A New TCP End-to-End Congestion Avoidance Algorithm Through Output Feedback

Yi Guo*, **Zhihua Qu*** and **Nageswara Rao**†

* Department of Electrical and Computer Engineering
University of Central Florida
e-mails: yguo@ee.ucf.edu, qu@mail.ucf.edu

† Center for Engineering Science Advanced Research
Oak Ridge National Laboratory
e-mail: raons@ornl.gov

Abstract

We design end-to-end congestion avoidance algorithm for TCP window dynamics. Instead of using explicit feedback from the router like that in RED, we propose the idea of dynamically estimating the queue size from the noisy throughput acknowledgement, and driving TCP window accordingly. A linear quadratic Gaussian controller is constructed, and stability of the composed system is proved. In this setting, complex functions are placed in the hosts and not inside the network.

1 Introduction

TCP is the most widely used end-to-end protocol in the Internet. It employs a bandwidth probing and window control scheme for traffic congestion avoidance and control. Specifically, TCP adjusted its window dynamics using the additive-increase multiplicative-decrease (AIMD) strategy ([4]). There has been a continuing interest from different mathematical and engineering research communities to improve TCP's dynamics and performances in congestion avoidance and control, see [11, 13] and the references therein. In this paper, we study the end-to-end congestion avoidance problem and design TCP-window control law based on noisy throughput feedback.

Among the early papers, the authors in [2] formulate the problem of congestion avoidance and control of computer networks as a system control problem, in which the system senses its state and feeds this back to its users who adjust their controls. Since the recent work of [5], striking progress has been made in analytic modelling of the Internet, and practical congestion control mechanisms as those in TCP and its variants haven been analyzed and examined using mathematically grounded feedback control, see, for example, [1, 3, 7, 8, 10]. In [5, 7, 8], the system problem of optimizing end users' utilities subject to the available resources was transformed into a user (or source) problem and a network (or link) problem. Optimizing algorithms were designed at both the user and the link, and explicit feedback from the link to the user were used to solve the system problem. Other explicit feedback control schemes such as random early detection (RED) were designed in [1, 3] on Active Queue Management (AQM) routers. However, due to the large amount of commercial routers in use in the Internet, it is not realistic to change router algorithms to provide explicit feedback for end users.

Due to the modern network applications such as distributed computation and remote experimentation, unprecedented end-to-end performance is required by the end users from the network ([12]). A self-managed network should be able to autonomously adjust its sending rate at the user ends according to the network operating condition. The ideal network operating condition is in the congestion avoidance stage, that is, network has the near maximized throughput but yet no excessive queue build-up in routers to cause package drop. In [2], such stage is called a "knee" position; while package drop occurs, the performance degrades rapidly and the network is at the "cliff" position and in congestion control stage. The idea employed in RED is similar: mark/drop each arriving package with some mark/drop probability whenever the queue length exceeds some level to avoid the router to be completely full. Using the same principle in this paper, but unlike previous results which consider explicit link to source feedback, we model the throughput (package received by the receiver end) and consider source-receiver feedback. We estimate online the queue size in router from the noisy throughput feedback, and drive the TCP window dynamics accordingly. Since the estimator and control are based on the throughput feedback (acknowledgement from receiver), we place the complex functions in the hosts and not inside the network. A block diagram representation of our control scheme is shown in Figure 1.

The rest of the paper is organized as follows. In Section 2, we present the fluid-flow based model for TCP window dynamics and router queue dynamics, and define our problem of "output feedback stabilizing control for TCP congestion avoidance". Then in Section 3, we design a linear-quadratic-Gaussian controller to solve the proposed problem. And finally, we conclude and discuss future research issues in Section 4.

2 Models and Control Problem Formulation

Let N TCP flows traverse a single bottleneck router. We adopt a dynamic model of TCP derived in [9, 3] using fluid-flow and stochastic differential equation analysis. The model relates the average value of key network variables by the following nonlinear differential equations:

$$\begin{aligned} \dot{W}(t) &= \frac{1}{R(t)} - \frac{W(t)W(t-R)}{2R(t-R(t))}p(t) + \omega_1(t) && (1) \\ \dot{q}(t) &= \frac{W(t)}{R(t)}N(t) - C + \omega_2(t) && (2) \\ R(t) &= \frac{q(t)}{C} + T_p && (3) \end{aligned}$$

where W is the average TCP window size in packets; $q(\cdot)$ is the average queue length in packets; $R(\cdot)$ is the round-trip time in seconds; T_p is the propagation delay in seconds; C is the bottleneck link capacity in packets/sec; N is the load factor (number of TCP sessions); $\omega_1(t), \omega_2(t)$ are stochastic disturbances (which include random parameter variations and uncertainties from un-modelled dynamics); and p is the probability of packet mark.

Note that (1) describes the window-size dynamics of TCP source, and (2) describes the queue dynamics at the bottleneck router. Unlike the control configuration in RED ([3]), where $p(t)$ is driven by an AQM control law taking $q(t)$ as the input, our new TCP congestion avoidance algorithm is to design $p(t)$ based on the on-line estimation of $q(t)$ from the throughput measurement (acknowledgement received by the source from the receiver). Therefore, the time delay in the term $p(t)$ does not exist in our model comparing to the model in ([3]).

We model the network throughput $Y(t)$ (in packets/sec), that is, the amount of data received by the receiver in unit time (assuming infinite receiver buffer). We have the following algebraic equation:

$$NW(t) = Y(t)R(t) + q(t) \qquad (4)$$

which means that the total packets W is equal to the number YR of packets in transit plus the number q of packets in the queue of the router.

Denote the acknowledgement received by the source as $z(t)$. Assume that it is imperfect measurement and takes the following form:

$$Z(t) = Y(t) + n(t) \qquad (5)$$

where $n(t)$ is the measurement uncertainty.

The control block diagram representation is shown in Figure 2. It can be seen that, instead of using an AQM controller at the router as in [3], we design congestion avoidance controller which takes the throughput measurement as input. *Using this scheme, we place the complex functions at the source instead of the router.*

Assume $N(t) = N, R(t) = R_0$, where N, R_0 are constants. The equilibrium of (1) and (2) are obtained by letting $\dot{W} = 0$ and $\dot{q} = 0$, which gives:

$$\begin{aligned} W_0^2 p_0 &= 2 \\ W_0 &= \frac{R_0 C}{N} && (6) \end{aligned}$$

Linearize the nominal system of (1) and (2) around the equilibrium (6). Following the same procedure as shown in the Appendix of [3], the following delayed differential equations are obtained:

$$\begin{aligned} \Delta\dot{W}(t) &= -\frac{N}{R_0^2 C}\left[\Delta W(t) + \Delta W(t-R_0)\right] \\ &\quad -\frac{R_0 C^2}{2N^2}\Delta p(t) && (7) \\ \Delta\dot{q}(t) &= \frac{N}{R_0}\Delta W(t) - \frac{1}{R_0}\Delta q(t) && (8) \end{aligned}$$

Furthermore, if

$$\frac{N}{R_0^2 C} = \frac{1}{W_0 R_0} \ll \frac{1}{R_0},$$

that is,

$$W_0 \gg 1,$$

by examining the transfer function from the input to output of (7), the delay is not significant and can be ignored for the stability analysis (see Appendix II in [3] for a detail justification). The following simplified dynamics is obtained for (1) and (2):

$$\begin{aligned} \Delta\dot{W}(t) &= -\frac{2N}{R_0^2 C}\Delta W(t) - \frac{R_0 C^2}{2N^2}\Delta p(t) + \omega_1(t) && (9) \\ \Delta\dot{q}(t) &= \frac{N}{R_0}\Delta W(t) - \frac{1}{R_0}\Delta q(t) + \omega_2(t) && (10) \end{aligned}$$

where

$$\Delta W(t) = W(t) - W_0,$$

$$\Delta q(t) = q(t) - q_0,$$
$$\Delta p(t) = p(t) - p_0,$$

and W_0, q_0, p_0 are quantities at the equilibrium.

The measurement output takes the form:

$$\Delta Z(t) = \frac{1}{R_0}(N\Delta W(t) - \Delta q(t)) + n(t) \quad (11)$$

where $\Delta Z(t) = Z(t) - Z_0$.

We further assume the uncorrelated disturbance and measurement error are white, zero-mean Gaussian random processes:

$$\begin{aligned} E[\omega_1(t)] &= 0 && (12)\\ E[\omega_1(t)\omega_1^T(\tau)] &= W_1\delta(t-\tau) && (13)\\ E[\omega_2(t)] &= 0 && (14)\\ E[\omega_2(t)\omega_2^T(\tau)] &= W_2\delta(t-\tau) && (15)\\ E[n(t)] &= 0 && (16)\\ E[n(t)n^T(\tau)] &= V\delta(t-\tau) && (17) \end{aligned}$$

where $\omega(t) = [\omega_1(t)\ \omega_2(t)]^T$, and W_1, W_2, V are spectral densities.

We review the stochastic stability definition which is of interest to us ([6]).

Definition 1 *For the stochastic system*

$$\dot{x}(t) = ax(t) + b\omega(t) \quad (18)$$

where a, b are constant matrices and ω is a zero-mean Gaussian white noise process, it is asymptotically stable in the mean if

$$\lim_{t\to\infty} E\{x(t)\} = 0 \quad (19)$$

where $E\{x(t)\}$ is the expected value of $x(t)$.

Our control problem is defined as:

Output Feedback Stabilizing Control for TCP Congestion Avoidance: Given $\Delta W, \Delta Z$ observable by the TCP source, design a dynamic output feedback control law of the form

$$\dot{\sigma} = \nu(\Delta W, \Delta Z, \sigma), \qquad \Delta p = \mu(\Delta W, \sigma) \quad (20)$$

such that the system (9) and (10) are asymptotically stable in the mean.

3 An Output Feedback Stabilizing Controller

To solve the problem of output feedback stabilizing control defined above, we first design an state estimator to dynamically estimate δq.

To ease the presentation, we denote:

$$x_1 = \Delta W, \quad x_2 = \Delta q, \quad z = \Delta Z, \quad u = \Delta p$$

and $x = [x_1 \quad x_2]^T$ is the state vector, yz is the output variable, u is the input variable, $\omega = [\omega_1\ \omega_2]^T$ is the disturbance vector. The system equations are re-written as:

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} -\frac{2N}{R_0^2 C} & 0 \\ \frac{N}{R_0} & -\frac{1}{R_0} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} -\frac{R_0 C^2}{2N^2} \\ 0 \end{bmatrix} u \\ &\quad + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \omega \\ &\triangleq Ax + Bu + F\omega && (21)\\ z &= \begin{bmatrix} \frac{N}{R_0} & -\frac{1}{R_0} \end{bmatrix} x + n \\ &\triangleq Hx + n && (22) \end{aligned}$$

Since the queue length $x_2(t)$ is not directly measurable in the end host, we have to estimate it from the noisy output measurement z. To achieve this, we first design a reduced-order Kalman-Bucy filter.

3.1 The Reduced-Order Filter Gain Computation

Since $x_1(t)$ is directly measurable, we design a reduced-order filter instead of a full-order one. The full-order Kalman-Bucy filter design can be found in [14].

Define the reduced-order error covariance, P_r, which is evolved according to the reduced-order dynamics:

$$\dot{P}_r = A_r P_r + P_r A_r^T + F_r W_2 F_r^T - P_r H_r^T V^{-1} H_r P_r \quad (23)$$

where $A_r = -\frac{1}{R_0}, F_r = 1, H_r = -\frac{1}{R_0}$. So the covariance estimate equation turns to:

$$\dot{P}_r = -\frac{2}{R_0}P_r + W_2 - \frac{1}{R_0^2}P_r^2 V^{-1} \quad (24)$$

And we construct the linear optimal filter gain as:

$$K_r = P_r H_r^T V^{-1} = -\frac{1}{R_0}V^{-1}P_r \quad (25)$$

3.2 The Reduced-Order State Estimate

Denoting $\hat{x}_2$ as the estimated value of x_2, the reduced-order dynamic estimator is constructed as the following:

$$\dot{\hat{x}}_2 = \frac{N}{R_0}x_1 - \frac{1}{R_0}\hat{x}_2 + K_r(z - \hat{z}) \quad (26)$$

where K_r is the filter gain obtained from (25), and $\hat{z} = \frac{1}{R_0}(Nx_1 - \hat{x}_2)$.

Define $\tilde{x}_2 = x_2 - \hat{x}_2$. Then the state error dynamics is:

$$\dot{\tilde{x}}_2 = -\frac{1}{R_0}(1 - K_r)\tilde{x}_2 + \omega_2 + K_r n \qquad (27)$$

We have the following proposition for the stability of the designed filter.

Proposition 1 *The error dynamics (27) with (24) and (25) is asymptotically stable in the mean.*

The proof of Proposition 1 is given in the Appendix.

3.3 The Controller Gain Construction

We construct a linear quadratic controller:

$$u = L \begin{bmatrix} x_1 \\ \hat{x}_2 \end{bmatrix} \qquad (28)$$

where $\hat{x}_2$ is the output of the estimator (26), and

$$L = -B^T \Gamma, \qquad (29)$$

where Γ is the solution to the following algebraic Riccati equation

$$A^T\Gamma + \Gamma A - 2\Gamma B B^T + Q = 0 \qquad (30)$$

where Γ, Q are positive-definite symmetric matrices.

We are now in the position to state the following theorem:

Theorem 1 *The controller (28) solves the problem of output feedback stabilizing control for TCP congestion avoidance.*

The proof of Theorem 1 is given in the Appendix. The output feedback control with the dynamic estimator construction is illustrated in the block diagram of Figure 3.

4 Conclusions

We proposed the idea of using output feedback control for the TCP end-to-end congestion avoidance. Instead of changing router algorithms to provide explicit feedback signals for the user like RED, we dynamically estimate the queue size in the router from the noisy throughput feedback. We adopt the models derived in [3, 9] describing TCP window and queue dynamics. A linear quadratic Gaussian controller is constructed, and the TCP window dynamics is driven by the online estimation of queue length. Asymptotical stability in the mean of both the window and filter dynamics is proved. The paper is to illustrate the basic idea of the proposed end-to-end feedback control through online estimation. Future work in modelling the packet loss such as random loss and other TCP dynamics (timeout *etc.*) will be conducted, as well as designing controllers for non-white uncertainties. Also, implementation in NS and extensive simulations are planned.

5 Appendix

Proof of Proposition 1: From (27), take the expectation of each side of the equation and we get:

$$\frac{d}{dt}E\{\tilde{x}_2\} = -\frac{1}{R_0}(1 - K_r)E\{\tilde{x}_2\} \qquad (31)$$

Choose the positive definite Lyapunov function:

$$V[E\{\tilde{x}_2(t)\}] = P_r(E\{\tilde{x}_2\})^2 \qquad (32)$$

From (24), the steady-state solution is:

$$0 = -\frac{2}{R_0}P_r + W_2 - \frac{1}{R_0^2}P_r^2 V^{-1} \qquad (33)$$

The expectation of time derivative of the Lyapunov function is:

$$\begin{aligned} \dot{V}[E\{\tilde{x}_2(t)\}] &= 2P_r E\{\tilde{x}_2\}\frac{d}{dt}E\{\tilde{x}_2\} \\ &= 2P_r E\{\tilde{x}_2\}\left[-\frac{1}{R_0}(1 - K_r)E\{\tilde{x}_2\}\right] \\ &= (E\{\tilde{x}_2\})^2\left[-\frac{2}{R_0}P_r - \frac{2}{R_0^2}P_r^2 V^{-1}\right] \end{aligned} \qquad (34)$$

From (33), (34) turns to:

$$\dot{V}[E\{\tilde{x}_2(t)\}] = -(E\{\tilde{x}_2\})^2\left[W_2 + \frac{1}{R_0^2}P_r^2 V^{-1}\right] \qquad (35)$$

which is negative definite.

Therefore, the system (31) is asymptotically stable in the mean. This concludes the proof of Proposition 1.

Proof of Theorem 1: From (21) and (27), the overall system dynamics is governed by:

$$
\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{\tilde{x}}_2 \end{bmatrix} = \begin{bmatrix} -\frac{2N}{R_0^2 C} & 0 & 0 \\ \frac{N}{R_0} & -\frac{1}{R_0} & 0 \\ 0 & 0 & -\frac{1}{R_0}(1-K_r) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \tilde{x}_2 \end{bmatrix} + \begin{bmatrix} -\frac{R_0 C^2}{2N^2} \\ 0 \\ 0 \end{bmatrix} u + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & K_r \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ n \end{bmatrix} \tag{36}
$$

Take the expectation of both sides:

$$
\begin{bmatrix} \frac{d}{dt}E\{x\} \\ \frac{d}{dt}E\{\tilde{x}_2\} \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & -\frac{1}{R_0}(1-K_r) \end{bmatrix} \begin{bmatrix} E\{x\} \\ E\{\tilde{x}_2\} \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} E\{u\} \tag{37}
$$

Substitute (28) into (37), we get

$$
\begin{bmatrix} \frac{d}{dt}E\{x\} \\ \frac{d}{dt}E\{\tilde{x}_2\} \end{bmatrix} = \begin{bmatrix} A+BL & -L_2 B \\ 0 & -\frac{1}{R_0}(1-K_r) \end{bmatrix} \begin{bmatrix} E\{x\} \\ E\{\tilde{x}_2\} \end{bmatrix} \tag{38}
$$

Since the matrix of (38) is block triangular, from its characteristic equation, it can been seen that the sets of poles of the combined system consists of the union of the control poles and the estimator poles. Since both of the control poles and estimator poles have negative real parts, the overall system is asymptotically stable in the mean.

References

[1] Y. Chait, C. V. Hollot, V. Misra, H. Han, and Y. Halevi. Dynamic analysis of congested TCP networks. In *Proceedings of 2002 American Control Conference*, 2002.

[2] D. M. Chiu and R. Jain. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Computer Networks and ISDN Systems*, 17:1–14, 1989.

[3] C. V. Hollot, V. Misra, D. F. Towsley, and W. B. Gong. A control theoretic analysis of RED. In *INFOCOM*, pages 1510–1519, 2001.

[4] V. Jacobson. Congestion avoidance and control. *ACM Computer Communication Review; Proceedings of the Sigcomm '88 Symposium in Stanford, CA, August, 1988*, 18, 4:314–329, 1988.

[5] F. Kelly. Mathematical modelling of the internet. In *Bjorn Engquist and Wilfried Schmid (Eds.), Mathematics Unlimited - 2001 and Beyond@ Springer*. 2001.

[6] H. J. Kushner and G. G. Yin. *Stochastic Approximation Algorithms and Applications*. Springer, New York, 1997.

[7] S. H. Low and D. E. Lapsley. Optimization flow control — I: basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, 7(6):861–874, 1999.

[8] S. H. Low, F. Paganini, and J. C. Doyle. Internet congestion control. *IEEE Control Systems Magazine*, 22(1):28–43, 2002.

[9] V. Misra, W. B. Gong, and D. F. Towsley. Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. In *SIGCOMM*, pages 151–160, 2000.

[10] F. Paganini. Flow control via pricing: A feedback perspective. In *Proceedings 2000 Allerton Conference, Monticello, IL.*, 2000.

[11] L. L. Peterson and B. S. Davie. *Computer Networks: A Systems Approach*. Morgan Kaufmann Publishers, 2000.

[12] N. S. V. Rao and L. O. Chua. On dynamics of network transport protocols. In *Proc. of Workshop on Signal Processing, Communications, Chaos and Systems*, 2002.

[13] R. Srikant. Control of communication networks. In *Tariq Samad (Ed.), Perspectives in Control Engineering: Technologies, Applications, New Directions*, pages 462–488. IEEE Press, 2000.

[14] R. F. Stengel. *Optimal Control and Estimation*. Dover Publications, New York, 1994.

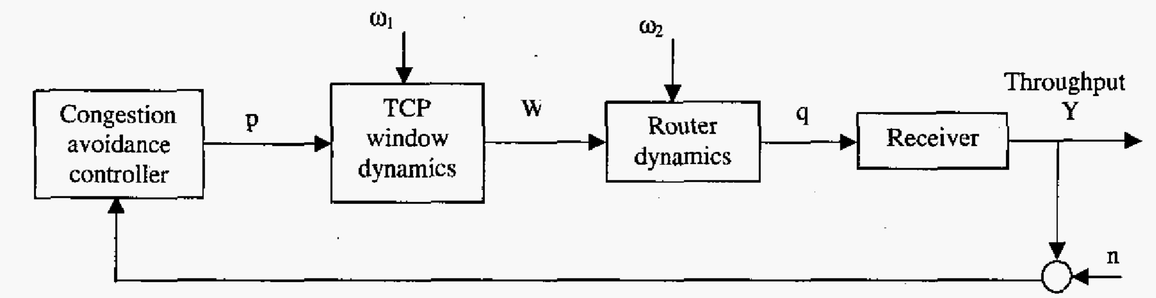

Figure 1: Block-diagram of congestion avoidance scheme

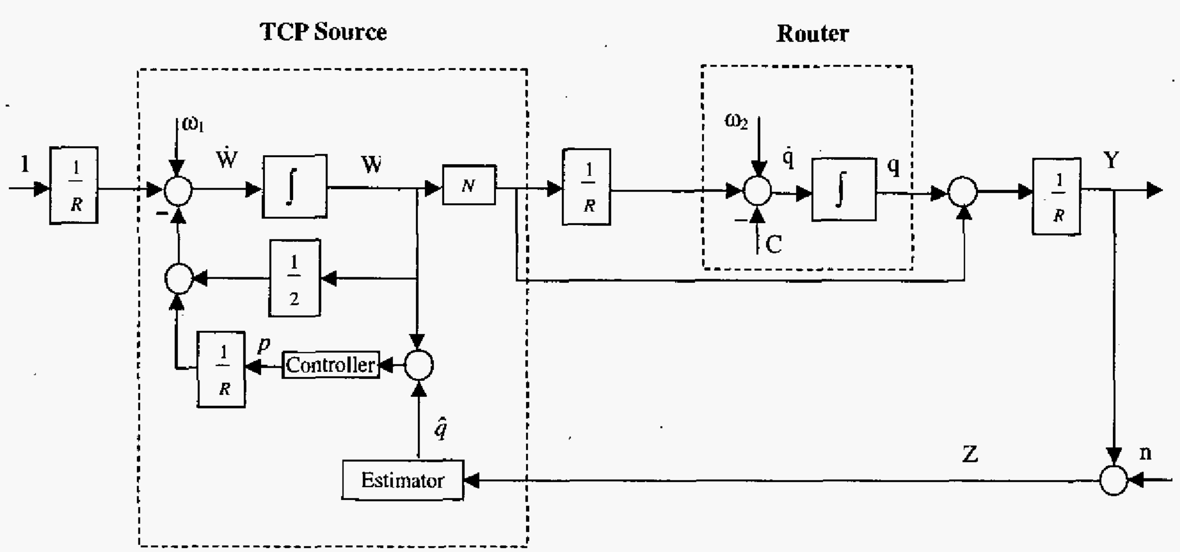

Figure 2: Block-diagram of TCP-based congestion avoidance flow-control scheme

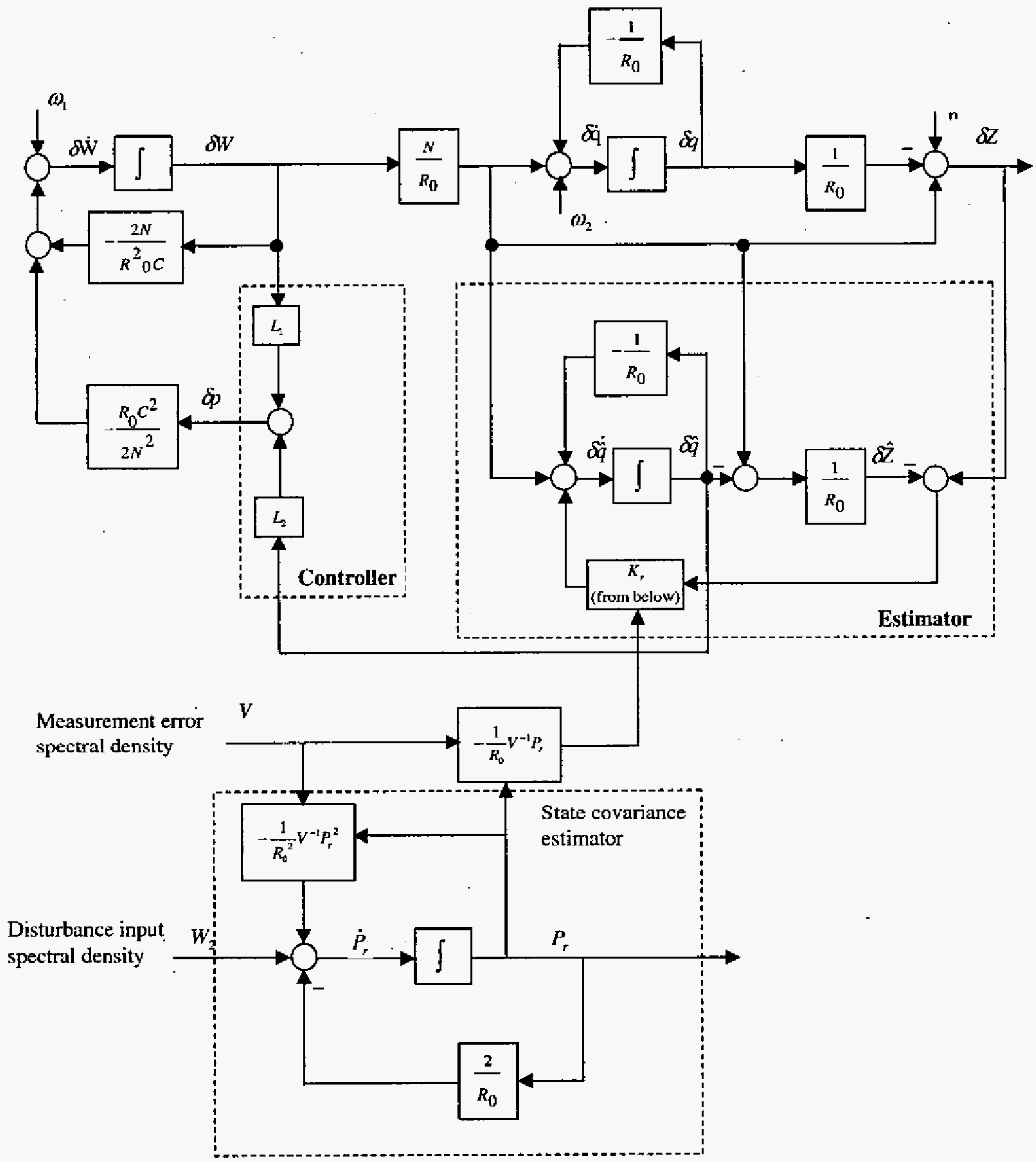

Figure 3: Block-diagram of the output feedback control scheme