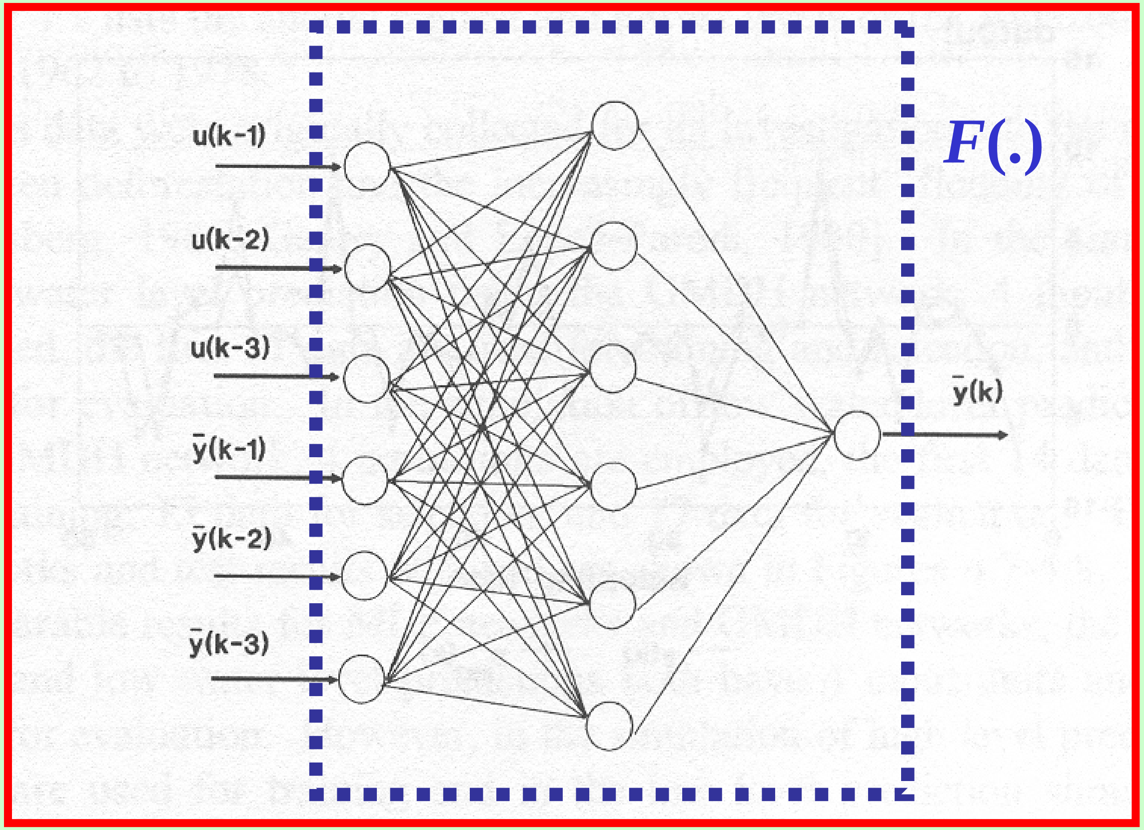
Application Examples

Neural Networks for

Non-linear Identification, Prediction and Control

Nonlinear *Dynamic* System

- Take a static NN
- From static to dynamic NN
- "Quasi-static" NN
- Add inputs, outputs and delayed signals



$$\tilde{y}(k) = F\left(u(k-1), u(k-2), u(k-3), \tilde{y}(k-1), \tilde{y}(k-2), \tilde{y}(k-3)\right)$$

Example of Quasi-static NN
with 3 delayed inputs and outputs

Nonlinear System Identification

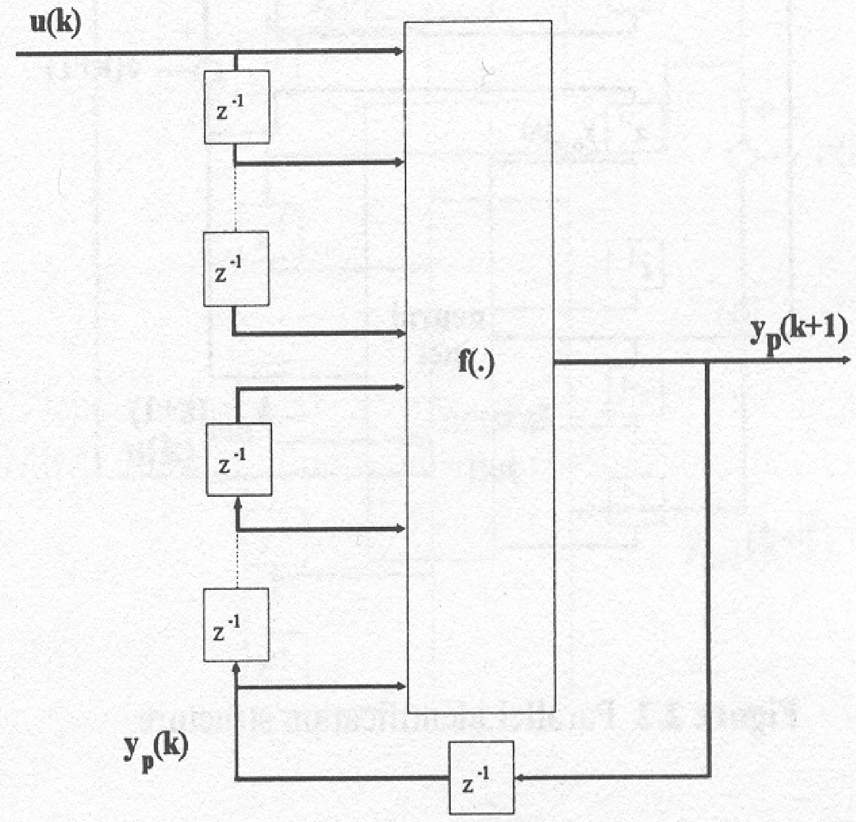


Figure 2.1 Input-output model

- f(.), unknown target function
- Nonlinear dynamic model
- Approximated via a quasi-static NN
- Nonlinear dynamic system identification
- Recall "*linear system identification*"

Silvio Simani

Nonlinear System Identification

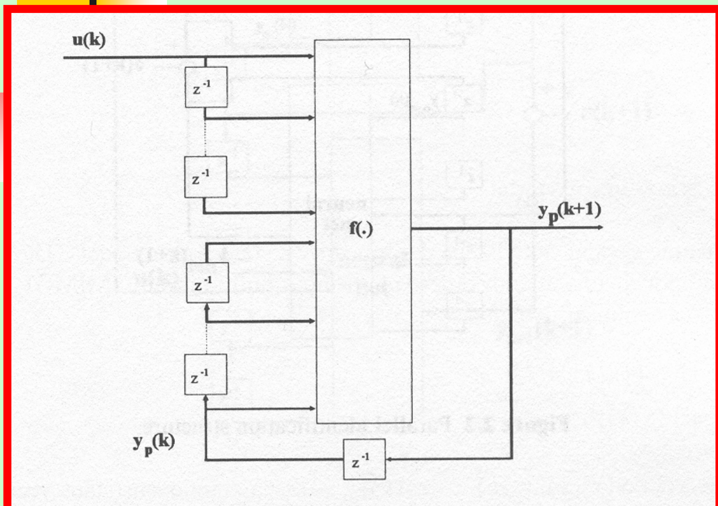
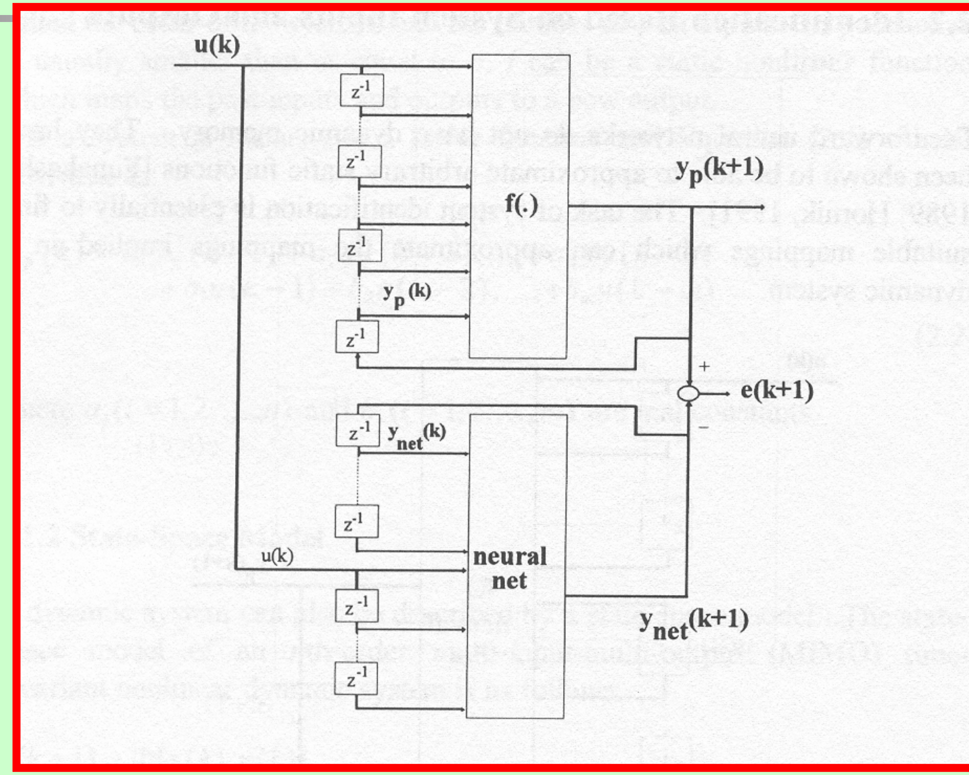
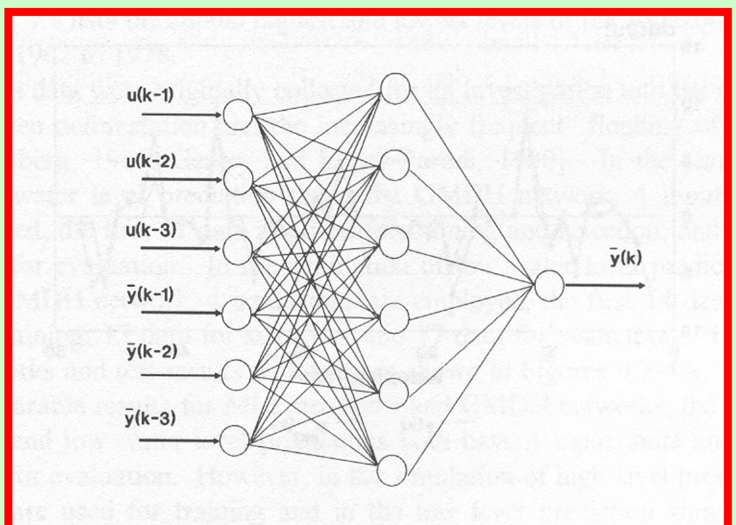


Figure 2.1 Input-output model



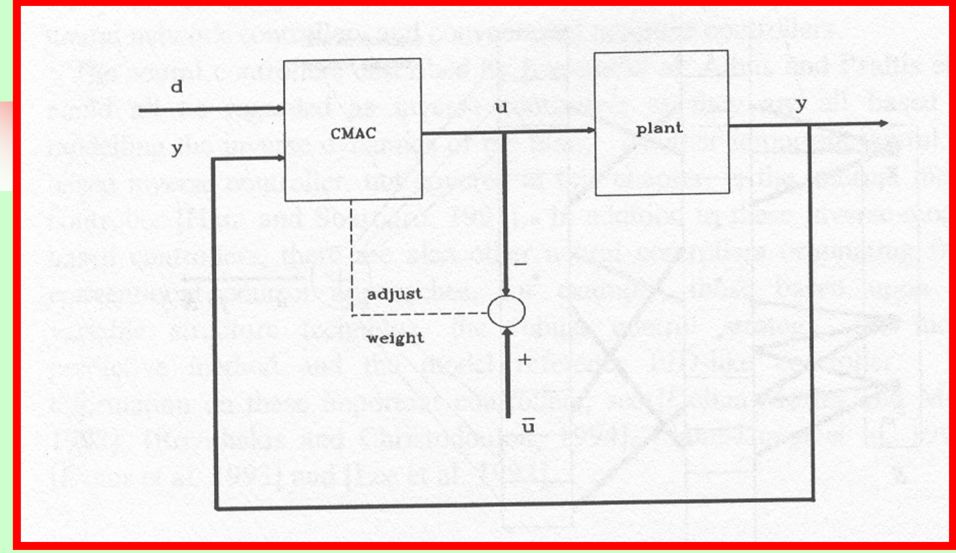


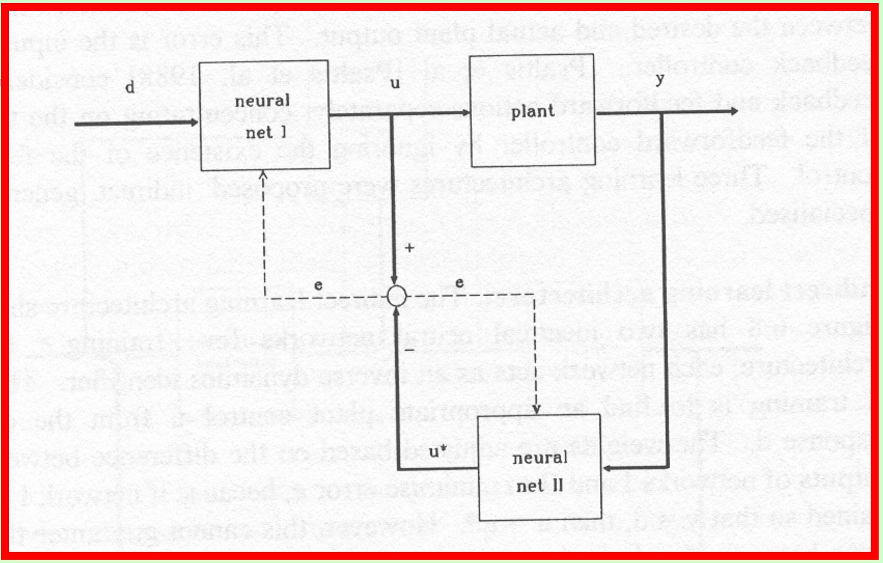
Target function: $y_p(k+1) = f(.)$

Identified function: $y_{NET}(k+1) = F(.)$

Estimation error: $e(k+1)$

Nonlinear System Neural Control





d: *reference/desired response*
y: *system output/desired output*
u: *system input/controller output*
ū: *desired controller input*
u*: *NN output*
e: *controller/network error*

The goal of training is to find an appropriate plant control u from the desired response d. The weights are adjusted based on the difference between the outputs of the networks I & II to minimise e. If network I is trained so that y = d, then u = u*. Networks act as inverse dynamics identifiers.

Neural Networks for Control

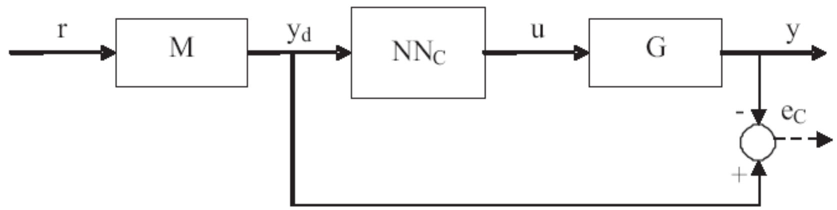


Figure 1: Direct Inverse Control using neural networks

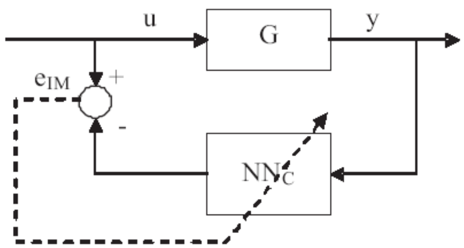


Figure 3: Training the neural network NN_C

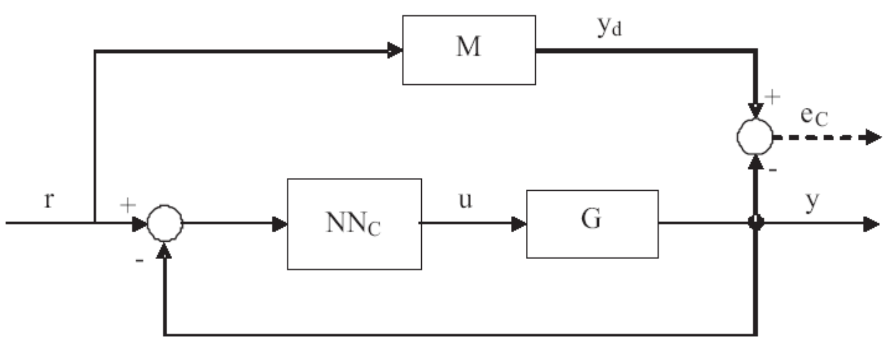


Figure 2: Model Reference Control using neural networks

Figures 1 and 3 Problems.

- Open-loop unstable models
- Disturbances

Neural Model Reference Adaptive Control

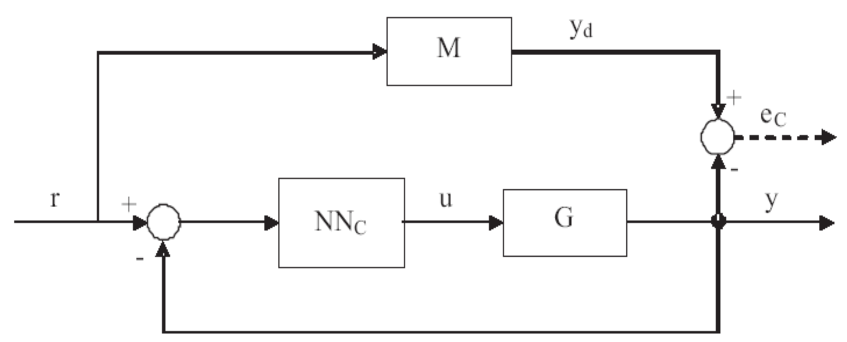


Figure 2: Model Reference Control using neural networks

The signal e_C is used to train or adapt online the weights of the controller NN_C. Two are the approaches used to design a MRAC control for an unknown plant: **Direct and Indirect Control**.

Direct Control: This procedure aims at designing a controller without having a plant model. As the knowledge of the plant is needed in order to train the neural network which corresponds to the controller (*i.e.* NN_C), until present, no method has been proposed to deal with this problem.

Neural Model Reference Adaptive Control

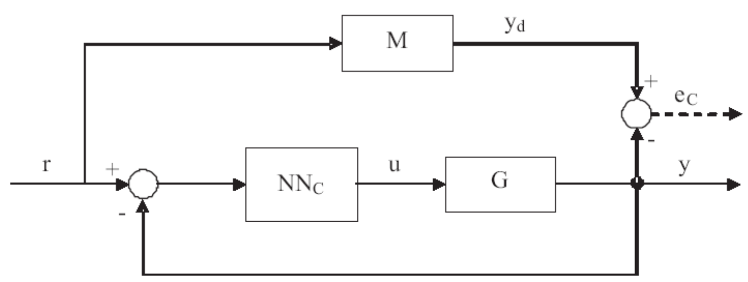


Figure 2: Model Reference Control using neural networks

Indirect Control

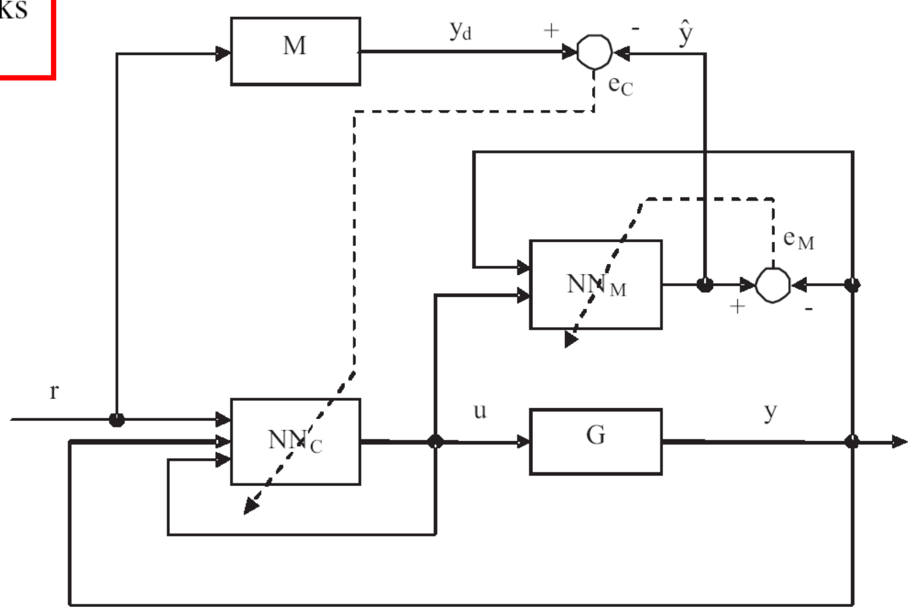


Figure 4: Indirect MRAC

- The signal e_C is used to train or adapt online the weights of the neural controller NN_C.

Indirect Control: NN_M & NN_C

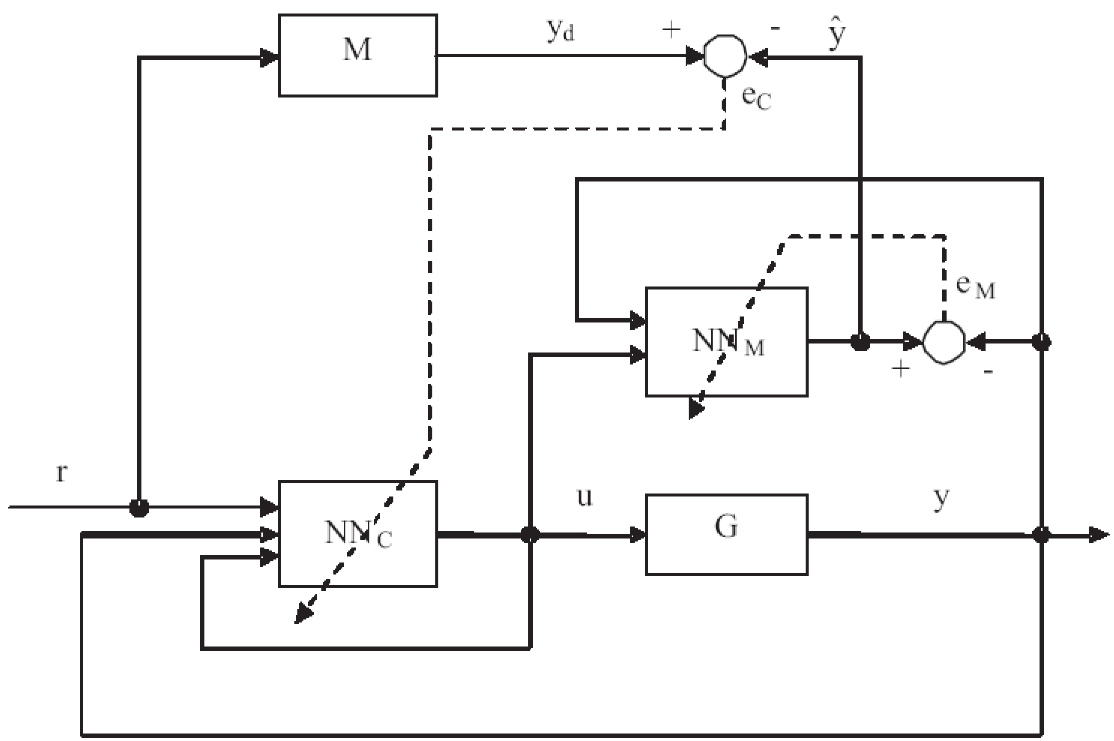


Figure 4: Indirect MRAC

- This approach uses two neural networks: one for modelling the plant dynamics (NN_M), and another one trained to control the real plant (G) so as its behaviour is as close as possible to the reference model (M) via the neural controller (NN_C).

Indirect Control (1)

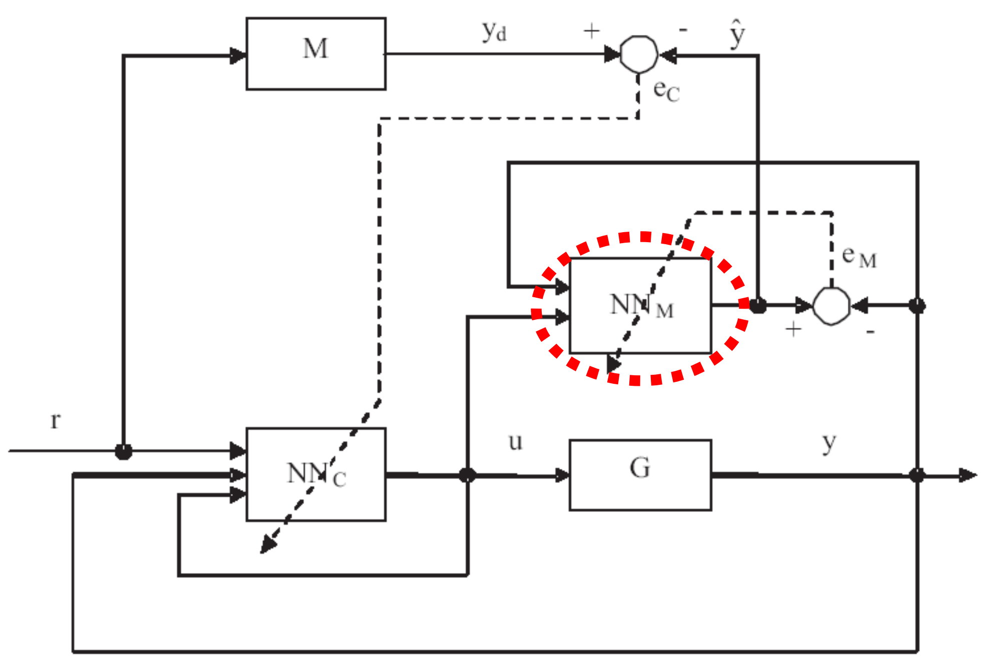


Figure 4: Indirect MRAC

The **neural network NN_M** is trained to approximate the plant **G** input/output relation using the signal e_M. This is usually done offline, using a batch of data gathered from the plant in open loop.

Indirect Control (2)

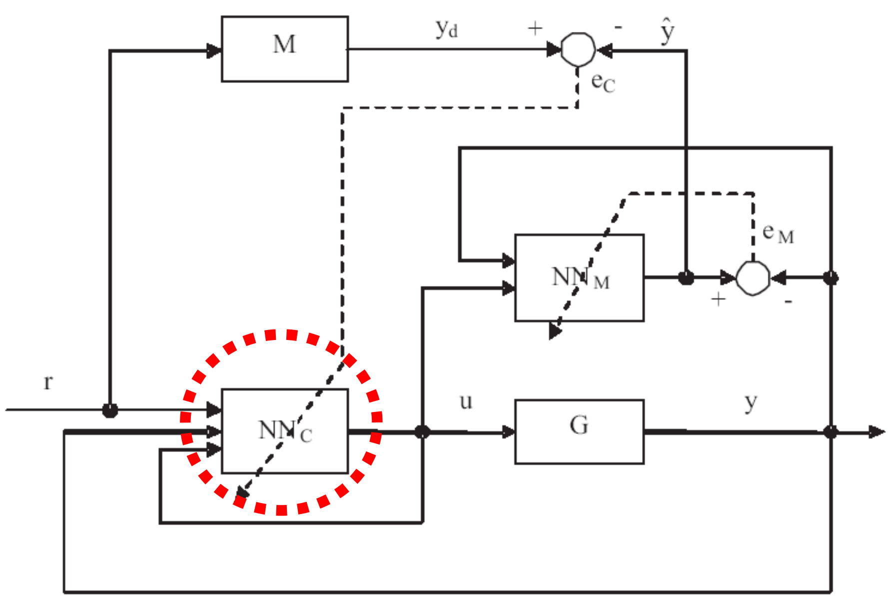


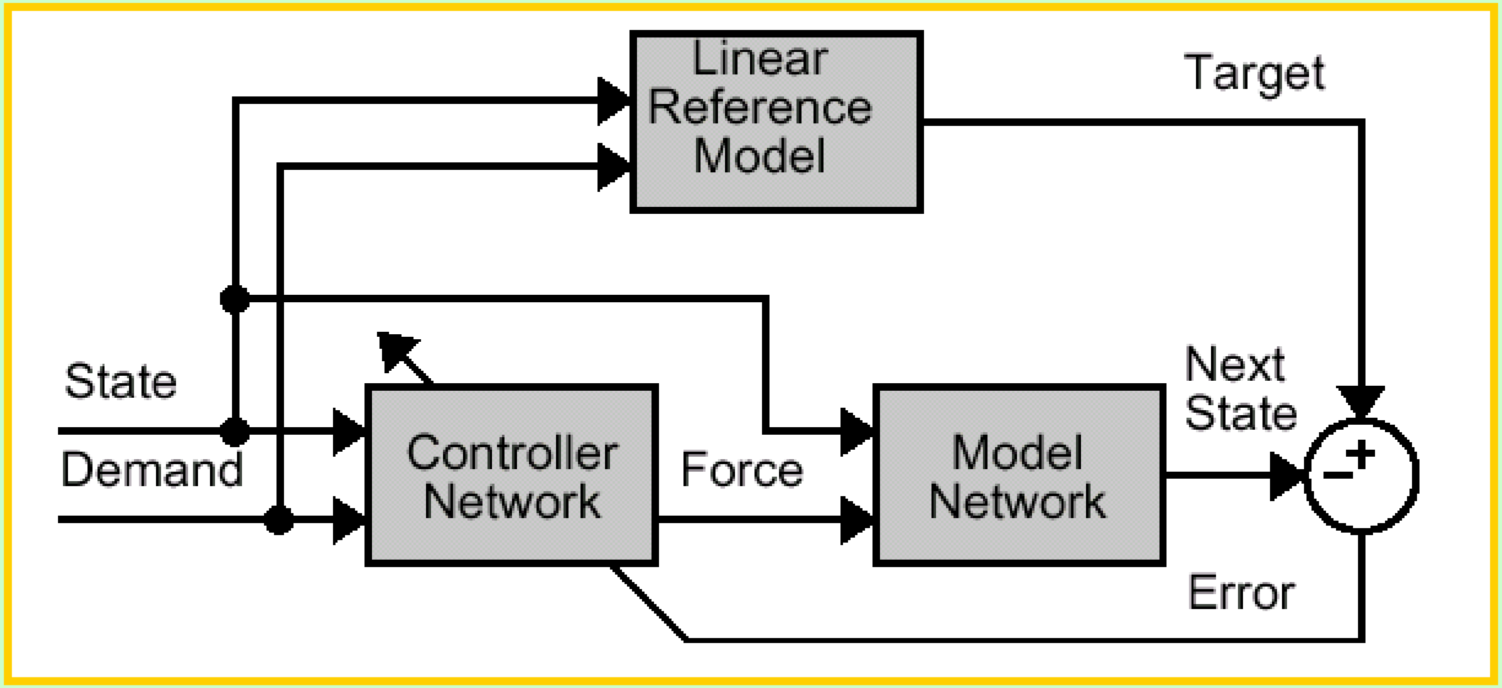
Figure 4: Indirect MRAC

Once the model NN_M is trained, it is used to train the network NN_C which will act as the controller. The model NN_M is used instead of the real plant's output because the real plant is unknown, so back-propagation algorithms can not be used. In this way, the control error e_C is calculated as the difference between the desired reference model output y_d and $\hat{y}$, which is the closed loop predicted output.

Then, NN_M is fixed, its output and behaviou are known and easy to compute.

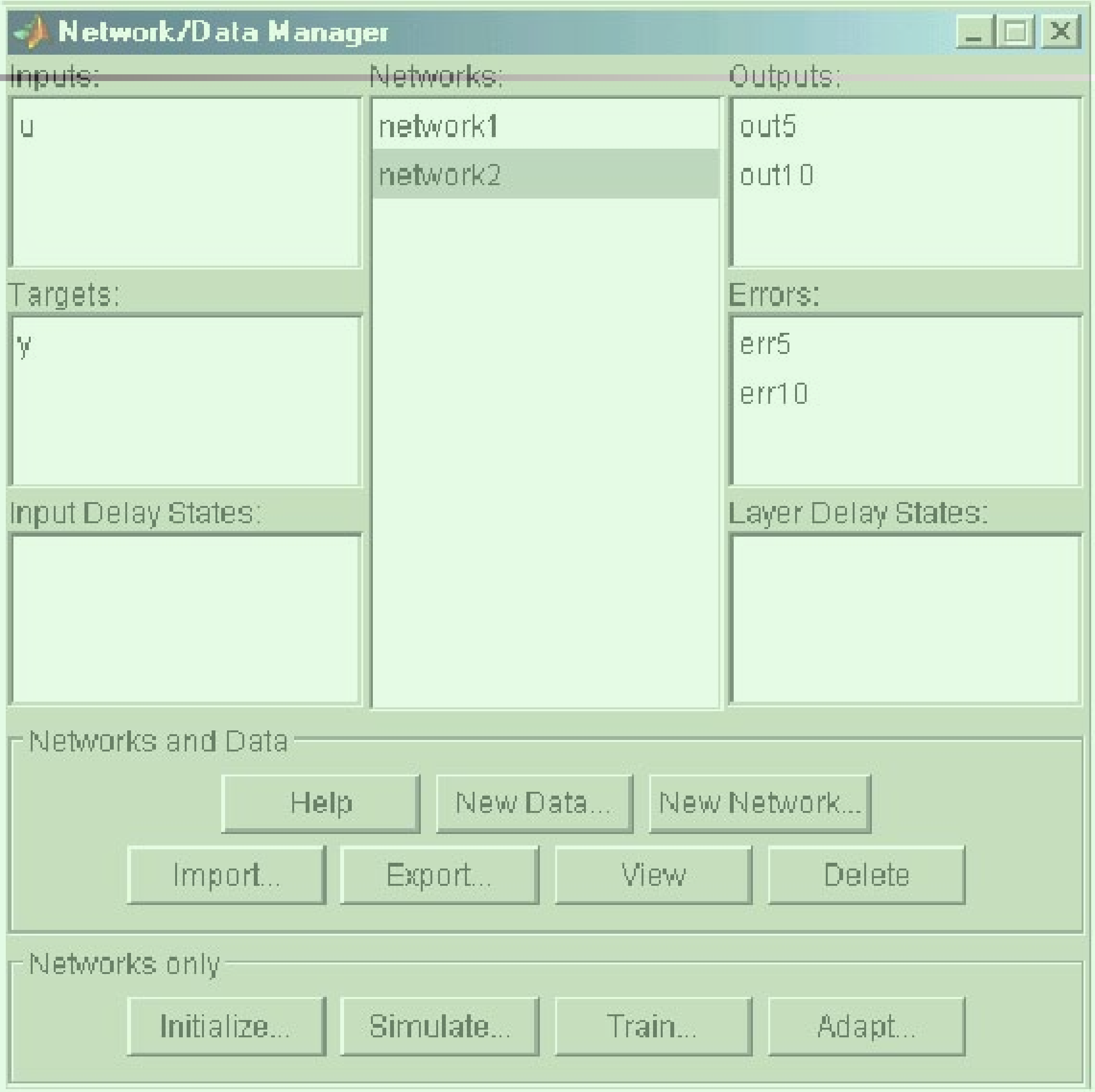
Model Reference Control

Matlab and Simulink solution

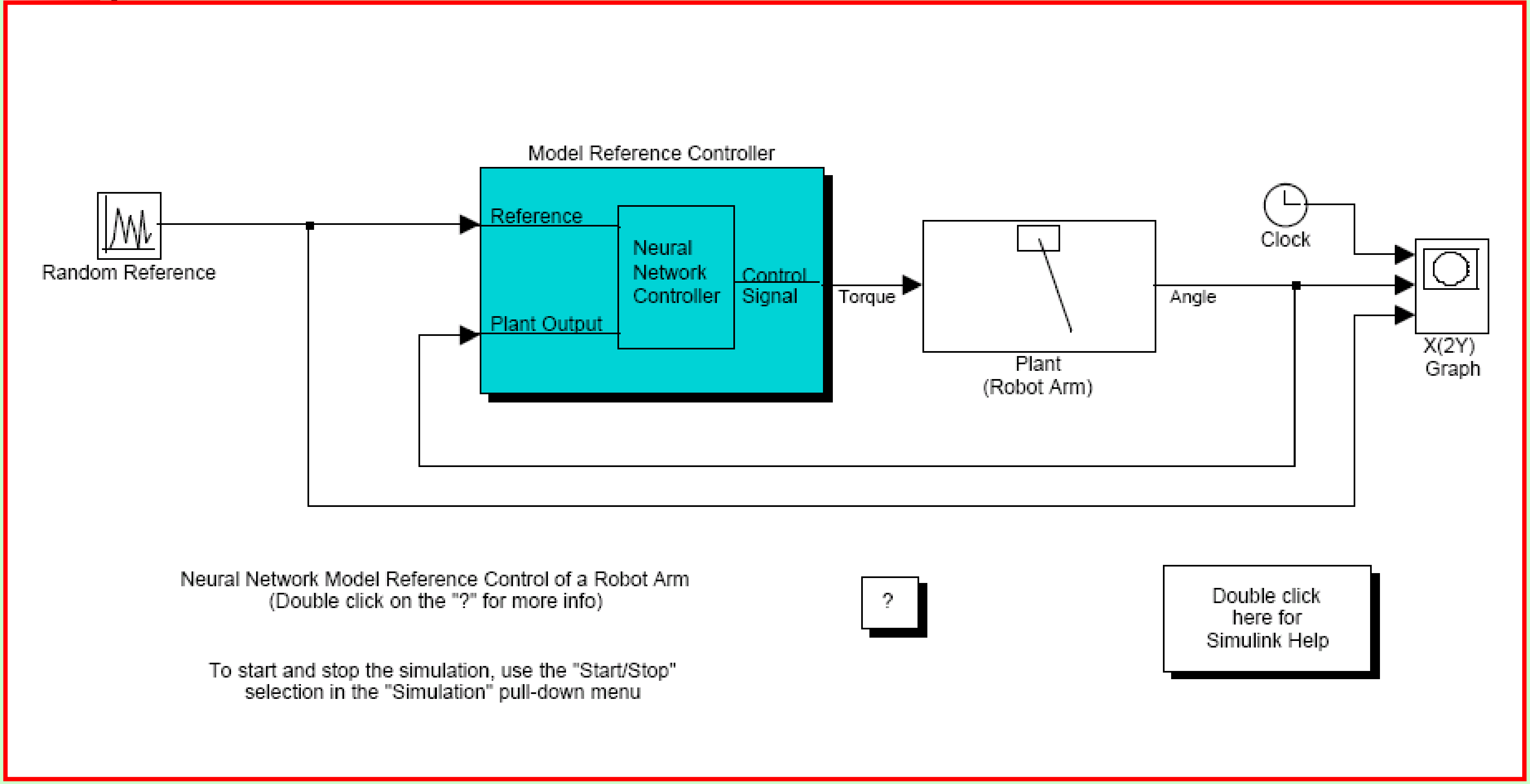


Neural controller, reference model, neural model

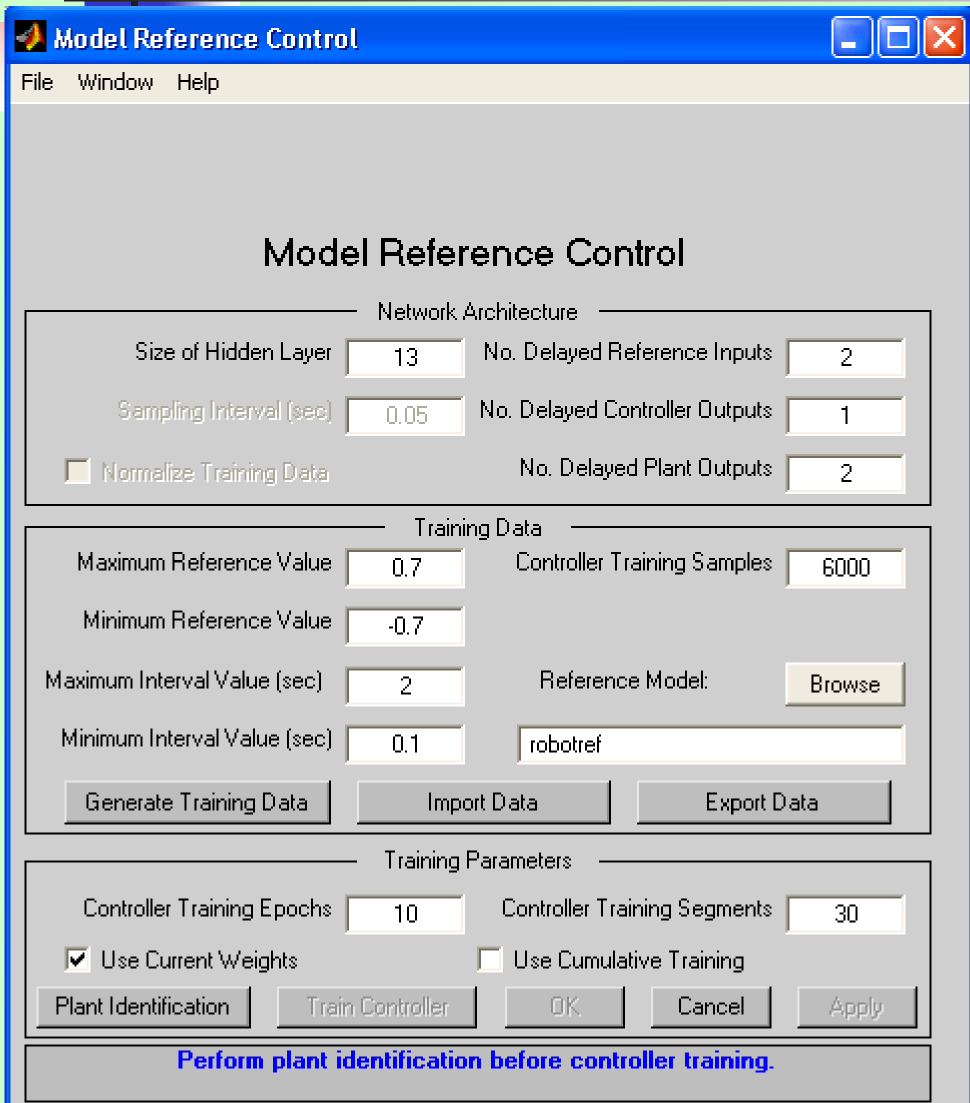
Matlab NNtool GUI (Graphical User Interface)

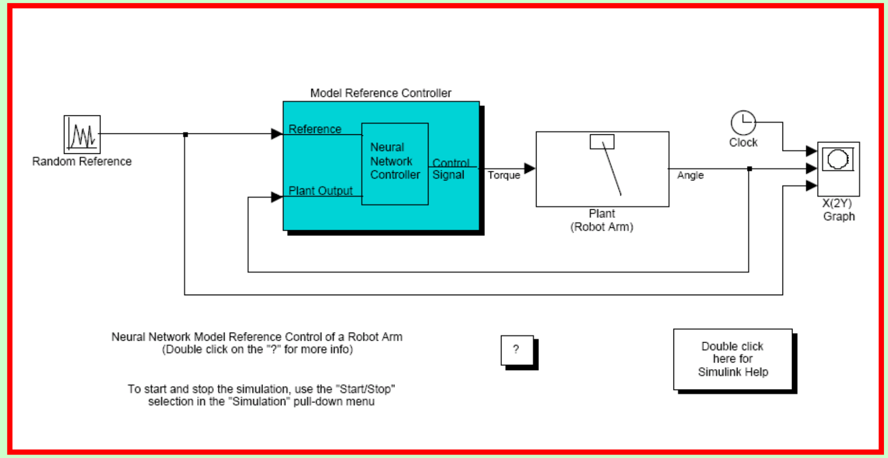


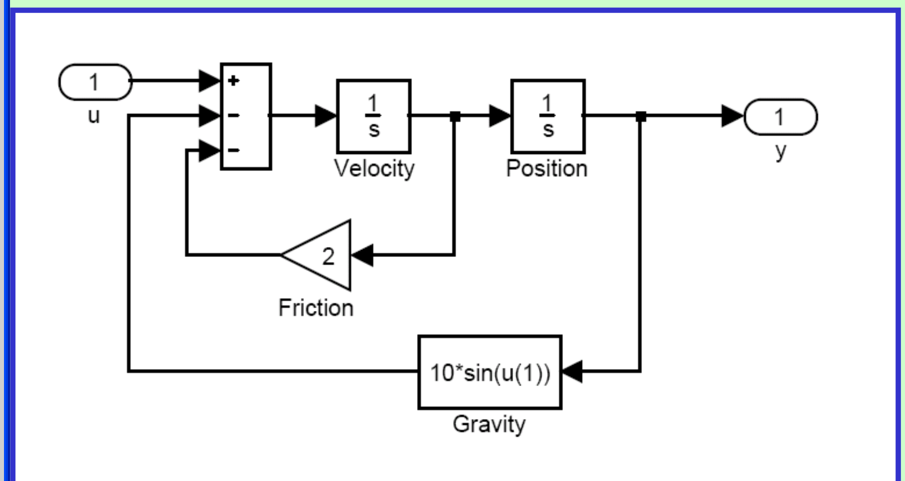
Control of a Robot Arm Example



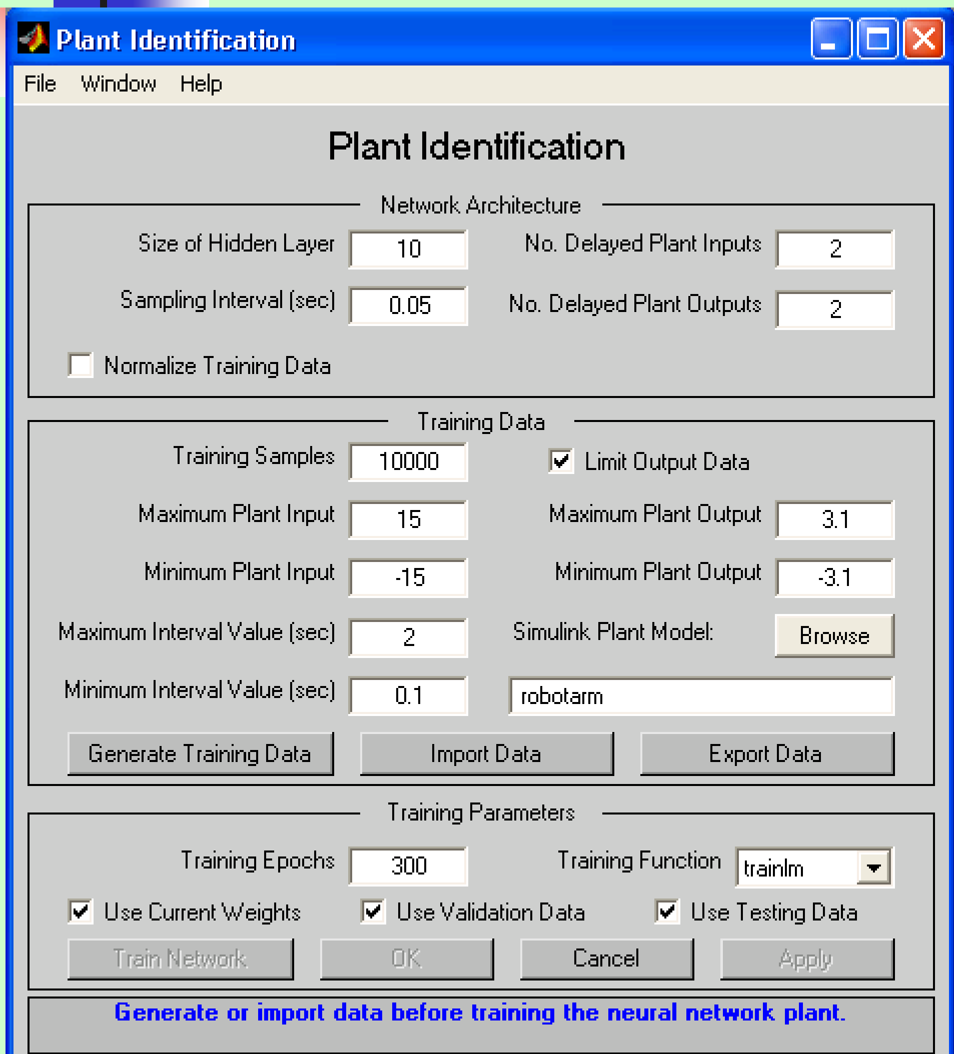
Control of a Robot Arm Example

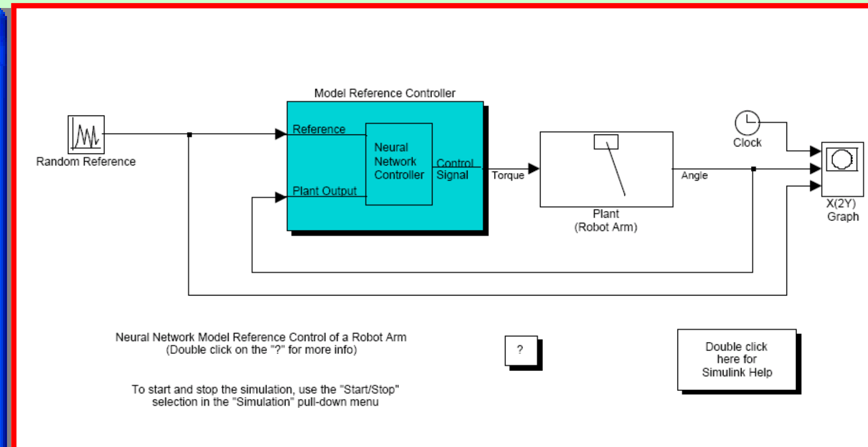






Control of a Robot Arm Example

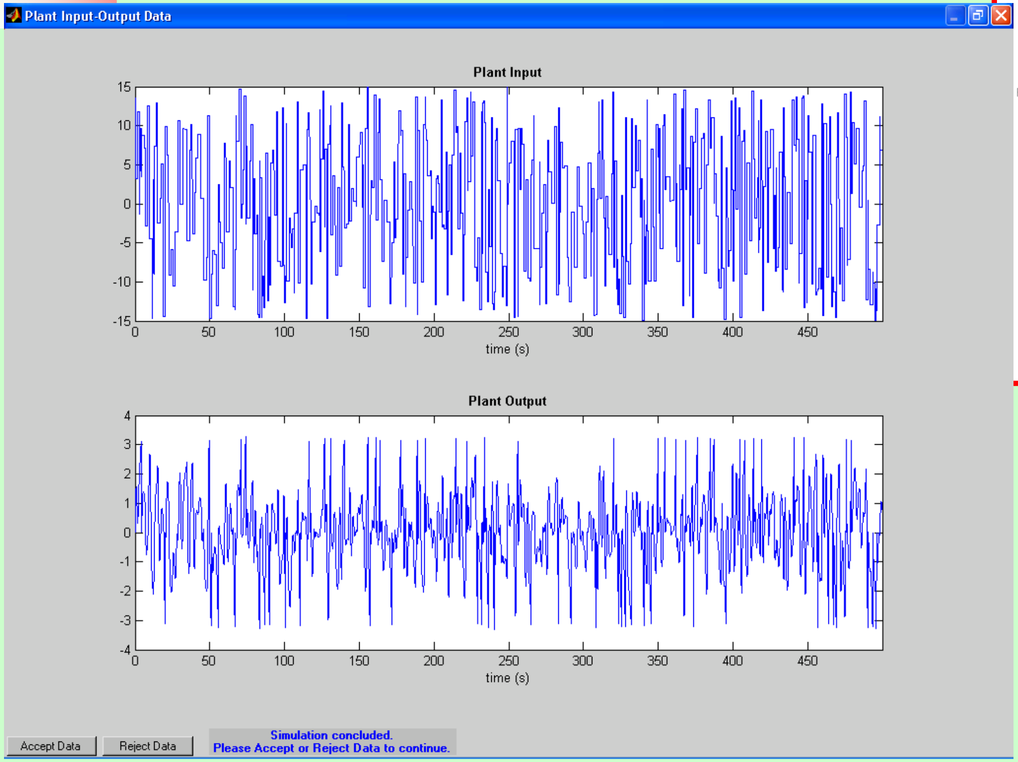


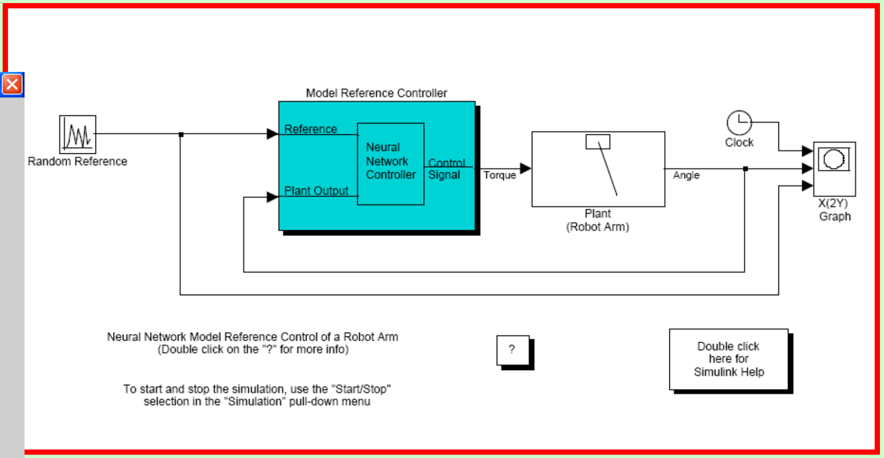


Plant Identification:

Data generation from the Reference Model for Neural Network training

Control of a Robot Arm Example

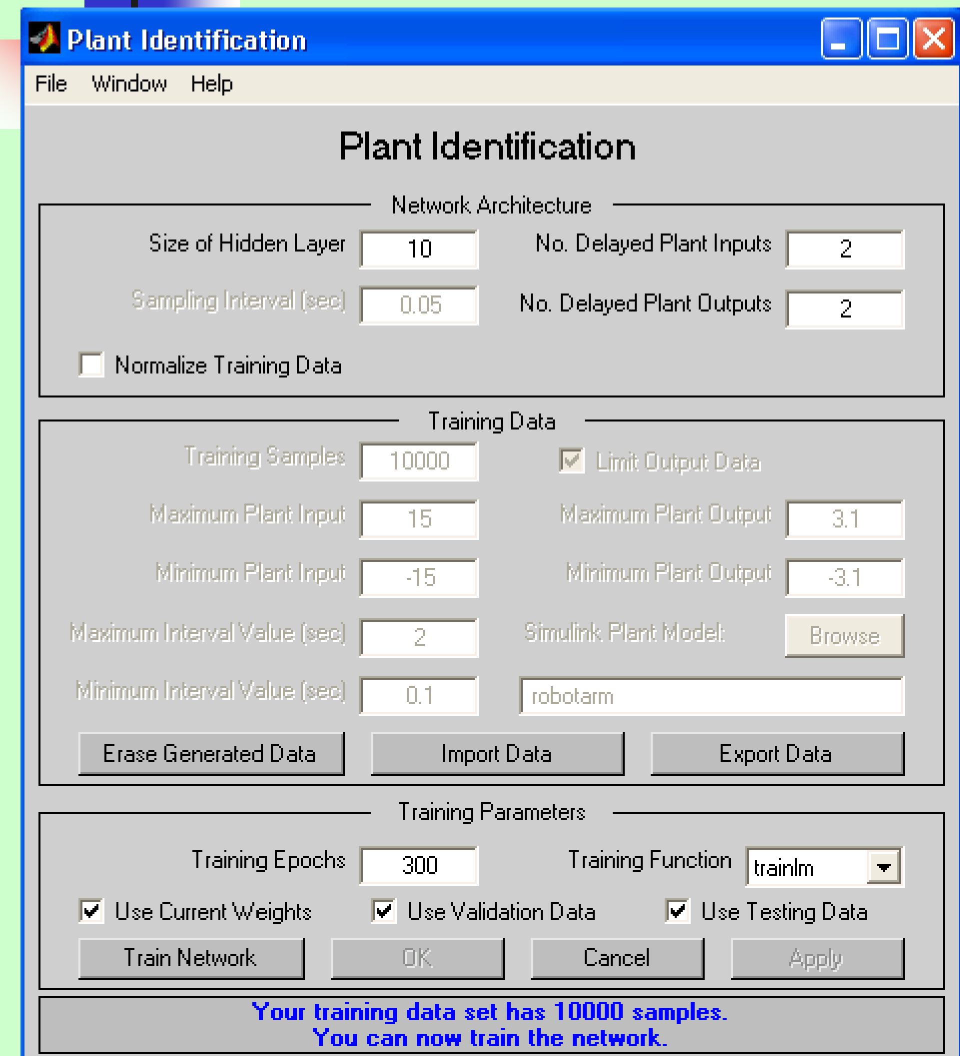


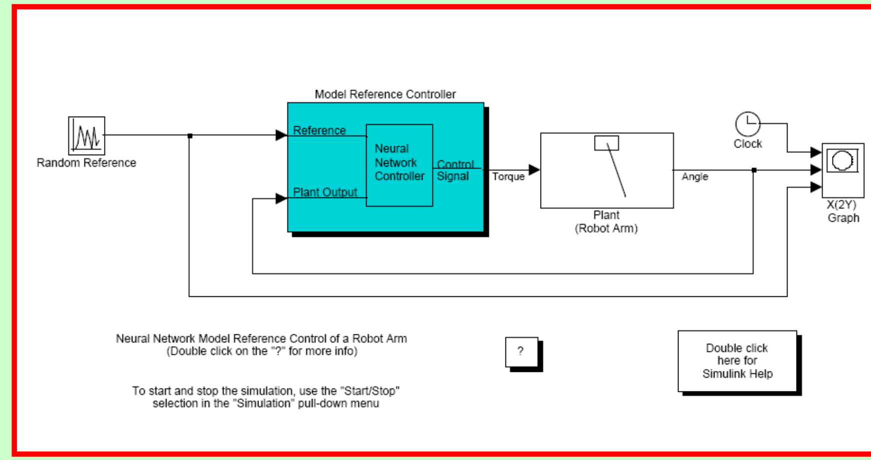


After Plant Identification:

Neural Network training

Control of a Robot Arm Example

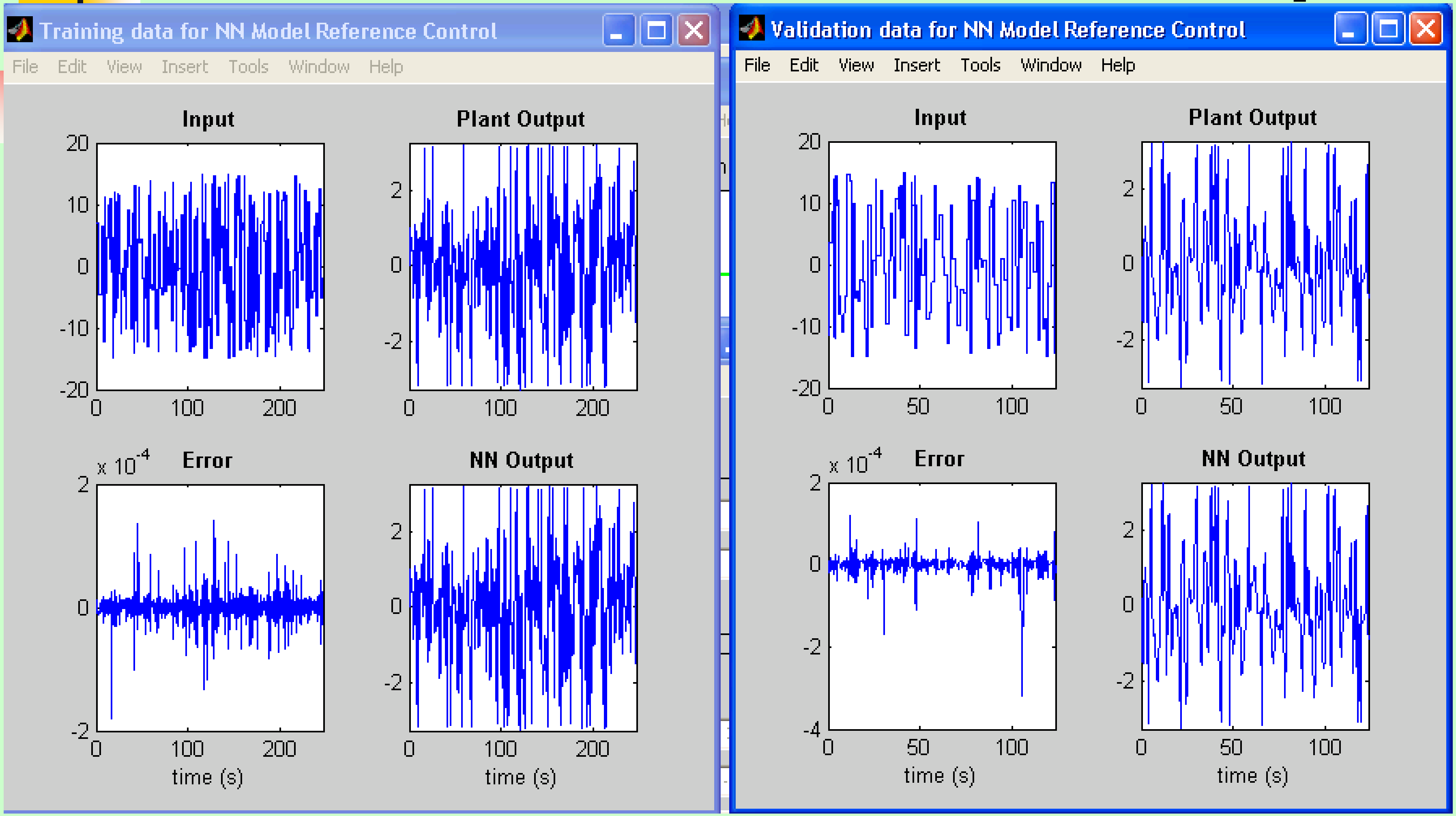




After Plant Identification:

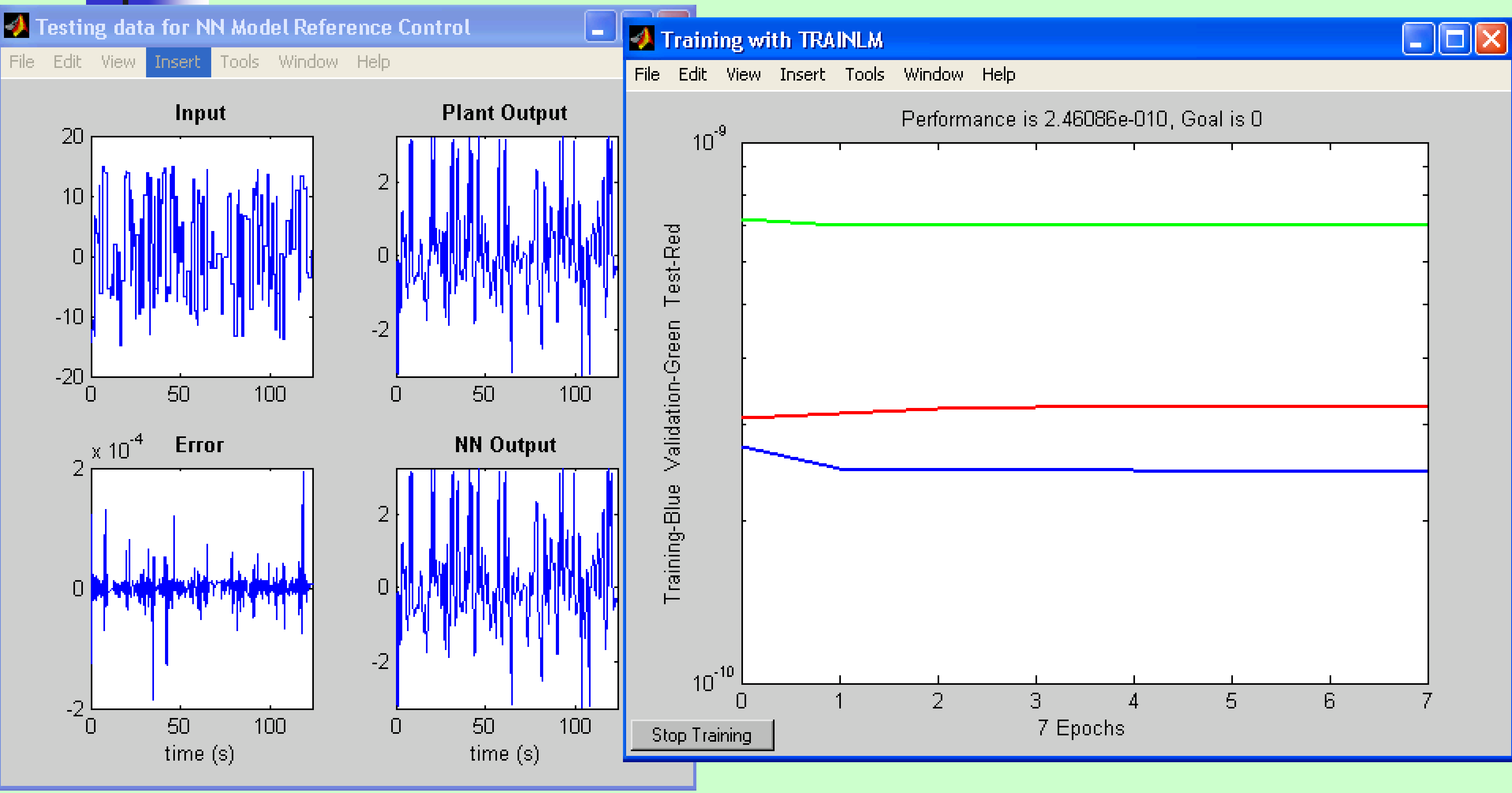
Neural Network training

Control of a Robot Arm Example



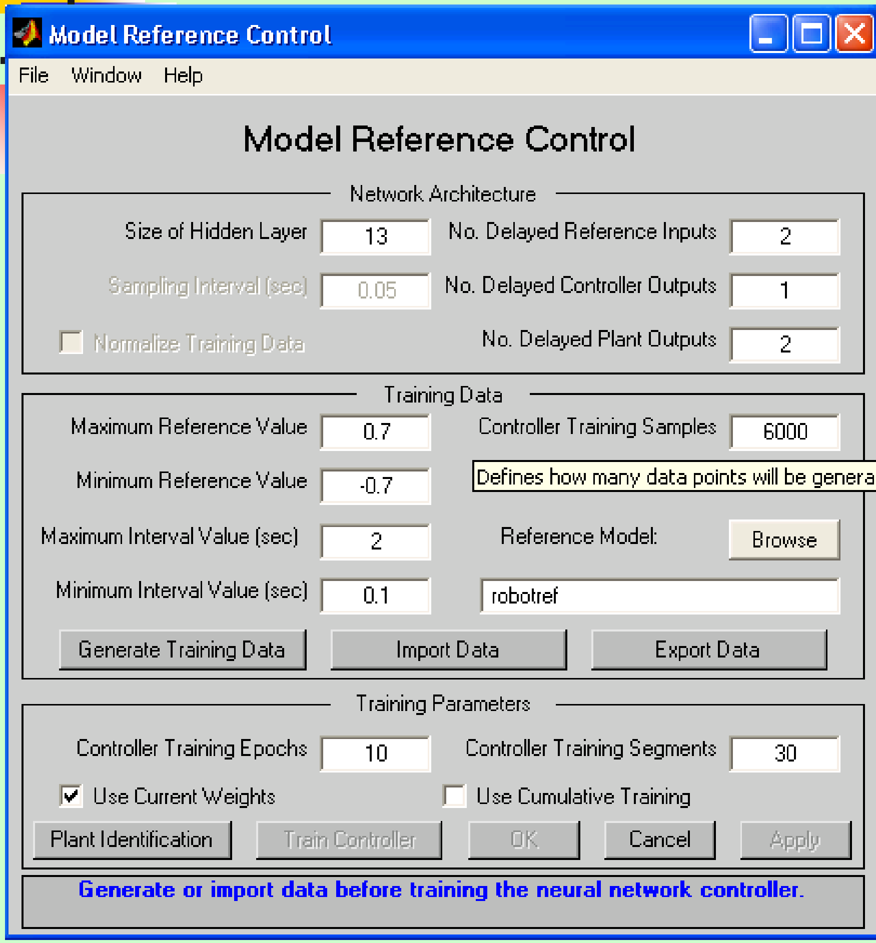
Training and Validation Data

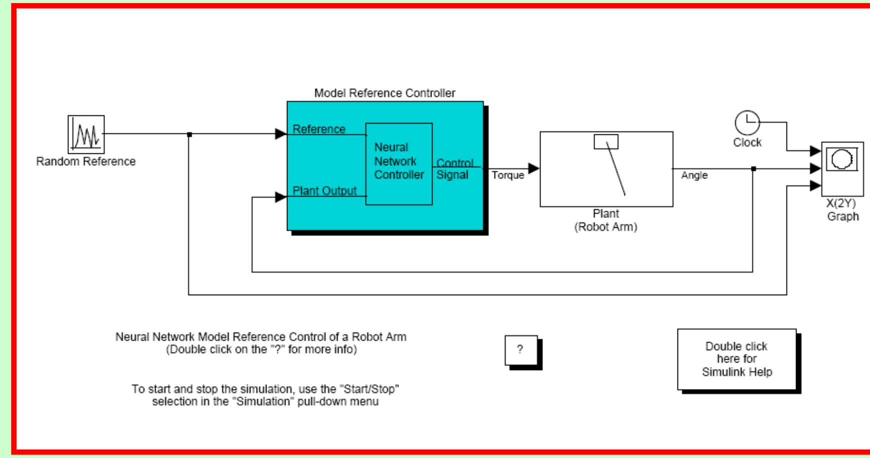
Control of a Robot Arm Example



Testing Data and Training Results

Control of a Robot Arm Example

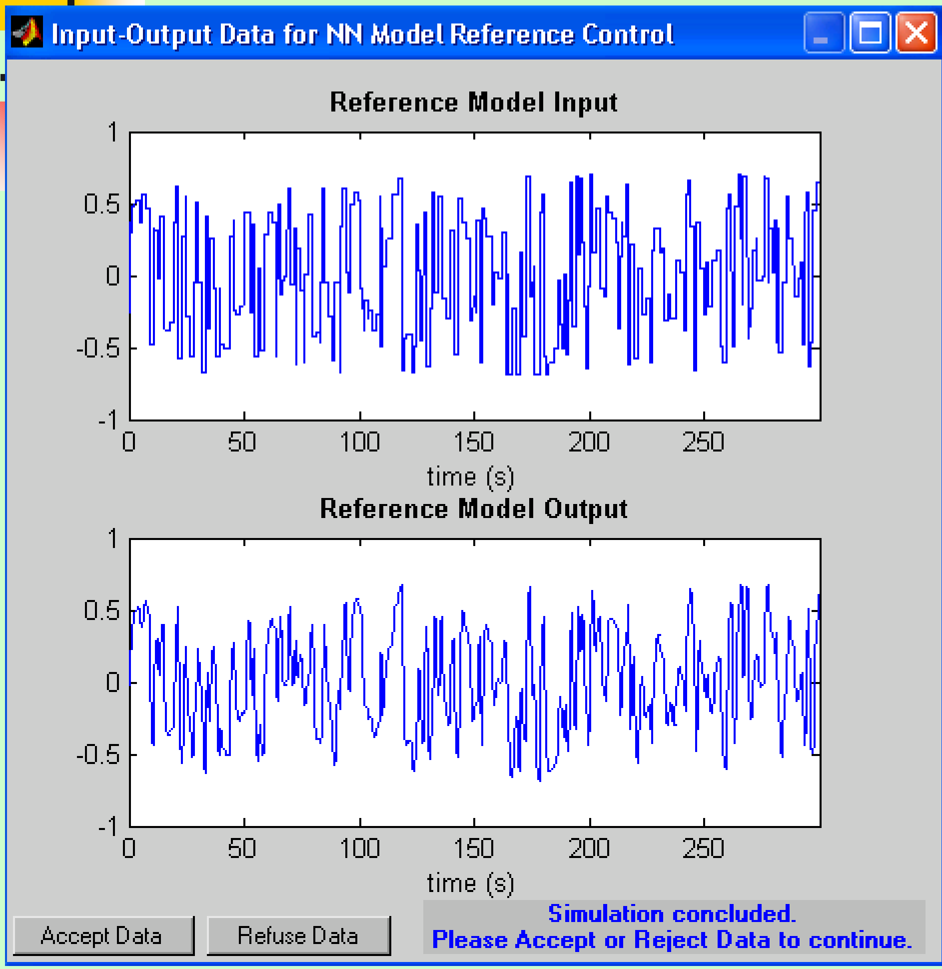


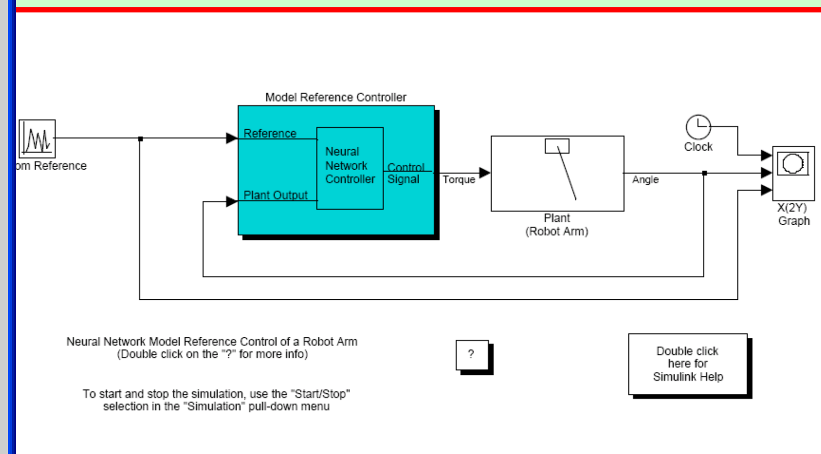


Plant identification with a NN

Data Generation for NN Controller Identification

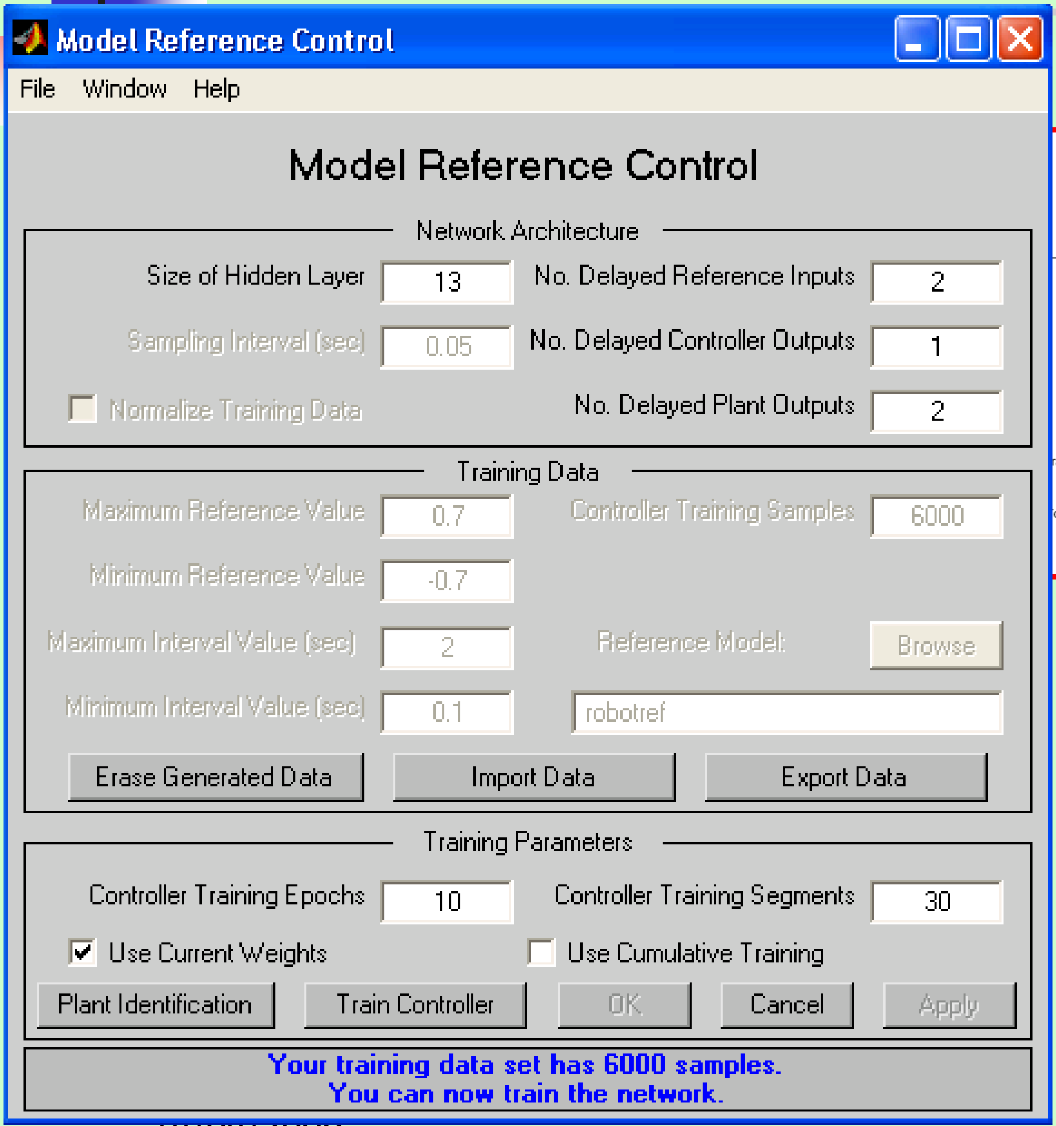
Control of a Robot Arm Example

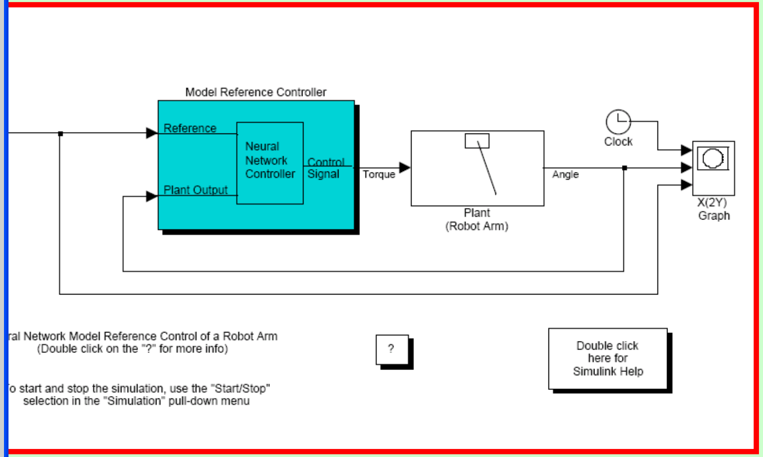




Accept the Data Generated for NN Controller Identification

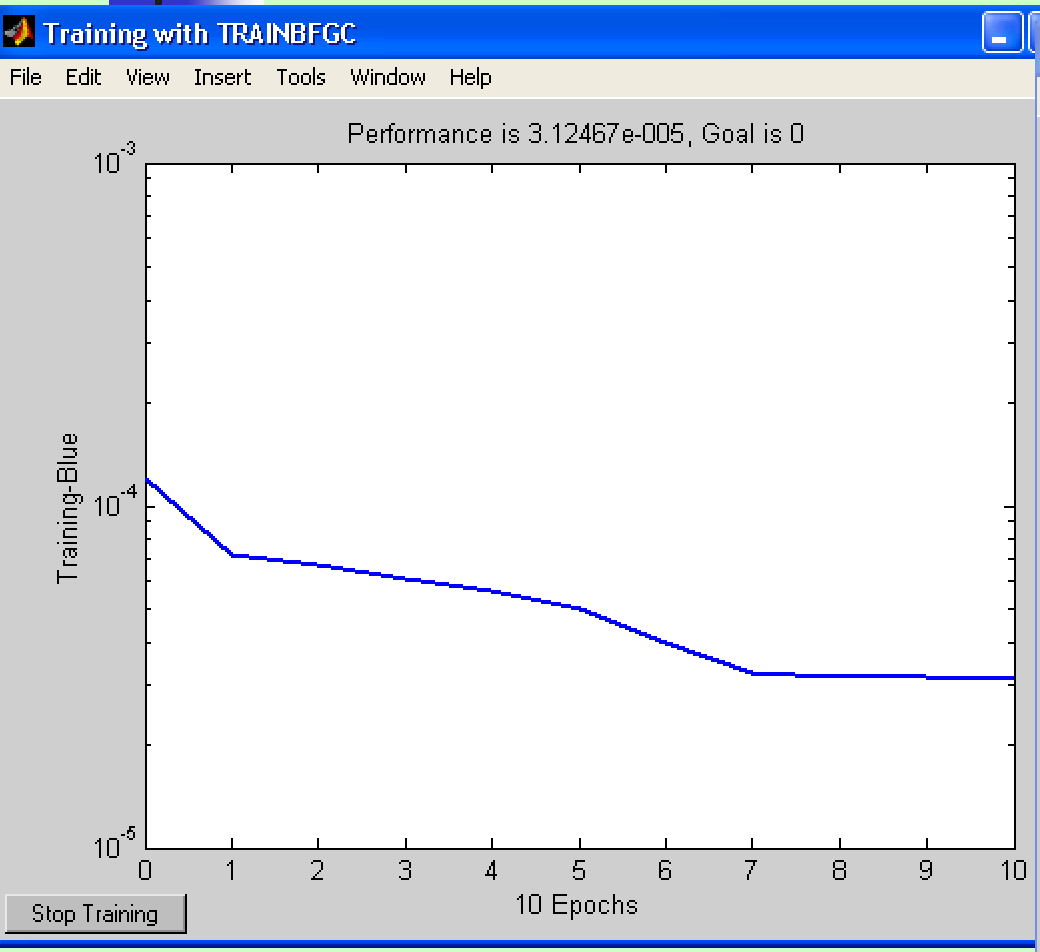
Control of a Robot Arm Example

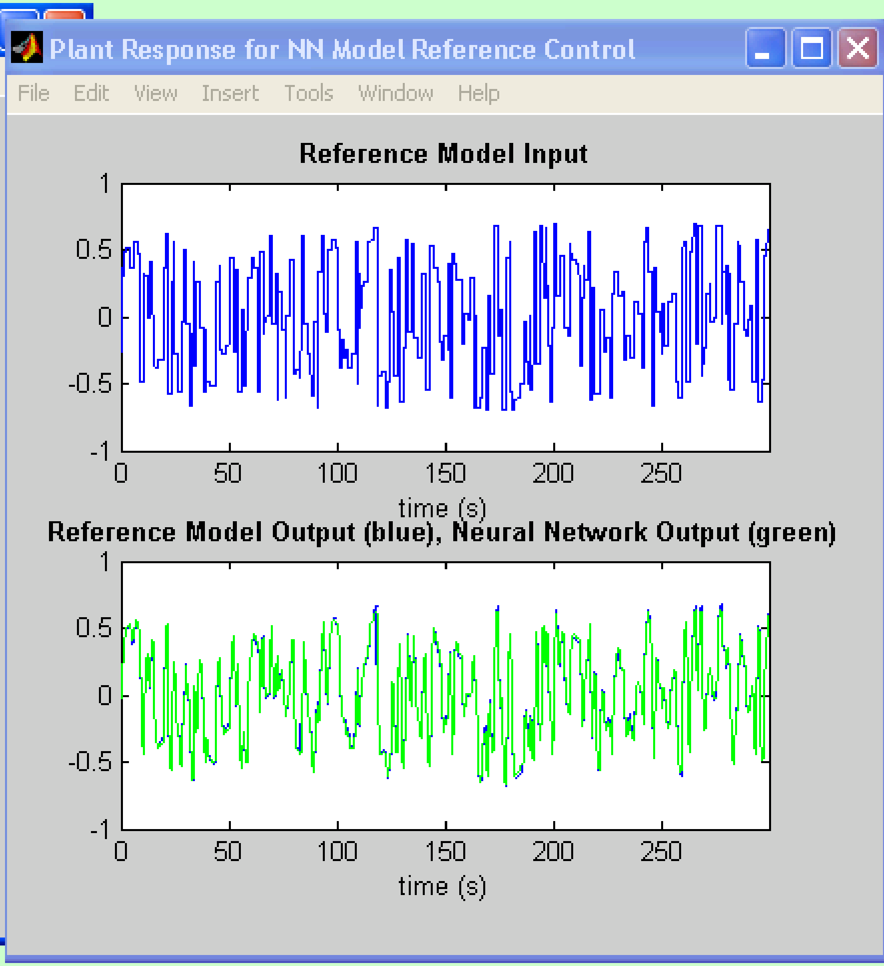




NN Controller Training

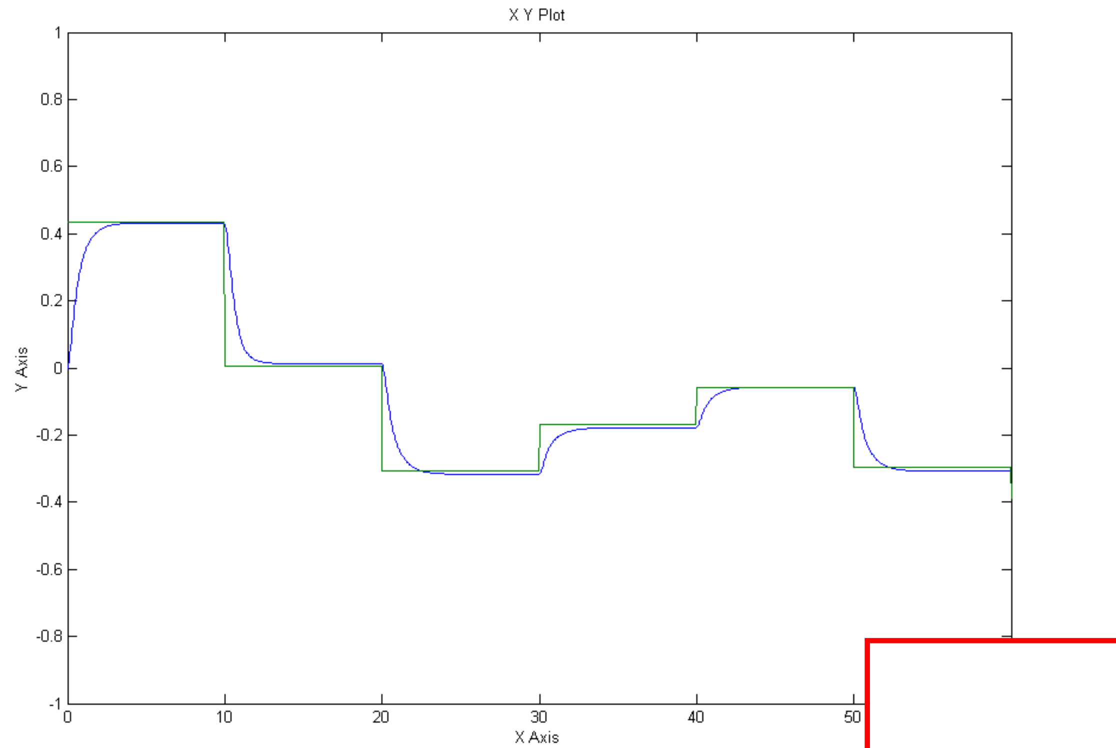
Control of a Robot Arm Example



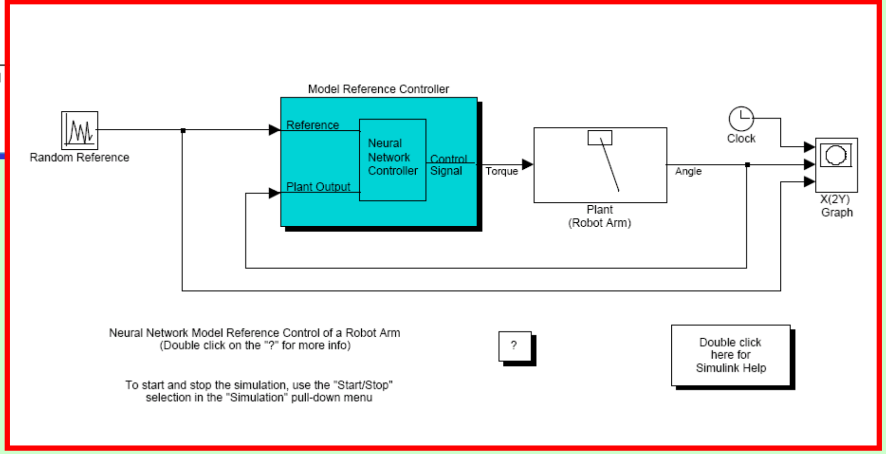


NN Controller Training and Results

Control of a Robot Arm Example



Reference and Tracked Output Signals



Simulation Final Results