A New PI and PID Control Design Method for Integrating Systems with Time Delays

DENİZ ÜSTEBAY
Bilkent University
Dept. of Electrical & Electronics Eng.
06800 Ankara
TURKEY
deniz@ee.bilkent.edu.tr

HİTAY ÖZBAY
Bilkent University
Dept. of Electrical & Electronics Eng.
06800 Ankara
TURKEY
hitay@bilkent.edu.tr

NAZLI GÜNDEŞ
Univ. of California, Davis
Dept. of Electrical & Computer Eng.
95616 CA
USA
angundes@ucdavis.edu

Abstract: Allowable PI and PD controller gains for a class of unstable MIMO systems with input/output delays have been investigated in [5, 12]. Using the results of these studies, we propose a new method for tuning the parameters of PI and PID controllers for integrating processes with time delays. As an application of this method, we design PI and PID controllers for Active Queue Management of TCP flows and illustrate performance of these controllers by packet level simulations in ns-2.

Key–Words: PID Controllers, Time Delay, Integrating Processes, Active Queue Management

1 Introduction

Proportional-Integral-Derivative (PID) controllers are widely used in various engineering applications since they are easy to implement using low storage memory and their computational complexity is low, [1]. Many different types of tuning guidelines have been proposed in the literature, e.g. [1], [17], as design of PID controllers imply tuning of PID parameters. In this paper we consider integrating systems with time delays (i.e. the plant contains an integrator, a time delay, and possibly other stable terms). Tuning of PID controllers for this type of unstable time delay systems is an active research area, [8], [15], [17].

In a recent study, stabilizing PID controllers are obtained for a class of MIMO (multi-input multi-output) unstable plants with delays in the input and output channels, [12]. Using the results of this study resilient PI (Proportional-Integral) and PD (Proportional-Derivative) controllers with largest allowable controller gain intervals are investigated for plants with at most two unstable poles, [5]. In this paper, we use the results of these two studies to obtain the largest allowable gain intervals for a given set of nominal plant parameters and we propose to select the center of these intervals as the optimal gains of the controller. Controllers with perturbed gains can be seen as "optimal" controllers for plants with perturbed nominal parameters. So, the controllers proposed here are expected to work for a wide range of plant parameters. Controllers using this method can be applied to integrating time delay systems appearing in data flow control in computer networks, target tracking problems in robotics applications, and material transport systems encountered in process control.

In this work, we consider the Active Queue Management (AQM) of TCP flows as an application example. AQM tries to find a solution to congestion which is one of the most important problems in computer networks. When some of the links are congested, buffers at the routers are full, which means incoming packets are lost and re-transmission occurs. This leads to long return-trip-times (RTT), i.e. delays, and may even result in an instability in the network (e.g. congestion collapse in the Internet), [4]. On the other hand, having an empty queue at a buffer means that the link capacity is not fully used, i.e. network resources are under-utilized in this case. Therefore, the goal of AQM is to keep the queue size at the buffers at a certain desired level. For this purpose, most AQM schemes mark the Explicit Congestion Notification (ECN) bit of the packets passing through the link according to a certain rule. In early AQM methods, such as RED [3] and REM [2], packet marking probability (control signal) is a static function of the queue (or average queue). First attempts to design a dynamic controller appear in [6, 7], where the authors use the fluid model of the TCP flows developed in [10] to design a PI controller. This AQM scheme is currently implemented in ns-2, [11]. It will be taken as the "benchmark" design for comparisons with the alternative AQM controller tuning method proposed here.

Remaining of the paper is organized as follows. Details of the controller parameter tuning is given in Section 2. Application of this tuning method to the

AQM problem and comparisons with the benchmark PI design can be found in Section 3. Concluding remarks are made in Section 4.

2 PI/PID Control for Integrating Processes with Time Delay

One of the most attractive features of PID (Proportional-Integral-Derivative) controllers is their simplicity: in the most general form they are second order systems

$$K_{pid}(s) = K_p + \frac{K_i}{s} + K_d s \tag{1}$$

and can be implemented digitally by discretization. The main issue in PID control design is the "optimal" tuning of the gains, K_p, K_i, K_d.

Allowable ranges of PI and PD controller gains has been investigated in [5, 12]. Here we use these results to derive a controller for a plant whose transfer function $G_{pq}(s)$ is in the form

$$G_{pq}(s) = \frac{K_o}{s}\, e^{-hs} g(s), \quad h > 0 \tag{2}$$

where $g(s)$ is an arbitrary stable rational transfer function satisfying $g(0) = 1$. For notational convenience we define $f(s) := e^{-hs} g(s)$ and let $G(s) = \frac{K_o}{s}\, g(s)$ be the finite dimensional part of the plant. A coprime factorization of $G(s)$ is in the form $G(s) = X(s)Y(s)^{-1}$ with $X(0) = sG(s)\ |_{s=0} \neq 0$. In our case, $Y(s) = s/(as+1)$, for any $a > 0$, and since $g(0) = 1$, we have $X(0) = K_o$. Then a PI controller is in the form

$$K_{pi}(s) = \alpha X(0)^{-1}(1 + \frac{\gamma}{s}) \tag{3}$$

for $\alpha \epsilon \mathbb{R}$ and $\gamma \epsilon \mathbb{R}$. The bound on the integral action gain γ can be defined by $0 < \gamma < \gamma_{max}$, where $\gamma_{max}^{-1} = \| \frac{\frac{\alpha}{s} f(1+\frac{\alpha}{s} f)^{-1} - 1}{s} \|_\infty$ with α satisfying $0 < \alpha < \| \frac{f(s)-1}{s} \|_\infty^{-1}$. Let us define $\theta := \| \frac{f(s)-1}{s} \|_\infty$. Then the lower bound for γ_{max}^{-1} can be found as

$$\gamma_* = \alpha(1 - \alpha\theta) \leq \gamma_{max}. \tag{4}$$

With straightforward calculation it can be shown that the optimal α maximizing γ_* is $\alpha_* = \frac{1}{2\theta}$ and the maximal γ_* corresponding to this optimal α is $\gamma_{*,max} = \frac{\alpha_*}{2}$.

Here we propose to choose controller's proportional gain as α_* and integral gain as $\frac{\gamma_{*,max}}{2}$, which is the center of the maximal interval. The resulting PI controller is in the form

$$K_{pi}(s) = \frac{1}{2\theta}\, \frac{1}{K_o}\, (1 + \frac{1}{8\theta s}). \tag{5}$$

The computation of (5) above is rather simplified thanks to conservatism introduced by (4). In fact, γ_{max}^{-1} is a function of α:

$$\gamma_{max}^{-1} \;=\; \sup_\omega \left| \frac{1}{j\omega + \alpha f(j\omega)} \right| =: Q(\alpha).$$

This norm should be calculated for every α in the interval $0 < \alpha < 1/\theta$. Then γ_{max}^{-1} could be chosen as the minimum of $Q(\alpha)$. However, this method requires $M * N$ calculations for M being the ω grid points for $\mathcal{H}_\infty$ norm computation and N being the α grid points. Using (4) we skip the H_∞ norm computation for every fixed α. The conservatism introduced by (4) is currently under investigation.

For a PD controller written in the form $\widehat{K}(1 + \widehat{K}_d s)$, [12] determined the optimal $\widehat{K}_d$ maximizing the allowable interval for the overall controller gain $\widehat{K}$ within the framework of the approach introduced in [5]. In this paper, we propose to take this optimal derivative action gain, and the center of the maximal allowable interval for $\widehat{K}$, for the plant (2). For this purpose, define $\rho(\omega) := |f(j\omega)|$ and $\phi(\omega) := \angle f(j\omega)$ as the magnitude and phase functions. Then set $\eta(\omega) := \left| \frac{\rho(\omega) - cos(\phi(\omega))}{\omega} \right|$. Assume that $\eta(\omega)$ has a maximum, η_o, which is attained at a single frequency ω_o. Then, define $\widehat{K}_d := -\frac{\sin(\phi(\omega_o))}{\omega_o\, \rho(\omega_o)}$. With the above definitions, the PD controller is

$$K_{pd}(s) = \frac{1}{2K_o\eta_o}\, (1 + \widehat{K}_d s). \tag{6}$$

The above PD controller design technique is valid for all plants in the form (2) where $g(s)$ can be any stable rational transfer function satisfying $g(0) = 0$, with one caveat: one has to check that the corresponding η has a single maximum. For a first order strictly proper $g(s)$ (as in the AQM problem, to be shown below) this is automatically satisfied.

A PID controller can be obtained from the product of PI and PD controllers. So, we will design a PD controller, then design a PI controller for the PD controlled open loop plant $G_{pq}(s)K_{pd}(s)$. For this purpose we define

$$G_{pq}(s)K_{pd}(s) \approx \frac{K_o}{s}\, e^{-hs} g(s)\, \frac{1}{2K_o\eta_o}\, (1 + \widehat{K}_d s)$$

$$=: \frac{1}{2\eta_o} \frac{e^{-hs} g_o(s)}{s} \quad g_o(s) := g(s)\, (1 + \widehat{K}_d s)$$

$$\theta_o := \| \frac{g_o(s) - 1}{s} \|_\infty.$$

$$K_{pi}(s) = \frac{1}{2\theta_o}\, 2\eta_o\, (1 + \frac{1}{8\theta_o s}).$$

So, the PID controller is

$$K_{pid}(s) = \frac{(1+\widehat{K}_d s)}{16K_o\theta_o^2}\,\frac{(1+8\theta_o s)}{s}. \qquad (7)$$

3 Application to AQM

3.1 Fluid Model of TCP Flows

For AQM design a PI tuning method is proposed in [6, 7] using a dynamical model of the TCP. This model is introduced in [10], and has been tested and used by many researchers, see e.g. [9], [13], [14], [16], [19], and references therein. By making fluid approximations the model is expressed in continuous time domain as follows, [6]

$$\dot{W}(t) = \frac{1}{R(t)} - \frac{W(t)\,W(t-R(t))}{2\;R(t-R(t))}p(t-R(t)) \quad (8)$$

$$\dot{q}(t) = N(t)\frac{W(t)}{R(t)} - c(t) \quad \text{when } q(t) > 0 \quad (9)$$

where W is the TCP window size, q is the queue length, N is the number of TCP flows passing through the link, c is the link capacity (outgoing flow rate), $R(t) = T_o + \frac{q(t)}{c(t)}$ is the RTT (total delay in the feedback path), T_o is the propagation delay, and p is the probability of a packet being marked. In this system p can be seen as the control input generated by a feedback from q, i.e. $p = \mathcal{K}(q)$ where $\mathcal{K}$ is the feedback control operator. First AQM schemes developed, e.g. RED, use static maps from q to p. Dynamic controllers are developed recently using control theoretic design methods. Clearly, the above system is nonlinear and it depends on an implicit function $R(t-R(t))$. In general, "optimal" controllers for such nonlinear dynamical systems are difficult to obtain. Typically, linearization is done around an equilibrium. Let $q(t) = q_o + \delta_q(t)$, $W(t) = W_o + \delta_W(t)$, $c(t) = c_o + \delta_c(t)$, $N(t) = N_o + \delta_N(t)$, $p(t) = p_o + \delta_p(t)$ and $R(t) = R_o + \delta_R(t)$, with $R_o = T_o + \frac{q_o}{c_o}$, where $q_o, W_o, c_o, N_o, p_o, R_o$ are the nominal values determined from equilibrium conditions. Then, a transfer function, $G_{pq}(s)$, from the control input δ_p to the output to be regulated δ_q can be obtained using small signal analysis, see e.g. [6, 7, 14],

$$G_{pq}(s) = \frac{R_o c_o\, K}{(R_o s + \frac{1}{K})}\,\frac{e^{-R_o s}}{(R_o s+1)},\; K = \frac{R_o c_o}{2N_o}. \quad (10)$$

If we assume that $K \gg 1$, then the "plant" (10) is approximately in the form (2) with $K_o := c_o K$, $h = R_o$ and $g(s) = (1+R_o s)^{-1}$. Hence, we can apply the technique proposed in Section 2.

Note that for (10) we can design a PI controller and then the PI controlled system can be considered as an open loop plant with an integrator so that we can apply PD control. This will be an alternative method of obtaining a PID controller. For the plant (10) the transfer function of the PI controlled system can be written as

$$G_{pq}(s)K_{pi}(s) = \frac{KR_o}{16\theta^2}\frac{(8\theta s+1)e^{-R_o s}}{s(KR_o s+1)(R_o s+1)} =: \frac{KR_o}{16\theta^2}\,\frac{g_o(s)}{s}$$

For designing a PD controller, first define $\rho_\nu(\omega) := |g_\nu(j\omega)|$ and $\phi_\nu(\omega) := \angle g_\nu(j\omega)$ as the new magnitude and phase functions. Then set $\eta_\nu(\omega) := |\frac{\rho_\nu(\omega)-cos(\phi_\nu(\omega))}{\omega}|$. Assume that $\eta_\nu(\omega)$ has a maximum, $\eta_{\nu,o}$, which is attained at a single frequency $\omega_{\nu,o}$. With this assumption define $\widehat{K}_{d\nu} := -\frac{\sin(\phi_\nu(\omega_{\nu,o}))}{\omega_{\nu,o}\;\rho_\nu(\omega_{\nu,o})}$. Then the PD controller is

$$K_{\nu,pd}(s) = \frac{1}{2\eta_{\nu,o}}\,\frac{16\theta^2}{KR_o}(1+\widehat{K}_{d\nu}s).$$

Thus the overall PID controller for the plant (10) is the product of $K_{pi}(s)$ and $K_{pd}(s)$, that is,

$$K_{pid}(s) = \frac{(1+\widehat{K}_{d\nu}s)}{2\eta_{\nu,o}c_o R_o K^2}\,\frac{(1+8\theta s)}{s}. \qquad (11)$$

3.2 Simulation results

The simulation topology is a dumbbell network with a single bottleneck link shared by N TCP connections which generate FTP flows as in [18]. This bottleneck link is of capacity $C_0 = 10$ Mbps with $T_0 = 20$ ms delay. The links between the TCP sources and the first router, and the links between the TCP sinks and the second router are $C_1 = C_2 = 10$ Mbps links with $T_1 = T_2 = 40$ ms propagation delay, see Figure 1. The buffer sizes of both routers are set to 300 packets and each packet is of size 1000 Bytes.

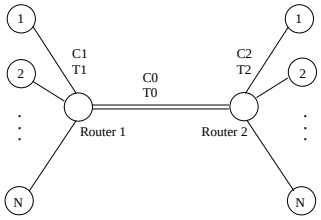

Figure 1: Network Topology

The nominal system parameters are: $N_o = 30$ TCP flows, $c = C_0 = 1250$ packets/s, $q_o = 150$ packets, $R_o = 0.32$ s. The queue length plots of the controllers are given in Figure 2.

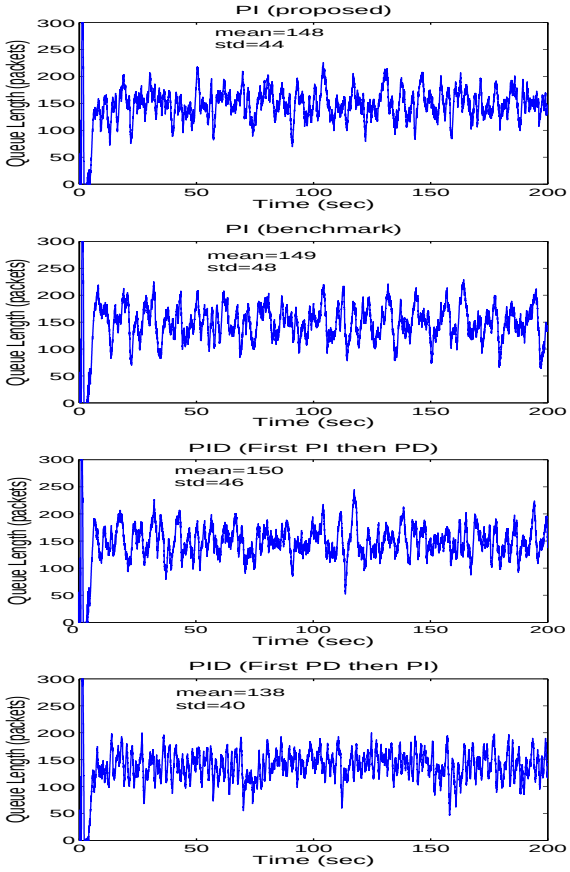

Figure 2: ns-2 simulations: queue length vs time.

In order to evaluate these simulation results we first look at the RMS value of the percentage error of the queue length with respect to the desired queue length, $q_d = 150$ packets, more precisely,

$$\text{RMS error} = \left(\frac{1}{M} \sum_{k=1}^{M} \left(\frac{q(k) - q_d}{q_d} \right)^2 \right)^{\frac{1}{2}}, \quad (12)$$

here M is the total number of the samples generated by ns-2. Secondly, we specify a tolerance for tracking of the desired queue: let $\mathsf{Q_d} := [q_d - q_{\mathsf{tol}}, q_d + q_{\mathsf{tol}}]$ be a neighborhood of the desired queue. In our case we have $q_d = 150$ packets, and we choose $q_{\mathsf{tol}} = 30$ packets. Define T_0 to be the total length of the time intervals for which $q(t) \notin \mathsf{Q_d}$, and let T_{total} be the length of the total simulation time (in our setting $T_{\mathsf{total}} = 200$ seconds). Then the second error metric we are interested in is $\mathcal{E} = T_0/T_{\mathsf{total}}$. This error provides a better look on the variations of the queue length, and its small values bound the delay jitter.

Table 1 shows that the proposed PI controller performs better than the benchmark PI controller for both error metrics defined above. Note that all cost functions are normalized with respect to the time interval. The new proposed PI design gives an improvement of 3% to 9% depending on the cost taken; when integrated over time, the benefits of the new design can be significant.

Table 1: The analysis of simulation results

Controller	RMS error	$\mathcal{E}$
PI (proposed)	0.29	**0.28**
PI (benchmark)	*0.32*	*0.37*
PID (first PI then PD)	0.30	0.30
PID (first PD then PI)	**0.27**	0.29

Furthermore, we investigate the robustness of the proposed controller schemes under variations of number of TCP connections (load) N, the round trip time RTT and, the link capacity c. The controller designs are for the nominal values of the system parameters, as in the previous section, and are fixed throughout the robustness tests. The results of these tests are given in the Appendix. In Figures 3, 4 we vary the number of TCP connections from 10 to 60. In Figures 5, 6 the round trip time is varied between 160 msec and 480 msec. In Figures 7, 8 the capacity is changed from 625 packets/second to 1875 packets/second (from 5 Mbps to 15 Mbps). These figures illustrate the mean and standard deviations of the queue lengths and the error metric comparisons. We omitted the results for the second PID controller (first PD then PI), because it yields similar results to that of the first PID controller (first PI then PD).

As seen in all the figures related to performance robustness analysis, the proposed PI controller performs better than the PI benchmark design except for small values of N. The PI controller scheme that we propose here gives better results than PI benchmark for standard deviation and for both of the error metrics. Simulation results showed that in certain situations the proposed PI controller is better than the PID controller as far as the robustness is concerned.

4 Conclusions

In this paper we proposed new PI, PD and PID controller tuning methods for integrating systems with time delay using the analysis of allowable PID gains done in [5, 12]. The PI and PD controller expressions (5), (6) appearing in Section 2 are valid for all integrating systems whose transfer function is in the form (2) where $g(s)$ is an arbitrary stable transfer function containing a time delay and satisfying $g(0) = 0$.

We have illustrated this method on the AQM problem for a single bottleneck network of TCP flows.

Performance of the proposed designs are compared with the benchmark design of [6, 7] implemented in ns-2. Simulations show that steady state queue variations are lower in the new design, and the transient response performance is better. Robustness of the designs with respect to variation in R, N and c are also analyzed and we have seen that in almost all cases the new PI controller performs better than the benchmark PI controller.

References:

[1] K. J. Aström, T. Hagglund, *PID Controllers: Theory, Design, and Tuning* 2nd Ed., Instrument Society of America, Research Triangle Park, NC, 1995.

[2] S. Athuraliya, S. H. Low, V. H. Li, Q. Yin, "REM: Active Queue Management," *IEEE Network*, vol. 15 (2001), no. 3, pp. 48-53.

[3] S. Floyd, V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1 (1993), no. 4, pp. 397-413.

[4] S. Floyd, K. Fall, "Promoting the use of end-to-end congestion control in the Internet," *IEEE/ACM Transactions on Networking*, vol. 7 (1999), pp. 458–472.

[5] A. N. Gündeş, H. Özbay, A. B. Özgüler, "PID Controller Synthesis for a Class of Unstable MIMO plants with I/O Delays," *Automatica*, vol. 43 (2007), pp. 135–142.

[6] C. Hollot, V. Misra, D. Towsley, W. Gong, "On designing improved controllers for AQM routers supporting TCP flows," in *IEEE INFOCOM*, Alaska, April 2001, pp. 1726-1734.

[7] C. Hollot, V. Misra, D. Towsley, W. Gong, "Analysis and design of controllers for AQM routers supporting TCP flows," *IEEE Transactions on Automatic Control*, vol. 47 (2002), no. 6, pp. 945-959.

[8] Y. Lee, J. Lee, S. Park, "PID controller tuning for integrating and unstable processes with time delay," *Chemical Engineering Science* , vol. 55, 2000, pp. 3481-3493.

[9] S. Low, F. Paganini, J. Wang, S. Adhalka, J. Doyle, "Dynamics of TCP/RED and a scalable control," in *Proc. IEEE INFOCOM*, New York, June 2002, pp 239-248.

[10] V. Misra, W. Gong, D. Towsley, "A fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED," in *Proc. ACM SIGCOMM*, Stockholm, Sweeden, Sep. 2000, pp. 151-160.

[11] ns-2 Network Simulator, version: 2.27. [Online]. Available: http://www.isi.edu/nsnam/ns/.

[12] H. Özbay, A. N. Gündeş, "Resilient PI and PD Controllers for a Class of Unstable MIMO plants with I/O Delays," Proc. of 6th IFAC Workshop on Time Delay Systems, L'Aquila, Italy, July 2006.

[13] E-C. Park, H. Lim, K-J. Park, C-H. Choi, "Analysis and design of the virtual rate control algorithm for stabilizing queues in TCP networks," *Computer Networks*, vol. 44 (2004), pp. 17-41.

[14] P-F. Quet, H. Özbay, "On the Design of AQM Supporting TCP Flows Using Robust Control Theory ," *IEEE Transactions on Automatic Control*, vol. 49 (2004), no. 6, pp. 1031-1036.

[15] E. Poulin, A. Pomerlau, "PI Settings for Integrating Processes Based on Ultimate Cycle Information" *IEEE Transactions on Control Systems Technology*, vol. 7, 1999, pp. 509-511.

[16] S. Ryu, C. Rump, C. Qiao, "A predictive and robust active queue management for Internet congestion control,"in *Proc. of the 8th IEEE Symposium on Computers and Communications*, Kemer, Antalya, Turkey, June-July 2003, pp. 991–998.

[17] G. J. Silva, A. Datta, S. P. Bhattacharyya, "PID Controllers for Time-Delay Systems," 2nd Ed., Birkhäuser, Boston, 2005.

[18] P. Yan, Y. Gao, H. Özbay, "A Variable Structure Control Approach to Active Queue Management for TCP with ECN," *IEEE Transactions on Control Systems Technology*, vol. 13 (2005), no. 2, pp. 203-215.

[19] P. Yan and H. Özbay, "Robust Controller Design for AQM and H_∞ Performance Analysis," in *Advances in Communication Control Networks*, S. Tarbouriech, C. Abdallah, J. Chiasson Eds., Springer-Verlag LNCIS, Vol. 308, 2004, pp. 49-64.

Acknowledgements: This work is supported by the European Commission (contract no. MIRG-CT-2004-006666) and by TUBITAK (grant no. EEEAG-105E156)

5 Appendix: Simulation Results

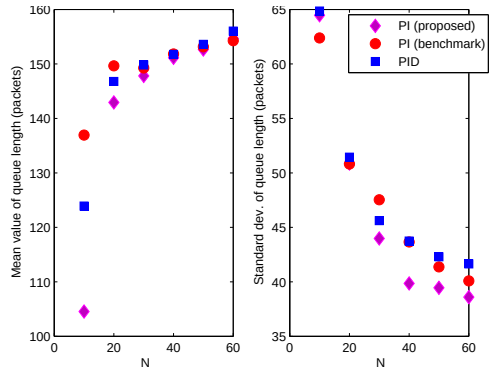

Figure 3: Mean and standard deviation of queue length under load variations

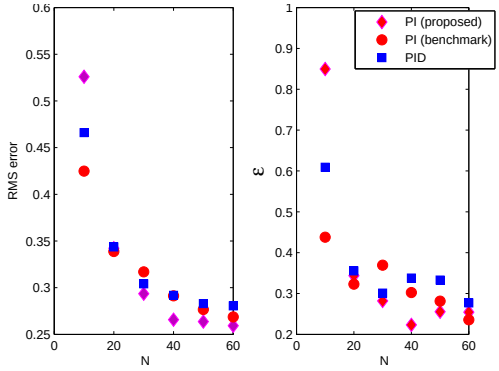

Figure 4: Errors under load variations

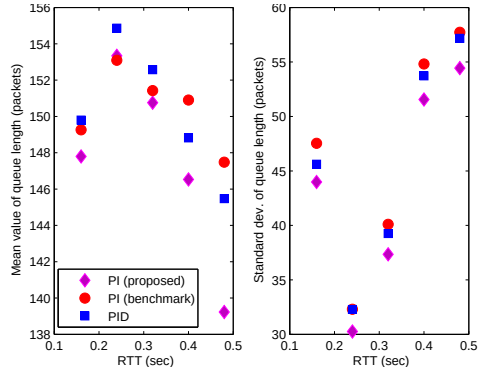

Figure 5: Mean and standard deviation of queue length under RTT variations

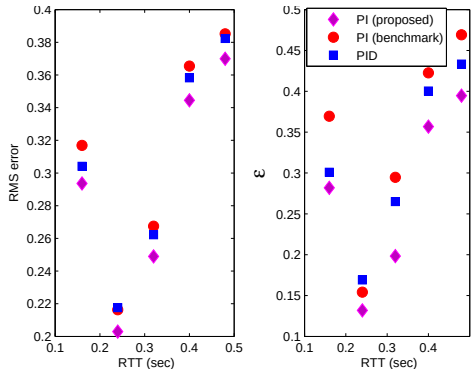

Figure 6: Errors under RTT variations

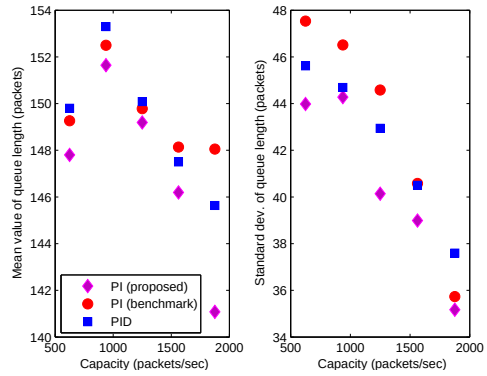

Figure 7: Mean and standard deviation of queue length under capacity variations

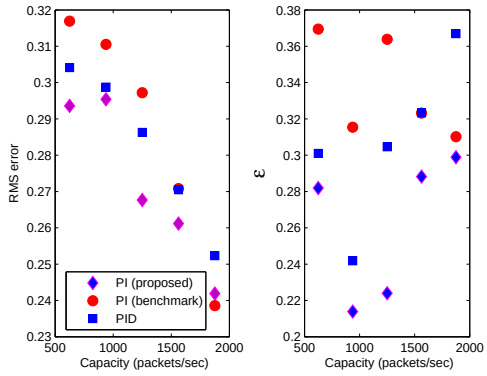

Figure 8: Errors under capacity variations