

Military Technical College  
Kobry El-Kobba  
Cairo, Egypt



12-th International Conference  
on  
Aerospace Sciences &  
Aviation Technology

## TIME DELAY COMPENSATION USING ADAPTIVE LINEAR NEURAL NETWORKS FOR NETWORKED CONTROL SYSTEMS

AHMED\* E. ABDALLA, MOHAMMED\*\* H. ASSAL, and HYDER\*\*\* O. ELBASHEIR

### ABSTRACT

In real time systems, particularly control systems, delays or dropped packets may cause performance degradation and system destabilization. In order to consider the uncertainty of communication delays and packet losses, intelligent computational approaches such as fuzzy logic, neural networks, and genetic algorithm can be used. In this paper, The effect of time delay is compensated via building undelayed plant model based on delayed model data using the Adaptive Linear Neuron networks (ADALINE). In ADALINE the linear networks are adjusted at each time step based on new input and target vectors which can find weights and biases that minimize the network's sum-squared error for recent input and target vectors. The proposed works are applied on distributed control of a DC servo system. The network is built using the true time MATLAB toolbox. Several simulation examples are applied using CAN network to clarify the efficiency of the proposed methods.

### KEYWORDS

Networked Control Systems, ADALINE, Time Delay, Neural Networks, Distributed Control, and CAN Network.

---

\* Egyptian Army

\*\* Modern Academy, Cairo, Egypt

\*\*\* Sudanese Army

## INTRODUCTION

Due to extensive functionality and low cost of microcontroller, many industrial control systems are changing into more intelligent digital systems. As a system becomes more intelligent and flexible, the system requires more sensors, actuators, and controllers, often referred to as field devices. As the number of devices in a system grows and the functions of a system need to be more intelligent, these devices need to exchange a large amount of data among them. Unlike regular control systems; in networked control systems the synchronization between different sensors, actuators and control units is not guaranteed. Furthermore, there is no guarantee for zero delay or even constant delay in sending information from sensors to the control units and control signals from the control units to the actuators. When there is overcrowding in the communication network, some packets are dropped to either reduce the queue size in the path or to inform the senders to reduce their transmission rates. In real time systems, particularly control systems, delays or dropped packets may cause performance degradation and system destabilization. In order to consider the uncertainty of communication delays and packet losses, intelligent computational approaches such as fuzzy logic, neural networks, and genetic algorithm can be used.

In [1], a modified Fuzzy PID Controller is introduced to implement real-time control adaptively to deal with random delays in Networked Control systems. By adjusting the control signal dynamically, the system performance is improved. An intelligent controller using fuzzy logic on top of a PI gain to adaptively compensate for the IP network-induced time delay in time-delay sensitive Networked Control System applications are proposed in [2-6]. Another approach of intelligent NCS is presented in [7], in which the constant time delay is estimated through the use of neural networks (NN), where the NNs are trained by the filtered noisy and delayed signal and the noise-free signal.

In this work, The effect of time delay is compensated via building undelayed plant model based on delayed model data using ADALINE [8]. The proposed works are applied on distributed control of a DC servo system. More details are given in the following sections.

## DELAYS IN NETWORK-BASED CONTROL SYSTEMS

The network-induced delay in NCSs occurs when sensors, actuators, and controllers exchange data across the network. This delay can degrade the performance of control systems designed without considering it and can even destabilize the system. Major types of these communication delays, along with a block diagram of a discrete-time model are depicted in Fig.1. and the timing diagram of delay generations with  $t < h$ ; where,  $h$  is the sampling time; is shown in Fig.2.

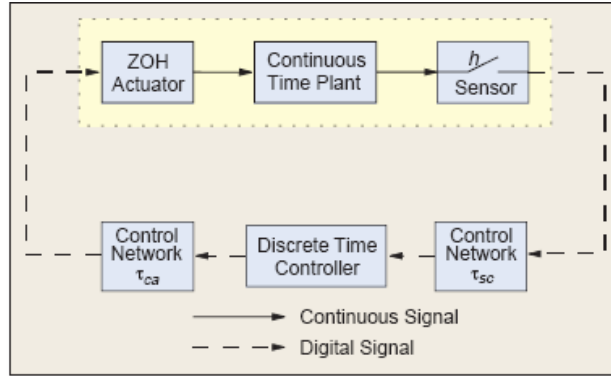


Fig.1. Block diagram of a discrete-time model of a NCS

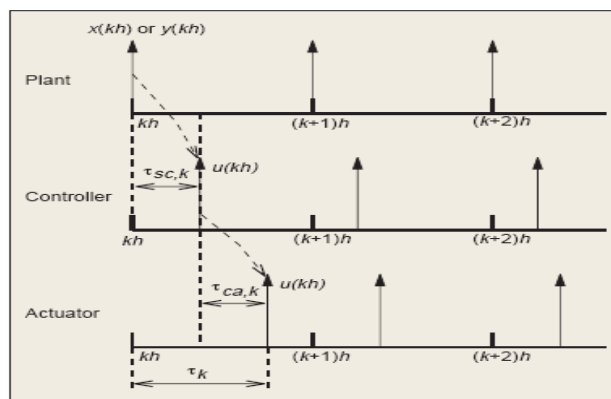


Fig.2. Timing diagram of delay generations.

### MODELING OF NCS CONSIDERING NETWORK-INDUCED DELAY

The NCS model considering network-induced delay is shown in Fig.3. The model consists of a continuous plant:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \tag{1}$$

and a discrete controller:

$$u(kh) = -Kx(kh) \quad k = 0,1,2,\dots \tag{2}$$

Here,  $x \in \mathbb{R}^n$ ,  $u \in \mathbb{R}^m$ , and  $y \in \mathbb{R}^p$  and  $A, B, C, K$  are of compatible dimensions. There are two sources of delays from the network: sensor-to-controller  $\tau_{sc}$  and controller-to-actuator  $\tau_{ca}$ . Any controller computational delay can be absorbed into either  $\tau_{sc}$  or  $\tau_{ca}$  without loss of generality [9]. For fixed control law (time-invariant controllers), the sensor-to-controller delay and controller-to-actuator delay can be

lumped together as  $\tau = \tau_{ca} + \tau_{sc}$  for analysis purposes. We consider the setup with: Clock-driven sensors that sample the plant outputs periodically at sampling instants. Event-driven controllers, which can be implemented by an external event, interrupt mechanism and which calculates the control signal as soon as the sensor data arrives, and Event-driven actuators, which mean the plant inputs, are changed as soon as the data become available.

Consider the case where the delay of each sample,  $x(kh)$  is less than one sampling period,  $h$ . This constraint means that, at most, two control samples  $u((k-1)h)$  and  $u(kh)$ , need be applied during the sampling period. The system equations can be written as

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \quad t \in [kh + \tau_k, (k+1)h + \tau_{k+1}),$$

$$u(t^+) = -Kx(t - t_k), \quad t \in \{kh + \tau_k, k = 0, 1, 2, \dots\} \quad (3)$$

where  $u(t^+)$  is piecewise continuous and only changes value at  $kh + \tau_k$ . Sampling the system with period  $h$ , we obtain the following discrete form [10].

$$\begin{aligned} x((k+1)h) &= \Phi x(kh) + \Gamma_0(\tau_k)u(kh) + \Gamma_1(\tau_k)u((k-1)h), \\ y(kh) &= Cx(kh) \end{aligned} \quad (4)$$

where

$$\begin{aligned} \Phi &= e^{Ah} \\ \Gamma_0(\tau_k) &= \int_0^{h-\tau_k} e^{As} B ds \\ \Gamma_1(\tau_k) &= \int_{h-\tau_k}^h e^{As} B ds \end{aligned}$$

Defining

$$z(kh) = [x^T(kh), u^T((k-1)h)]^T \quad (5)$$

as the augmented state vector, the augmented closed-loop system

$$z((k+1)h) = \tilde{\Phi}(k)z(kh), \quad (6)$$

where

$$\tilde{\Phi}(k) = \begin{bmatrix} \Phi - \Gamma_0(\tau_k)K & \Gamma_1(\tau_k) \\ -K & 0 \end{bmatrix}$$

If the delay is constant (i.e.,  $\tau_k = \tau$ , for  $k=0,1,2,\dots$ ) the system is still time invariant, which simplifies the system analysis. Thus we can envision static scheduling network protocols, such as token ring or token bus, which can provide constant delay. Even in this simplified setup, the next question is: "how much delay can the system tolerate?"

Another observation is that the sensor-controller delay can be compensated by an estimator if the messages sent out by sensors are time stamped [11]. Traditional one-step prediction estimation can compensate delays less than one sampling period, since the estimate of  $x(kh)$  only depends on the value of  $y((k-1)h)$ .

### ADAPTIVE NEURAL CONTROL OF TIME DELAY NETWORKED SYSTEMS

A linear neuron with R inputs is shown below [ 8,12 ] in Fig.3.

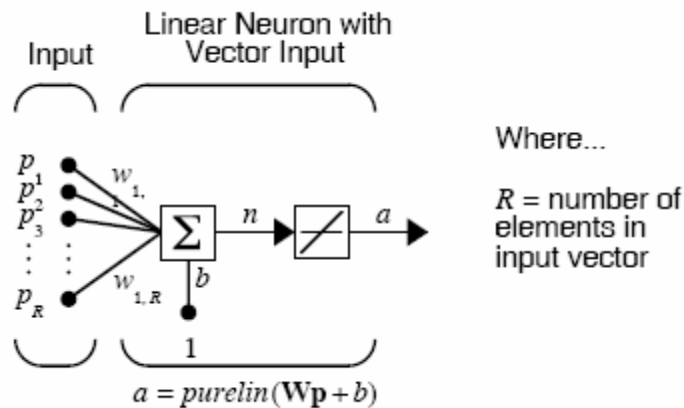


Fig.3. Linear neuron model

This network has the same basic structure as the perceptron. The only difference is that the linear neuron uses a linear transfer function, which we name "purelin", see Fig.4. and described by

$$a = \text{purelin}(n) = \text{purelin}(W_p + b) = W_p + b \tag{7}$$

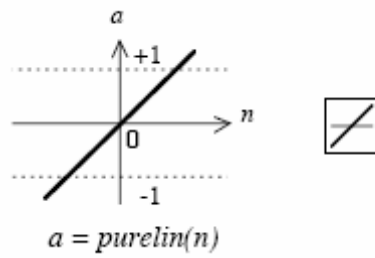


Fig.4. Linear transfer function

This neuron can be trained to learn an affine function of its inputs, or to find a linear approximation to a nonlinear function. A linear network cannot, of course, be made to perform a nonlinear computation.

The Adaptive Linear Network (ADALINE) architecture is shown in Fig.5. has one layer of  $S$  neurons connected to  $R$  inputs through a matrix of weights  $W$ .

This network is sometimes called a MADALINE for Many ADALINES. Note that the figure on the right defines an  $S$ -length output vector  $\mathbf{a}$ . The Widrow-Hoff rule can only train single-layer linear networks. This is not much of a disadvantage; however, as single-layer linear networks are just as capable as multilayer linear networks. For every multilayer linear network, there is an equivalent single-layer linear network.

Like the perceptron, the ADALINE has a decision boundary that is determined by the input vectors for which the net input  $n$  is zero, see Fig.6. For  $n=0$  the equation  $Wp + b = 0$  specifies such a decision boundary as shown below. Input vectors in the upper right gray area lead to an output greater than 0. Input vectors in the lower left white area lead to an output less than 0. Thus, ADALINE can be used to classify objects into two categories.

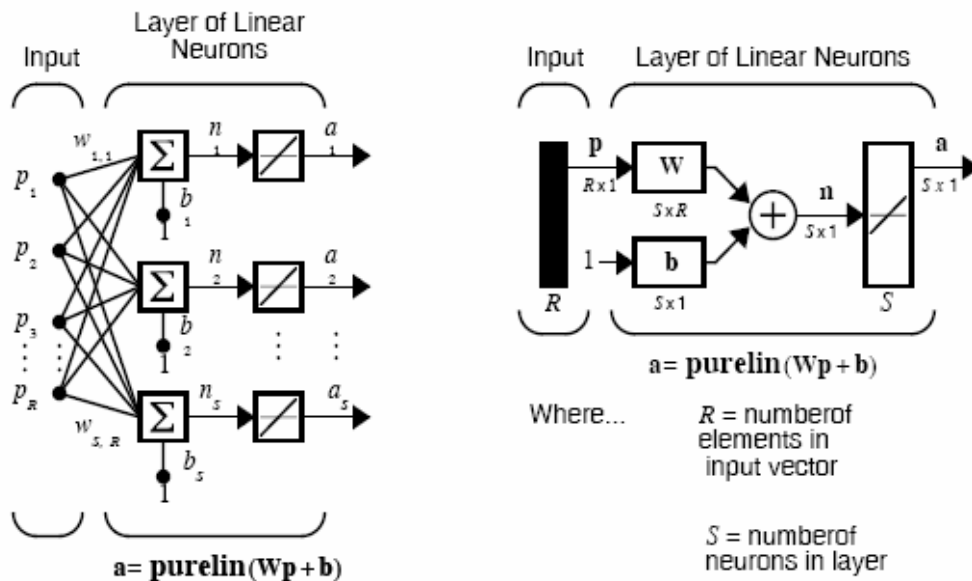


Fig.5. the ADALINE network

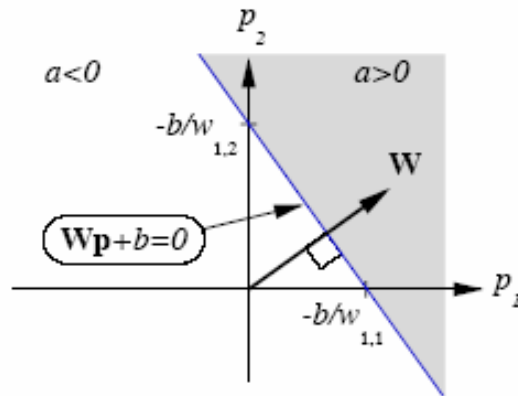


Fig.6. ADALINE decision boundary

The learning rule, is the least mean square error (LMS) algorithm for the desired network behavior which is described by  $(p_1, t_1), (p_2, t_2), \dots, (p_q, t_q)$  where  $p_q$  is an input to the network, and  $t_q$  is the corresponding target output. As each input is applied to the network, the network output is compared to the target. The error is calculated as the difference between the target output and the network output. We want to minimize the average of the sum of these errors.

$$mse = \frac{1}{Q} \sum_{k=1}^Q e(k)^2 = \frac{1}{Q} \sum_{k=1}^Q (r(k) - a(k))^2 \tag{8}$$

The LMS algorithm adjusts the weights and biases of the ADALINE so as to minimize this mean square error.

### NCS TIME DELAY COMPENSATION BASED ON ADALINE

We assume that the update time  $h$  is larger than the delay time  $\tau_k$ , and both are constants. At times  $k(h - \tau)$  the sensor transmits the state data to the controller/actuator. This data will arrive  $\tau_k$ . So, at times  $kh$  the controller/actuator receives the state vector value  $x(kh - k\tau_k)$ . The main idea is to use the plant model in the controller/actuator to calculate the present value of the output state. The approximate output state obtained can be used to update the controller's input. The ADALINE (Adaptive Linear Neuron networks) model uses the plant model and the past values of the control input  $u(t)$  to calculate an estimate of actual output state  $y(kh)$  from the received data  $y(kh - \tau_k)$ . This estimate is then used to update the model that, with the controller, will generate the control signal for the plant.

## SIMULATION RESULTS

Consider the SIMULINK model NCS of distributed DC servo system shown in Fig.7. The system contains four computer nodes connected by one network block. The time-driven sensor node contains a periodic task, which at each invocation samples the process and transmits the sample package to the controller node. The controller node contains an event-driven task that is triggered each time a sample arrives over the network from the sensor node. Upon receiving a sample, the controller computes a control signal, which is then sent to the event-driven actuator node, where it is actuated. The model also contains an interference node with a periodic task generating random interfering traffic over the network. We assume a CAN network where transmission of simultaneous messages is decided based on package priorities.

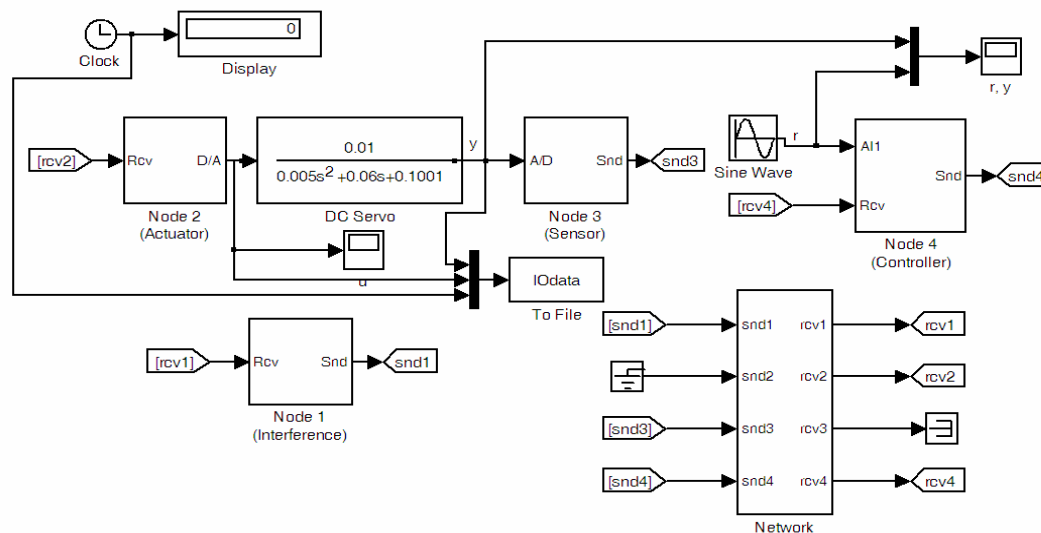


Fig.7. SIMULINK model NCS of distributed DC servo system

The step respons of the system is obtained using digital PID controller and neural network with CAN networks for different network-induced time delays (0.1,0.15, 0.2) are shown in Fig.8.

From simulation results, it was observed that the applied neural network (ADALINE) performed relatively better than the conventional digital PID controller; neural network (ADALINE) for NCS is a very appropriate choice due to its robustness in terms of system parameters. Especially, the measurement of network delay may become very difficult, if not impossible, because the delay varies with many factors including implementation method of the network communication process, communication traffic of the network, and the number of stations.



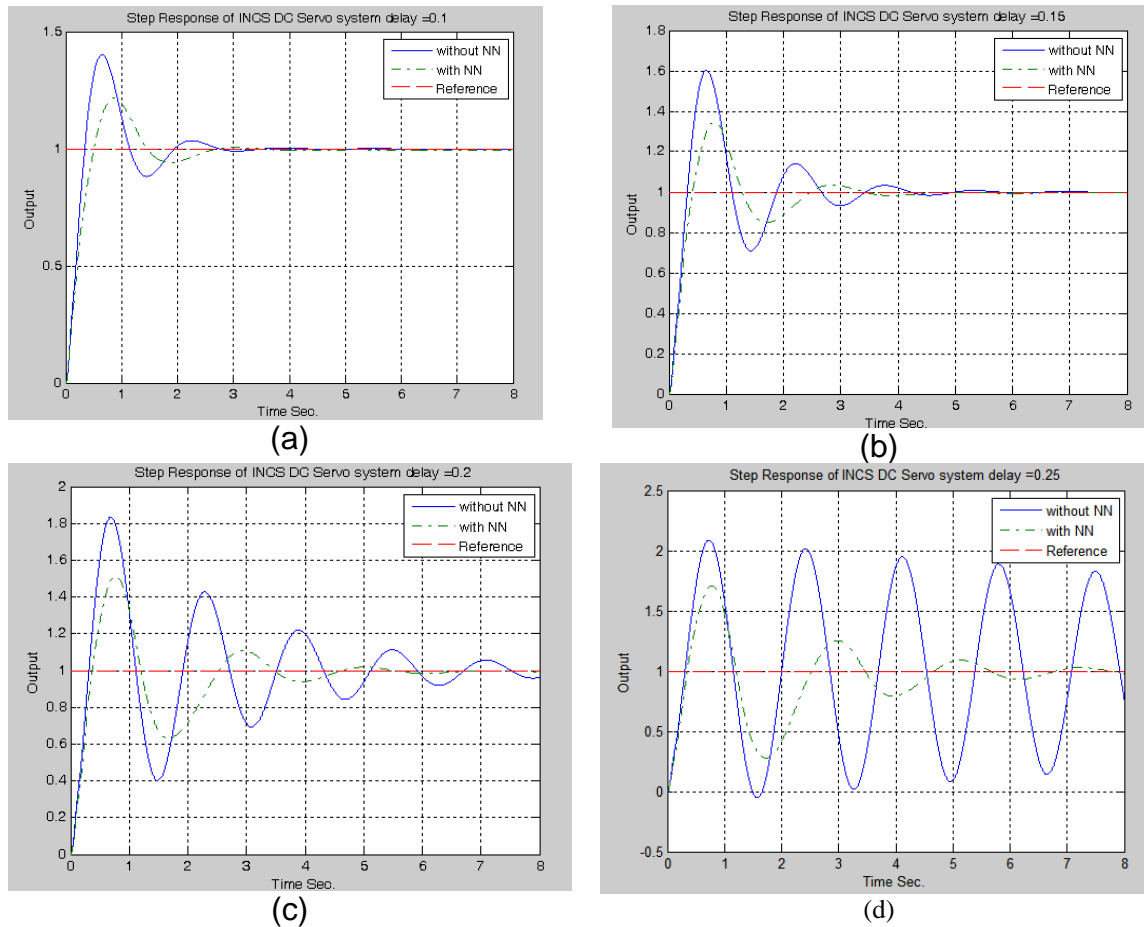


Fig.8. Step response of NCS DC servo system for different network-induced time delays (0.1, 0.15,0.2, 0.25) without and with applying NN.

## 9 CONCLUSION

The change of communication architecture from point-to-point connection to the common-bus approach, however, introduces different forms of time-delay uncertainty and in closed-loop system dynamics. These time delays come from time sharing of the communication medium as well as the computation time required for physical signal coding, communication processing. The time delays in a control application can degrade a system's performance and even cause system instability. In order to guarantee the stability and performance of an NCS, the time delays due to networked communications should be characterized, the impact of network-induced delays on control performance should be analyzed, and a methodology to compensate or minimize these effects should be investigated. In this work the effect of time delay is compensated via building undelayed plant model based on delayed model data using ADALINE. In ADALINE the Linear networks are adjusted at each time step based on new input and target vectors via which the weights and biases can be obtained that minimize the network's sum-squared error for recent input and target vectors. The proposed works are applied on distributed control of a DC servo system using CAN network. The network is built using the TrueTime MATLAB toolbox. The simulation result reflects the efficiency of the proposed method of network induced delays compensation.

## REFERENCES

- [1] Yong-can Cao and Wei-dong Zhang, "Modified Fuzzy PID Control for Networked Control Systems with Random Delays", Transactions On Engineering, Computing And Technology V12 MARCH 2006 ISSN 1305-5313.
- [2] N. B. Almutairi & M. -Y. Chow, "Adaptive fuzzy modulation (AFM) for IP networked PI control systems. Part I: Partial adaptation," for possible presentation at IECON 2002, Sevilla, Spain, 2002.
- [3] N. B. Almutairi & M. -Y. Chow, "Adaptive fuzzy modulation (AFM) for IP networked PI control systems. Part II: Full adaptation," for possible presentation at IECON 2002, Sevilla, Spain, 2002.
- [4] Yong-can Cao and Wei-dong Zhang, "Modified Fuzzy PID Control for Networked Control Systems with Random Delays", Transactions On Engineering, Computing And Technology V12 MARCH 2006 ISSN 1305-5313
- [5] K.J. Astrom and B. Wittenmark, Computer-Controlled Systems: Theory and Design, Prentice Hall, Upper Saddle River, NJ, 1997. Third Edition.
- [6] F.-L. Lian, J. R. Moyne, and D. M. Tilbury, "Performance evaluation of control networks: Ethernet, ControlNet and DeviceNet," IEEE Contr. Syst. Mag., vol. 21, no. 1, pp. 66–83, Feb. 2001.
- [7] Samir Shaltaf, Neural-Network-Based Time-Delay Estimation EURASIP Journal on Applied Signal Processing 2004:3, 378–385.
- [8] MathWorks (2001): Simulink: A Program for Simulating Dynamic Systems—User's Guide. The MathWorks Inc.
- [9] B. Wittenmark, J. Nilsson, and M. Torngren. Timing problems in real-time control systems. In Proceedings of American Control Conference, pages 2000-2004, 1995.
- [10] Dawson, J. G. and Gao, Z. (1994). Fuzzy logic control of linear systems with variable time delay, LQ IEEE.
- [11] Richard P. Lipmann . An Introduction to Computing with Neural Nets, IEEE ASSP Magazine, April 1987.
- [12] K.M. Lee, D.H. Kwak, H. Leekwang, Tuning of fuzzy models by fuzzy neural networks, Fuzzy Sets and Systems 76 (1995) 47–61.