FUZZY-GENETIC CONTROLLER FOR CONGESTION AVOIDANCE IN COMPUTER NETWORKS[1]

Prof. Dr. Mohammed Z. Al-Faiz[2], MIEEE **Assistant Lec. Ali M. Mahmood[3]**

Abstract

In this paper a fuzzy proportional-Integral (FPI) controller is designed as an Active Queue Management (AQM) in internet routers to improve the performance of PI controller for congestion avoidance in computer networks. Firstly the parameters of FPI controller are selected by trial and error method, but to get the best controller parameters the Genetic Algorithm (GA) is used as an optimization method for tuning the FPI parameters. The analytical results for linearized TCP/AQM model are presented in MATLAB version 7.0. From the obtained results, a faster response time as well as the regulation of the output to a constant value by the designed FPI controller is clearly observed and it is noted that the FPI controller provides good tracking performance under different circumstances for congestion avoidance in computer networks.

Keywords: *Active Queue Management, fuzzy logic controller, Genetic Algorithm, computer networks.*

الخلاصة

في هذا البحث تم تصميم المسيطر المضبب Fuzzy Proportional Integral (FPI) كمدير للطابور الفعال موجهات الانترنيت ولتحسين أداء المسيطر(PI) لتجنب الاختناق في شبكات الحاسوب. في بداية تصميم المسيطر المضبب تم اختيار معاملات معادلاته بطريقة المحاولة والخطأ. لكن للحصول على المعاملات المثلى للمسيطر استخدمت الخوارزمية الجينية كطريقة للحصول على أفضل المعاملات للمسيطر. إن النتائج وتحليلها للنموذج الخطي لنظام (TCP/AQM) تم تمثيلها بواسطة نظام MATLAB الإصدار السابع ولوحظ من النتائج التي تم الحصول عليها استجابة سريعة ومنتظمة للنظام مع المسيطر المضبب (FPI) مع أداء جيد لتجنب الاختناق في شبكات الحاسوب تحت مختلف الظروف.

[1] This paper was presented in the Engineering Conference of Control, Computers and Mechatronics Jan. 30- 31/2011, University of Technology.

[2] Computer Engineering Dept. Nahrain University.

[3] Control and Systems Eng. Dept.

1. Introduction

Congestion can be defined as filled to excess, or overcrowded; for example, highway congestion. Although, the best solution of congestion is to simply avoid situations where and when congestion is likely to occur, this strategy isnt always possible. Unfortunately, congestion occurs in many real world networking environments because there is always a bottleneck of some sort a slow computer, a low-speed link, or an intermediate node with low throughput [1]. Congestion in a computer network is a state

in which performance degrades due to the saturation of network resources such as communication links, processor cycles, and memory buffers. Network congestion has well recognized as a resource-sharing problem. When too many packets are contending for the same link, the queue overflows and packets have to be dropped. When such drops become common events, the network is said to be congested. It is anticipated that this results in better response compared to linear controllers due to the nonlinear nature of NNPI.

Several researches have been done in the field of congestion avoidance in traffic of computer networks. A brief description of these researches is submitted in the following paragraphs. In the last 80’s Jacobson and Karels [2] proposed the end-to-end congestion control is algorithms which forms the basic for the TCP congestion control. Its content that a TCP sender keeps a sending window (packets) rate according to the rate of dropped packets when a buffer becomes full in the router queue.

In the last 90’s, Floyd and Jacobson [3] presented the RED. Its mechanism is that packets are randomly dropped before the buffer of queue overflows. And, Braden et al. [4] proposed the enhanced end-to-end congestion control for AQM.

Misra et al.[5] developed a methodology to model and obtain numerically expected transient behavior of networks with AQM routers supporting TCP flows. The solution methodology scales well to a large number of flows. This modeling/solution methodology has a great potential in analyzing and understanding various network congestion control algorithms. Hollot et al. [6] used linearization to analyze a previously developed Non-linear model of the TCP/AQM. They linearized the model by using small-signal linearization about an operating point to gain insight for the purpose of feedback control to analyze a combined TCP and AQM model from a control theoretic standpoint and ability to present design guidelines for choosing parameters that lead to stable operation of the linear feedback control system.

Hollot et al. [7] proposed PI controller based on the linear control theory. The main contribution is to convert the congestion control Algorithm into the controller design problem within the framework of control theory in AQM system by studying a previously developed linearized model of TCP and AQM. The controller showed better theoretical properties than the well known RED controller.

Waskasi et al. [8] developed a new AQM algorithm based on neural networks. PI controller based on Artificial Neural Networks (ANN) is applied to AQM for the objective of congestion avoidance and control in middle nodes. The proposed controller is simple and can be easily implemented in high-speed routers. Neural Network PI (NNPI) dynamically adapts its parameters with respect to changes in the system.

Yann et al. [9] proposed an AQM based on the Lyapunov theory for time delay systems. With the help of Lyapunov-Krasovskii functional and using a state space representation of a linearized fluid model of TCP/AQM which is extended to the robust case where the delay in the loop is unknown.

Al-Hammouri [10] developed analytical characterization of the complete stability region of the PI controller for TCP/ AQM model .The analytical challenge is the presence of time-delays in the TCP-AQM feedback loop. The complete stability region provides an in-depth understanding of the performance of PI controller under different network parameters.

Kang et al. [11] proposed the LQ-Servo controller for AQM routers. The proposed controller structure is made by taking a traditional servo mechanism based on Linear Quadratic (LQ) approach. The proposed LQ-Servo controller can deal with a good tracking performance comparing with PI controller.

Kang et al. [12] developed the LQ-Servo controller based on loop shaping method for TCP/AQM router in order to meet such frequency domain design specifications as good disturbance rejection. The simulation results show that the proposed controller is more effective in getting the good tracking responses than PI controller for the varying reference queue size in AQM routers.

2. TCP/AQM Model

AQM has been extensively analyzed using control-theoretical methods. Control-theoretical approaches lead to stable, effective, and robust congestion control operation. In [5], the non-linear dynamic model for multiple TCP flows control has been developed based on fluid-flow theory to model the interactions of a set of TCP flows and AQM routers in computer networks which consist of a system of nonlinear differential equations. For the control theoretical analysis, it was approximated as a linearized constant model by small signal linearization about an operating point(W_0, q_0, p_0), see [6] for linearization details, which leads to the following:

$$\delta \dot{W}(t) = -\frac{2N}{R_0^2 C}\delta W(t) - \frac{R_0 C^2}{2N^2}\delta p(t-R_0)$$
$$\delta \dot{q}(t) = \frac{N}{R_0}\delta W(t) - \frac{1}{R_0}\delta q(t) \qquad (1)$$

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

$\dot{W}(t)$ denotes the time-derivative of $W(t)$, $\dot{q}(t)$ denotes the time derivative of $q(t)$, and

W : Expected TCP window size (packets)

q : Expected queue length (packets)

R_0: Round-trip time (seconds)

C : Link capacity (packets/second)

N : Load factor (number of TCP sessions)

p : Probability of packet mark/drop

t : Time

The expected queue length q and the expected TCP window size W are positive value and bounded quantities. And also, the probability of packet (mark /drop) p takes value only in [0 ,1] .

Taking the Laplace transform of equation (1) and rearranging the following transfer Taking the Laplace transform of equation (1) and rearranging the following transfer functions are obtained:

$$P_{tcp}(s) = \frac{W(s)}{p(s)} = \frac{\frac{R_0 C^2}{2N^2}}{s + \frac{2N}{R_0^2 C}} \qquad (2)$$

$$P_{queue}(s) = \frac{q(s)}{W(s)} = \frac{\frac{N}{R_0}}{s + \frac{1}{R_0}} \qquad (3)$$

So, the overall plant transfer function becomes:

$$P(s) = P_{tcp}(s) P_{queue}(s) e^{-sR_0} \qquad (4)$$

And can be expressed as:

$$P(s)=\frac{\delta q(s)}{\delta p(s)}=\frac{\frac{C^2}{2N}e^{-sR_0}}{(s+\frac{2N}{R_0^2C})(s+\frac{1}{R_0})} \qquad (5)$$

Thus, the block diagram of linearized AQM control system is shown in Fig.1. In this diagram *Ptcp* (*s*) denotes the transfer function from loss probability δ*p*(*t*) to window size δ*W*(*t*) , *Pqueue* (*s*) denotes the transfer function from δ*W*(*t*) to queue length δ*q*(*t*) , and *C*(*s*) denotes the transfer function of controller. Taking the Z-transform to Eq.(5),the designed plant transfer function is obtained after considering the sampling time half of R_0. Precisely and for consider the case study with N=60, C=3750 packets/sec and R_0=0.253 sec the following discrete transfer function are obtained.

$$P(z)=\frac{q(z)}{p(z)}=\frac{11252.46\,z^{-3}}{1-1.545\,z^{-1}+0.569\,z^{-2}} \qquad (6)$$

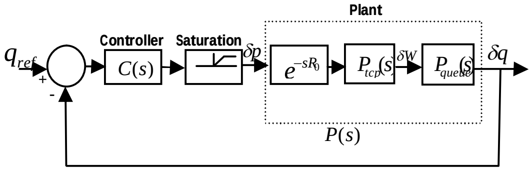

Fig.1:Block diagram of a linearized AQM as feedback control.

3. Controller Design

3.1. Conventional PI controller

The continuous-time PI controller is described by the following expression:

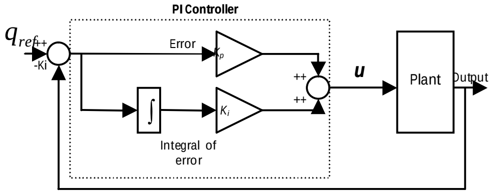

Fig.2: Block diagram of PI controller.

3.2. Fuzzy PI (FPI) Controller

The previous equation (6) gives a conventional PI-controller .To provide a formal methodology for representing, and implementing a human's heuristic knowledge about how to control a system, which may provide a new paradigm for non-linear systems and to get more robust controller that can cover a much wider range of operating conditions than PI controller and can operate with noise and disturbances of different nature, a fuzzy PI controller is designed as an intelligent controller for congestion avoidance in AQM routers. A block diagram of a fuzzy PI controller system looks like Fig.3.

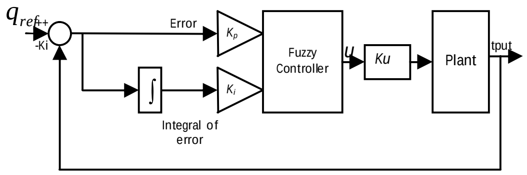

Fig.3: Block Diagram of FPI Controller with (e & ∫e).

From Fig3 it is noted that the formulation of control rules is difficult with the input variable sum-of-error (∑*e*) because its steady-state value is unknown for most control problems, because it may have the very wide universe of discourse [13]. So another configuration gaining more popular utilization by the designers which depends on the inputs (e , $\dot{e}$) which is used by moving the integration from the part preceding to a fuzzy controller to part following it, and integrate the output of a controller not the input. When the derivative, with respect to time, of the Equation (6) is taken, it is transformed into the following equivalent expression:

$$u(t)=K_p\times e(t)+K_i\times\int e(t)dt \qquad (7)$$

where, K_p and K_i are the proportional and the integral gain coefficients. A block diagram for a PI controller is shown in Fig.2.

the control signal it needs to integrate the output of controller as shown in Fig.4

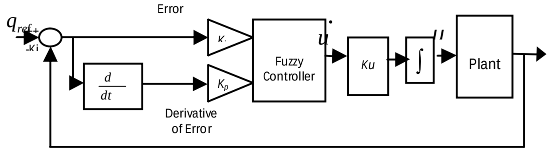

Fig.4: Block Diagram of FPI Controller with (e , $\dot{e}$)

3.3. Specifications of Fuzzy PI Controller

a) The (Universe of Discourse) UOD Partitions and Membership Type

The UOD of each input control variable (e , $\dot{e}$) is decomposed into five fuzzy sets .The linguistic values of these inputs are: (Negative Big) **NB**, (Negative Small) NS, (Zero) Z, (Positive Small) PS, (Positive Big) PB and the control Signal of output variable (u) is decomposed into seven fuzzy sets that have the linguistic values: (Negative Big) NB, (Negative Medium) NM, (Negative Small) NS, (Zero) Z, Medium) NM, (Negative Small) NS, (Zero) Z, (Positive Small) PS, (Positive Medium) PM, (Positive Big) PB. The UOD for inputs (e , $\dot{e}$) and output control variable (u) are normalized between (-1,1) and for simplicity and effectiveness, triangular and trapezoidal shapes are chosen as membership functions for inputs (e , $\dot{e}$) and Singleton for output (u) as shown in Fig.5.

b) Formation of Rule Bases and Defuzzification Method

The proposed rule base contains 25 rules, with a linguistic description of domain expert knowledge in the “if-then” form. The FPI controller rules are obtained by expertise and the trial-and-error method. The rule base of the designed FPI controller is shown in Table (1), where Mamdani fuzzy rules are used to perform the fuzzy rules with Mamdani

$$du(t)/dt = K_p \times \frac{de}{dt} + K_i \times e(t) \qquad (8)$$

So the controller output is not a control signal, but as a derivative of a control signal. To get

and the above ever mentioned sets give the better response. The first rule is outlined below, Rule 1:

If (e) is NB AND ($\dot{e}$) is NB THEN u is NB

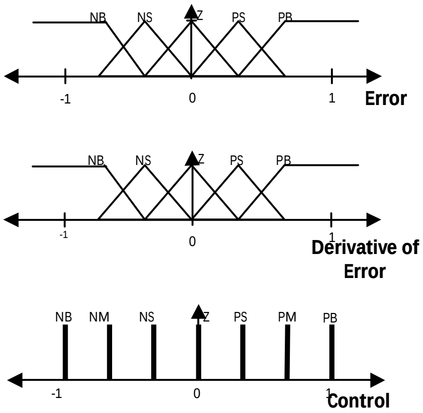

Fig.5: Fuzzy Input and Output Memberships with Normalized UOD.

Table (1): Fuzzy Rule Base.

u		*e*				
		NB	**NS**	**Z**	**PS**	**PB**
$\dot{e}$	**NB**	NB	NB	NM	NS	Z
	NS	NB	NM	NM	Z	PS
	Z	NM	NM	Z	PM	PM
	PS	NS	Z	PM	PM	PB
	PB	Z	PS	PM	PB	PB

3.4. FPI Based Genetic Algorithm

GA method is used as alternative to trial and error method since it is a suitable optimization method to find best FPI parameters (K_i, K_p, Ku) according to minimization of the criterion ITAE set in equation (8) to get the maximized fitness of minimum inference method that is used in a fuzzy rule to determine the rule outcome from the given rule input information. The center of gravity method is used in defuzzification to convert fuzzy sets to a real number. It worth mentioning that other sets of input and output membership function and fuzzy rules are used

is achieved or the value of fitness is fixed for next generations.

$$ITAE = \int_{0}^{t_f} t \left| e \right| dt \qquad (9)$$

where, t_f : the final time of simulation.

$$\text{Fitness}=1/(1+\text{ITAE}) \qquad (10)$$

4. Network Topology Scenario

Fig.6. shows the network case study taken, where the simulation is conducted for a single link (Bottleneck link) that has a bandwidth capacity **C=3750** packets/sec (corresponds to a **15** Mbps with packet size **500** bytes), and the same bandwidth capacity is used at other links, the Round Trip Time (**R$_0$**) is **0.253** second where the desired queue size is **200** packets and the propagation delay is **0.2** second. The number of TCP sessions (**N**) is **60** for source and destination, where applying the above parameters in equation (3.17) gets the overall TCP/AQM system transfer function as shown in equation (10). The maximum queue length in the AQM router Router1 is **800** packets. The AQM mechanism (PI or FPI) is configured at Router1, and drop Tail is used at other gateways.

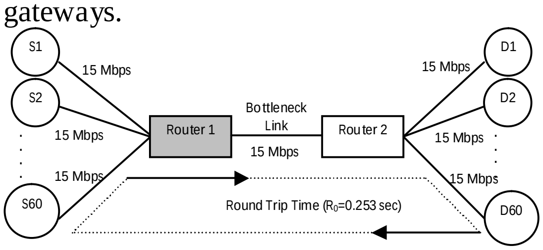

Fig.6: Network Topology Case Study.

5.Linearized TCP/AQM Model Simulation Results

The simulation of the linearized TCP/AQM model was done in MATLAB 7.0. Consider the TCP/AQM model with network parameters as set in pervious section and the reference input (queue size) which has rectangular form changes every 50 seconds as shown in equation (4.2). First the simulation is done for the system without controller as shown Fig.7.

$$q_{ref}=\begin{cases}300 & 0<t<50;\\ 200 & 50<t<100;\\ 400 & 100<t<150;\\ 200 & 150<t<200;\end{cases} \qquad (11)$$

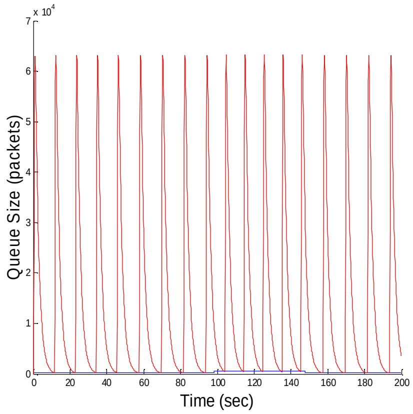

Fig.7: System Response without Controller.

From Fig.7 it is shown that the system without controller is unable to track the queue length around the queue length to the desired level, where the system goes into a sustained oscillation with high congestion exceeding the maximum buffer size. In order to eliminate this sustained oscillation and get better tracking performance a classical control (PI-controller) is applied. As it is shown in Fig.8. The PI controller parameters are selected by trial and error method. Thus the PI coefficients are $K_p = 0.3\times 10^{-5}$ and $K_i = 0.1\times 10^{-5}$ respectively. Although the PI controller shows good performance the system response is slow, so to overcome this drawback a FPI controller is designed to speed up the system response. Fig.9 shows the system response with PI and FPI

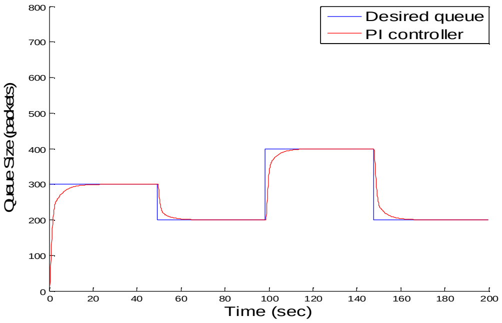

Fig.8: System Response with PI Controller.

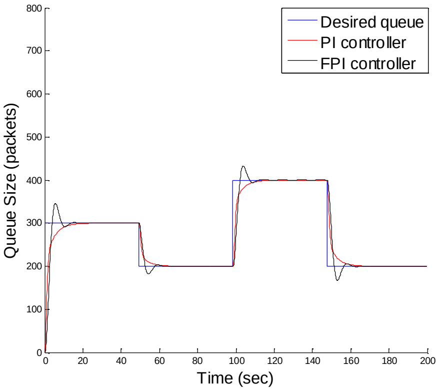

Fig.9: System Response with PI and FPI Controllers.

From Fig.9 above it is shown that the FPI controller speeds up the response comparing with PI controller, where the FPI controller parameters are selected by trial and error method as follows $K_p = 0.5\times10^{-3}, K_i = 0.9\times10^{-3}, Ku = 0.2\times10^{-2}$, so to get the best parameters to controller and to enhance the system response the Genetic Algorithm (GA) is used as a suitable optimization method for tuning FPI Parameters, where the population size is 100, crossover probability pc=0.9, mutation probability pm=0.05, for varying queue size in AQM the system response enhanced especially in decreasing the rising time and the settling time which means that the FPI with GA could speed up the system response, as a result it gives better congestion avoidance compared with FPI and classical PI controllers, as shown in Table (2) and Figure (10). It is worth mentioning that other sets of GA parameters

Table (2): TCP/QAM System Response Performance of PI, FPI and FPI Based GA with error criteria 5%.

Controller	Rise Time (t_r) (sec)	Overshoot $M_p\%$	Peak Time (t_p) (sec)	Settling Time (t_s) (sec)
PI	18	-	-	20
FPI	4	15	5	8
FPI based GA	2.5	7	3	4

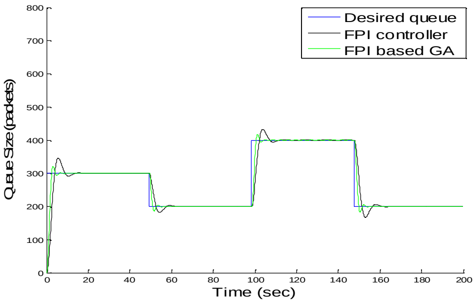

Fig.10: System Response with FPI and FPI Based GA.

6. Conclusions

From the design and the simulation results, it can be concluded that:

1- The designed FPI controller can deal with congestion problem with a good tracking performance about the desired queue size with high link utilization and faster system response observed as compared with routers.

2- By using the GA to optimally select the best FPI parameters the system response is improved as shown in table (2) which prove the efficiency of GA as suitable optimization method

3- the modeling and linearization of window and queue dynamics of the TCP/AQM model about an operating point to gain insight for the purpose of feedback control in design and analysis of AQM schemes.

References

[1] D. E. *Mcdysan* and D.Paw, "ATM & MPLS Theory & Application: Foundations of Multi-Service Networking", McGraw-Hill, 2002.

[2] V. Jacobson and M. J. Karels, "Congestion Avoidance and Control", *In SIGCOMM'ss*, 1988.

[3] S. Floyd and V. Jacobson, "Random early detection gateway for congestion avoidance", IEEE/ACM Transactions on Networking Vol.1, No. 4, August 1993.

[4] B. Braden, D. Clark and J. Crowcroft, "Recommendations on Queue

Management and Congestion Avoidance in the Internet", Network Working Group ,Request for Comments: 2309 *(RFC2309),* April 1998.

[5] 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", in Proc. ACM SIGCOMM, 2000. pp. 151-160.

[6] C. V. Hollot, V. Misra, D. Towsley, and W. B. Gong, "A Control Theoretic Analysis of RED", *In Proceedings of IEEE INFOCOM*, 2001.

[7] C. V. Hollot, V. Misra, D. Towsley, and W. B. Gong, "On Designing Improved Controllers for AQM Routers Supporting TCP Flows". *In Proceedings of IEEE INFOCOM, pages 1726–1734, 2001.*

[8] M. Y. Waskasi, M. J. Yazdanpanah and N. Yazdani, "a New Active Queue Management Algorithm Based On Neural Networks PI", *Control and Intelligent Processing Center of Excellence Electrical and Computer Engineering Department University of Tehran*, 2005

[9] Y. Labit, Y. Ariba and Gouaisbaut, "On Designing Lyapunov-Krasovskii Based AQM for Routers Supporting TCP Flows", UPS, 118 Route de Narbonne, F-31062 Toulouse, France, December, 2007.

[10] A. T. Al-Hammouri "Internet Congestion Control Complete Stability Region for PI AQM and Bandwidth Allocation in Networked Control". Ph.D. Thesis, January, 2008.

[11] K. M. Lee, J. H. Yang and B. S. Suh," Congestion Control of Active Queue Management Routers Based on LQ-Servo Control", *Engineering Letters, July*, Korea, 2008, 332-338.

[12] K. M. Lee, J. H. Yang and B. S. Suh,"Lq-Servo Control Design Based On Loop-Shaping Method For TCP/AQM Router" International Conference Modeling , Identification, and Control MIC February 16 -18 ,2009 Innsbruck ,Austria

[13] L. Reznik, "*Fuzzy Controllers*", Newnes, first edition, 1993.