AQM Control of TCP/IP Networks using Generalized Predictive Control

T. Alvarez, S. Cristea

Department of Systems Engineering and Automatic Control,
School of Computer Engineering, University of Valladolid,
Valladolid, Spain
(e-mail:{ tere, smaranda}@autom.uva.es)

Abstract: This paper presents how generalized predictive control can improve the performance of TCP/IP networks when dealing with control congestion. Predictive control and GPC (Generalized Predictive Control), in particular, can be seen as an improved AQM (Active Queue Management) method. Predictive controllers, constrained and unconstrained, are compared with other control methods, such as PI control or RED/AQM, showing the advantages of the proposed technique, as it makes the consideration of constraints possible in the manipulated and controlled variables.

Keywords: predictive control, AQM, congestion control, TCP, PI

1. INTRODUCTION

As the growth of Internet increases and users demand new applications and better performance, the rise in data volume generates problems such as long delays in delivery, lost and dropped packets, oscillations and synchronization problems (Azuma et al, 2006; Xiong *et al.*, 2005; Deng *et al.*, 2003). These troubles are due to congestion, which happens when there are too many sources sending too much data too fast for the network to handle, and it is a very serious drawback. Thus, it is necessary to reduce as much as possible this problem. At present, there are methodologies to deal with this issue (Jacobson, 1988; Ryu *et al.*, 2004): *congestion control* which is used after the network is overloaded and *congestion avoidance* which takes action before the problem appears. This paper deals with congestion control because it is where feedback control techniques can be openly and easily applied.

Internet congestion control is carried out in the transport layer at the sources (end systems) and has two parts: the end-to-end protocol TCP (Transmission Control Protocol), and the active queue management (AQM) scheme, implemented in routers. As explained in Hollot *et al.* (2002), Hayes *et al.* (2007) and Sun *et al.* (2007), among others, the most common AQM objectives are: *efficient queue utilization* (to minimize the occurrences of queue overflow and underflow, thus reducing packet loss and maximizing link utilization), *queueing delay* (to minimize the time required for a data packet to be serviced by the routing queue) and *robustness* (to maintain closed-loop performance in spite of changing conditions).

These AQM schemes enhance the performance of TCP. During recent years, this has been a subject of research, and different algorithms have been proposed: RED (Floyd and Jacobson, 1993), PI (Hollot *et al.*, 2002) or REM (Athuraliya *et al.*, 2001). These AQM schemes have advantages and disadvantages as they do not work perfectly under every traffic situation. For instance, RED can detect and respond to long-term traffic patterns, but it cannot detect congestion caused by short-term traffic load changes. From the moment that researches published mathematical models of AQM, control theory based approaches have been used to analyze and design AQM (Hollot *et al.*, 2002; Kelly, 2001; Misra *et al.*, 2000; Pagano and Secchi, 2004; Srikant, 2004) schemes (Deng *et al.*, 2003; Durresi *et al.*, 2006; Hayes *et al.*, 2007; Hollot *et al.*, 2002; Marami *et al.*, 2007; Quet and Özbay, 2004; Ryu *et al.*, 2004 and the references therein; Xiong *et al.*, 2005).

Model Based Predictive Control (MBPC) (Camacho and Bordons, 2007) has been successfully applied to several real systems. It is one of the few control techniques that can handle constraints, systems with time-delay and time varying simultaneously parameters. The only drawback is that it can sometimes be computationally expensive. The most relevant MBPC approaches to the TCP congestion control problem found in the literature are now presented. Their solutions are innovative and give good results. Chiera and White (2007) worked with a model-free LQC subspace predictive control. Results are very good, but no constraints are considered. Zhu *et al.* (2004) applied a GPC (Clarke and Mohtadi, 1987) to solve the problem and compared the results with a P and PI controller. Again, no constraints are taken into account. The same can be said for Marami *et al.* (2007) and Marami and Haeri (2007). The predictive controllers are very smart and non-computationally taxing. Rahnamai *et al.* (2006) considered a neural predictive controller and presented a comparison with a PI.

In this paper we apply a constrained GPC and a non-constrained GPC to the TCP congestion control problem and compare the results with PI and RED controllers. From the results it can be concluded that constraints on inputs allow a smoother response and constraints on outputs limit the queue size to a desired value. Both predictive controllers give better performance than standard procedures.

This paper is organized as follows. Section 2 introduces the TCP/AQM dynamic model. The AQM control problem RED and PI are discussed in section 3. Section 4 presents the proposal of using predictive control as an AQM control strategy. Section 5 shows simulation results and comparison between the different control techniques applied to the system. Finally, in section 6 some conclusions and future work are given.

2. DYNAMIC MODEL OF AN AQM ROUTER

This section presents the modelling of an AQM router. First, a non-linear model is given and then the linear version is derived.

2.1 Non-linear model

In Hollot *et al.* (2002), a dynamic model of TCP behaviour was presented using fluid-flow and stochastic differential equation analysis. The model was developed using fluid-flow and stochastic differential equation analysis. For simplicity, this paper considers a reduced version that ignores the TCP timeout mechanism. The model relates the average value of the network variables and is described by the following coupled, nonlinear differential equations:

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

$$\dot{q}(t) = \begin{cases} -C + \dfrac{N(t)}{R(t)}W(t), & q > 0 \\ \max\left\{0,\ -C + \dfrac{N(t)}{R(t)}W(t)\right\}, & q = 0 \end{cases} \qquad (1)$$

Where

$W \approx$ average TCP window size (packets),
$\dot{q} \approx$ average queue length (packets),

$R \approx$ round-trip time = $\dfrac{q}{C} + T_p$ (secs),

$C \approx$ link capacity (packets/sec),
$T_p \approx$ propagation delay (secs),
$N \approx$ load factor (number of TCP sessions),
$p \approx$ probability of packet mark.

As explained by Hollot *et al.* (2002), the first differential equation in (1) describes the TCP window control dynamic and the second equation models the bottleneck queue length as an accumulated difference between packet arrival rate and link capacity. The queue length and window size are positive, bounded quantities, i.e., $q \in [0, \bar{q}]$ and $W \in [0, \bar{W}]$, where $\bar{q}$ and $\bar{W}$ denote buffer capacity and maximum window size, respectively. In this formulation, the congestion window size $W(t)$ is increased by one every round-trip time if no congestion is detected, and is halved when congestion is detected. Moreover it has been assumed that the AQM scheme implemented at the router marks packets using Explicit Congestion Notification (ECN, Ramakrishnan and Floyd, 1999) to inform the TCP sources of impending congestion.

2.2 Linear model

Although an AQM router is a non-linear system, in order to analyze certain types of properties and design controllers we need a linear model which is presented in this sub-section.

To linearize (1), we assume that the number of active TCP sessions and the link capacity are constant, i.e., $N(t)=N$ and $C(t)=C$. The dependence of the time delay argument $t-R$ on queue length q, is ignored and it is assumed to be fixed to $t-R_0$. Then, local linearization of (1) about the operating point results in the following equation:

$$\begin{aligned} \partial\dot{W}(t) &= -\frac{N}{R_0^2 C}\left(\partial W(t) - \partial W(t-R_0)\right) \\ &\quad - \frac{1}{R_0^2 C}\left(\partial q(t) - \partial q(t-R_0)\right) - \frac{R_0^2 C}{2N^2}\partial p(t-R_0) \\ \partial\dot{q}(t) &= \frac{N}{R_0}\partial W(t) - \frac{1}{R_0}\partial q(t) \end{aligned} \qquad (2)$$

Where $\partial\dot{W}(t) = W - W_0$, $\partial q = q - q_0$ and $\partial p = p - p_0$, represent the perturbed variables. The operating point for a desired equilibrium queue length q_0 is given by:

$$R_0 = \frac{q_0}{C} + T_p,\ W_0 = \frac{R_0 C}{N} \text{ and } p_0 = \frac{2}{W_0^2} \qquad (3)$$

Equation (2) can be further simplified by separating the low frequency ('nominal') behaviour (*P(s)* in (4)) of the window dynamic from the high frequency behaviour (Δ(s) in (4)) which is accounted as a parasitic.

$$P(s) = \frac{C^2/(2N)}{\left(s + (2N)/(R_0^2 C)\right)\left(s + 1/R_0\right)},\ \Delta(s) = \frac{2N^2}{R_0 C^3}\left(1 - e^{-R_0 s}\right) \quad (4)$$

3. THE AQM CONTROL PROBLEM

This section introduces the control formulation of an AQM router and how RED and PI control can be applied.

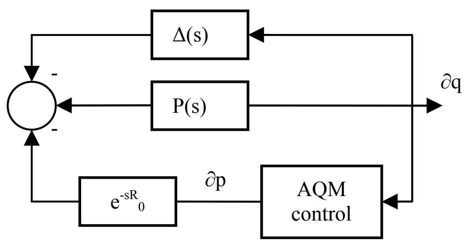

Figure 1: AQM as feedback control

3.1. AQM as feedback control

Taking (4) as starting point, Hollot *et al.* (2002) give a feedback control system of AQM (Figure 1). The blocks can be rearranged as shown in Figure 2.

The action of an AQM control law is to mark packets with probability p, as a function of the measured queue length q. Following (4), the transfer function Δ(s) denotes the high-frequency window dynamics and *P(s)* (plant dynamics) relates how p dynamically affects q.

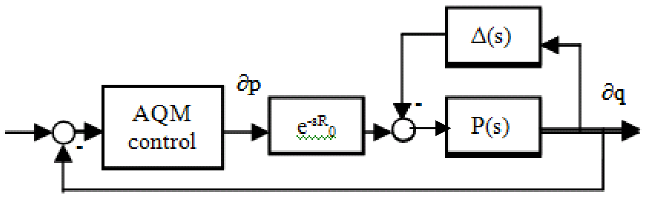

Figure 2: Block diagram of AQM as feedback control system

3.2. AQM using RED

Random Early Detection (known as RED) was presented by Floyd and Jacobson (1993). A RED gateway calculates the average queue size, using a low-pass filter with an exponential weighted moving average. The average queue size is compared to two thresholds (minimum and maximum). When the average queue size is less than the minimum threshold, no packets are marked. When the average queue size is greater than the maximum threshold, every arriving packet is marked. If marked packets are in fact dropped, or if all source nodes are cooperative, this ensures that the average queue size does not significantly exceed the maximum threshold. When the average queue size is between the minimum and the maximum threshold, each arriving packet is marked with probability *p*, where *p* is a function of the measured queue length *q*.
Hollot *et al.* (2002) proposed the following transfer function model for the RED controller:

$$C(s)=\frac{K\cdot L_{red}}{s+K}=\frac{K_{red}}{s/k_{red}+1} \quad (5)$$

Following the guidelines in Deng *et al.* (2003), Durresi *et al.* (2006), Hollot *et al.* (2002) and Pagano and Secchi (2004), the parameters can be evaluated as follows:

$$K_{red}=\frac{R_0^3C^2}{(2N)^2}L_{red}\,, \qquad L_{red}=\frac{p_{\max}}{\max_{th}-\min_{th}} \quad \text{and}$$

$$k_{red}=-C\ln(1-\alpha_{red}) \quad (6)$$

Where α_{red} is RED's queue-averaging weight. The corresponding block diagram is shown in Figure 3,

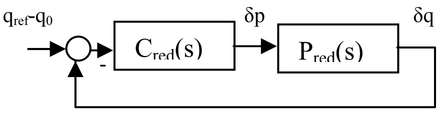

Figure 3: RED linearized block diagram

where: $P_{red}(s)=\frac{-C^2/(2N)}{\left(s+(2N)/(R_0^2C)\right)\left(s+1/R_0\right)}e^{-R_0s}$ (7).

3.3. AQM using Proportional-Integral Control

The transfer function of a PI controller can be written as:

$$C(s)=K_{PI}\frac{s/z+1}{s} \quad (8)$$

This controller is very well known by the control community. Its parameters can be tuned following methods proposed in the control literature. For example, Hollot *et al.* (2002) gave guidelines based on the Bode diagram tuning technique:

$$z=\frac{2N}{R_0^2C}\,,\ K_{PI}=w_g z\left|\frac{jw_g+\frac{1}{R_0}}{C^2/(2N)}\right| \ \textit{and}\ w_g=\frac{\beta}{R_0} \quad (9)$$

where w_g is the unity-gain crossover frequency. The block diagram is the same as in the RED case.

4. AQM USING GENERALIZED PREDICTIVE CONTROL

Model Based Predictive Control (MBPC) (Camacho and Bordons, 2007; Maciejowski, 2001 and Rossiter, 2003) is a control strategy based on the explicit use of a model to predict the process output over a long-range time period. A receding control horizon technique is used: only the first control signal is applied (so all the changes that take place between two control signal calculations are considered). A cost function is minimized at each sampling time. It is usually defined as a linear combination of the tracking error and the manipulated variables (the most common formulation being a quadratic cost function).

Although a non-linear model can be used for prediction, computational and robustness considerations makes the use of a linear model more adequate. Following the Generalized Predictive Controller (GPC) described in Clarke and Mohtadi (1987), a MIMO controller is described here. Considering a process with M inputs, N outputs and R measurable disturbances represented by the model:

$$A_i\left(q^{-1}\right)y_i(t)=\sum_{j=1}^{N}B_{ij}\left(q^{-1}\right)u_j(t)+\sum_{j=1}^{R}D_{ij}\left(q^{-1}\right)v_j(t)+\frac{T_i}{\Delta}\xi_i(t) \quad (10)$$

where: i =1, ..., M and A_i, B_{ij}, D_{ij} and T_i are polynomials in the q^{-1} operator, $\Delta=1-q^{-1}$ and ξ_i is white noise. The predicted future values of the controlled variables are:

$$\hat{y}_i(t+j)=\sum_{j=1}^{M}\sum_{k=1}^{j}\sigma_j g_{ij_k}\Delta u_j(t-k+j)+p_i(t+j), i=1,\ldots,N \quad (11)$$

where g_{ij} is the step response between y_i and u_j and p_i is the free response of y_i. The coefficients σ and η take the value 1 if the corresponding input is included in the predicted output and 0 if it is not.

The control algorithm objective is the calculation of the sequence of (optimal) changes of the control variables in a control horizon Nu: $\Delta u_i(t+j)$, j=0,..., Nu_i-1 so that the predicted outputs $\hat{y}_i$ are as close as possible to the internal reference $r_i(t+j)$. This is translated into an optimization problem where a quadratic cost function of the tracking error and the manipulated variables is minimized taking into account constraints on **Δu**, **u**, **y**, and any other constrained variable that depends on **Δu**. This optimization problem can be stated as:

$$J=\sum_{i=1}^{N}\sum_{j=N1i}^{N2i}\eta_i\left(\gamma_i\left(\hat{y}_i(t+j)-r_i(t+j)\right)^2\right)$$
$$+,\sum_{i=1}^{M}\sum_{j=0}^{Nu_i-1}\sigma\left(\beta\Delta u_i(t+j)\right)^2 \quad (12)$$

$$D_{m_i}\le\Delta u_i(t+j)\le D_{M_i},\, j=0,\ldots,N_{u_i}-1$$
$$U_{m_i}\le u_i(t+j)=u_i(t-1)+\sum_{i=0}^{j}\Delta u_i(t+j)\le U_{M_i}$$
$$L_{m_i}\le\hat{y}_i(t+j)\le L_{M_i},\, j=N3_i,\ldots,N4_i \quad (13)$$

where the coefficients γ and β give the relative weight of every prediction error or change in the control variables, while the coefficients σ and η taking the value 1 or 0 allow a variable to be included in the index or excluded. When no constraints are considered, there exists an explicit solution of (12).

4.1. Implementation in TCP

In the AQM scheme in Figure 2, there is one input (*p*), one output (*q*) and no measured disturbances. Thus, the transfer function that will be used as a model to predict the future outputs is given by the low frequency component of the model in (4):

$$P_{GPC}(s)=\frac{-C^2/(2N)}{\left(s+(2N)/(R_0^2C)\right)\left(s+1/R_0\right)} \quad (14)$$

The system delay will be taken into account in the control and prediction horizons. Choosing the sampling time T_s as $0.2\sqrt{\tau_1^2+\tau_2^2}$ where τ_1 and τ_2 are the time constant of $\frac{C^2/(2N)}{s+2N/(R_0^2C)}$ and $\frac{1}{s+1/R_0}$ and applying the zero-order-hold transformation, the discrete transfer function of (14) is represented by:

$$P_{GPC}(z^{-1})=z^{-d}\,\frac{b_1z^{-1}+b_2z^{-2}}{1+a_1z^{-1}+a_2z^{-2}} \quad (15)$$

where the coefficients a_i and b_i are directly calculated from the discretization, and *d* represents the system delay in time samples. In our case, simplifying, if R_0 is the continuous time-delay, then d=round_upper(R_0/T_s).

5. SIMULATION

In this section simulation results under different working conditions will be presented. The non-linear simulations have been carried out using EcosimPro (2007) considering (1) as the model. The GPC controller, implemented in C, is called as an external library.

The working scenario is defined by the following values (Marami *et al.*, 2007; Quet and Özbay, 2004):

i. Nominal values known to the controller: N=50 TCP sessions, C = 300 packets/s, Tp=0.2 s. Thus R_0=0.533 and W_0= 3.2 packets. As a fluid model is assumed, no packetization issues are considered.
ii. Plant real values: N=40 TCP sessions, C = 250 packets/sec., Tp=0.3 sec., so R_0=0.7 and W_0= 4.375 packets.

For comparison, the PI AQM scheme proposed in Hollot *et al.* (2002), the RED AQM (Floyd and Jacobson, 1993), an unconstrained GPC that minimizes (12) and a constrained GPC that minimizes (12) subject to (13) were implemented and simulated under different situations.

The parameters of the PI are computed as suggested by Hollot *et al.* (2002) and then is applied fine tuning to improve the system's response. Thus, the PI transfer function is given by:

$$C_{PI}(s)=-0.0013\left(\frac{1}{1.17}+\frac{1}{s}\right) \quad (16)$$

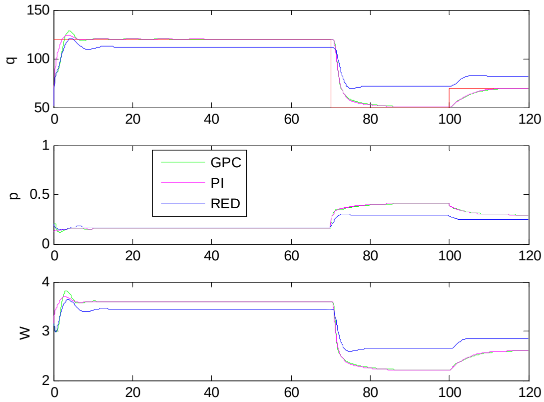

Figure 4: Experiment 1. Nominal parameters

RED has the following parameters: p_{max}=0.1, min_{th}=80, max_{th}=150, L_{red}= 0.0014 and queue averaging weight α=0.001 (As explained in Durresi *et al.* (2006), decreasing this value gives a more stable but slower response). The transfer function is:

$$C_{red}(s)=\frac{-0.0039}{1+s/0.3} \quad (17)$$

The GPC controller has the following settings:

- Sampling time: T_s=0.2013 s.
- Model: $A(z^{-1})=T(z^{-1})=1-1.476z^{-1}+0.5416z^{-2}$, $B(z^{-1})=z^{-3}\left(-14.9-12.15z^{-1}\right)$
- Prediction horizon: N1=4, N2=40
- Control horizon: Nu=4 (it should be at least equal to the plant delay).
- γ=100, β=500.
- The following constraints are included in the constrained GPC:
    - $0\le p\le 1$, $-0.05\le\Delta p\le 0.05$
    - $0\le q\le 200$ (these values are changed in experiments 3 and 4).

Some illustrative experiments are now described.

5.1. Experiment 1: Nominal parameters

In this simulation, the performance of PI, RED, unconstrained GPC and constrained GPC are compared on the non-linear plant defined by (1) with nominal parameters. The desired queue length is q_{ref} =120 pac. at t=0 sec., q_{ref} =70 at t=50 and finally, q_{ref} =100 at t=100. In this case no performance differences can be observed between the unconstrained and the constrained GPC as the constraints are not reached and the variables remain within limits. The PI and GPC controllers give a good performance during all the simulation, but the RED one cannot reach the reference value as there is a steady state error inherent to the technique.

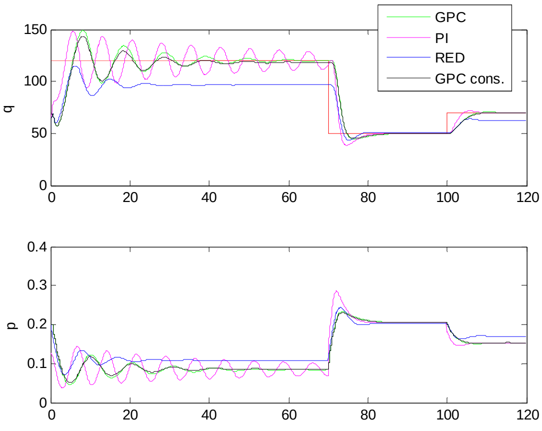

Figure 5: Experiment 2: Variation of parameters

5.2. Experiment 2: Variation of parameters

Now, the previous experiment is repeated but considering the real values of the plant (different from the nominal values that were used for controller tuning). The RED controller, again, presents steady state error. The PI controller presents significant oscillations, specifically on the control signal which is absolutely inadequate until t=70 secs. The constrained GPC gives a response smoother than the unconstrained GPC, due to the constraints on the rate of change of *p* (Figure 5).

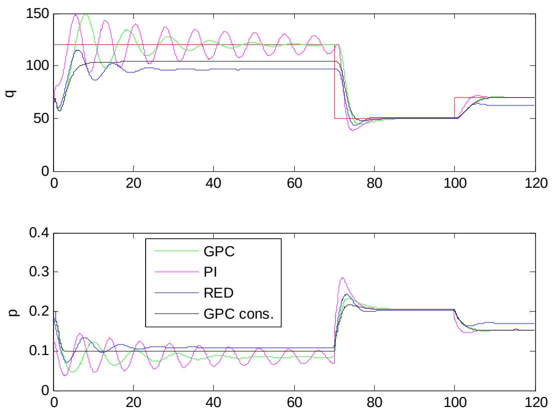

Figure 6: Experiment 3. Lower limit on p: p>0.1

5.3. Experiment 3. Constraints on *p*

One of the advantages of considering constraints is that the desired final value of inputs and outputs can be limited. For instance, if the dropping probability *p* must be smaller than 0.1. If we repeat the experiment (Figure 6) we can see that the probability is always maintained at values greater than or equal to 0.15. The queue length cannot reach the reference value when q_{ref} =120.

5.4. Experiment 4: Random TCP flows, propagation delay and link capacity

In this simulation, we compare the performance of the PI, RED, unconstrained and constrained GPC on a non-linear plant defined by (1) with real values, but when the number of TCP flows N is a normally distributed random signal (with mean 45.0 and variance 49.0) added to a pulse of period 50 secs., duration 30 and amplitude 8.

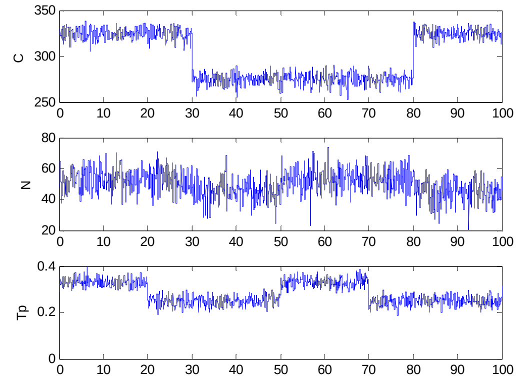

Figure 7: Random variables in experiment 4

The propagation delay T_p is a normally distributed random signal (with mean 0.25 and variance 0.0004) added to a pulse of period 50 secs., duration 20 and amplitude 0.08 . The outgoing link capacity C is a normally distributed random signal with mean 275 and variance 36 added to a pulse of period 80, duration 30 and amplitude 50 (Figure 7). The desired queue length is q_{ref} =150.

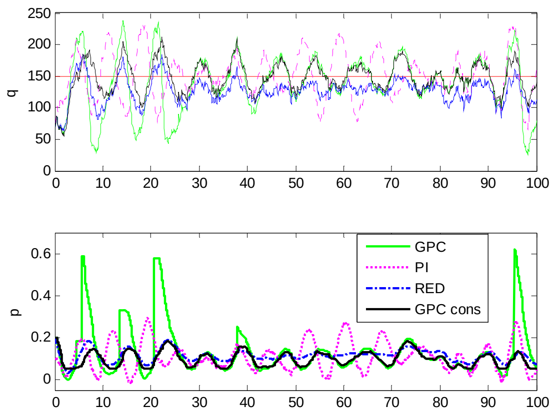

Figure 8: Experiment 4. Random disturbances

Figure 8 shows the input and output variables in the presence of these disturbances, using different controllers. Due to the random nature of *N*, *C* and T_p, the manipulated variable (*p*) can change very abruptly, which is non adequate. Thus, the constrained GPC considers the following constraints: $-0.02 \le \Delta p \le 0.02$, $0.05 \le p \le 0.25$ and $0 \le q \le 200$.

Table 1 summarizes the mean, variance and standard deviation of each controller during this experiment. Although the PI controller better approaches the reference value, the variance is quite big. The constrained GPC gives a smoother response and remains within limits. Moreover, the PI controller calculates negative values for the dropping probability.

Table 1: q mean, variance and standard deviation

Controller	GPC	Cons. GPC	PI	RED
Mean	138.9	146.8	148.7	126.9
Variance	1940	661.87	1272.0	383.0
Stn. Dev.	43.6	25.7	35.7	19.6

6. CONCLUSIONS

This paper has presented the application of the GPC predictive controller as an active queue management methodology for TCPI/IP networks with promising results. The constrained GPC makes it possible to consider constraints on signals. The main advantage of considering constraints is that we can limit the range of values and even the rate of change of the manipulated variables.

Further work includes a MIMO and time varying network models, so changing conditions can be included. This would help to control, the congestion at several nodes. Moreover as, the number of packets in a queue is an integer value, also hybrid predictive control could also be explored.

REFERENCES

Athuraliya, S., S.H. Low, V.H. Li, Q. Yin (2001). REM: active queue management. *IEEE Networking*, **15**, 48-53.

Azuma, T., T. Fujita, M. Fujita (2006). Congestion control for TCP/AQM networks using State Predictive Control. *Electrical Engineering in Japan*, **156**, 1491-1496.

Camacho, E.F., C. Bordons, C. (2007). *Model Predictive Control.* Springer Verlag, 1995, 2nd ed. 2004., Corr. 2nd printing, 2007.

Chiera, B.A. and L.B. White (2007). Application of model-free LQC subspace predictive control to TCP congestion control. *International Journal of Adaptive Control and Signal Processing*, published on-line 22/08/2007.

Clarke, D.W., Mohtadi, C., and Tuffs, P.S.: 'Generalised predictive control-Part I: The basic algorithm and Part II: Extensions and Interpretations', *Automatica*, 1987, 23, (2), pp. 137-160

Deng, X., S. Yi, G. Kesidis. C.R. Das (2003). A control theoretic approach for designing adaptive AQM schemes. GLOBECOM'03, **5**, 2947 – 2951.

Durresi, A., P. Kandikuppa, M. Sridharan, S. Chellappan, L. Barolli, R. Jain (2006). LED: Load Early Detection: A Congestion Control Algorithm based on Router Traffic Load. *Journal of Information Processing Society of Japan*, **47**, 94 - 107.

EcosimPro. The El Modelling language. www.ecosimpro.com.

Floyd, S., V. Jacobson (1993). Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, **1**, 397-413.

Hayes. M.J., S.M. Mahdi Alavi, P. Van de Ven (2007). An Improved Active Queue Management scheme using a two-degree-of-freedom feedback controller. European Control Conference (ECC 2007), July 2-5, 2007, Kos, Greece.

Hollot, C.V., V. Misra, D. Towsley, W. Gong (2002). Analysis and Design of Controllers for AQM Routers Supporting TCP flows. *IEEE Transactions on Automatic Control*, **47**, 945-959.

Jacobson, V. (1988). Congestion avoidance and control. ACM SIGCOMM'88.

Kelly, F. (2001). Mathematical Modeling of the Internet, in Mathematics Unlimited-2001 and beyond, B. Enquist and W. Schmid, Eds, Berlin, Germany, Springer-Verlag.

Maciejowski, J.M. (2001). *Predictive control with constraints*. Pearson.

Marami, B., N. Bigdeli and M. Haeri (2007). Active Queue Management of TCP/IP Networks Using Rule-Based Predictive Control. IEEE International Symposium on Industrial Electronics (ISIE 2007), June 4-7, 2007, Vigo, Spain.

Marami, B.and M. Haeri (2007). Neural network approximation of model predictive controller for congestion control of TCP/AQM networks. International Conference on Control, Automation and Systems, Seoul, Korea, 2591-2596.

Misra, V., W.B. Gong, D. Towsley (2000). Fluid Based Analysis of a network of AQM routers supporting TCP flows with an application to RED. ACM/SIGCOMM 2000, 151-160.

Pagano, M., R. Secchi (2004). A survey on TCP performance evaluation and modelling. Proc. of Int'l Working Conf. on Performance Modelling and Evaluation of Heterogeneous Networks (HET-NET 2004), Ilkley, UK.

Quet, P.-F., H. Özbay (2004). On the design of AQM supporting TCP flows using robust control theory. IEEE Transactions on Automatic Control, **49**, 1031-1036.

Rahnamai, K., K. Gorman and A. Gray (2006). Model predictive neural control of TCP flow in AQM network. IEEE NAFIPS 2006, 493-498.

Ramakrishnan, K., S. Floyd (1999). Explicit Congestion Notification. *ACM SIGCOMM Computer Communications Review*, **24**, 8-23.

Rossiter, J.A. (2003). *Model-Based Predictive Control: A Practical Approach*. Taylor & Francis, CRC Press.

Ryu, S., C. Rump, C. Qiao (2004). Advances in Active Queue Management (AQM) based TCP congestion control. *Telecommunication Systems*, **25**, 317-351.

Srikant, R. (2004). *The Mathematics of Internet Congestion Control*. Boston, Birkhauser Verlag.

Sun, J., S. Chan, K.-T. Ko, G. Chen, M. Zukerman (2007). Instability effects of two-way traffic in a TCP/AQM system. *Computer Communications*, 30, 2172-2179.

Xiong, N., Y. He, Y. Yang, B. Xiao, X. Jia (2005). On designing a novel PI controller for AQM routers supporting TCP flows. APWEB 2005, 991-1002.

Zhu, R., H. Teng and W. Hu (2004). A predictive controller for AQM router supporting TCP with ECN. In AWCC 2004, C.-H. Chi and K.-Y. Lam (Eds), 131-136.