Neuronal Implementation of Predictive Controllers

Computational Intelligence Group

Basque Country University (Spain)

José Manuel López-Guede, Ekaitz Zulueta, Manuel Graña, Alicia D'Anjou

This work was supported in part by the Spanish Ministerio de Educación y Ciencia, DPI2006-15346-C03-03.

Index

1. Introduction

- Well known control schemas as PID controllers could have problems to control some systems.
- There are techniques that emulate the human brain that could control these systems: Model Predictive Control.
- These techniques have some drawbacks: they are very computational expensive.

2. Objectives

- Use Predictive Controllers to control complex systems that classics controllers can’t.
- The main objective is get a computational inexpensive implementation of Predictive Controllers.
- Get a fast and cheap implementation.

3. Model Predictive Control (MPC)

- Is it really necessary?
- We tried to control a system using a discrete version of PID:

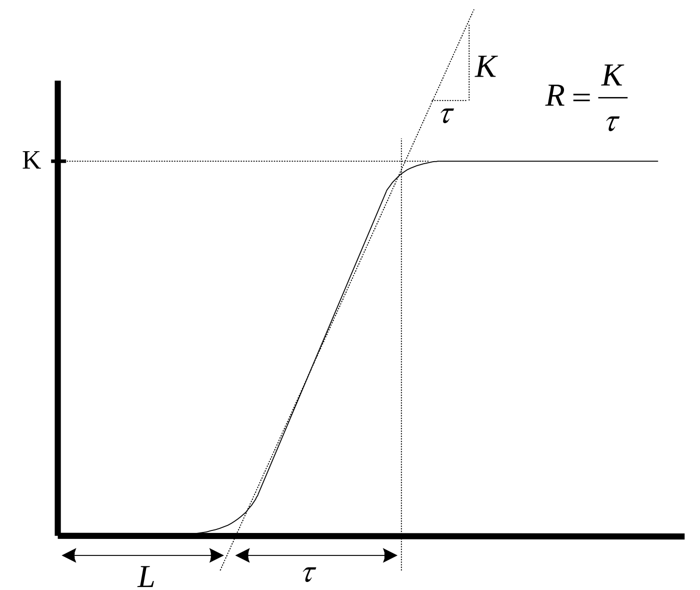
$$u_{PID}(t) = K_P\left[e(t) + \frac{1}{T_I}\int_0^t e(t)\,dt + T_D\,\frac{de(t)}{dt}\right]$$

$$C_{PID}(z) = K_P.\left[1 + T_d\,\frac{(z-1)}{z\,T_m} + \frac{1}{T_I}\,\frac{T_m}{2}\,\frac{(z+1)}{(z-1)}\right]$$

3. Model Predictive Control (MPC)

- Parameter adjust: Ziegler-Nichols

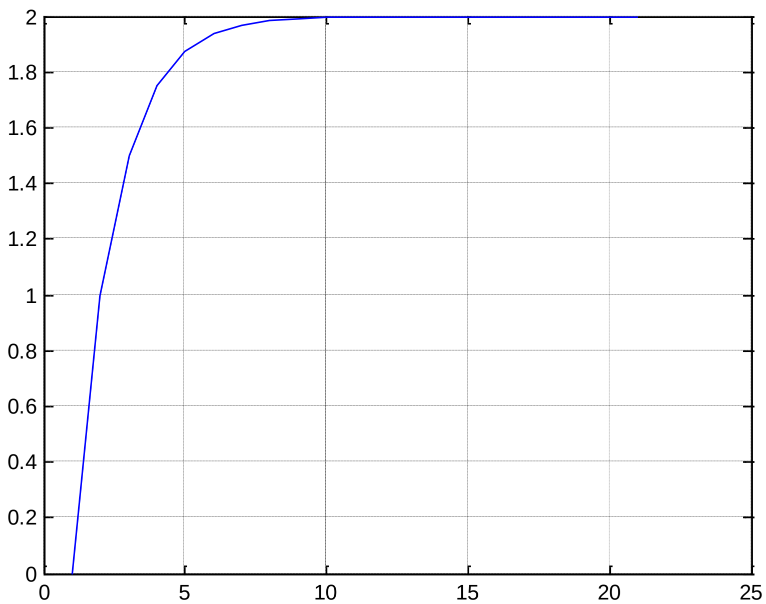
	K_P	T_I	T_D
P	$1/RL$		
PI	$0.9/RL$	$3L$	
PID	$1.2/RL$	$2L$	$0.5L$



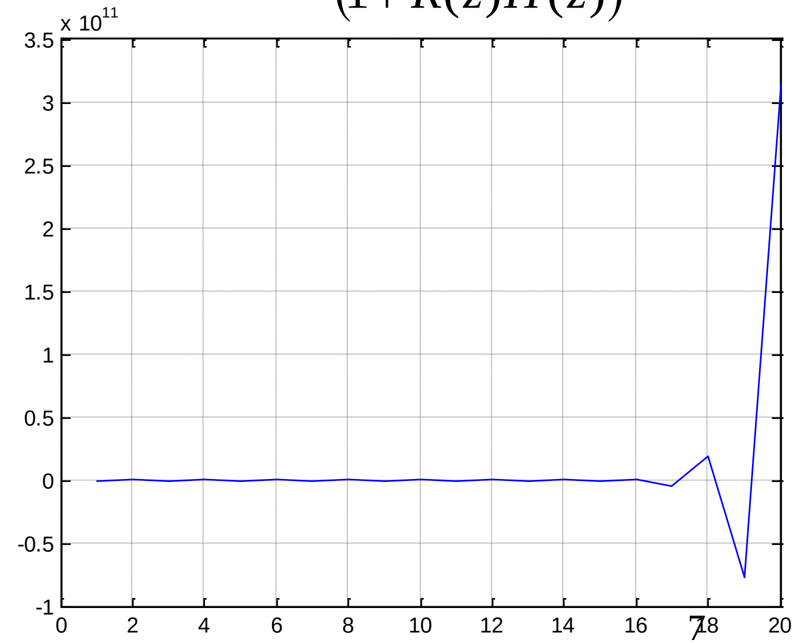
3. Model Predictive Control (MPC)

- Example: $H(z)=\dfrac{1}{z-0.5}$ (stable)

Output H(z)



Output $\dfrac{R(z)H(z)}{\left(1+R(z)H(z)\right)}$



3. Model Predictive Control (MPC)

- It works like human brain:
  - It doesn’t use past error between the output of the system and the desired value.
  - It predicts the value of the output in a short time.
  - It generates a signal to get that the output of the system was as closer as possible of the desired value.

3. Model Predictive Control (MPC)

- A set of techniques that use:
  - Plant model, to get a prediction of the system’s output over a prediction horizon *p*,
  - Objective function to minimize,
  - Control law to minimize the objective function over the prediction horizon *p* using actions in the control horizon *m*, generally $m<p$.

3. Model Predictive Control (MPC)

- Advantages:
  - It is an open methodology,
  - It can include constraints,
  - Generalization of MIMO systems.

- Drawbacks:
  - It is computational expensive in its tuning phase,
  - It is computational expensive in its working phase.

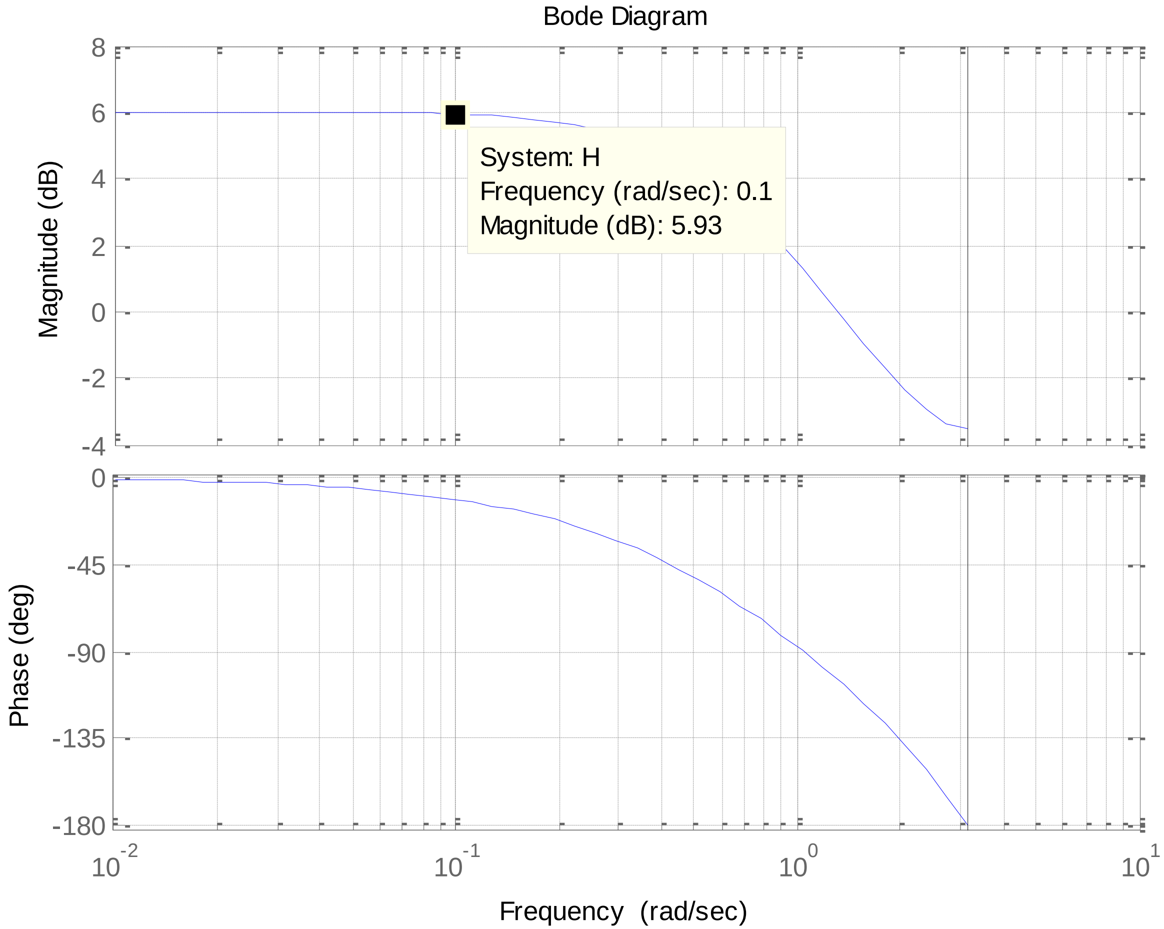
4. Dynamic Matrix Control (DMC)

- It’s a concrete MPC technique:
  - Subsystem model: Step response.
  - Objective function: measures the difference between the reference signal and the predicted output.
  - Control law:

$$\Delta u = \left(G^t G + \lambda I\right)^{-1} G^t \left(w - f\right)$$

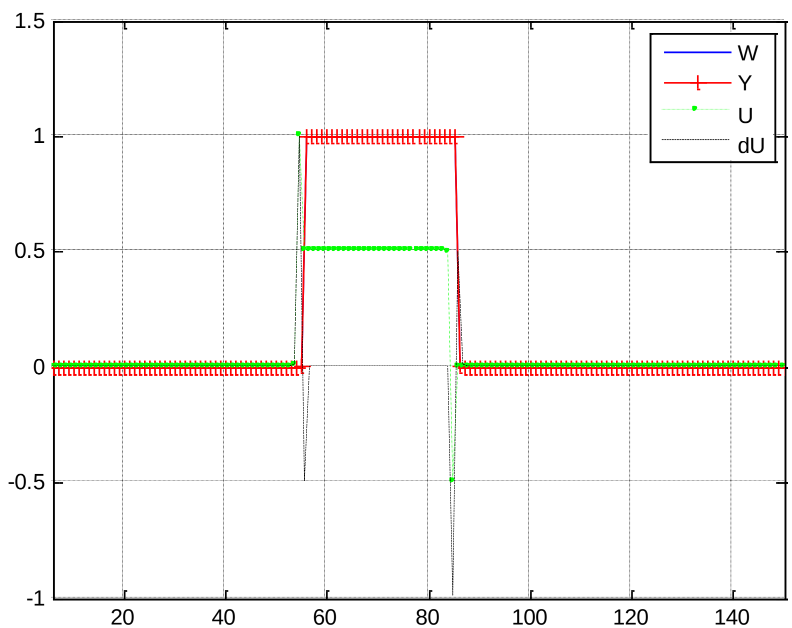
4. Dynamic Matrix Control (DMC)

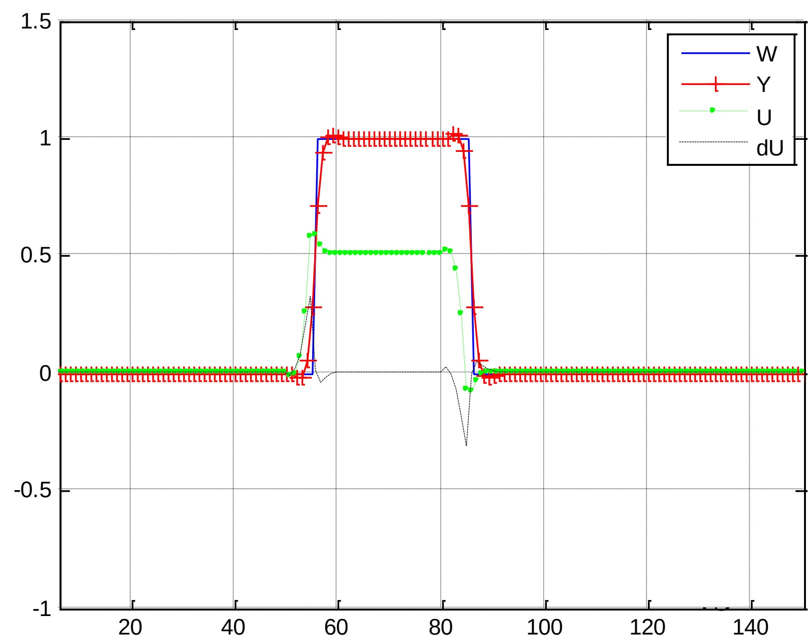
- Working point: $H(z) = \frac{1}{z - 0.5}$



4. Dynamic Matrix Control (DMC)

- Parameters: p=5, m=3
  - λ=0.001
    mce = 2.1e-007
    Mp = 0.06 %
  - λ=1
    mce = 0.005
    Mp = 1.8 %





4. Dynamic Matrix Control (DMC)

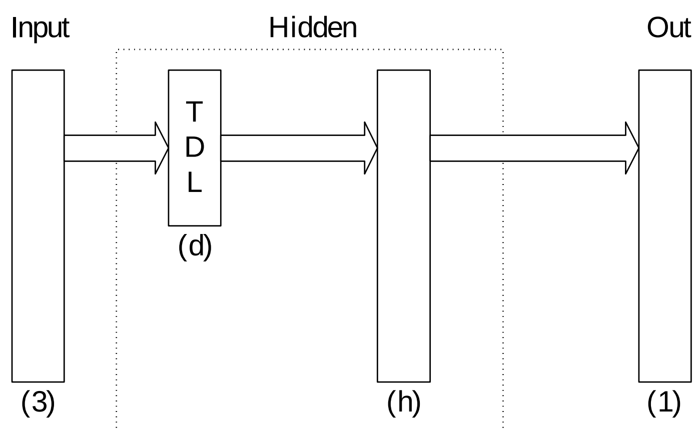
- But it has the MPC general drawbacks:
  - It is computational expensive in its tuning phase, but it is carried out only one time.
  - It is computational expensive in its working phase:
    - Each sample: $\Delta u = \left(G^t G + \lambda I\right)^{-1} G^t \left(w - f\right)$
    - It obtains a set of *m* signals, but only the first of them is used in this sample time, the rest are ignored.

5. Time Delayed N. N. (TDNN)

- Main characteristics:
  - They are a kind of multi-layer perceptron neural networks.
  - They are dynamics.
  - Delayed versions of the input signals are introduced to the input.
  - They are ANNs (fast and generalizing responses).
- We use TDNN to model a tuned Predictive Controller.

5. Time Delayed N. N. (TDNN)

- Structural parameters:
  - Hidden layers: 1,
  - Size of the time delay line,
  - Number of neurons of the hidden layer.



6. Results

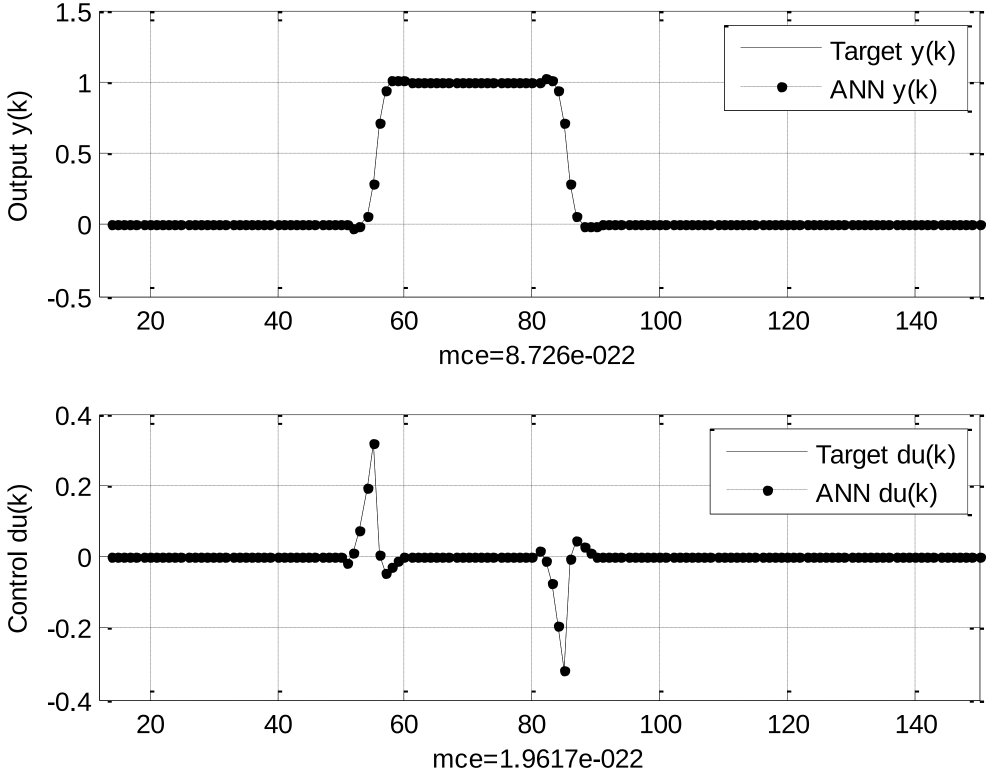
- Training experiments with multiple structures, varying:
  - Number of hidden layer neurons h.
  - Number of delays of the time delay line d.
- The Levenberg-Marquardt method has been used to carry out the training:
  - Target vector: $P = [w(k)\,,\; y(k)\,,\; \Delta u(k-1)]'$
  - Output: $\Delta u(k)$

6. Results

- The control of neuronal controller is right even with noisy references that haven't been used in the training phase.
- The chosen structure:
  - Number of hidden layer neurons $h = 5$.
  - Number of delays of the time delay line $d = 7$.

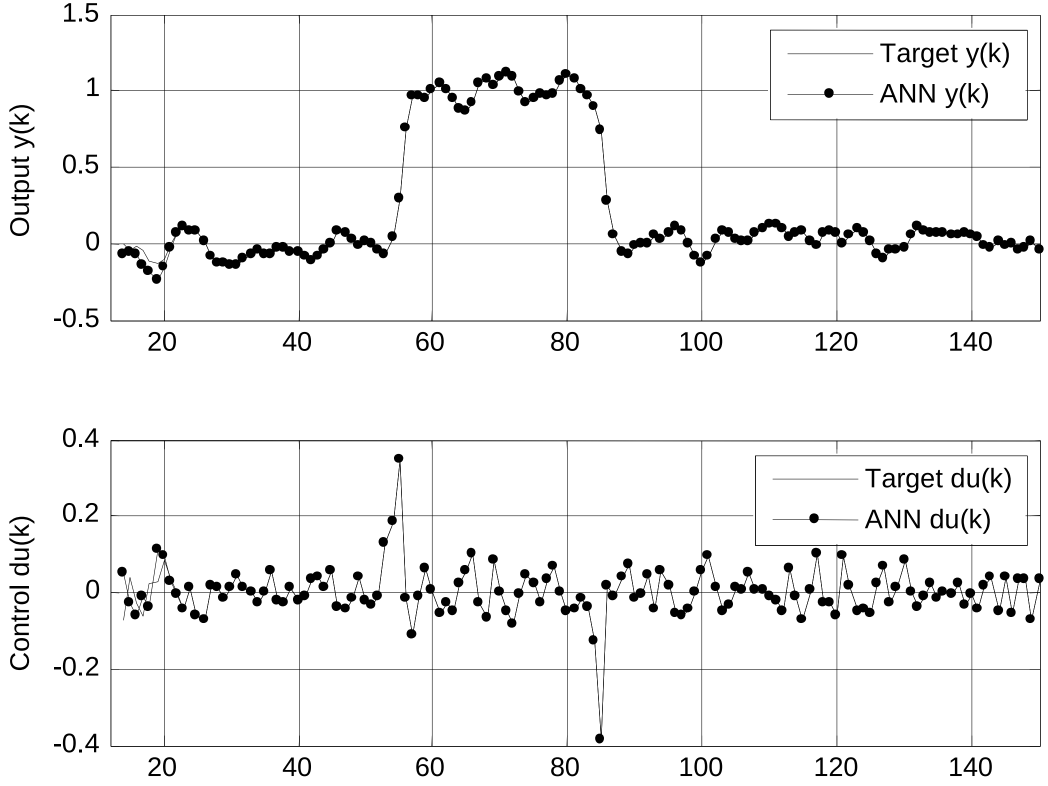
6. Results

- Example 1: the reference to follow has been used in the training phase.



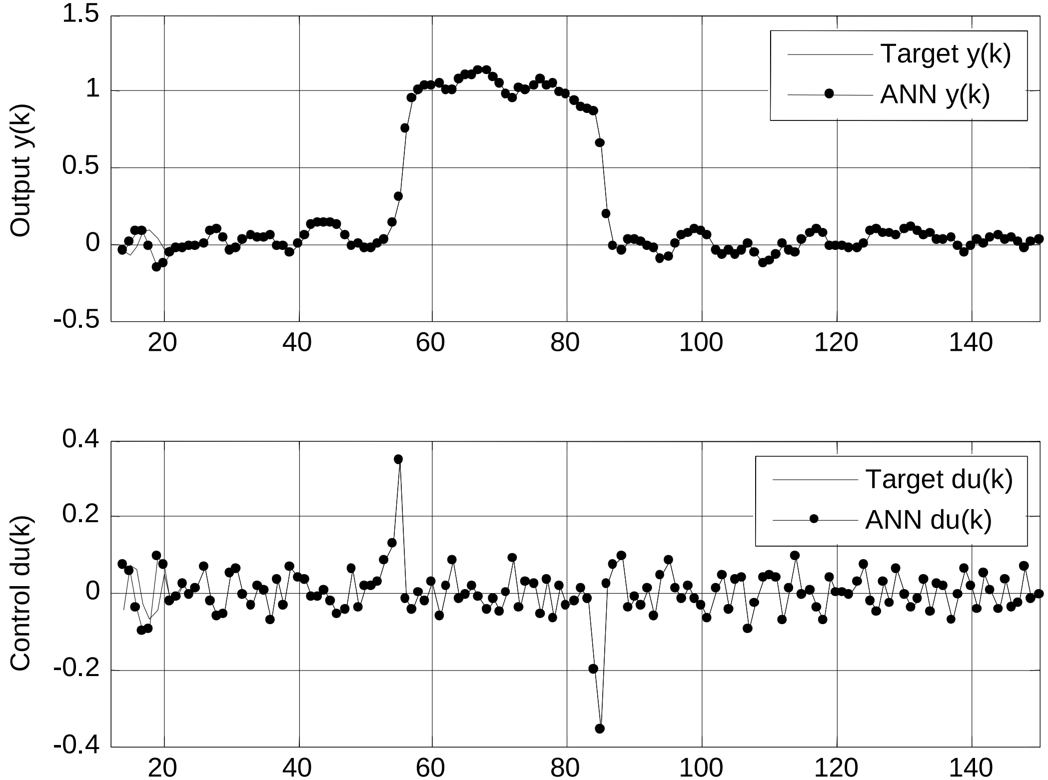
6. Results

- Example 2: the noisy reference to follow hasn't been used in the training phase.



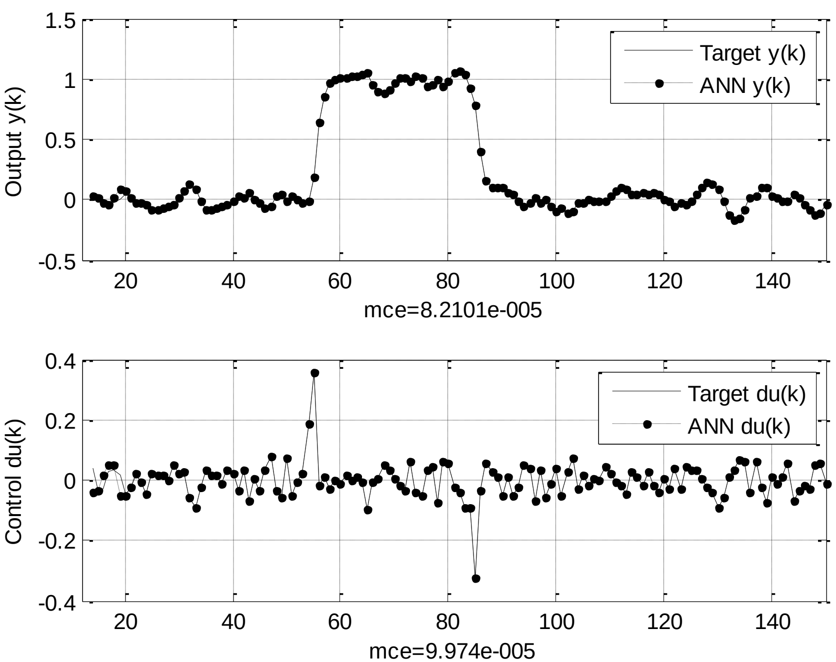
6. Results

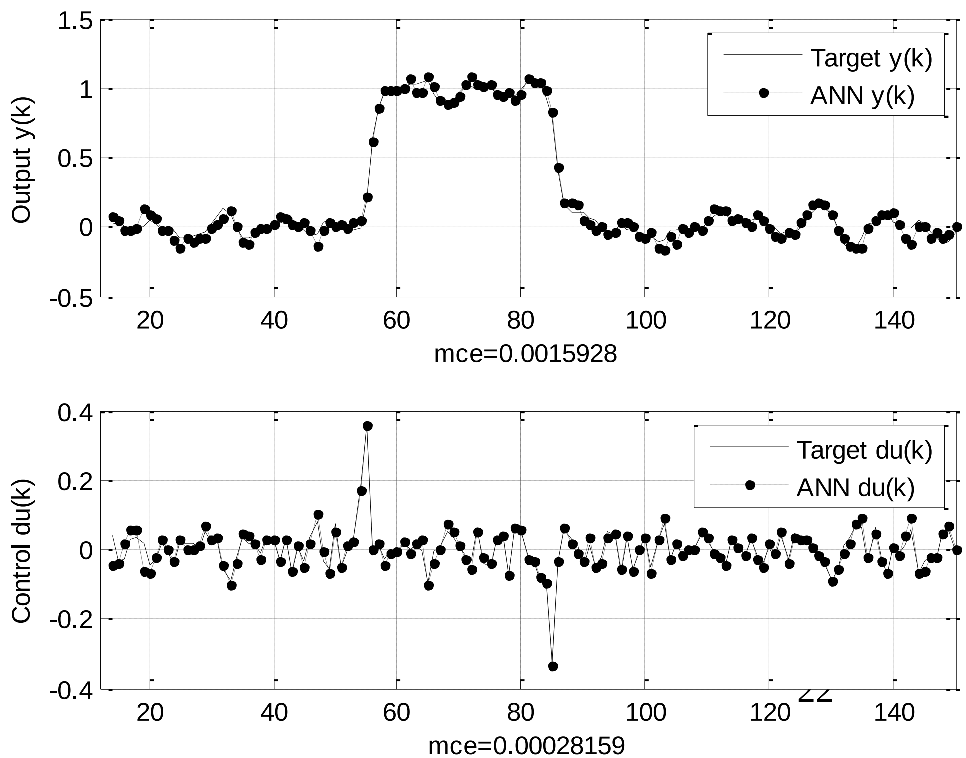
- <u>Example 3</u>: the noisy reference to follow hasn't been used in the training phase.



6. Results

- Example 4: Presence of control and measurement perturbations, white noise of mean zero and variance $\sigma^2 = 10^{-3}$





7. Conclusions

- Predictive Control is a technique that can control systems that classic controllers can’t.
- Time Delayed Neural Networks are a kind of ANN that can model Dynamic Matrix Controllers.

7. Conclusions

- In this way, we can overcome the main drawback of these kind of controllers, including:
  - New situations that haven't been used in the training phase,
  - The existence of perturbations in the control and/or measurement signals.

Thanks.

Questions?