Adaptive neuro-fuzzy controller trained by genetic-particle swarm for active queue management in internet congestion

Mohammed I. Berbek, Ahmed A. Oglah
Control and Systems Engineering Departement, University of Technology-Iraq, Baghdad, Iraq

Article Info

Article history:

Received Sep 4, 2021
Revised Jan 14, 2022
Accepted Feb 4, 2022

Keywords:

Active queue management
ANFIS like-PID
Fuzzy like-PID
Genetic algorithms
Particle swarm optimization

ABSTRACT

Routers are vital during network congestion. All routers have input and output packet buffers. Various congestion control strategies have been suggested. Some controller-based proportional-integral derivative (PIDs) have recently been offered as active queue management (AQM) solutions to alleviate the deterioration of transmission control protocol (TCP) congestion management system performance. However, the time delay is large, the data retention decreases, and oscillation occurs, suggesting that the present PID-controller is unable to fulfill quality of service (QoS) criteria. Some research is developed on new control technologies such as neural networks and fuzzy logic. This paper proposes the adaptive neuro-fuzzy inference system (ANFIS) like PID controller for AQM. This model employs genetic algorithms (GAs) and particle swarm optimization (PSO) to learn and optimize all variables for ANFIS like PID controller. Simulations were used to investigate the effects of using fuzzy like PID based on single sign-on (SSO), and (ANFIS like PI, ANFIS like PID with GA-PSO) controllers on the length of the queue for an AQM router, respectively. Then we compared the findings to see which approach should be utilized to manage the queue length for AQM routers. In simulations, ANFIS like PID has superior stability, convergence, resilience, loss ratio, goodput, lowest rising time, overshoot, and settling time.

Corresponding Author:

Mohammed I. Berbek
Control and Systems Engineering Departement, University of Technology-Iraq
Baghdad, Iraq
Email: cse.19.08@grad.uotechnology.edu.iq

1. INTRODUCTION

Because a growing number of people are using the Internet, it has captured the public's interest during the previous decade. As the number of Internet users grows, so does the Internet's capacity to meet the demand. As a result, Congestion occurs in the network when aggregate demand exceeds available capacity. This may result in lower link utilization, longer round-trip times, and possibly network inaccessibility. Buffer management for Internet routers is important for congestion control. Droptail is a conventional buffer management strategy that works with transport layer protocols such the transmission control protocol (TCP). The round trip time (RTT) of each packet increases as a result of droptail's passive behavior, and more crucially, As a result of the loss of correlation between packet drops, the well-known "TCP synchronization" issue occurs [1]. As a result, a variety of congestion control techniques have been developed. Active queue management (AQM) has been proposed as a mechanism to provide senders with congestion notifications in advance of the queue overflowing and packet loss occurring. AQM has been the subject of numerous studies [2]-[7].

The primary purpose of AQM is to maintain queue lengths at a certain target, allowing for a good balance of high throughput and low end-to-end delay; and to ensure high link usage. The first AQM technique was random early detection (RED), which used the average queue length to calculate the risk of discarding packets [2]. AQM has been developed in recent years as an effective strategy for mitigating "TCP synchronization" and avoiding network congestion. FAPIDNN is proposed as a novel algorithm for AQM in [3]. The neural network's PID controller calculates the probability of discarding a packet depending on the fuzzy controller's rate of learning. The FAPIDNN technique is the most accurate in terms of the PID controller's queue, convergence speed, and time delay, according to simulation findings. A linear quadratic optimum controller was constructed using linear control theory. The design specifications are determined by the weighting matrices Q and R used. where the genetic algorithm (GA) was used to find the appropriate weighting matrices, As an efficient technique that acts on intermediate nodes to facilitate end-to-end congestion control [4].

Feng *et al.* [5] introduced a nonlinear RED-based, three-section random early detection (TRED) is a system that divides the packet dropping probability function into three parts to differentiate between light, moderate, and high loads. Xu *et al.* [6] A robust PID non-fragile control technique is devised for the TCP/AQM network to be controlled. To adjust network queue congestion, a PID controller is proposed. A PD-type feedback controller for efficiently regulating the queue length and enhancing the system's stability was presented. Additionally, it provides guidance for selecting control gain parameters [7].

Fezazi *et al.* [8] proposed a strategy for internet congestion control based on a static feedback law. To achieve stability over a vast variety of permissible initial conditions, a congestion controller for network TCP/IP Routers is developed. By measuring the average influence of an accepted/dropped packet on the aggregated packet arriving rate, this research provides an innovative information compression model simply from the standpoint of routers. The proposed technique enhances performance in terms of providing a good tradeoff between reducing latency and maximizing goodput, according to simulation results [9]. For queue length stabilization with a short delay and a quick settling period, the LQ-servo controller is given. The recommended controller parameters are adjusted using the particle swarm optimization (PSO) method. The PSO algorithm was used to find the appropriate controller parameters, Q and R, in order to obtain a decent output response.When compared to the PI controller [10].

Liu *et al.* [11] The authors suggested the finite-time performance function (FTPF), which introduces a novel finite-time control design approach and solves the finite-time control problem for a class of transmission control protocol/active queue management (TCP/AQM) networks. By using machine learning approaches to exploit the explicit congestion notification (ECN) information to improve AQM algorithms, to predict congestion, the intelligent technique employs an artificial neural system (ANN) and a rein-forcement learning-based AQM parameter tuner. The results of the study suggest that by utilizing current TCP congestion control mechanisms, this strategy can improve the performance of deployed AQM [12]. A new AQM algorithm was established based on model predictive control (MPC) theory commonly used in non-linear and time-delay systems [13]. A revised model for TCP/AQM networks with multiple bottleneck routers was presented, based on the previous network model with one bottleneck router, and the usual control approach of backstepping was applied to a TCP network with several bottleneck routers. backstepping congestion control for multi-router (BCCM) is a proposed network congestion solution based on integrated backstepping [14]. In the paper, Wang *et al.* [15] Investigations were done on TCP congestion management issues in a finite time $H\infty$ frame. For the finite-time convergence of the tracking error, a novel $H\infty$ finite time controller design technique is described using the backstepping methodology, finite-time control method, and $H\infty$ control theory.

All the above researches are based on the network congestion problem is solved by many AQM algorithms to detect and control congestion. This paper proposes AQM algorithm based on adaptive neuro fuzzy inference system (ANFIS) like-PID Controller that can be trained by GA-PSO technique, composed of the conventional PID control, The ANFIS is an ANN based on the Takagi-Sugeno fuzzy inference system (FIS). ANFIS is a system that combines neural networks and fuzzy logic principles, allowing it permission to use both in a single framework. ANFIS can be utilized to automatically control and optimize the system. Therefore, it was used in this research to Control the queue length to guarantee to keep the router queue length as close to the goal queue length and to achieve a stable performance when the number of TCP connections changes.

The following is how the paper is organized: The AQM linear model explains in section 2. Section 3 provides describes of the ANFIS like-PID controller (ANFIS-PID) structure. An overview of the PSO and GA technique is explained in section 4. Section 5 contains the simulation results that were used to verify the proposed controller using MATLAB *(R2020a)* software. Section 6 concludes with a conclusion.

2. THE TCP/AQM MATHMATICAL MODEL

Models for TCP have been represented based on the theories of renewal, fluid models, fixed point, processor sharing, and control theorie [16]. Hollot and his colleagues' [17] model is used. This model is a simplified version of the nonlinear dynamic model proposed by Misra and colleagues [18], which is based primarily on fluid flow and stochastic differential equations. The TCP timeout mechanism is ignored in the simplified model.

The following are the model's nonlinear differential (1):

$$\dot{W}(t)=\frac{1}{R(t)}-\frac{W(t)W(t-R(t))}{2R(t-R(t))}\,\mathrm{P}\,(t-R(t)) \tag{1}$$

$$\dot{q}(t)=\frac{W(t)}{R(t)}\,N(t)-C \tag{2}$$

$\dot{W}(t)$ Signifies W(t)'s time derivative, while $\dot{q}(t)$ denotes q(t)'s time derivative, q: Expected queue length (packets), W: is the TCP window size (packets), N: Load factor (number of TCP sessions); R_0: Round-trip time (seconds), C: Link capacity (packets/second), p: Packet mark/drop probability, t: Time. In (1), TCP window control dynamic is described, a multiplicative decline of the TCP formula is described when the sender is in congested avoidance, each RTT's congestion window grows by one data segment size, and when congestion is detected, the window shrinks to half its size. In (2), it depicts variations in the router's buffer space the number of packets queued in the router changes as a result of packets sent to the router is deducted from the packet that was sent out. The likelihood of a packet being marked for dropping is known as the probability of marking. This value falls between (0, 1). Figure 1 is a representation of an (1). This Figure focuses on TCP window control and queue dynamics.

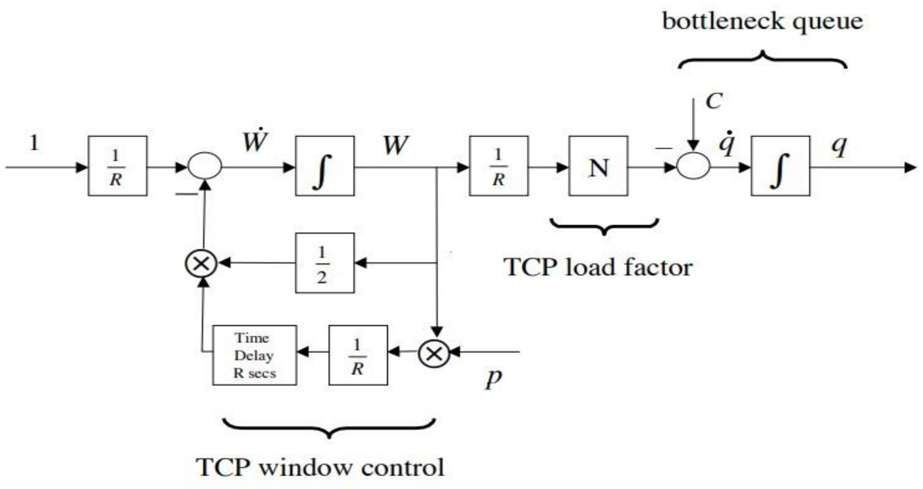


Figure 1. TCP's congestion avoidance flow-control mode block diagram [17]

The number of TCP sessions and round-trip times are assumed to be constant N(t) ≡ N and R(t) ≡ R_0 in order to linearize the model, Then, for feedback control (AQM), an approximated linearized model can be built using small-signal linearization around an operating point (W_o, p_o, q_o) [17]. The right sides of the differential in (1) and (2) are defined as follows:

$$f(W,W_R,q,p)=\frac{1}{R}-\frac{W(t)W_R}{2R}P\,(t-R) \tag{3}$$

$$g(W,q)=\frac{N(t)}{R}W(t)-C \tag{4}$$

where, $W_R(t)\cong W(t-R)$. Recall the operating point relationships:

$$\dot{W}=0 \quad \Longrightarrow W_0^2p_0=2 \tag{5}$$

$$\dot{q} = 0 \quad \Rightarrow W_0 = R_0 C/N$$

where:
$R_0 = \frac{q_0}{C} + T_p$
T_p: Propagation delay (seconds).
Evaluating partials at these operating points (W_o, p_o, q_o). In (5) gives the dynamics of linearized TCP/AQM router is:

$$\delta \dot{W}(t) = -\frac{2N}{R_0^2 C}\left(\delta W(t) + \delta W(t - R_0)\right) - \frac{R_0 C^2}{2N^2}\delta P(t - R_0) \tag{6}$$

$$\delta \dot{q}(t) = \frac{N}{R_0}\delta W(t) - \frac{1}{R_0}\delta q(t) \tag{7}$$

where:

$$\delta W(t) \cong W - W_0, \delta q(t) \cong q - q_0, \delta p(t) \cong p - p_0$$

If ($W_0 >> 1$), the delay term in the TCP window-control dynamic $\delta W(t - R_0)$ in (4) can be omitted. For typical network conditions, ($W_0 >> 1$) is a plausible assumption, thus this delay will be ignored in the rest of the equations, and the simplified dynamics in the following equations can be considered:

$$\delta \dot{W}(t) = -\frac{2N}{R_0^2 C}\delta W(t) - \frac{R_0 C^2}{2N^2}\delta P(t - R_0) \tag{8}$$

$$\delta \dot{q}(t) = \frac{N}{R_0}\delta W(t) - \frac{1}{R_0}\delta q(t) \tag{9}$$

The following transfer functions can be constructed from the Laplace transform of (8), (9):

$$P_{tcp(S)} = \frac{W(S)}{P(S)} = \frac{\frac{R_0 C^2}{2N^2}}{S + \frac{2N}{R_0^2 C}} \tag{10}$$

$$P_{queue(S)} = \frac{q(S)}{w(S)} = \frac{\frac{N}{R_0}}{S + \frac{1}{R_0}} \tag{11}$$

As a result, the plant transfer function is written as (12).

$$P(S) = P_{tcp(S)} P_{queue(S)} e^{-sR_0} \tag{12}$$

And it can be written as (13).

$$P(s) = \frac{W(s)}{P(s)} = \frac{\frac{C^2}{2N} e^{-sR0}}{\left(S + \frac{2N}{R_0^2 C}\right)\left(S + \frac{1}{R_0}\right)} \tag{13}$$

Figure 2 a schematic of a linear AQM control block. In this diagram $P_{tcp(S)}$ illustrates the loss probability transfer function $\delta p(t)$ to the size of window $\delta W(t)$, $P_{queue(S)}$ illustrates the function of transmission from $\delta W(t)$ to queue length $\delta q(t)$, C(s) system controller, and delay term e^{-sR_0}.

In (13) is a general model, and for the sake of analyzing the controllers' performance and simulation work, a special model for the TCP/AQM is determined, the case study of the network in this work, N = 60, R0 = 0.2530 seconds and C = 3750 packet/second, where the variables are used in an (13), that got transfer function in (14). The network topology is displayed in Figure 3.

$$P(s) = \frac{117187 \cdot 5\, e^{-0.253S}}{S\,2 + 4 \cdot 4524S + 1 \cdot 9759} \tag{14}$$

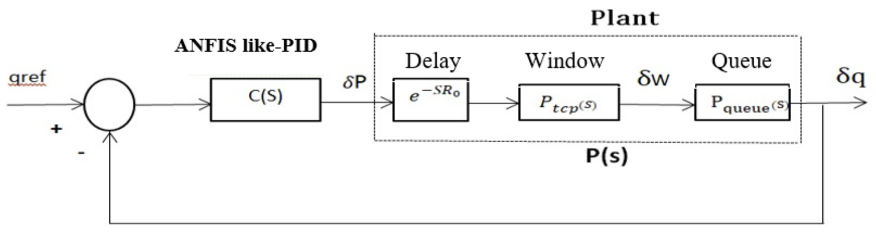


Figure 2. A linearized AQM block is utilized as a feedback signal [17]

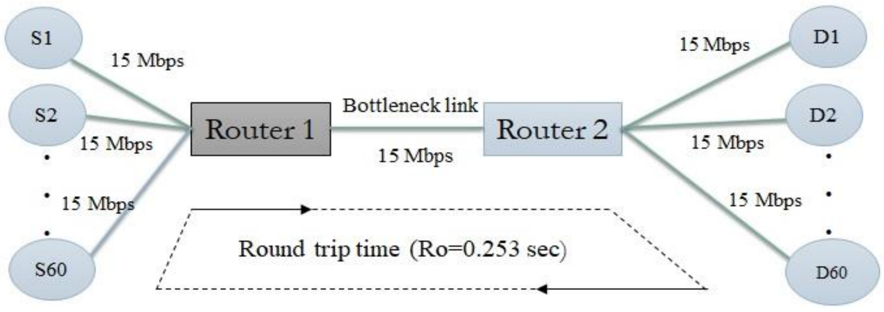


Figure 3. The case study network's topology

3. DESIGN OF A CONTROLLER

Strong control system synthesis' main goal is to find a control rule that keeps the authorized signal, system output, and error signals constant. This Section introduces the design of a fuzzy like-PID, and ANFIS like-PID controller for congestion control.

3.1. Fuzzy like-PID controller

Because of its low complexity, ease of implementation, robustness, and the limited number of parameters to adjust, the PID controller is frequently employed. Below is a description of the PID controller [19].

$$u(t)= K_P e(t) + K_D \bar{e}(t) + K_I \int e(t)dt \tag{15}$$

Where:
K_P = Proportional gain, K_I = Integral gain, and K_D = Differential gain, e(t) = The difference between the output and the inputs.

For classical control theory, An example of a standardized control structure is a PID controller. However, due to nonlinearity in the process plant, the performance is substantially affected and the efficiency is lowered. Fuzzy PID Controller is a natural extension from conventional PID controllers, which maintain the linear structure of the PID controller. Fuzzy logic and is a type of Artificial Intelligence. Its goal is to implant human intelligence into the system, allowing it to think intelligently such a human being [20]. In AQM routers, intelligent congestion avoidance control is provided by a fuzzy PID controller that is able to handle a wider range of operational situations than traditional controllers. Figure 4 depicts the suggested controller, which is made up of a fuzzy like-PID controller.

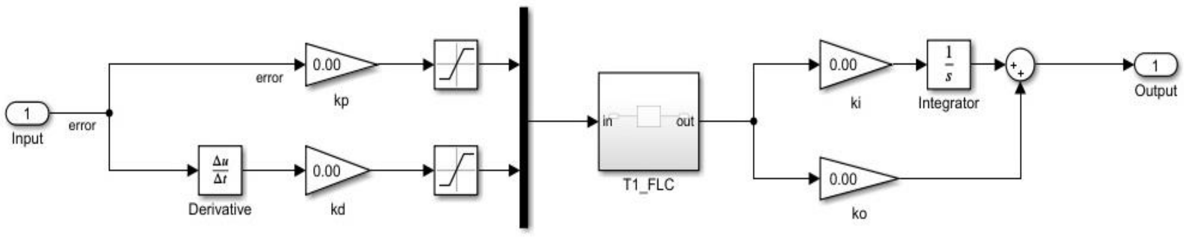
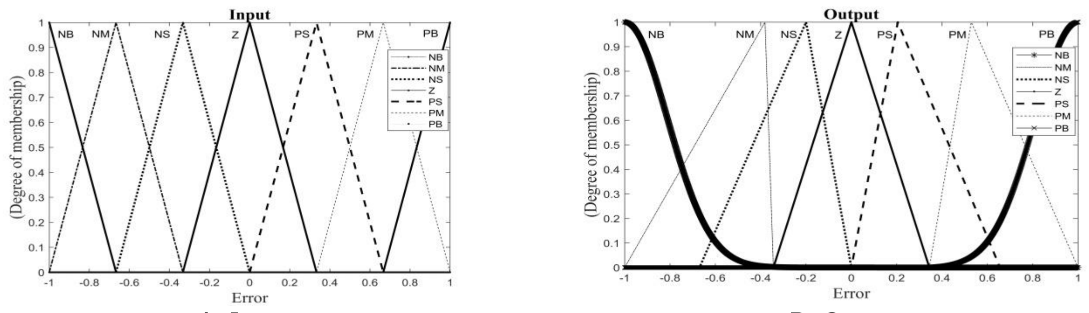


Figure 4. A fuzzy's structure like PID logic controller

The fuzzy PID controller has two inputs. The inputs are the classical error (e) and the error rate (ė) [21]. An Fuzzy logic controller consists of four major components: Fuzzifier, Rules, Inference engine, and De-fuzzifier. seven triangular and Gaussian membership functions are used for input variables, and for the output variable u, seven triangular and Gaussian membership functions are defined as in Figure 5. Inputs and output membership functions which are spread over their universes of discourse (-1, 1) range that is modified by ko is the output gain that is utilized to adjust the membership range of the outputs into the probability factor range's [Zero(0) One(1)] zone. The center of gravity is chosen as a defuzzification method. Table 1 shows the fuzzy-PID controller rule base, composed of a total of 49 (7×7) rules.



A -Input *B -Output*

Figure 5. The fuzzy logic control (FLC'S) membership function

Table 1. Fuzzy logic control rule's base

e \ ė	NB	NM	NS	Z	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	Z
NM	NB	NB	NB	NM	NS	Z	PS
NS	NB	NB	NM	NS	Z	PS	PM
Z	NB	NM	NS	Z	PS	PM	PB
PS	NM	NS	Z	PS	PM	PB	PB
PM	NS	Z	PS	PM	PB	PB	PB
PB	Z	PS	PM	PB	PB	PB	PB

Where positive big (PB); positive medium (PM); positive-small (PS); zero (Z); stands for negative-small (NS), for negative-medium (NM) and for negative-big (NB) are the table's abbreviations. Additionally, the number of MFs we should use to create a precise and Fuzzy system of superior quality, resulting in a precise response. Matlab was used to create this (mamdani fuzzy inference systems). The AND operation is often used to include the rules.

3.2. ANFIS-structure

Jang [22] first introduced in 1993 the ANFIS method. It is a technique that combines the advantages of fuzzy logic and neural network systems. To tune the parameters of a fuzzy inference system (FIS), an ANFIS uses neural network learning methods. As illustrated in Figure 6, The ANFIS architecture is made up of five layers.

Figure 6 depicts the usual architecture of an ANFIS controller; here all the ANFIS inputs use 7 membership functions for two input signals error (e) and rate of error (ė). ANFIS trains these memberships' features in order to improve network congestion control and provide optimal regulated signal characteristics, to tune the gains of the controller. It is made up of five layers in total. The inputs are represented by the first layer, the fuzzification by the second layer, the rules by the third layer, the defuzzification by the fourth layer, and the summation by the fifth layer [23].

Layer 1 the crisp inputs are given fuzzy values in this layer, which is a simple Fuzzification layer.

Layer 2 (Fuzzification): The nodes are adaptable with a function in this layer. The outputs are the inputs' Fuzzy membership grades, which are determined by:

$$O_i^1 = \mu_{Ai}(x), \qquad i=1, 2 \tag{16}$$

$$O_j^1 = \mu_{Bj}(y), \qquad j=1, 2 \tag{17}$$

Membership functions, such as triangular, trapezoidal, Gaussian, and Singleton, are used to fuzzifying the inputs. Among the MFs listed above, the gaussian function was chosen for this study due to its clear and concise notation, as shown in (18).

$$\mu_{Ai}(x), \mu_{Bj}(y) = exp\ [\frac{-(x-c_i)^2}{2\sigma_i^2}] \qquad i, j= 1,2 \tag{18}$$

Where the antecedent (premise) parameters are c_i and σ_i. Where $\{c_i , \sigma_i\}$ are the MFS characteristics that influence the form of MFs. The membership function's width is σ_i and c the center of the function is c_i.

Layer 3 (fuzzy rule weighting): Every fixed node in this layer is represented by the symbol Π. The firing strength W_k is calculated using the membership values computed in the fuzzification layer. The following is how the outputs are calculated.

$$O_k^3 = W_k = \mu_{Ai}(x) * \mu_{Bj}(y) \qquad i, j= 1,2 \tag{19}$$

Layer 4 (de-fuzzification): In each node of this layer, weighted consequent values of rules are calculated as shown in (20).

$$O_k^4 = W_k f_k = W_k(p_k x + q_k y + r_k) \qquad k=1, 2 \tag{20}$$

Where:
W_k Represents the third layer's output, and$(p_k; q_k ; r_k)$ are consequent parameters.
Layer 5 (summation): De-fuzzification layer's outputs are summed together to produce ANFIS's total output.

$$O^5 = \sum_{k=1}^{n} W_k f_k = \frac{\sum_1^n W_k * f_k}{W1+W2} \tag{21}$$

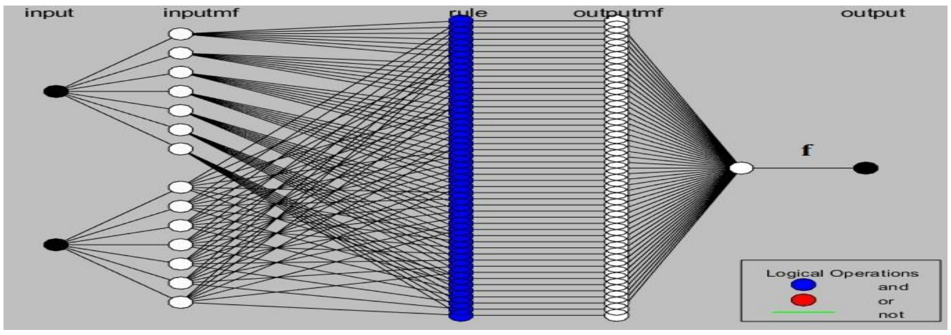


Figure 6. ANIFS architecture of two inputs Sugeno-fuzzy model with 49 rules

3.3. ANFIS like-PID controller

To provide a robust controller in AQM routers, an ANFIS like-PID controller is constructed as an intelligent congestion avoidance controller. Figure 7 illustrates the structure of an ANFIS like-PID controller [24], [25].

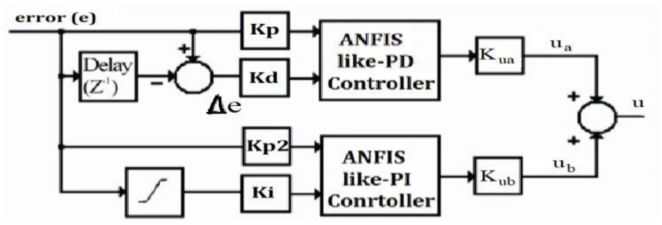


Figure 7. General block diagram of ANFIS like PID controller [25]

The PID equation is described in (15).

$$u(t)= K_P e(t) + K_D \bar{e}(t) + K_I \int e(t)dt$$

The ANFIS like-PID controller can be built as a parallel structure of an ANFIS like-PD controller and an ANFIS like-PI controller, as shown in Figure 7, with the output approximated as (23).

$$u(t)= u_a + u_b = (K_{P1}e(t) + K_D \bar{e}(t)) + (K_{P2}e(t) + K_I \int e(t).dt)$$

$$u_a(t) = K_{P1}e(t) + K_D\bar{e}(t) \tag{23}$$

However, the ANFIS PD controller is represented by the first portion of (22). The control signal (u_a) is calculated by the PD controller for any pair of e and Δe values. The ANFIS controller should do the same thing for any pair of errors (e) and change of error (Δe), it should work out the control signal through Neuro fuzzy inference system. To choose the best values for the input (premise parameters) and output (consequent parameters) which will be modified in the training phase. Various forms of gaussian-shaped membership functions can be obtained when the values of these parameters change. In (22)'s second portion depicts a PI controller, which represents ANFIS like-PI controller. Where both the (error and sum of error) signal are entered into the ANFIS to get output control signal (u_b).

$$u_b(t) = K_{P2}e(t) + K_I \int (t).dt \tag{24}$$

The parameters of Ku_a and Ku_b are the output scaling factors of ANFIS PD and ANFIS PI controllers respectively, the proposed ANFIS like-PID will consist of:

Layer 1: This layer is a basic Fuzzification layer that assigns relative fuzzy values to crisp inputs.
Layer 2: 7 gaussian MF for $e(t)$, 7 gaussian MF for $\bar{e}(t)$, and 7 gaussian MF for $\int e(t).dt$.
Layer 3: 49 Rules as mentioned in Table 1.
Layer 4: first-order Sugeno: $O_i^4 = W_i^3(p_i \times e(t) + q_i \times \bar{e}(t) + r_i)$ or $O_i^4 = W_i^3(p_i \times e(t) + q_i \times \int e(t).dt + r_i)$.
Layer5: $f = \text{sum}(O_i^4) = u_a$ or u_b

Figure 8 depicts the overall schematic diagram of the ANFIS like-PID controller system.

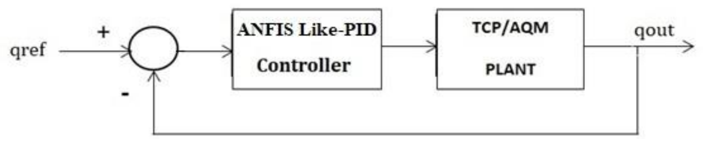


Figure 8. ANFIS like-PID control system, general block diagram

4. AN OVERVIEW OF THE GENETIC ALGORITHM AND PARTICLE SWARM OPTIMIZATION

Solving difficult optimization issues has remained a popular topic of study, posing several obstacles to researchers. The majority of the time, there are several local solutions to complex real-world optimization problems. Researchers were inspired by natural events in solving challenges of optimization. Mimicking the behavior or natural occurrences of natural systems has resulted in numerous methods of optimization, such as genetic algorithm (GA), and PSO. These algorithms were created by adapting naturally occurring processes. They are known by various names, such as evolutionary algorithms and metaheuristic techniques. Metaheuristic techniques typically incorporate problem-specific heuristic algorithms in a more generic framework. As a result, metaheuristics are defined as tactics used to arrive at an optimal or near-optimal solution.

4.1. Genetic algorithm

Genetic algorithms are computational algorithms based on natural selection and natural genetics. The principle of natural selection and evolution was embraced by genetic algorithms. The basic goal of the genetic algorithm is to mimic the natural selection process, which is based on the principle of survival of the fittest. Each answer is represented as a chromosome in the genetic algorithm. The fitness of these chromosomes is assessed and ranked. The genetic algorithm's flow chart is presented in Figure 9 [26].

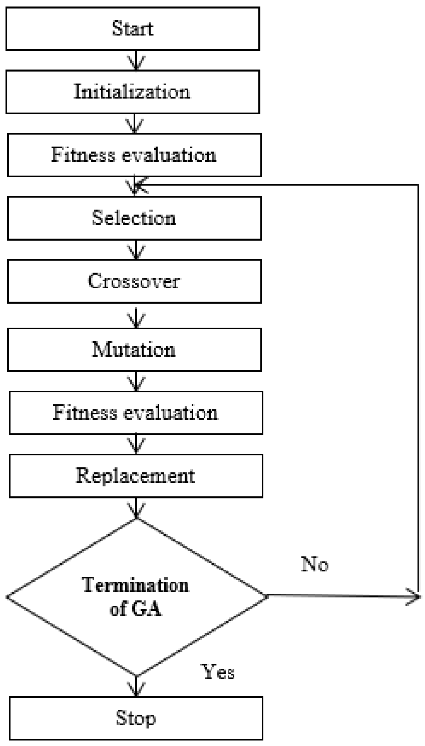


Figure 9. Flowchart for genetic algorithm

These randomly populated first chromosomes are known as parent chromosomes, while subsequent generations of chromosomes are known as child or offspring chromosomes. The goal of genetic algorithms is to incorporate better parents in the reproduction process so that better children can be produced. The stronger chromosomes survive to the next round of natural selection, whereas the weaker chromosomes are eliminated. These chromosomes represent the solution in any optimization issue, and the fittest chromosome is expressed as the best outcome for the optimization problem at hand. To assist with this, natural selection process, genetic algorithms use three distinct procedures: selection, crossover, and mutation operators. The selection operator is employed in the first stage of selection to choose the better chromosomes. These chromosomes are ranked according to the fitness function that corresponds to the scenario of optimization.

After that, the individual solutions are rated according to their fitness value. Individual chromosomes are sorted based on the type of optimization problem at hand, which is to maximize (or) reduce a certain goal. The chromosomes with the highest fitness value are ranked first if the goal is to maximize. If the goal is to reduce the size of a function, the chromosomes with the lowest fitness value are prioritized, and the highest-ranking chromosomes frequently migrate to the next level and participate in the Generations process. The top-ranked chromosomes frequently move to the next stage of the process via a mechanism known as "Elite Count." Typically, the Elite Count is fixed at 10% of the population size. The elite count is set at two in the case of a population of twenty. Many ways for selecting chromosomes are available in the literature; these chromosomes are selected using two methods: roulette selection and tournament-based selection. A crossover operator is used to generate new populations once the parent chromosomes have been selected.

To make new chromosomes (solutions), the crossover operator normally combines two different parents. In the literature, there are various types of crossover operators. The crossover probability is a measure of a person's proclivity to spawn stronger persons. Because only the most powerful individuals are allowed to engage in population creation, this may have an impact on population diversity and, as a result, solution diversity. To avoid becoming trapped inside a local minima (or) maxima, the mutation operator is employed to infuse diversity into the solution. Individual chromosomes are mutated by the mutation operator to make new children. Individual chromosomes are altered to provide an unique mutation factor, which adds diversity to the solution while also assuring excellent convergence. There are many different types of mutation operators available, and which one to use depends on the type and necessity of the optimization problem.

4.2. Particle swarm optimization

Kennedy and Eberhart were the first to present the PSO approach in 1995 [27]. This method explores the space of possible solutions to an optimization problem by utilizing a cooperative swarm of particles to search the space of possible solutions. Each particle that is a candidate solution is either randomly or heuristically initialized before being allowed to 'fly' in the parameter space with a velocity that is dynamically changed based on its own and its companions' flight experiences [28]. The following formulas can be used to compute each particle's adjusted velocity and position.

$$vij(t+1) = wv_{ij}(t) + c_1r_1\left[pbest_{ij}\,(t) - x_{ij}(t)\right] + c_2r_2\left[gbest_{ij}\,(t) - x\_ij\,(t)\right] \quad (25)$$

$$x_{ij}(t+1) = x_{ij}(t) + vij(t+1) \quad (26)$$

i= 1, 2,...,
j= 1, 2,…,

Where (i) denotes the number of particles in a group, (j) denotes the number of members in a particle, (t) is the number of iterations, The velocity of particle (i) at iteration t is $\mathrm{v_{ij}(t)}$, w is the weight inertia factor, (c1, c2) the constants of acceleration,($\mathrm{r_1}$, $\mathrm{r_2}$) Random numbers range from (0 to 1), The current position of particle (i) at t iteration is $\mathrm{x_{ij}(t)}$, $\mathrm{pbest_{ij}(t)}$ is the pbest of particle (i), and finally, $\mathrm{gbest_{ij}(t)}$ is the gbest of the group. In (25); is made up of three terms: $\mathrm{wv_{ij}(t)}$ a displacement physical component, where (w) is a variable parameter that can be used to alter the particle's displacement in the following iteration, $\mathrm{c_1r_1}\left[\mathrm{pbest_{ij}}(t) - x_{ij}(t)\right]$ a cognitive displacement component and $\mathrm{c_2r_2}\left[\mathrm{gbest_{ij}}(t) - x_{ij}(t)\right]$ a social component of displacement.

5. SIMULATION RESULTS AND DISCUSSION

5.1. Linearized TCP/AQM model simulation results

The parameters of the network topology simulation results given in Figures 3 are the follows [29]: The bottle-neck connection capacity of 15 Mega bit per second between routers 1 and 2 is rectangular in shape, with the delay time of propagation 0.2 second, 500 bytes as a packet size and N = 60, and the ideal I/P queue size is rectangular in shape, as indicated in (27).

$$q_{ref} = [\;3\;,\;2\;,\;4\;,\;2\;]*100 \quad (27)$$
$$Time = [\;(0) < t < (50)\;,\;(50) < t < (100)\;,\;(100) < t < (150)\;,\;(150) < t < (200)\;]$$

The suggested controllers are evaluated in this section using MATLAB *(R2020a)*. The simulation is shown in Figure 10 without a controller. Because to the heavy traffic, that surpasses the Max. Size buffer, the device enters a long-term oscillation, and the system without the controller is unable to track queue length to the appropriate level through queue length. To minimize continuous oscillations and enhance tracking efficiency, the system using an FLC like-PID is first simulated using an optimization approach (SSO). Second, the simulation is done for the system with (ANFIS like-PI, ANFIS like-PID) Controller that can be trained by GA-PSO algorithm technique to control AQM systems. Figure 11 demonstrates the results of using the social optimization method (SSO) settings (FLC like -PID).

As an appropriate tuning optimization technique, the SSO (PID like-FLC) The SSO parameter was 50 spiders, 0.65 lower female 0.9% higher female % 50 iterations, and the integral-time absolute-error, was used as the cost function (ITAE).

$$\text{ITAE} = \int t\,|(e)|\;dt \quad (28)$$

Where: $e = qref - qout$

Figure 12 shows the simulation of the ANFIS like-PI Controller that can be trained by GA-PSO Algorithm technique to control AQM systems using (m.file code). It has been suggested to test the effectiveness of the GA-PSO technique for optimizing both the antecedent and the consequent parameters of the ANFIS acting like PID controller. The GA-PSO parameters are set to the following values: PSO parameters are set: Population size (Swarm size) 100, the inertia weight factor (w = 0.73), acceleration constant (c1=1.5) and (c2=1.5). Genetic algorithm parameters have been set: Population size (100), Crossover= (0.8), Mutation = (0.3) and Mutation Rate=0.03. The maximum number of iterations is 100.

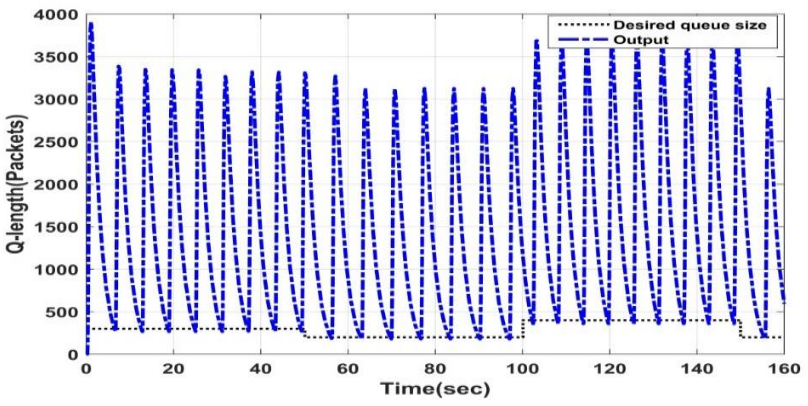


Figure 10. Time response of the system in the absence of a controller

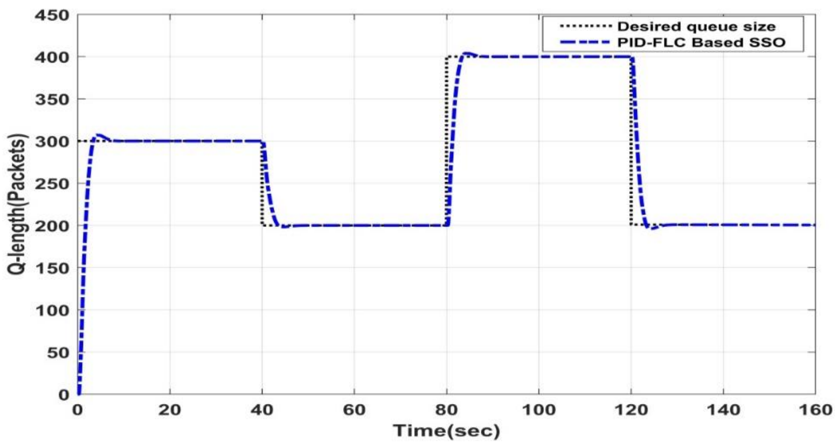


Figure 11. The system responds with an FLC like- PID based on SSO

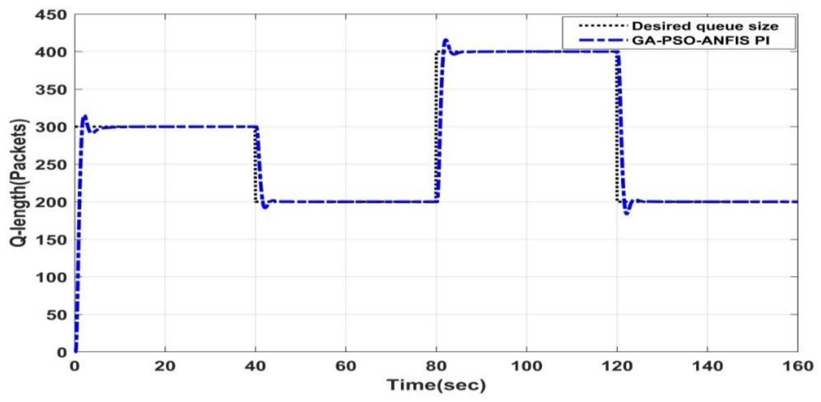


Figure 12. Controller response for GA-PSO-ANFIS like PI controller

Figure 13 Displays the system response of using ANFIS like PID with trained by GA-PSO algorithm. The comparison among the three controller's results (FLC like-PID Based SSO, GA-PSO ANFIS like-PI, GA-PSO ANFIS like-PID) is in Figure 14. It is showing clearly GA-PSO ANFIS like-PID is the most efficient in terms of Rise time, Overshot, and settling time.

Table 2 compares the steady state properties of all controllers using optimization techniques. The comparison of PID gains parameters for all algorithms is shown in Table 3. Table 2 indicates that GA- the PSO ANFIS PID controller outperforms the others controllers in terms of rise time, overshoot, and settling time.

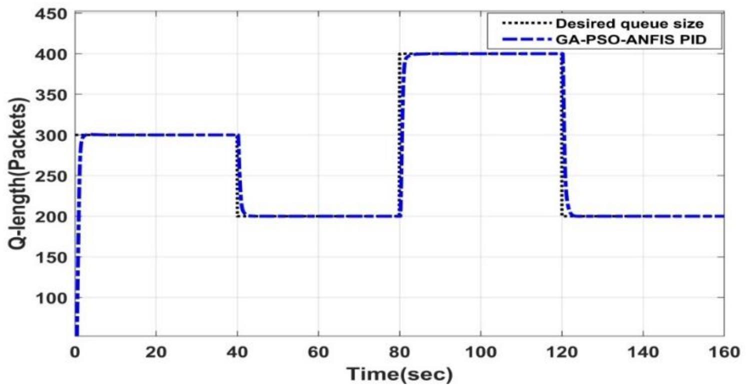


Figure 13. Controller response for GA-PSO-ANFIS like PID controller

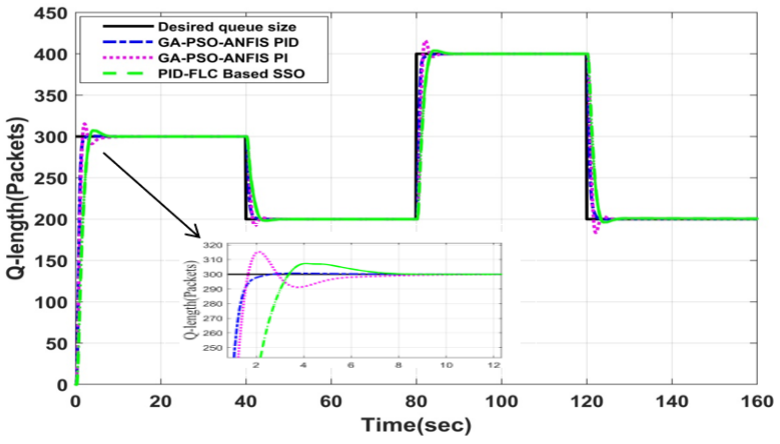


Figure 14. A comparison of the results by the three controllers for AQM

Table 2. Comparison of steady-state responses

Controllers Type	Rise-time [sec.]	Settling-time [sec].	Max.Over. %
GA-PSO-ANFIS PID	0.9	2.6	0.2
GA-PSO-ANFIS PI	1	5.3	4.866
SSO-FLC	2	7.3	1.666

Table 3. For all controllers, a comparison of PID gain settings is made

Controller	Kp1	Kd	Kp2	Ki	Kua	Kub
GA-PSO-ANFIS PID	0.116	0.0054	0.0130	0.0064	0.0087	0.0067
GA-PSO-ANFIS PI	-	-	0.0198	0.0054	-	0.0103
FLC based on SSO	0.004118	0.002814	-	0.008345	0.007636	

This part demonstrates that without the controller, the model is unable to monitor the target queue length; however, when the controllers are applied to the model, the time response is high levels of acceptability and extremely efficient, and the model can keep track of the desired queue length. This suggests that the controllers' efficiency and capacity in maintaining system stability and tracking the AQM necessary value are both high. As a result, the rate of dropped packets can be reduced, and overall network performance can be improved. When comparing the results with the search [30]. A PID fuzzy-neural controller (FNC) was designed as an active queue management (AQM) in internet routers to improve the performance of a fuzzy proportional integral (FPI) controller used for congestion avoidance in computer networks. Particle swarm optimization (PSO) is utilized as an optimization approach for tuning the PID parameters to acquire the best controller parameters. Table 4 compares the results of this study with those of research [30], in terms of rising time, overshoot, and settling time. GA-PSO ANFIS PID Controller output response has given rise time and settling time of 0.9 sec, 2.6 sec respectively which is 5.6 sec and 4.4 sec respectively less than that of FNC-PID-PSO controller. However, it gives the percentage overshoot of 0.2%.

Table 4. TCP/AQM system response performance compared by the number of controllers

Controllers Type	Rise-time [sec].	Settling-time [sec].	Max.Over %
GA-PSO-ANFIS PID	0.9	2.6	0.2
GA-PSO-ANFIS PI	1	5.3	4.866
SSO-FLC	2	7.3	1.666
FNPI-PSO [30]	5.7	13	5
FNC-PID [30]	3.8	15	29
FNC-PID-PSO [30]	6.5	7	-

6. CONCLUSION

The following conclusions are reached based on the design and simulation results: the designed SSO-FLC, GA-PSO-ANFIS PI, and GA-PSO-ANFIS PID controllers can handle congestion problems with strong tracking performance about the desired queue size, high link utilization, and speedier system response, where the simulation showed that the GA-PSO-ANFIS PID controller has better performance in terms of Rise time, settling time and overshoot than the rest of the controllers. The initial step in our research work was to examine SSO-FLC, GA-PSO-ANFIS PI, and GA-PSO-ANFIS PID controller independently, and then evaluated all of the results to present a comparison of which technique should be employed in AQM scheme control. A system should always be built to have a lower percentage overshoot, as well as fewer rising and settling times. The comparative investigation in this study shows that GA-PSO-ANFIS PID controller is a lot better than SSO-FLC and GA-PSO-ANFIS PI controller as it gives a percentage overshoot of 0.2% than that of SSO-FLC and GA-PSO-ANFIS PI controller which is 1.666%, 4.866% with 87.9% and 95.8% percentage of enhancement in overshoot respectively. Percentage overshoot describes what happens when a signal exceeds its steady-state value. The designed GA-PSO-ANFIS PID controller can decrease the rise time and settling time, with (10%, 55%) percentage of improvement in Rise time and decreasing its settling time by (50.9%, 64.3%) percentage for the GA-PSO-ANFIS PI and SSO_FLC controller respectively, and reduce the system's rise and settling times to make it more responsive and stable. This means the designed GA-PSO-ANFIS PID controller It has a better response than the rest of the controllers.

REFERENCES

[1] B. Braden *et al.*, "Recommendations on queue management and congestion avoidance in the internet," *RFC,* vol. 2309, pp. 1–17, 1998.

[2] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking,* vol. 1, no. 4, pp. 397–413, 1993, doi: 10.1109/90.251892.

[3] Y. Qiao and Q. Lei, "A new active queue management algorithm based on self-adaptive fuzzy neural-network PID controller," *2011 International Conference on Internet Technology and Applications, iTAP 2011 - Proceedings,* pp. 0–3, 2011, doi: 10.1109/ITAP.2011.6006116.

[4] M. Z. Al-Faiz and S. S. Sabry, "Optimal linear quadratic controller based on genetic algorithm for TCP/AQM router," *2012 International Conference on Future Communication Networks, ICFCN 2012,* 2012, pp. 78–83, doi: 10.1109/ICFCN.2012.6206877.

[5] C. Feng, L. Huang, C. Xu, and Y. Chang, "Analysis based on nonlinear RED," pp. 1–8, 2014.

[6] W. Xu, Z. Zhu, and X. Gao, "A robust PID non-fragile control theoretic analysis to TCP/AQM network," *Proceedings of the 2015 27th Chinese Control and Decision Conference, CCDC 2015*, 2015, pp. 2188–2192, doi: 10.1109/CCDC.2015.7162284.

[7] S. K. Bisoy and P. K. Pattnaik, "Design of feedback controller for TCP/AQM networks," *Engineering Science and Technology, an International Journal,* vol. 20, no. 1, pp. 116–132, 2017, doi: 10.1016/j.jestch.2016.10.002.

[8] N. E. Fezazi, F. E. Haoussi, E. H. Tissir, and T. Alvarez, "Design of robust H∞ controllers for congestion control in data networks," *Journal of the Franklin Institute*, vol. 354, no. 17, pp. 7828–7845, 2017, doi: 10.1016/j.jfranklin.2017.09.026.

[9] Z. Liu, J. Sun, S. Hu, and X. Hu, "An adaptive AQM algorithm based on a novel information compression model," *IEEE Access,* vol. 6, no. 1, pp. 31180–31190, 2018, doi: 10.1109/ACCESS.2018.2844407.

[10] S. S. Sabry and T. M. Nayl, "Particle swarm optimization based lq-servo controller for congestion avoidance," *Iraqi Journal of Computer, Communication, Control and System Engineering*, pp. 63–70, 2019, doi: 10.33103/uot.ijccce.19.1.8.

[11] Y. Liu, Y. Jing, and X. Chen, "Adaptive neural practically finite-time congestion control for TCP/AQM network," *Neurocomputing,* vol. 351, pp. 26–32, 2019, doi: 10.1016/j.neucom.2019.03.022.

[12] C. Gomez, X. Wang, and A. Shami, "Intelligent active queue management using explicit congestion notification," *CoRR,* vol. abs/1909.0, 2019, doi: 10.20944/preprints201909.0077.v1.

[13] Q. Xu, G. Ma, K. Ding, and B. Xu, "An adaptive active queue management based on model predictive control," *IEEE Access*, vol. 8, pp. 174489–174494, 2020, doi: 10.1109/ACCESS.2020.3025377.

[14] L. Ma, X. Liu, H. Wang, and X. Deng, "Congestion tracking control for multi-router TCP/AQM network based on integral backstepping," *Computer Networks*, vol. 175, no. February, 2020, doi: 10.1016/j.comnet.2020.107278.

[15] K. Wang, X. Liu, and Y. Jing, "Robust finite-time H∞ congestion control for a class of AQM network systems," *Neural Computing and Applications*, vol. 33, no. 8, pp. 3105–3112, 2021, doi: 10.1007/s00521-020-05168-z.

[16] J. Olsen, *Stochastic modeling and simulation of the TCP protocol orgen Ols ´*. 2003.

[17] Y. Zhang and M. Ahmed, "A control theoretic analysis of XCP," *Proceedings - IEEE INFOCOM*, no. 2, 2006, doi: 10.1109/INFOCOM.2006.348.

[18] V. Misra, W.-B. Gong, and D. Towsley, "Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED," *Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, 2000, pp. 151–160, doi: 10.1145/347059.347421.

[19] M. J. Er and Y. L. Sun, "Hybrid fuzzy proportional-integral plus conventional derivative control of linear and nonlinear systems," *IEEE Transactions on Industrial Electronics*, vol. 48, no. 6, pp. 1109–1117, 2001, doi: 10.1109/41.969389.

[20] M. J. Mohamed, "Design a Fuzzy PID Controller for Trajectory Tracking of Mobile Robot," *Engineering and Technology Journal*, vol. 36, no. 1, 2018, doi: 10.30684/etj.36.1a.15.

[21] A. A. O. and M. M. Msallam, "Real-time implementation of fuzzy logic controller based on chicken swarm optimization for the ball and plate system," *International Review of Applied Sciences and Engineering*, 2021, doi: 10.1556/1848.2021.00360.

[22] J. S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993, doi: 10.1109/21.256541.

[23] N. Farhan, A. Humod, and F. Hasan, "Field oriented control of AFPMSM for electrical vehicle using adaptive neuro-fuzzy inference system (ANFIS)," *Engineering and Technology Journal*, vol. 39, no. 10, pp. 1571–1582, 2021, doi: 10.30684/etj.v39i10.1969.

[24] A. J. Humaidi, A. A. Oglah, S. J. Abbas, and I. K. Ibraheem, "Optimal augmented linear and nonlinear PD control design for parallel robot based on PSO tuner," *International Review on Modelling and Simulations*, vol. 12, no. 5, pp. 281–291, 2019, doi: 10.15866/iremos.v12i5.16298.

[25] M. Arao, "Fuzzy controllers," *Journal of Agricultural Machinery Society*, vol. 51, no. 3, pp. 119–122, 1989, doi: 10.11357/jsam1937.51.3_119.

[26] E. Fjær *et al.*, "Preface to the 1992 edition," *Developments in Petroleum Science*, vol. 53, 2008, doi: 10.1016/S0376-7361(07)53015-3.

[27] R. Eberhart and J. Kennedy, "New optimizer using particle swarm theory," *Proceedings of the International Symposium on Micro Machine and Human Science*, pp. 39–43, 1995, doi: 10.1109/mhs.1995.494215.

[28] H. Y. Meng, X. H. Zhang, and S. Y. Liu, "A co-evolutionary particle swarm optimization-based method for multiobjective optimization," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3809 LNAI, no. 5, pp. 349–359, 2005, doi: 10.1007/11589990_37.

[29] H. M. Kadhim and A. A. Oglah, "Interval type-2 and type-1 fuzzy logic controllers for congestion avoidance in internet routers," *IOP Conference Series: Materials Science and Engineering*, vol. 881, no. 1, 2020, doi: 10.1088/1757-899X/881/1/012135.

[30] M. Z. Al-Faiz and S. A. Sadeq, "Particle swarm optimization based fuzzy-neural like PID controller for TCP/AQM router," *Intelligent Control and Automation*, vol. 03, no. 01, pp. 71–77, 2012, doi: 10.4236/ica.2012.31009.

BIOGRAPHIES OF AUTHORS

Mohammed I. Berbek received a B.Sc. degree in Control and System Engineering from the University of Technology, Baghdad, IRAQ in 2003 and 2007 respectively. Currently, he is working as a M.Sc. Student in University of Technology/control and system Engineering Department. He can be contacted at email: cse.19.08@grad.uotechnology.edu.iq.

Ahmed A. Oglah received his B.Sc. and M.Sc. degrees in Control engineering from Al-Rasheed College of Engineering and Science, University of Technology, Iraq, in 1998 and 2004, respectively. He received his Ph.D. degree from the University of Basrah in 2015 with a specialization in control and computers. He is presently the assistant professor of the computer engineering branch at the Control and Systems Engineering Department at the University of Technology. His field of interests includes intelligent control systems, nonlinear control systems, robotics, computer networks, image processing, digital systems design, real-time systems and optimization techniques. He can be contacted by email: Ahmed.A.Oglah@uotechnology.edu.iq.