Design of LQ-Servo Controller for Active Queue Management Routers

Kang Min Lee, Ji Hoon Yang, and Byung Suhl Suh

Abstract— **On AQM(Active Queue Management) routers, it is so well known that traditional method of control algorithm is used to maintain command of the reference queue size such as P(proportional), PI(P-Integral) and PID(PI-Derivative), initially RED(Random Early Detection). Also, performance reliance for these controllers is relatively strong in now days. Actual router that has complicated functions and is under the unpredictable perturbed environment need to be implemented to cope with so many troubles. On every adequate method, reserving all usable working areas is one of mandatory conditions. Usually, these working areas mean reservation of RAM (Random Access Memory) or SMA (Shared Memory Area) on ordinary router systems. The specified method for obtaining some working memories is getting the area which is already reserved for another task by dynamic method. For this job, the action for changing the command of reference queue size is strongly needed. In this paper, we adopt the LQ-Servo (Linear Quadratic Servo) controller into routers for getting system robustness.**

Index Terms— **AQM Router, Congestion Control, LQ-Servo Controller, Trajectory Tracking.**

I. INTRODUCTION

It comes to more important matters that congestion controls for the information network under the circumstances as the more and the faster data through the information network increasing. Jacobson and Karels[1] proposed the end-to-end congestion control algorithms which forms the basic for the TCP(Transmission Control Protocol) congestion control. It is a 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[2] presented the RED(Random Early Discard) which is that packets are randomly dropped before the buffer is overflow, and Braden et al. [3] proposed the enhanced end-to-end congestion control for Active Queue Management.

In recent years, The more needs for the congestion controllers having enough ability which is more logically predictable and reliable are occurred. For this reason, the traditional control algorithms which have been used only for mechanical or electrical systems are adopted to the area of congested network and their performances which are known as relatively good. Consequently, the more controllers which have various featured types have been adopted to the network congestion control area using control theories

On the issue of applying control theories to the network, especially AQM Router, Misra *et al.*[4] developed a methodology to model and obtain expected transient behavior of networks with Active Queue Management Routers supporting TCP flows. And Hollot *et al.* approximated its linearized model[5] using small-signal linearization about an operating point to gain insight for the purpose of feedback control, and designed the PI controller[6] based on the linear control theory. More recently, the robust AQM controllers have developed by the optimal control theory based on the Linear Matrix Inequalities (LMI) [7] and the robust μ-analysis technique [8] for the stability and performance issues in AQM. These controllers maintain the queue size which is given at system starting moment without any fluctuation. In this type of system operation case, it is impossible to use all resources (memories) being occupied by other applications because all those resources were privately dominated by previous one.

In order to cope with various unpredictable critical problems and share the memory resource peacefully, we induce congestion controller to more flexible one based on LQ-Servo structure of Athen [9]. Since it can follow the rectangular command input, we can change input reference command as the reference queue size in AQM system. Of course, we already knows the fact that the controllers don't have a servo structure can follow the varying reference command, however, it cannot have been satisfied the system stability, and most of all, system performances at all. Basically, this 'servo' operation enhances the usability of resources and furthermore, reduces unnecessary memory reservations. Above all, servo can induce condition that strengthen the system stability and can prevent router system from fatal error.

In this paper, we design LQ-Servo controller for the AQM router by combining two features as LQ optimization and servo structure. Consequently, we can get both satisfactions for the stability robustness and performance robustness. Each one is inborn attribute of LQ-Optimization and augmented servo structure.

This paper is structured as follows: In section II, rational linearized AQM router model is reviewed. We briefly consider the LQ optimization method and servo structure in section III and we finally derive implemented controller from consisting of partial state feedback and partial output feedback structures in section IV. In section V, some

Kang Min Lee is with the DSC Div. Advanced Development Group, SAMSUNG Techwin, Suwon, Korea (e-mail: kmry.lee@samsung.com , edwardjunior@hitel.net).

Ji Hoon Yang is with the Electrical Engineering Department, Hanyang University, Seoul, Korea (e-mail: openyj@hanyang.ac.kr).

Byung Suhl Suh is with the Electrical Engineering Department, Hanyang University, Seoul, Korea (corresponding author to provide phone: +82-2-2220-0364; fax: +82-2-2220-1856; e-mail: bssuh@hanyang.ac.kr).

simulation works are performed for non-servo structured controller and for the designed LQ-Servo controller with ns-2 simulator.

II. AQM MODEL

This paper uses most popular model of TCP/AQM that has linear formation[5]. It makes controller designing more easy and simple than non linear formation[4]. Brief reviews for the linear model are shown below.

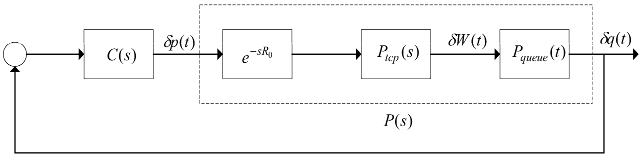

Figure 1. AQM Router Block Diagram

In Fig.1, $P_{tcp}(s)$ denotes the transfer function from loss probability $\delta p(t)$ to window size $\delta W(t)$, $P_{queue}(s)$ denoted the transfer function from $\delta W(t)$ to queue length $\delta q(t)$, and $C(s)$ denoted the transfer function of controller.

At first, the non linear differential equations are

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

$$\dot{q}(t) = \frac{W(t)}{R(t)} N(t) - C \quad (1)$$

where $\dot{W}$ denotes the time-derivative of W and subscript '0' means initial state.

$W \doteq$ Expected TCP window size (packets)
$q \doteq$ Expected queue length (packets)
$R \doteq$ Round-trip time ($= \frac{q}{C} + T_p$) (seconds)
$C \doteq$ Link capacity (packets/second)
$T_p \doteq$ Propagation delay (seconds)
$N \doteq$ Load factor (number of TCP sessions)
$p \doteq$ Probability of packet mark/drop

and linearization for 'small signal variation' at operating point is accomplished by conditions,

$$\dot{W}(t) = 0 \;\Rightarrow\; W_0^2 p_0 = 2$$
$$\dot{q}(t) = 0 \;\Rightarrow\; W_0 = \frac{R_0 C}{N} \quad (2)$$

And take the first chance of RTT as a input delay

$$R_0 = \frac{q_0}{C} + T_p \quad (3)$$

We finally get the dynamics of TCP/AQM router

$$\delta\dot{W}(t) = -\frac{2N}{R_0^2 C}\delta W(t) - \frac{R_0 C^2}{2N^2}\delta p(t-R_0)$$
$$\delta\dot{q}(t) = \frac{N}{R_0}\delta W(t) - \frac{1}{R_0}\delta q(t) \quad (4)$$

where

$$\delta W(t) \doteq W - W_0$$
$$\delta q(t) \doteq q - q_0$$
$$\delta p(t) \doteq p - p_0 \quad (5)$$

Let the state variable $x(t)$ of (4) be defined as:

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} \delta W(t) \\ \delta q(t) \end{bmatrix} \quad (6)$$

The (4) can be represented with state-space mode:

$$\dot{x}(t) = A_p x(t) + B_p u(t-R_0)$$
$$y(t) = C_p x(t) \quad (7)$$

where $y(t) = \delta q(t)$ is an output variable, $u(t) = \delta p(t)$ is a control variable.

And also, the system matrix, input matrix and output matrix of (7) can be expressed as following:

$$A_p = \begin{bmatrix} -\frac{2N}{R_0^2 C} & 0 \\ \frac{N}{R_0} & -\frac{1}{R_0} \end{bmatrix},\; B_p = \begin{bmatrix} -\frac{R_0 C^2}{2N^2} \\ 0 \end{bmatrix},$$
$$C_p = \begin{bmatrix} 0 & 1 \end{bmatrix} \quad (8)$$

In this paper, R_0 as an input delay on (7) is ignored because of its mere affection on entire control system. And one of the AQM system state variables W that can't be measured exactly on router itself is simply approximated to the average incoming packet size. In practice, router can observe and detect total incoming packets which have mixed form. Thus, it can count the total number of packets that comes from every end nodes at every sampling period.

The window size is

$$W = Incoimg\ Packets\ /\ N \quad (9)$$

III. LQ-SERVO

A. Tracking Method

In order to follow a reference input command trajectory which is continuously varying, our controller must have at least one integrator which is associated with proper state variables. For the AQM system model, the reference queue size is the reasonable state variable. In this paper, rectangular reference input commands are used for the test of continuous varying command. Thus just one integrator is adapted to

AQM system.

Integrator adoption process which is finished to augmented system matrix A_p is formed via hiring new state variable

$$x_3(t) = \int_0^t q(\tau)\, d\tau \tag{10}$$

So, the augmented state-variable descriptions become

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} = \begin{bmatrix} \delta W(t) \\ \delta q(t) \\ \int_0^t q(\tau)\, d\tau \end{bmatrix} \tag{11}$$

Then, the state-space model of (7) becomes as following (12)

$$\begin{aligned} \dot{x}(t) &= A_{aug} x(t) + B_{aug} u(t - R_0) \\ y(t) &= C_{aug} x(t) \end{aligned} \tag{12}$$

And also, the system matrix, input matrix and output matrix can be presented, respectively,

$$A_{aug} = \begin{bmatrix} A_p & 0 \\ C_p & 0 \end{bmatrix},\ B_{aug} = \begin{bmatrix} B_p \\ 0 \end{bmatrix},$$
$$C_{aug} = \begin{bmatrix} C_p & 0 \end{bmatrix} \tag{13}$$

It is certain that both system matrices A_p and A_{aug} has same eigen values.

B. LQ(Linear Quadratic) Optimization

For every control systems, stability and performance robustness satisfaction are the most important issues. Stability and certain portion of performance are satisfied by bringing LQ design method.

Shortly, in LQ design method, consider the cost function as:

$$J = \int_0^\infty \{x^T(t) \cdot Q_J \cdot x(t) + u(t) \cdot R_J \cdot u(t)\} dt \tag{14}$$

with the assumption that a weighting matrix Q_J is symmetric and positive semi-definite, and a weighting factor R_J is positive value.

Then, we use the general control law for regulating

$$u(t) = -G\, x(t) \tag{15}$$

where $G = -R_J^{-1} B_{aug}^T K$ and $K = K^T$ is a solution matrix of the algebraic Riccatti's equation:

$$KA_{aug} + A_{aug}^T K + Q_J - KB_{aug} R_J^{-1} B_{aug}^T K = 0 \tag{16}$$

For the LQ-Servo structure, this optimization method properly will be applied to the AQM System for tracking after system augmentation job will be completed. At the results, performance robustness would be better than current regulator's.

IV. Implementation of LQ-Servo For AQM

In order to implement for LQ-Servo, the complete system matrix, input matrix and output matrices of (13) can be obtained by substituting (8) into (13), respectively,

$$A_{aug} = \begin{bmatrix} -\dfrac{2N}{R_0^2 C} & 0 & 0 \\ \dfrac{N}{R_0} & -\dfrac{1}{R_0} & 0 \\ 0 & 1 & 0 \end{bmatrix},\ B_{aug} = \begin{bmatrix} -\dfrac{R_0 C^2}{2N^2} \\ 0 \\ 0 \end{bmatrix},$$
$$C_{aug} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \tag{17}$$

We already have all enough conditions. The rest job is just locating system, input and output matrices to every appropriate position and get control gains that previously denoted G in (15) through the CARE(Control Algebraic Riccati Equation) (16).

Remark 1. It is noted that we can set the weighting matrix $Q_J = I$ and the weighting factor $R_J = 2$ for convenience in this paper.

In practical Implementation, conditions (5) must be verified. At first, W_0 can be derived by certain constant values R_0, C, N. And q_0 have to be reference input that is continuous varying. State variables W and q are coupled with observing queue length and (9) respectively. The rest state variable x_3 is a value of accumulated $\delta q(t)$. This x_3 may strengthen the performance robustness on tracking issue.

It is noted that the controller structure of the proposed controller, which is LQ-Servo controller, is employed to a control law like as (15) directly, using ns-2 [10] in this paper

V. Simulation

We can verify the tracking performance of some network congestion controller by using ns-2 [10].

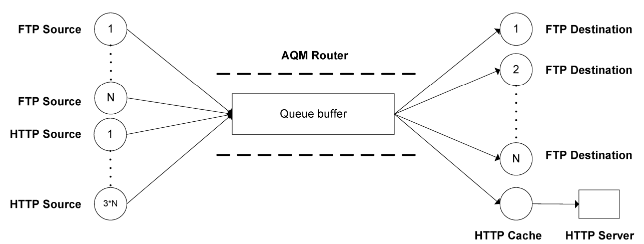

Figure 2. AQM Router Diagram

For the first time, PI controller [6] which has non trajectory tracking structure is tested as a comparative reference. Fig. 3 shows its performance for the continuous varying reference command. The reference input (queue size) has rectangular form and changes every 50seconds. Initial value start with queue size 300 and then, reference queue size is changes to 200 and 500, 200. Sampling time parameter is set to 1/160second for all experiments

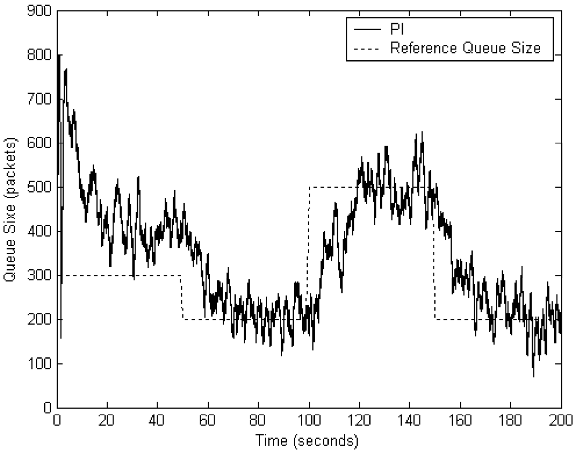

Figure 3. Tracking performance of the PI controller

Next, some experiments can verify the tracking performance of the LQ-Servo controller with some different AQM model parameters.

A. Experiment I

To evaluate tracking performance, we compare LQ-Servo with PI controller at first on the ns-2. (17), the continuous (analog) system and input matrices are

$$A_{aug} = \begin{bmatrix} -0.5288 & 0 & 0 \\ 243.9024 & -4.0650 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \; B_{aug} = \begin{bmatrix} -480.4688 \\ 0 \\ 0 \end{bmatrix}$$

In order to simulate, this continuous model must convert to discrete one. Simply, command ‘c2d’ makes continuous model into appropriate one on Matlab.

Finally, we can get the control gain for the experiment 1.

$$G = \begin{bmatrix} -0.4373 & -0.1607 & -0.1662 \end{bmatrix}$$

Both comparative reference PI and LQ-Servo controllers have same values of the network parameters.

R_0 = 0.246 (second)
C = 15Mbps (3750 packets/sec)
N = 60

The reference input (queue size) also has rectangular form and changes every 50seconds. Initial value start with queue size 300 and then, queue size is changes to 200 and 500, 200. The result is shown in Fig. 4.

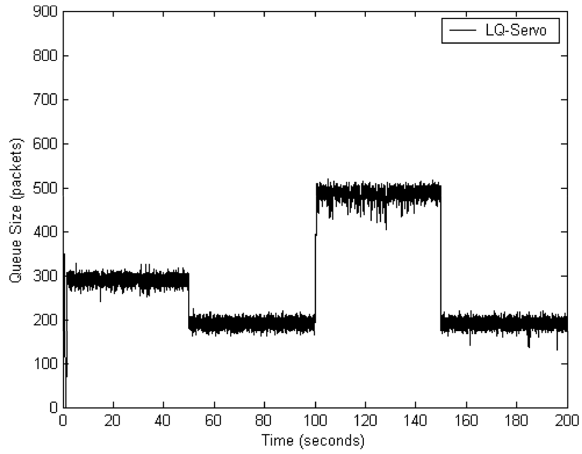

Figure 4. Tracking performance of LQ-Servo controller for experiment I

B. Experiment II

Simulation is accomplished with similar to experiment I, except that more load factors(N s) are added and also link capacity (C) is reduced.

R_0 = 0.246 (second)
C = 8Mbps (2000 packets/sec)
N = 120

The continuous (analog) system and input matrices are

$$A_{aug} = \begin{bmatrix} -1.9829 & 0 & 0 \\ 487.8049 & -4.0650 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \; B_{aug} = \begin{bmatrix} -34.1667 \\ 0 \\ 0 \end{bmatrix}$$

Control gain for the experiment II.

$$G = \begin{bmatrix} -3.4654 & -0.4220 & -0.4440 \end{bmatrix}$$

The reference input (queue size) also has rectangular form and changes every 50seconds. Initial value start with queue size 300 and then, queue size is changes to 400 and 200, 350. The result is shown in Fig. 5.

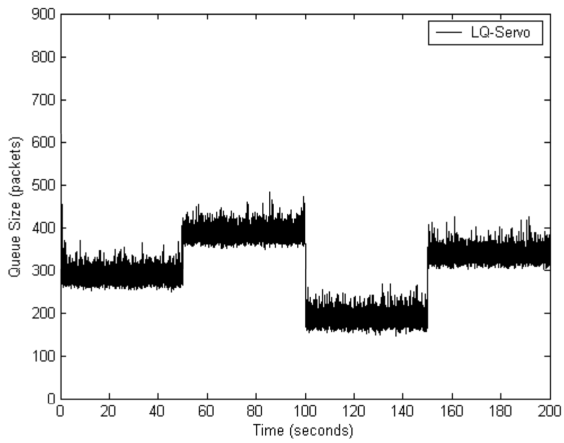

Figure 5. Tracking performance of LQ-Servo controller for experiment II

Experiment I and II show the results comparison of the proposed LQ-servo controller and PI controller with respect to changed reference queue size. These results show that the proposed controller has the better performance in the trajectory tracking problem on AQM routers.

VI. Conclusion

This paper presents the LQ-Servo controller deal with a good tracking performance. This controller structure is made by taking a traditional servo mechanism based on LQ optimization method. 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. On the LQ-Servo simulation result, not only steady state wave form has a stable tendency but also it shows the better performances at every reference changing point.

References

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

[2] S. Floyd and V. Jacobson. "Random Early Detection gateways for Congestion Avoidance," IEEE/ACM Transactions on Networking, vol 1, no. 4, 1997.

[3] B. Braden et al. "Recommendations on Queue Management and Congestion Avoidnace in the Internet," RFC2309, April 1998.

[4] V. Misra, W.-B. Gong, and D. Towsley. "Fluid-based analysis of a nerwork of AQM routers supporting TCP flows with an application to RED," in Proc. ACM SIGCOMM, 2000. pp. 151-160.

[5] C. V. Hollot, V. Misra, D. Towsley, and W. B. Gong, "A control theoritic analysis of RED," in Proc. IEEE INFOCOM, 2001.

[6] C. V. Hollot, V. Misra, D. Towsley, W.-B. Gong. "Analysis and Design of Controller for AQM Routers Supporting TCP Flows," IEEE Transcations on Automatic Control, vol. 47, no. 6, pp. 945-959. 2002.

[7] M. M. A. E. Lima, N. L. S. Fonseca and J. C. Geromel. "An Optimal Active Queue Managemnet Controller," IEEE Communications Society, pp. 2261-2266. 2005

[8] Q. Chen and O. W. W. Yang. "On Designing Traffic Controller for AQM Routers Based on Robust u-Analysis," IEEE Globecom, pp. 857-861, 2005.

[9] M. Athen, "Multivarible Control System," Athen's Control Lecture Note, MIT.

[10] "ns-2 Network Simulator," Obtain from http://www.isi.edu/nsnam/ns/.