

Model Predictive Neural Control of TCP Flow in AQM Network

Kourosh Rahnamai, Kevin Gorman

*Electrical Engineering Department
Western New England College
Springfield, MA 01119 USA*

krahnama@wnec.edu

Andrew Gray

*Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109*

gray@jpl.nasa.gov

Abstract - Many research papers have been published on RED (Random Early Detection) and variants of RED [1-12]. Recently many articles have been presented on modeling a Transmission Control Protocol (TCP) flow in an Active Queue Management (AQM) of a bottlenecked network link [1-3]. Classical control theories have also been applied to achieve or improve stability of the network flow [4]. In this paper we present a Neural Network (NN) Model Predictive Control (MPC) of TCP flows. We show the robust adaptive behavior of the MPC optimal controller under modeling errors and system dynamic changes. We also show the superior transient and steady state behavior as well as general stability of MPC as compared to the Classical PI controller.

I. INTRODUCTION

The computer network congestion control problem has been a recurring theme for many years. A feedback-based congestion control scheme is required to ensure trouble-free service for data transmission on a bottlenecked network link. The main objective has always been to use network resources efficiently to prevent the data packet loss, which requires retransmission of the lost information. Most currently in-use networks have a window-based flow control for the Transmission Control Protocol. The size of the window length is used as a feedback mechanism to control the network's congestion problem. As an example, packet losses are used as the feedback signal for reducing the queuing window size in TCP Reno. This congestion control protocol increases the window length gradually until such time that a packet loss occurs at the gateway, indicating that the transmission bandwidth limits have been reached. At this point, if the transmission source keeps increasing its window length more packet losses will occur. Therefore, as soon as a packet loss is detected, the source host will reduce its window size to avoid network congestion.

Another popular congestion control scheme is TCP RED. In this algorithm for controlling queuing window size, RED gateway randomly drops the incoming packets proportional to the average queuing length. The longer the window length the higher would be the probability of dropping received packets. The fundamental idea behind TCP RED is to keep the queuing length to a minimum.

Since TCP RED is a feedback control problem, we can apply classical [4] as well as modern control theories to adjust the size of the queuing window. Since most control system designs are based on a mathematical model of the system we

first present a [1-3] model for a typical TCP flow, based on the fluid dynamics theories.

I. TCP MODELING

The following nonlinear equations describe the behavior of a typical bottlenecked network gateway

$$\frac{dW(t)}{dt} = \frac{1}{R(t)} - \frac{W(t)}{2} \frac{W(t-R(t))}{R(t-R(t))} p(t-R(t)) \quad (1)$$

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

$$R(t) = \frac{q(t)}{C} + T_p \quad (3)$$

where

W : Expected TCP window length (packets)
q : Queue length (packets)
R : Round trip time (sec)
C : Link capacity (packets/sec)
T_p: Propagation delay (sec)
N : Number of TCP sessions
P : Probability of packet marked/drop

and

$$W \in [0, W_{\max}], q \in [0, q_{\max}]$$

Now linearizing [2-3] these equations about an operating point $Q_0 = (W_0, R_0, q_0, p_0)$, we obtain

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

$$\dot{\delta q}(t) = +\frac{N}{R_0} \delta W(t) - \frac{1}{R_0} \delta q(t) \quad (5)$$

where

$$\delta W(t) \doteq W(t) - W_0$$

$$\delta q(t) \doteq q(t) - q_0$$

$$\delta p(t) \doteq p(t) - p_0$$

The total transfer function of the system can be obtained by taking the Laplace transform of these equations

$$H_{queue}(s) = \frac{\delta q(s)}{\delta W(s)} = \frac{\frac{N}{R_0}}{\left[s + \frac{1}{R_0} \right]} \quad (6)$$

$$H_{tcp}(s) = \frac{\delta W(s)}{\delta p(s)} = \frac{-\frac{R_0 C^2}{2N^2} e^{-sR_0}}{\left[s + \frac{2N}{R_0^2 C} \right]} \quad (7)$$

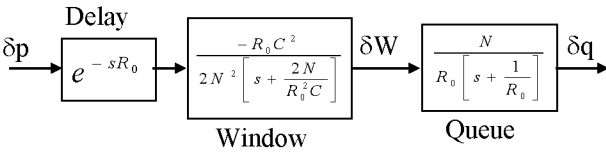


Figure 1: TCP model

Using Pade's approximation to represent the time delay with a rational number we obtain

$$e^{-R_0 s} \cong \frac{1 - \frac{R_0}{2} s}{1 + \frac{R_0}{2} s} \quad (8)$$

$$H_T(s) = H_{tcp}(s)H_{queue}(s) = \frac{1 - \frac{R_0}{2} s}{1 + \frac{R_0}{2} s} \frac{-\frac{R_0 C^2}{2N^2}}{\left[s + \frac{2N}{R_0^2 C} \right]} \frac{\frac{N}{R_0}}{\left[s + \frac{1}{R_0} \right]} \quad (9)$$

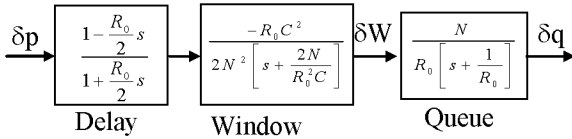


Figure 2: AQM linear model

Figure 2 shows the linear model of the TCP window-control system. The standard RED controller is shown in Figure 3.

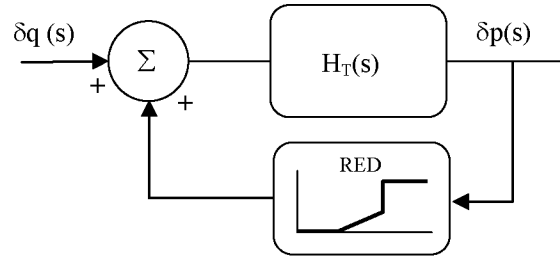


Figure 3: RED feedback Block

The positive feedback is due to an internal 180 degrees phase shift between the input and output of the $H_T(s)$.

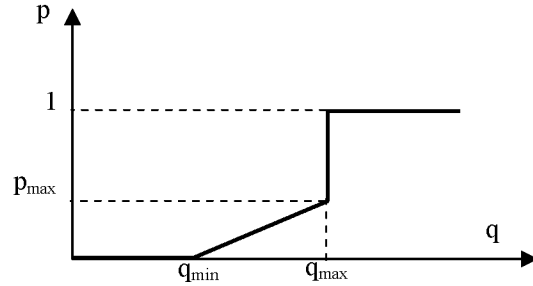


Figure 4: RED Nonlinear feedback function

$$p(q) = \begin{cases} 0 & 0 \leq q \leq q_{\min} \\ \frac{q - q_{\min}}{q_{\max} - q_{\min}} p_{\max} & q_{\min} \leq q \leq q_{\max} \\ 1 & q_{\max} \leq q \end{cases} \quad (10)$$

Equation (10) is the piecewise function describing the standard RED algorithm.

I. NEURAL NETWORK MODEL PREDICTIVE CONTROLLER (NN-PMC)

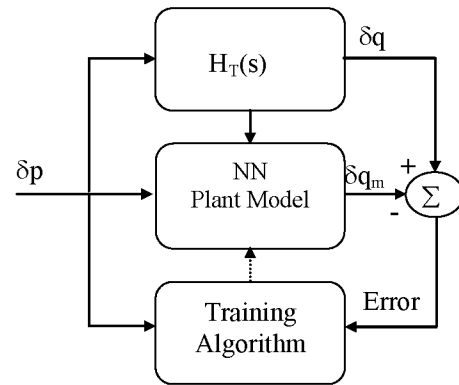


Figure 5: Predictive Model

Plant forward dynamics identification is at the first stage of any model predictive control structure. Figure 5 shows a neural structure for the system model identification [14]. The

neural plant model and the actual system outputs are compared and an error signal is generated which is used to train the neural network. The plant parameters, and the network weights are changes such that the error signal is minimized. The standard form for model identification is given by

$$\delta q(k+d) = h[\delta q(k), \delta q(k-1), \dots, \delta q(k-n+1), \delta p(k), \delta p(k-1), \dots, \delta p(k-m+1)] \quad (11)$$

Where δp is the system input, δq is the system output and d is the system delay. The main objective of the identification process is to find the function h . In this application we train a neural network to approximate the nonlinear function h .

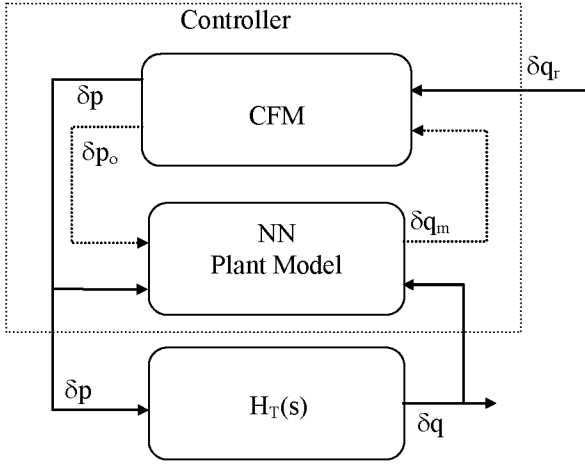


Figure 6 Neural MPC

Figure 6 shows the structure of a typical neural model predictive control. The reference output signal, δq_r which could be the output of a reference model, is the input to the cost function minimization (CFM) block. CFM block produces input signals for the NN and the plant. The output of the plant is fed back into the NN and the output of this block is used to adjust the CFM parameters. CFM block uses a predictive numerical optimization algorithm to generate the input signal for the system such that the following cost function is minimized

$$J = \sum_{j=N_1}^{N_2} [\delta q_r(k+j) - \delta q_m(k+j)]^2 + \rho \sum_{j=1}^{N_u} [\delta p_o(k+j-1) - \delta p_o(k+j-2)]^2 \quad (12)$$

where

N_1 : is the minimum tracking horizon
 N_2 : is the maximum tracking horizon
 N_u : is the control signal horizon

δq_m : is the predicted NN response

δq_r : is the desired reference trajectory

ρ : is the control input weight

It is important to mention that N_1 , N_2 , N_u , and ρ are design parameters. With proper selection of these parameters a balance between the reference tracking trajectory and the control input effort is achieved.

II. SIMULATION RESULTS

Figure 7 shows the simulation block diagram of the model predictive and the PI controllers. The network topology that we have considered consisted of a single bottleneck network with 40 ms round trip delay and channel capacity of 2000 packets/sec. The network traffic consists of 50 identical long-lived TCP sessions with RED parameters set to $q_{\min} = 200$ packets, $q_{\max} = 800$ packets, and $p_{\max} = 0.5$.

Figure 8 shows the simulation results with q_{ref} set to 560 packets and 10 TCP sessions dropping at 5 seconds into the simulation. As can be seen both algorithms perform well under normal operating conditions, but PI has much worse transient response at onset of the dropped session which causes the dynamics of the plant to change.

Figure 9 shows that, as the number of TCP drop out sessions increases, the overshoot transient behavior of the PI controller becomes unacceptable and approaches instability with oscillatory queuing window size changes between 900 to 200 packets in less than a second.

We can see from Figure 10 that the PI controller becomes unstable when 25 TCP sessions drop, but the MPC controller performs gracefully with no overshoot. Note that the queuing window length is inversely proportional to the number of TCP sessions, so as the number of the TCP sessions drops, the queue window length increases.

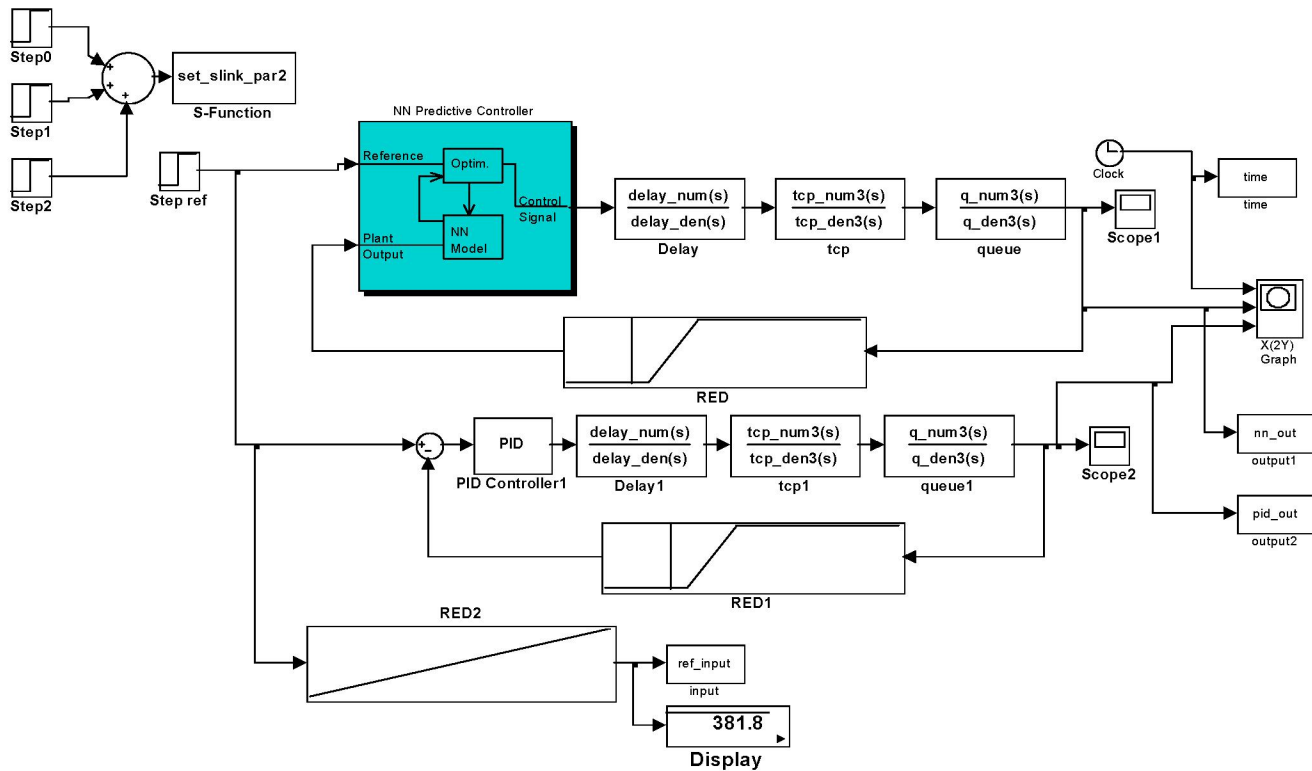


Figure 7: Simulation block diagram of the MPC and the PI

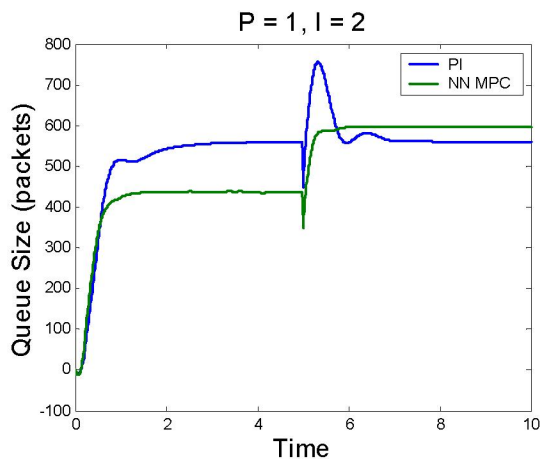


Figure 8: qref = 560 packets, 10 TCP session drop out at 5 sec

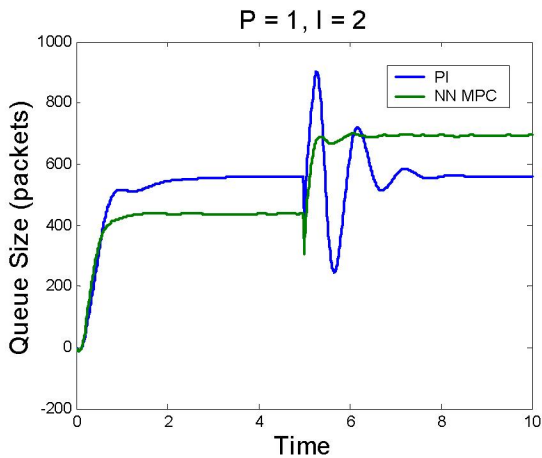


Figure 9: qref = 560 packets, 15 TCP session drop out at 5 sec

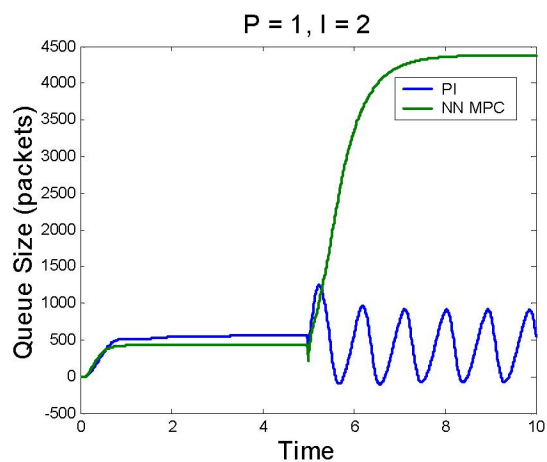


Figure 10: $q_{ref} = 560$ packets, 25 TCP sessions drop out at 5 sec

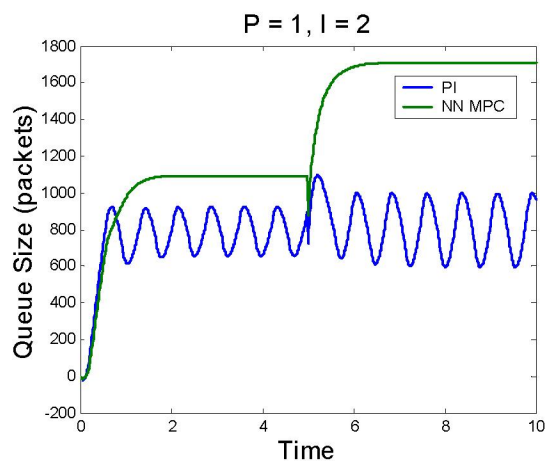


Figure 11: $q_{ref} = 800$ packets, 10 TCP sessions drop out at 5 sec

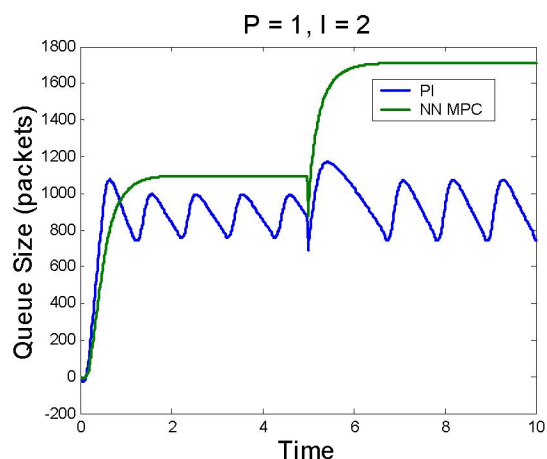


Figure 12: $q_{ref} = 1280$ packets, 10 TCP sessions drop out at 5 sec

Figures 11 and 12 show the oscillatory behavior of the PI controller as q_{ref} is increased.

III. CONCLUSION

In this work, we have shown the superior, transient, steady state behavior and the general stability of MPC over the PI controller as it is applied to optimize the length of a TCP queuing window in a bottlenecked AQM link. The adaptive nature of the MPC controller algorithm exhibits a stable behavior even under severe modeling error. Although PI controller performs very well under normal operating conditions, it produces extreme oscillatory responses under modeling error.

REFERENCES

- [1] Dong Lin and Robert Morris, "Dynamics of random early detection," in Proceedings of ACM/SIGCOMM, 1997.
- [2] Vishal Misra, Wei-Bo Gong, and Don Towsley, "Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED," in Proceedings of ACM/SIGCOMM, 2000.
- [3] Hollot C., Misra V., Towsley D., Gong W., "A Control Theoretic Analysis of RED," Proceedings of IEEE Infocom 2001.
- [4] Hollot C., Misra V., Towsley D., Gong W., "On designing improved controllers for AQM routers supporting TCP flows," Proceedings of IEEE Infocom 2001.
- [5] Floyd S., Jacobson V., "Random Early Detection Gateways for Congestion Avoidance," IEEE/ACM Transactions on Networking, pp.397- 413, August 1993.
- [6] Jacobson V., "Congestion avoidance and control," in Proceedings of SIGCOMM '88, pp. 314-329, August 1988.
- [7] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," IEEE/ACM Transactions on Networking, vol. 1, pp. 397-413, August 1993.
- [8] Martin May, Thomas Bonald, and Jean-Chrysostome Bolot, "Analytic evaluation of RED performance," in Proceedings of IEEE Infocom, March 2000.
- [9] M. Christiansen, K. Jeffay, D. Ott, and F. D. Smith, "Tuning RED for web traffic," in Proceedings of ACM Sigcomm, 2000.
- [10] Fernando Paganini, John C. Doyle, and Steven H. Low, "Scalable laws for stable network congestion control," in Proceedings of Conference on Decision and Control, December 2001.

[11] Kisimoto, M.; Ohsaki, H.; Murata, M., "On transient behavior analysis of random early detection gateway using a control theoretic approach," Proceedings of the 2002 International Conference on Control Applications, Volume: 2, September 18-20, 2002 Glasgow, Scotland, U.K.

[12] Hiroyuki, H.; Ohsaki, H.; Murata, M., "Steady state and transient behavior analyses of tcp connections considering interactions between tcp connections and network," in Proceedings of the 2003 Symposium on Applications and the Internet (SAINT 03), 27-31 Jan. 2003

[13] Jiwei Chen, Fernando Paganini, Ren Wang, M.Y. Sanadidi, Mario Gerla, "Fluid-flow Analysis of TCP Westwood with RED," Department of Electrical Engineering and Computer Science, UCLA CSD Technical Report.

[14] D. Soloway, P. Haley, "Neural Generalized Predictive Control," Proceedings of the 1996 IEEE International Symposium on Intelligent Control, Dearborn, MI, September 15-18, 1996.

[15] Gene F. Franklin, J. David Powell, and Abbas Emami-Naeini, Feedback Control of Dynamic Systems, Addison-Wesley, 2002.