An Adaptive Inverse Controller for Explicit Rate Congestion Control with Guaranteed Stability and Fairness

by

Kenneth P. Laberteaux, Charles E. Rohrs, and Panos J. Antsaklis

As Appeared In

The International Journal of Control, Volume 76, Number 1, pages 24-47, 2003.

Abstract – This paper examines explicit rate congestion control for data networks. The Available Bit Rate (ABR) service category of Asynchronous Transfer Mode (ATM) networks serves as an example explicit rate system. However, the results of this paper are applicable to other systems as well. After a plant model is established, a control strategy based on approximate inverse concepts is introduced. The control process includes a linear digital filter (with a DC or drift tap) that uses Normalized Least Mean Square (NLMS) adaptation. The convergence, stability and fairness properties of this control scheme are discussed. This work differentiates itself from the other contributions in the area of rate-based congestion control in its balanced approach of retaining enough complexity as to afford attractive, analytically-proven performance properties, but not so much complexity as to make implementation prohibitively expensive.

Please Direct Correspondences to klaberte@alumni.nd.edu

An Adaptive Inverse Controller for Explicit Rate Congestion Control with Guaranteed Stability and Fairness

by

Kenneth P. Laberteaux, Charles E. Rohrs, and Panos J. Antsaklis

Abstract – This paper examines explicit rate congestion control for data networks. The Available Bit Rate (ABR) service category of Asynchronous Transfer Mode (ATM) networks serves as an example explicit rate system. However, the results of this paper are applicable to other systems as well. After a plant model is established, a control strategy based on approximate inverse concepts is introduced. The control process includes a linear digital filter (with a DC or drift tap) that uses Normalized Least Mean Square (NLMS) adaptation. The convergence, stability and fairness properties of this control scheme are discussed. This work differentiates itself from the other contributions in the area of rate-based congestion control in its balanced approach of retaining enough complexity as to afford attractive, analytically-proven performance properties, but not so much complexity as to make implementation prohibitively expensive.

1 Introduction

In 1984, the Consultative Committee on International Telecommunications and Telegraph (CCITT), a United Nations organization responsible for telecommunications standards, selected Asynchronous Transfer Mode (ATM) as the paradigm for broadband integrated service digital networks (B-ISDN) [2]. ATM networks provide 6 service categories. Each category of service is customized for a particular type of traffic. Of these 5 categories, only one, Available Bit Rate (ABR), uses a feedback mechanism to create closed-loop congestion control.

Congestion control is a process by which networks use feedback to adjust the influx of data such that the customer's Quality of Service (QoS) requirements are met while simultaneously attempting to maximize the utilization of the network's resources. Networks that attempt to deliver more data than their capacity will experience congestion, leading to undesirable data loss, excessive delays, or both. The closed-loop nature of congestion control implies communication between the network and customer throughout the life of the connection. Generally this communication comes in the form of instructions to the customer to increase or decrease the data rate. Well suited for data that is not strongly delay sensitive, closed-loop congestion control uses a feedback mechanism and thus can draw heavily on the feedback control theory.

The complete ABR congestion control mechanism is described in [1] and [2]. This paper focuses on explicit rate congestion control. The plant description of Section 2 is an approximation to the mechanisms specified in [1]. The present challenge is to devise a controller that resides at the output queue of an ATM switch port and produces a single *explicit rate* to be sent to all ABR sources passing through the queue. The explicit rate must be chosen such that the incoming ABR bandwidth matches the available ABR bandwidth in some appropriate sense. Specifying a single explicit rate at time *n* for all sources ensures fairness. Matching the incoming ABR bandwidth to the available ABR bandwidth attains efficiency.

Kenneth P. Laberteaux (klaberte@alumni.nd.edu) and Charles E. Rohrs (crohrs@mit.edu) are affiliated with the Tellabs Research Center, 3740 Edison Lakes Parkway, Mishawaka, IN 46545, USA, Ph. 219-258-6400. Panos J. Antsaklis (antsaklis.1@nd.edu) is affiliated with the University of Notre Dame, Dept. of Electrical Engineering, 275 Fitzpatrick Hall, Notre Dame, IN 46556, USA, Ph. 219-631-5792.

The creation of a control mechanism for a switch that can work with the closed-loop congestion control mechanism specified by the ATM Forum [1] is one result of this paper. As is frequently the case when conducting research on practical problems such as this, there is a tension between solutions that are simple to implement and solutions that are simple to analyze. The solution offered here is simple to implement[1]; it should have an intuitive feel for those with a basic understanding of adaptive filters. This necessitates rather cumbersome analysis, which appears in Section 4, to fully characterize the behavior of the proposed system. This analysis provides strong guarantees of proper performance. The reader must never lose sight that despite the complexity of this analysis, the implementation of the congestion controller is relatively straightforward.

The contributions of this paper are as follows: A plant model is created to characterize the congestion control mechanism of ATM ABR. Several control strategies are presented and evaluated. One control strategy is chosen for its comparatively low computational cost, realizability and attractive convergence and fairness properties. These properties are determined through analysis. Further algorithm enhancements are briefly presented as well as corroborating results from simulation experiments. Comparisons to other schemes are made.

This paper's treatment of ATM ABR Congestion Control is quite general. The extent of ATM ABR deployment is still uncertain, however issues studied by this paper are likely to arise in future networking protocols and should not be considered applicable only to ATM ABR. Given the rate at which bandwidth consumption is increasing and computational costs are decreasing, it seems inevitable that at some point in the future, data networks will employ high-performance explicit rate congestion control mechanisms.

1.1 Available Bit Rate (ABR) Congestion Control

The standard for ABR service category [1] states that "the ABR service category provides a low cell loss ratio" and that "no numeric commitment is made about cell transfer delay," but both should be minimized. Key to this goal is avoiding congestion at any switching node in the ATM network; cells that arrive to a nearly full switch buffer will experience excessive delay, while cells arriving to a completely full buffer are lost entirely.

Congestion control for ABR traffic utilizes a feedback mechanism, namely *Resource Management* (RM) cells. An ABR Source periodically inserts RM cells into the stream of data cells. These RM cells pass through each switch along the path to the destination of the *virtual connection* (VC). The destination then returns the RM cell to the ABR source along the same path (but in reverse order) used for the forward virtual connection from source to destination[2]. RM cells moving from source to destination are called Forward RM cells and RM cells returning to the source are called Backward RM cells.

1.2 Related Work

Significant contributions to the understanding of congestion control in ATM ABR networks have been made in the past decade. Contributors include [2]-[34]. Benmohamed and Meerkov made a significant early contribution in plant modeling with [7] - [8]. The assumptions developed in [7] are widely employed. Reference [7] treats the single-node case, while [8] treats the multiple-node case, although in the end, through careful reasoning and imposing judicious assumptions, [8] arrives back essentially at the single bottleneck node case described in [7]. This paper makes a strong case for

[1] There do exist solutions to this problem with even simpler implementations, e.g. [20], [22], and [25]. These are discussed in Section 7.

[2] If for some reason a network does not send Backwards RM cells along the reverse path of the data flow, this network can generally use the same congestion control techniques, but with much longer action delays, with the expected performance degradation.

simplifying the congestion control problem to a single node study. Few investigators have deviated from this since. Benmohamed and Meerkov content themselves to place the closed-loop poles. No effort is made to cancel or affect the plant (and thus closed-loop) zeros. Importantly, the number of responsive sources and their action delays are assumed known, thereby avoiding computational complexity usually associated with congestion controllers.[3] Kolarov and Ramamurthy extend the results of [7] by using two proportional derivative controllers, one with high gain, one with low gain [9]. Switching between the two controllers is accomplished by comparing the utilization against a threshold.

Altman et al. make several contributions [11]-[16]. Of particular relevance, [11] discusses how a pure rate-matching algorithm, i.e. where the bandwidth available to ABR traffic is completely apportioned without regard of the current queue depth, will produce unacceptably long queues. However, [12] shows that under fairly general restrictions, under-allocating the available bandwidth, using either an additive or multiplicative constant, will ensure stability in the queue length. This gives some credibility to the rate matching schemes proposed by others and proposed here. Note that throughout [11]-[16], the number of sources and their action delays are assumed to be known. Also note that their models do not include the presence of ABR traffic which is controlled by other switches.

Raj Jain has made the best known contributions to the field of ATM ABR congestion control. His implementation-friendly Explicit Rate Indication for Congestion Avoidance (ERICA) algorithm [17], its predecessor, the ERPCA+ [19], and its successor, ERICA+ [20], work well in a large number of situations and appear to be favored by ATM switch designers. ERICA is computationally inexpensive to implement (as compared to the other contributions mentioned above) and has been shown, via simulations, to rapidly achieve max-min fairness in many cases. As such, it demonstrated the viability of explicit rate schemes at a time that many considered explicit rate congestion control to be an extravagant luxury. However, further study discovered various scenarios where max-min fairness was not achieved. In a 1998 contribution [22], persisting fairness concerns of ERICA+ prompted a new approach. The switch determines an effective number of sources. This effective number of sources N_{eff} assigned a specific fractional value to sources unable to use their fair share allocation. This approach is very similar to that suggested by Fulton and Li in 1997 [25] and marks an intersection of these two bodies of work. Imer also proposes a controller in the same vein [27]. A comparison of these approaches to the control scheme presented in this paper occurs in Section 7. Su et al. extend the approach of [25] by including queue-control in their calculation of the effective number of sources.

In addition, there has been significant contributions made in the ATM Forum. The ATM Forum has approved guidelines for the operation of ABR congestion control by defining the required behaviors and properties of ABR Sources, Destinations and Resource Management (RM) Cells [1]. As compared to the algorithms in favor of the ATM Forum, a slight increase in complexity can reap significant returns in performance and predictability. However, this push towards greater complexity must be a small one. Implementation cost should be a very important consideration. It is believed that the results of this paper will provide performance advantages that justify their implementation costs.

This work differentiates itself from the other contributions in the area of rate-based congestion control in its balanced approach of retaining enough complexity as to afford attractive, analytically-proven performance properties, but not so much complexity as to make implementation prohibitively expensive.

Another vein of congestion control research focuses on lower complexity-lower performance solutions for the Internet. Floyd proposed Random Early Detection (RED) [37] and Explicit Congestion

[3] The controller presented in Section 3.3.4 does not assume that the number of sources and delays (**B**) is given. This accounts for a significant amount of the controller's complexity.

Notification (ECN) [38], two methods for routers to signal congestion by probabilistically dropping (RED) or marking (ECN) packets. These two concepts have given rise to several suggestions for Internet congestion control using one-bit marking strategies (see [39] and [41] and references therein). This work has been greatly advanced by Misra's model relating packet loss probability to TCP throughput [40]. This group of one-bit, Internet-specific algorithms and our ATM ABR algorithm occupy very different places on the performance-cost curve, specifically our algorithm gives more performance with more cost. At best, the one-bit marking algorithms can match bandwidth to capacity only in the mean, requiring large buffers (as discussed in Section 7).

The general flow control problem has also received extensive investigation. This problem contemplates a network where internal links constantly advertise an updated price for its resource (generally price goes up as the demand approaches capacity). Sources purchase a part of the link's resource in order to maximize the source's utility-minus-cost function (see [36] and references within). This general approach is helpful in framing specific flow control problems, but required communication and computational resources are not generally available in ATM and TCP applications.

1.3 Outline of Paper

The remainder of this paper is as follows: Section 2 presents an appropriate model for the congestion control problem. Section 3, after presenting several candidates, identifies one control methodology for the plant specified in Section 2. Section 4 then examines the convergence and stability properties of the selected controller. Fairness is addressed in Section 5. Simulations presented in Section 6 demonstrate the proper functioning of the proposed controller as well as minor algorithm enhancements. Section 7 compares the proposed scheme to other proposals. Conclusions are made in Section 8.

2 Plant Definition

In this section, the plant is defined.

Since each switch implements its own, independent controller, one may consider the plant from the perspective of a single switch *SW*. A discrete-time model is used, where sample intervals correspond to control intervals, i.e. a new control action $u(n)$ is calculated for each n. port j of switch *SW* carries N simultaneous Available Bit Rate (ABR) sessions, and serves as an output port for data cells and an input port for backward resource management (RM) cells.

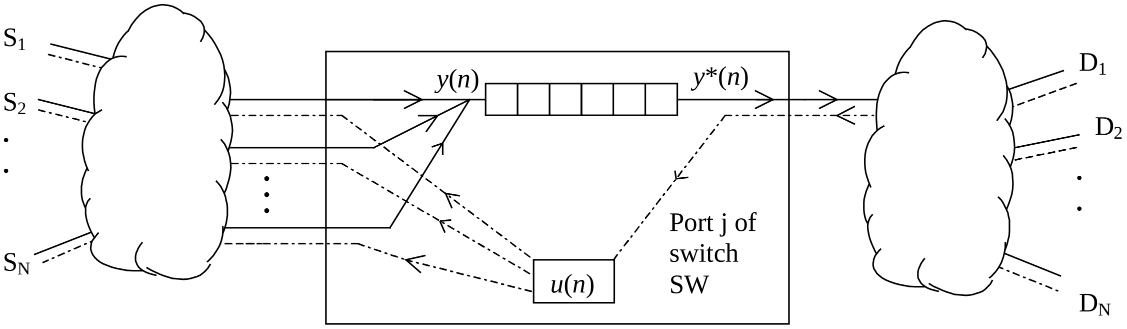


Figure 1 Plant from perspective of Switch Output Port

The present challenge is to devise a controller that resides at output port j of switch *SW* and produces a single explicit rate u to be sent to all ABR sources passing through the port. The explicit rate u must be chosen such that the incoming ABR bandwidth y matches the available ABR bandwidth y^* in some appropriate sense. Specifying a single explicit rate at time *n* for all sources ensures fairness. Matching y to $y*$ attains efficiency.

Port *j* generates a single desired rate $u(n)$ for all connections. As resource management (RM) cells for the *N* ABR virtual connections (VC's) pass through *j* on their return from destination to source, port *j* examines each, specifically the contents of each explicit rate (ER) field. If port *j* finds the ER field contains a rate above its current $u(n)$, port *j* overwrites the ER field with $u(n)$. The RM cell transports this explicit rate $u(n)$ to each ABR Source. It is assumed that for each of the *N* ABR virtual connections, at least one RM cell passes *j* during each sample interval. Rates $u(n)$, $y(n)$, and $y*(n)$ are in units of cells/second.

Output port *j* will observe changes to its input rate $y(n)$ as various sources (S_i) react to previously specified explicit rates $u(n-m)$. The *reaction delay, m,* as viewed by *j* for source S_i, is the time between *j*'s adjustment of its explicit rate to the time *j* measures this explicit rate as its input rate from S_i. These reaction delays will vary for different sources. Assume that there are b_0 sources that respond with reaction delay *d*, b_1 sources that respond with delay $d+1$, and b_{dB} with delay $d+dB$, where dB is a known upper bound on *j*'s reaction delay. In addition, some sources will be unresponsive to the explicit rate generated at j, e.g. they are bottlenecked by another port or the explicit rate of j is less than the source's promised Minimum Cell Rate. These unresponsive flows are modeled to have a total constant[4] bandwidth C. It is assumed that C, b_0, $b_1,\ldots,$ b_{dB} remain constant for periods of time long enough for adaptive identification to occur. Faster convergence speed of the adaptive algorithm results in better tracking of these time-varying parameters. The plant is therefore given by

$$y(n)=b_0u(n-d)+\cdots+b_{dB}u(n-d-dB)+C \tag{1}$$

$$y(n)=B\left(z^{-1}\right)u(n-d)+C \tag{2}$$

$$y(n)=\mathbf{B}^T\mathbf{u}(n-d)+C \tag{3}$$

where

$$\mathbf{B}\equiv\left[b_0,b_1,...,b_{dB}\right]^T \text{ and } \mathbf{u}(n)\equiv\left[u(n),u(n-1),...,u(n-dB)\right]^T.$$

Note that for convenience, filters in z^{-1} (denoting unit time delay) and time sequences in *n* are mixed in expressions, e.g. (2). Matrix notation is also used. Equations (1), (2), and (3) are equivalent.

Since the minimum delay in the plant is *d*, adjustments in $u(n)$ will not be observed until $n+d$. Therefore to generate $u(n)$, it must be decided at time *n* what the desired value of $y(n+d)$ should be. This desired bandwidth, which is notated as $y*(n+d\,|\,n)$, may reflect both bandwidth and buffer measurements[5] made up to time *n* (this may be generated by a prediction filter as in [16]). By extension,

[4] The generalized case where *C* is a random process is explored in [35], [34].

[5] Requested bandwidth can be reduced to shrink the buffer if it is too large.

in many cases, the input of the algorithm will be $y^*(n+d+V \mid n)$ (for some non-negative *V*), i.e. the desired value of $y(n+d+V)$ chosen at time *n*.

The goal of the congestion control mechanism of *SW* is to choose at time n the control signal $u(n)$ so as to minimize $E\left[\left(y(n+d+V)-y^*(n+d+V \mid n)\right)^2\right]$.

3 Selection of Controller for the Congestion Control Plant

This section presents several candidate control strategies for the plant model identified in Section 2. Rejected control strategies are briefly proposed and evaluated in Sections 3.1 and 3.2. The most attractive control scheme is presented in Section 3.3. Stability, convergence and convergence rate issues are presented in Section 4.

3.1 Control of Stable Non-Minimum Phase Plants

The plant (3) is an FIR filter and is thus bounded input-bounded output (BIBO) stable. The controller proposed in [24] cancels the dynamics of the plant by placing controller poles where plant zeros are located (all plant poles are at the origin). Specifically, using Normalized Least Mean Square (NLMS) adaptation [49], at each time n, determine an estimate of $\mathbf{B}(n)$, $\hat{\mathbf{B}}(n)$,

$$\hat{\mathbf{B}}(n+1)=\hat{\mathbf{B}}(n)+\frac{\mu\boldsymbol{\varphi}(n)}{\boldsymbol{\varphi}(n)^T\boldsymbol{\varphi}(n)}\left[y(n)-\hat{\mathbf{B}}(n)^T\boldsymbol{\varphi}(n)\right], \tag{4}$$

$\boldsymbol{\varphi}(n)\equiv\left[u(n),u(n-1),\ldots,u(n-dB),\phi_{DC}\right]^T$, and control signal $u(n)$ is calculated with

$$u(n)=\frac{y^*(n+d+V \mid n)}{\hat{\mathbf{B}}(n+1)}. \tag{5}$$

As discussed in [24], congestion controllers that determine an effective number of sources, e.g. [25] and [22], can be viewed as special one-tap version of (4)-(5), specifically the one-tap case, i.e. $\hat{\mathbf{B}}(n)=\hat{N}_{eff}(n)$. Reference [24] describes the advantages of expanding the one-tap controller of [25] and [22] to a controller using two or more taps.

One of the assumptions required for stable operation of controller (5) is that the zeros of $B\left(z^{-1}\right)$ lie within the unit disk, i.e. that the plant $B\left(z^{-1}\right)$ is minimum-phase. However, as discussed in [24], the underlying physical plant does not suggest that this assumption is appropriate. A non-minimum phase plant is not only possible, but quite likely. Thus a controller capable of controlling a non-minimum phase (NMP) plant is needed.

There has been significant progress in the control of non-minimum phase (NMP) plants in the past twenty years. This section reviews some of the better-known NMP-plant controllers and discusses their appropriateness for the explicit rate congestion control problem. To simplify the presentation of basic concepts, consider the plant (2) with zero DC offset, i.e. $C=0$,

$$y(n)=B\left(z^{-1}\right)u(n-d). \tag{6}$$

In Section 3.3.2, a coefficient, the *DC tap*, is added to the identification filter for the purposes of matching DC offsets, thereby extending the following comments.

3.1.1 Approximate Inversion using FIR Filters

This section introduces the concept of approximately inverting one finite impulse response (FIR) filter with another FIR filter. This concept is attractive due to its simplicity and its attractive stability properties. It is therefore the common theme of controllers in the rest of this paper.

The concept of approximately inverting one FIR filter with another FIR filter is not new, e.g. [42], [44]. This concept of *Adaptive Approximate Inverse Control* seems to have gained relatively little attention despite its attractive characteristics. Its most attractive attribute is its ability to control non-minimum phase stable plants without introducing the potential for instability.

A general plant $B\left(z^{-1}\right)$ can have zeros inside and outside the unit circle. Consider the ideal inverting IIR filter $B^{-1}\left(z^{-1}\right) \equiv 1/B\left(z^{-1}\right)$. The time-domain realization $b^{-1}(n) \equiv \mathcal{Z}^{-1}\left\{B^{-1}\left(z^{-1}\right)\right\}$, where $\mathcal{Z}^{-1}\left\{x\left(z^{-1}\right)\right\}$ is the inverse Z-transform of $x\left(z^{-1}\right)$ [43], is not specified until a region of convergence is specified. If the region of convergence is chosen to include the unit circle, the impulse response is generally two-sided, i.e. non-zero for both positive and negative *n*. However, unless there is a root of $B\left(z^{-1}\right)$ on the unit circle, i.e. the region of convergence includes the unit circle, $\left|b^{-1}(n)\right|$ converges to zero exponentially as $n \to \pm\infty$ [43]. Delaying $b(n)^{-1}$ by V samples, the resulting $b(n-V)^{-1}$ can be truncated to form an FIR filter if $b(n-V)^{-1} \approx 0$ for $n<0$ and $n>dQ$ ($V \geq 0$). The resulting causal $(dQ+1)$ tap FIR filter $q(n)$ approximates $b(n-V)^{-1}$ increasing well with increasing choices of V and dQ (if $B(z)$ has no roots on the unit circle).

$$Q\left(z^{-1}\right) = q_0 + q_1 z^{-1} + \ldots + q_{dQ} z^{-dQ} \approx \frac{z^{-V}}{B\left(z^{-1}\right)}$$

Note that adding delay is a common characteristic of non-minimum phase plant control, given the large phase lags inherent in non-minimum-phase plants. The above explanation does not appear in [42] or [44], although the more recent [45] makes brief, similar comments.

3.1.2 Indirect Adaptive Control using an Approximate Inverse FIR Filter

In this section, a previously published control strategy is briefly outlined, as this controller is an example of the concept of Approximate Inverse FIR Control described in Section 3.1.1 and motivates the control strategies presented in Section 3.3. Yahagi and Lu propose the following FIR, direct controller [42]: Estimate $\hat{\mathbf{B}}(n+1)$ using (4), then calculate $\hat{\mathbf{Q}}(n+1) \equiv \left[\hat{q}_0(n+1), \ldots, \hat{q}_{dQ}(n+1)\right]^T$ with

$$\hat{\boldsymbol{\beta}}(n+1) \equiv \begin{bmatrix} \hat{b}_0(n+1) & \hat{b}_1(n+1) & \cdots & \hat{b}_{dB}(n+1) & 0 & 0 & 0 \\ 0 & \hat{b}_0(n+1) & \hat{b}_1(n+1) & \cdots & \hat{b}_{dB}(n+1) & 0 & 0 \\ \vdots & & & & & & \vdots \\ 0 & 0 & 0 & \cdots & \hat{b}_0(n+1) & \hat{b}_1(n+1) \cdots & \hat{b}_{dB}(n+1) \end{bmatrix},$$

$$\hat{\mathbf{Q}}(n+1)=\arg_{\mathbf{Q}}\left(\hat{\boldsymbol{\beta}}(n+1)^T\mathbf{Q}-\mathbf{e}_v\right)^T\left(\hat{\boldsymbol{\beta}}(n+1)^T\mathbf{Q}-\mathbf{e}_v\right),\mathbf{e}_v\equiv[0,0,...,0,1,0,...0]^T \tag{7}$$

with the $(V+1)$'th element of $\mathbf{e}_v$ equal to 1. The solution of (7) is given by the Wiener solution [49]

$$\hat{\mathbf{Q}}(n)=\left(\hat{\boldsymbol{\beta}}(n)\hat{\boldsymbol{\beta}}(n)^T\right)^{-1}\hat{\boldsymbol{\beta}}(n)\mathbf{e}_v \tag{8}$$

Finally, produce the control signal as $u(n)=\hat{\mathbf{Q}}(n+1)^T\mathbf{y}^*(n+d+V\mid n)$

The computational cost of evaluating (8) can be reduced by using a Levinson algorithm [42], but is still $O\left(dQ^2\right)$. A computationally less expensive alternative is presented next.

3.2 Rejected Direct Adaptive Control using an Approximate Inverse FIR Filter

In this section, two adaptive controllers are presented. Although neither can be used in their presented form, both controllers presented in Section 3.2 motivate the controller of Section 3.3.

The controllers in Sections 3.2 and 3.3 are direct adaptive controllers. The term *direct* specifies that controller parameters are directly identified using an adaptive identification method. In contrast, the *indirect* controller of Section 3.1.2 identifies the plant parameters first and then derives controller parameters from the estimates of the plant parameters. The controllers in this section were developed in an attempt to find a direct formulation of the indirect controller presented in Section 3.1.2. The motivation for finding a direct formulation is reducing computational cost by eliminating the calculation of (8).

3.2.1 A Potentially Non-Convergent Adaptive Controller

Consider a direct controller where $\hat{\mathbf{Q}}$ is directly estimated from plant input and output signals, shown in Figure 2. Using the Normalized Least Mean Squares (NLMS) method [49], adaptively estimate $\hat{\mathbf{Q}}_{y^*}(n)$ to obtain the ideal $\mathbf{Q}_{y^*,0}$ that minimizes the least squares criterion $\mathbf{Q}_{y^*,0}=\arg\min_{\hat{\mathbf{Q}}_{y^*}}E\left[e^2(n)\right]$.

A careful study of Figure 2 shows that convergence cannot be assured. Briefly stated, the update error $e(n)$ is not the required inner product of the parameter error vector $\hat{\mathbf{Q}}_{y^*}(n)-\mathbf{Q}_{y^*,0}$ and input vector $\mathbf{y}^*(n+V\mid n-d)$, but instead this inner product is filtered by the FIR filter $B\left(z^{-1}\right)$. Since $B\left(z^{-1}\right)$ is not strictly positive real (SPR), except for the case of $dB=0$, i.e. $B\left(z^{-1}\right)=b_0$, convergence cannot be assured [48]. Therefore, the controller of Figure 2 is disqualified as viable explicit rate congestion controller.

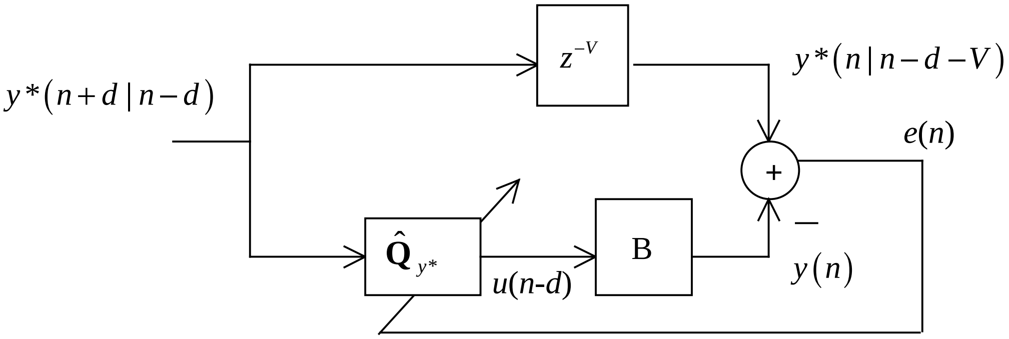


Figure 2 A Direct Adaptive Controller System for Controlling MA Plant That MAY NOT CONVERGE.

3.2.2 An Unrealizable Controller

Consider a second control method by inverting the order of $\hat{\mathbf{Q}}$ and $\mathbf{B}$ in Figure 2, as in Figure 3. The auxiliary signal $t(n)$ is introduced. The issue of filtering the coefficient error vector is overcome.

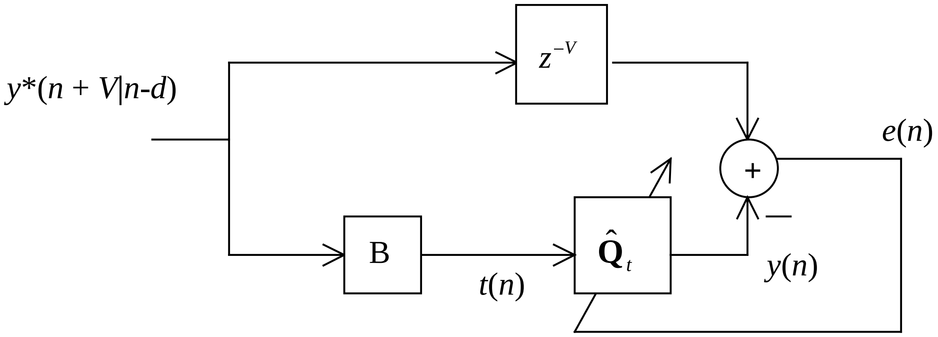


Figure 3 Inverting Plant and Controller, AN UNREALIZABLE CONTROLLER ($t(n)$ is not available).

Comparing Figure 3 with Figure 2, clearly $\mathbf{Q}_{t,0} = \mathbf{Q}_{y*,0} = \arg\min_{\hat{\mathbf{Q}}_t} E\left[e^2(n)\right]$.

Then the Wiener solution, as given by [49], is

$$\mathbf{Q}_{t,0} = \mathbf{Q}_{y*,0} = \left(\mathbf{B}\mathbf{R}_{y^*}\mathbf{B}^T\right)^{-1}\mathbf{B}\mathbf{R}_{y^*}\mathbf{e}_V \qquad (9)$$

where $\mathbf{R}_{y^*} \equiv E\left[\mathbf{y}^*(n+d+V\,|\,n)\,\mathbf{y}^*(n+d+V\,|\,n)^T\right]$ is a dB+dQ+1 by dB+dQ+1 auto-correlation matrix assumed to be full rank, i.e. there is sufficient excitation. Note that if $\{y*\}$ is white noise with $\mathbf{R}_{y^*} = \sigma^2\mathbf{I}$, then (9) is equivalent to (8).

However, there is a problem. Since $B\left(z^{-1}\right)$ is unknown, $t(n)$ cannot be created. The formulation of Figure 3 is useful for insight and intuition, but cannot be implemented.

3.3 Direct Adaptive Approximate Inverse Control

In this section, a control strategy is presented that was developed expressly for an ABR explicit rate congestion controller. It is based in part on the identification scheme shown in Figure 3. The goal of this control strategy is to minimize $E\left[\left(e_u(n)\right)^2\right]$ as given by Figure 4, as discussed below.

However, further investigation revealed that the control methodology presented in this section is nearly identical to *Adaptive Inverse Control*, a methodology previously proposed by Widrow and Walach [44]. The approach here distinguishes itself from the approach of [44] in its use of the Normalized Least Mean Square (NLMS) adaptation scheme; Widrow uses Least Mean Square (LMS). The advantage of NLMS is that it allows setting the adaptive gain to its optimal value (=1), resulting in the fastest possible stable convergence. Use of NLMS requires a new proof of convergence, which appears in Section 4.

3.3.1 A Convergent, Realizable Adaptive Control Strategy

An attractive adaptive control strategy must converge to desirable parameters and be realizable; the controllers of Sections 3.2.1 and 3.2.2 each failed in one of these respects. Yet both motivate the controller presented in this section. When placed in series with **B**, an ideal FIR controller will approximate a delayed impulse. The adaptation error must not be filtered by a non-SPR filter (e.g. by **B**, as it was in Figure 2). Figure 3 achieves this. Unfortunately $t(n)$ is not available. However, if in Figure 3, the signals $y^*(n|n-d-V)$, $t(n)$, and $y(n)$ are replaced respectively with $u(n-d)$, $y(n)$, and $\hat{u}(n-V-d)$, as in Figure 4, all necessary signals are available.

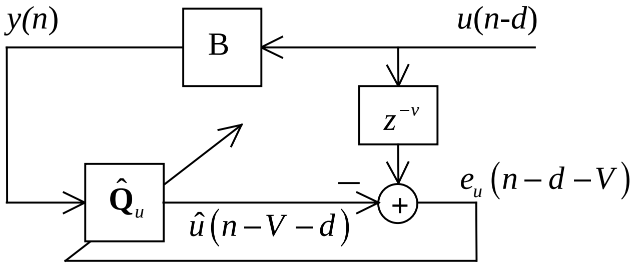


Figure 4 Direct Inverse Plant Modeling

Figure 4 specifies the suggested structure for controller identification. It will be shown that $\mathbf{Q}_{u,0} \approx \mathbf{Q}_{t,0}$, and that $\hat{\mathbf{Q}}_u$ can be found using a NLMS estimation process.

Define $\mathbf{Q}_{u,0} \equiv \arg\min_{\mathbf{Q}_u} E\left[e_u(n)^2\right]$ and the dB+dQ+1 by dB+dQ+1 auto-correlation matrix $\mathbf{R}_u \equiv E\left[\mathbf{u}(n)\mathbf{u}(n)^T\right]$, (assumed to be full rank). Then the Wiener solution gives

$$\mathbf{Q}_{u,0} = \left(\mathbf{B}\mathbf{R}_u\mathbf{B}^T\right)^{-1}\mathbf{B}\mathbf{R}_u\mathbf{e}_V \tag{10}$$

The goal of this control strategy is to find the best estimate $\hat{\mathbf{Q}}(n)$ of $\mathbf{Q}_{u,0}$ such that $E\left[e_u(n)^2\right]$ is minimized. Section 4 contains two proofs of convergence. The proof in Section 4.1 shows that $\hat{\mathbf{Q}}(n)$

converges to $\mathbf{Q}_{u,0}$ assuming that $\{u\}$ is Gaussian (Assumption 1). The proof in Section 4.2 gives a second, independent proof showing that $y(n)$ converges to $y^*(n)$ while making no assumption on the signals $\{u\}$ and $\{y^*\}$ (although Assumption 7 disqualifies a potential pathological relationship between $y^*(n)$ and $\hat{\mathbf{Q}}(n)$).

Although (10) and (9) are not equal, except for the case of $B(z^{-1}) = b_0$, both provide an approximate inverse of **B**. To better compare $\mathbf{Q}_{y^*,0}$ and $\mathbf{Q}_{u,0}$, consider the formulation of Figure 5.

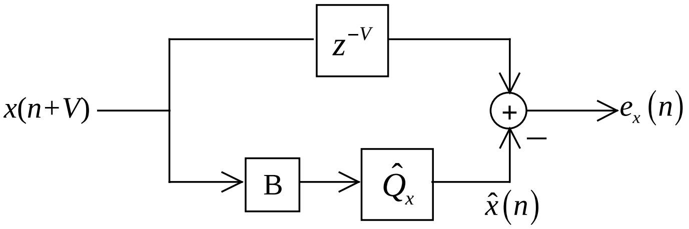


Figure 5 $\mathbf{Q}_{x,0}$ is a function of $\{x\}$.

The error power $E\left[e_x(n)^2\right]$ is to be minimized as a function of $\hat{\mathbf{Q}}_x$. Clearly $\hat{\mathbf{Q}}_x$ must approximately invert **B**, but the specific $\mathbf{Q}_{x,0} = \left(\mathbf{B}\mathbf{R}_x\mathbf{B}^T\right)^{-1}\mathbf{B}\mathbf{R}_x\mathbf{e}_V$ is a function of the spectral content of excitation signal $\{x\}$. For example, if $\{x\}$ is primarily a low-frequency signal, then $\hat{\mathbf{Q}}_x$ can only hope to match the inverse of **B** at these low frequencies; $\hat{\mathbf{Q}}_x$ may not be a good match for the inverse of **B** at higher frequencies not represented by $\{x\}$.

For $\mathbf{Q}_{y^*,0}$, the driving signal is $\{y^*\}$, while the driving signal of $\mathbf{Q}_{u,0}$ is $\{u\}$. Therefore the difference between $\mathbf{Q}_{y^*,0}$ and $\mathbf{Q}_{u,0}$ is simply attributable to the differences of the spectral characteristics of $\{y^*\}$ and $\{u\}$ due to the filtering of $\hat{\mathbf{Q}}_x$. When $\{y^*\}$ and $\{u\}$ have similar spectral characteristics, i.e. when the filter $\hat{\mathbf{Q}}_x$ is relatively spectrally flat, then by (9) and (10), $\mathbf{Q}_{y^*,0} \approx \mathbf{Q}_{u,0}$.

In practice, it is more important to note that if $\hat{\mathbf{Q}}_x$ could perfectly invert $\mathbf{B}$ in Figure 5, then assuming sufficient excitation in $\{x\}$, $\mathbf{Q}_{x,0} = \mathbf{B}^{-1}$ regardless of the specific spectrum $\{x\}$. For Figure 2 and Figure 4, if both $\hat{\mathbf{Q}}_{y^*}$ and $\hat{\mathbf{Q}}_u$ have enough taps to well approximate the inverse of $\mathbf{B}$, then assuming sufficient excitation, $\mathbf{Q}_{y^*,0} \approx \mathbf{B}^{-1} \approx \mathbf{Q}_{u,0}$.

3.3.2 Removing DC Offsets with a DC tap

For Sections 3.1 - 3.3.1, the analysis has been simplified by assuming the plant parameter $C = 0$. To extend these results to the non-zero C case, a DC tap is appended to the estimator and controller. This simply requires increasing $\hat{\mathbf{Q}}_u(n)$ by one tap, i.e. incrementing dQ by one, and appending a constant y_{DC} to the vectors $\mathbf{y}$ and $\mathbf{y}^*$. Redefine

$$\mathbf{y}(n) \equiv \left[y(n), y(n-1), \ldots, y(n-dQ), y_{DC} \right]^T \tag{11}$$

$$\mathbf{y}^*(n+d+V \mid n) \equiv \left[y^*(n+d+V \mid n), \ldots, y^*(n+d+V-dQ \mid n-dQ), y_{DC} \right]^T . \tag{12}$$

The final tap of $\hat{\mathbf{Q}}_u(n)$ is called the DC tap, and once converged, ensures that $E\left[u(n-V-d)\right] = E\left[\hat{u}(n-V-d)\right]$. The DC tap is further discussed in Section 4.1.2.

3.3.3 Normalized Least Mean Square Adaptive Mechanism

Since the identification schemes presented in Sections 3.1.2, 3.2.1, and 3.2.2 will receive no further treatment, the subscript u on $\mathbf{Q}_{u,0}$ and $\hat{\mathbf{Q}}_u(n)$ are henceforth dropped.

Unlike $\mathbf{Q}_{y^*,0}$, $\mathbf{Q}_{u,0}$ can be estimated using the Normalized Least Mean Square algorithm [49]. At time *n*, calculate

$$u(n) = \hat{\mathbf{Q}}(n)^T \mathbf{y}^*(n+V+d \mid n) \tag{13}$$

$$\hat{u}(n-V-d) = \hat{\mathbf{Q}}(n)^T \mathbf{y}(n) \tag{14}$$

$$e(n) = e_u(n-V-d) = u(n-V-d) - \hat{u}(n-V-d) \tag{15}$$

$$\hat{\mathbf{Q}}(n+1) = \hat{\mathbf{Q}}(n) + \frac{\mu \mathbf{y}(n)}{\mathbf{y}(n)^T \mathbf{y}(n)} e_u(n-V-d) \tag{16}$$

The scalar *d* is the minimum plant delay, *V* is an operator chosen (non-negative) inversion polynomial delay (discussed at length in Section 3.1.1), and μ is the adaptive gain chosen such that $0 < \mu < 2$. The constant y_{DC} is operator-chosen, appended to the delay-chain values of $\{y\}$ in (11) so that the final tap of $\hat{\mathbf{Q}}(n)$ becomes a DC tap $\hat{q}_{DC}(n)$ (discussed further in Section 4.1.2), $\hat{\mathbf{Q}}(n) \equiv \left[\hat{\mathbf{Q}}_{lin}(n)^T, \hat{q}_{DC}(n) \right]^T$.

Defining the error parameter vector $\tilde{\mathbf{Q}}(n) \equiv \hat{\mathbf{Q}}(n) - \mathbf{Q}_0$, Section 4 shows that if $0 < \mu < 2$ and certain other assumptions are met, then $\tilde{\mathbf{Q}}(n)$ converges to the zero vector. Section 4 also addresses global stability.

3.3.4 Complete Control Architecture

Figure 6 shows the complete control architecture. The Identification section uses NLMS adaptation to determine $\hat{\mathbf{Q}}(n)$ (shown with $\hat{q}_{DC}(n)$ separated from the remaining linear taps, $\hat{\mathbf{Q}}_{lin}(n)$, and with $y_{DC} = 1$) by creating estimate $\hat{u}(n-V-d)$ using (14). $\hat{\mathbf{Q}}(n)$ is copied into the Controller, which produces $u(n)$ from the set point $y^*(n+V+d \mid n)$. The Plant is represented by (3).

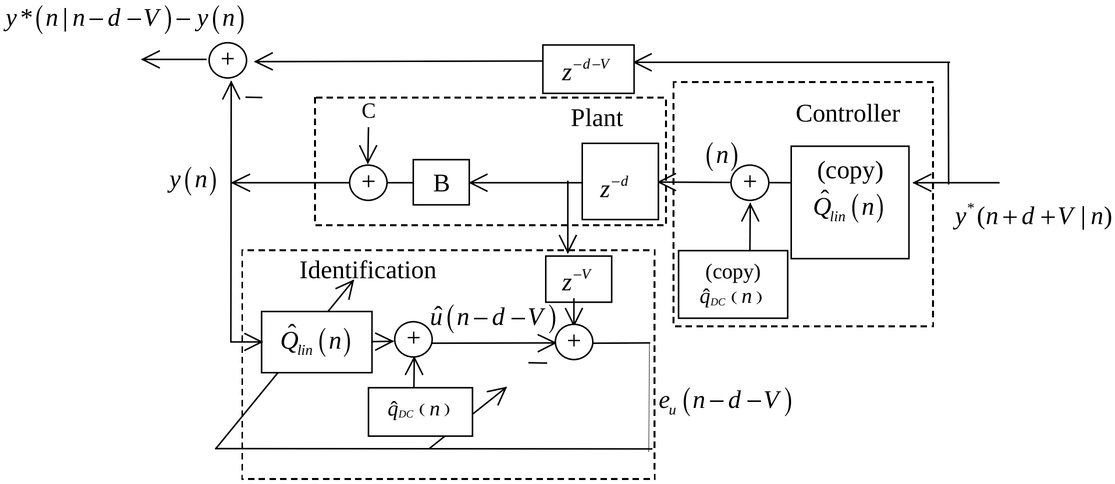


Figure 6 Complete Control Architecture ($y_{DC}=1$)

3.4 Summary

After reviewing the ATM ABR congestion control plant in Section 2, four control mechanisms based on adaptive linear control theory are presented. The first, described in Section 3.1.2, is a previously published controller that approximates the inverse of the moving average (MA) plant with a BIBO stable FIR filter. This approach is an indirect controller and is judged to be unnecessarily computationally complex, yet it inspires the ensuing three controllers. The first two of these three are impractical choices, as discussed in Section 3.2, but provide intuition to the proposed controller in Section 3.3. The selected controller can be viewed as a direct adaptive controller based on the controller of Section 3.1.2. This controller can employ a NLMS adaptation mechanism.

Convergence, fairness and other issues pertaining to this control architecture are discussed in Sections 4 and 5. Other algorithm modifications to effect improved performance are given in Section 6.3. Comparisons to other schemes are found in 7.

4 Convergence and Stability

In this section, the convergence and stability properties of the controller proposed in Section 3.3 are examined. Section 4 contains two separate yet complimentary proofs. Each proof has its own set of assumptions. The first proof, in Section 4.1, is summarized by Theorem 1 and Theorem 2. The second proof, in Section 4.2, is summarized by Theorem 3. Each of these two proofs demonstrates desirable qualities of the controller presented in Section 3.3.

The system operates optimally if, at time n, $\hat{\mathbf{Q}}(n)$ produces the control signal $u(n)$ that minimizes $E\left[\left(y(n+d+V)-y^*(n+d+V|n)\right)^2\right]$. This occurs if $\hat{\mathbf{Q}}(n)$ converges to its stationary, minimum mean square error optimal value, $\mathbf{Q}_0$ (defined by (17)). Loosely stated, the following proofs demonstrate that $\hat{\mathbf{Q}}(n)$ converges to $\mathbf{Q}_0$, and that $y(n)$ appropriately emulates $y^*(n|n-d-V)$.

The two proofs are necessitated by lack of perfect modeling afforded by attempting to invert one FIR filter with another. When perfect modeling is assumed, as it is in the second proof, strong results are possible. This perfect modeling assumption well approximates practical situations in a great many cases, e.g. when the FIR plant has no roots on the unit circle and the FIR controller has a sufficiently large number of taps. However, as this approximation becomes less accurate, e.g. when the FIR plant has roots near or on the unit-circle or when the FIR controller has an insufficient number of taps, it is important to know that the controller will converge to a desirable filter. The first proof offers assurances that the controller will converge to a minimum mean square error solution. Therefore the two proofs should be viewed as complimentary. Of course it would be possible to combine these two proofs by combining their assumptions and results. However, this would obscure important understanding of the system under study.

4.1 A Proof of Controller Convergence – the Inaccurate Plant Inversion Case

This section shows that the controller parameters converge to their optimal values in both the mean and the mean square sense. Further, the form of the controller ensures stability.

The convergence analysis in this section is based in part on a proof of convergence for the NLMS algorithm by Tarrab and Feuer [47]. However different assumptions are made; see Section 4.1.1. Most notably, this proof does not require zero-mean signals, which are required by [47]. Further, the filter $\hat{\mathbf{Q}}(n)$ includes a DC tap (drift tap) that ensures that the mean of the estimated signal equals the mean of the signal being estimated. To improve readability, many technical details have been omitted. A full, continuous presentation can be found in [33].

4.1.1 Assumptions

For notational convenience, define a vector of current and past $\{y\}$ (identical to (11) except for the DC term) $\mathbb{Y}(n) \equiv \left[y(n), y(n-1), ..., y(n-dQ)\right]^T$.

Here are the assumptions made throughout Section 4.1:

Assumption 1 $u(n)$ is Gaussian.

Assumption 2 $\mathbf{y}(n)$ and $\tilde{\mathbf{Q}}(n)$ are independent. Also $u(n-V-d)$ and $\tilde{\mathbf{Q}}(n)$ are independent.

Assumption 3 The auto-covariance matrix, $\boldsymbol{\sigma}^2 \equiv E\left[\left(\mathbb{Y}(n)-E\left[\mathbb{Y}(n)\right]\right)\left(\mathbb{Y}(n)-E\left[\mathbb{Y}(n)\right]\right)^T\right]$, is full rank.

Assumption 4 $\alpha_0 \leq \left\|\mathbf{y}(n)\right\|^2, \alpha_0 > 0$

Assumption 4 ensures that finite adaptation adjustments in (16) will occur. In implementation, it is common to impose Assumption 4 by simply skipping the adaptation of (16) unless Assumption 4 is satisfied.

Assumption 3 is a sufficient excitation condition. From Assumption 3, it follows that $E\left[\mathbf{y}(n)\mathbf{y}(n)^T\right]$ is full rank (see [34], [33]), which ensures that the plant will be fully identified, allowing the discovery of a unique $\mathbf{Q}_0$; see (17).

Assumption 2 is an often-made assumption in convergence proofs of adaptive filters. If $\{y\}$ were white (an assumption generally not made, but considered here only for illustration), both $u(n-V-d)$ and $\mathbf{y}(n)$ would be independent of $\tilde{\mathbf{Q}}(n-dQ)$, and if μ <<1, then $\tilde{\mathbf{Q}}(n) \approx \tilde{\mathbf{Q}}(n-dQ)$. More generally, signals $u(n-V-d)$ and $\mathbf{y}(n)$ make their most significant contribution to $\tilde{\mathbf{Q}}(n+1)$. For ease of computation, the much smaller impact on $\tilde{\mathbf{Q}}(n)$ is ignored.

Assumption 2 replaces an assumption made in [47]. The proof of [47] assumes that the excitation signal is Gaussian and further, if $\mathbf{y}(n)$ is a vector of the excitation signal at time *n*, $E\left[\mathbf{y}(n)\mathbf{y}(m)\right] = \mathbf{0}$ for $n \neq m$, even if $m = n+1$. This assumption tends not to be even approximately true if $\{y\}$ is the output of an FIR filter, thus it is replaced with Assumption 2.

Assumption 1 ensures that $u(n-d)$ and $\mathbf{y}(n)$ are jointly Gaussian. It is significant to note that [47] further requires $\mathbf{y}(n)$ to be zero-mean. No such assumption is made here. Broadening [47] beyond the zero-mean case is the primary contribution of this proof.

4.1.2 DC Identification

Consider the Identification section of Figure 6 ((14) - (16)), redrawn as Figure 7.

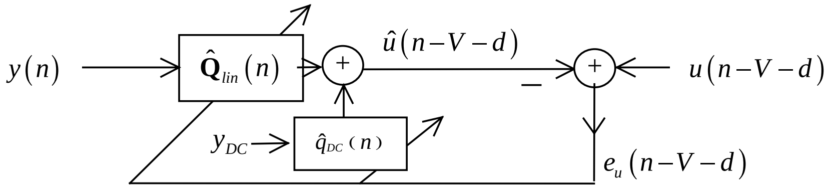


Figure 7 Adaptive System for Calculating $\hat{\mathbf{Q}}(n)$

From Figure 7, the optimal solution $\mathbf{Q}_0$ for the adaptive coefficients $\hat{\mathbf{Q}}(n)$ is defined as $\mathbf{Q}_0 \equiv \arg \min_{\hat{\mathbf{Q}}} \left\{ e(n)^2 \right\}$. Defining $\mathbf{R} \equiv E\left[\mathbf{y}(n)\mathbf{y}(n)^T\right]$, $\mathbf{Q}_0$ is known to be [49]

$$\mathbf{Q}_0 = \mathbf{R}^{-1} E\left[\mathbf{y}(n) u(n-V-d)\right]. \tag{17}$$

This solution exists and is unique since $\mathbf{R}$ is full rank ([34],[33]).

Now consider a different yet similar scheme where the DC tap is not employed but the means of y and u are removed, as in Figure 8.

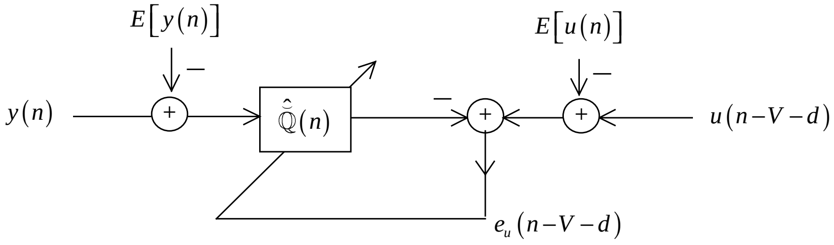


Figure 8 Adaptive System with Means Explicitly Removed

Define $\breve{\mathbb{Y}}(n) \equiv \mathbb{Y}(n) - E[\mathbb{Y}(n)]$ and $\breve{u}(n) \equiv u(n) - E[u(n)]$. The optimal solution for the adaptive coefficients $\hat{\breve{\mathbb{Q}}}(n)$ is $\breve{\mathbb{Q}}_0$, which solves

$$E\left[\breve{\mathbb{Y}}(n)\breve{\mathbb{Y}}(n)^T\right]\breve{\mathbb{Q}}_0 = E\left[\breve{\mathbb{Y}}(n)\breve{u}(n-V-d)\right]. \tag{18}$$

The solutions $\breve{\mathbb{Q}}_0$ and $\mathbf{Q}_0$ are closely related, as shown by Lemma 1.

Lemma 1 The unique solution of (17) is

$$\mathbf{Q}_0 = \left[\breve{\mathbb{Q}}_0{}^T, \frac{-E[y(n)]\sum_i \breve{\mathbb{Q}}_{0,i} + E[u(n-V-d)]}{y_{DC}}\right]^T.$$

Proof: As noted before, $\mathbf{Q}_0$ is unique due to $\mathbf{R}$ being full-rank [33]. Describe

$$\left[\breve{\mathbb{Q}}_0{}^T, \frac{-E[y(n)]\sum_i \breve{\mathbb{Q}}_{0,i} + E[u(n-V-d)]}{y_{DC}}\right]^T \tag{19}$$

as the *proposed solution* to (17). Directly substituting the proposed solution into (17) verifies that it is indeed a solution ([34], [33]), and thus the unique solution, completing the proof.

Lemma 1 demonstrates that by using a DC tap in the adaptive estimator as in Figure 7, the optimal solution for $\mathbf{Q}_{lin,0}$ is equivalent to $\breve{\mathbb{Q}}_0$. To gain intuition, consider that for a linear estimator not using a DC tap, the optimal solution would create the best possible match between the frequency spectrum of the desired signal (u) and the spectrum of the estimated signal ($\hat{u}$), given the regressor (y). This match would consider all frequencies, including DC. A DC tap, if included, can only affect the spectrum of the estimated signal at DC, but by doing so, allows the linear taps to ignore DC in their spectrum matching, as if there was no DC content in either regressor signal (y) or the desired signal (u).

The DC tap creates an additional similarity between Figure 7 and Figure 8 – a zero-mean optimal error. By defining the optimal error $e^*(n) \equiv u(n-d-V) - \mathbf{Q}_0(n)^T \mathbf{y}(n)$, with (19), it is easy to show ([34], [33]) that

$$E[e^*(n)] = 0. \tag{20}$$

4.1.3 Other Notation

Now that the control and estimation methods have been described, what remains is to show convergence. Before proceeding with the proofs, notation needs to be introduced. For matrix $\mathbf{R}$, let the matrices of ortho-normalized eigenvectors and eigenvalues of be $\mathbf{W}^T$ and $\mathbf{\Lambda}$ respectively, where $\mathbf{\Lambda} = \mathrm{diag}(\lambda_i),\ i = 1, \cdots, (dQ+1)$,

$$\mathbf{W}^T\mathbf{W} = \mathbf{I},\ \mathbf{W}\mathbf{R}\mathbf{W}^T = \mathbf{\Lambda} \tag{21}$$

Because **R** is full-rank, **W** is full-rank. Now define a linear transformation of the random vector $\mathbf{y}(n)$ as follows.

$$\boldsymbol{\psi}(n) \equiv \mathbf{W}\mathbf{y}(n),\ \mathbf{W}^T\boldsymbol{\psi}(n) = \mathbf{W}^T\mathbf{W}\mathbf{y}(n) = \mathbf{y}(n) \tag{22}$$

$$\mathbf{L}(n) \equiv \mathbf{W}\tilde{\mathbf{Q}}(n),\ \mathbf{W}^T\mathbf{L}(n) = \tilde{\mathbf{Q}}(n) \tag{23}$$

$$E\left[\boldsymbol{\psi}(n)\boldsymbol{\psi}(n)^T\right] = \mathbf{\Lambda} \tag{24}$$

Substituting (14) and (15) into (16), pre-multiplying by **W** and adding and subtracting a term, then subtracting $\mathbf{W}\mathbf{Q}_0$ from both sides produces

$$\mathbf{L}(n+1) = \left(I - \mu\frac{\boldsymbol{\psi}(n)\boldsymbol{\psi}(n)^T}{\boldsymbol{\psi}(n)^T\boldsymbol{\psi}(n)}\right)\mathbf{L}(n) + \frac{\mu\ \boldsymbol{\psi}(n)e^*(n)}{\boldsymbol{\psi}(n)^T\boldsymbol{\psi}(n)} \tag{25}$$

and

$$\begin{aligned}
\mathbf{L}(n+1)\mathbf{L}(n+1)^T &= \mathbf{L}(n)\mathbf{L}(n)^T - \mu\frac{\boldsymbol{\psi}(n)\boldsymbol{\psi}(n)^T}{\boldsymbol{\psi}(n)^T\boldsymbol{\psi}(n)}\mathbf{L}(n)\mathbf{L}(n)^T \\
&\quad -\mu\mathbf{L}(n)\mathbf{L}(n)^T\frac{\boldsymbol{\psi}(n)\boldsymbol{\psi}(n)^T}{\boldsymbol{\psi}(n)^T\boldsymbol{\psi}(n)} + \mu^2\frac{\boldsymbol{\psi}(n)\boldsymbol{\psi}(n)^T}{\boldsymbol{\psi}(n)^T\boldsymbol{\psi}(n)}\mathbf{L}(n)\mathbf{L}(n)^T\frac{\boldsymbol{\psi}(n)\boldsymbol{\psi}(n)^T}{\boldsymbol{\psi}(n)^T\boldsymbol{\psi}(n)} \\
&\quad +\mu\frac{e^*(n)}{\boldsymbol{\psi}(n)^T\boldsymbol{\psi}(n)}\left[\mathbf{L}(n)\boldsymbol{\psi}(n)^T + \boldsymbol{\psi}(n)\mathbf{L}(n)^T\right] \\
&\quad -\mu\frac{e^*(n)}{\boldsymbol{\psi}(n)^T\boldsymbol{\psi}(n)}\left[\frac{\boldsymbol{\psi}(n)\boldsymbol{\psi}(n)^T}{\boldsymbol{\psi}(n)^T\boldsymbol{\psi}(n)}\mathbf{L}(n)\boldsymbol{\psi}(n)^T + \boldsymbol{\psi}(n)\mathbf{L}(n)^T\frac{\boldsymbol{\psi}(n)\boldsymbol{\psi}(n)^T}{\boldsymbol{\psi}(n)^T\boldsymbol{\psi}(n)}\right] \\
&\quad +\frac{\mu^2\boldsymbol{\psi}(n)\boldsymbol{\psi}(n)^T\left(e^*(n)\right)^2}{\left(\boldsymbol{\psi}(n)^T\boldsymbol{\psi}(n)\right)^2}
\end{aligned} \tag{26}$$

The following notations is used extensively:

$$\mathbf{A} \equiv E\left[\frac{\boldsymbol{\psi}(n)\boldsymbol{\psi}(n)^T}{\boldsymbol{\psi}(n)^T\boldsymbol{\psi}(n)}\right],\ \mathbf{C}(n) \equiv E\left[\mathbf{L}(n)\mathbf{L}(n)^T\right],$$

$$\mathbf{D}(n) \equiv E\left[\frac{\boldsymbol{\psi}(n)\boldsymbol{\psi}(n)^T}{\boldsymbol{\psi}(n)^T\,\boldsymbol{\psi}(n)}\ \mathbf{C}(n)\ \frac{\boldsymbol{\psi}(n)\boldsymbol{\psi}(n)^T}{\boldsymbol{\psi}(n)^T\,\boldsymbol{\psi}(n)}\right],\ \mathbf{H} \equiv E\left[\frac{\boldsymbol{\psi}(n)\boldsymbol{\psi}(n)^T}{\left(\boldsymbol{\psi}(n)^T\,\boldsymbol{\psi}(n)\right)^2}\right]$$

4.1.4 Parameter Convergence in the Mean

In this section it is shown that $\lim_{n\to\infty} E\left[\mathbf{L}(n)\right] = \mathbf{0}$, and thus, by (23), $\lim_{n\to\infty} E\left[\tilde{\mathbf{Q}}(n)\right] = \mathbf{0}$.

Note a few key independencies. From Assumption 2, and the fact that **W** provides a one-to-one mapping, Section 5.4 of [46] shows that $\boldsymbol{\psi}(n)$ and $\mathbf{L}(n)$ are independent. Similarly $u(n-d-V)$ and $\mathbf{L}(n)$ are independent.

Note that $e^*(n)$ and $\mathbf{y}(n)$ are jointly Gaussian, and uncorrelated ($E\left[\mathbf{y}(n)e^*(n)\right] = \boldsymbol{\rho} - \mathbf{R}\ \mathbf{R}^{-1}\ \boldsymbol{\rho} = \mathbf{0}$). The auto-covariance matrix of two jointly-Gaussian, uncorrelated random variables, where at least one is zero-mean ($e^*(n)$), is diagonal. Therefore, $\mathbf{y}(n)$ and $e^*(n)$ are independent. By a similar argument, $\boldsymbol{\psi}(n)$ and $e^*(n)$ are also independent. The auto-covariance $\boldsymbol{\psi}(n)$ is diagonal ((21) gives $\mathbf{W}_i^T\mathbf{W}_j = 0$ where $\mathbf{W}_j$ is the *j*'th row of **W**), thus the elements of $\boldsymbol{\psi}(n)$ are independent.

Lemma 2 $\mathbf{A}$ and $\mathbf{H}$ are diagonal matrices (See Appendix for proof).

Lemma 3 $0 < \alpha_1 < A_{ii} \le 1$ (See Appendix for proof).

Here the main result of Section 4.1.4:

Theorem 1 Given Assumption 1 – Assumption 4 and $0 < \mu < 2$, $\lim_{n\to\infty} E\left[\tilde{\mathbf{Q}}(n)\right] = \mathbf{0}$.

Proof: From (20),

$$E\left[\frac{\mu\,\boldsymbol{\psi}(n)e^*(n)}{\boldsymbol{\psi}(n)^T\,\boldsymbol{\psi}(n)}\right] = \mu\ E\left[\frac{\boldsymbol{\psi}(n)}{\boldsymbol{\psi}(n)^T\,\boldsymbol{\psi}(n)}\right] E\left[e^*(n)\right] = 0$$

From (25), since $\boldsymbol{\psi}(n)$ and $\mathbf{L}(n)$ are independent, $E\left[\mathbf{L}(n+1)\right] = (\mathbf{I} - \mu\ \mathbf{A})\ E\left[\mathbf{L}(n)\right]$

From Lemma 2, the linear system completely decouples:

$$E\left[L_i(n+1)\right] = (1-\mu A_{ii})\,E\left[L_i(n)\right],\ i = 1,\ \ldots,\ dQ+1$$

From Lemma 3 and if $0 < \mu < 2$, then $\left|(1-\mu A_{ii})\right| < 1$, $i = 1, \ldots, dQ+1$. Thus, $\lim_{n\to\infty} E\left[\mathbf{L}(n)\right] = \mathbf{0}$. Equation (23) gives $\lim_{n\to\infty} E\left[\tilde{\mathbf{Q}}(n)\right] = \mathbf{0}$, thus completing the proof.

Theorem 1 states that given Assumption 1 – Assumption 4, if μ is bounded as $0 < \mu < 2$, then an NLMS adaptive system of estimating $\hat{\mathbf{Q}}$, given by (14) - (16), converges in the mean to the ideal $\mathbf{Q}_0$, given by Lemma 1. Theorem 1 is distinctive from the related proof of [47] in that it uses (20), instead of a zero-mean assumption for $\mathbf{y}(n)$**,** to eliminate the expectation of the second term of (25).

4.1.5 Parameter Convergence in the Mean Square

Given Theorem 1, a statement bounding the variance of $\tilde{\mathbf{Q}}(n)$ would give much additional credibility to the proposed controller, and is the goal of this section.

The proof in [47] relies heavily on a zero-mean assumption on $\mathbf{y}(n)$, an assumption not made here. However, Lemma 4 shows that almost all of the terms of $\boldsymbol{\psi}(n)$ are zero-mean. Because of this, a strategy similar to that of [47] is adopted.

Lemma 4 Given the independence of the terms of $\boldsymbol{\psi}(n)$, (24) is the necessary and sufficient condition that no less than dQ of the $dQ+1$ elements of $\boldsymbol{\psi}(n)$ are zero mean (See Appendix for proof).

The element of $\boldsymbol{\psi}(n)$ that generally has non-zero mean is notated $\psi_\zeta(n)$, i.e.

$$E\left[\psi_i(n)\right]=0, i=1,\dots,dQ+1,\ i\neq\zeta\ , \tag{27}$$

If $E\left[y(n)\right]=0$, $E\left[\psi_\zeta(n)\right]=0$.

Lemma 5 The expectation of the fifth through eighth term of (26) is zero.

Proof: Can be shown ([30], [34], [33]) term by term using (20) and independencies noted above.

With Lemma 5, the expectation of (26) is equivalent to

$$\mathbf{C}(n+1)=\mathbf{C}(n)-\mu\left(\mathbf{AC}(n)+\mathbf{C}(n)\mathbf{A}\right)+\mu^2\mathbf{D}(n)+\mu^2\varepsilon^*\mathbf{H} \tag{28}$$

where ε^* is the minimal mean-square error $\varepsilon^*\equiv E\left[\left(e^*(n)\right)^2\right]$.

Since (28) is a discrete, linear, time-invariant difference equation, its convergence is guaranteed if its homogeneous part is asymptotically stable, as this implies BIBO stability, and if its forcing term $\mu^2\varepsilon^*\mathbf{H}$ is bounded. After Lemma 6 is presented, these are shown in turn.

Lemma 6 Since the elements of $\boldsymbol{\psi}(n)$ are Gaussian and independent, given (27), the expectation of an Odd Function[6] of $\boldsymbol{\psi}(n)$ around $\psi_i(n)$ for $i\neq\zeta$ is zero (Proof: See [34], [33]. Also mentioned without proof in [47].).

The row *i*, column *j* element of $\mathbf{D}(n)$, $D_{ij}(n)$, can be computed as

$$D_{ij}(n)=2G_{ij}C_{ij},\ i\neq j \tag{29}$$

[6] A function of $\boldsymbol{\psi}(n)$, $\Gamma_i(\boldsymbol{\psi}(n))$, is defined as an *Odd Function around* $\psi_i(n)$ if $\Gamma_i(\psi_1(n),\psi_2(n),\dots,\psi_i(n),\dots,\psi_{dQ+1}(n))=-\Gamma_i(\psi_1(n),\psi_2(n),\dots,-\psi_i(n),\dots,\psi_{dQ+1}(n))$

$$D_{ii}(n) = \sum_{J=1}^{dQ+1} G_{iJ} C_{JJ} \tag{30}$$

where **G** is defined as

$$G_{ij} \equiv E\left[\frac{\left(\psi_i(n)\right)^2 \left(\psi_j(n)\right)^2}{\left(\boldsymbol{\psi}(n)^T \boldsymbol{\psi}(n)\right)^2}\right]. \tag{31}$$

By direct substitution,

$$D_{ij}(n) = \sum_{K=1}^{dQ+1} \sum_{J=1}^{dQ+1} E\left[\frac{\psi_i(n)\psi_j(n)\psi_J(n)\psi_K(n)}{\left(\boldsymbol{\psi}(n)^T \boldsymbol{\psi}(n)\right)^2}\right] C_{JK}\,. \tag{32}$$

Since the elements of $\boldsymbol{\psi}(n)$ are independent and all but one are zero-mean, the numerator of the expectation in (32) will equal zero in many cases. Using Lemma 4 and Lemma 6, References [30] and [34] demonstrate that (32) equals (29) if $i \neq j$, and (32) equal to (30) if $i = j$.

Having shown (28), (29), and (30), the techniques used for remainder of the mean-square proof are nearly identical to those presented in [47], and thus will only be outlined here (see [34] and [33] for details). Off-diagonal elements of $\mathbf{C}(n)$ are treated separately from the diagonal elements. The off-diagonal term of (28) is

$$C_{ij}(n+1) = \gamma_{ij} C_{ij}(n), i \neq j \tag{33}$$

$$\gamma_{ij} = 1 - \mu\left(A_{ii} + A_{jj}\right) + 2\mu^2 G_{ij}, i \neq j\,, \tag{34}$$

with $\left|\gamma_{ij}\right| < 1$ ([34], [33]), and thus (33) goes to zero as *n* approaches infinity.

Focussing now on the diagonal entries of $\mathbf{C}(n)$, define a vector of the diagonal entries of $\mathbf{C}(n)$, $\boldsymbol{\Omega}(n) \equiv \left[C_{11}(n), C_{22}(n), \ldots, C_{(dQ+1)(dQ+1)}(n)\right]^T$. From (28),

$$\boldsymbol{\Omega}(n+1) = \mathbf{F}\,\boldsymbol{\Omega}(n) + \mu^2 \varepsilon^* \mathbf{h}\,, \tag{35}$$

$$\mathbf{F} = \text{diag}\left\{\left(1 - 2\mu \mathbf{A}_{ii}\right)\right\} + \mu^2 \mathbf{G}\,,\ \mathbf{h} \equiv \left[\mathbf{H}_{11}, \mathbf{H}_{22}, \ldots, \mathbf{H}_{(dQ+1)}\right]^T. \tag{36}$$

It can be shown ([47], [33], [34]) that (35) is BIBO stable. Assuming $0 < \mu < 2$, the forcing term of (35) is bounded, i.e.$\left|\mu^2 \varepsilon^* \mathbf{H}_{ii}\right| < \alpha_2, \alpha_2 < \infty$, $\left|\mu^2 \varepsilon^* \mathbf{H}_{ij}\right| = 0, i \neq j$, since $\left|H_{ii}\right| \leq 1/\left(dQ-2\right)\lambda_{\min}$ ([34], [33]). Equations (33) and (35) show that each element of $\mathbf{C}(n)$ is bounded at each *n*, that the off-diagonal elements of $\mathbf{C}(n)$ converge to zero, and that the diagonal elements also converge, $\lim_{n\to\infty} \boldsymbol{\Omega}(n) = \mu^2 \varepsilon^* \left(\mathbf{I} - \mathbf{F}\right)^{-1} \mathbf{h}$. This is formalized in Theorem 2, the main result of Section 4.1.5:

Theorem 2 Given Assumption 1 - Assumption 4 and $0 < \mu < 2$,

$$\lim_{n\to\infty} E\left[\tilde{\mathbf{Q}}(n)\tilde{\mathbf{Q}}(n)^T\right] = \mu^2\varepsilon^* \mathbf{W}^T (\mathbf{I}-\mathbf{F})^{-1}\mathbf{HW}.$$

Proof: Linear time-invariant system (35) is BIBO stable ([47], [33], [34]). Its input signal is bounded, thus $\lim_{n\to\infty}\mathbf{\Omega}(n) = (\mathbf{I}-\mathbf{F})^{-1}\mu^2\varepsilon^*\mathbf{h}$. Then with (33) and $|\gamma_{ij}| < 1$, $\lim_{n\to\infty}\mathbf{C}(n) = diag\left\{(\mathbf{I}-\mathbf{F})^{-1}\mu^2\varepsilon^*\mathbf{h}\right\}$. The definitions of $\mathbf{C}(n)$ and $\mathbf{L}(n)$ give the final result, concluding the proof.

The key contribution of Section 4.1.5 comes from showing (28) - (30) without requiring $E[\mathbf{y}(n)] = \mathbf{0}$. Expressions (28) - (30) exist in [47], but required $E[\mathbf{y}(n)] = \mathbf{0}$. By demonstrating (28) - (30) without requiring $E[\mathbf{y}(n)] = \mathbf{0}$, the results of [47] are significantly extended.

4.1.6 Global Stability

Global Stability has been built into the control structure. Both plant (3) and controller (13) are FIR filters. The controller simply conditions the set point $\{y^*\}$; all other control is open loop. The parameters of the plant are obviously bounded. The parameters of the controller are random variables that have been shown to have mean square values that are finite for all *n* and converge, implying a bounded mean-square gain for the controller. From an implementation view, the controller parameters can be kept bounded at each time *n* by a simple limiter after the adaptation of (16). The FIR structure of the controller then guarantees BIBO stability for the modified system. A stronger stability proof is given in Section 4.2.2.

4.1.7 Discussion

Consider the pathological case where $\mathbf{B} = \mathbf{0}$, $y(n) = C < y^*(n \mid n-d-V)$. In this case, the port can be described as a *non-controlling port*, since there are no VCs responsive to its explicit rates. This under-utilized port must only carry VCs that are bottlenecked by other ports. Let the largest bandwidth of any bottlenecked VC passing through this port be u'. Ideally this non-controlling port will converge such that its explicit rate u is $\geq u'$. If it does, all VCs passing through the port will ignore its explicit rate, making the non-controlling port essentially transparent to the network. From (15), to minimize $E[e(n)]$, $\hat{\mathbf{Q}}(n)$ must converge to $\left[\mathbf{0}^T, \tilde{q}_{DC}\right]^T$, where $\tilde{q}_{DC}$ is a constant. As it does, $e(n)$ converges to zero and $\hat{\mathbf{Q}}(n)$ and $u(n) = \tilde{q}_{DC}y_{DC}$ become constant.

The explicit rate $u(n) = \tilde{q}_{DC}y_{DC}$ cannot converge to a value $< u'$. If it did, one (or more) VC(s) would become responsive to this explicit rate. The adaptive filter $\hat{\mathbf{Q}}(n)$ would then identify this responsiveness and the port would attempt to increase $y(n)$ to meet $y^*(n)$ by increasing $u(n)$. This would continue until $u(n) \geq u'$, when the responsive VC(s) would again become non-responsive. Therefore, the explicit rate converges to $u(n) = \tilde{q}_{DC}y_{DC} \geq u'$. In short, a port with $\mathbf{B} = \mathbf{0}$, $y(n) = C < y^*(n \mid n-d-V)$ is appropriately described as a non-controlling port, as its constant explicit rate is ignored by all VCs.

This concludes the first proof of Section 4. Theorem 1 and Theorem 2 prove that the controller parameters converge to their optimal values in the mean and mean square sense. It is observed that the controller runs essentially open loop, only conditioning the set point, thus global convergence is assured. Unlike the proof presented next in Section 4.2, no assumption about the FIR invertibility of the plant is made in Section 4.1.

4.2 A Proof of Controller Convergence and Global Stability– the Accurate Plant Inversion Case

Section 4.2 contains the second of the proofs of Section 4. The plant and controller used in Section 4.2 are identical to that described in Sections 2 and 3.3.3 except the NLMS update equation (16) is replaced by

$$\hat{\mathbf{Q}}(n+1) = \hat{\mathbf{Q}}(n) + \frac{\mu \mathbf{y}(n)}{\delta + \mathbf{y}(n)^T \mathbf{y}(n)} e(n),\ \delta > 0 . \tag{37}$$

4.2.1 All-Pole Plant Approximation and other Assumptions

As discussed in Section 3.1.1, the possibility of a Non-Minimum Phase (NMP) $B(z^{-1})$ encourages the use of an all-pole plant approximation (with a corresponding all-zero controller). Assume that $B(z^{-1})$ has no zeros on the unit circle and that $B(z^{-1}) = B^+(z^{-1}) B^-(z^{-1})$, with $B^+(z^{-1})$ having zeros exclusively inside the unit circle and $B^-(z^{-1})$ having zeros exclusively outside the unit circle. By long division, $N^+(z^{-1}) = \left(B^+(z^{-1})\right)^{-1} = n_0^+ + n_1^+ z^{-1} + \ldots$, with $\left|n_i^+\right| \ge \xi_1 \left|n_{i+1}^+\right|$, $i \ge 0$, $0 \le \xi_1 < 1$ (a causal filter) and $N^-(z^{-1}) = \left(B^-(z^{-1})\right)^{-1} = n_\gamma^- z^\gamma + n_{\gamma+1}^- z^{\gamma+1} + \ldots$, with $\left|n_i^-\right| \ge \xi_2 \left|n_{i+1}^-\right|$, $i \ge \gamma$, $\gamma > 0$, $0 \le \xi_2 < 1$ (a non-causal filter). Since the coefficients are decreasing exponentially, with enough taps, N^+ and N^- can be approximated by truncated FIR filters, $\bar{N}^+(z^{-1}) = n_0^+ + n_1^+ z^{-1} + \ldots + n_{dN^+}^+ z^{-dN^+}$ and $\bar{N}^-(z^{-1}) = n_\gamma^- + n_{\gamma+1}^- z^{\gamma+1} + \ldots + n_V^- z^V$. Therefore, an FIR

$$Q(z^{-1}) \equiv z^{-V} \bar{N}^+(z^{-1}) \bar{N}^-(z^{-1}) + C' \tag{38}$$

can well approximate $z^{-V}\left(B(z^{-1}) + C\right)^{-1}$. This approximation is formalized by the following Assumption used throughout Section 4.2.

Assumption 5 $B(z^{-1})$ has no zero on $|z| = 1$ and the plant (3) is equivalently expressed (using (11)) as

$$\mathbf{Q}\mathbf{y}(n) = u(n - d - V), \tag{39}$$

where $\mathbf{Q} = \left[q_0, q_1, \ldots, q_{dQ}, q_{DC}\right]$.

In addition to the all-pole assumption given by Assumption 5, here are the other assumptions made throughout Section 4.2:

Assumption 6 $\left\|\hat{\mathbf{Q}}_{lin}(n)\right\| > 0$ for all n.

Assumption 7 At each n, $z = e^{-j\omega'}$ is not a root of $\hat{\mathbf{Q}}_{lin}\left(z^{-1}\right) = 0$ if $y*(n)$ contains the frequency ω'.

Assumption 6 requires at least one of the linear taps of $\hat{\mathbf{Q}}_{lin}(n)$ to be non-zero. Note that this excludes non-controlling ports.

Assumption 7 states that the controller cannot null a frequency present in the set-point signal. Such an occurrence would make it impossible for the plant output to match the set-point at frequency ω'. Intuitively, the controller should not place a zero on the unit circle since the plant has no marginally stable poles.

Both of these assumptions prevent pathological cases; neither pose significant limitations in practice.

4.2.2 Convergence and Global Stability

In this section, some of the more limiting assumptions of Section 4.1 are lifted. In their place, Assumption 5 is made, as well as the minor Assumption 6 and Assumption 7. This leads to a cleaner proof with stronger global stability results.

4.2.2.1 Proof of Convergence and Global Stability

The update equation (39) is identical to Equation (3.3.19) of [48]. From (15), (39) and (14), $e(n) = -\tilde{\mathbf{Q}}(n)^T \mathbf{y}(n)$, and from Lemma 3.3.2 of [48],

$$\lim_{n\to\infty} \frac{e(n)}{\left(\delta + \mathbf{y}(n)^T \mathbf{y}(n)\right)^{1/2}} = 0, \qquad \lim_{n\to\infty} \left\| \hat{\mathbf{Q}}(n-k) - \hat{\mathbf{Q}}(n) \right\| = 0 \text{ for any finite } k. \tag{40}$$

From (15), (13), and (14),

$$\begin{aligned} e(n) &= \hat{\mathbf{Q}}(n-d-V)^T \mathbf{y}*(n \mid n-d-V) - \hat{\mathbf{Q}}(n)^T \mathbf{y}(n) \\ &= \left(\hat{\mathbf{Q}}(n-d-V) - \hat{\mathbf{Q}}(n)\right)^T \mathbf{y}*(n \mid n-d-V) + \hat{\mathbf{Q}}(n)^T \left(\mathbf{y}*(n \mid n-d-V) - \mathbf{y}(n)\right) \end{aligned} \tag{41}$$

Then from (40)

$$0 = \lim_{n\to\infty} \frac{e(n)}{\left(\delta + \mathbf{y}(n)^T \mathbf{y}(n)\right)^{1/2}} = \lim_{n\to\infty} \frac{\hat{\mathbf{Q}}(n)^T \boldsymbol{\chi}(n)}{\left(\delta + \mathbf{y}(n)^T \mathbf{y}(n)\right)^{1/2}}$$

$$\lim_{n\to\infty} \frac{\left(\hat{\mathbf{Q}}(n)^T \boldsymbol{\chi}(n)\right)^2}{\delta + \mathbf{y}(n)^T \mathbf{y}(n)} = 0 \tag{42}$$

where the set-point error is $\boldsymbol{\chi}(n) \equiv \mathbf{y}*(n \mid n-d-V) - \mathbf{y}(n)$.

To show that (42) implies that $\lim_{n\to\infty} \left(\hat{\mathbf{Q}}(n)^T \boldsymbol{\chi}(n)\right)^2 = 0$, note that $\left\|\mathbf{y}(n)\right\| \le \kappa_1 + \kappa_2 \max_{0 \le \tau \le n} \left\|\mathbf{y}(\tau)\right\|$, $0 < \kappa_1, \kappa_2 < \infty$, and since $\mathbf{y}*(n \mid n-d-V)$ is bounded, and since $\left\|\boldsymbol{\chi}(n)\right\| \ge \left\|\mathbf{y}(n)\right\| - \left\|\mathbf{y}*(n \mid n-d-V)\right\|$, together with Assumption 6,

$$\left\|\mathbf{y}(n)\right\| \leq \kappa_3 + \kappa_4 \max_{0\leq\tau\leq n}\left|\hat{\mathbf{Q}}(\tau)^T \boldsymbol{\chi}(\tau)\right|,\ 0 < \kappa_3, \kappa_4 < \infty \tag{43}$$

With (42) and (43), the Key Technical Lemma [48] asserts that

$$\left\|\mathbf{y}(n)\right\| \text{ is bounded, and } \lim_{n\to\infty}\left(\hat{\mathbf{Q}}(n)^T \boldsymbol{\chi}(n)\right)^2 = 0\,. \tag{44}$$

Note that (44) does not require use of Assumption 7. However, Assumption 7 is needed to show that $\lim_{n\to\infty}\chi(n) = 0$. As *n* approaches infinity, $\hat{\mathbf{Q}}(n)^T \boldsymbol{\chi}(n)$ can be viewed as a signal $\chi(n)$ filtered by a constant FIR filter $\hat{\mathbf{Q}}_{lin}(n)$; see (40). Assumption 7 prevents $\hat{\mathbf{Q}}_{lin}(n)$ from nulling frequencies present in $\boldsymbol{\chi}(n)$. Therefore, from (44), we have the following theorem.

Theorem 3 Given Assumption 5-Assumption 7, the plant (2), which is equivalent to (39), controlled by (13)-(16) and (37), gives $\lim_{n\to\infty}\chi(n) = 0$.

Proof: The equations of (40) give (42). The Key Technical Lemma gives (44), which with Assumption 7, gives the result. This completes the proof.

4.2.2.2 Discussion

The result above is a strong statement on global stability. Note that no a-priori assumption on the boundedness of $\left\|\mathbf{y}(n)\right\|^2$ is made, nor are any of Section 4.1.1's restrictions placed on $y^*(n)$ (aside from boundedness).

The main assumption made in this proof is Assumption 5. When this restriction is violated, e.g. $B\left(z^{-1}\right) = 1 + z^{-1}$, Theorem 1 and Theorem 2 as well as simulation experiments suggest that the control structure behaves stably. Thus the results of this section and the convergence results of Section 4.1 should be viewed as complimentary, both examining the control system of Section 3.3.3, each starting with different assumptions, and both producing desirable results.

4.3 Summary

In Section 4, the convergence and stability properties of the controller proposed in Section 3.3 are examined extensively. Section 4 contains two separate yet complimentary proofs. Theorem 1 and Theorem 2 summarizes the first proof in Section 4.1. The second proof, in Section 4.2, is summarized by Theorem 3. Each of these two proofs demonstrates desirable qualities of the controller presented in Section 3.3. Each proof starts with its own set of assumptions. The first proof focuses on the convergence of the controller parameters $\hat{\mathbf{Q}}$ to an optimal $\mathbf{Q}_0$. The second proof requires that perfect inversion of plant $\mathbf{B}$ by FIR $\hat{\mathbf{Q}}$ is a reasonable approximation to assume. Taken together, the proofs of Section 4 make a convincing case that Adaptive Approximate Inverse Control has attractive convergence and stability properties.

5 Fairness

Section 4 demonstrates that the control structure presented in Section 3 stably converges. In this section, the fairness of this converged controller is addressed. In the context of this paper, fairness refers to the distribution of ABR bandwidth $y*$ among ABR VCs. Specifically, the controller (13)-(16) is shown to be both max-min fair and fair in a dynamic sense.

5.1 Max-Min Fairness

Implicitly, max-min fairness [50] (described below) assumes static link bandwidth availability. If the bandwidth of one or more links changes, a new max-min solution results, and the network should transition to this new max-min solution. The definition for max-min fairness does not formally describe how a network in transition allocates its bandwidth, only its final allocation once the transition is completed. The notion of an *Available* Bit Rate service category suggests that ABR link bandwidths are dynamic, i.e. $y_i*(n)$ for link i changes with n. However, to discuss whether a particular ABR congestion controller converges to a max-min fair solution, it is reasonable to assume that the ABR bandwidth $y_i*(n)$ of each link remains constant, $y_i*(n) = y_i*$.

The max-min fair [50] distribution of bandwidth for a network implies that each flow has a bottlenecked link. Consider a particular bottleneck port j, the output port of Figure 1. From Section 4.1.7, it is clear that a non-controlling port does not control bandwidth allocations. Therefore, in this discussion of fairness, all ports, including j, are assumed to control at least one VC. If a network of such ports is fair, it may be augmented with non-controlling ports with no effect on bandwidth allocations.

Whether or not j's allocation of bandwidth is max-min fair, the VCs that flow through j can be separated into three groups. *Group 1* VCs consists of VCs bottlenecked by another port at a rate below j's offered explicit rate. *Group 2* VCs consists of N_u (≥ 1 since j is a controlling port) VCs bottlenecked at j that respond to j's offered explicit rate. *Group 3* VCs are constricted to a rate above j's offered explicit rate and therefore do not respond to j. For example, if a VC negotiates a Minimum Cell Rate (MCR) above j's offered explicit rate, the VC transmits at its MCR.

Consider the max-min allocation of bandwidth $y*$ of j. Identify the VCs that are in Groups 1, 2 and 3 under the max-min solution as $[\mathbf{VC}]_1$, $[\mathbf{VC}]_2$, and $[\mathbf{VC}]_3$ respectively. Under the max-min solution, identify the combined bandwidth of VCs in Groups 1 and 3 as C cells/sec. Using this notation, the max-min fair bandwidth allocation matches the total bandwidth from all three groups to the available bandwidth $y*$. It meets the bandwidth requirement of each VC in $[\mathbf{VC}]_1$ and $[\mathbf{VC}]_3$. Further, it equally divides the remaining bandwidth, $y*-C$, among the N_u VCs in $[\mathbf{VC}]_2$. This requires the congestion controller of port j to generate

$$u(n) = u_{\max-\min} = (y*-C)/N_u . \tag{45}$$

By definition, each VC in $[\mathbf{VC}]_1$ ($[\mathbf{VC}]_3$) has a bandwidth requirement less (greater) than $u_{\max-\min}$.

For the port to be max-min fair, the controller (13)-(16) must simply produce an explicit rate given by (45). Despite the lack of sufficient excitation from $y*(n)$, (40) assures that the adaptive coefficients, and therefore $u(n)$, becomes fixed in a practical sense. Theorem 3 assures that convergence occurs and that the controller generates u such that the available bandwidth is fully utilized, $y(n) = y*(n) = y*$. It is now shown that given Theorem 3 and static link bandwidth availability, u must converge to $u_{\max-\min}$. This is shown by contradiction.

Consider the case when u remains fixed at $u^- < u_{\max-\min}$ and compare the resulting bandwidth generated by each VC to the bandwidth the VC generates when $u = u_{\max-\min}$. First consider a VC m in $[\mathbf{VC}]_1$ that generates bandwidth $u_1 < u_{\max-\min}$ when $u = u_{\max-\min}$. VC m will not generate bandwidth less than $\tilde{u}_m$ (e.g. $\tilde{u}_m$ is m's MCR). When $u = u^-$, m could remain in Group 1 or join either Group 2 or 3. VC j remains in Group 1 if $\tilde{u}_m \le u_1 \le u^-$. If m remains in Group 1 when $u = u^-$, m continues to produce bandwidth u_1. VC m joins Group 2 if $\tilde{u}_m \le u^- < u_1$. In this case, m generates bandwidth u^-. VC m in $[\mathbf{VC}]_1$ joins Group 3 if $u^- < \tilde{u}_m \le u_1$. In this case, m generates bandwidth $\tilde{u}_m$.

Second consider a VC k in $[\mathbf{VC}]_2$ that generates bandwidth $u_{\max-\min}$ when $u = u_{\max-\min}$ and will not generate bandwidth less than $\tilde{u}_k$ (e.g. $\tilde{u}_k$ is k's MCR). When $u = u^-$, k could remain in Group 2 or join Group 3. It cannot join Group 1. VC k remains in Group 2 if $\tilde{u}_k \le u^-$. If k remains in Group 2, k continues to track u, i.e. k produces bandwidth u^- when $u = u^-$. From above, at least one VC in $[\mathbf{VC}]_2$ remains in Group 2 when $u = u^-$. VC k will join Group 3 if $u^- < \tilde{u}_k \le u_{\max-\min}$. In this case, k generates bandwidth $\tilde{u}_k$.

Third consider a VC i in $[\mathbf{VC}]_3$ that generates bandwidth $u_3 > u_{\max-\min}$ when $u = u_{\max-\min}$ and will not generate bandwidth less than u_3 (e.g. u_3 is k's MCR). When $u = u^-$, $u < u_3$, so VC i must remain in Group 3. Thus i generates bandwidth u_3.

Summarizing the above cases, each VC produces less or equal bandwidth when $u = u^-$ than when $u = u_{\max-\min}$. The (at least one) VCs in $[\mathbf{VC}]_2$ that remain in Group 2 produce strictly less bandwidth. Summing across all VCs, the bandwidth y^- generated when $u = u^-$ must be $y^- < y^{\max-\min} = y^*$. This violates Theorem 3. Thus given static ABR link capacities, it is impossible for u to converge to a value less than $u_{\max-\min}$. By a similar argument, it can be shown that it is impossible for u to converge to a value greater than $u_{\max-\min}$. Therefore u must equal $u_{\max-\min}$ in steady state and the controller (13)-(16) a steady-state solution that is max-min fair.

Given that the port j may serve VCs constrained outside of j's control, the above discussion shows that the port finds a locally max-min allocation. To show that a network of such ports converges to the global max-min fair solution, consider port j_1, the port in the network with the smallest explicit rate in the global max-min fair allocation. Port j_1 has no Group 1 VCs, and VCs are in Group 3 solely based on the number of VCs that pass through j_1, j_1's available bandwidth, and the lower bound rate for each VC passing through j_1. The explicit rate to which j_1 converges is not affected by other ports. Therefore, from above, j_1 converges to its globally max-min explicit rate, and each VCs passing through j_1 will be locked by j_1 throughout the network at its appropriate max-min rate. For port j_2, the port in the network with the second smallest explicit rate in the global max-min fair allocation, the only VCs in its Group 1 are those bottlenecked (at their correct rate) by port j_1. Port j_2 will separate VCs between Group 2 and 3 independent of ports j_t, $t > 2$. Therefore, j_2 must converge to an explicit rate which is the globally max-min fair rate, locking each VCs passing through j_2 at its appropriate global max-min rate. Iterating through all ports in the network, it is clear that each port converges to its globally max-min fair rate.

5.2 Dynamic Fairness

In addition to being max-min fair, the controller (13)-(16) is also fair in a dynamic sense. When the available bandwidth $y^*(n)$ varies with n, the controller generates a single explicit rate $u(n)$ at each n. Each Group 1 VC that is constrained to a bandwidth less than $u(n)$ by a link other than j will receive its constrained bandwidth at j. Each Group 3 VC will also receive its constrained bandwidth at j. Each Group 2 VC is granted an identical explicit rate $u(n)$. The sending rate of each greedy Group 2 VC is a time-shifted version of the series $\{u(n)\}$. Therefore the controller is dynamically fair to all VCs while allocating all the available bandwidth (Theorem 3).

6 Practical Results

The focus of this paper is introduce a congestion controller and to analytically investigate its appropriateness to the intended task. The depth of the above analytical treatment considerably assists the formulation of algorithmic enhancements designed to address practical considerations beyond stability, convergence and fairness. There is much to say about these practical considerations. Space limitations allow for only a cursory treatment of these considerations in this section. The interested reader is directed to [35], [32] and Chapter 4 of [34], which are dedicated to practical issues of the proposed controller.

6.1 Simulation Framework

To provide a baseline for comparisons in this paper, a common simulation framework is now defined. These simulations use the Matlab [52] simulation tool.

The plant, defined in Section 2 and shown in Figure 1, envisions a switch *SW* having an output port *j* containing a congestion controller. For the purpose of a common simulative framework, the output port rate of port *j* is 2488 Mbps (million bits per second) = 5.869 Mcps (million cells per second), i.e. an OC48, which is a realistic port speed for ATM switches currently under development. Of that, some subset (10-20% seems reasonable) of the bandwidth will be allocated for ABR traffic, and in the current framework, 1 Mcps is used as the average ABR rate for port *j*. Let C=200 Kcps (thousand cells per second) of this 1 Mcps constitute ABR traffic controlled by other ports, leaving on average 800 Kcps of ABR traffic responsive to the port *j*. The set-point y^* is therefore chosen to be a white Gaussian process with mean $E[y^*]$=1 Mcps and a standard deviation σ_{y^*} of 22 Kcps[7].

It seems plausible that the complexity of ABR will discourage its use for short-lived connections (e.g. Domain Name Server queries, individual e-mail deliveries, etc.). Instead ABR connections in a single port will likely constitute a small number of large bandwidth aggregations of traffic, e.g. connecting sites of a college or industrial campus. Therefore, for these simulations, let the 800 Kcps of responsive ABR traffic be comprised of 22 high-capacity, greedy sources, each averaging 15.4 Mbps = 36.4 Kcps. If the number of ABR cells that must include one RM cell, N_{RM}, is 32, then the per-connection rate of RM cells corresponding to responsive ABR sources is 1.14 Kcps, or one RM cell every 880 microseconds. The measurement and control sample time is $T_s = 1$ msec.

The minimum response delay d=10 msec. The distribution of the delays of the 22 sources is given by $B(z^{-1}) = z^{-10}(2+9z^{-1}+8z^{-2}+3z^{-3})$. This corresponds to a plant with one non-minimum phase

[7] These deviations about the mean of the desired ABR rate are determined by the extent that the port measures and re-allocates bandwidth from higher-level service category flows. It is somewhat uncertain how aggressively ports will attempt to re-allocate unused bandwidth. Very small variances are possible.

zero and a pair of complex minimum phase zeros. The number of taps in the controller is $dQ = 30$, with $V = 10$. The adaptation gain is set at its optimal value $\mu = 1$. Cell rates are not strictly limited to be non-negative, although manual inspections reveal that this rarely occurs after an initial transient.

6.2 Simulation Results without Algorithm Enhancements

Figure 9 shows the set-point error for one simulation. Considerable convergence occurs in the first few seconds. Convergence to zero set-point error is ensured by the results of Section 4, however this convergence is relatively slow. The rate of convergence is a function of the ratio of the smallest to the largest eigenvalues of $E\left[\mathbf{y}(n)\mathbf{y}(n)^T\right]$. In the example of Figure 9, this ratio is 2.16×10^{-7}, resulting in slow convergence.

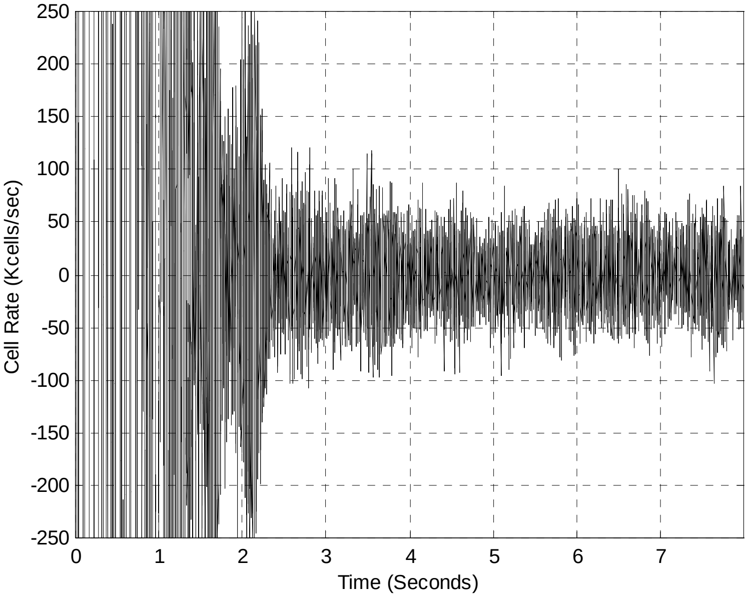


Figure 9 Set Point Error $y^*(n\,|\,n-d-V)-y(n)$.

6.3 Algorithm Enhancements

In this section, three additions to the Congestion Control mechanism are briefly mentioned. Each addition provides necessary mortar in cementing together theoretical analysis and practical design. These three modifications are singled out for attention here since each addresses a general issue likely to appear in many complex congestion control schemes, not just that of ATM ABR congestion control. A full presentation of these enhancements appears in [35], [32], [34].

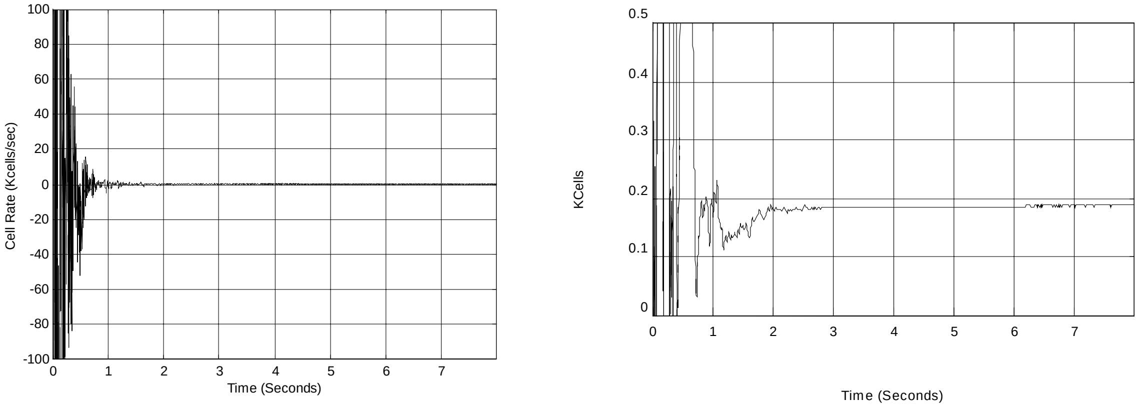
The first algorithm enhancement addresses the convergence rate of the controller. The results of Section 4 ensure that the originally proposed congestion controller eventually converges. However, without modification, convergence rates are unnecessarily, and possibly unacceptably, slow. The convergence rate decreases as eigenvalue spread of $E\left[\mathbf{y}(n)\mathbf{y}(n)^T\right]$ increases. This eigenvalue spread is often large due to the non-zero mean of the rate $y(n)$. The essence of the enhancement is to strategically subtract and add estimates of $E\left[y(n)\right]$ and $E\left[u(n)\right]$ throughout the system, thereby decreasing the eigenvalue spread of $E\left[\mathbf{y}(n)\mathbf{y}(n)^T\right]$. This produces a significant speedup in the rate of convergence.

The second algorithm enhancement responds to an addition to the plant. Specifically, a model of the buffer queue size is added to the plant, prompting a method to control this queue size. It is argued that size of output queue should be neither too long nor too short. Many congestion control schemes that directly control buffer size are computationally complex. The proposed enhancement measures the current queue size and appropriately scales $y^*(n+d+V \mid n)$ in order to bring the queue size to a desired value.

The third algorithm enhancement also responds to an enhancement in the plant model. The enhanced model generalizes the behavior of the non-responsive ABR sources, allowing them non-constant rates. This is modeled by adding to C a zero-mean noise source in the plant model. This noise causes biasing in the parameter estimates used for the controller. The enhancement reparameterizes the plant, thereby significantly reducing the bias of the adaptive estimates. Unlike previously published remedies for bias, this solution requires only a trivial amount of added calculations. Further, unlike other methods, this new method does not jeopardize convergence.

6.4 Simulation Results with Convergence Rate Algorithm Enhancement

Drawing from [35], [32], [34], Section 6.3 proposes several enhancements to the algorithm presented in Sections 3.3.3 and 3.3.4. To demonstrate the improvements obtained from the first two algorithm enhancements, the experiment shown in Figure 9 is repeated. Figure 10 shows the results. Note that the vertical axis scaling of Figure 9 and Figure 10a are not the same. Figure 10 shows much improved convergence.



a

b

Figure 10 After enhancements. (a) Set point error $y^*(n \mid n-d-V)-y(n)$; (b) Queue depth when queue target is 100-200 cells, $\sigma_{y^*}=22$

7 Comparisons to Less Complex Schemes

This section compares the proposed control scheme to less complex schemes. Other approaches to congestion control have been outlined in Section 1.2. Included in this list are approaches that claim to provide satisfactory performance with a lower computational cost than (13)-(16). In what follows, it is shown that the added computational cost of (13)-(16) provides better performance than less computationally complex schemes, specifically [22], [23], [25]. Also, (13)-(16) in its simplest ($dQ = 0$) case, is essentially equivalent in performance and complexity to the simpler schemes.

Consider the proposed controller (13)-(16), specifically the identification (16), with only one adaptive tap, i.e. $dQ = 0$, $V = 0$, $\mu = 1$:

$$\begin{aligned} \hat{q}(n+1) &= \hat{q}(n) + \frac{1}{y(n)}\big(u(n-d) - \hat{q}(n)\,y(n)\big) \\ &= \frac{u(n-d)}{y(n)} \end{aligned} \tag{46}$$

Note that in the $dQ = 0$ case, the NLMS adaptation devolves into a single division. Compare this to Fulton's identification [25]:

$$\hat{N}_{eff,\text{Fulton}}(n+1) = \frac{y(n)}{\overline{u}(n\text{-}1)}$$

where $\overline{u}(n-1)$ is the time average of a sequence of previous values of *u*. Fulton does not explicitly estimate d , therefore requires the averaging on u for convergence (unsurprisingly, the recommended time interval for averaging is d samples). Similarly, Imer [23] calculates

$$\hat{N}_{eff,\text{Imer}} = \frac{y}{u}\,.$$

every $d\,'$ samples, $d\,' \geq d$, where u is kept constant over the past $d\,'$ samples. The Fahmy parameter *Effective Number of active VCs,* or $\hat{N}_{eff,\text{Fahmy}}$, is defined similarly [22], albeit in an indirect manner.

Clearly $\hat{N}_{eff,\text{Laberteaux}} = 1/\hat{q}$, $\hat{N}_{eff,\text{Fulton}}$, $\hat{N}_{eff,\text{Imer}}$, and $\hat{N}_{eff,\text{Fahmy}}$ are adaptive estimates that attempt to capture the same information. In each controller, this value is used to divide the amount of available bandwidth y^* so that a future y will match a future y^* in some appropriate way. Thus the controller (13)-(16), in its simplest version ($dQ = 0$), is essentially equivalent, in performance and complexity, to the suggestions made by Fulton, Imer, and Fahmy.

Consider how these one-tap controllers perform. The controller in each case consists of dividing a future estimate of $y^*(n)$ by the associated $\hat{N}_{eff}$. All provide fair and efficient allocation of y^* in the long-term. The best such a controller could accomplish is that the incoming ABR bandwidth matches the available ABR traffic in the mean, i.e. $E\big[\chi(n)\big] = 0$, $\chi(n) \equiv y^*(n) - y(n)$.

While the authors of [22], [23], [25] make a fair and stable allocation their performance goal, here a fair (Section 5) and stable (Section 4) allocation is taken to be a minimum acceptable performance objective. This difference in the performance objective may be based on a modeling assumption. Clearly for ATM ABR congestion control systems, two quantities change with time: the amount of bandwidth allocated to ABR and the number of competing ABR connections vying for this bandwidth. Since operational experience with ABR is limited, it is difficult to know with certainty the time-scales over which these two quantities change. However, this paper assumes that an *Available* Bit Rate controller is likely to see its available bandwidth change more rapidly than the number of connections.

If the available bandwidth $y*(n)$ remains constant for long periods (e.g. multiples of the maximum round trip time, d), or $\sigma_{y*}{}^2 \approx 0$, then the single-tap schemes discussed above work effectively. Note that Imer, both in his development and simulations, assumes that $y*(n)$ is constant. Fulton uses $\overline{y}^*(n)$, the sample mean of $y*(n)$, in her calculation of the explicit rate $u(n)$.

Since this paper assumes that $y*(n)$ changes more quickly than changes in the number of ABR connections, $y*(n)$ is modeled as a noise source. Using this paper's notation, in the one-tap case, $y(n)$ can be modeled as

$$y(n) = y*(n+d+V \mid n)\frac{B\left(z^{-1}\right)+C}{\hat{N}_{eff}}. \tag{47}$$

i.e. a noise source filtered through the FIR filter $\left(B\left(z^{-1}\right)+C\right)/\hat{N}_{eff}$. From (47),

$$\chi(n) = y*(n \mid n-d-V) - y(n) = y*(n \mid n-d-V)\left(1-\frac{B\left(z^{-1}\right)+C}{\hat{N}_{eff}}\right)$$

Unless, $B\left(z^{-1}\right) = b_o$, the variance of $\chi(n)$, $\sigma_{\chi}{}^2$, increases as $\sigma_{y*}{}^2$ increases. The queue size, $queue(n)$, is the integral of $\chi(n)$. From the definition of variance, for a one-tap controller N_{eff}, the variance of $queue(n)$, σ^2_{queue}, also increases as $\sigma_{y*}{}^2$ increases (these observations will be supported by simulations below). This increases the necessary buffer size if overflow is to be avoided. Also, if buffer underflow is to be avoided, a larger average queue size must be targeted as σ^2_{queue} increases.

Since larger queue sizes require a larger memory cost and also increases the delay through the switch, both of which are preferably avoided, this paper views minimizing $\sigma_{\chi}{}^2$, and thus $queue(n)$, as a desirable performance goal.

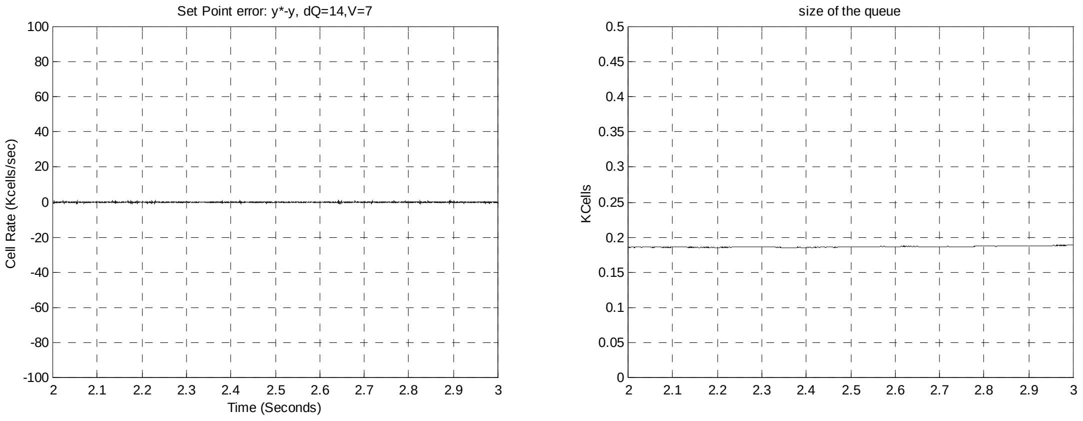


Figure 11a Set-point error, $y^*(n)-y(n)$, and size of queue, $queue(n)$, with $dQ=14, V=7$, $\sigma_{y^*}=22$

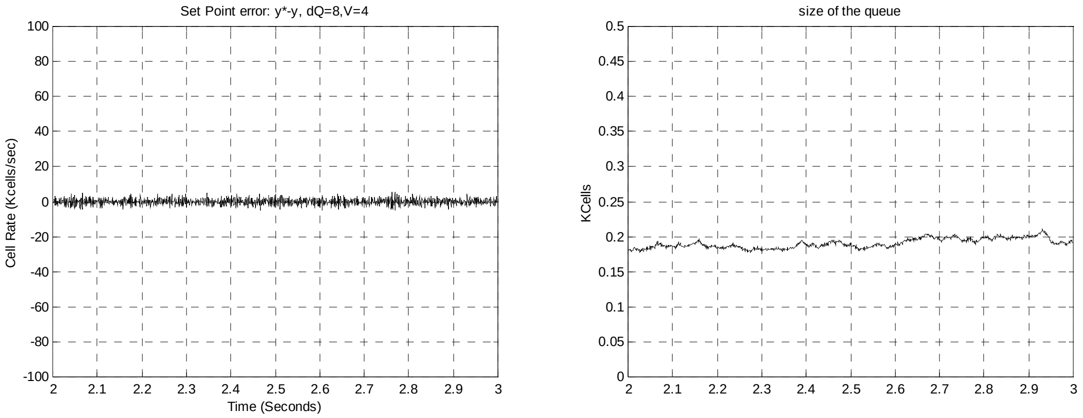


Figure 11b Set-point error, $y^*(n)-y(n)$, and size of queue, $queue(n)$, with $dQ=8, V=4$, $\sigma_{y^*}=22$

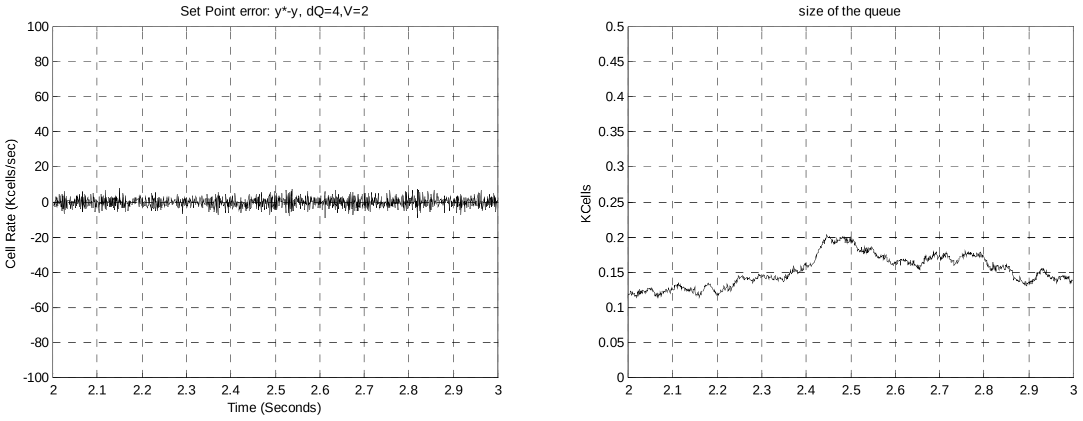


Figure 11c Set-point error, $y^*(n)-y(n)$, and size of queue, $queue(n)$, with $dQ=4, V=2$, $\sigma_{y^*}=22$

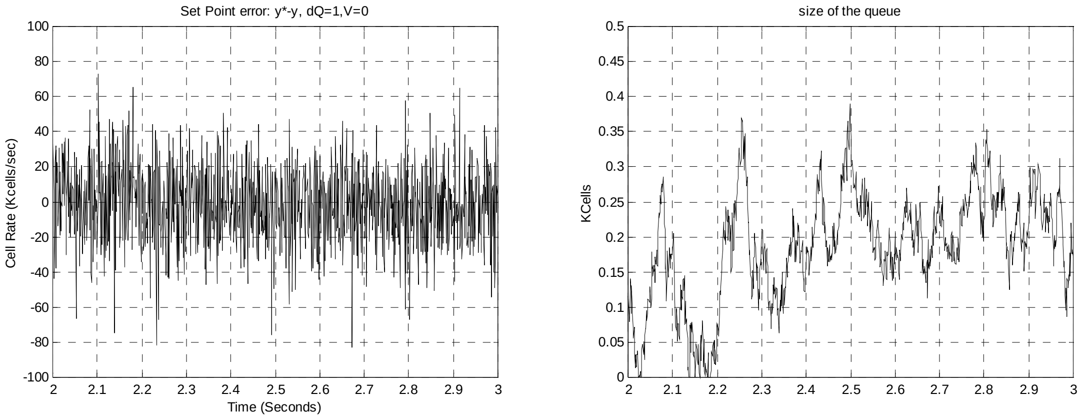


Figure 11d Set-point error, $y^*(n)-y(n)$, and size of queue, $queue(n)$, with $dQ=1, V=0$, $\sigma_{y^*}=22$

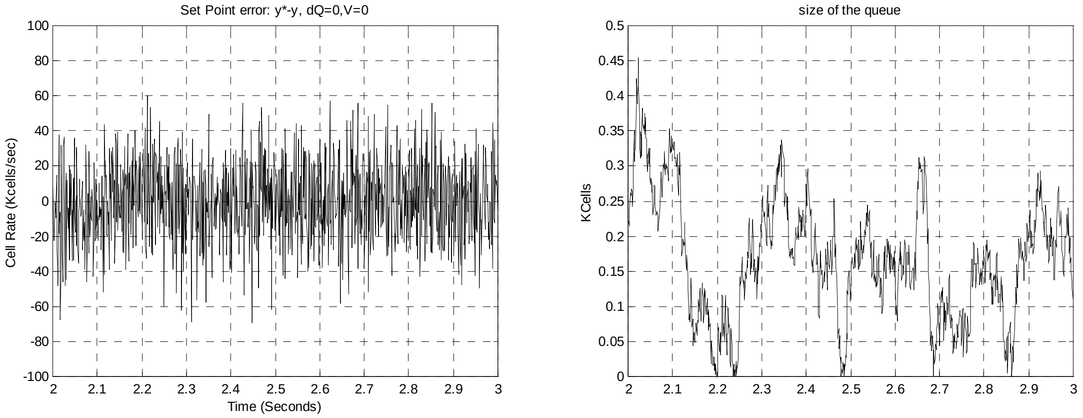


Figure 11e Set-point error, $y^*(n)-y(n)$, and size of queue, $queue(n)$, with $dQ=0, V=0$, $\sigma_{y^*}=22$

Using the simulation environment established in Section 6.1, Figure 10 shows how effectively $\sigma_\chi{}^2$ and queue size can be minimized by using 31 ($dQ=30$) taps in the controller (13)-(16). Figure 11a-Figure 11e show the performance, both in terms of set-point error, $y^*(n)-y(n)$, and the size of the queue, $queue(n)$, gradually degrading as the number of taps dQ decreases. These simulations are identical to those shown in Figure 10—where the desired queue size is 100-200 cells—except for changes in dQ and V as noted. In the limiting one-tap case ($dQ=0$), the performance is essentially equal to the performance of Fulton's UT algorithm, shown in Figure 12a. This supports the near-equivalence of performance predicted in the discussion above.

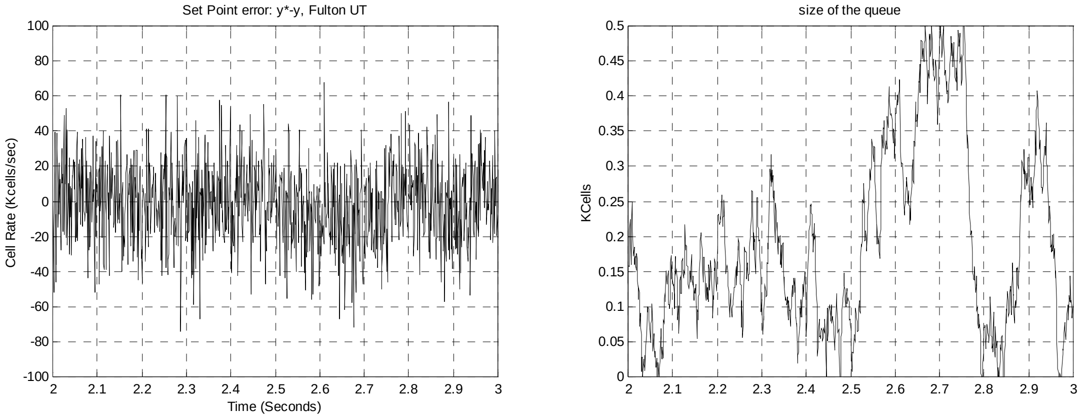


Figure 12a Set-point error, $y^*(n)-y(n)$, and size of queue, $queue(n)$, with Fulton's UT algorithm, $\sigma_{y^*}=22$

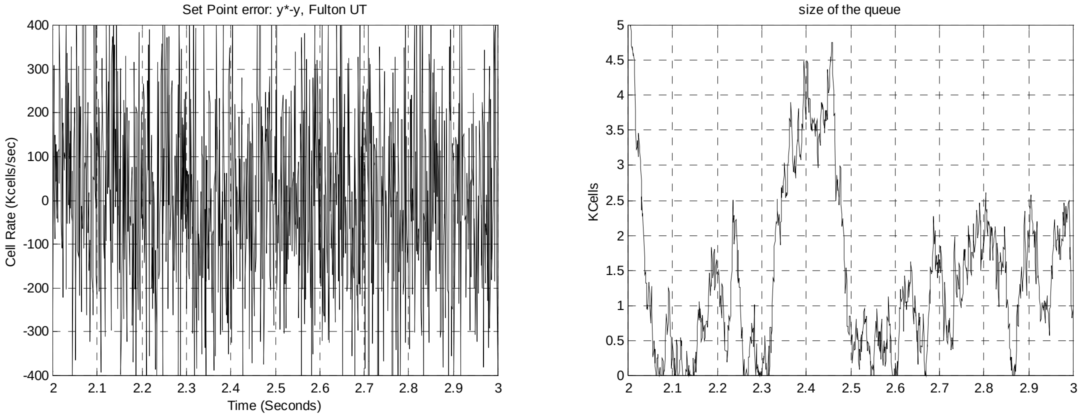


Figure 12b Set-point error, $y^*(n)-y(n)$, and size of queue, $queue(n)$, with Fulton's UT algorithm, $\sigma_{y^*}=223$

Some may be satisfied with the performance of the simple, one-tap controllers shown in Figure 11e and Figure 12a. However, it is important to note that performance of one-tap controllers is highly dependent on the standard deviation of the set-point, σ_{y^*}. When σ_{y^*} is increased an order of magnitude from 22 to 223, the performance is observed to degrades an order of magnitude (compare Figure 11e to Figure 13a and Figure 12a to Figure 12b). In contrast, the multi-tap controllers ($dQ>0$) improve performance as the number of taps increase, with the original $dQ=30$ case showing no performance impairment due to increased σ_{y^*} (compare Figure 13d and Figure 10).

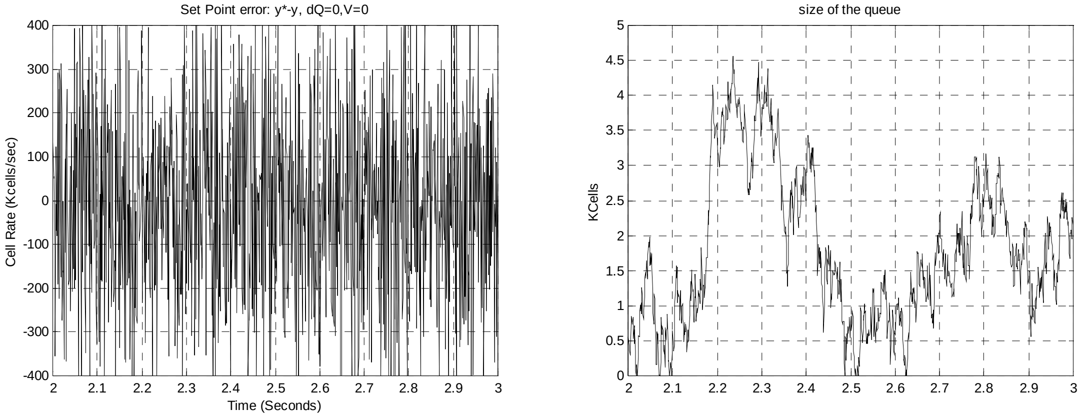


Figure 13a Set-point error, $y^*(n)-y(n)$, and size of queue, $queue(n)$, with $dQ=0, V=0$, $\sigma_{y^*}=223$

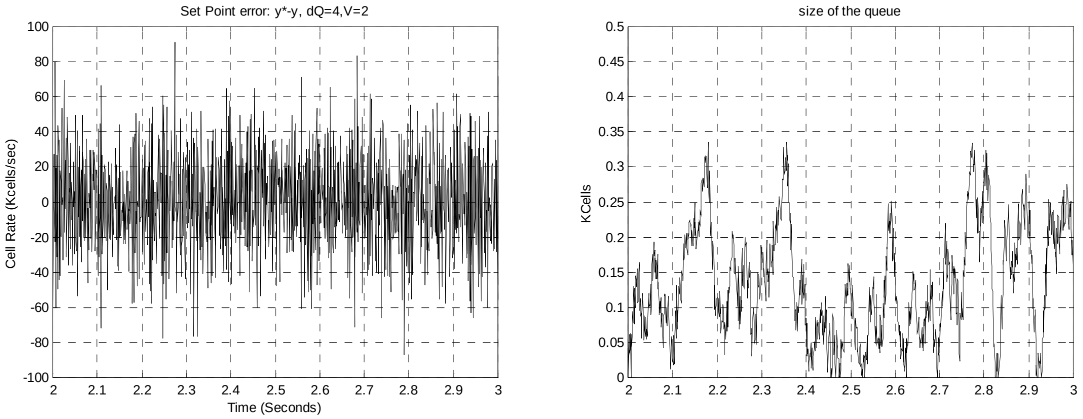


Figure 13b Set-point error, $y^*(n)-y(n)$, and size of queue, $queue(n)$, with $dQ=4, V=2$, $\sigma_{y^*}=223$

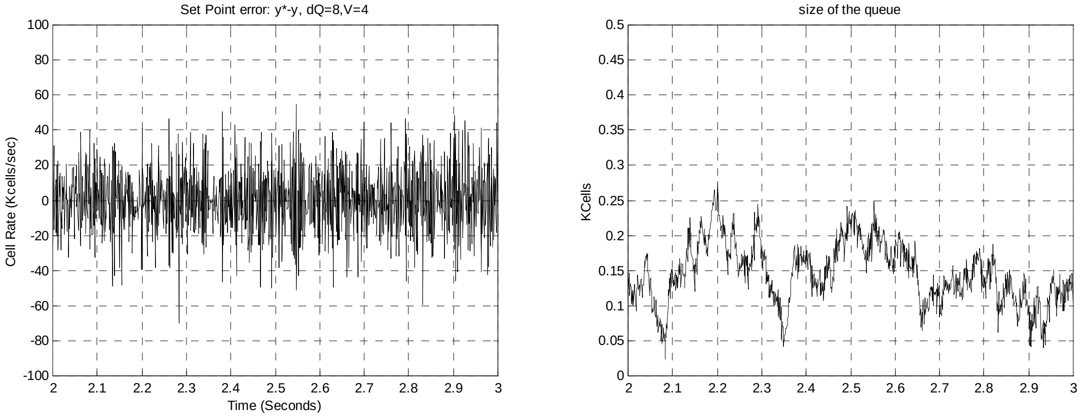


Figure 13c Set-point error, $y^*(n)-y(n)$, and size of queue, $queue(n)$, with $dQ=8, V=4$, $\sigma_{y^*}=223$

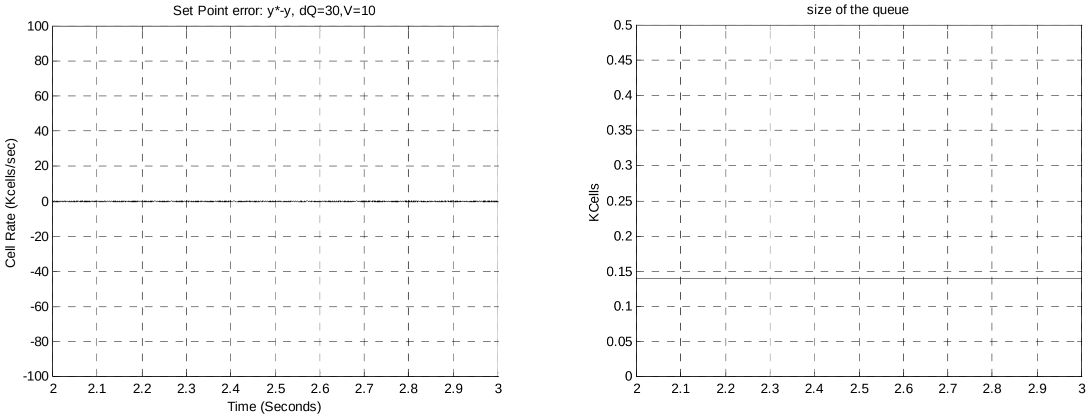


Figure 13d Set-point error, $y*(n)-y(n)$, and size of queue, $queue(n)$, with $dQ=30, V=10$, $\sigma_{y*}=223$

To summarize, the added complexity of (13)-(16) does indeed provide much improved performance over those of [22], [23], [25]. Further, (13)-(16) can be simplified in implementation (by reducing dQ), thereby gradually reducing its performance and complexity to that of the popular one-tap solutions [22], [23], [25]. Controller (13)-(16) provides a continuum of performance and complexity. For example, if the complexity budget for a specific available bit rate application allows five taps ($dQ=4$), then the added complexity of these five taps appears justified.

If it is later shown that available bit rate network applications change their number of sharing connections rapidly as compared to the amount of available bandwidth–corresponding to the small σ_{y*} case–then a large number of taps may be unwarranted. In fact, using a small number of taps would improve coefficient convergence times. However, if available bandwidth changes rapidly as compared to the number of sharing connections–the large σ_{y*} case–then a large number of coefficients is warranted.

Other even computationally simpler congestion control schemes have been presented for the Internet (see Section 1.2). Generally these schemes are one-bit marking approaches. These approaches occupy a very different location on the performance/complexity curve of congestion control. At best, these one-bit schemes will match the arriving bandwidth to the available bandwidth in the mean, with even greater error variance σ_{χ}^{2}. The comparisons made above can therefore be extended to the one-bit Internet proposals.

8 Summary

This paper takes up the challenge of finding an effective control strategy for the explicit rate congestion controller. The problem is introduced in Section 1. Section 2 defines a plant model. Section 3 then recognizes that the plant model defined in Section 2 is frequently non-minimum phase. Several strategies appropriate for the control of non-minimum phase plants are reviewed. In the end, one control strategy is chosen for its comparatively low computational cost, realizability, and attractive convergence and fairness properties. Formal convergence analysis is covered in Section 4, which contains two complimentary proofs. Each of these proofs begins with different assumptions; both suggest attractive analytical convergence properties. Section 5 shows that the final solution is fair in both the max-min and the dynamic sense. Other practical considerations for the proposed controller are reported in other papers and are briefly summarized in Section 6. The question of complexity verses performance is addressed in

Section 7, where it is shown that the current scheme performs better than other, less-complex schemes. The simulations in Sections 6 and 7 demonstrate proper functioning of the congestion control system and support the findings in the previous sections.

APPENDIX

Proof of Lemma 2: The proofs for $\mathbf{A}$ and $\mathbf{H}$ are nearly identical; only the former is shown. Let A_{ij} indicate the element of $\mathbf{A}$ in the *i*'th row, *j*'th column. Then

$$\left|A_{ij}\right| = \left| E\left[\frac{\boldsymbol{\psi}(n)\boldsymbol{\psi}(n)^T}{\boldsymbol{\psi}(n)^T\,\boldsymbol{\psi}(n)}\right]_{ij}\right| = \left|\int_{\mathbf{X}} \frac{x_i x_j}{\|\mathbf{X}\|^2} f_{\boldsymbol{\psi}}(\mathbf{X})\,d\mathbf{X}\right| \le \frac{1}{\alpha_0}\left|\int_{\mathbf{X}} x_i x_j f_{\boldsymbol{\psi}}(\mathbf{X})\,d\mathbf{X}\right| = \frac{1}{\alpha_0}\left|E\left[\psi_i(n)\psi_j(n)\right]\right|$$

where the inequality is from Assumption 4, and from (24), $\left|A_{ij}\right| \le 0$ if $i \ne j$.

Proof of Lemma 3: $A_{ii} = \int_{\mathbf{X}} \frac{x_i(n)^2}{\sum_{j=1}^{dQ+1} x_j(n)^2} f_{\boldsymbol{\psi}(n)}(\mathbf{X})\,d\mathbf{X}$. $0 \le \frac{x_i(n)^2}{\sum_{j=1}^{dQ+1} x_j(n)^2} \le 1$ and $0 \le f_{\boldsymbol{\psi}(n)}(\mathbf{X}) \le 1$ for any $\mathbf{X}$. Then it is possible to choose a closed, bounded set of $\mathbf{X}$, $\mathbf{X}_{set}$, that simultaneously satisfies four constraints: 1. $x_i \ne 0$, 2. $\sum_j x_j^2 < \infty$, 3. $0 < f_{\boldsymbol{\psi}(n)}(\mathbf{X})$, 4. $\int_{\mathbf{X}_{set}} \frac{x_i(n)^2}{\sum_{j=1}^{dQ+1} x_j(n)^2} f_{\boldsymbol{\psi}(n)}(\mathbf{X})\,d\mathbf{X} > \alpha_1 > 0$. Since $\frac{x_i(n)^2}{\sum_{j=1}^{dQ+1} x_j(n)^2} f_{\boldsymbol{\psi}(n)}(\mathbf{X})$ is non-negative for every $\mathbf{X}$ outside of $\mathbf{X}_{set}$, the proof is completed.

Proof of Lemma 4: (Sufficiency) By contradiction. For $i \ne j$, $E\left[\psi_i(n)\psi_j(n)\right] = E\left[\psi_i(n)\right]E\left[\psi_j(n)\right]$. If less than dQ elements of $\boldsymbol{\psi}(n)$ are zero mean, $E\left[\boldsymbol{\psi}(n)\boldsymbol{\psi}(n)^T\right]$ is not diagonal, contradicting (24). (Necessity) If no less than dQ of the $dQ+1$ elements of $\boldsymbol{\psi}(n)$ are zero mean, then the independence of the terms of $\boldsymbol{\psi}(n)$ gives (24), concluding the proof.

BIBLIOGRAPHY

[1] J. Kenney, Editor, *Traffic Management Specification Version 4.1*, available from [51].

[2] R. Jain, "Congestion Control and Traffic Management in ATM Networks: Recent Advances and A Survey," Computer Networks and ISDN Systems, Vol. 28, No. 13, October 1996, pp. 1723-1738.

[3] F. Bonomi, and K. Fendick, "Rate-based flow control framework for the available bit rate ATM service," IEEE Network 9 2 Mar-Apr 1995 p. 25-39.

[4] C. Rohrs, R. Berry, and S. O'Halek, "Control engineer's look at ATM congestion avoidance," Computer Communications. v 19 n 3 Mar 1996. pp. 226-234.

[5] C. Rohrs and R. Berry, "A Linear Control Approach to Explicit Rate Feedback in ATM Networks," IEEE Infocom '97 Kobe, Japan v. 3 1997 pp. 277-282

[6] J. Bennett and G. Tom Des Jardins, "Comments on the July PRCA Rate Control Baseline," AF-TM 94-0682, available from [51], July 1994.

[7] L. Benmohamed and S. M. Meerkov, "Feedback Control of Congestion in Packet Switching Networks: The Case of a Single Congested Node," IEEE/ACM Transactions on Networking, Vol. 1, No. 6, December 1993.

[8] L. Benmohamed and S. M. Meerkov, “Feedback control of congestion in packet switching networks: The case of multiple congested nodes,” International Journal of Communication Systems Vol. 10, No. 5, p 227-246, Sep-Oct 1997.

[9] A. Kolarov and G. Ramamurthy, “A Control Theoretic Approach to the Design of Closed Loop Rate Based Flow Control for High Speed ATM Networks,” IEEE/ACM Transactions on Networking, vol. 7, Oct. 1999.

[10] J-C. Bolot, “A self-tuning regulator for adaptive overload control in communication networks,” Proc. 31st IEEE Conference on Decision and Control, Tucson, AZ, December 1992.

[11] E. Altman, F. Baccelli, J-C. Bolot, “Discrete-time analysis of adaptive rate control mechanisms,” Proc. 5th Intl. Conf. on Data Communication Systems and their Performance, pp. 121-140, Raleigh, NC, Oct. 1993. Apparently reprinted in *High Speed Networks and their performance*, H. G. Perros and Y. Viniotis Eds., North Holland, pp. 121-140, 1994.

[12] O. Ait-Hellal , E. Altman and T. Basar , “Rate based flow control with bandwidth information,” (invited paper) European Trans. on Telecom., special issue on ABR, pp. 55-66, 1996. A short version (invited paper) the proceedings of the 35th IEEE Conference on Decision and Control, Kobe, Japan, Dec. 1996.

[13] E. Altman and T. Basar, “Optimal Rate Control for High Speed Telecommunication Networks,” 34th IEEE Conference on Decision and Control, December 1995, New Orleans, Louisiana, (invited paper). A more detailed version: Research report UILU-ENG-95-2235, DC-169, University of Illinois at Urbana-Champaign, October, 1995.

[14] Z. Pan, E. Altman and T. Basar, “Robust Adaptive Flow Control in High Speed Telecommunication Networks,” (invited paper) the proceedings of the 35th IEEE Conference on Decision and Control , Kobe, Japan, Dec. 1996.

[15] E. Altman, T. Basar, and R. Srikant, “Multi-User Rate-Based Flow control with Action Delays: A Team-Theoretic Approach,” IEEE Conference on Decision and Control, Vol. 3, p. 2916-2921, Dec 10-12 1997 1997.

[16] E. Altman, Eitan, T. Basar, and R. Srikant, “Robust rate control for ABR sources,” Proceedings - IEEE INFOCOM v 1, p 166-173, Mar 29-Apr 2 1998.

[17] R. Jain, S. Kalyanaraman, and R. Viswanathan, "A Sample Switch Algorithm," AF-TM 95-0178R1, February 1995.

[18] R. Jain, S. Kalyanaraman, and R. Viswanathan, "The OSU Scheme for Congestion Avoidance Using Explicit Rate Indication," AF-TM 94-0883, September 1994.

[19] R. Jain, S. Kalyanaraman, and R. Viswanathan, "The EPRCA+ Scheme" AF-TM 94-0988, October 1994.

[20] R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy, F. Lu, "ERICA+: Extensions to the ERICA Switch Algorithm," AF-TM 95-1145R1, October 1995.

[21] B. Vandalore, R. Jain, R. Goyal, S. Fahmy, “Design and Analysis of Queue Control Functions for Explicit Rate Switch Schemes,” Proceedings of IC3N '98, Lafayette, LA, pp. 780-786, October 1998.

[22] S. Fahmy, R. Jain, S. Kalyanaraman, R. Goyal and B. Vandalore, “On Determining the Fair Bandwidth Share for ABR Connections in ATM Networks,” Proceedings of the IEEE International Conference on Communications (ICC) 1998, Atlanta, GA, June 1998, Vol. 3, pp. 1485-1491.

[23] O. Imer et al., “ABR Congestion Control in ATM Networks”, IEEE Control System Magazine, February 2001.

[24] K. Laberteaux and C. Rohrs, “Application Of Adaptive Control To ATM ABR Congestion Control,” Globecom ’98, Sydney, Australia, (available at http://www.geocities.com/klaberte/contributions.html), 1998.

[25] C. Fulton and S. Q. Li , “UT: ABR Feedback Control with Tracking,” Proc. IEEE Infocom'97 Conference, April 1997.

[26] Y. D. Zhao, S. Q. Li and S. Sigarto , ``A Linear Dynamic Model for Design of Stable Explicit-Rate ABR Control Schemes," ATM Forum Contribution 96-0606, April. 1996.

[27] O. Imer et al., “ABR Congestion Control in ATM Networks”, IEEE Control System Magazine, February 2001.

[28] C.F. Su, G. de Veciana, and J. Walrand, “Explict Rate Flow Control for ABR Services in ATM Networks,” IEEE/ACM Transactions on Networking, vol. 8, June 2000.

[29] S. Mascolo, “Smith Principle for Congestion Control in High-Speed Data Networks,” IEEE Trans. Auto. Control, Vol. 45, No. 2, pp. 358-364, Feb. 2000.

[30] K. Laberteaux and C. Rohrs, “On the Convergence of a Direct Adaptive Controller for ATM ABR Congestion Control,” Int Conf Comm 2000, (available at http://www.geocities.com/klaberte/contributions.html), June 2000.

[31] K. Laberteaux and C. Rohrs, “A Direct Adaptive Controller for ATM ABR Congestion Control,” Amer Control Conf 2000, Chicago, IL, (available at http://www.geocities.com/klaberte/contributions.html), June 2000.

[32] K. Laberteaux, C. Rohrs, and P. Antsaklis “A Pragmatic Controller for Explicit Rate Congestion Control,” Globecom 2001, San Antonio, TX, (available after presentation at http://www.geocities.com/klaberte/contributions.html), Nov 2001.

[33] K. Laberteaux and C. Rohrs, “A Proof of Convergence for a Direct Adaptive Controller for ATM ABR Congestion Control,” Technical Report ND-ISIS-2000-02, (available from http://www.nd.edu/~isis), 2000.

[34] K. Laberteaux, “Explicit Rate Congestion Control for Data Networks,” a Dissertation, Dept. of Elec. Eng., Univ. of Notre Dame (available at http://www.geocities.com/klaberte/contributions.html), 2000.

[35] K. Laberteaux, C. Rohrs, and P. Antsaklis “A Practical Controller for Explicit Rate Congestion Control,” IEEE Transactions on Automatic control, Vol 47, p. 960-978, (available at http://www.geocities.com/klaberte/contributions.html), June 2002.

[36] S. Low and D. Lapsley, “Optimization Flow Control, I: Basic Algorithm and Convergence,” IEEE/ACM Trans. On Networking, 7(6), pp. 861-874, Dec 1999.

[37] S. Floyd, and V. Jacobson, “Random Early Detection gateways for Congestion Avoidance,” IEEE/ACM Transactions on Networking, Vol.1 No. 4, pp. 397-413, August 1993.

[38] S. Floyd, “TCP and Explicit Congestion Notification”. ACM Computer Communication Review, V. 24 N. 5, p. 10-23, October 1994.

[39] S. Athuraliya et al., “REM: Active Queue Management”, Proc. 17th Intl. Teletraffic Congress, September, 2001.

[40] V. Misra, W. Gong, D. Towsley, "Stochastic Differential Equation Modeling and Analysis of TCP Windowsize Behavior", Proc. Performance'99, Istanbul, Turkey, October 1999.

[41] C.V. Hollot et al., "On Designing Improved Controllers for AQM Routers Supporting TCP Flows", IEEE Infocom, April 2001.

[42] Yahagi, T. and Lu, J., “On Self-Tuning Control of Nonminimum Phase Discrete-Time Systems Using approximate Inverse Systems,” ASME Journal of Dynamic Systems, Measurement, and Control, Vol. 115, p. 12-18, March 1993.

[43] A. Oppenheim and R. Schafer, *Digital Signal Processing*, Prentice-Hall, 1975.

[44] B. Widrow and E. Walach, *Adaptive Inverse Control*, Prentice-Hall, 1996.

[45] B. Widrow and G. Plett, “Nonlinear adaptive inverse control,” IEEE CDC 1997, San Diego, CA, pp. 1032-1037, vol. 2., December 1997.

[46] P. Peebles, *Probability, Random Variables, and Random Signal Principles, 2nd Ed.* McGraw-Hill, New York, NY, 1987.

[47] M. Tarrab and A. Feuer, “Convergence and Performance Analysis of the Normalized LMS Algorithm with Uncorrelated Gaussian Data,” *Trans Info Theory*, Vol. 34, No. 4, p. 680-691, July 1988.

[48] G. Goodwin and K. Sin, *Adaptive Filtering Prediction and Control*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1984.

[49] Haykin, Simon S., *Adaptive Filter Theory*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1991.

[50] D. Bertsekas and R. Gallager, *Data Networks, 2nd ed.*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1992.

[51] ATM Forum web site, http://www.atmforum.org.

[52] Mathworks, Inc., *Matlab*, R11, http://www.mathworks.com/.