

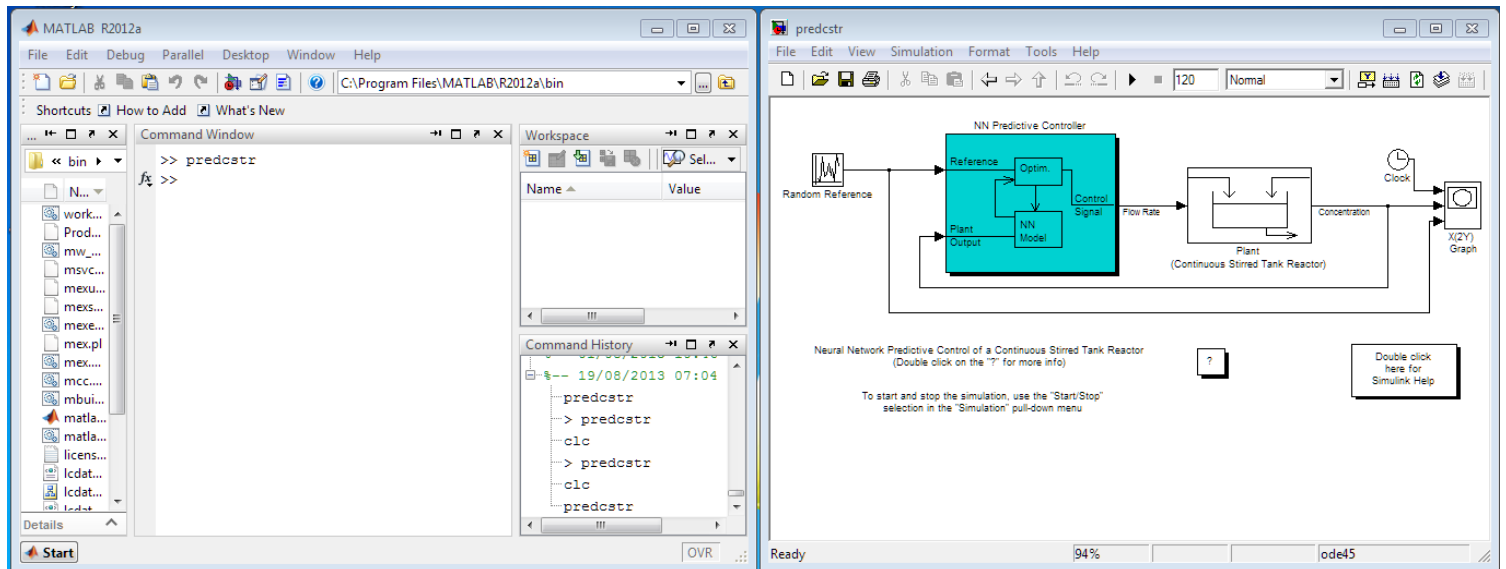
Carlos Antonio Tovar García.

Control Inteligente.- Redes Neuronales.

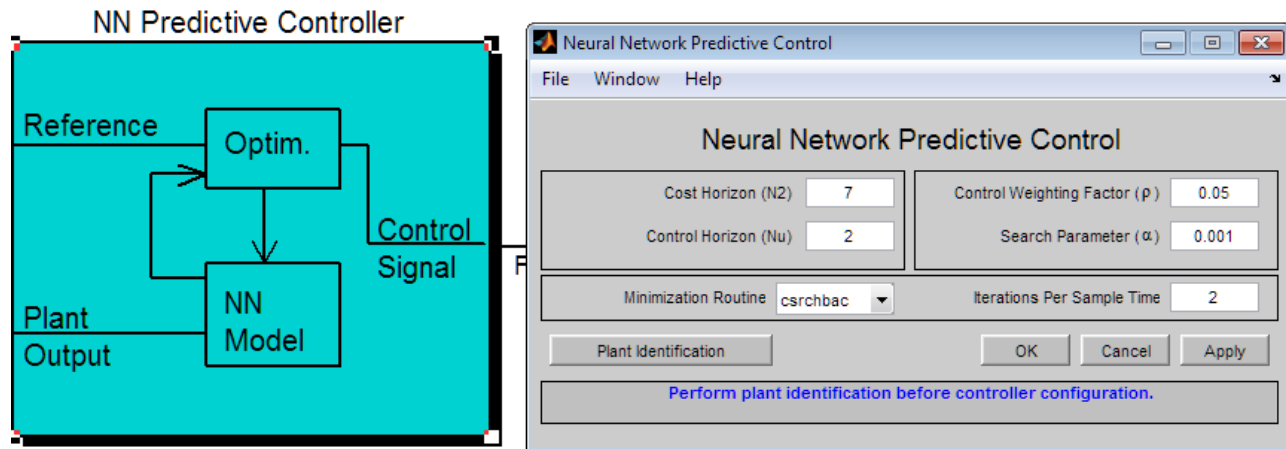
### Entrenamiento de una red neuronal para control predictivo.

En este documento se muestra como se entrena una planta y como se obtienen datos a partir de ese entrenamiento con el fin de poder predecir, regular ó controlar el funcionamiento de dicha planta para obtener un comportamiento deseado.

En primer lugar se teclea en la ventana de comandos de Matlab "predcstr" y a continuación se abre un diagrama de bloques de Simulink, como se muestra a continuación:



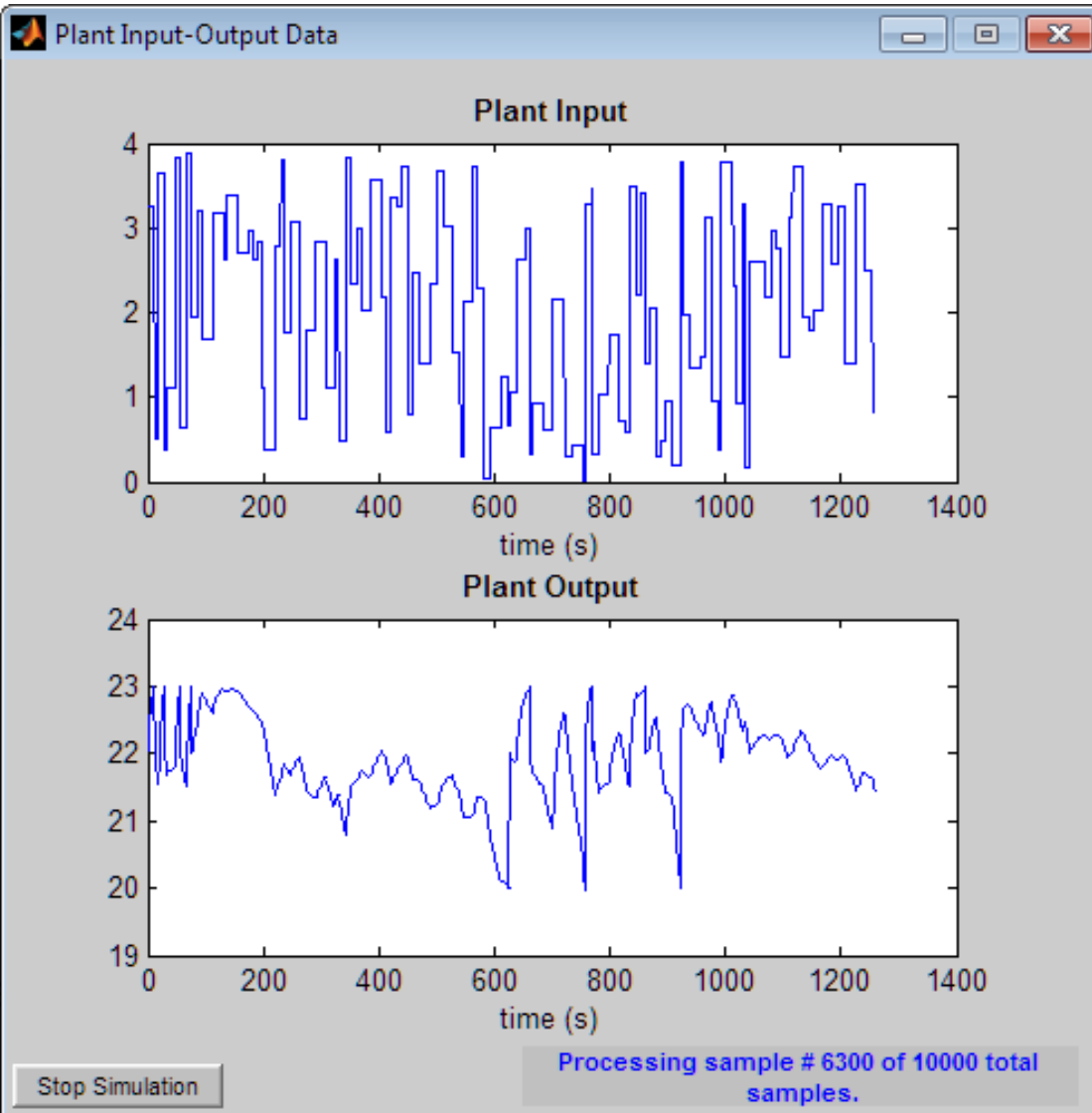
Para entrenar la red neuronal es necesario ajustar los parámetros que pide el control predictivo, en la siguiente imagen se muestran los parámetros que deben de ajustarse, y enseguida se muestran los valores que se propusieron:



En la siguiente tabla se muestran los valores elegidos, con un tiempo de simulación de 120 segundos:

Prueba	Muestras de entrenamiento	Horizonte de predicción	Horizonte de Control	Tamaño de la capa oculta	Integral del error cuadrático
1	10000	9	5	9	45.71
2	2000	7	3	3	58.08
3	8000	7	5	7	38.1

Después de cambiar los parámetros se corre el entrenamiento de la red neuronal, la figura siguiente muestra este proceso:



Se termina de entrenar la neurona:

The screenshot shows the 'Neural Network Training (nntraintool)' window. It features a neural network diagram with an input of 4, two hidden layers (9 and 1 nodes), and an output of 1. The training progress is shown as 58 iterations out of 200, with a performance of 3.34e-06. The window includes buttons for 'Performance', 'Training State', and 'Regression', a 'Plot Interval' slider set to 1 epoch, and a 'Training neural network...' status bar with 'Stop Training' and 'Cancel' buttons.

**Neural Network**

Input: 4

Layer 1: 9 nodes

Layer 2: 1 node

Output: 1

**Algorithms**

Data Division: Index (divideind)  
Training: Levenberg-Marquardt (trainlm)  
Performance: Mean Squared Error (mse)  
Derivative: Default (defaultderiv)

**Progress**

Epoch:	0	58 iterations	200
Time:		0:00:12	
Performance:	509	3.34e-06	0.00
Gradient:	551	0.00397	1.00e-10
Mu:	0.00100	0.0100	1.00e+10
Validation Checks:	0	0	6

**Plots**

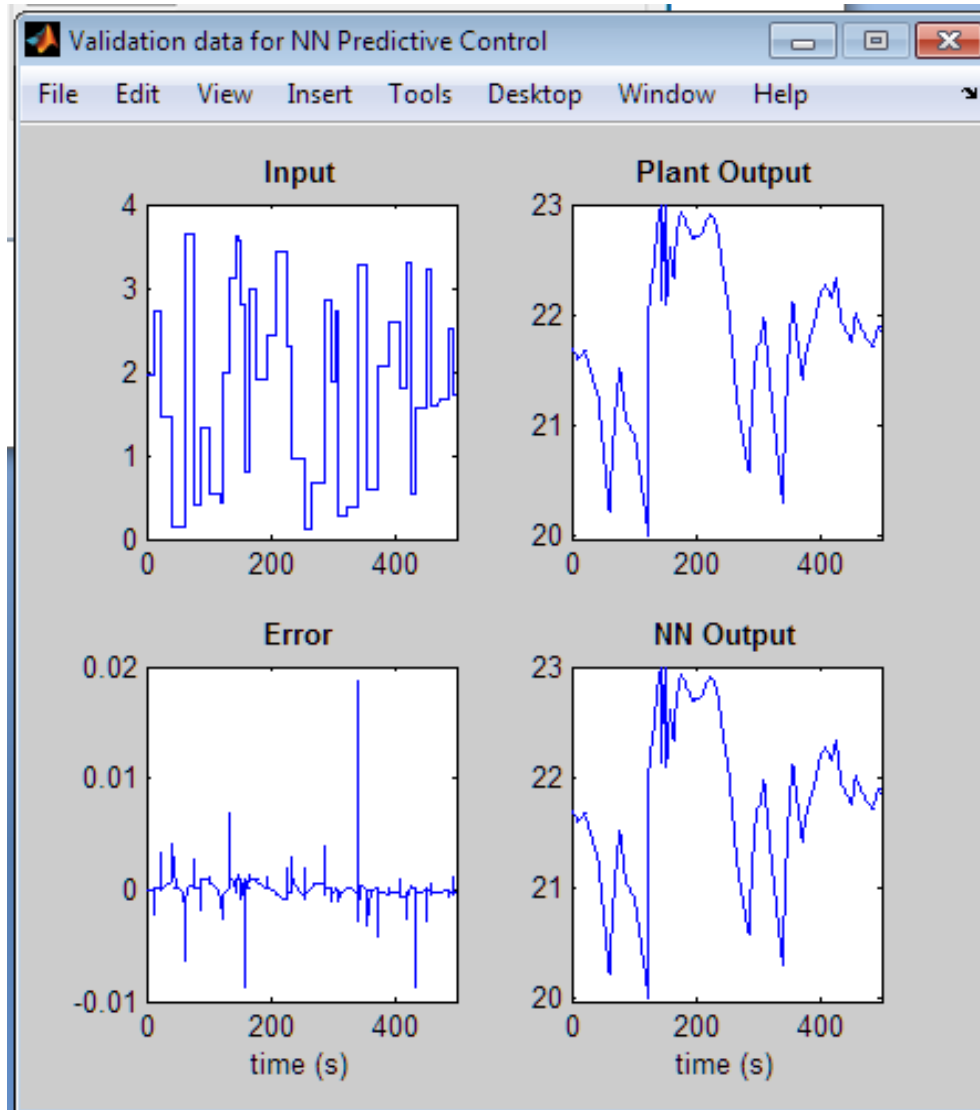
Performance (plotperform)  
Training State (plottrainstate)  
Regression (plotregression)

Plot Interval: 1 epochs

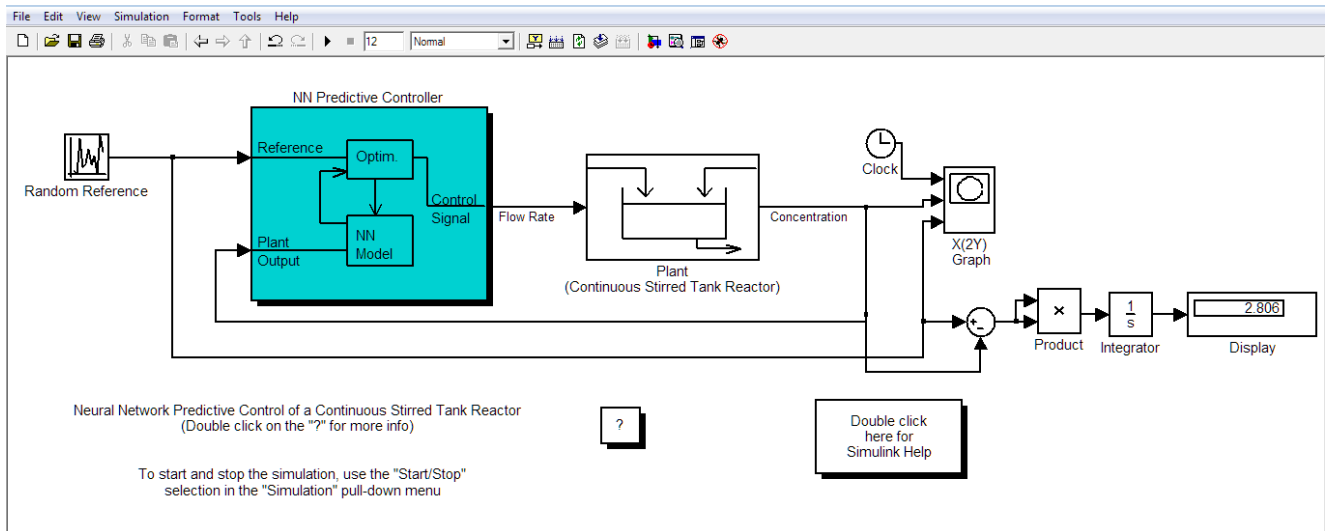
Training neural network...

Stop Training Cancel

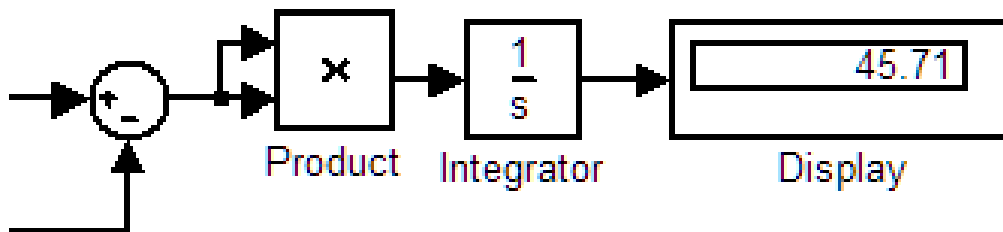
Con la red neuronal entrenada se obtienen los datos de entrada, la salida, el error y la salida de la red neuronal.



Se obtiene la integral del error cuadrático, con el siguiente diagrama de bloques:



Se da a correr el programa y al finalizar se observa el valor obtenido:



El resultado obtenido del XYGraph es el siguiente:

