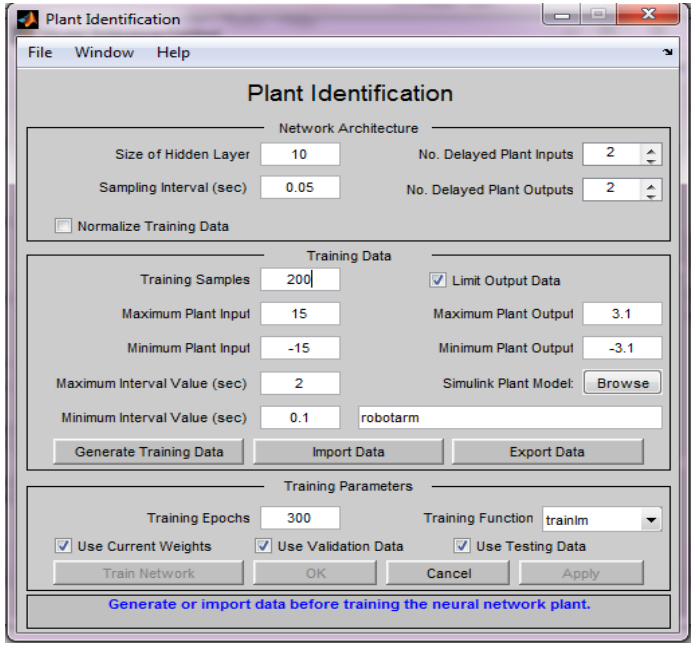


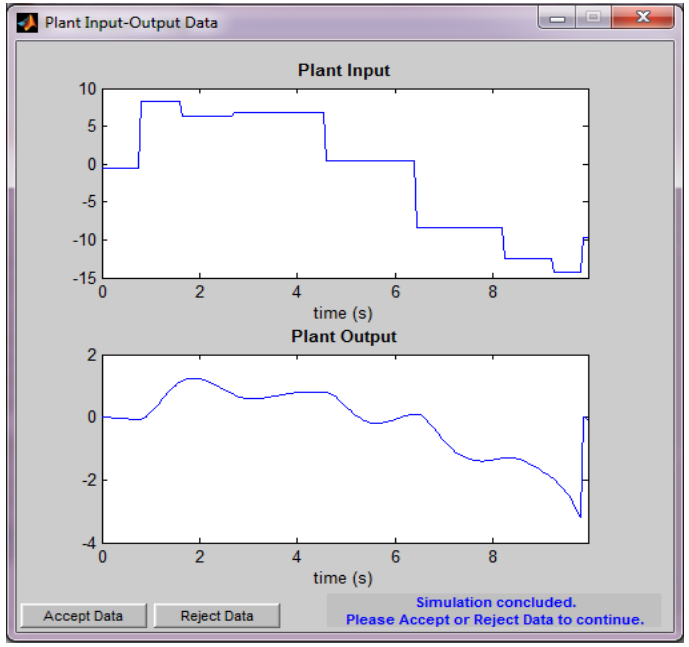


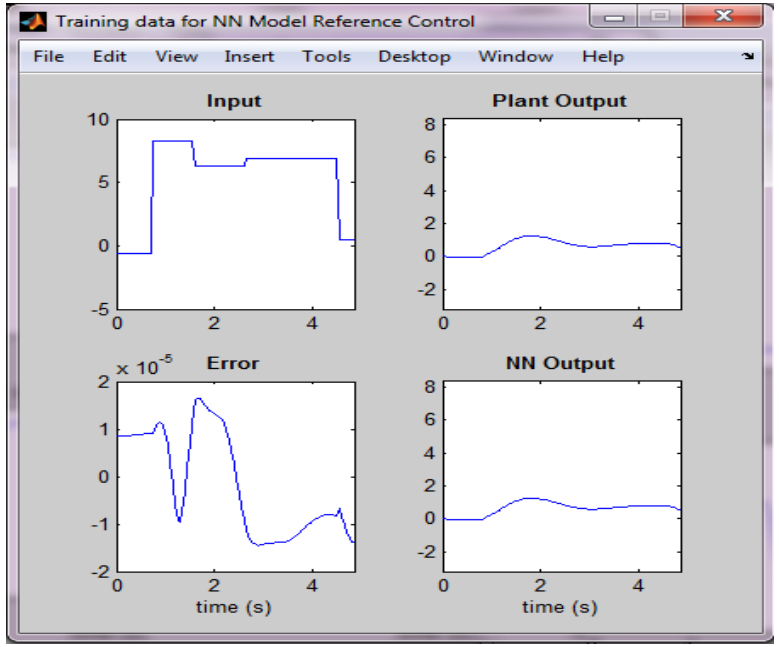


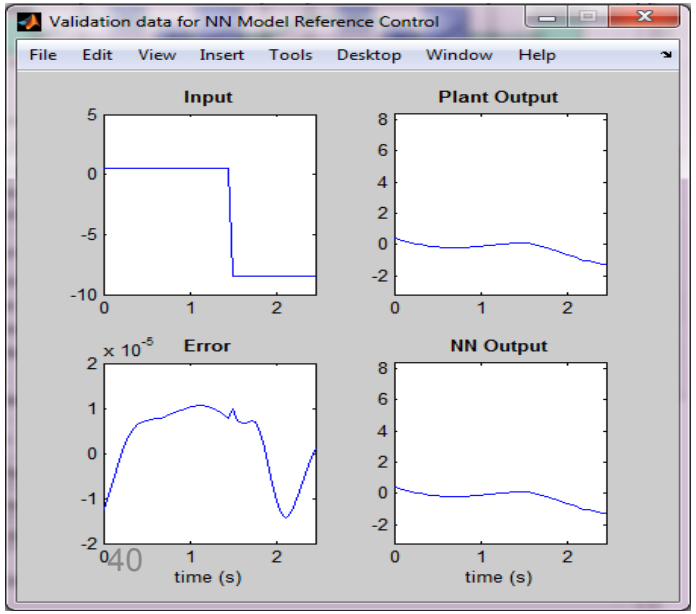
شبکه برای نتیجه خوب باید برای ۶۰۰۰ داده ورودی تولید شده آموزش ببیند که اینجا برای نشان دادن بهتر نمودار های ورودی و خروجی فقط ۲۰۰ داده تولید شده است.

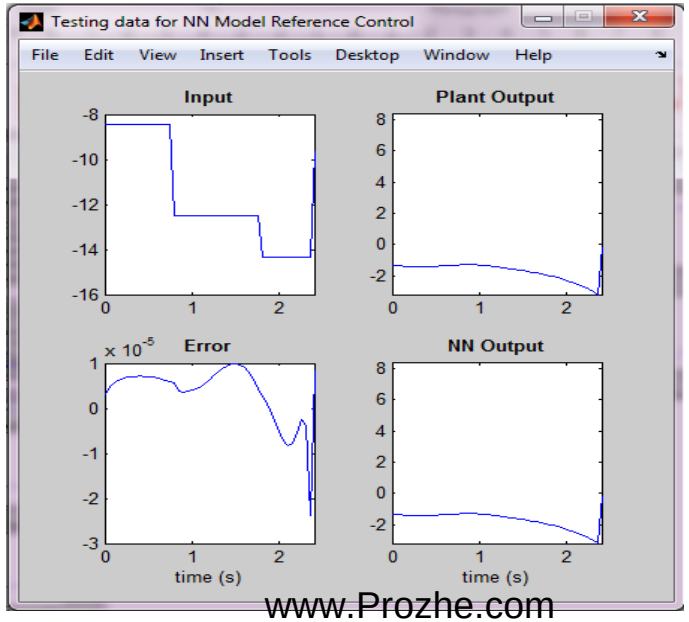
آموزش شبکه عصبی شناسایی plant

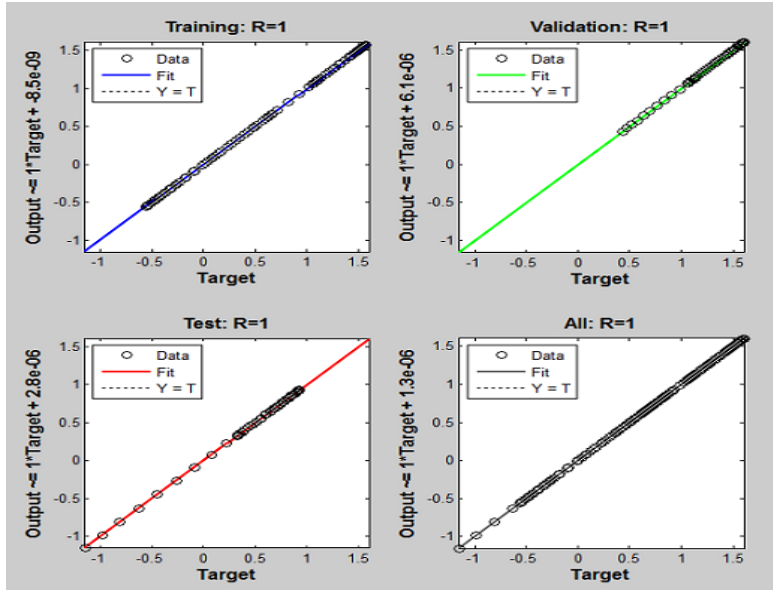






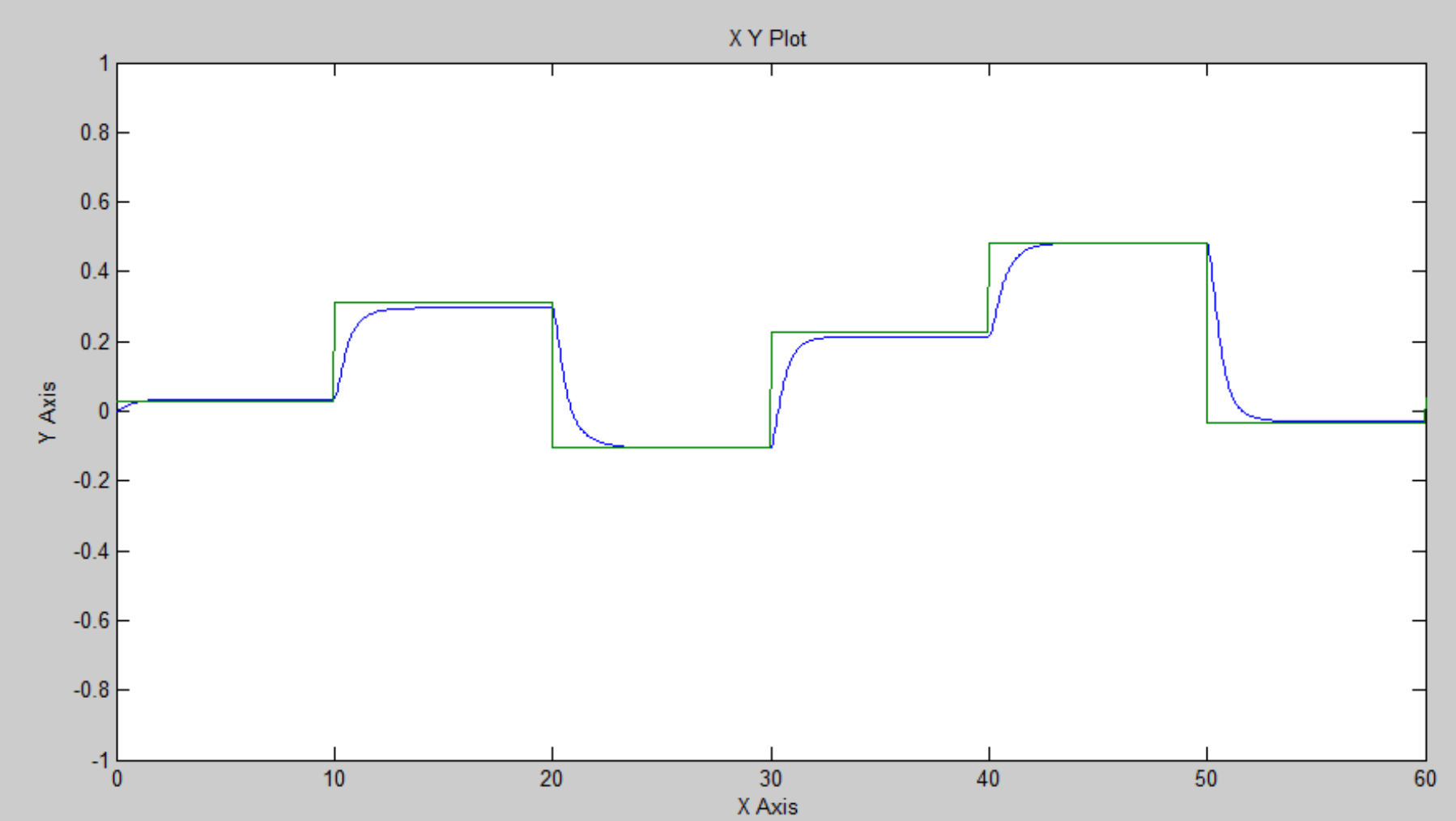




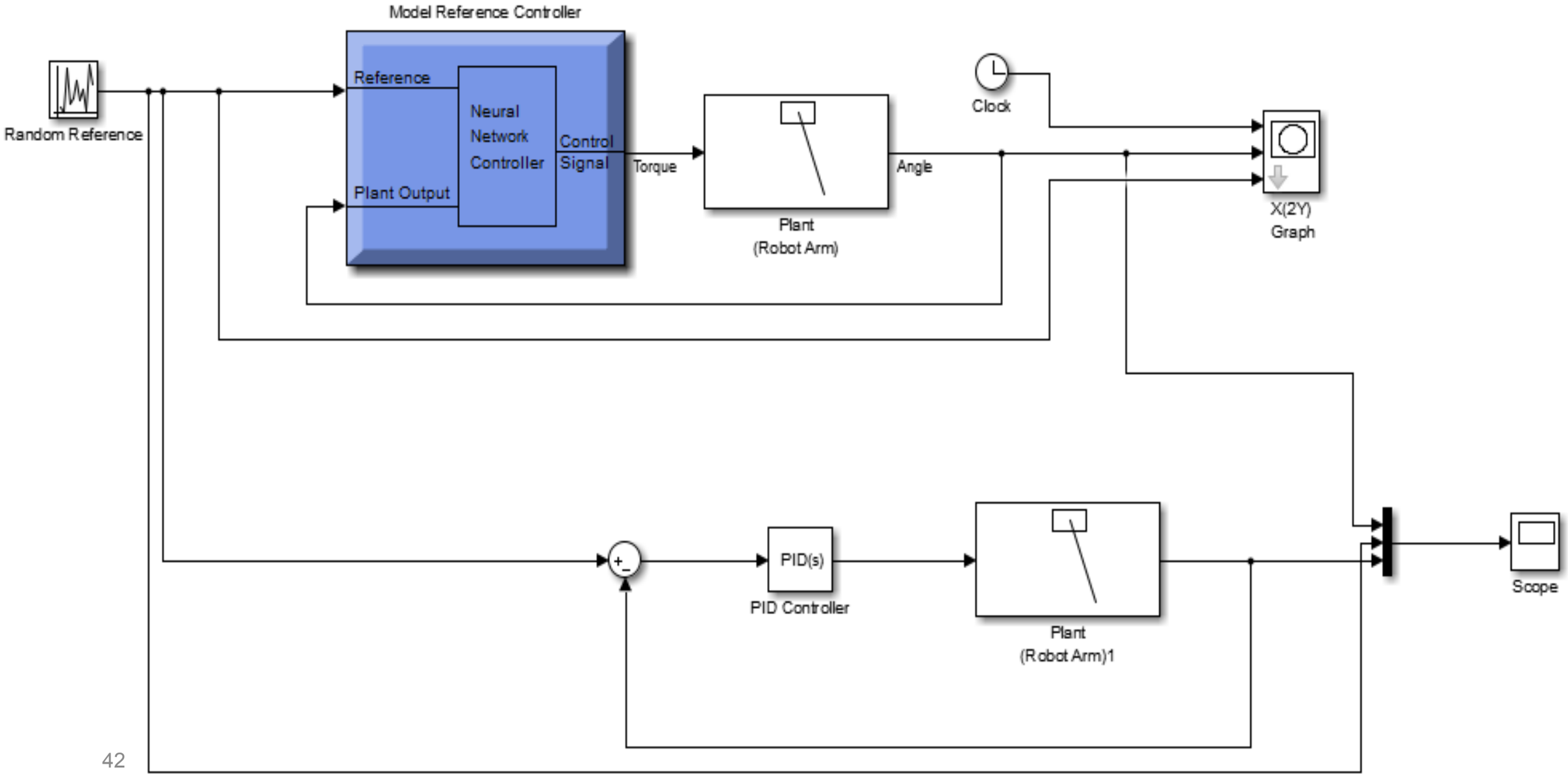


دادن یک ورودی تصادفی به سیستم وبررسی رفتار سیستم

نتیجه رفتار مناسب شبکه نسبت به ورودی تصادفی به سیستم است



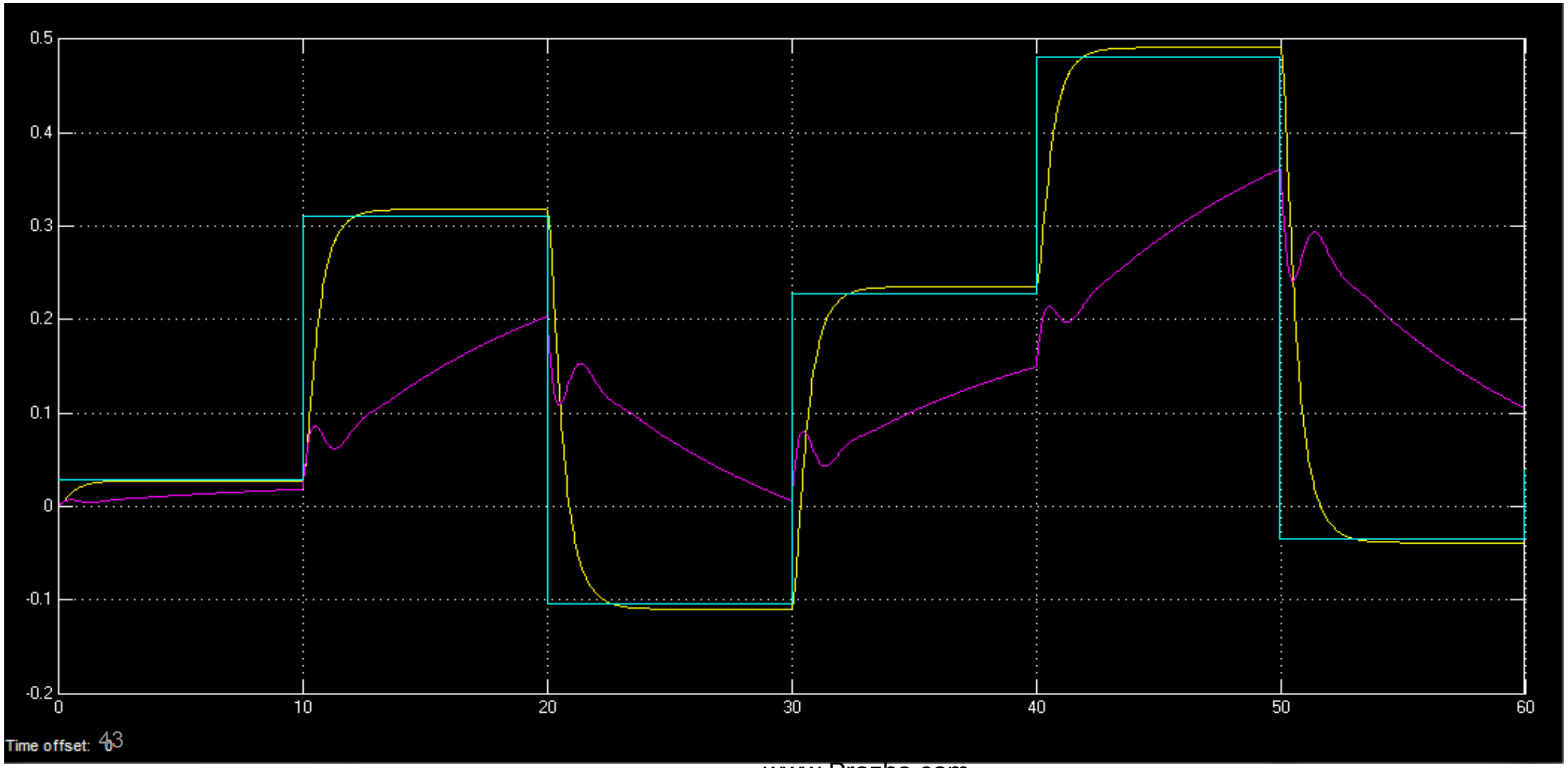
شبیه سازی سیستم با کنترلر PID و نروکنترلرمدل مرجع



مقایسه رفتار سیستم با وجود کنترلرPID و کنترلر شبکه عصبی مدل مرجع

رنگ زرد : پاسخ سیستم با کنترلر شبکه عصبی مدل مرجع

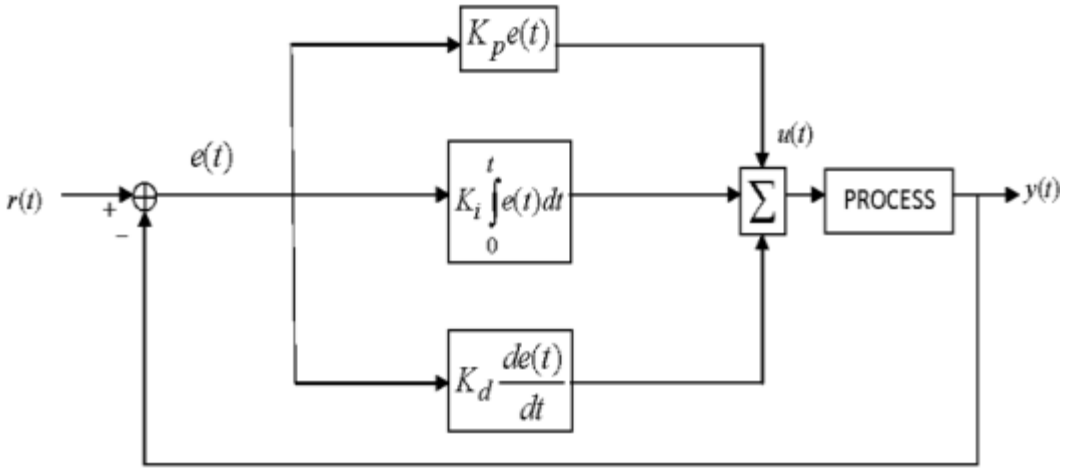
رنگ صورتی : پاسخ سیستم با کنترلر PID



تنظیم پارامترهای کنترلر PID با استفاده از شبکه عصبی

- یکی از پر کاربردترین روش های کنترلی ،کنترل کننده PID است، این کنترل کننده دارای قابلیت هایی از جمله در کنترل سیستم هایی که دارای مدل دقیق نمی باشند، و دارای عدم قطعیت می باشند.
- از ویژگی های کنترل کننده PID می توان سادگی و پایداری را نام برد.
- ساختار کنترل کننده PID از سه بخش با پارامترهای تناسبی،انتگرالی و مشتق گیر تشکیل شده است. عملکرد صحیح کنترل کننده منوط به تنظیم درست این ضرائب می باشد.
- کنترل کننده PID یک مکانیسم با فیبدک حلقه بسته است که بطور گسترده در صنعت مورد استفاده قرار می گیرد.کنترلر PID مقدار خطای بین خروجی فرآیند و مقدار ورودی مطلوب را دریافت می کند .
- هدف کنترل کننده تلاش برای به حداقل رساندن خطا با تنظیم ورودی های کنترل فرآیند است.

$$u(t) = k_p e(t) + k_i \int e(k)dt + k_d \left(\frac{de(t)}{dt} \right)$$



کنترل PID یک فرآیند

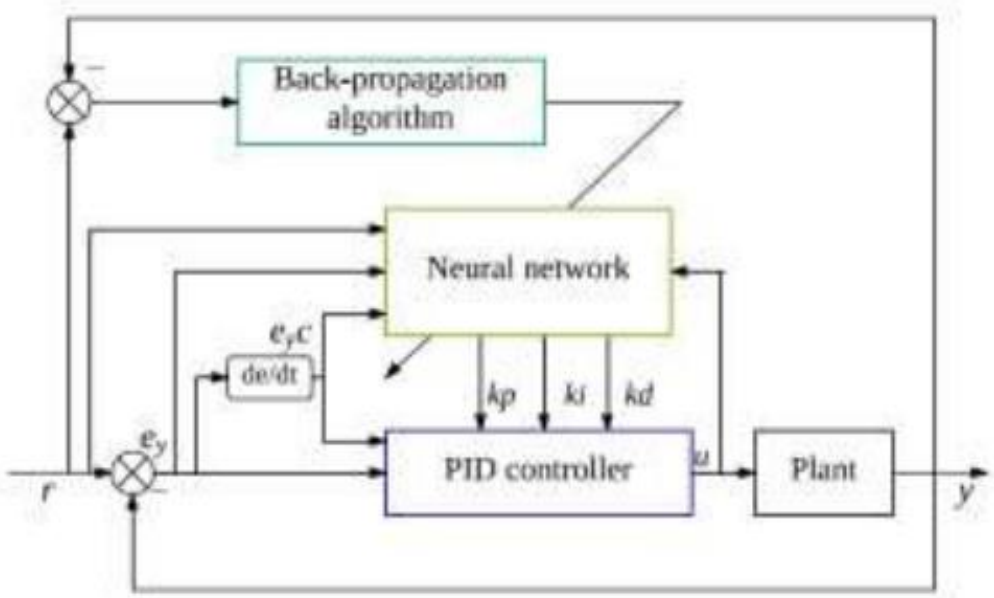
انواع روش های کلاسیک و آفلاین تعیین پارامترهای کنترلر PID:

- Ziegler and Nichols و روش های توسعه ای بر پایه این روش
- Åström
- Cohen

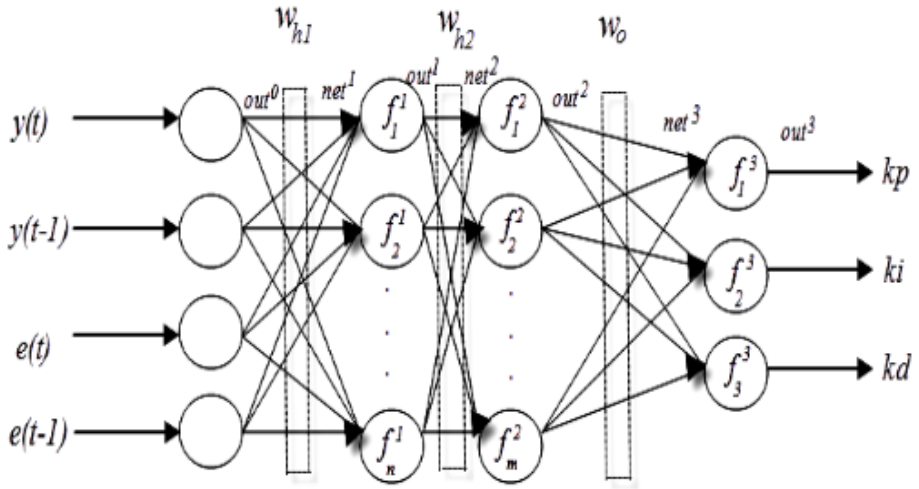
PID خود تنظیم

- برای آپدیت online پارامترهای کنترلرPID می بایست حلقه تطبیقی را به ساختارکنترل اضافه کرد که این امر موضوع را کمی پیچیده می کند.
- از آنجا که تنظیم آفلاین پارامترها زمان و هزینه دارد، استفاده از حلقه تطبیقی مورد استقبال قرار گرفت.
- با تغییر در دینامیک های سیستم(تغییر پرامترهای سیستم) ،تنظیم کنترل کننده اهمیت بیشتری پیدا می کند.
- پس از آنکه PID های خود تنظیم مورد استقبال صنعت قرار گرفتند،محققین بر آن شده اند با استفاده از تئوری های مختلف کنترل،الگوریتم های مختلفی برای خود تنظیمی ارائه دهند که از آن جمله به کنترل تطبیقی می توان اشاره کرد.
- از آنجایی که اغلب سیستم ها متغیر با زمان بوده و عملکرد انها تحت تاثیر نویز است لذا بررسی پایداری سیستم ها با این روش ها بسادگی نیست.
- از آنجایی که شبکه های عصبی قابلیت زیادی برای رفع اینگونه مشکلات دارند،لذا تحقیقات وسیعی در مورد ساختارهای مختلف آن جهت تنظیم پارامترهای PID صورت گرفته است.

استفاده از شبکه عصبی MLP برای خود تنظیمی کنترلر PID

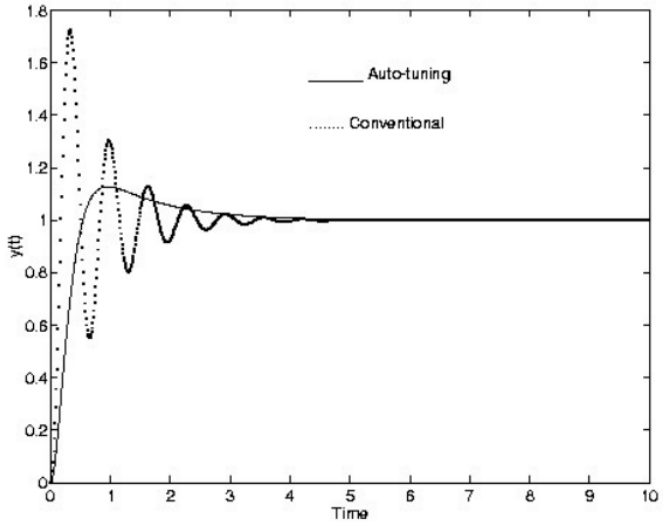


BPNN based PID control scheme



شبکه عصبی MLP با دو لایه فعالساز میانی برای تنظیم

کنترل‌کننده PID



Close loop responses with neural

منابع و مراجع

- جزوه کنترل مدرن
  - دکتر علی غفاری
- کتاب شبکه های عصبی در MATLAB
  - سید مصطفی کیا
- مقاله AN INTRODUCTION TO THE USE OF NEURAL NETWORKS IN CONTROL SYSTEMS
  - MARTIN T. HAGAN 1 , HOWARD B. DEMUTH 2 AND ORLANDO DE JESÚS 1
- مقاله Neural Networks for Control
  - Martin T. Hagan
- مقاله کنترل وضعیت موتورDC تحریک جداگانه با کمک کنترلر شبکه عصبی NARMA-L2
  - جهان جمشیدی-رضا حق مرام
- سمینار ارائه شده کاربرد شبکه های عصبی در کنترل پیش بین
  - حمیدرضا ایرانمنش
- طراحی و شبیه سازی کنترل کننده PID خودتنظیم بر پایه شبکه های عصبی MLP و پویا
  - مهدی علی بخشی
- Neural Network-based Auto-Tuning for PID Controllers
  - FRANCKLIN RIVAS-ECHEVERRÍA
- تعدادی از مقالات مطالعه شده دیگر

سپاس از توجه شما

هزینه استفاده

پنج صلوات جهت تعجیل فرج صاحب الزمان