ISTANBUL TECHNICAL UNIVERSITY

INTELLIGENT CONTROL SYSTEMS

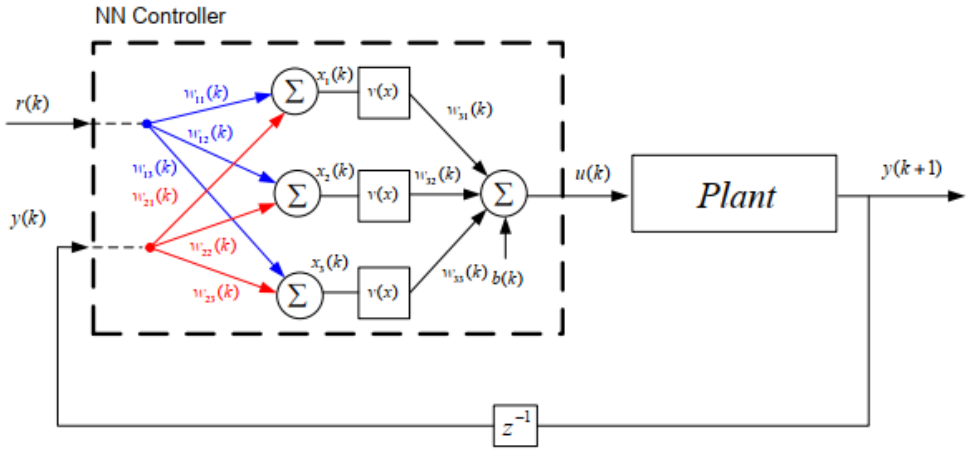
KON-426E

HOMEWORK II

ECEM IŞILDAR

040170450

Question 1: Neural Network structure can be used as a controller directly as in fig.1. The performance of the controller is going to be evaluated on the following nonlinear time-varying plant :



$$y(k+1)=\frac{1.2(1-0.8e^{-0.1k})\ y(k)}{1+\ y^2(k)}+u(k)$$

a) Give the details related to the update rules for neural network controller parameters using chain rule so as to minimize the following objective function

First the forward phase is done:

$$x_1(k) = r(k) * w_{11}(k) + y(k) * w_{21}$$

$$x_2(k) = r(k) * w_{12}(k) + y(k) * w_{22}$$

$$x_3(k) = r(k) * w_{13}(k) + y(k) * w_{23}$$

Also, u(k) is calculated:

$$u(k) = w_{31}(k) * v(x_1) + w_{32}(k) * v(x_2) + w_{33}(k) * v(x_3) + b(k)$$

Update rules of the neural network controller consists of two parts, first it should be updated the weights in the output layer $w_{31}(k), w_{32}(k), w_{33}(k)$ are the output weights. The second part is updating the weights in the input layer, the weights $w_{11}(k), w_{12}(k), w_{13}(k), w_{21}(k), w_{22}(k), w_{23}(k)$ are updated in the second part of the update rule.

For the output layer:

$$\Delta w_{31}(k) = -\eta \frac{\partial J(k)}{\partial e(k)} * \frac{\partial e(k)}{\partial y(k+1)} * \frac{\partial y(k+1)}{\partial u(k)} * \frac{\partial u(k)}{\partial v(k)} * \frac{\partial v(k)}{\partial w_{31}(k)}$$

$$\Delta w_{32}(k) = -\eta \frac{\partial J(k)}{\partial e(k)} * \frac{\partial e(k)}{\partial y(k+1)} * \frac{\partial y(k+1)}{\partial u(k)} * \frac{\partial u(k)}{\partial v(k)} * \frac{\partial v(k)}{\partial w_{32}(k)}$$

$$\Delta w_{33}(k) = -\eta \frac{\partial J(k)}{\partial e(k)} * \frac{\partial e(k)}{\partial y(k+1)} * \frac{\partial y(k+1)}{\partial u(k)} * \frac{\partial u(k)}{\partial v(k)} * \frac{\partial v(k)}{\partial w_{33}(k)}$$

The results of the derivatives above are given in below:

$$\frac{\partial J(k)}{\partial e(k)} = e(k), \quad \frac{\partial e(k)}{\partial y(k+1)} = -1, \quad \frac{\partial y(k+1)}{\partial u(k)} = \frac{y(k) - y(k-1)}{u(k) - u(k-1)}, \quad \frac{\partial u(k)}{\partial v(k)} = 1 - v(x)^2, \quad \frac{\partial v(k)}{\partial w_{33}(k)} = x_1$$

Then the new weights can be calculated:

$$w_{31}(k+1) = w_{31}(k) + \Delta w_{31}$$
$$w_{32}(k+1) = w_{32}(k) + \Delta w_{32}$$
$$w_{33}(k+1) = w_{33}(k) + \Delta w_{33}$$

Then $w_{11}(k), w_{12}(k), w_{13}(k), w_{21}(k), w_{22}(k), w_{23}(k)$ can be calculated using the formula below:

First 3 weights are updated using input r(k);

$$\Delta w_{11}(k) = \eta * e(k) * \frac{\partial v_1(k)}{\partial x(k)} * w_{31} * r(k)$$

$$\Delta w_{12}(k) = \eta * e(k) * \frac{\partial v_2(k)}{\partial x(k)} * w_{32} * r(k)$$

$$\Delta w_{13}(k) = \eta * e(k) * \frac{\partial v_3(k)}{\partial x(k)} * w_{33} * r(k)$$

Second 3 weights are updated using y(k);

$$\Delta w_{21}(k) = \eta * e(k) * \frac{\partial v_1(k)}{\partial x(k)} * w_{31} * y(k)$$

$$\Delta w_{22}(k) = \eta * e(k) * \frac{\partial v_2(k)}{\partial x(k)} * w_{32} * y(k)$$

$$\Delta w_{23}(k) = \eta * e(k) * \frac{\partial v_3(k)}{\partial x(k)} * w_{33} * y(k)$$

$$w_{11}(k+1) = w_{11}(k) + \Delta w_{11}$$
$$w_{12}(k+1) = w_{12}(k) + \Delta w_{12}$$
$$w_{13}(k+1) = w_{13}(k) + \Delta w_{13}$$
$$w_{21}(k+1) = w_{21}(k) + \Delta w_{21}$$
$$w_{22}(k+1) = w_{22}(k) + \Delta w_{22}$$
$$w_{23}(k+1) = w_{23}(k) + \Delta w_{23}$$

Also bias should be updated for all iterations:

$$\Delta b = -\eta \frac{\partial J(k)}{\partial e(k)} * \frac{\partial v(k)}{\partial w_{33}(k)} = -\eta * e * x$$

$$b(k+1) = b(k) + \Delta b$$

So that the cost funciton is minimized.

b) Design (Write matlab m-file code) an adaptive NN controller using gradient descent in order to minimize the objective function in (3) for the given reference signal in fig 2. The initial values of weights in NN controller are initialized as in Table 1. You can download reference signal in dost system and upload using given m-file.
Plot
i) The alternation of the controller parameters versus time
ii) Reference-system signal versus time.
iii) Control Signal versus time.

Results are obtained with 100 iterations

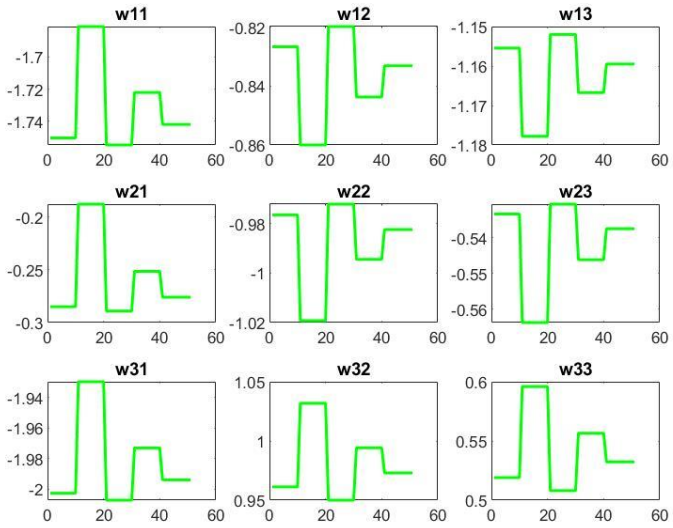


Figure 1: controller parameters versus time

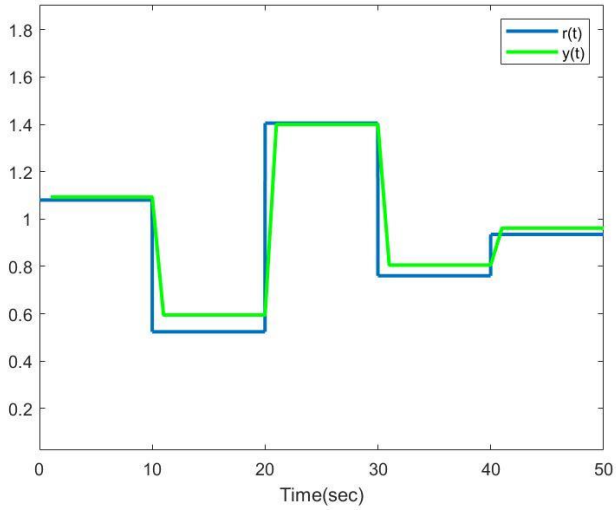


Figure 2: reference signal versus time

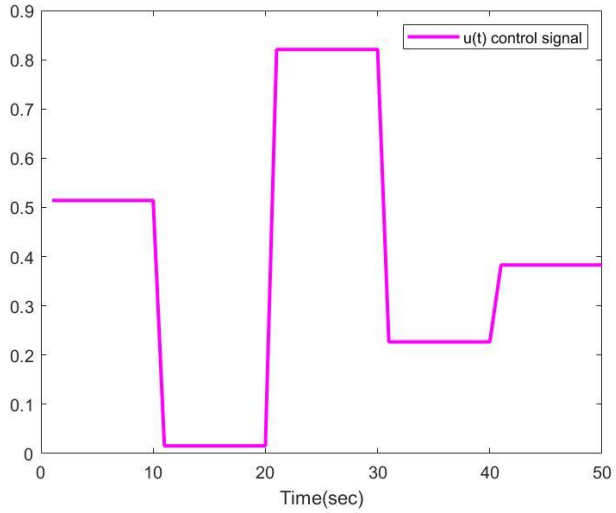


Figure 3: control signal versus time

Question 2: The PID type fuzzy controller is illustrated, where e_{tr_n} is the tracking error and $u_{FPID_{n+1}}$ is the control signal produced by fuzzy pid controller at time index n. In this assignment, triangular membership functions with {-1,-0.4,0,0.4,1} are chosen for each linguistic value of the e and $\dot{e}$ as shown in fig. 5.

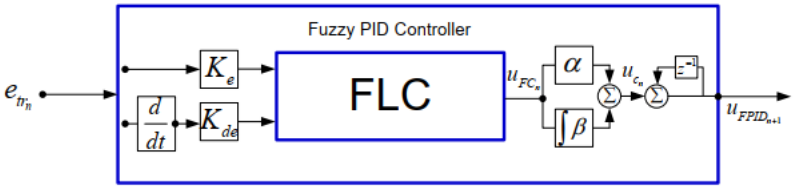


Fig.3 PID Type Fuzzy Controller(F PID C)

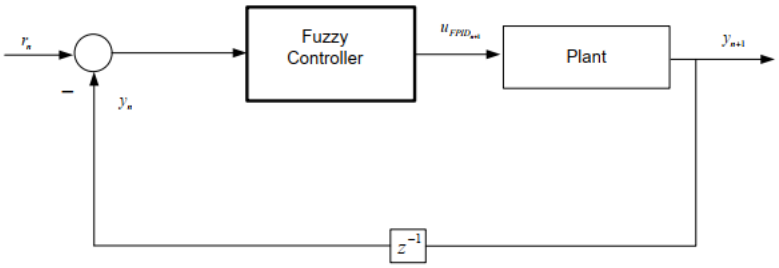


Figure 4: fuzzy control

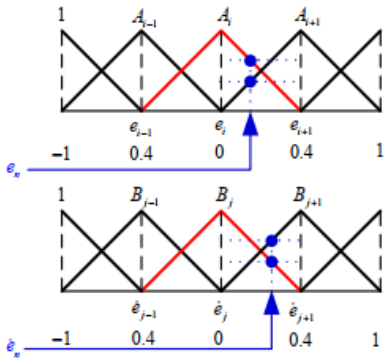


Figure 5: The Membership Functions of Ai and Bj

a) Obtain the control procedure for the given problem step by step.

The control procedure of CSTR using a fuzzy controller can be put in the following order:

1. Tracking error calculation using the current error and its derivative

2. Obtaining the inputs of the controller using Ke, Kde and tracking error

3. Determining the output of the fuzzy controller with fuzzified error and derivative of error values and correct the control rule

4. Current control signal calculation with the controller outside of the fuzzy controller

5. Applying the control signal to the system and recalculating its new states

b) Design(Write matlab m-file code) a PID type fuzzy controller in order to minimize the tracking error for the given reference signal.

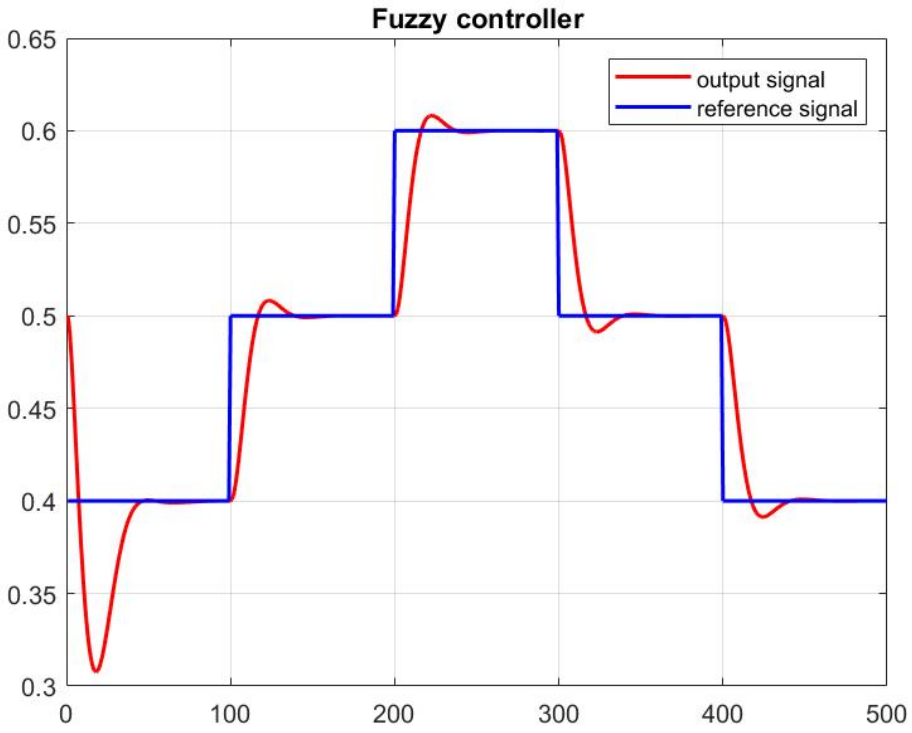


Figure 6: Fuzzy controller

d) Propose any adaptation mechanism for this controller based on neural network or fuzzy logic or any other intelligent method. Ilustrate with figures and explain the working principle of the proposed mechanism

An adaptive neuro-fuzzy inference system or adaptive network-based fuzzy inference system (ANFIS) is a kind of artificial neural network that is based on Takagi–Sugeno fuzzy inference system.

Since it integrates both neural networks and fuzzy logic principles, it has potential to capture the benefits of both in a single framework. Its inference system corresponds to a set of fuzzy IF–THEN rules that have learning capability to approximate nonlinear functions.

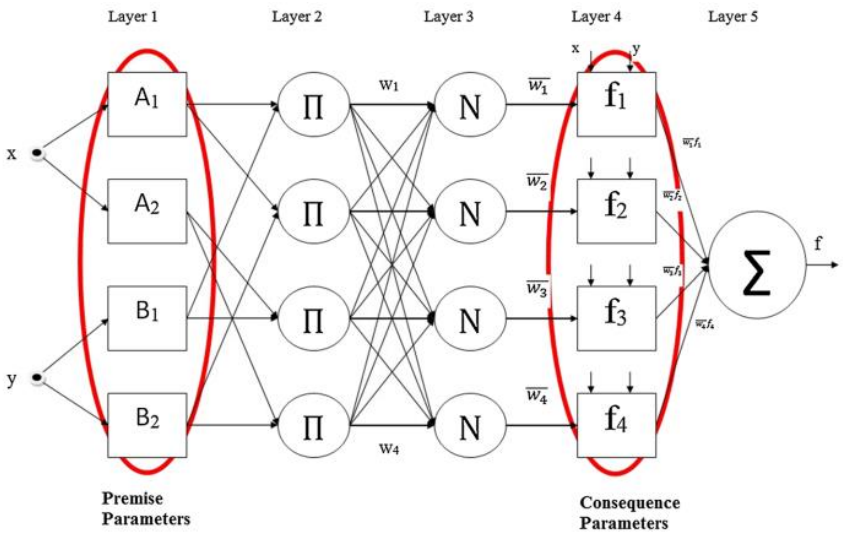


Figure 7: ANFIS structure

It is possible to identify two parts in the network structure, namely premise and consequence parts. In more details, the architecture is composed by five layers. The first layer takes the input values and determines the membership functions belonging to them. It is commonly called ‘fuzzification’ layer. The membership degrees of each function are computed by using the premise parameter set, namely {a,b,c}. The second layer is responsible of generating the firing strengths for the rules. Due to its task, the second layer is denoted as "rule layer". The role of the third layer is to normalize the computed firing strengths, by dividing each value for the total firing strength. The fourth layer takes as input the normalized values and the consequence parameter set {p,q,r}. The values returned by this layer are the defuzzificated ones and those values are passed to the last layer to return the final output.

References:

- https://en.wikipedia.org/wiki/Adaptive_neuro_fuzzy_inference_system
- Karaboga, Dervis; Kaya, Ebubekir (2018). "Adaptive network based fuzzy inference system (ANFIS) training approaches: a comprehensive survey". Artificial Intelligence Review. 52 (4): 2263–2293. doi:10.1007/s10462-017-9610-2. ISSN 0269-2821. S2CID 40548050.