Controller Design for Congestion Avoidance Based on Several Optimization Techniques

Ghufran abdulqader alhaddad [1], Anass yousif abass [2], Nora Ahmed Mohammed [3]

[1,3] College of Engineering, Al-Qadisiyah University, Iraq

[2] College of Medicine, Al-Qadisiyah University, Iraq

Emails: Ghufran@qu.edu.iq; Anass.alkhalidi@qu.edu.iq ; Nora.mohammed@qu.edu.iq

Abstract

The world has become more like a small community thanks to the internet, which connects millions of people, businesses, and pieces of technology for a variety of uses. Because of the significant influence these networks have on our lives, maintaining their efficiency is important, which necessitates addressing issues like congestion. In this study, PI-controller gains are adjusted using a variety of optimization strategies to regulate the nonlinear TCP/AQM model. This controller commits controlled pressured signaling characteristics and modifies computer network congestion. First manual tune PI-Controller are used; then several optimization techniques were used to tune PI-controller gains (Particle Swarm Optimization (PSO), Ant-Colony Optimization (ACO) and Simulated Annealing algorithm (SA)) and then Linear Quadratic Regulator theory are used. To test the reliability and effectiveness of each of the suggested controllers, several tests utilizing varied network parameter values, different queue sizes, and extra disturbances were conducted. MATLAB was used for all experiments., the results show the superiority of the LQR controller over PI controller with both manual and optimal tuning techniques.

Keyword: AQM; network congestion control; LQR; PSO; ACO; SA; PI controller.

1. Introduction

Due to the rapid development of communication networks and the internet, it is now necessary to transmit a huge volume of created data in a reliable and efficient manner in order to prevent network congestion. In order to strengthen and improve the quality of service, it is crucial to prevent network congestion. One of the most widely used transport protocols, transmission control protocol (TCP), offers an end-to-end congestion control method [1]. When the packets are transmitted to the receiver in this protocol, the average congestion window size rises. In contrast, when a data transfer fails, the average congestion window size shrinks. This method, meanwhile, suffers from poor network resource use and unreliable stream synchronization [2]. In order to prevent these limitations at gateways, Active Queue Management (AQM) has been suggested to improve performance. Numerous AQM variations have been proposed in recent years to ease congestion in TCP networks. Prior to the queue buffer overflowing, random early detection (RED), which randomly drops the packet with a specified probability, was proposed in [1]. Congestion can be overcome using AQM by enhancing the performance of TCP network middle nodes. Sabry and Nayl developed the linear quadratic (LQ)-servo controller as an AQM utilized in a TCP network to control congestion [3]. It has been suggested that the AQM system be used at gateways to enhance congestion control in TCP/IP networks. As a reliable response to handle network congestion, we created a Linear Quadratic Regulator (LQR) controller. The findings of our investigation show that system performance has significantly improved. We were able to successfully reduce network congestion by putting the LQR controller into use, which led to a more effective and responsive network infrastructure.

Doi: https://doi.org/10.54216/FPA.140123
Received: July 28, 2023 Revised: October 25, 2023 Accepted: December 22, 2023

Finally, this work emphasizes the significance of resolving internet network congestion. We have shown significant improvements in system performance through the use of sophisticated control techniques and the creation of the AQM model. The continued efforts to optimize and streamline internet network operations are greatly aided by this work, which ultimately benefits both users and companies.The incorporation of AQM is crucial for minimizing packet loss and enhancing network efficiency. To alleviate network congestion, Berbek and Oglah have introduced a novel approach, combining a hybrid intelligent system employing a PID controller with type 1 fuzzy logic. Additionally, they utilize Social Spider Optimization (SSO) as a router for AQM to optimize control parameters, ultimately reducing queue size errors [4]. This innovative approach simulates human-like decision-making processes, making it a noteworthy advancement in network congestion control.
The fuzzy-PID controller is utilized in [5] to regulate the nonlinear TCP/AQM model. The computer network's congestion is adjusted by this controller, which also commits regulated, pressured signaling characteristics.
In this work, the PI-controller is made to reduce traffic jams and try to maintain a good level of service in a variety of network conditions. Following is a list of the paper contributions:

- It is advised to use a PI controller to stabilize queue length with the least amount of delay and the quickest settling times.
- Through the use of various optimization techniques, the PI controller gains are adjusted.
- Linear Quadratic Regulator (LQR) controller is designed to enhance system performance.
- A comparison between the results of the used optimization techniques and the LQR controller is made in this work.

2. System Dynamics Of TCP/AQM

The stochastic differential equation analysis and fluid flow were used to construct the dynamic behavior of the TCP model, which is represented as follows [3] & [4]:

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

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

Where W is representing the typical TCP window size in packets, q is representing the length of the anticipated queue in packets, R is $Q/C + T_P$ the round-trip time in seconds, C is representing the link capacity in packets per second, T_P is representing the transmission delay, N is representing the load factor in TCP sessions, P is representing the likelihood of packet marking or loss, and t is representing the Time in seconds.
A multiplicative decline formulation TCP technique was established in equation (1). When a sender evaded congestion, the congestive window increased in one data segment in any R. If there is congestion, the window size is decreased to 50% of its initial size [3], [4].
The number of packets queued in the router varies when a packet is deducted from a packet that has already been sent out, which is explained by equation (2). Both the TCP window size W and the anticipated queue length q are positive and tiny. The packet probability (mark/drop) p values are also restricted to the range {0, 1}. Taking Laplace transformation of equations (1) and (2) is used to rearrange [4]:

$$\mathrm{P_{tcp}}(s) = \frac{W(S)}{P(S)} = \frac{\frac{R_{0C^2}}{2N^2}}{S + \frac{2N}{R_0^2 C}} \quad (3)$$

$$\mathrm{P_{queue}}(s) = \frac{q(S)}{W(S)} = \frac{\frac{N}{R_0}}{S + \frac{1}{R_0}} \quad (4)$$

That yield the following transfer function, as following:

$$P(s) = P_{tcp}(s) P_{queue}(s) e^{-sR_0} \quad (5)$$

Additionally, it might be stated as follows:

$$P(s) = \frac{\delta q(s)}{\delta p(s)} = \frac{\frac{c^2}{2N} e^{-sR0}}{(S + \frac{2N}{R_0^2 C})(S + \frac{1}{R_0})} \quad (6)$$

Doi: https://doi.org/10.54216/FPA.140123
Received: July 28, 2023 Revised: October 25, 2023 Accepted: December 22, 2023

Figure 1 displays a block schematic for a linear AQM control system. $P_{tcp}(s)$ describes the transfer function from the loss of the probability $p(t)$ to the window size $W(t)$, P_{queue}(S) describes the transfer function from $\delta W(t)$ to the queue length $q(t)$, the e^{-sR0} delay term, and the $C(s)$ system controller.

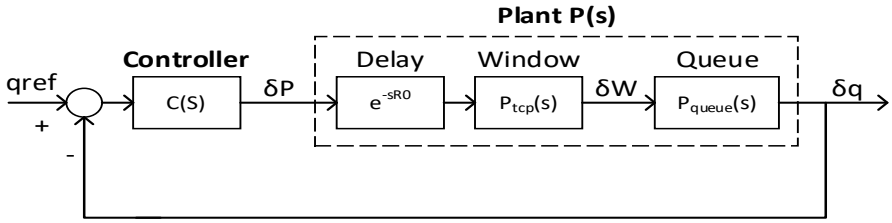

Figure 1: An AQM block diagram linearized as a feedback control.

Let suppose in the case of the study with $N = 60$, $R_0 = 0.253$ sec, and $C = 3750$ packets/sec. The resultant transfer function is in the equation (7). The topology of the network is depicted in Figure 2.

$$P(S) = \frac{117187.5\, e^{-0.253S}}{S^2 + 4.4524\, S + 1.9759} \tag{7}$$

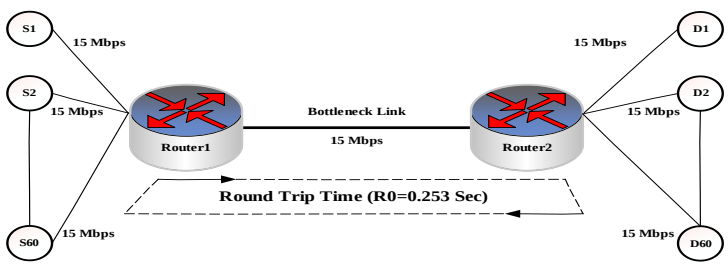

Figure 2: Case study network topology.

3. Controller Design

The primary objective of robust control system synthesis is to identify a control rule that maintains system output and error signals within the allowable signal. It is possible to implement feedback control in the form of a Proportional-Integral (PI) controller. It enhances the responsiveness and stability of the system by a combination of proportional and integral control actions. The equations describing the PI controller in the time domain as following [6] - [11]:

$$U(t) = K_p(t) + K_i \int e(t).dt \tag{8}$$

$$U(t) = r(t) - y(t) \tag{9}$$

Where $U(t)$ is representing the control signal at time t, $K_p(t)$: is representing the time-varying proportional gain, K_i is representing the integral gain, $e(t)$ is representing the error at time t, r(t) is representing the reference signal or setpoint at time t,and y(t) is representing the system's output or process variable at time t.

4. Evolutionary Algorithms (EA) for the Optimisation

The EA is different from experimenting with different values for the controller gains to find the sweet spot. Under specific conditions, the maximum and minimum of a function can be calculated using a method called an optimization algorithm.

4.1 Particle Swarm Optimization (PSO)

The PSO method is an optimization strategy based on stochastic inhabitance. Each member of the population is denoted by the term "particle," and each "flies" approximately in the multidimensional search space with a velocity that is constantly updated by the background of both the individual particle and the background of its neighbours or the entire swarm. It has found use in many contexts, including as in the optimization of functions, the study of artificial neural networks, the classification of patterns, and the management of fuzzy systems. The quick convergence to an optimal solution, simple computation and execution, and absence of complex calculations are all benefits of PSO. The PSO's accuracy is constrained by the fact that it can't improve upon a solution that's already nearly ideal. The algorithm

Doi: https://doi.org/10.54216/FPA.140123
Received: July 28, 2023 Revised: October 25, 2023 Accepted: December 22, 2023

can get caught in local optimum solutions, and as it matured, the intersection rate dropped drastically. The fundamental idea behind the PSO technique may be summed up as speeding each individual particle toward its P_{best} and G_{best} sites while experiencing unexpected or unpredictable metric accelerations at each time grade. The following velocity and position equations are used to modify the particles' velocity and location [12] – [15]:

$$v_i^{t+1} = q \text{ x } v_i^t + c_1 \text{ x } U \text{ x } (X_i^b - X_i^t) + c_2 \text{ x } U \text{ x } ((X_i^g - X_i^t) \quad (10)$$

$$X_i^{t+1} = X_i^t + X_i^{t+1} \quad (11)$$

Where v_i^t: is representing the particle velocity, X_i^t is representing the current particle position, q is representing the inertia weight, X_i^b and X_i^g are representing the optimum and the global optimum values, U is representing the random function, its values usually in the range of {0 and 1)}, and finally, c_1 and c_2 are learning factors.

In PSO, several parameters influence the optimization process. Here are the values of PSO parameters that used in this work: $number\ of\ particles\ =\ 300$, number of iterations = 1000, inertia weight = 0.2, cognitive coefficient = 1.5 and social coefficient = 1.5.

The fitness function (objective function) used in code to minimize is:

$$Fitness\ =\ \sum_{i=1}^{N} |q_{ref} - y_i| \quad (12)$$

Where N is representing the total number of time steps or data points over the specified time interval, q_{ref} is the desired reference value at time step i, and y_i is representing the actual system output at time step i.

The flowchart depicted in Figure 3 provides a visual representation that outlines the fundamental structure of the PSO algorithm.

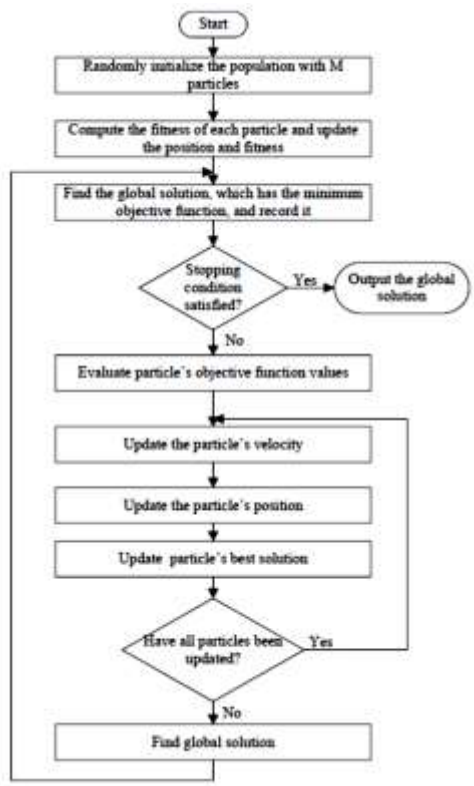

Figure 3: The PSO algorithm's overall structure.

4.2 Ant Colony Algorithm

It is an optimization technique that draws inspiration from nature and is based on how ants forage. It is employed to address a variety of combinatorial optimization issues. The fastest route that actual ants take between their colony and food sources is what inspired ACO. Pheromone is a molecule that ants leave on the ground while they investigate their surroundings. Ants may follow trails with increased pheromone concentrations thanks to this pheromone, which also acts as a means of communication among them. The choice of pathways is reinforced by a positive feedback loop

Doi: https://doi.org/10.54216/FPA.140123
Received: July 28, 2023 Revised: October 25, 2023 Accepted: December 22, 2023

since paths with more ants going on them build up more pheromone and become more alluring to other ants. ACO have three major equations the first one is pheromone update equation [16] – [20]:

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \alpha\eta_{ij}(t) \quad (13)$$

Where $\tau_{ij}(t+1)$ is representing the pheromone level on edge (i,j) at time $t+1$, $\tau_{ij}(t)$ is the pheromone level on edge (i,j) at time t, α is a constant representing the pheromone deposit factor, and $\eta_{ij}(t)$ is the heuristic information associated with edge (i,j) at time t.

The second equation is pheromone evaporation equation:

$$\tau_{ij}(t+1) = (1-\alpha)\,\tau_{ij}(t) \quad (14)$$

The final equation is the ant decision rule which used to determine the probability of an ant choosing a particular path in ant colony optimization (ACO). The equation is typically of the form:

$$P_{ij}(t) = \frac{\tau_{ij}(t)}{\sum_k \tau_{kj}(t)} \quad (15)$$

Where P_{ij} is representing the probability that an ant selects edge (i,j) at iteration t and the summation $\sum_k$ is taken over all possible edges k that an ant can consider for selection at iteration t.

In ACO several parameters influence the optimization process. Here are the values of ACO parameters that used in this work: number of $ants = 50$, number of $iterations = 100$, $alpha = 1$, $beta = 2$ and $evaporation\ rate = 0.5$. The fitness function (objective function) used is the same as equation (12).

Figure 4 depicts the Ant colony algorithm flowchart. The following flowchart, which is presented in Figure 4, serves as an illustration of the overall structure of the ACO algorithm.

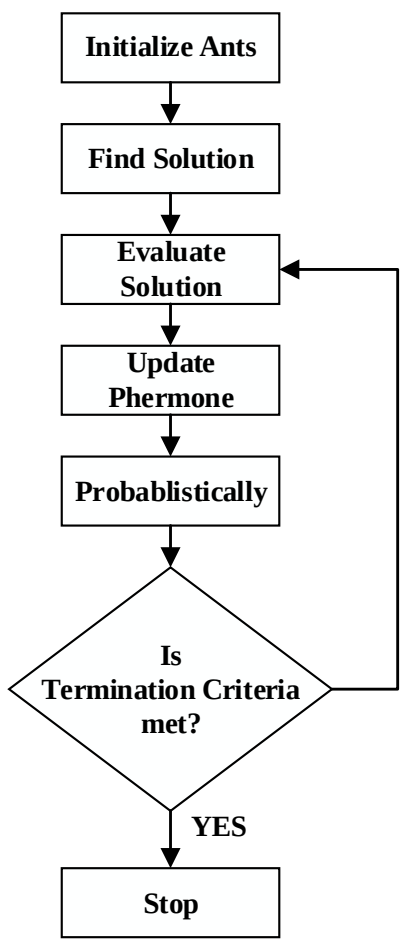

Figure 4: The ACO algorithm's overall structure.

Doi: https://doi.org/10.54216/FPA.140123
Received: July 28, 2023 Revised: October 25, 2023 Accepted: December 22, 2023

4.3 Simulated Annealing Algorithm

It is a probabilistic optimization approach that is modeled after the metallurgical annealing procedure, which involves progressively cooling a material to reduce imperfections and produce a more stable structure. By repeatedly looking for the best solution in a complicated search space that may contain several local optima, SA is used to tackle optimization challenges.

The fundamental idea behind the technique is to start out by allowing "bad" moves and then gradually becoming pickier as the optimization goes on. This idea is similar to the metallurgical annealing process, in which a material is heated first and then gradually cooled to a lower-energy state [21], [22].

$$P(accept) = e^{-T\Delta E} \tag{16}$$

Where ΔE is representing the change in energy (cost) between the new solution and the current solution. It's calculated as new cost−current cost, and T is the current temperature. A predetermined cooling schedule is used to lower the temperature itself. Exponential cooling is a popular cooling schedule in which the temperature is updated after each repetition.

In SA algorithm several parameters influence the optimization process. Here are the values of SA algorithm parameters that used in this work: $initial\ temperature = 100$, $final\ temperature = 0.1$, $cooling\ factor = 0.95$ and $max\ iterations = 1000$. The fitness function (objective function) used is the same as equation (12).

The following flowchart, which is presented in Figure 5, serves as an illustration of the overall structure of the SA algorithm.

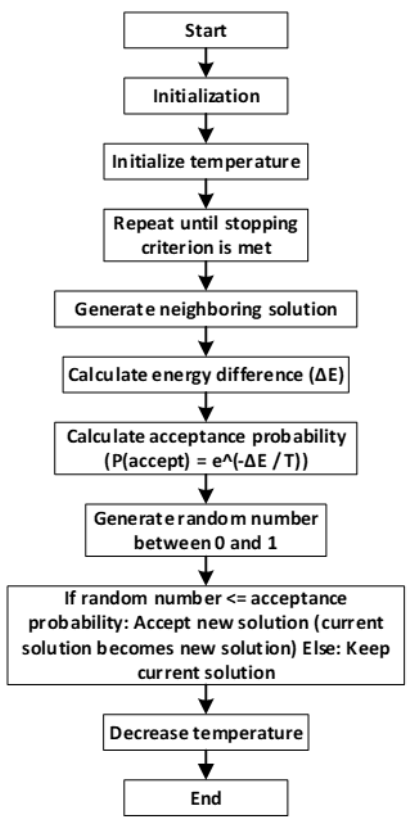

Figure 5: The SA algorithm's overall structure.

5. LQR controller design

This stage involves optimizing the control input based on a quadratic cost function. Here are the mathematical equations [3], [23]:

a) State-Space Equations: The system can be represented in state-space form as follows:

Doi: https://doi.org/10.54216/FPA.140123
Received: July 28, 2023 Revised: October 25, 2023 Accepted: December 22, 2023

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (17)$$
$$y(t) = Cx(t) + Du(t) \quad (18)$$

Where $\dot{x}(t)$ is representing the state vector at time t, $u(t)$ is the control input at time t, $y(t)$ is the output at time t, A is the state matrix, B is the input matrix, C is the output matrix, and D is the feedforward matrix.

b) **LQR Cost Function:** The LQR cost function to be minimized is given by:

$$J = \int_0^\infty (x^T Q x + u^T R u).dt \quad (19)$$

Where Q is representing the state weighting matrix, and R is the control input weighting matrix.

c) **LQR Gain Matrix K:** The LQR gain matrix K can be computed as:

$$K = \mathrm{R}^{-1}\mathrm{B}^{\mathrm{T}}\mathrm{P} \quad (20)$$

Where P is representing the solution of the algebraic Riccati equation which can be expressed as following:

$$A^{TP} + PA - PBR^{-1}B^T P + Q = 0 \quad (21)$$

d) **Control Law:** The control input u can be computed using the control law:

$$u = -Kx \quad (22)$$

e) **Closed-Loop System Equations:** The closed-loop system dynamics with the LQR controller is given by:

$$\dot{x}cl(t) = (A - BK)xcl(t) \quad (23)$$
$$ycl(t) = Cxcl(t) \quad (24)$$

Where $xcl(t)$ is representing the closed-loop state vector at time t and $ycl(t)$ is the closed-loop output at time t.

6. Results And Discussion

The parameters for the network topology simulation results presented in Figure 2 are as follows: With a propagation time of 0.2 seconds, N = 60, an average packet size of 500 bytes, and a bottleneck connection capacity of 15 Mbps between routers 1 and 2, the required input queue size has an irregular shape every 50 seconds. The maximum queue length for the AQM Router 1 is 800 packets. Other gateways use the AQM and drop Tail configurations that Router1 sets up.

$q_{ref} = \{300\ 0 < t < 50$
$200\ 50 < t < 100$
$400\ 100 < t < 150$
$200\ 150 < t < 200\}$

The q_{ref} and the close loop system response without any controller are shown in Figure 6.

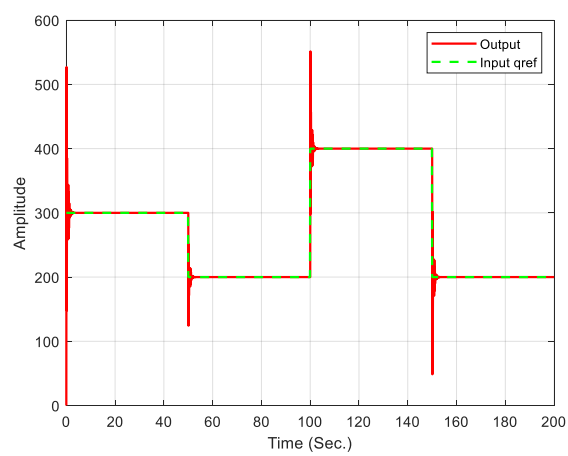

Figure 6: Output response and input reference signal $\boldsymbol{q_{ref}}$.

Doi: https://doi.org/10.54216/FPA.140123
Received: July 28, 2023 Revised: October 25, 2023 Accepted: December 22, 2023

From the close loop response, it is obvious that the output response does not match the input q_{ref} signal and the output signal have an overshoot and a high settling time where: settling time (T_s): 1.7530 seconds, rise time (T_r): 0.030359 seconds and overshoot Percentage: 75.6779%. That led us that we have to design a controller in order to improve system performance, manual tune PI controller is designed and the output response of PI controller is shown in Figure 7. The manual tuning gains of PI controller K_P and k_I are 0.0002 and 0.00007 respectively.

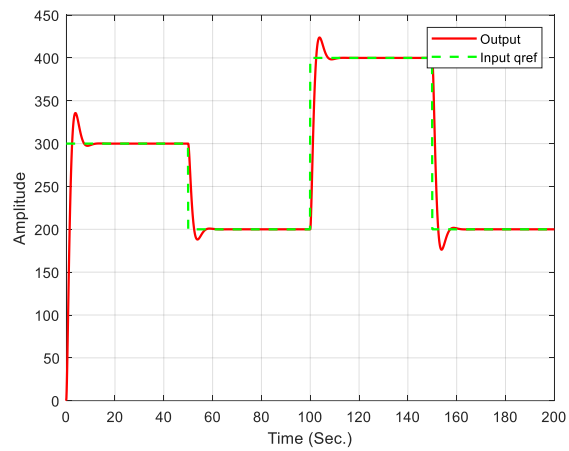

Figure 7: Output response of PI controller and input reference signal $\boldsymbol{q_{ref}}$.

The manual tuning PI controller parameters which are set as T_s: 6.4713 seconds, T_r: 2.1 seconds and Percentage overshoot: 11.9%.

The manual tuning of PI controller is based on personal experience by using trial and error and this method takes too long time and effort so the PI gains are tuned using an optimal technique, the output response of optimal techniques that used in this work are illustrated in Figures (8 to 10). The optimal techniques gains and the response parameters are shown in Table 1.

Table 1: Optimal techniques gains and response parameters.

Technique	K_P	K_I	Settling time (Sec.)	Rise time (Sec.)	Percentage Overshoot (%)
PSO Algorithm	0.13529	0.14283	2.3	1.4	$5e^{-5}$
ACO Algorithm	6.6042	0.61935	1.79	1.1	$-1.862e^{-06}$
SA Algorithm	5.6936	0.61401	1.8	1.1	$-1e^{-06}$

Doi: https://doi.org/10.54216/FPA.140123
Received: July 28, 2023 Revised: October 25, 2023 Accepted: December 22, 2023

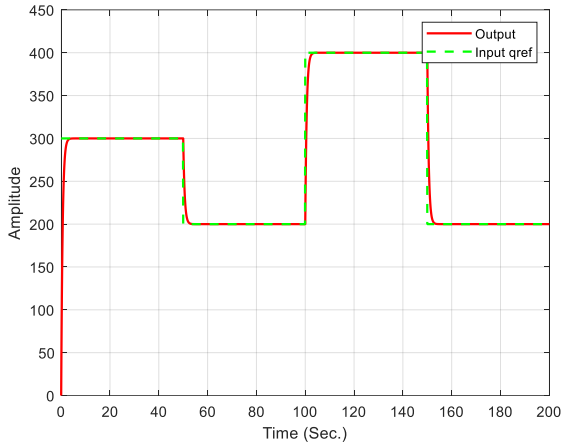

Figure 8: Output response of PSO-PI controller and input reference signal $\boldsymbol{q_{ref}}$.

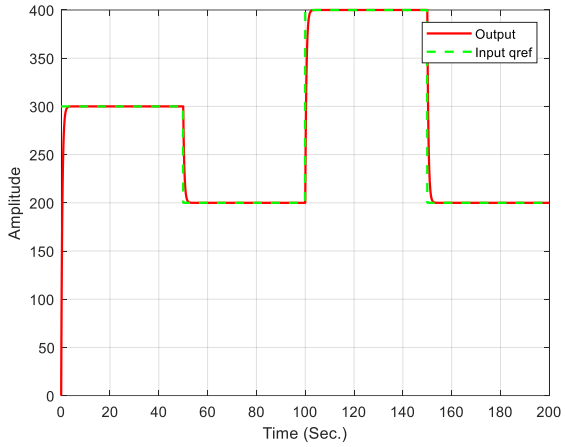

Figure 9: Output response of ACO-PI controller and input reference signal $\boldsymbol{q_{ref}}$.

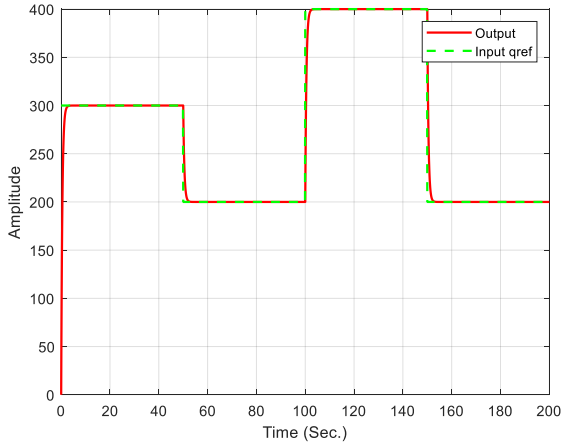

Figure 10: Output response of SA-PI controller and input reference signal $\boldsymbol{q_{ref}}$.

Doi: https://doi.org/10.54216/FPA.140123
Received: July 28, 2023 Revised: October 25, 2023 Accepted: December 22, 2023

The LQR controller gives the following parameters and the output response is shown in Figure 11. Where the response parameters are Ts = 0.8748, rise time (Tr) = 0.4913 and percentage overshoot = 0%.

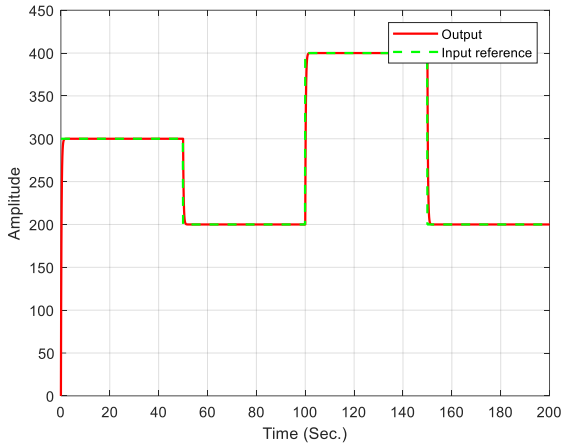

Figure 11: Output response of LQR controller and input reference signal q_{ref}.

7. Conclusion

Congestion, which can negatively affect network performance, is one of the major problems with internet networks. To solve this urgent problem, we presented an AQM model in this work. In order to fine-tune the *PI* controller, our method involved using a range of techniques, including manual tuning and efficient algorithms like PSO, ACO, and SA. With the intention of successfully controlling network congestion, this fine-tuning procedure was carried out using MATLAB.

As a reliable response to handle network congestion, we created a LQR controller. The findings of our investigation show that system performance has significantly improved. We were able to successfully reduce network congestion by putting the LQR controller into use, which led to a more effective and responsive network infrastructure. Finally, this study emphasizes the significance of resolving internet network congestion. We have shown significant improvements in system performance through the use of sophisticated control techniques and the creation of the AQM model. The continued efforts to optimize and streamline internet network operations are greatly aided by this work, which ultimately benefits both users and companies.

References

[1] E. Abharian and M. Alireza, "Hybrid GA-BF based intelligent PID active queue management control design for TCP network," in Proceedings of the 3rd International Conference on Electronics, Communications, and Technologies (ICECT), 2011.

[2] S. Jaineet and T. Maher, "Stability and Control of Network Congestion Control System," 2020.

[3] S. Sana and N. Thaker, "Particle swarm optimization-based LQ-servo controller for congestion avoidance," Iraqi Journal of Computers, Communications, Control & Systems Engineering (IJCCCE), vol. 19, no. 1, 2019.

[4] M. I. Berbek and A. A. Oglah, "Fuzzy Like PID Controller Based on SSO Design for Congestion Avoidance in Internet Router," Iraqi Journal of Computers, Communications, Control, and Systems Engineering (IJCCCE), vol. 21, no. 2, 2021.

[5] H. M. Kadhim and A. A. Oglah, "Congestion Avoidance and Control in Internet Router Based on Fuzzy AQM," Engineering and Technology Journal, vol. 39, part A, no. 2, pp. 233-247, 2021.

[6] R. P. Borase, D. K. Maghade, S. Y. Sondkar, and S. N. Pawar, "A review of PID control, tuning methods and applications," International Journal of Dynamics and Control, vol. 9, no. 2, pp. 818-827, 2021.

[7] O. Karahan, "Design of optimal fractional order fuzzy PID controller based on cuckoo search algorithm for core power control in molten salt reactors," Progress in Nuclear Energy, vol. 139, 103868, 2021.

[8] L. Wang, "PID control system design and automatic tuning using MATLAB/Simulink," John Wiley & Sons, 2020.

[9] X. Jin, K. Chen, Y. Zhao, J. Ji, and P. Jing, "Simulation of hydraulic transplanting robot control system based on fuzzy PID controller," Measurement, vol. 164, 108023, 2020.

Doi: https://doi.org/10.54216/FPA.140123
Received: July 28, 2023 Revised: October 25, 2023 Accepted: December 22, 2023

[10] A. Stooke, J. Achiam, and P. Abbeel, "Responsive safety in reinforcement learning by PID Lagrangian methods," in Proceedings of the International Conference on Machine Learning (ICML), 2020, pp. 9133-9143.
[11] P. J. Gaidhane, M. J. Nigam, A. Kumar, and P. M. Pradhan, "Design of interval type-2 fuzzy precompensated PID controller applied to two-DOF robotic manipulator with variable payload," ISA Transactions, vol. 89, pp. 169-185, 2019.
[12] D. Wang et al., "Particle swarm optimization algorithm: an overview," Soft Computing, vol. 22, no. 2, pp. 387-408, 2018.
[13] X. Yang, J. Yuan, and H. Mao, "A modified particle swarm optimizer with dynamic adaptation," Applied Mathematics and Computation, vol. 189, no. 2, pp. 1205-1213, 2007.
[14] L. Jia and X. Zhao, "An improved particle swarm optimization (PSO) optimized integral separation PID and its application on central position control system," IEEE Sensors Journal, vol. 19, no. 16, pp. 7064-7071, 2019.
[15] H. Ali et al., "LQR/sliding mode controller design using particle swarm optimization for crane system," Al-Nahrain Journal for Engineering Sciences, vol. 23, no. 1, pp. 45-50, 2020.
[16] A. H. Alaidi, C. S. Der, and Y. W. Leong, "Systematic review of enhancement of artificial bee colony algorithm using ant colony pheromone," International Journal of Interactive Mobile Technologies, vol. 15, no. 16, p. 173, 2021.
[17] M. Dorigo and T. Stützle, "Ant colony optimization: overview and recent advances," Cham: Springer, 2019.
[18] B. Guan, Y. Zhao, and Y. Li, "An improved ant colony optimization with an automatic updating mechanism for constraint satisfaction problems," Expert Systems with Applications, vol. 164, 114021, 2021.
[19] Charchekhandra, B., " The Reading and Analyzing Of The Brain Electrical Signals To Execute a Control Command and Move an Automatic Arm", Pure Mathematics for Theoretical Computer Science, Vol 1, 2023.
[20] M. Paniri, M. B. Dowlatshahi, and H. Nezamabadi-pour, "Ant-TD: Ant colony optimization plus temporal difference reinforcement learning for multi-label feature selection," Swarm and Evolutionary Computation, vol. 64, 100892, 2021.
[21] Nadweh, R., " On The Fusion of Neural Networks and Fuzzy Logic, Membership Functions and Weights", Galoitica Journal Of Mathematical Structures and Applications, Vol 7, 2023.
[22] G. Crescenzio and C. Vito, "A simulated annealing algorithm for scheduling problems," Journal of Applied Mathematics and Physics, vol. 7, pp. 2579-2594, 2019.
[23] T. Mohmmed, "LQR controller design for stabilization of non-linear DIP system based on ABC algorithm," Eastern-European Journal of Enterprise Technologies, vol. 2, pp. 36-44, 2023.

Doi: https://doi.org/10.54216/FPA.140123
Received: July 28, 2023 Revised: October 25, 2023 Accepted: December 22, 2023