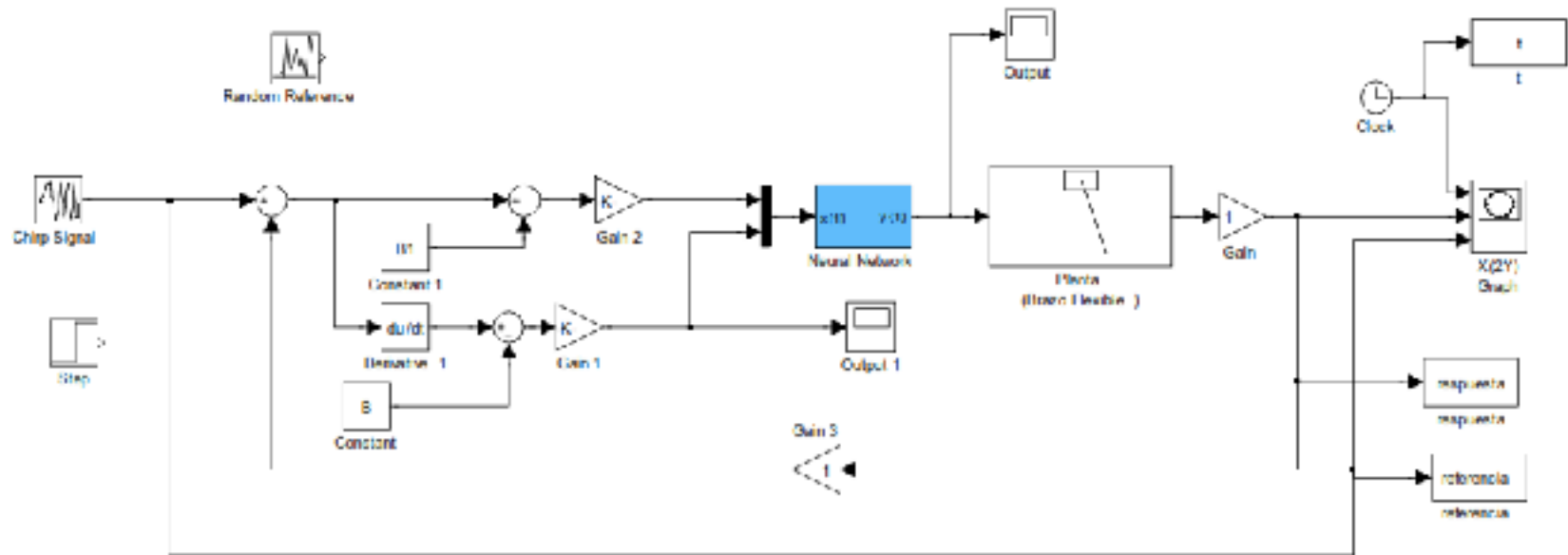
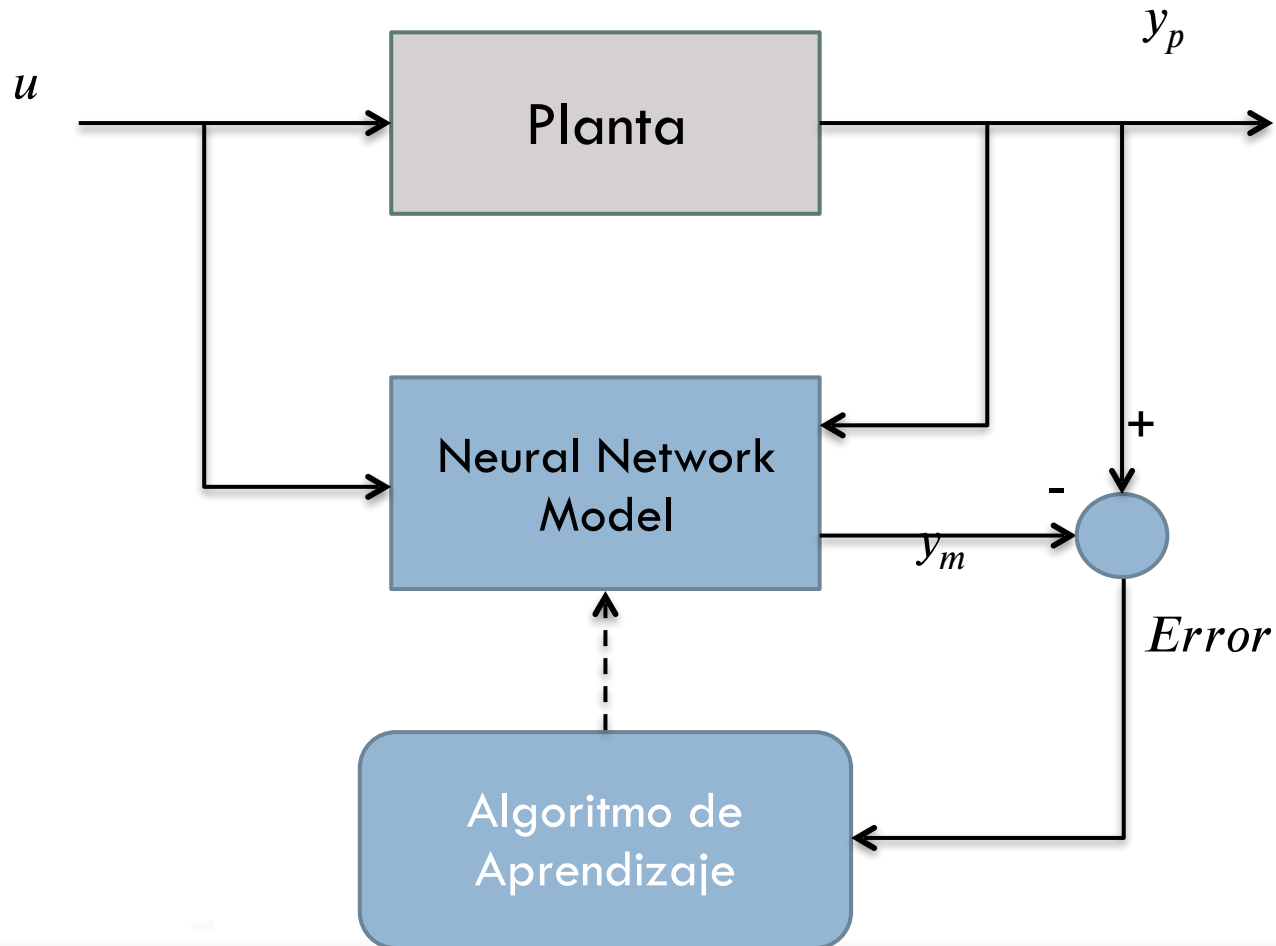


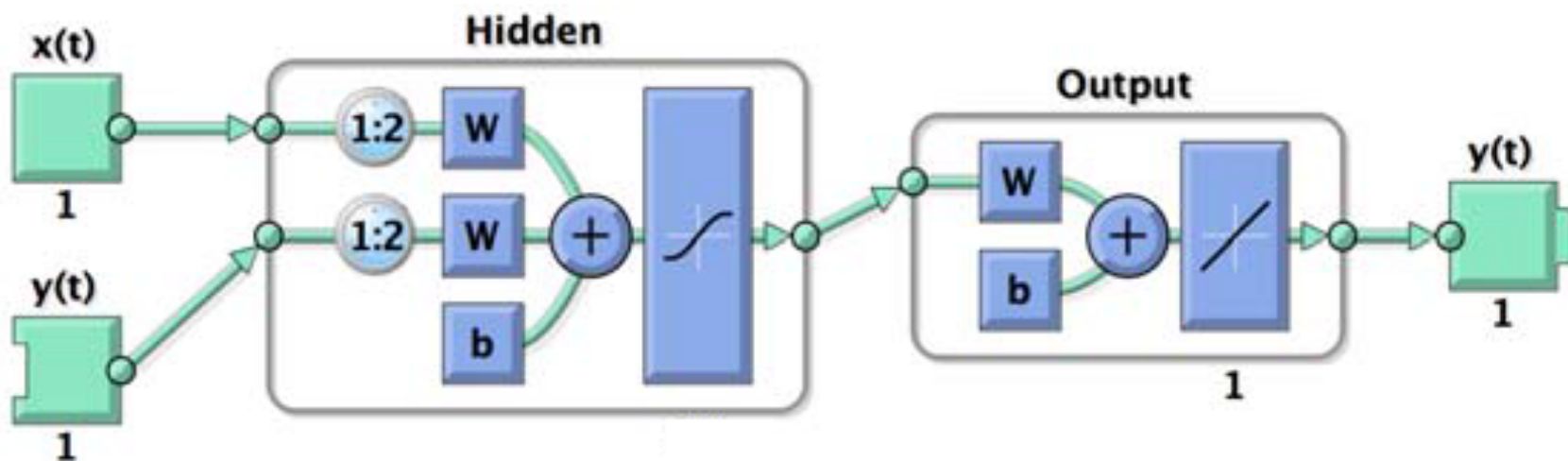
# Identificación y Control de Sistemas con Redes Neuronales



# Identificación del modelo de una planta



# Aproximador universal



# Modelo del motor de D.C

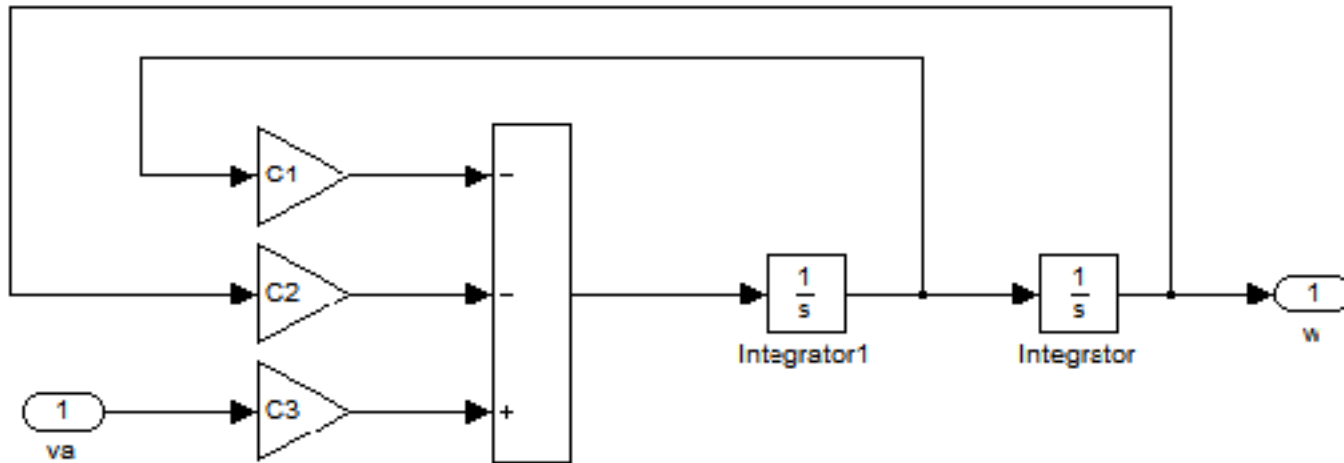
$$L \frac{di_a}{dt} = v_a - R_a i_a - K w \quad (1)$$

$$J \frac{dw}{dt} = K_t i_a - B w \quad (2)$$

Reemplazando la ecuación (2) en la (1) se obtiene una expresión que no depende de la corriente de armadura.

$$JL \frac{d^2 w}{dt^2} = - (R_a J + L B) \frac{dw}{dt} - (K^2 + R_a B) w + K v_a$$

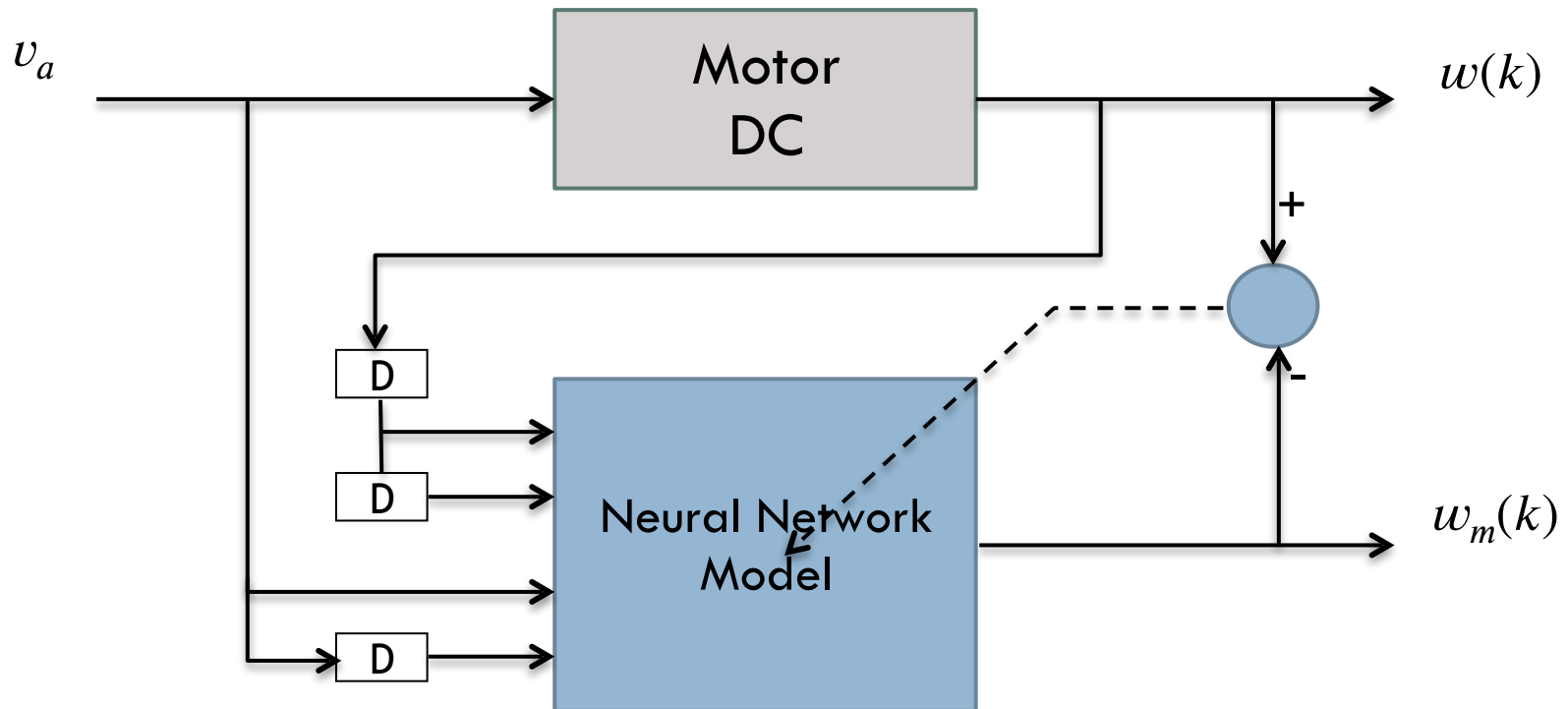
# Modelo del motor de D.C



$$\begin{aligned}R_a &= 7.56; \\L &= 0.055; \\K &= 3.475; \\J &= 0.068; \\B &= 0.03475;\end{aligned}$$

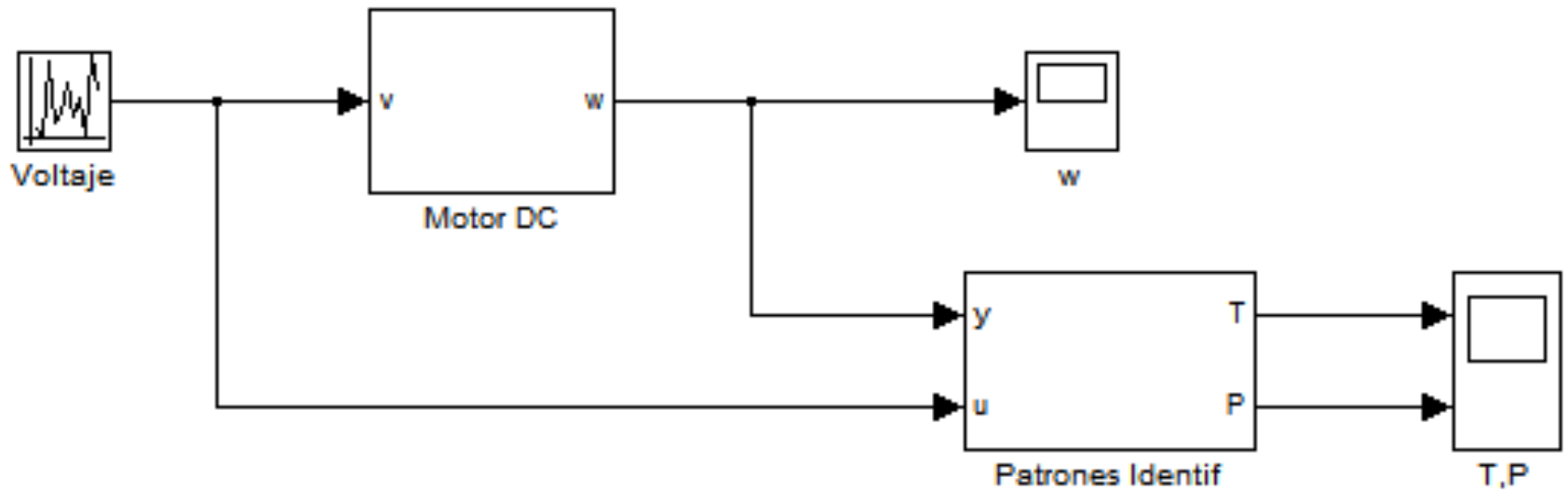
$$\begin{aligned}C1 &= (R_a * J + L * B) / (J * L) \\C2 &= (K^2 + R_a * B) / (J * L) \\C3 &= K / (J * L)\end{aligned}$$

# Identificación - ARX

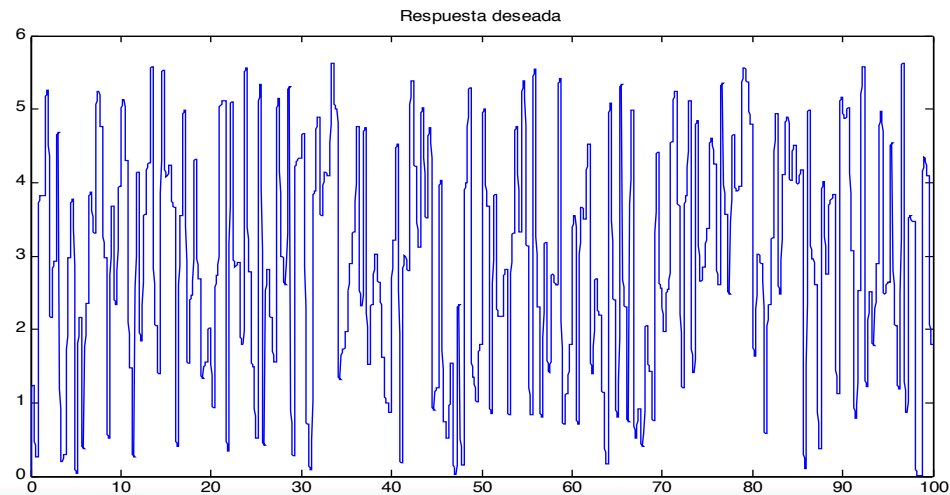
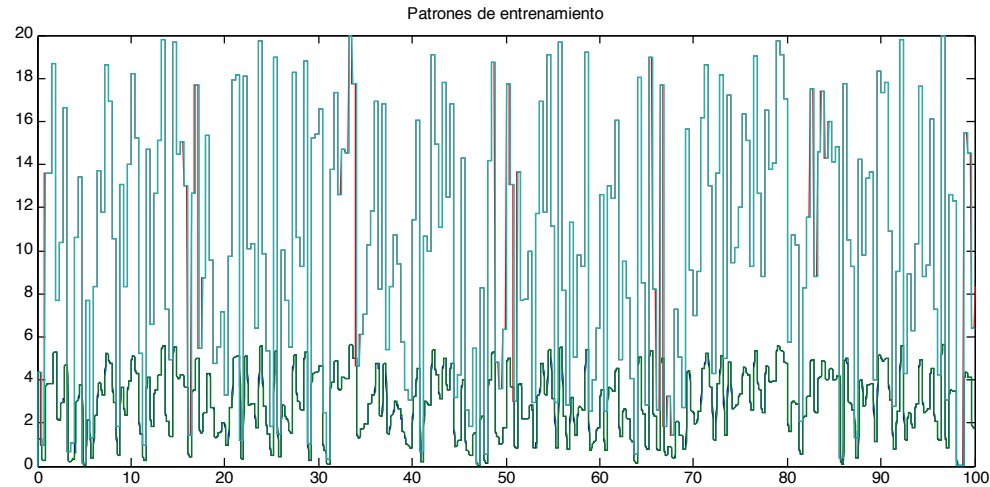


$$w_m(k) = f(w(k-1), w(k-2), v_a(k), v_a(k-1))$$

# Patrones de entrenamiento

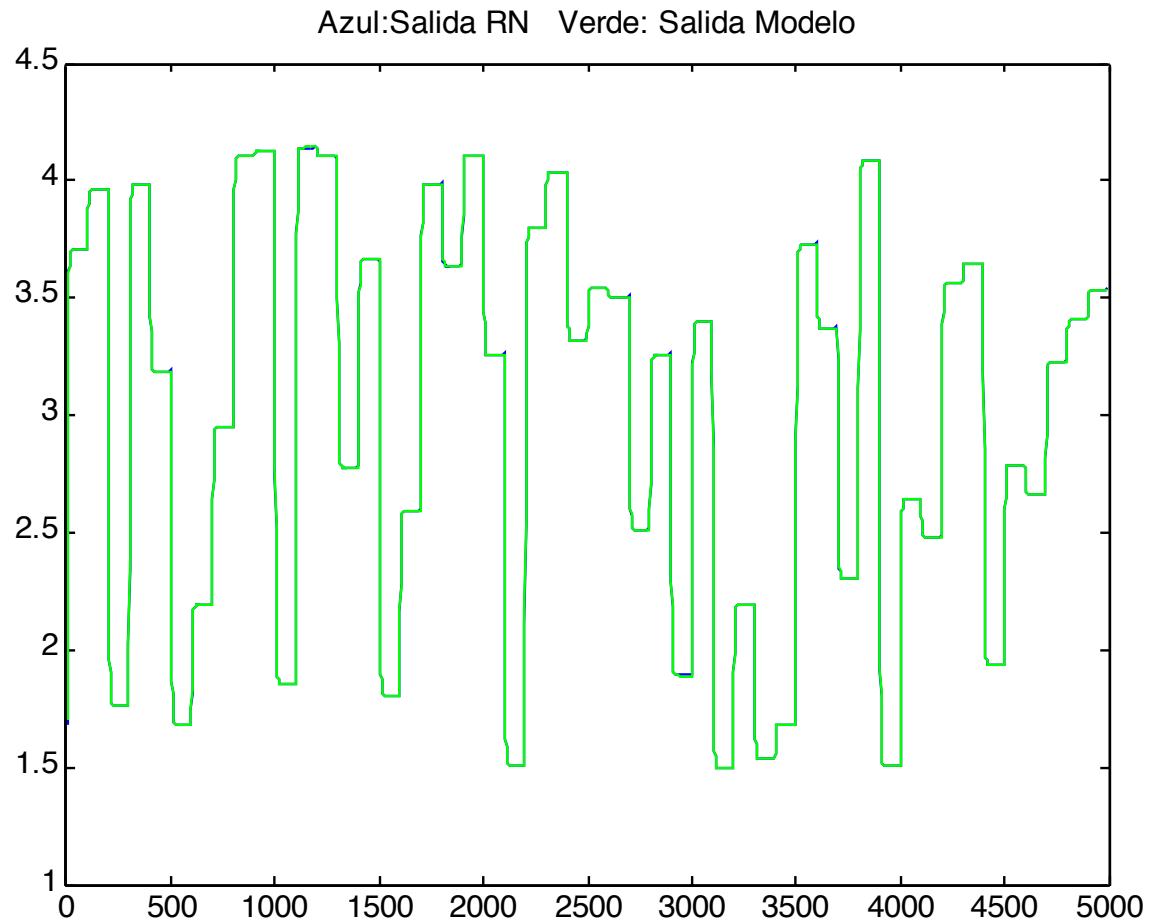


# Patrones de Entrenamiento de la RN



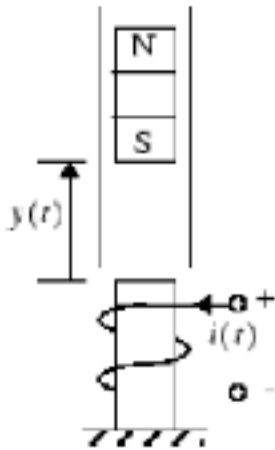


# Respuesta



# Levitador Magnético

□ ballrepel0



$$\frac{d^2 y(t)}{dt^2} = -g + \frac{\alpha}{M} \frac{i^2(t)}{y(t)} - \frac{\beta}{M} \frac{dy(t)}{dt}$$

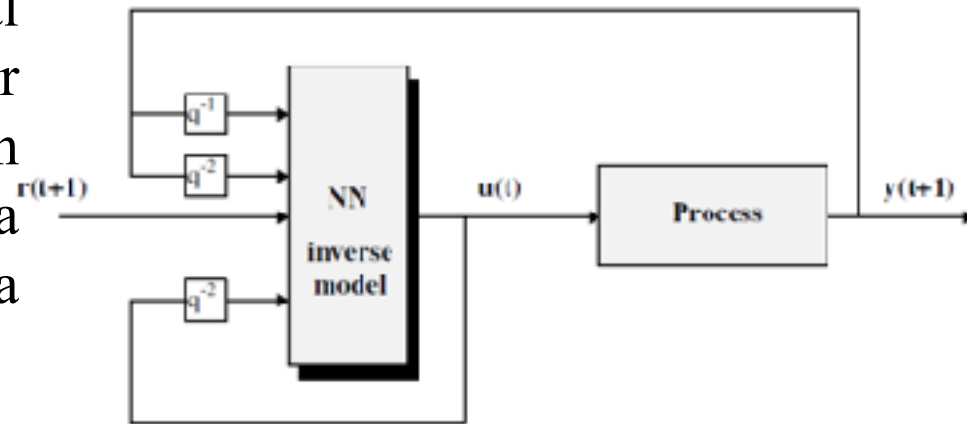
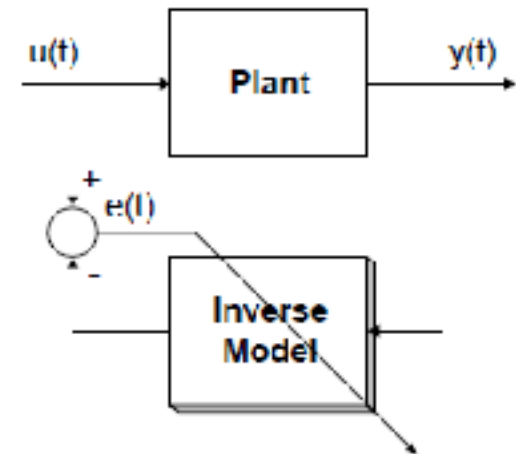
# Esquemas de control

- Control directo por modelo inverso
- Control con modelo de referencia
- Control híbrido
- Control adaptativo
- Control predictivo
- Control NARMA-L2

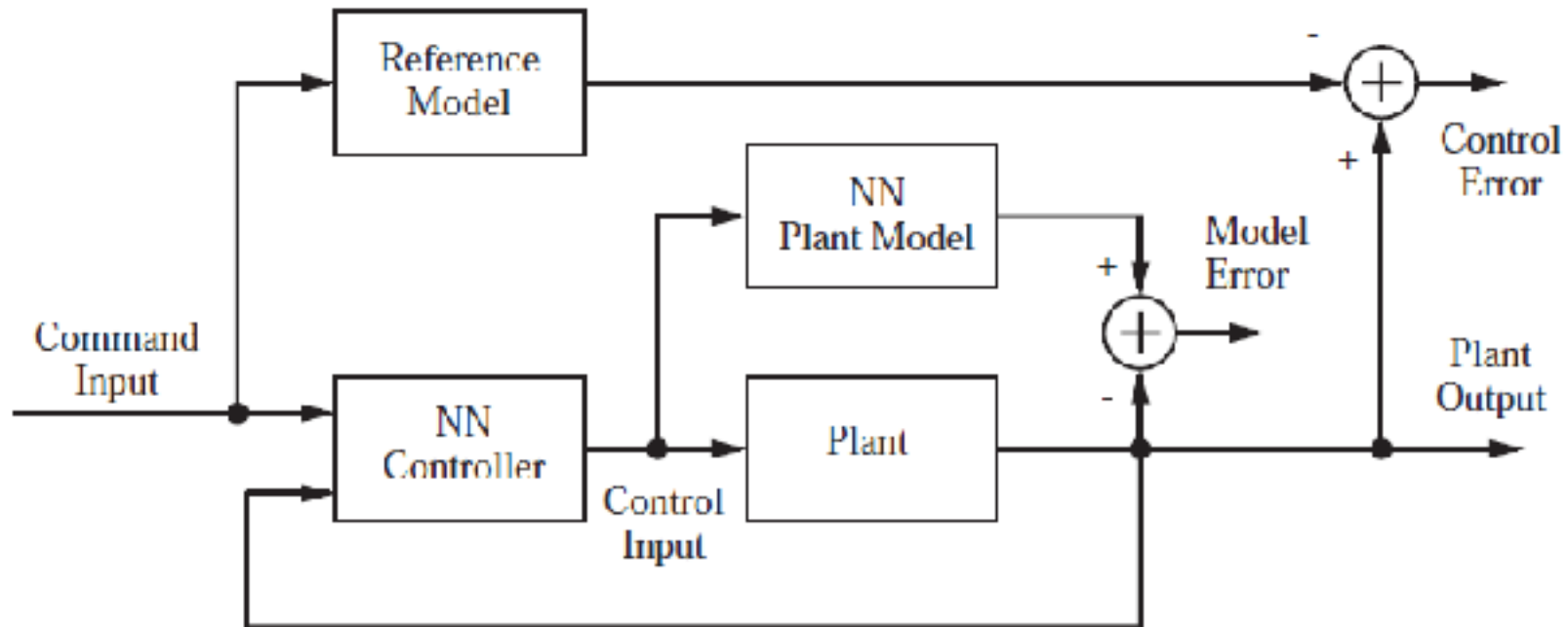
# Control directo por modelo inverso

El control por modelo inverso es una técnica, que busca cancelar la dinámica de la planta al colocar un elemento en cascada con ella, en este caso una red neuronal, siendo esta una aproximación matemática del inverso de la planta.

Se realiza un entrenamiento general usando datos off-line para encontrar el modelo inverso de la planta. En seguida la RNA entrenada se usa como controlador pues ella cancela la dinámica de la planta.

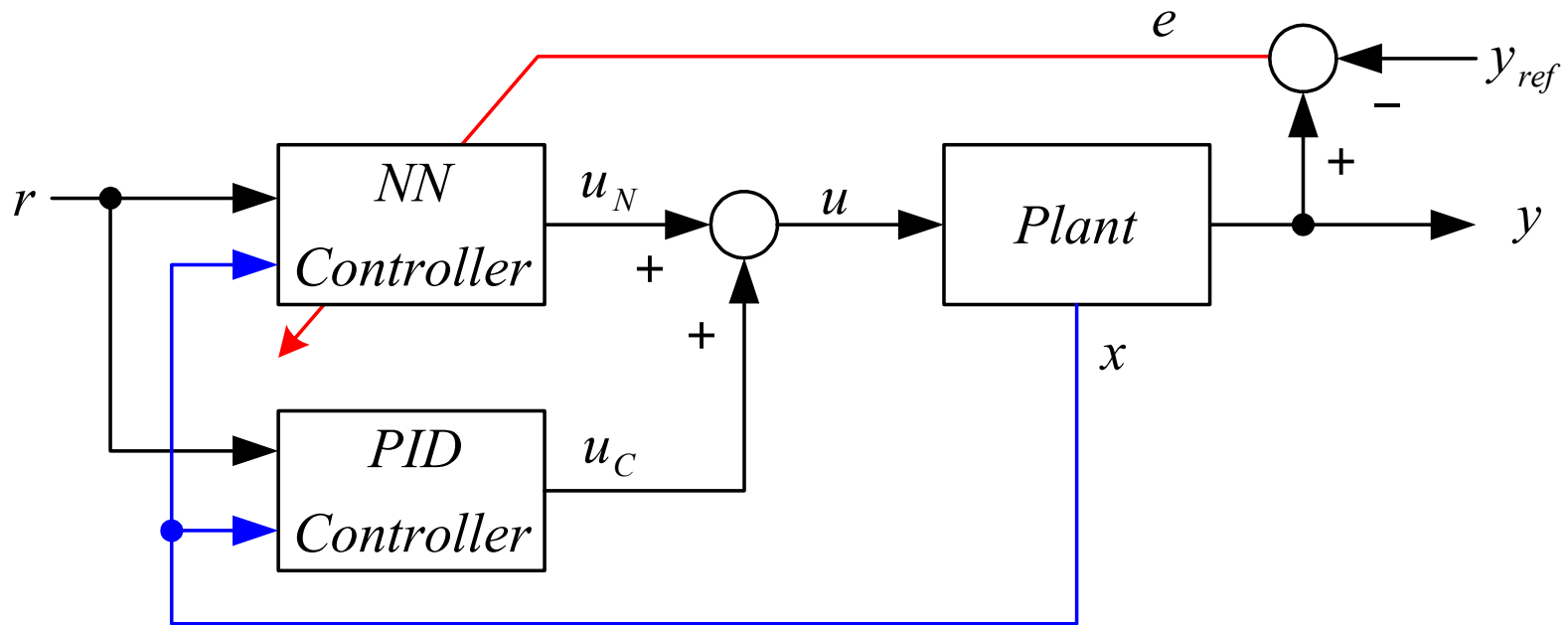


# Control con modelo de referencia



El control con modelo de referencia intenta lograr que la salida de la planta tienda asintóticamente a la salida del modelo de referencia.

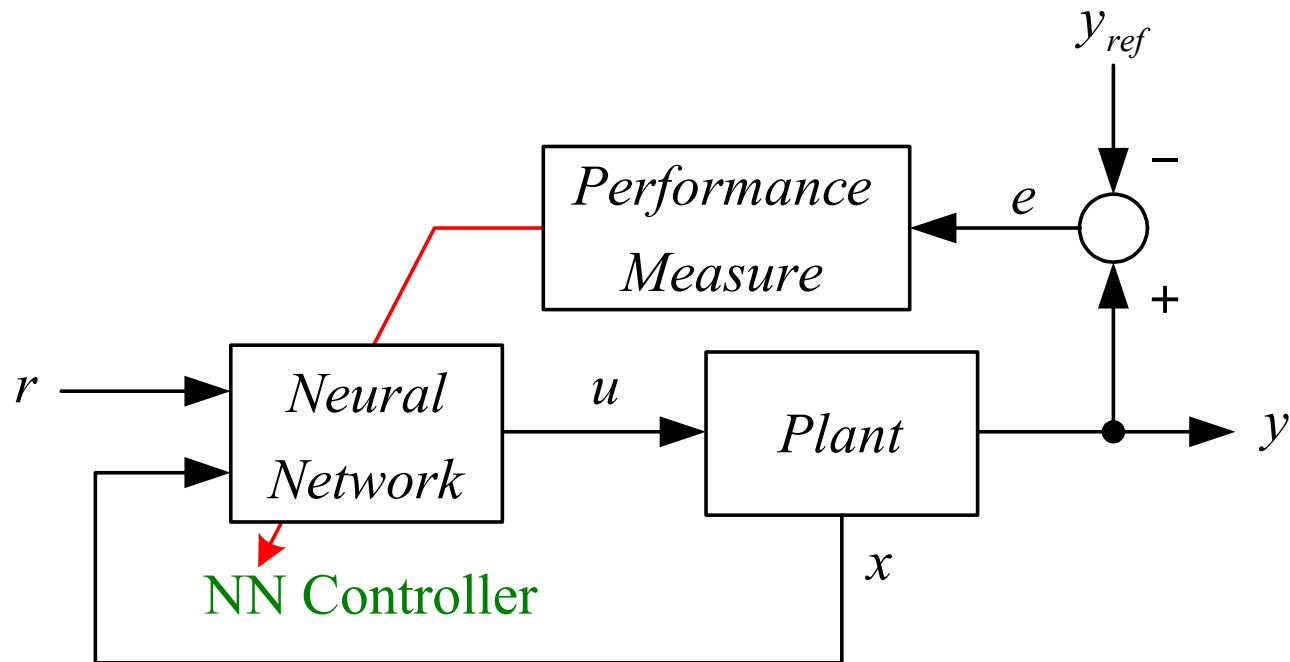
# Control híbrido



**Aprendizaje off-line:** Una aproximación a la ley de control deseada

**Aprendizaje on-line:** Mejora el control proporcionado por el PID

# El control adaptativo



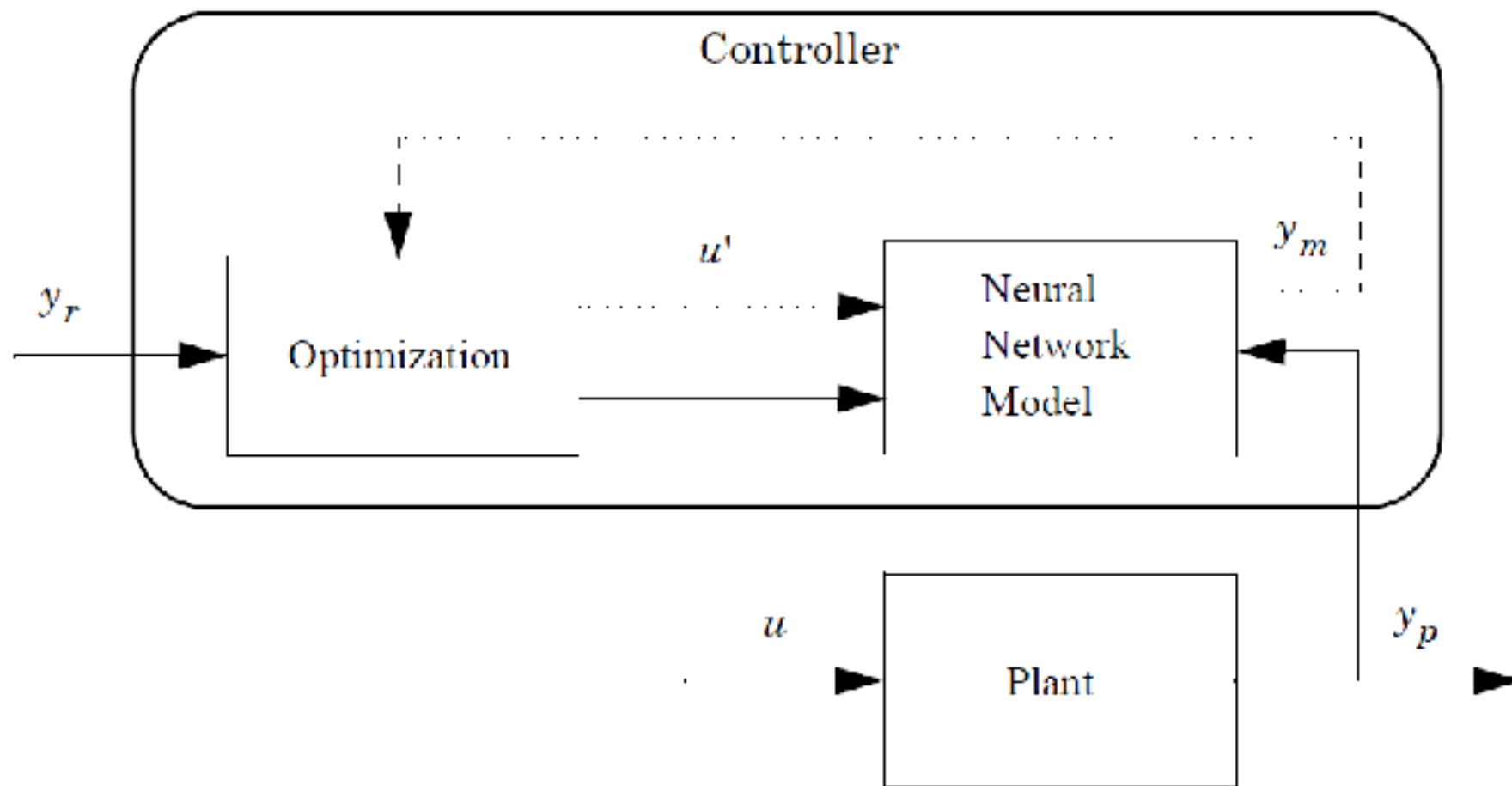
El error de seguimiento se evalúa de acuerdo a algún índice de desempeño. Los pesos se ajustan on-line.

# Control Predictivo

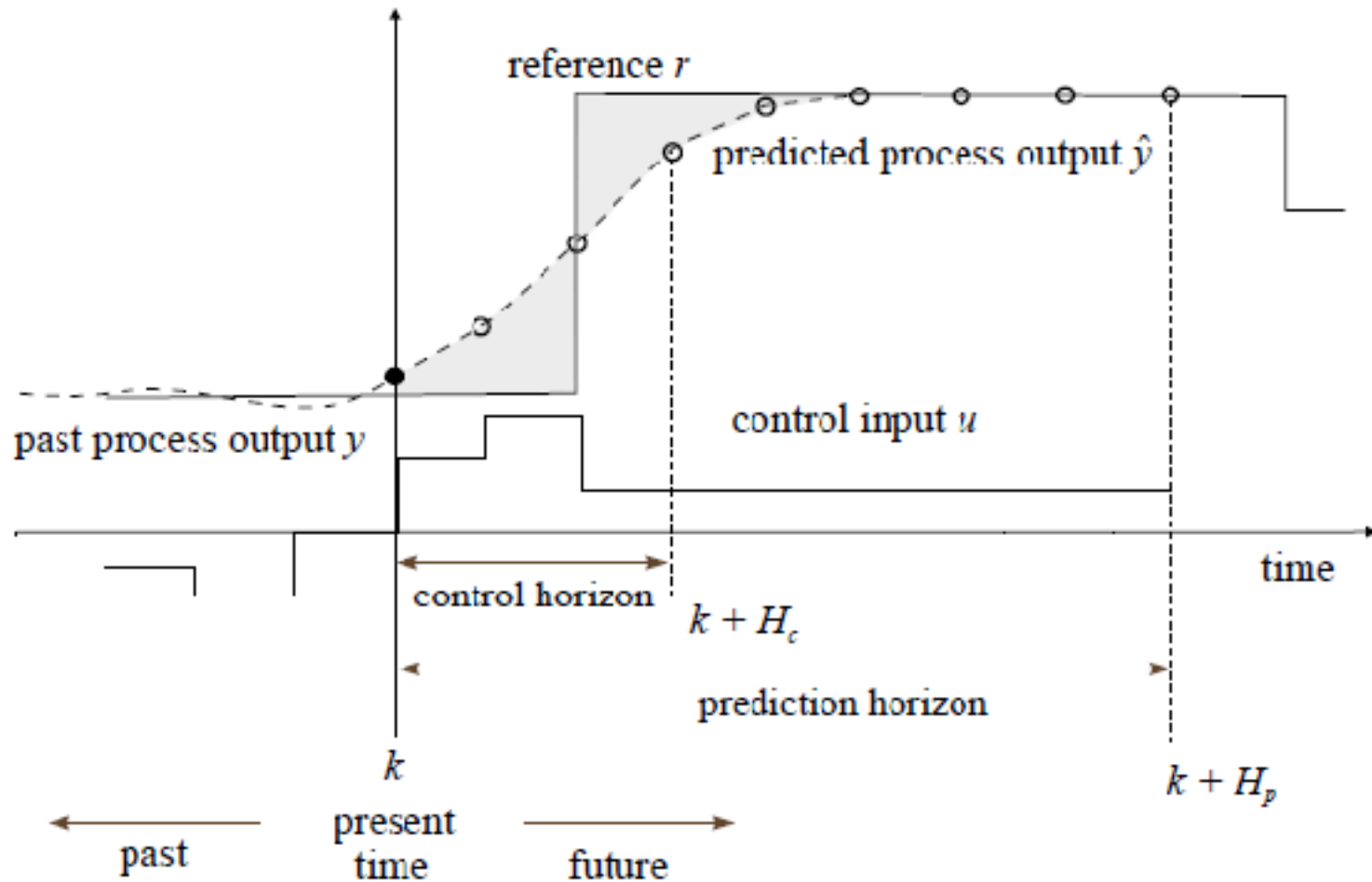
- Se utiliza el modelo de la planta para predecir el comportamiento futuro.
- La red neuronal predice la respuesta de la planta en un horizonte específico.
- Utiliza un algoritmo de optimización para seleccionar la señal de control que optimiza el desempeño futuro.



# Control Predictivo



# Control Predictivo



# Control Predictivo

- La predicción utiliza optimización numérica para determinar la señal de control que minimiza una función de desempeño.

$$J = \sum_{j=N_1}^{N_2} (y_r(t+j) - y_m(t+j))^2 + \rho \sum_{j=1}^{N_u} (u'(t+j-1) - u'(t+j-2))^2$$

- $N_1$ ,  $N_2$ , y  $N_u$  definen el horizonte
- $u'$ : es la señal de control tentativa
- $y_r$ : es la respuesta deseada
- $y_m$ : es la respuesta del modelo neuronal
- $\rho$ : determina el peso de la suma al cuadrado de los incrementos de control

# Control NARMA-L2

- Nonlinear Autoregressive-Moving Average-L2
- **Linealización por realimentación:** La idea central de este tipo de control es transformar las dinámicas no lineales en dinámicas lineales cancelando las no linealidades.

# Linealización por realimentación

- **Ejemplo:** Modelo dinámico de un robot manipulador,

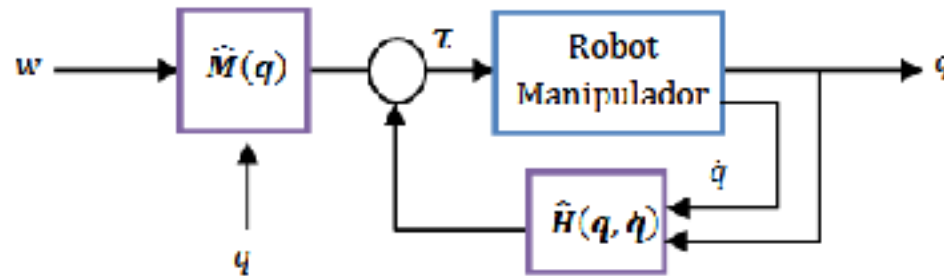
$$\mathbf{M}(q)\ddot{q} + \mathbf{H}(q, \dot{q}) = \tau$$

- En (Khalil & Dombre, 2002) se propone la siguiente linealización:

$$\hat{\mathbf{M}}(q)\mathbf{w}(t) + \hat{\mathbf{H}}(q, \dot{q}) = \tau$$

- donde  $\mathbf{w}(t)$  representa un nuevo vector de control, siendo  $\hat{\mathbf{M}}$  y  $\hat{\mathbf{H}}$  las estimaciones respectivas de  $\mathbf{M}$  y  $\mathbf{H}$ . Cuando el modelo dinámico es perfectamente conocido, se tiene que  $\hat{\mathbf{M}} = \mathbf{M}$  y  $\hat{\mathbf{H}} = \mathbf{H}$ .

# Linealización por realimentación



$$\hat{M}(q)w(t) + \hat{H}(q, \dot{q}) = M(q)\ddot{q} + H(q, \dot{q})$$

$$\ddot{q} = w(t)$$

Considerando el caso ideal de ausencia de disturbios y conocimiento perfecto del modelo, se obtiene que el problema se reduce al control de  $n$  sistemas lineales, desacoplados y de segundo orden (doble integrador).

# Control NARMA-L2

Un modelo NARMA es utilizado para representar un sistema no lineal.

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

Donde  $u(k)$  es la entrada y  $y(k)$  es la salida.

El proceso de identificación consiste en identificar  $N$ .

# Control NARMA-L2

Si se desea que el sistema siga una trayectoria de referencia,

$$y(k + d) = y_r(k + d)$$

Se utiliza un modelo que aproxima al sistema de la forma,

$$y = f + gu$$

$$\hat{y} = y_r = \hat{f} + \hat{g}u(k)$$

Donde la ley de control es,

$$u(k) = \frac{y_r - \hat{f}}{\hat{g}}$$



# Control NARMA-L2

El modelo del sistema es,

$$y = f + gu$$

El modelo del controlador es,

$$u = \frac{y_r - \hat{f}}{\hat{g}}$$

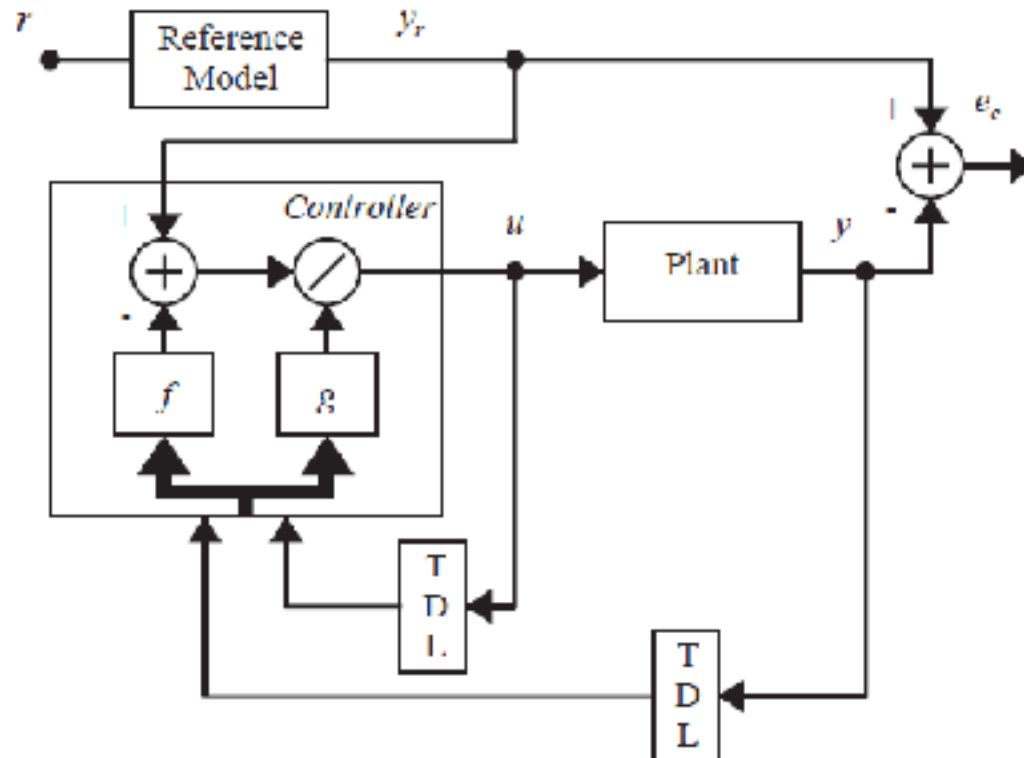
Reemplazando la ley de control  $u$  en el modelo,

$$y = f + g \left( \frac{y_r - \hat{f}}{\hat{g}} \right)$$

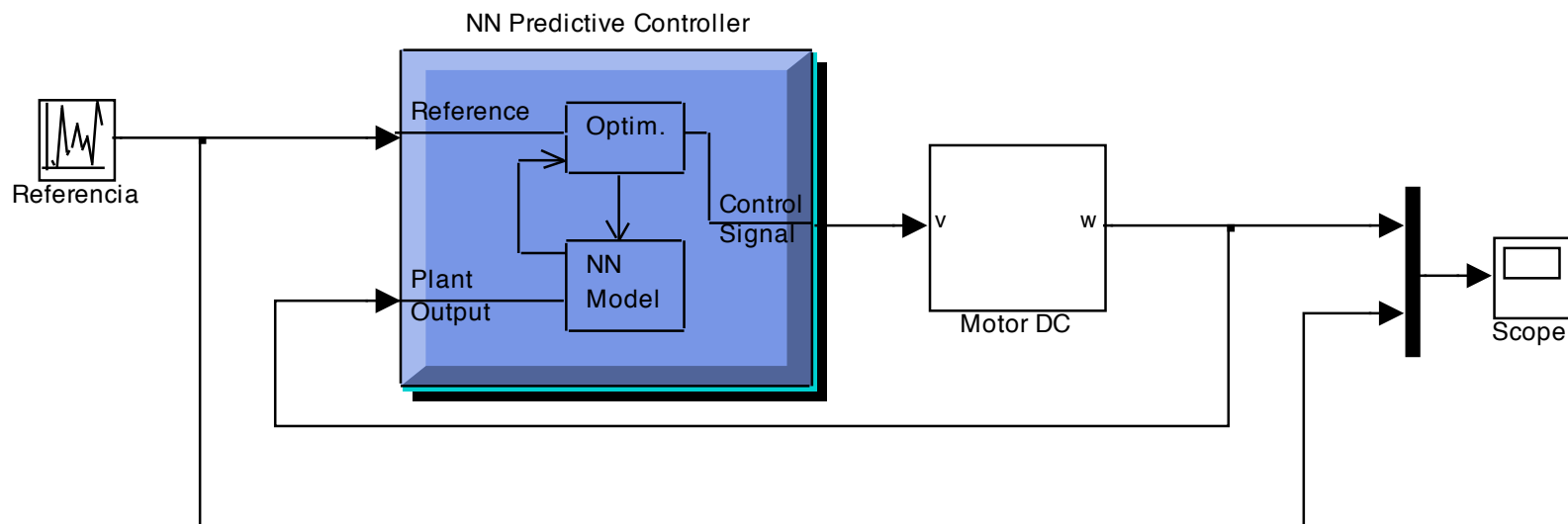
Si  $f = \hat{f}$  y  $g = \hat{g}$ , notar que  $y = y_r$

# Control NARMA-L2

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



# Control predictivo motor DC



# Parámetros para identificación del modelo

The image shows a MATLAB/Simulink dialog box titled "Plant Identification". It is divided into three main sections: Network Architecture, Training Data, and Training Parameters.

**Network Architecture**

- Size of Hidden Layer: 7
- No. Delayed Plant Inputs: 2
- Sampling Interval (sec): 0.01
- No. Delayed Plant Outputs: 2
- ☐ Normalize Training Data

**Training Data**

- Training Samples: 1000
- ☒ Limit Output Data
- Maximum Plant Input: 1.5
- Maximum Plant Output: 5
- Minimum Plant Input: 0
- Minimum Plant Output: 0
- Maximum Interval Value (sec): 0.4
- Simulink Plant Model: Browse...
- Minimum Interval Value (sec): 0.2
- Model to Train: ModeloMotor

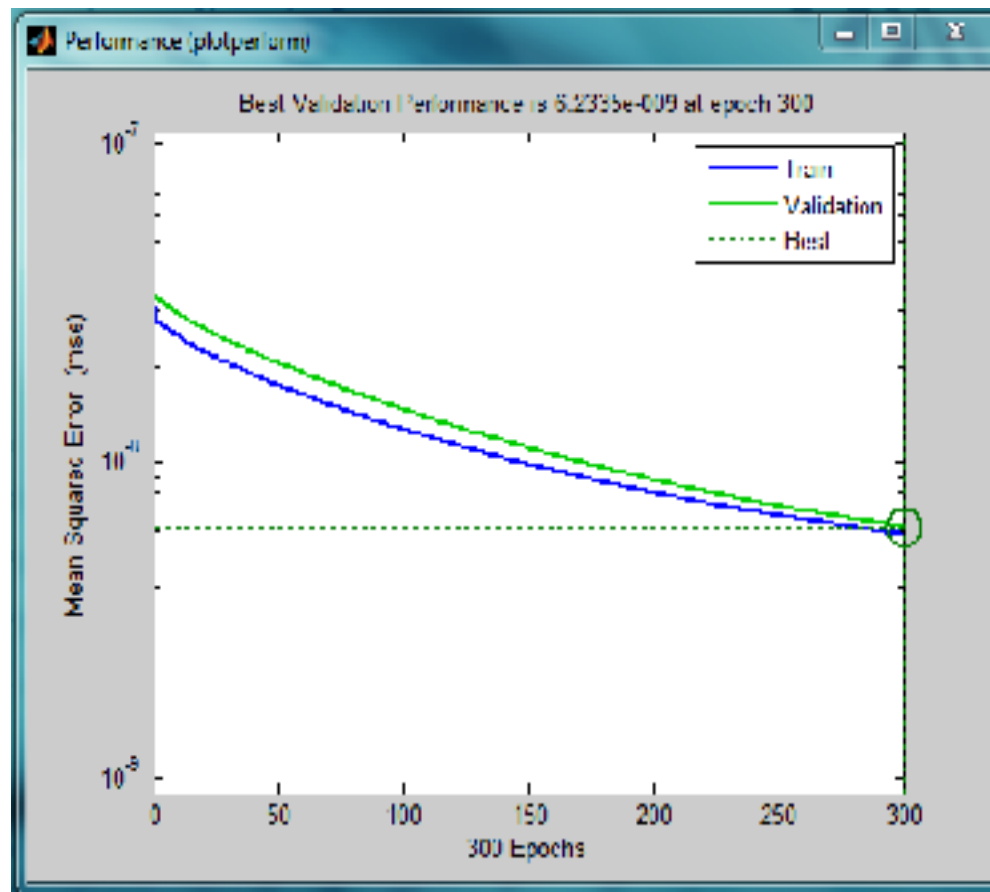
**Training Parameters**

- Training Epochs: 300
- Training Function: trainlm
- ☒ Use Current Weights
- ☒ Use Validation Data
- ☐ Use Testing Data

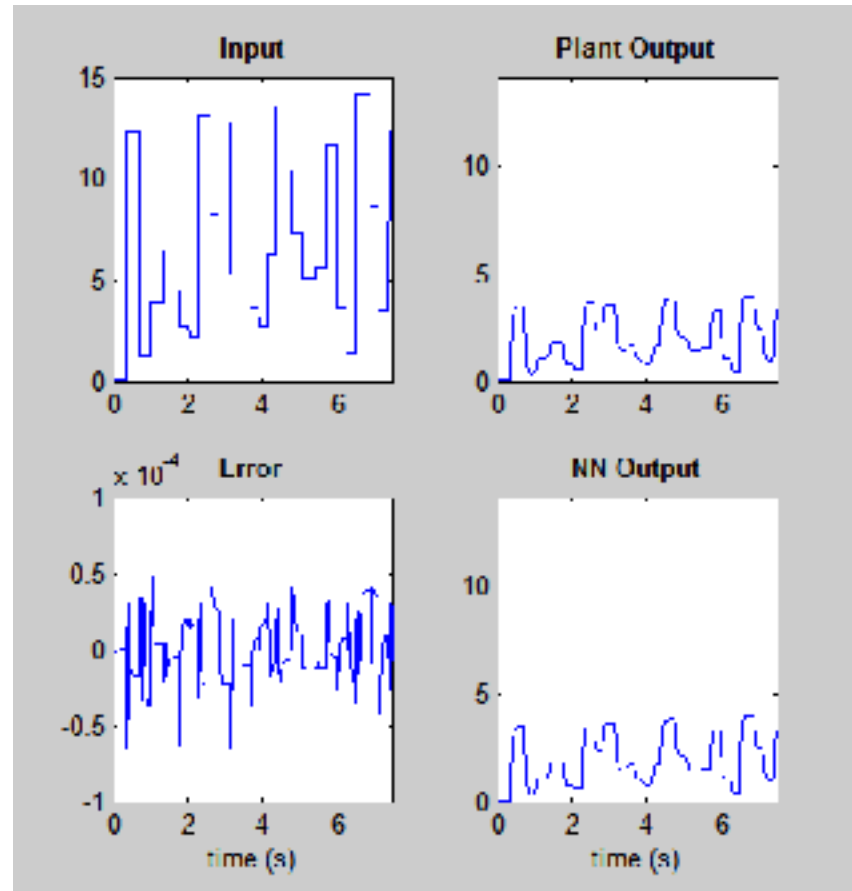
Buttons: Generate Training Data, Import Data, Export Data, Train Network, OK, Cancel, Apply.

Footer text: Generate or import data before training the neural network plant.

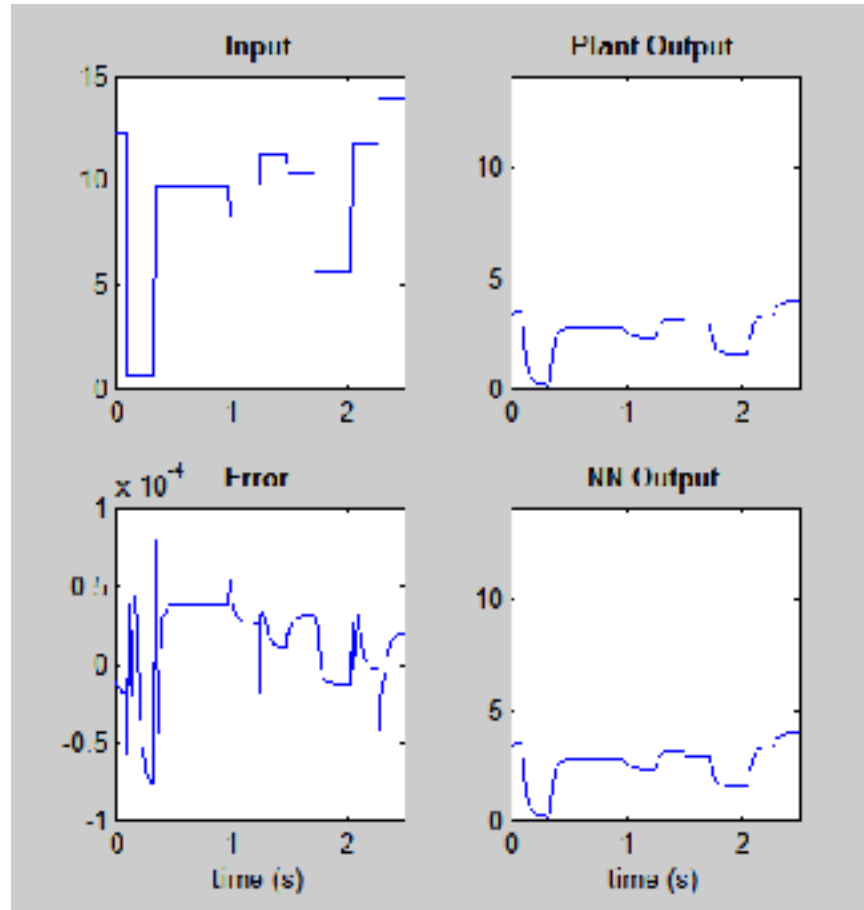
# Error alcanzado en 300 epoch



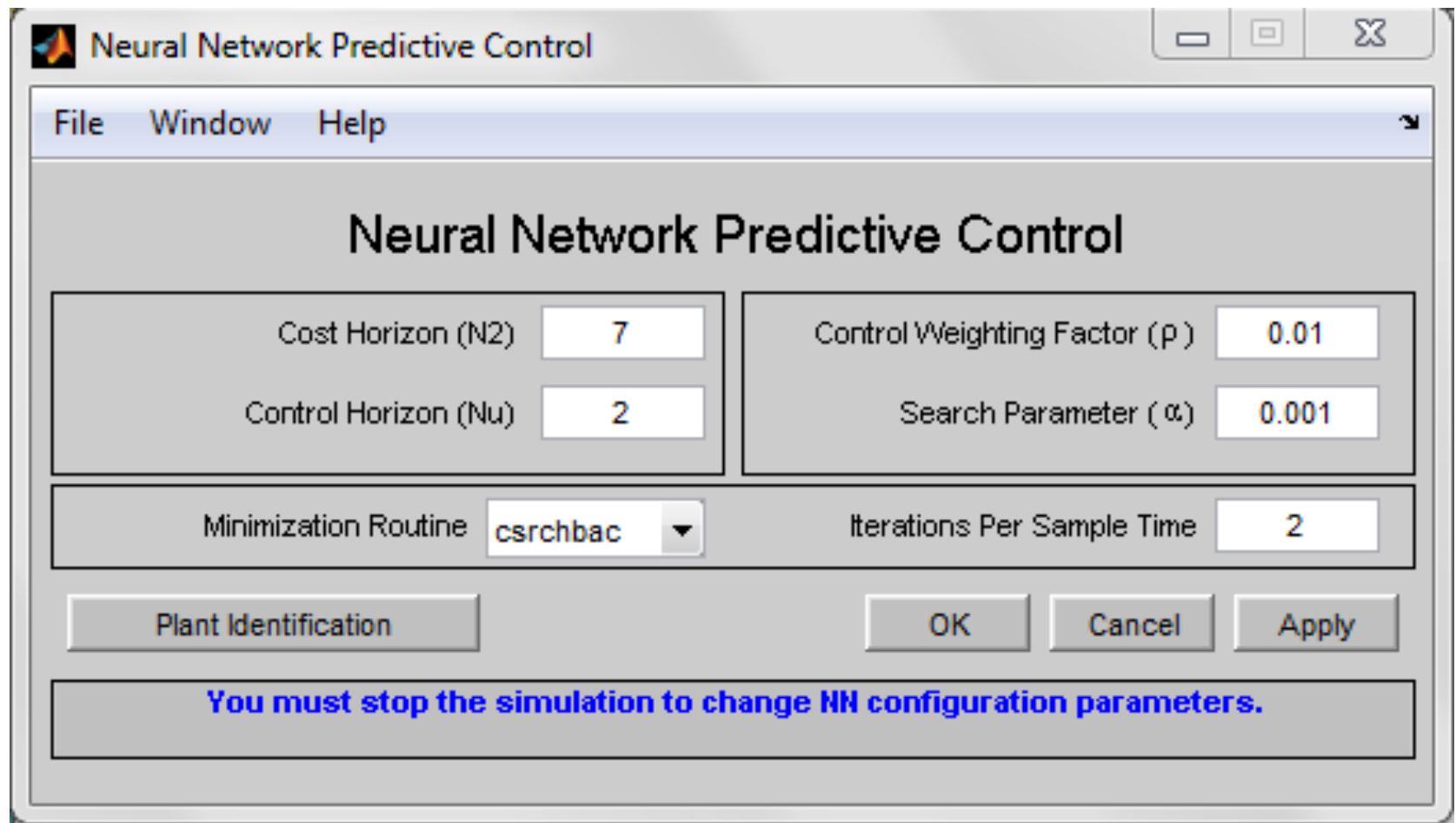
# Resultados del entrenamiento



# Resultado Datos de Validación



# Parámetros controlador predictivo RN



The image shows a software window titled "Neural Network Predictive Control". It features a menu bar with "File", "Window", and "Help". The main area is titled "Neural Network Predictive Control" and contains several input fields and buttons. The "Cost Horizon (N2)" is set to 7, and the "Control Horizon (Nu)" is set to 2. The "Control Weighting Factor ( $\rho$ )" is set to 0.01, and the "Search Parameter ( $\alpha$ )" is set to 0.001. The "Minimization Routine" is set to "csrchbac" via a dropdown menu, and the "Iterations Per Sample Time" is set to 2. There are buttons for "Plant Identification", "OK", "Cancel", and "Apply". A message at the bottom states: "You must stop the simulation to change NN configuration parameters."

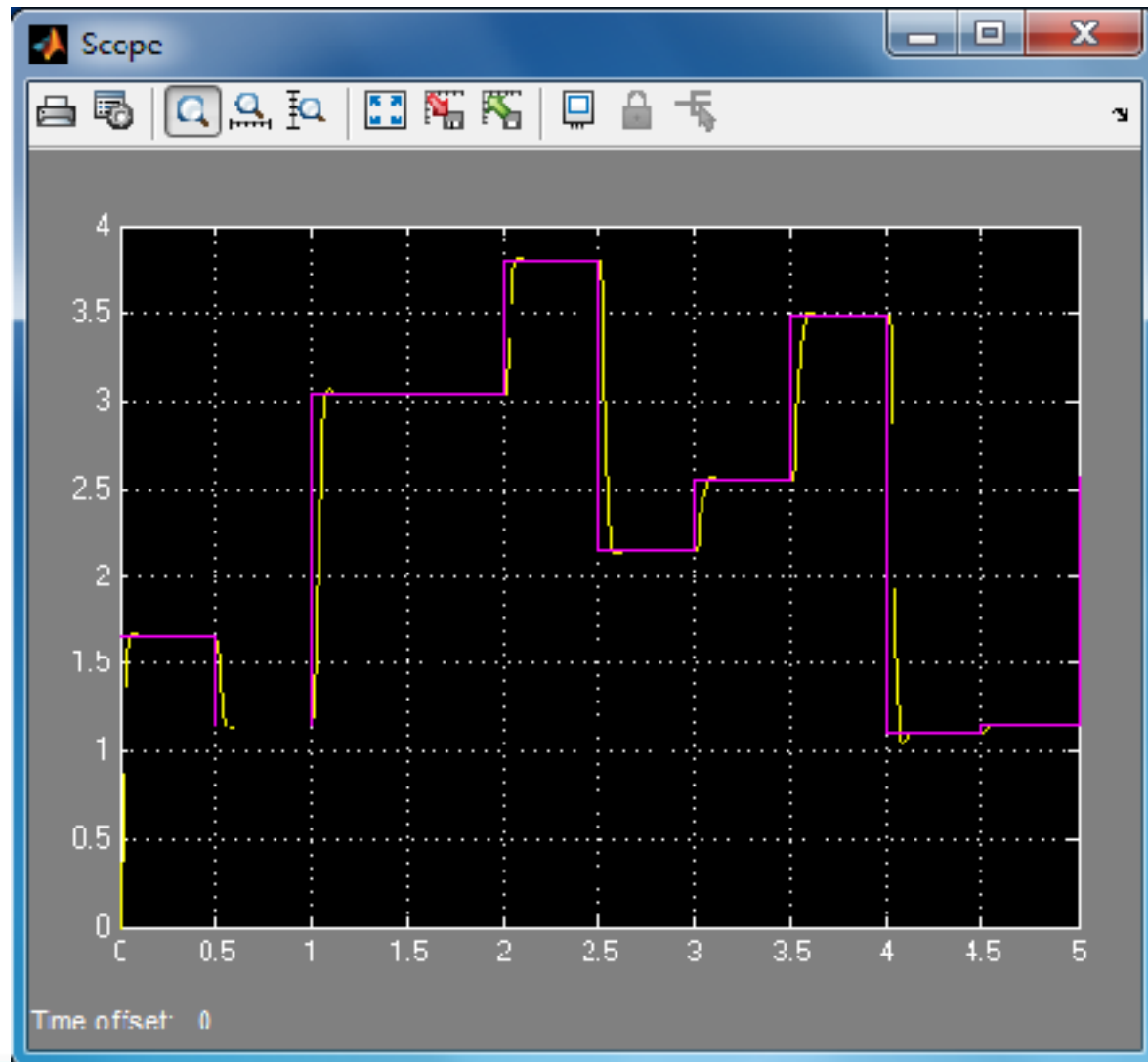
| Parameter                           | Value    |
|-------------------------------------|----------|
| Cost Horizon (N2)                   | 7        |
| Control Horizon (Nu)                | 2        |
| Control Weighting Factor ( $\rho$ ) | 0.01     |
| Search Parameter ( $\alpha$ )       | 0.001    |
| Minimization Routine                | csrchbac |
| Iterations Per Sample Time          | 2        |

Buttons: Plant Identification, OK, Cancel, Apply

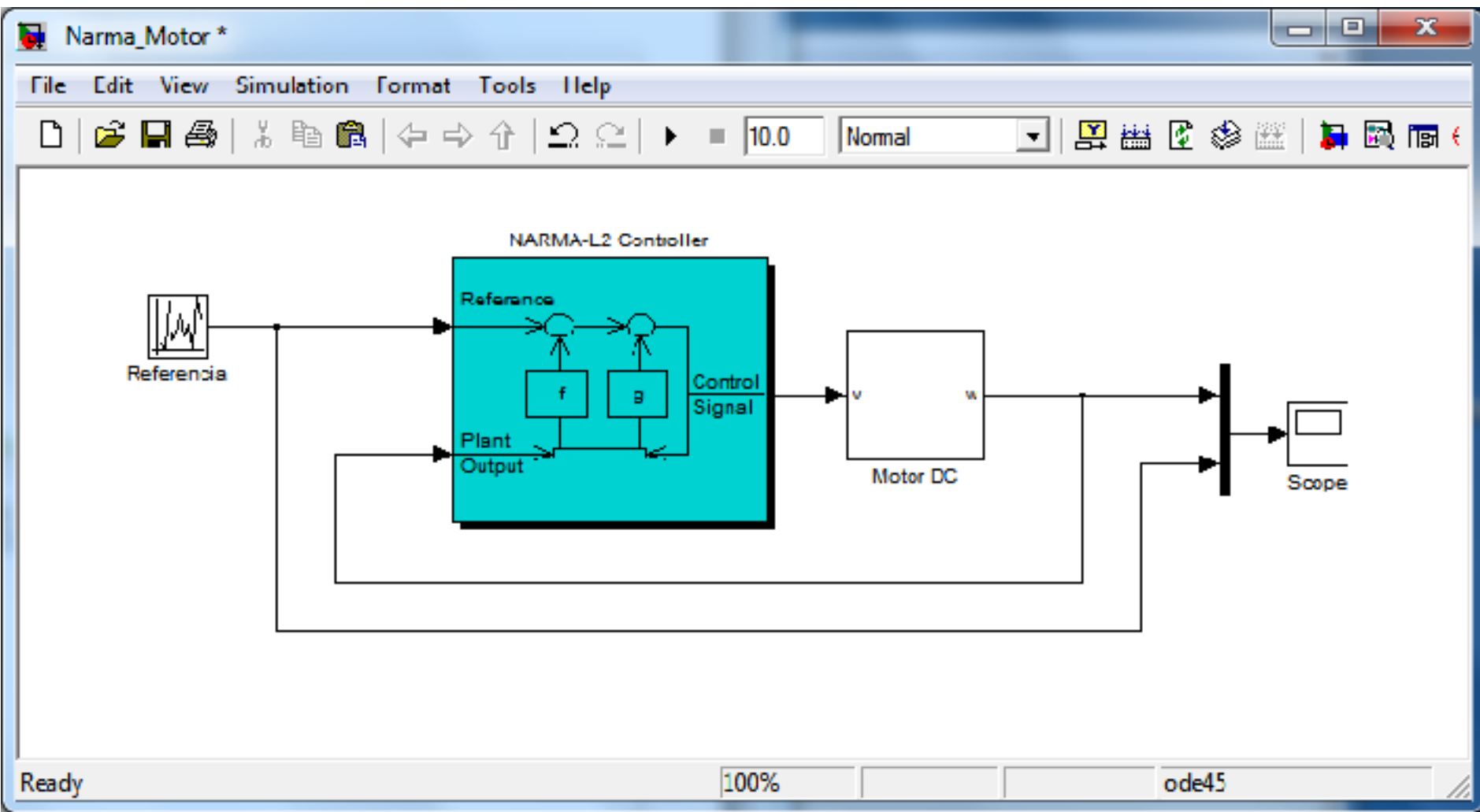
**You must stop the simulation to change NN configuration parameters.**



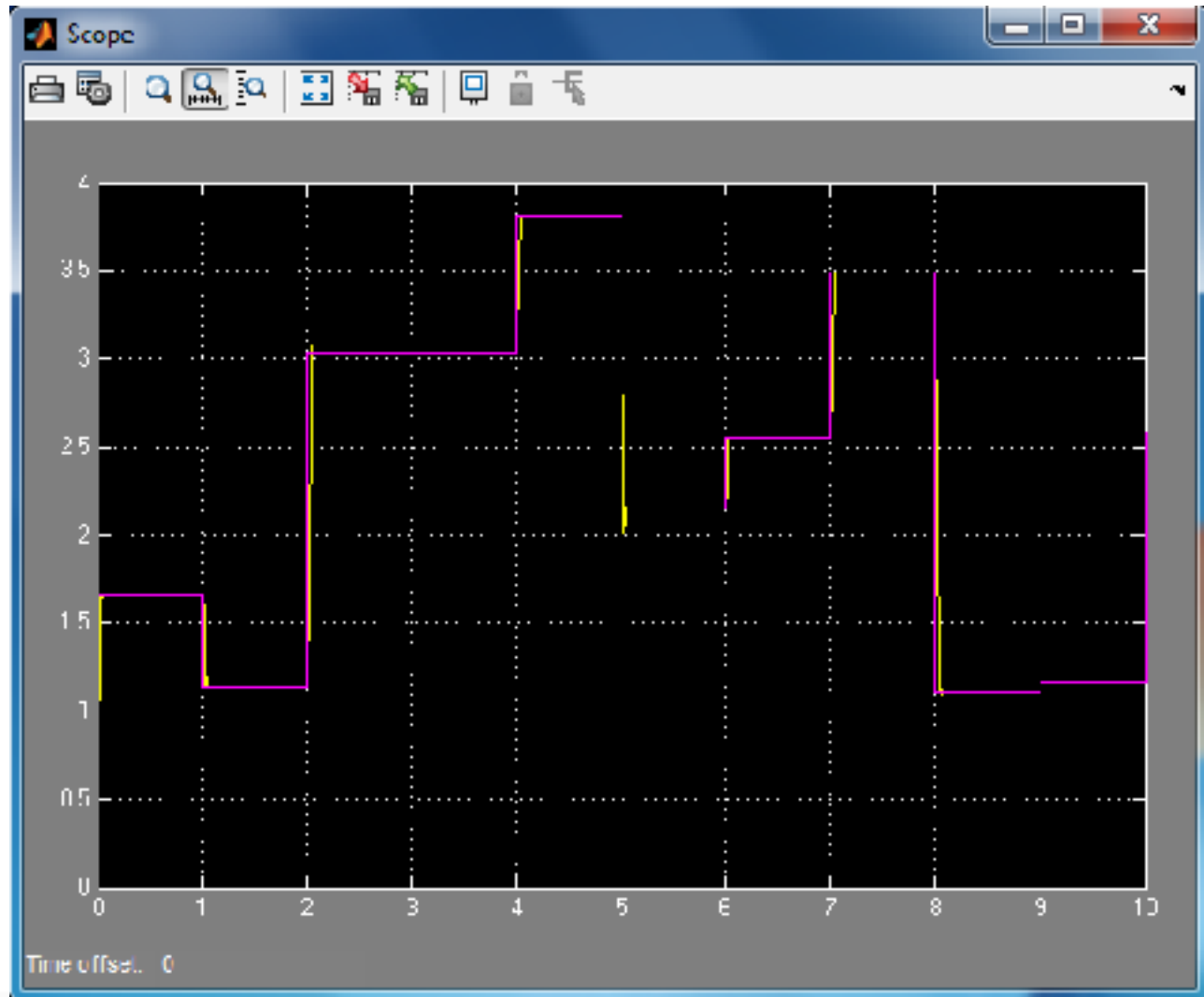
# Respuesta del controlador predictivo RN



# Control Narma – L2



# Respuesta del controlador



# ¿ Preguntas ?

