EXPLICIT RATE CONGESTION CONTROL FOR DATA NETWORKS

A Dissertation

Submitted to the Graduate School
of the University of Notre Dame
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

by

Kenneth Patrick Laberteaux, B.S.E., M.S.

Panos J. Antsaklis, Director

Charles E. Rohrs, Director

Department of Electrical Engineering
Notre Dame, Indiana
October 2000

EXPLICIT RATE CONGESTION CONTROL FOR DATA NETWORKS

Abstract

by

Kenneth Patrick Laberteaux

This dissertation addresses congestion control for explicit rate controlled data networks. Congestion control is a closed-loop technique to regulate the influx of data into a network. In the application considered here, an internal switching node employs congestion control to specify source input rates such that the traffic arriving at the node matches the node's available resources in a fair and efficient manner. Due to the closed-loop and dynamic nature of this problem, adaptive control techniques are utilized extensively. The specific context for this study is the Available Bit Rate (ABR) service category of Asynchronous Transfer Mode (ATM) networks. However, the obtained results apply beyond this specific protocol due to the generality of the derived plant model. It 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.

To my parents

CONTENTS

FIGURES

ACKNOWLEDGEMENTS

Dr. Panos Antsaklis was my advisor throughout my graduate education and served as dissertation co-director. As such, he played a fundamental role in shaping this dissertation. I am extremely grateful for his considerable intellect, dedication, and experience. Dr. Antsaklis is a man of insight, wit, and charm; spending time with him is always rewarding.

Dr. Charlie Rohrs introduced me to the research developed by this dissertation and served as dissertation co-director. Charlie has contributed tremendously to my learning throughout my graduate education. By spending time with Charlie, one discovers that learning and thinking are two of life's greatest joys. Charlie has made an indelible impression on my life. I am truly in his debt.

I thank Drs. Daniel Costello, Thomas Fuja, and Michael Lemmon for serving on my Graduate Committee. Their careful review and insightful questions substantially improved the quality of this dissertation.

Tellabs Operations, Inc. generously supported this research. In addition, several members of the Tellabs family made important contributions. Dr. Mike Schafer arranged the sabbatical during which I completed this work. Dr. John Kenney, truly an expert in the field of this dissertation, patiently answered countless questions, often within hours of

important deadlines. When I came closest to abandoning this project, Mike and John's strong support played a key part in my decision to continue. I also thank Marilynn Anson, who masterfully typed, set equations, and drew figures for large portions of this dissertation, and Kathy Eaton, who by editing the final draft joins a very short list of people who have or will read this dissertation.

I thank my sisters, Kathy and Kristy, and my new brothers, John and Chris. Their love, encouragement, and humor sustained me throughout this experience.

Finally I thank my parents, Tom and Rita. I have felt their love every day of my life. I owe them everything. No words can adequately express the appreciation, admiration, and affection I feel for them. It is to them that I dedicate this work.

Ken Laberteaux

CHAPTER 1 INTRODUCTION

1.1 Congestion Control for Data Networks

From the first bugles and smoke signals to the Internet and video-conferencing links of today, data and communication networks have tremendously impacted our society. Commerce, recreation, governance, warfare, and personal relationships have all been drastically affected. Today the desire to exchange data and communicate shows no signs of abating. In fact, the amount of data carried by the Internet backbone now doubles every 100 days [55].

As communicated data grows in both scale and diversity, networks are becoming increasingly complex. This increasing complexity encourages network designers to draw upon diverse expertise from a variety of fields. Congestion control, the topic of this dissertation, is a prime example of such cross-discipline research, incorporating both fields of data networking and control theory.

To gain some initial intuition for congestion control, consider the analogy of air travel on a hypothetical, low-cost airline, LCA. LCA uses small planes with limited range. A typical LCA passenger expects four to ten airport connections, most of which last only a few minutes, but occasionally can last several hours. Further, LCA does not take reservations from passengers until they are ready to leave for the airport. A prospective passenger calls an LCA booking agent, specifies his current location and destination, and then receives a departure time.

The LCA booking agent performs two tasks before informing the passenger of his departure time. Routing is the first task. If the passenger is in Detroit and wishes to travel to Los Angeles, the agent might determine that the best route includes stops in Chicago, St. Louis, Denver and Las Vegas. The agent's second task is that of congestion control. This task determines when the passenger should begin his journey in order to minimize his travel time. If the agent discovers that Denver's airport is currently experiencing severe congestion, he must do his part to reduce the congestion at Denver by decreasing the rate of departures routed through Denver. By creating such an itinerary, the agent delays the passenger's departure. However, this delay actually improves the passenger's travel experience–who would not prefer to wait at home than in an airport?

The key virtue of congestion control is that both a network provider and its customers frequently benefit when the provider admits the *correct* number of customers in a given time period. Admitting more customers causes heavy congestion. Admitting fewer customers allows resources to go under-utilized, creating an opportunity cost that is eventually transferred to customers. Congestion control addresses exactly this issue.

More precisely, 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 experience congestion, leading to undesirable data loss, excessive delays, or both[1]. 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 its sending rate. Well suited for data that is not strongly delay sensitive, closed-loop congestion control uses a feedback mechanism and draws heavily on feedback control theory.

This dissertation studies congestion control as it applies to the Available Bit Rate (ABR) service category of Asynchronous Transfer Mode (ATM) networks, or ATM ABR. As demonstrated in Section 1.3.1, ATM ABR specifies a large collection of tools that enables sophisticated explicit rate (ER) congestion control. There have been many studies on how to best use these tools towards the goal of ATM ABR congestion control (see Section 1.3.2 and references therein). This dissertation is dedicated to this goal. However, the characteristics of ATM ABR congestion control that are likely to reemerge in future network protocols receive the majority of the attention. For this reason, the results of this dissertation should have application outside of the particular ATM ABR protocol.

[1] Note that the issues addressed by congestion control can also be attacked using open-loop methods, including policing, shaping, and call admission control. In a call admission control environment, a customer makes a request of the network to deliver data meeting certain specifications. The network then examines its current state and, if it is capable, makes a forward-looking commitment to the customer. Once the network informs the customer of an affirmative decision, the network takes no further action to limit the customer’s flow of data, assuming the customer abides by his part of the contract. One common but simplistic example of an open-loop call admission control decision is a telephone system’s decision to make a constant bit rate connection from one phone to another, signaled by either a ring or a busy signal. More generally, the customer requests a connection that varies in rate as some unknown function of time. The source characterizes this desired flow, and the network determines if it can adequately support this new connection in addition to its previously-made commitments to other customers. Generally the network attempts to achieve some statistical multiplexing gain, requiring a non-trivial decision process. Call admission control is well suited for traffic that is predictable or consistent. Congestion control is better suited for traffic with unpredictable or highly varying requirements.

For many years, ATM ABR was touted as “the next big thing,” not in small part due to its extensive support of sophisticated congestion control. However, long before the completion of this dissertation, the rising star of ATM ABR began to fade. Other next big things commanded the headlines of the trade magazines. ATM ABR, despite being well suited for the explosively popular applications of web browsing, e-mail, and data backup, is yet to be widely utilized. Few personal computers are ATM-enabled, much less ABR-enabled. Instead, the dominant protocol of today’s Internet remains TCP/IP, with its comparatively less sophisticated congestion control ([40]-[42],[44]-[46]).

However, ATM deployment is on the rise. “Worldwide revenue for ATM equipment and services combined is projected to reach almost $9.5 billion in 2001, up from $2.4 billion in 1997–a compound annual growth rate (CAGR) of 41 percent” [52]. Today ATM is primarily employed as a transport medium for TCP/IP traffic, including so-called Voice over IP (VoIP) Networks [53]. New technologies, such as FAST [56], Differentiated UBR [57], and Guaranteed Frame Rate [1], respond to this new role for ATM.

The extent of future usage of the ABR service category of ATM is still unknown. Yet despite the ambiguities of the marketplace, there are at least two reasons to continue research in ATM ABR congestion control. The first reason is that ABR may yet see widescale adoption. Although no longer the newest technology, ATM ABR has yet to be outperformed by newer technologies in its stated task–providing efficient, fair, and reliable transport for non-real-time, large bandwidth data applications. In fact, ABR’s critics contend that its high-performance-through-high-complexity approach exceeds, both in capability and cost, the needs of the network marketplace of tomorrow. These critics claim that cheaper and simpler solutions, albeit less robust, are possible by extending the TCP/IP paradigm. Examples of these innovations include Differentiated Services (diffserv) [47], Random Early Detection (RED) [48],[49], Prioritized Switch Fabrics, and Multi-Protocol Label Switching (MPLS) [51].

Yet these TCP/IP-centric approaches also have their critics:

> Customers aren't asking for less reliable networks–on the contrary, they are demanding more stringent service guarantees and higher levels of performance. Service providers are unlikely–and unwilling–to move toward network architectures that offer less accountability and control than they have today [with ATM networks] . . . Despite the formidable hype for an all-IP communications world, most service providers will bank on the most reliable and most mature technologies. That spells ATM for the time being, and for quite some time to come [54].

The outcome of the current ATM verses TCP/IP battle remains uncertain. ATM ABR is by now a well-defined technology. The onus is on the new TCP/IP enhancements to prove their claims of doing well enough with less.

The second reason for continuing study of ATM ABR congestion control is that it reveals and attempts to answer basic issues likely to arise in future networking protocols. Whether or not ATM ABR is widely deployed, future networks will almost certainly require high quality congestion control. In the case of this dissertation, Section 1.2 examines the ATM ABR protocol in detail, but the model developed in Chapter 2 remains general[2]. This general plant description is studied to discover an appropriate general control strategy. Therefore, as the title of this dissertation suggests, the results of this work are applicable to the basic study of explicit rate congestion control 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 any protocol likely to dominate future data networks will employ a high-performance explicit rate congestion control mechanism.

1.2 Congestion Control in ATM Networks

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 services digital networks (B-ISDN) [2]. ATM networks provide six service categories [1]; a given ATM network may implement some or all of these service categories:

1. Constant Bit Rate (CBR) is used for traffic requiring a constant cell rate.
2. Real Time Variable Bit Rate (rt-VBR) is used for traffic with a varying cell rate which can be expressed with a few traffic specifications, e.g. maximum cell rate, sustainable cell rate, etc.

[2] This general plant model assumes that some network element is tasked with fairly and efficiently allocating a time-varying quantity of bandwidth to consumers. The number of these bandwidth customers is initially unknown. Some customers are likely not to be responsive. The network element employs a congestion controller that controls the rate influx of the bandwidth consumers by explicitly feeding back a maximum send rate.

3. Non-Real Time Variable Bit Rate (nrt-VBR) is similar to rt-VBR, except no guarantees are made about the delivery delay or delay jitter; thus, nrt-VBR is only appropriate for non-real time traffic.
4. Available Bit Rate (ABR) is used for traffic with unpredictable rate requirements, although a minimum cell rate may be imposed; ABR is ideal for many data transfer applications.
5. Unspecified Bit Rate (UBR) is a best effort service used for traffic that is content to use whatever network capacity remains. UBR is somewhat the ATM peer of the Internet Protocol (IP).
6. Guaranteed Frame Rate (GFR) is designed to specify a minimum cell rate (MCR) and a peak cell rate (PCR). All traffic sent above the MCR will be treated as best effort. Conformance definitions are based on frames of data, not cells.

Each category of service is customized for a particular type of traffic. Of these six categories, only one, Available Bit Rate (ABR), uses a feedback mechanism to create closed-loop congestion control. The other categories use either open-loop traffic management strategies, e.g. call admission control (see Footnote 1 on page 3), or in the case of UBR, no Quality of Service (QoS) component to traffic management. The creation of a control mechanism for a switch that can work with the closed-loop congestion control mechanism specified by the ATM Forum is the focus of this dissertation, although, as discussed in Section 1.1, the results are likely applicable beyond ATM ABR.

1.3 Available Bit Rate (ABR) Congestion Control

The ATM *Traffic Management Specification* [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[3]. RM cells moving from source to destination are called Forward RM cells, and RM cells returning to the source are called Backward RM cells.

RM cells contain fields that support two methods of feedback control. The first method, called Relative Rate Marking, uses the Congestion Indication (CI) bit and No Increase (NI) bit together for binary feedback control. Essentially, switches use these two bits to request the ABR source to increase or decrease its rate using a fixed mechanism[4]. In the second method, an explicit rate field within the RM cell can be used

[3] 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.

[4] Note that although two bits (the CI bit and the NI bit) are used, this control mechanism is usually called *binary*, since the single CI bit tends to have the dominant effect.

by switches to request a specific rate. ABR sources must abide by both the binary and the explicit rate mechanisms, adjusting their rates to that specified by the mechanism that specifies the lower rate [1]. However, since a given switch need only use one or the other, researchers tend to consider the binary and the explicit rate mechanisms separately.

1.3.1 ATM ABR Explicit Rate Congestion Control

ATM ABR explicit rate congestion control occurs as follows: ABR source S periodically inserts (at least every N_{RM} cells) resource management (RM) cells into its stream of data cells. These RM cells generally contain an explicit rate (ER) field that is initialized to the maximum possible sending rate of the source, its peak cell rate (PCR). As the RM cell moves from switch to switch, each switch can reduce the rate indicated by the explicit rate field. When the RM cell is returned to the source, the source is required to adjust its allowed cell rate (ACR), an upper bound on its sending rate, to be no greater than the rate indicated by the explicit rate field. Thus, the ACR of an ABR source equals the minimum rate allowed by the switches in the path of the flow as indicated by the most recently received RM cell.

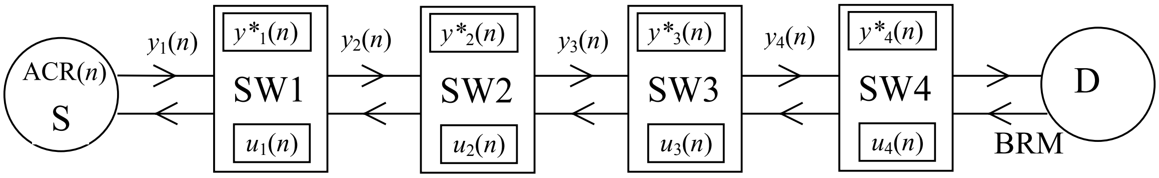

Figure 1.1 Congestion Control Mechanism From Perspective of Source/Destination Pair

Consider Figure 1.1, which shows the congestion control mechanism from the perspective of a single ABR virtual connection (VC) from source S to destination D via switches *SW1-SW4*. Assume a digital model, such that at time *n*, *S* transmits at the allowed cell rate, $\mathrm{ACR}(n)$ cells/sec (*S* is a greedy source). The arrival rate at *SW1* is $y_1(n) = \mathrm{ACR}(n - d_1)$, where d_1 is the propagation delay between *S* and *SW1*.

The output port bandwidth is $y*$. Due to the special configuration of Figure 1.1, $y*_i = y_{i+1}$, although this is generally not true. At each time *n*, each switch independently performs a congestion control calculation to produce $u_i(n)$. As a resource management cell returns from *D* to *S*, each switch examines the RM cell's explicit rate field. If the switch's current u_i is smaller than the contents of the ER field, the switch copies its current u_i into the explicit rate field. When this resource management cell returns to *S*, it contains the smallest explicit rate u_i it encountered along its path. The source's ACR is then updated, thereby creating a feedback control process.

Two key metrics for congestion control are efficiency and fairness. Efficiency means that traffic allowed into the network closely matches the resources of the network. Over-allocating the network causes delays and data loss. Under-allocating the network generally reduces the return on the network's investment by the network provider. This opportunity cost is likely eventually to be transferred to customers. Various definitions of fairness exist. Max-min fairness [4] is one frequently used definition. When all ABR connections specify a minimum cell rate equal to zero, the definition of max-min fairness is unambiguous. Consider the specific switch *SW* of Figure 1.2 carrying N ABR connections through output port *j*. These connections can be divided into two groups. The first group are the constrained connections–connections that cannot use their fair share allocated by port *j* because they are limited to a rate below their fair share elsewhere in the network[5]. The second group are the unconstrained connections– connections that can reach their fair share of bandwidth, i.e. are limited in bandwidth by the allocation provided by port *j*.

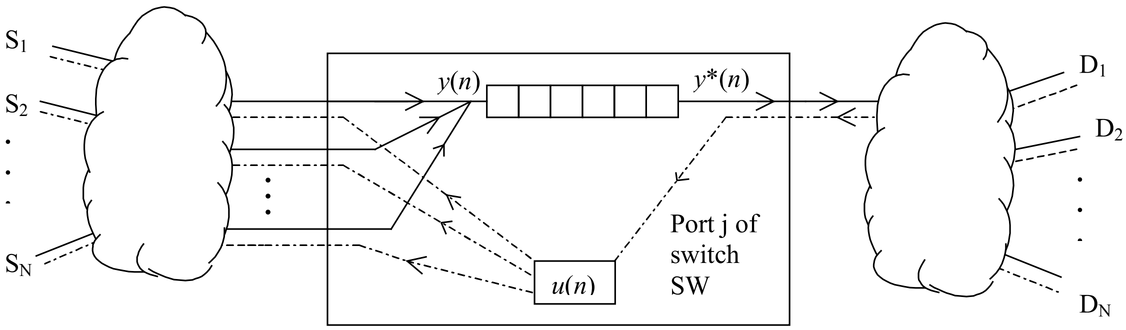

Figure 1.2 Plant from perspective of Switch Output Port

Max-min allocation occurs by giving constrained connections the bandwidth they require and splitting the remaining bandwidth evenly among the unconstrained connections. Let N_u and N_c be the number of unconstrained and constrained connections at port *j*, respectively. Further, define *C* as the total bandwidth consumed by the N_c connections and y^* as the total bandwidth available at port *j*. Then the max-min fair share bandwidth is

$$fair\ share_{\max-\min} = \frac{y^* - C}{N_u}. \tag{1.1}$$

Note that the definition of fair share depends on N_u, and the definition of N_u depends on the fair share. In many practical situations, finding the fair share is non-trivial.

[5] Note that one possible definition of max-min fairness with non-zero minimum cell rate (MCR) is to define connections with MCR greater than their fair share as a constrained connection. Generally the case with non-zero MCR is not considered here, but when it is, this definition is used.

1.3.2 Previous Contributions

The past decade has seen significant contributions to the understanding of congestion control in ATM ABR networks. Contributors include Rohrs, Berry and O'Halek from M.I.T.; Benmohamed and Meerkov from the University of Michigan; Altman, Baccelli and Bolot from INRIA (France); Altman, Basar and Srikant from the University of Illinois; Jain from the Ohio State University; Fulton and Li from the University of Texas; and Mascolo from Politecnico di Bari (Italy). These contributions are summarized in Sections 1.3.2.1 through 1.3.2.6.

The ATM Forum has also made significant contributions. The ATM Forum "is an international non-profit organization formed with the objective of accelerating the use of ATM products and services through a rapid convergence of interoperability specifications. In addition, the Forum promotes industry cooperation and awareness" [3]. The ATM Forum approved what has become the de facto 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]. This specification intentionally leaves the method by which switches determine explicit rates unspecified[6]. However, several candidate algorithms have been proposed in the ATM Forum, many by switch designers. Not too surprisingly, these designers tend to show interest in implementation-friendly solutions. Of the above-cited authors, Raj Jain clearly made the most significant contributions in the ATM Forum, and his ERICA algorithm and its derivatives are very popular. Mike Hluchyi, Andy Barnhart and Larry Roberts also made early contributions to the ATM Forum [3], [1]. The other cited author's contributions

[6] This is precisely the issue addressed in this dissertation.

tend to promote controllers well studied in the literature of control theory with many attractive analytical features. However, implementation costs rarely enter their cost functions to be optimized, and thus the resulting algorithms are often viewed as too complex by those designing "real-world" ATM network components, many of whom are represented by the ATM Forum.

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, a push towards greater complexity must be minimal. Implementation cost should be a very important consideration. It is believed that the results of this dissertation will provide performance advantages that justify their implementation costs.

1.3.2.1 Rohrs

Rohrs, Berry, and O'Halek explored binary congestion control in their 1996 article [5]. Generally, switches mark the CI bit to one when their buffers exceed some threshold and set the CI bit to zero otherwise [1]. Rohrs, et al. demonstrated that such a non-linear control scheme leads to oscillations in both arrival rates and buffer sizes. To overcome these oscillations, [5] proposes a method to communicate from the switch to the source a parameter p, $0 \leq p \leq 1$, which indicates the level of congestion at the switch. The parameter p is transmitted to the source by the switch's marking the CI bit with probability p. Thus the source receives a noisy estimate of p. With the model (almost) linearized, Rohrs et al. uses linear control theory to design a controller that demonstrates much improved performance and substantially reduces oscillations. The lessons of [5] could be applied to other binary congestion control situations.

Binary feedback mechanisms are often unfair due to the beat down effect [7]. Consider a VC that spans a relatively large number of switches. Such a VC is more likely to traverse a congested switch than a VC spanning relatively few switches. Even though the longer VC may not in any way contribute to the congestion, it is more likely to have its CI bit marked, causing it to further reduce its rate. Many schemes have been proposed to improve the fairness of binary congestion control, e.g. intelligently mark cells from VCs that exceed their fair share, but it is always possible to design an explicit rate scheme that is at least as fair and efficient as any given binary scheme.

1.3.2.2 Benmohamed and Meerkov

An important, early contribution to explicit rate congestion control comes from Lotfi Benmohamed's Ph.D. thesis while working with Semyon Meerkov at the University of Michigan. The work was published in 1993 [8]. Benmohamed makes important modeling contributions, providing detailed assumptions, including:

1. greedy sources
2. a discrete-time model
3. a fluid flow model for traffic
4. that link rates, not processing time, limit traffic throughput
5. that input traffic patterns are piecewise constant for periods of time long enough for controller transients settle between changes
6. first come, first served (FCFS) queue service

The strategy consists of designing a controller to drive the queue to some target depth.

Benmohamed and Meerkov made another contribution in 1997 [9], this time considering multiple congested nodes. Again the contribution is substantially that of modeling. In the end, through careful reasoning and imposing judicious assumptions, they essentially arrive back at the single bottleneck node case described in [8]. This contribution [9] makes a strong case for simplifying the congestion control problem to a single node study. Few investigators have deviated from this since.

A few comments are in order. First, the integral action of the queue plant necessitates control to provide stability. Second, 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. Their costly calculation requires a large matrix inversion and multiplication. Finally, this work is widely cited and many have adopted their models and conclusions.

1.3.2.3 Bolot

Jean-Chrysostome Bolot, then of The French National Institute for Research in Computer Science and Control (INRIA), appears to be the first to suggest a self-tuning regulator for congestion control of communication networks [10]. He models a one-node network with only one source and model noise. He finds even this simple model difficult to directly analyze and therefore proposes a general ARMAX model to represent the progression of the queue length.

$$A\left(z^{-1}\right)q\left(n\right)=B\left(z^{-1}\right)u\left(n\right)+C\left(z^{-1}\right)w\left(n\right) \tag{1.2}$$

where $queue(n)$ is the queue length, $u(n)$ is the requested rate at time n, and $w(n)$ is an independent and identically distributed zero-mean, Gaussian process. He suggests using Recursive Least Squares (RLS) to estimate $A(z^{-1})$, $B(z^{-1})$, and $C(z^{-1})$ and to use a set point of $queue*(n) = maxqueue/2$, where $maxqueue$ is the length of the buffer. Although his summary suggests that future work will further explore the use of self-tuning regulators in congestion control problems, no evidence of this has been found in his later publications. Instead he began work with E. Altman.

1.3.2.4 Altman et al.

Eitan Altman from INRIA has co-authored several papers relevant to ATM congestion control with individuals such as J-C Bolot (see 1.3.2.3), F. Baccelli, T. Basar, O. Ait-Hellal, and R. Srikant. The first major contribution, which came in 1993 [11], investigates a single node with a single source with unit action delay. Randomness is introduced from an available service rate that changes in time according to an unknown ARMA process. Also, noisy measurements of the queue length are assumed. The control mechanism is given by

$$u(n+1) = \max\left\{u(n) + \alpha\left(\hat{u}*(n) - u(n)\right) - \beta\left(queue_{est}(n) - queue^*\right), 0\right\} \qquad (1.3)$$

where $u(n+1)$ is the rate requested at the time n, $\hat{u}*(n)$ is the estimate of the desired rate arriving at time n, $queue_{est}(n)$ is the estimate of the queue length at time n, $queue^*$ is the desired queue length, and α and β are control parameters to be designed.

Significantly, this contribution discusses how a pure rate-matching algorithm, when $\beta = 0$, produces, in time, unacceptably long queues. More precisely, if the buffer is assumed to be infinitely large but queue lengths are lower bounded by zero, then for any finite $queue_0$,

$$P\left(queue(n) < queue_0\right) \to 0 \quad \text{as} \quad n \to \infty \tag{1.4}$$

The intuition here is that the queue integrates mismatches in the actual available rate and the requested rate. If the queue length grows because the requested rate exceeds the actual rate, the controller only acts to reduce the requested rate so as to stop the increase of the queue size. With $\beta = 0$, the controller will not further reduce the requested rate to decrease the queue size back to some desired, reasonable size.

In [12] Ait-Hellal, Altman, and Basar examine the use of a pure rate-matching algorithm where slightly less than the available (predicted) bandwidth is utilized. They show that under fairly general restrictions, under-allocating the available bandwidth, using either an additive or multiplicative constant, ensures stability in the queue length. This gives some credibility to the rate matching schemes proposed by others and proposed here (although this dissertation suggests extending this rate-matching scheme to include buffer matching. See Section 4.4).

Two more contributions came in 1995 [13] and 1996 [14]. These assume essentially the same model of [11] but propose a more complex H^∞ controller. Reference [13] assumes that the ARMA plant model parameters are known. Reference [14] assumes no knowledge of these ARMA parameters and instead of certainty equivalence, "combines identification . . . with the (H^∞) control in a novel way" [14].

In 1997 [15] and 1998 [16], Altman, Basar and Srikant admit multiple sources, each with potentially a different action delay. The service rate available to the ABR traffic is modeled with an AR process. The instantaneous cost function is given by

$$\left(queue(n)-queue^{*}\right)^{2}+\sum_{m=1}^{M}\frac{1}{c_{m}^{2}}\left(u_{m}(n)-a_{m}u^{*}(n)\right)^{2} \tag{1.5}$$

where $\{a_m\}$ allows flexibility in apportioning $u^*(n)$ to the M sources, $\{c_m\}$ differentiates relative importance to the M sources, as well as balances the priority of rate matching and queue matching. Several certainty-equivalence formulations are suggested and compared.

Note that throughout this body of work, 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.

1.3.2.5 Jain and Li

Although working separately, there has been an intersection of the most recent work of Raj Jain and his student Sonia Fahmy at the Ohio State University and San-qi Li and his student Cathy Fulton at the University of Texas, Austin.

Raj Jain made the best know 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.

The basic ERICA algorithm is characterized by calculating two rates for each VC and writing the larger rate into the explicit rate field of the VC's BRM cells. The first of the two rates calculated, *VCshare*, modifies a VC's current cell rate based on the switch's current loading. Specifically

$$loadfactor = \frac{y(n)}{y^*(n)}$$

$$VCshare_i = \frac{CurrentCellRate_i}{loadfactor}, \quad i = 1, \ldots, N,$$

that is, *VCshare* is increased when $loadfactor < 1$ and decreased when $loadfactor > 1$ for each of the *N* VCs.

The second rate calculated is the VC's *FairShare*,

$$FairShare = \frac{y^*(n)}{N}$$

where *N* is the total number of ABR VCs sharing the $y^*(n)$ of bandwidth. Note that if a VC is constrained at another point in the network, then that VC will not use its *FairShare*.

If every VC adjusted its rate to its VC share, then the allocation would be efficient ($loadfactor = 1$) but not fair. Conversely, if every VC adjusted its rate to its *FairShare*, then unless there are no VCs constrained at other points, the allocation is inefficient ($loadfactor < 1$) but fair. Thus the explicit rate given to each VC, $ERcalculated_i$, is

$$ERcalculated_i = \max\{FairShare, VCshare, MCR\}$$

$$ERcalculated_i = \min\{ERcalculated_i, y^*(n)\}$$

ERICA is computationally inexpensive to implement 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 when many considered explicit rate congestion control to be an extravagant luxury. However, further study discovered various scenarios where max-min fairness was not achieved. Several small modifications were made to address the more serious of these shortcomings.

Until recently, ERICA and ERICA+ were pure rate-matching algorithms. A recent contribution [21] acknowledged the usefulness of queue control. A modification to ERICA+ is proposed where explicit rates are multiplied by a factor corresponding to the current queue level.

In another recent contribution [22], persisting fairness concerns of ERICA+ prompted a new approach. The switch determines an effective number of sources. This effective number of sources, or 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 and marks an intersection in these two bodies of work[7].

The approach suggested by Cathy Fulton and San-qi Li, the Uniform Tracking (UT) method, assumes that one fair explicit rate results from equally dividing the contested bandwidth by the number of contesting sources. However the contested bandwidth and number of contesting sources are not found directly. Instead the fair rate is found iteratively by comparing past explicit rates to the current total input rate.

[7] The work of Fahmy et al. [22] does not mention Fulton and Li's work [24], the latter published a year before the former.

In steady state, both of Jain's algorithms [19] [22] as well as Li's UT algorithm [24] achieve fairness and efficiency if they equally divide the bandwidth available for ABR traffic among the competing sources. Since each source sends cells at no more than the minimum explicit rate specified by the switches in its path (each switch calculates its explicit rate independently), it is quite likely that a switch will carry traffic from a source constrained by another switch. Both algorithms supply the constrained sources their needed bandwidth, and, at least in steady state, equally divide the remaining bandwidth among the unconstrained sources.

1.3.2.6 Mascolo

Saverio Mascolo of Politecnico di Bari, Italy, explores congestion control using Smith Predictor principles [26]. The Smith Predictor eliminates the time delay from the closed-loop controlled system [27]. Mascolo views the response delay of arrival rates from each source to requested rates of the switch as fixed and known, $(T_1, T_2, \ldots T_N)$. He attempts to keep the queue length $x(t)$ equal to a set-point $r(t)$.

Using Smith principles, [27] finds the controller $G(s)$ such that Figure 1.3 and Figure 1.4 have the same transfer function $X(s)/R(s)$.

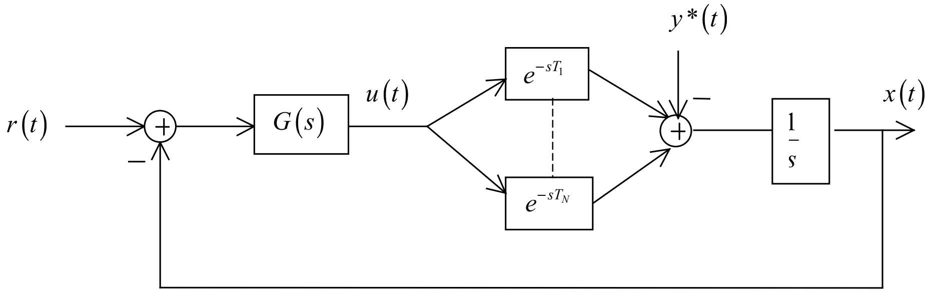

Figure 1.3 Actual Model of [27]

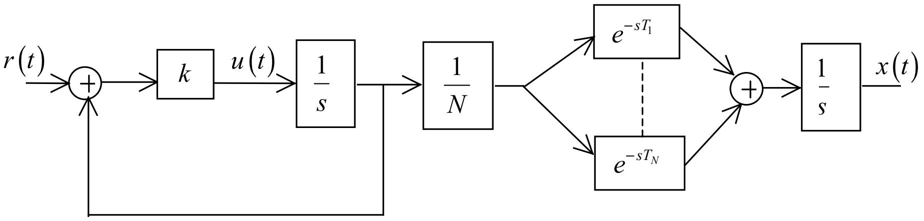

Figure 1.4 Equivalent Model of [27]

The controller $G(s)$ is found to be ([27])

$$G(s)=\frac{K/N}{1+\frac{k/N}{s}\left(N-\sum_{i=1}^{N}e^{-sT}\right)}.$$

With this model, [27] shows that the buffer never overflows or underflows.

The work of [27] shows promise but has some shortcomings. First, round trip delays are considered known (this dissertation assumes that delays must be adaptively determined). Second, [27] assumes that all ABR flows populating the queue $x(t)$ are responsive to the explicit rate $u(t)$. This appears to be a significant oversight, despite the author's acknowledgement of this possibility.

In addition to [27], Mascolo considers ATM ABR congestion control in [28] and TCP/IP control in [29]. All these works rely heavily on the Smith Predictor.

1.4 Outline of Dissertation

This introductory chapter introduces several basic congestion control concepts and reviews relevant previous contributions. The remainder of this dissertation is organized as follows: Chapter 2 selects an appropriate model for the congestion control problem. Chapter 3 identifies a control methodology for the plant specified in Chapter 2, then examines the convergence and stability properties of the selected controller. Chapter

4 introduces enhancements to the control algorithm selected in Chapter 3, in part by extending the plant model of Chapter 2. Conclusions are made in Chapter 5, as well as suggested future research directions.

CHAPTER 2 PLANT MODELING

The plant of the congestion control problem is developed in this chapter. Section 2.1 distills the description of the ATM ABR congestion control mechanism presented in Section 1.3.1 into analytic expressions. These expressions create the framework of the plant model used throughout this dissertation. Section 2.2 shows that a simplification of the plant developed in Section 2.1 inspires an adaptive control strategy taken from the literature of adaptive control. This well-understood control method is very similar to the Uniform Tracking [24] scheme. The connection between these two control schemes is explored, connecting Uniform Tracking to adaptive control theory. Section 2.3 shows that the original plant of Section 2.1, only slightly more complex than the plant of Section 2.2, inspires an improved control strategy. Section 2.4 further generalizes the plant of Section 2.1 and controller of Section 2.3. This generalized strategy justifies its added complexity with its ability to match arrival rates to available capacity for much finer time intervals. This results in much smaller queue sizes, reducing both hardware costs and delay experienced by the traffic. A further improved control strategy is presented in subsequent chapters. Extensions to the plant defined in Section 2.4 are further extended in Chapter 4 to include queue sizes and a noise disturbance. Section 2.5 explains the Blending Effect, by which cell rates are modulated by intervening switches between a source and its bottle-neck node.

2.1 Preliminaries and Plant Definition

Since each switch implements its own independent controller, one may consider the plant from the perspective of a single switch *SW*, as in Figure 2.1. 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 output port for data cells and input port for backward resource management cells.

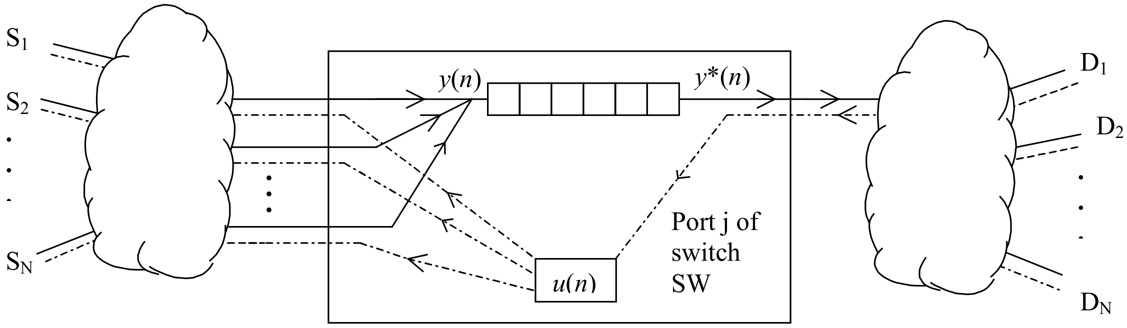

Figure 2.1 Plant from Perspective of Switch Output Port

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 (VCs) 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.

Although N sources share $y^*(n)$ of available bandwidth, it is assumed that a subset N_c of the N sources are constrained to a rate different than $u(n)$. There are at least two reasons for this possibility. First, a source may be controlled or bottlenecked by another switch along its path. Second, a source may have been guaranteed a minimum cell rate (MCR) greater than the rate assigned by port *j*, or have insufficient data to take advantage of the offered bandwidth. Thus only the $N_u \equiv N - N_c$ unconstrained sources will react to $u(n)$. These N_u sources are assumed greedy and will send cells continuously at the maximum allowed cell rate (ACR) dictated by the switch output ports through which they pass. The aggregate bandwidth of the $N_c(n)$ constrained sources[8]

$$C = \sum_{i \in N_c} y_i(n)$$

is assumed to be constant and independent of $u(n+a)$ for any positive or negative a. The switch is assumed not to initially know the value of N_u or C.

The round trip response delay for each of the N_u unconstrained sources, assumed to be equal and known[9] by the switch, is d, giving

$$y(n) = N_u \, u(n-d) + C. \tag{2.1}$$

[8] A non-constant C is introduced in Section 4.5.

[9] Section 2.4 removes this assumption on *d*.

It is assumed that C and N_u 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.

Since the minimum delay in the plant is *d*, adjustments in $u(n)$ will not be observed until time $n+d$. 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, notated as $y^*(n+d \mid n)$, may reflect both bandwidth and buffer measurements[10] made up to time *n* (this may be generated by a prediction filter as in [16]). By extension, 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 $y(n+d+V)$ decided at time *n*.

2.2 The One Parameter Plant

This section introduces an application to congestion control of an algorithm thoroughly understood in the literature of Adaptive Control. As such, its stability and convergence characteristics can be rigorously proven, even for generalized plants where the reaction times of various sources differ. Under certain conditions and assumptions, this algorithm bears a strong resemblance to the suggested algorithms of [22] and [24], briefly reviewed in Section 1.3.2.5. Contributions to the understanding of the modeling of rate control problems also appear in this section.

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

Fulton and Li propose [24] a similar plant for (2.1) that is slightly simpler than (2.1) but also has somewhat less fidelity to the real situation. At time n, define the desired fair rate at time $n-d$ as $u^*(n-d)$,

$$u^*(n-d) \equiv \frac{y^*(n)-C}{N_u}. \tag{2.2}$$

Fulton and Li implicitly define a new effective number of sources $N_{eff}(n)$ where

$$N_{eff}(n) \equiv \frac{y^*(n)}{u^*(n-d)}. \tag{2.3}$$

Thus, they define their plant as

$$y(n) = N_{eff}(n)u(n-d). \tag{2.4}$$

Note that Fulton and Li do not use forward-looking estimates of $y^*(n)$, therefore the notation $y^*(n)$ instead of $y^*(n+d \mid n)$ is used.

Assuming for now that the plant in (2.4) is a valid model, a simple Minimum Prediction Error Adaptive Controller (Direct Approach) [63] can be created to determine, at time n, the control signal $u(n)$ that minimizes

$$E\left[\left(y(n+d)-y^*(n+d \mid n)\right)^2\right].$$

As with the design of most adaptive controllers, for the purposes of analysis, it is assumed that the parameter $\theta_o = N_{eff}$ is constant within the time interval needed to generate an estimate $\hat{N}_{eff}(n)$ with accuracy. Similar assumptions are made in future sections.

Since knowledge of d is assumed, a Normalized Least Mean Squares (NLMS) [64] formulation is possible:

$$\hat{N}_{eff}\left(n+1\right)=\hat{N}_{eff}\left(n\right)+\frac{\mu}{u^2\left(n-d\right)}u\left(n-d\right)\left(y\left(n\right)-u\left(n-d\right)\hat{N}_{eff}\left(n\right)\right) \tag{2.5}$$

$$u\left(n\right)=\frac{y^*\left(n+d\,|\,n\right)}{\hat{N}_{eff}\left(n+1\right)} \tag{2.6}$$

Update equation (2.5) converges to the desired value if $0<\mu<2$ [63]. All poles and zeros of (2.4) are at the origin, thus within the unit circle, leading to the result of Lemma 2.1.

Lemma 2.1 For the adaptive controller of (2.5) and (2.6) applied to the plant (2.4):

1. $\left\{y\left(n\right)\right\}$ and $\left\{u\left(n\right)\right\}$ are bounded sequences,
2. $\lim\limits_{n\to\infty} y\left(n\right)-y^*\left(n\,|\,n-d\right)=0$, and
3. $\lim\limits_{N\to\infty}\sum\limits_{n=d}^{N}\left[y\left(n\right)-y^*\left(n\,|\,n-d\right)\right]^2<\infty$.

Proof: See [63].

Fulton and Li [24] propose the following adaptive controller, which is quite similar to (2.5) and (2.6):

$$\hat{N}_{eff}\left(n+1\right)=\frac{y\left(n\right)}{\overline{u}\left(n\text{ - }1\right)} \tag{2.7}$$

$$u\left(n\right)=\frac{\overline{y}^*\left(n\right)}{\hat{N}_{eff}\left(n+1\right)} \tag{2.8}$$

where $\overline{u}(n-1)$ is the time average of a sequence of previous values of *u.* (Comments on the time averaging of $y^*\left(n\right)$ are made in Section 2.3). The time index of $\hat{N}_{eff}$ is chosen to be $n+1$ instead of n to maintain notational consistency with other control schemes in this dissertation. This plant model and controller are simulated extensively in [22] and [24].

Compare the controller proposed here ((2.5) and (2.6)) and that of [24] ((2.7) and (2.8)). For a moment, ignore the time averaging of $u(n-1)$ in (2.7) and map (2.7) into something similar to (2.5),

$$\hat{N}_{eff}\left(n+1\right)=\frac{y\left(n\right)u\left(n-1\right)}{u\left(n-1\right)u\left(n-1\right)}$$

$$\hat{N}_{eff}\left(n+1\right)=\hat{N}_{eff}\left(n\right)+\frac{u\left(n-1\right)}{u^{2}\left(n-1\right)}\left(y\left(n\right)-u\left(n-1\right)\hat{N}_{eff}\left(n\right)\right), \tag{2.9}$$

which is equivalent to (2.5) with $\mu=1$ and d = 1. Thus the Fulton and Li controller without the averaging of $u\left(n-1\right)$ is equivalent to the one-parameter controller, (2.5) and (2.6), where d is assumed to be 1. Define the parameter estimation error as $\tilde{N}_{eff}\left(n\right)\equiv\hat{N}_{eff}\left(n\right)-N_{eff}$. Without the averaging of $u\left(n-1\right)$, if $d\neq1$, a non-zero steady state parameter estimation error will occur:

$$\begin{aligned}\tilde{N}_{eff}\left(n+1\right)&=\tilde{N}_{eff}\left(n\right)+\frac{u\left(n-1\right)}{u^{2}\left(n-1\right)}\left(u\left(n-d\right)N_{eff}-u\left(n-1\right)\hat{N}_{eff}\left(n\right)\right)\\&=\tilde{N}_{eff}\left(n\right)+N_{eff}\frac{u\left(n-d\right)}{u\left(n-1\right)}-\hat{N}_{eff}\left(n\right).\end{aligned} \tag{2.10}$$

The averaging of $u\left(n-1\right)$ proposed by Fulton and Li should bring the parameter error closer to zero.

Fulton and Li suggest that the time average “should be taken over the maximum expected round trip delay time of the ABR sources” [24]. In this case, $u\left(n-1\right)$ should be averaged over at least d steps. The purpose of the averaging becomes clear: to make $u\left(n-d\right)/\overline{u}\left(n-1\right)\approx1$.

2.3 The Two-Parameter Plant

Now reexamine the plant model developed in (2.2) and (2.3). Solving each for $y^*(n)$,

$$y^*(n) = N_u u^*(n-d) + C = N_{eff}(n) u^*(n-d) \tag{2.11}$$

Using the assumption that N_u and C are constant in the time-scale needed for parameter convergence, then (2.11) can be understood by examining Figure 2.2.

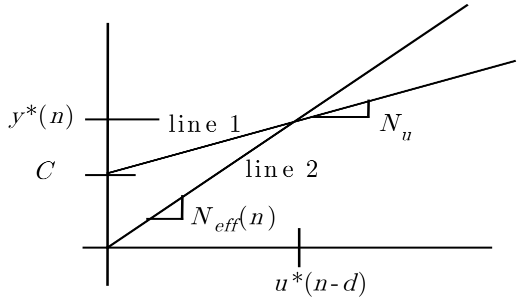

Figure 2.2 Graphical Interpretation of N_{eff}

Optimal explicit rate $u^*(n-d)$ is determined by finding the horizontal coordinate of line 1 that corresponds with its vertical component $y^*(n)$. $N_{eff}(n)$ can then be determined by calculating the slope of a line extending through the origin to the point $(u^*(n-d), y^*(n))$.

Clearly, if $y^*(n)$ varies time, $N_{eff}(n)$ is not constant, making the task of the adaptive controller of (2.7) and (2.8) (where the delay is not known and incorrectly assumed) or even (2.5) and (2.6) (where the correct delay is used) very difficult.

In short, [24], and similarly [22], avoid the task of determining the response delay d. This requires averaging $u(n-1)$ and $y^*(n)$, as shown by (2.10) and Figure 2.2. However this averaging allows $y(n)$ only to match $y^*(n)$ in the mean. The variance of

$y^*(n) - y(n)$ could be large, requiring larger queue sizes to smooth this variance. These larger queue sizes require a larger hardware cost and generally increase delay.

A two parameter controller is more appropriate when $y^*(n+d \mid n)$, and thus $N_{eff}(n)$, is not, or should not, be constrained in its variance. In such cases, the original plant model given by (2.1) is more appropriate. The controller suggested by [63] is as follows:

$$\boldsymbol{\theta} = \begin{bmatrix} N \\ \theta_{DC} \end{bmatrix}, \ \hat{\boldsymbol{\theta}}(n) = \begin{bmatrix} \hat{N}(n) \\ \hat{\theta}_{DC}(n) \end{bmatrix}, \ \boldsymbol{\varphi}(n) = \begin{bmatrix} u(n-d) \\ \phi_{DC} \end{bmatrix}$$

where ϕ_{DC} and θ_{DC} are constants such that $\phi_{DC}\theta_{DC} = C$. Then

$$\begin{aligned} y(n) &= \boldsymbol{\theta}^T \boldsymbol{\varphi}(n), \\ \hat{y}(n) &= \hat{\boldsymbol{\theta}}(n)^T \boldsymbol{\varphi}(n), \\ e(n) &= y(n) - \hat{y}(n). \end{aligned}$$

The parameter estimate vector is updated as follows:

$$\hat{\boldsymbol{\theta}}(n+1) = \hat{\boldsymbol{\theta}}(n) + \frac{\mu \boldsymbol{\varphi}(n)}{\boldsymbol{\varphi}(n)^T \boldsymbol{\varphi}(n)} \left[y(n) - \hat{\boldsymbol{\theta}}(n)^T \boldsymbol{\varphi}(n) \right] \tag{2.12}$$

with the adaptive gain μ given by $0 < \mu < 2$, and the control law given by

$$u(n) = \frac{y^*(n+d \mid n) - \hat{\theta}_{DC}(n+1)\phi_{DC}}{\hat{N}(n+1)}. \tag{2.13}$$

Define the parameter estimate error as $\tilde{\boldsymbol{\theta}}(n) \equiv \hat{\boldsymbol{\theta}}(n) - \boldsymbol{\theta}$. Then from (2.12),

$$\left\| \tilde{\boldsymbol{\theta}}(n+1) \right\|^2 = \left\| \tilde{\boldsymbol{\theta}}(n) \right\|^2 + [-2+\mu] \frac{\mu e^2(n)}{\boldsymbol{\varphi}(n)^T \boldsymbol{\varphi}(n)}. \tag{2.14}$$

Therefore the parameter error power $\left\|\tilde{\boldsymbol{\theta}}(n)\right\|^2$ never increases, since the right most term of (2.14) is negative. The parameter error $\tilde{\boldsymbol{\theta}}(n)$ converges to zero if the signal $y^*(n+d \mid n)$, and thus $u(n)$, is persistently exciting[11].

One more check must be made before declaring the controller (2.12) and (2.13) stable. Specifically, the inverse function mapping $y(n)$ to $u(n-d)$ must be stable [63] so that the control law produces well-behaved $u(n)$. From (2.1),

$$u(n-d) = -\frac{C}{N_u} + \frac{y(n)}{N_u},$$

and since both C and N_u are finite by assumption, stability is given by a minor generalization of Lemma 2.1, thus (2.12) and (2.13) provide a stable controller.

2.4 The Multi-Parameter Plant

A more realistic plant than that given by (2.1) would have sources responding to a switch's explicit rate $\{u\}$ with varying amounts of delay. 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 at time $n-m$ to the time *j* measures this explicit rate as its input rate from S_i. Reaction delays 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. It is assumed that C, b_0, b_1,…, b_{dB} remain constant for periods of time

[11] There are various definitions of persistent excitation. In this dissertation, an adaptive filter input signal is persistent exciting if its auto-correlation is full rank. See [63] for a full discussion.

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

$$y(n) = b_0 u(n-d) + \cdots + b_{dB} u(n-d-dB) + C \tag{2.15}$$

$$y(n) = B(z^{-1}) u(n-d) + C \tag{2.16}$$

$$y(n) = \mathbf{B}^T \mathbf{u}(n-d) + C \tag{2.17}$$

where $\mathbf{B} \equiv [b_0, b_1, ..., b_{dB}]^T$ and $\mathbf{u}(n) \equiv [u(n), u(n-1), ..., u(n-dB)]^T$. Note that for convenience, filters in z^{-1} and time sequences in n are mixed in expressions, e.g. (2.16). Matrix notation is also used. Equations (2.15), (2.16), and (2.17) are equivalent. This plant is a direct extension of the plants validated by simulation in [22] and [24]. Simulations of this plant, in combination with the controller of Chapter 3, appear in Chapter 4.

Defining the plant as (2.17) leads to a generalized controller that is a direct extension to the two-parameter controller presented in Section 2.3. Define

$$\hat{\boldsymbol{\theta}}^T(n) = \left[\hat{b}_0(n), \hat{b}_1(n), \ldots, \hat{b}_{dB}(n), \hat{\theta}_{DC}(n)\right],$$

$$\boldsymbol{\varphi}(n) \equiv \left[u(n), u(n-1), ..., u(n-dB), \phi_{DC}\right]^T$$

and reuse (2.12) to perform updates on the parameter estimates, copied again here:

$$\hat{\boldsymbol{\theta}}(n+1) = \hat{\boldsymbol{\theta}}(n) + \frac{\mu \boldsymbol{\varphi}(n)}{\boldsymbol{\varphi}(n)^T \boldsymbol{\varphi}(n)} \left[y(n) - \hat{\boldsymbol{\theta}}(n)^T \boldsymbol{\varphi}(n)\right] \tag{2.18}$$

The control law is then given by

$$\boldsymbol{\varphi}(n)^T \hat{\boldsymbol{\theta}}(n+1) = y^*(n+d \mid n),$$

or equivalently

$$u(n) = \frac{y^*(n+d \mid n) - \hat{b}_1(n+1)u(n-1) - \cdots - \hat{b}_{dB}(n+1)u(n-dB) - \hat{\theta}_{DC}(n+1)\phi_{DC}}{\hat{b}_0(n+1)}. \quad (2.19)$$

Clearly (2.13) is simply (2.19) with $\hat{N}(n+1) = \hat{b}_0(n+1)$ and

$$b_1 = b_2 = \ldots = b_{dB} = \hat{b}_1(n+1) = \hat{b}_2(n+1) = \ldots = \hat{b}_{dB}(n) = 0\,.$$

The parameter vector error power $\left\|\tilde{\boldsymbol{\theta}}(n+1)\right\|^2$ will converge to a constant, as is shown in (2.14), and will further converge to zero if the signal $u(n-d)$ is persistently exciting [63].

However, only if the inverse mapping from $y(n)$ to $u(n-d)$ is stable can the controller be deemed suitable. This requires that $\phi_{DC}\theta_{DC}(=C)$ is finite (true by assumption) and that zeros of the plant (2.16) lie within the unit disk $|z|<1$. Clearly there are situations where this is not so, e.g. if $b_0 = 1, b_1 = 3;\ b_2, b_3, \cdots B_L = 0$, (2.16) has a zero at $z = -3$. In such a case, the generated $u(n)$ may not behaved well. On a related note, the algorithm requires that $b_0 \neq 0$, i.e. the minimum delay *d* must not be underestimated. Underestimating *d* also has the effect of placing a plant zero outside the unit disk and thus produces an unstable controller. The practical consequences of these limitations are addressed in Chapter 3.

2.5 The Blending Effect of Queues

The plant model (2.15) implicitly assumes that cells that leave a source with a specific rate will arrive at a bottleneck port *j* at the same rate, i.e. cell rates are unchanged or unmodulated by intervening ports. This assumption is not appropriate in every scenario. This section reveals how intervening ports might modulate rates. However, including this mechanism in the plant model involves significant, perhaps intractable, complexity. The specific rate-modulating mechanism described here is the Blending Effect. However, as discussed in Section 2.5.3, this effect is not included in the model used in the following chapters.

The following simple example illustrates the Blending Effect. Consider an output port *p* that has a constant output service rate y_p^* cells/sec. Let port *p* service two virtual connections, VC_1 and VC_2, having fixed input rates $y_{1,in,p} = (1/2) y_p^*$ and $y_{2,in,p} = (3/4) y_p^*$. The resulting ratio of cells in p's buffer is approximately 2/5 and 3/5 for VC_1 and VC_2 respectively. The output rates are therefore $y_{1,out,p} = (2/5) y_p^*$ and $y_{2,out,p} = (3/5) y_p^*$.

A number of initial observations can be drawn from this example. First, the rates of both virtual connections are modified by port *p*, i.e., $y_{1,in,p} \neq y_{1,out,p}$, $y_{2,in,p} \neq y_{2,out,p}$. Second, the output rate of VC_1, $y_{1,out,p}$, is as much a function of $y_{2in,p}$ as it is $y_{1,in,p}$. By extension, $y_{1,in,p}$ is a function of the rates of all of the virtual connections sharing all of the upstream ports with VC_1, and likewise for VC_2. Third, if the source rate of VC_1 is controlled by a downstream port *j* ($\neq p$), and if VC_2 does not continue through *j*, port *j* has no way to directly measure or directly control VC_2, nor the impact of VC_2 on $y_{1,in,j}$. These observations reveal the significant difficulty that the Blending Effect poses to the modeling of explicit rate congestion control.

2.5.1 Modeling the Blending Effect

This section presents a model for the Blending Effect. The Blending Effect was first reported in [30], which included an approximate model. The model introduced here does not rely on some of the limiting assumptions of [30]. This algorithm takes as an input a $N \times 1$ vector $\mathbf{in}(n)$, which represents the number of cells from each of the N virtual connections that enter the queue at time n. The output of the algorithm is a $N \times 1$ vector $\mathbf{out}(n)$, which represents the number of cells from each of the N virtual connections that exit the queue at time n.

The Blending Effect algorithm is in discrete time and counts cells and cell rates as continuous (i.e. non-discrete) quantities. Queue service is first in-first out (FIFO). Let port *p* have an output rate of $y^*(n)$. The sample time is Δ seconds, therefore the input and output rates are $\mathbf{in}(n)/\Delta$ and $\mathbf{out}(n)/\Delta$ cells per second.

Section 2.5.1.1 presents the Blending Effect algorithm in pseudocode. Additional variable definitions and discussion follows in Section 2.5.1.2.

2.5.1.1 Pseudocode for Blending Effect

```
/* Initialize.*/
$n = 0$
$c = 0$
for each $n$
{
        $c = c + 1$
        $\mathbf{Cells}(c) = \mathbf{in}(n)$
        $remain(c) = \sum_{x=1}^{N} \mathbf{CELLS}_{xc}$
        $\mathbf{outcells} = \mathbf{0}$
        $M = y^*(n)\Delta$
        while $(M > 0)$
        {       if $(remain(1) \le M)$
```

```
                {       $\mathbf{outcells} = \mathbf{outcells} + \frac{remain(1)}{\sum_{x=1}^{N} \mathbf{CELLS}_{x1}} \mathbf{Cells}(1)$
                        $M = M - remain(1)$
                        if $(c > 1)$
                        {       for  $t = c, c-1, \ldots, 2$
                                {       $\mathbf{Cells}(t-1) = \mathbf{Cells}(t)$
                                        $remain(t-1) = remain(t)$
                                }
                        }
                        else
                        {   $\mathbf{Cells}(1) = \mathbf{0}$
                            $M = 0$
                        }
                        $c = c - 1$
                }
                else
                {       $\mathbf{outcells} = \mathbf{outcells} + \frac{M}{\sum_{x=1}^{N} \mathbf{CELLS}_{x1}} \mathbf{Cells}(1)$
                        $remain(1) = remain(1) - M$
                        $M = 0$
                }
        }
        $\mathbf{out}(n) = \mathbf{outcells}$
}
```

2.5.1.2 Discussion of Pseudocode for Blending Effect

The cells in the queue are represented by $N \times c$ matrix $\mathbf{CELLS}$. Row i of $\mathbf{CELLS}$, notated as the $N \times 1$ vector $\mathbf{Cells}(i)$, corresponds to the input cell vector $\mathbf{in}(n-(i-1))$, $1 \le i \le c$. The scalar c indicates the number of partial and complete input intervals remaining in the queue. Therefore c fluctuates depending on the rate of cell arrival as compared to the cell service rate. The operations at time *n* begin by appending $\mathbf{in}(n)$ to $\mathbf{CELLS}$, thereby incrementing *c*. Then $M = y^*(n)\Delta$ cells, if available, are removed from "the front of" $\mathbf{CELLS}$. At time n, some of the cells located at the front of the queue, i.e. in $\mathbf{Cells}(1)$, may have exited the queue at time $n-1$. Therefore a count of the remaining cells represented by $\mathbf{Cells}(1)$ is maintained as the scalar $remain(1)$ ($remain(c)$ is actually calculated when $\mathbf{in}(n)$ becomes $\mathbf{Cells}(c)$). The $N\times 1$ vector **outcells** counts the number of cells from each virtual connection as they exit. The specific manner in which the *M* cells are removed from $\mathbf{CELLS}$ is given by the above pseudocode in Section 2.5.1.1. Basically, $\mathbf{CELLS}$ keeps track of the relative population of each Virtual Circuit's cells throughout the queue.

2.5.2 Simulation

The following simple example demonstrates the Blending Effect as it is characterized by Section 2.5.1.1. Consider the case where only two virtual connections, VC_1 and VC_2, share port *p*. Port *p* has a constant service rate $y_p^*(n)=1$ million cells per second (Mcps). VC_1 presents a constant input load and VC_2 presents a sinusoidal load to port *p*:

$$y_{1,in,p}(n)=500 \text{ Kcps},$$

$$y_{2,in,p}(n)=\left(1+\sin\left(\pi n/20\right)\right)500 \text{ Kcps}.$$

The average load provided by VC_1 and VC_2 is equal to the service rate $y_p^*(n)$. The input and output rates are shown together for VC_1 in Figure 2.3 and for VC_2 in Figure 2.4. Due to the Blending Effect, the output rates depart significantly from their respective input rates, especially for VC_1.

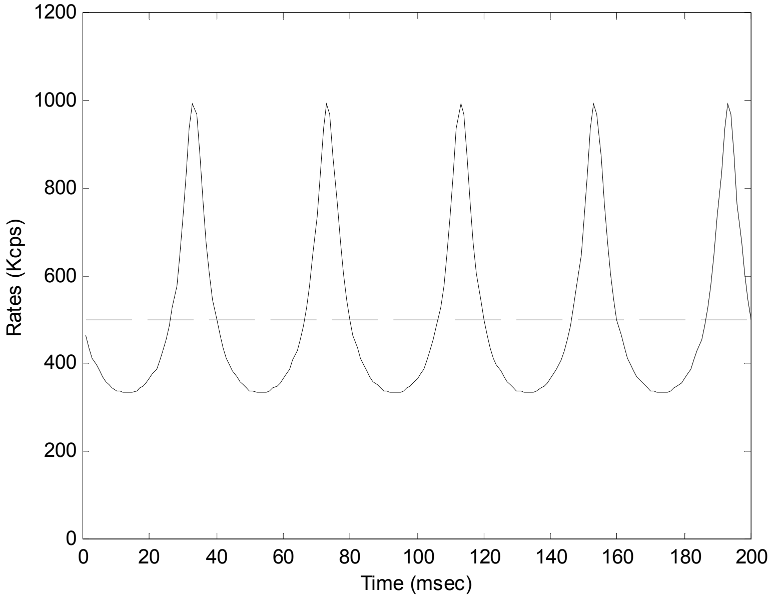

Figure 2.3 Input (dashed line) and Output (solid line) Rates for VC_1. The disparity is the result of the Blending Effect.

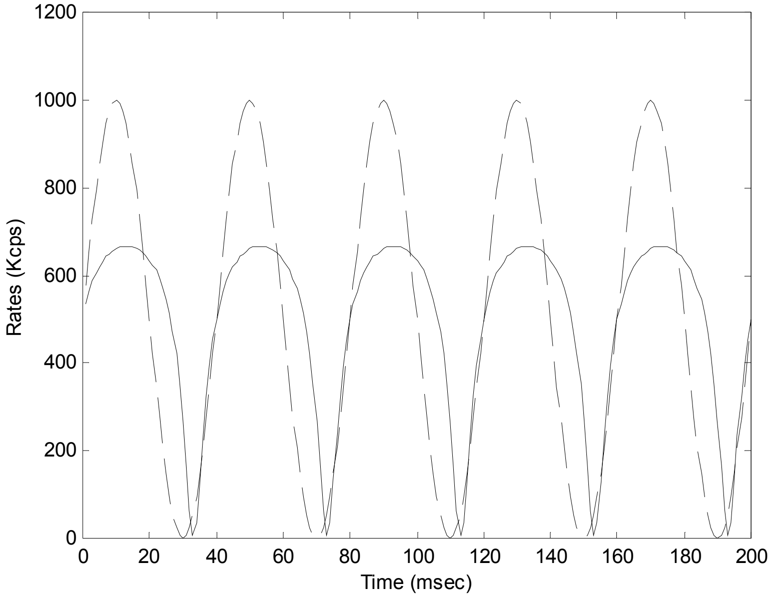

Figure 2.4 Input (dashed line) and Output (solid line) Rates for VC_2. The disparity is the result of the Blending Effect.

2.5.3 Discussion

Clearly the Blending Effect presents an enormous challenge to explicit rate congestion control. The algorithm of Section 2.5.1.1 characterizes the Blending Effect as it applies to one port, e.g. port *p*. However, before output rates of port *p* can be calculated, the input rates must be known. Generally these input rates are the output rates of ports immediately preceding port *p*. As the Blending Effect is presumably present in these ports as well, the algorithm of Section 2.5.1.1 must be executed for these ports before it is run for port *p*. By iterative logic, it appears that the model for the input rates of *p* may include a very large, perhaps intractable, number of virtual connection source rates and port rate blendings. Further, port *j* is very unlikely to measure or learn all of this information. Efforts to meaningfully model the effect from other ports as a disturbance has yet to show promise.

Consider the alternative of ignoring the Blending Effect altogether. Loosely stated, if port *j* is the sole bottleneck port for VC_1, the ports between VC_1's source and port *j* should presumably have adequate resources to serve VC_1. When this is not true, then it may be appropriate to move the designation of "controlling port for VC_1" to the port causing the blending. In other words, instead of augmenting the plant model of (2.15) to incorporate the Blending Effect, allow the plant model to rapidly add and remove responsive flows. Faster adaptive identification employed by the controller results in improved tracking of these time-varying parameters.

Furthermore, it may be argued that ports will by design conservatively allocate less bandwidth to ABR traffic than is actually available, as suggested by [11] and [12]. If ABR cells are served in a separate buffer from other service category cells, extremely short ABR cell queues should result. As the length of queue depths diminish, the associated Blending Effect also diminishes[12].

The Blending Effect can be diminished by lengthening sample intervals. In steady state, VC_1's input rate to a port must equal its output rate over large intervals of time. Another possible solution, although protocol specific to ATM ABR, is to avoid direct measurement of input cell rates. Resource management cells include a field that specifies the allowed cell rate of the source at the time the resource management cell was created. This field will be unaffected by the Blending Effect. Port j can then determine its input rate by summing the field values read from the RM cells of distinct virtual connections.

For these reasons, combined with the intractability problem, the Blending Effect is ignored for the remainder of this dissertation. Incorporating the Blending Effect into the results that follow would be a significant future contribution.

2.6 Summary

In this chapter, Minimum Prediction Error Adaptive Controllers are used as congestion control algorithms for ATM ABR traffic. A one-parameter controller was developed in Section 2.2 and shown to converge when the bandwidth available for ABR traffic is constant. The previously published algorithm of [24] and [22] is shown to be an approximation to the one-parameter controller. Section 2.3 introduces a mechanism for directly estimating and removing constrained source traffic. This improves convergence

[12] Note that Section 4.4 attempts to keep queue depths at a targeted, non-zero value.

when the available bandwidth for ABR traffic changes in time. Section 2.4 generalized the controller of Section 2.3 to the case of sources with non-identical response delays. This generalized controller of Section 2.4 can be proven to be stable only under certain conditions. Developing an improved controller is the topic of the next chapter. Section 2.5 explores the Blending Effect. Much of the material presented in this chapter is published as [23].

CHAPTER 3 A CONTROLLER FOR CONGESTION CONTROL

This chapter identifies a controller to operate in the congestion control plant developed in Chapter 2.

Section 3.1 examines several control strategies with regard to their appropriateness for controlling the congestion control plant. One of the strategies examined is selected as the best choice. The convergence and stability properties of this chosen control strategy is examined in Section 3.2.

3.1 Selection of Controller for the Congestion Control Plant

In the previous chapter, several models for the rate-controlled data network plant are discussed. These models differentiate themselves in part by the amount of detail used to represent sources' response delay to explicit rate assignments. For each of the models, an adaptive, indirect controller that inverts the estimate of the plant is suggested. The most detailed model is presented in Section 2.4, where the response delay of each source is explicitly expressed. It is noted in Section 2.4 that the multi-parameter control scheme (2.18) and (2.19) requires a minimum-phase plant assumption. This assumption has little basis in reality, and violation of this assumption causes undesirable results. Therefore the focus of present section is finding control strategies that effectively control non-minimum phase and minimum-phase plants alike.

This section begins with a review of the multi-parameter plant presented in Section 2.4. Several control strategies are proposed and evaluated. The most attractive control scheme is presented in Section 3.1.4. Stability, convergence and convergence rate issues are presented in Sections 3.2 and 4.3.

3.1.1 Plant Definition (Review)

This subsection briefly restates the plant presented in Section 2.4.

Since each switch implements its own, independent controller, the plant may be considered 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.

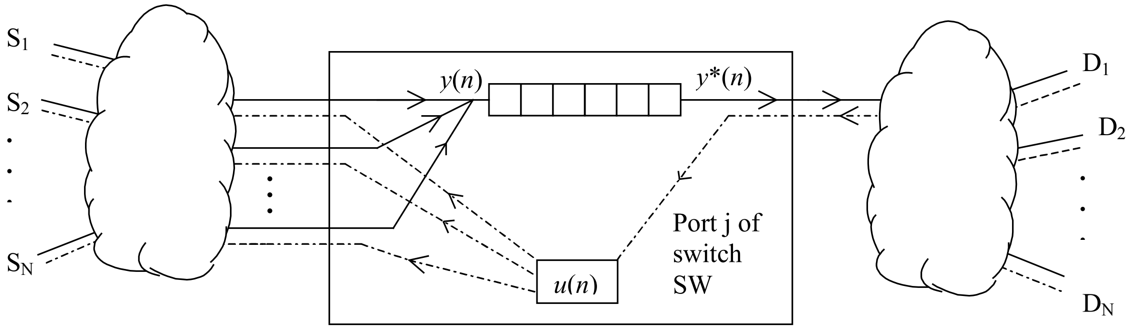

Figure 3.1 Plant from Perspective of Switch Output Port

Output port j observes changes to its input rate $y(n)$ as various sources (S_i) reacts 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 at time $n-m$ to the time *j* measures this explicit rate as its input rate from S_i. Reaction delays 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. It is assumed that C, b_0, b_1,..., 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

$$y(n) = b_0 u(n-d) + \cdots + b_{dB} u(n-d-dB) + C \tag{3.1}$$

$$y(n) = B(z^{-1}) u(n-d) + C \tag{3.2}$$

$$y(n) = \mathbf{B}^T \mathbf{u}(n-d) + C \tag{3.3}$$

where $\mathbf{B} \equiv [b_0, b_1, ..., b_{dB}]^T$ and $\mathbf{u}(n) \equiv [u(n), u(n-1), ..., u(n-dB)]^T$. Note that for convenience, filters in z^{-1} and time sequences in *n* are mixed in expressions, e.g. (2.16). Matrix notation is also used. Equations (2.15), (2.16), and (2.17) 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 $y^*(n+d \mid n)$, may reflect both bandwidth and buffer measurements[13] made up to time *n* (this may be generated by a prediction filter as in [16]). By extension, 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 $y(n+d+V)$ known at time *n*. Rates $u(n)$, $y(n)$, and $y^*(n)$ are in units of cells per second.

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

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

3.1.2 Control of Stable Non-Minimum Phase Plants

To simplify the presentation of basic concepts, consider the plant (2.16) with zero DC offset, i.e. $C=0$,

$$y(n)=B\left(z^{-1}\right)u(n-d). \tag{3.4}$$

In Section 3.1.4.2, a coefficient, the DC tap, is added to the identification filter for the purposes of matching DC offsets, thereby extending the following comments.

The plant (3.4) is an FIR filter $B\left(z^{-1}\right)$ and is thus Bounded Input/Bounded Output (BIBO) stable. The controller proposed in Section 2.4 cancels the dynamics of the plant by placing controller poles where plant zeros are located (all plant poles are at the origin). One of the assumptions required for stable operation of such a controller 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 noted in Section 2.4, 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. Section 3.1.2 reviews some of the better-known NMP-plant controllers and discusses their appropriateness for the explicit rate congestion control problem.

3.1.2.1 Minimum Variance Control Law for NMP Plants

The Minimum Variance Control Law for NMP Plants was introduced by Grimble in 1981 [35]. The following year, the same author proposed a self-tuning version [36]. A direct (or implicit) adaptive controller, where controller parameters instead of plant parameters are directly identified, was significantly extended by Niederlinski and Moscinski in 1988 [37]. For plant (3.4), Grimble's indirect and Niederlinski's direct controllers reduce to the same controller.

It is convenient to separate $B\left(z^{-1}\right)$ into two polynomals: $B^{+}\left(z^{-1}\right)$ includes all minimum-phase zeros of $B\left(z^{-1}\right)$ and $B^{-}\left(z^{-1}\right)$ includes all non-minimum-phase zeros of $B\left(z^{-1}\right)$.

$$y(n)=B^{+}\left(z^{-1}\right)B^{-}\left(z^{-1}\right)u(n-d) \tag{3.5}$$

Further, define a reflection polynomial of $B^{-}\left(z^{-1}\right)$

$$\tilde{B}^{-}\left(z^{-1}\right)\equiv z^{-dB^{-}}B^{-}(z) \tag{3.6}$$

If z_0 is a root of $B^{-}\left(z^{-1}\right)=0$, z_0^{-1} is a root of $\tilde{B}^{-}\left(z^{-1}\right)=0$.

The Generalized Minimum Variance Control Law for (3.5) is ([35])

$$u(n)=\frac{y^{*}(n+d\,|\,n)}{B^{+}\left(z^{-1}\right)\tilde{B}^{-}\left(z^{-1}\right)} \tag{3.7}$$

Substituting (3.7) into (3.5), the closed loop response is

$$y(n)=\frac{B^{+}\left(z^{-1}\right)B^{-1}\left(z^{-1}\right)y^{*}(n\,|\,n-d)}{B^{+}\left(z^{-1}\right)\tilde{B}^{-}\left(z^{-1}\right)}=\frac{B^{-}\left(z^{-1}\right)}{\tilde{B}^{-}\left(z^{-1}\right)}y^{*}(n). \tag{3.8}$$

$B^{-}\left(z^{-1}\right)/\tilde{B}^{-}\left(z^{-1}\right)$ has the property that

$$\left|\frac{B^{-}\left(z^{-1}\right)}{\tilde{B}^{-}\left(z^{-1}\right)}\right|=1, \tag{3.9}$$

and is called an all-pass filter; see Figure 3.2 for an example with $B(z)=z^{2}-\sqrt{8}z+4$.

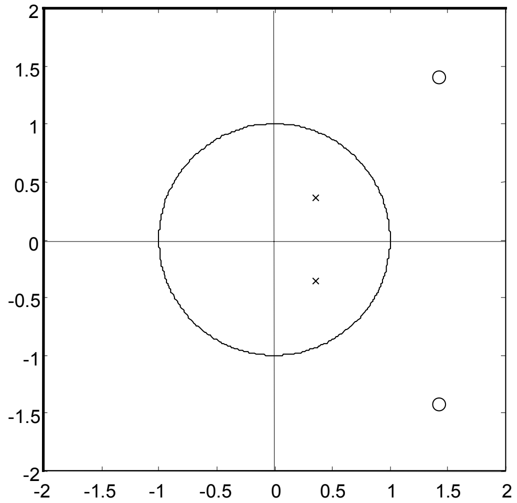

Figure 3.2 Poles and Zeros of $B^{-}\left(z^{-1}\right)/\tilde{B}^{-}\left(z^{-1}\right)$ with $B(z)=z^{2}-\sqrt{8}z+4$

The controller of (3.7) places closed-loop poles to cancel the minimum-phase zeros of the plant. Plant zeroes that are not minimum-phase are not cancelled. Instead, reflection poles are placed such that the reflection poles and non-minimum-phase plant zeros form an all pass filter, $B^{-}(z^{-1})/\tilde{B}^{-}(z^{-1})$. Note that Section 2.4 developed a controller very similar to (3.7). In fact, given the assumption of Section 2.4 that the plant be minimum-phase, i.e. $B^{-}\left(z^{-1}\right)=1$, the controller (3.7) reduces to the same multi-parameter controller of (2.19), repeated here:

$$u(n)=\frac{y^{*}(n+d\mid n)-\hat{b}_{1}(n+1)u(n-1)-\cdots-\hat{b}_{dB}(n+1)u(n-dB)-\hat{\theta}_{DC}(n+1)\phi_{DC}}{\hat{b}_{0}(n+1)}$$

The simple appearance of (3.7) masks a significant complication: the need for a root-finding algorithm to factor $B\left(z^{-1}\right)$ into $B^{-}\left(z^{-1}\right)$ and $B^{+}\left(z^{-1}\right)$. A survey of numerical algorithms that find the roots of a polynomial can be found in [38], and include the Secant method, False Position method, and the Newton-Ralphson method. High levels of precision generally require that the algorithm be run twice, once to find the general range of each root, the second time to find the location precisely. Each located root must be factored from the remaining polynomial before finding the next root.

This process is computationally expensive. Further, root location techniques are very sensitive to polynomial coefficients. Small inaccuracies in the estimate of $B\left(z^{-1}\right)$, common in many estimation techniques, could result in considerably inaccurate root calculations. Many simulations verify that the generalized minimum variance controller is not a good choice for explicit rate congestion control.

3.1.2.2 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 dissertation.

The minimum-phase plant limitation described in Section 2.4 is required since violation of this condition introduces non-stable closed-loop eigenvalues into the system via the controller's poles. These controller poles are placed in series with the plant and therefore do not effect feedback control with its associated desirable qualities [34]. This begs the question of whether control of plant (3.4) can be satisfactorily accomplished without adding potentially destabilizing controller poles. Except in special cases, one finite impulse response (FIR) filter cannot perfectly invert another FIR filter–an infinite impulse response (IIR) filter is required. However, as discussed below, in many practical situations, an FIR control filter placed in series with an FIR plant produces an impulse response with attractive qualities–loosely stated, the FIR controller approximately inverts the FIR plant.

The concept of approximately inverting one FIR filter with another FIR filter is not new, e.g. [39], [59]. Yet this concept 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. To control non-minimum phase plants, a delay of V samples is added to the controlled system. Given the large phase lags inherent in non-minimum-phase plants, adding delay is a common characteristic of non-minimum phase plant control.

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 $x\left(z^{-1}\right)$ [58], is not specified until a region of convergence is specified. Let $p_{+,\max}$ be the location of the largest magnitude pole of $B^{-1}\left(z^{-1}\right)$ inside the unit circle, and let $p_{-,\min}$ be the location of the smallest magnitude pole outside the unit circle.

Consider a region of convergence for $B^{-1}\left(z^{-1}\right)$ of $\left|p_{+,\max}\right| < |z| < \left|p_{-,\min}\right|$. With this region of convergence, the impulse response is 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 $n \rightarrow \pm\infty$ [58]. As an example, consider $B\left(z^{-1}\right)=2+9z^{-1}+8z^{-2}+3z^{-3}$, which has a pair of complex minimum-phase zeros and one non-minimum-phase zero, as shown in Figure 3.3.

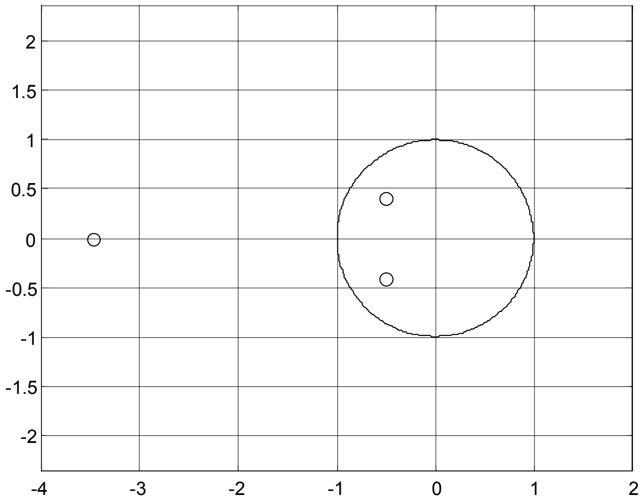

Figure 3.3 The Zeros of $B\left(z^{-1}\right)=2+9z^{-1}+8z^{-2}+3z^{-3}$

The inverse Z-transform of $B\left(z^{-1}\right)=2+9z^{-1}+8z^{-2}+3z^{-3}$, with a region of convergence that includes the unit circle, results in a two-sided time-domain filter that converges in magnitude rapidly to zero, as shown in Figure 3.4.

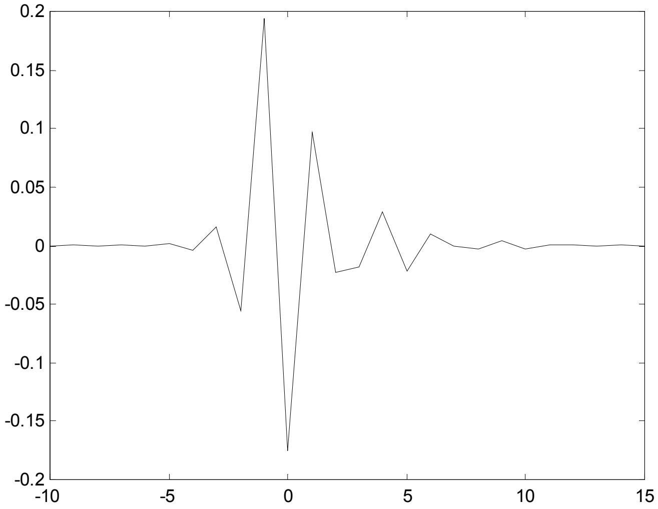

Figure 3.4 A Two-Sided, Causal Impulse Response $b^{-1}(n)$ if $B(z^{-1}) = 2 + 9z^{-1} + 8z^{-2} + 3z^{-3}$ and the Region of Convergence is Chosen to Include the Unit Circle.

If an FIR filter $b(n)^{-1}$, with impulse-response shown in Figure 3.4, is shifted in time to the right by 10 samples and truncated after 26 coefficients, the resulting causal, stable, FIR filter $q(n)$ would provide a good approximation to a delayed version of $b^{-1}(n)$; that is, $q(n) \approx b^{-1}(n-10)$ (see Figure 3.5), or

$$Q(z^{-1}) = q_0 + q_1 z^{-1} + \ldots + q_{25} z^{-25} \approx \frac{z^{-10}}{B(z^{-1})}.$$

If this delay is acceptable, $q(n)$ could be used to approximately invert the non-minimum-phase plant $B(z^{-1})$.

Now to generalize, let $Q(z)$ be a causal, FIR filter that attempts to invert a delayed version of $B(z)$. If $b(n-V)^{-1} \approx 0$ for $n < 0$ and $n > dQ$ ($V \geq 0$), then a causal $(dQ+1)$ tap FIR filter with impulse response $q(n)$ could approximate $b(n-V)^{-1}$

increasing well with increasing choices of V and dQ (if $B(z)$ has no roots on the unit circle).

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

The above explanation does not appear in [39] or [59], although the more recent [60] makes brief, similar comments.

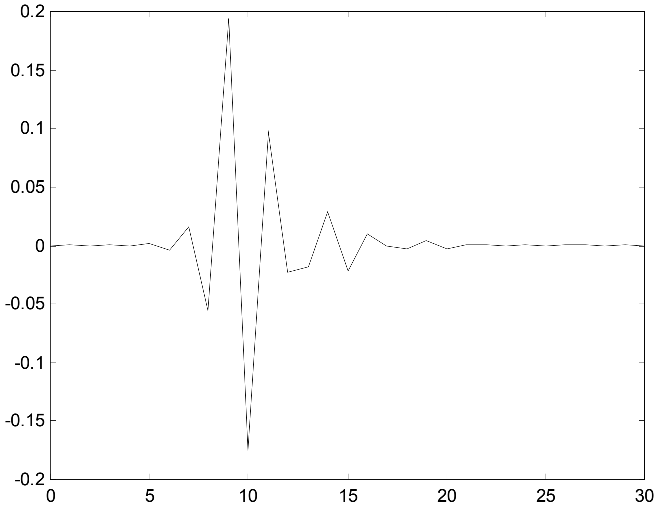

Figure 3.5 The Impulse Response of $q(n)$, a Delayed, Truncated Version of $b^{-1}(n)$

3.1.2.3 Indirect Adaptive Control Using an Approximate Inverse FIR Filter

This section briefly outlines a previously published control strategy. The controller is an example of the concept of Approximate Inverse FIR Control described in Section 3.1.2.2 and motivates the control strategies presented in Section 3.1.4.

Yahagi and Lu proposed an intuitive adaptive controller for non-minimum-phase plants in 1993 [39]. The controller consists of a time-varying, FIR filter $\hat{Q}\left(z^{-1}\right)$ (note that this notation drops the implicit dependence on time *n*), which when placed in series with the $B(z)$, approximately produces a delayed unit pulse, i.e.

$$B\left(z^{-1}\right)\hat{Q}\left(z^{-1}\right) \approx z^{-V} \tag{3.10}$$

The constant *V* is an operator-chosen delay which is non-negative, introduced "so that the accuracy of the approximate inverse system [(3.10)] becomes better" [39]. Further comments on this are made in 3.1.2.2.

The Approximate Inverse Indirect Controller for the current scenario is given as ([39]):

$$u(n) = \hat{\mathbf{Q}}(n+1)^T \mathbf{y}^*(n+d+V \mid n), \tag{3.11}$$

$$\hat{\mathbf{Q}}(n) \equiv \left[\hat{q}_0(n), \ldots, \hat{q}_{dQ}(n)\right]^T \tag{3.12}$$

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

Using the polynomial notation $\hat{Q}\left(z^{-1}\right)$ and vector notation $\hat{\mathbf{Q}}(n)$ interchangeably, the plant (3.4) and controller (3.11) give the closed loop response $y(n) = \hat{Q}\left(z^{-1}\right)B\left(z^{-1}\right)y^*(n+V \mid n-d)$. If the approximation of (3.10) is assumed to be exact, then $y(n) = y^*(n \mid n-d-V)$.

The least-squares fit to the estimated $1/\hat{\mathbf{B}}(n+1)$ is $\hat{\mathbf{Q}}(n+1)$, defined as:

$$\hat{\mathbf{Q}}(n+1) = \arg_{\mathbf{Q}}\left(\hat{\mathbf{B}}(n+1)^T \mathbf{Q} - \mathbf{e}_v\right)^T \left(\hat{\mathbf{B}}(n+1)^T \mathbf{Q} - \mathbf{e}_v\right), \tag{3.13}$$

$$\mathbf{B} \equiv \begin{bmatrix} b_0 & b_1 & \cdots & & b_{dB} & 0 & 0 & 0 \\ 0 & b_0 & b_1 & \cdots & & b_{dB} & 0 & 0 \\ \vdots & & & & & & & \vdots \\ 0 & 0 & 0 & \cdots & b_0 & b_1 & \cdots & b_{dB} \end{bmatrix}, \tag{3.14}$$

with an estimate of $\mathbf{B}$, $\hat{\mathbf{B}}(n+1)$, similarly defined. Also define

$$\hat{\mathbf{Q}} \equiv \left[\hat{q}_0, \hat{q}_1, ..., \hat{q}_{dQ}\right]^T, \;\; \mathbf{e}_v \equiv [0,0,...,0,1,0,...0]^T,$$

with the $(V+1)$th element of $\mathbf{e}_v$ equal to 1. The Wiener solution [64] solves (3.13):

$$\hat{\mathbf{Q}}(n+1) = \left(\hat{\mathbf{B}}(n+1)\hat{\mathbf{B}}(n+1)^T\right)^{-1}\hat{\mathbf{B}}(n+1)\mathbf{e}_v \tag{3.15}$$

The computational cost of evaluating (3.15) can be reduced by using a Levinson algorithm [39], but is still $O\left(dQ^2\right)$.

Operation of the Approximate Inverse Indirect Controller Algorithm consists of the following steps at each time *n*. First, update estimate $\hat{\mathbf{B}}(n)$ to $\hat{\mathbf{B}}(n+1)$ using an appropriate identification algorithm, e.g. Normalized Least Mean Squares. Second, calculate $\hat{\mathbf{Q}}(n+1)$ from (3.15), using the latest estimate of $\hat{\mathbf{B}}(n+1)$. Finally, calculate $u(n)$ from (3.11). A computationally less expensive alternative is presented next.

3.1.3 Rejected Direct Adaptive Controllers

In this section, two adaptive controllers are presented. Although neither can be used in their presented form, both controllers presented in Section 3.1.3 motivate the controller of Section 3.1.4.

Direct adaptive controllers are discussed in Sections 3.1.3 and 3.1.4. The term direct specifies that controller parameters are directly identified using an adaptive identification method. In contrast, the indirect controller of Section 3.1.2.3 identifies the plant parameters first and then derives controller parameters from the estimates of the plant parameters. The controllers in Sections 3.1.3.1, 3.1.3.2, and 3.1.4 were developed in an attempt to find a direct formulation of the indirect controller presented in Section 3.1.2.3. The motivation for finding a direct formulation is reducing computational cost by eliminating the calculation of (3.15).

3.1.3.1 A Potentially Non-Convergent Adaptive Controller

Consider a direct controller where $\hat{\mathbf{Q}}$ is directly estimated from plant input and output signals, as shown in Figure 3.6. Using the Normalized Least Mean Squares (NLMS) method [64], adaptively estimate $\hat{\mathbf{Q}}_{y*}(n)$ to obtain the ideal $\mathbf{Q}_{y*,0}$ that minimizes the least squares criterion

$$\mathbf{Q}_{y*,0} = \arg\min_{\hat{\mathbf{Q}}_{y*}} E\left[e^2(n)\right].$$

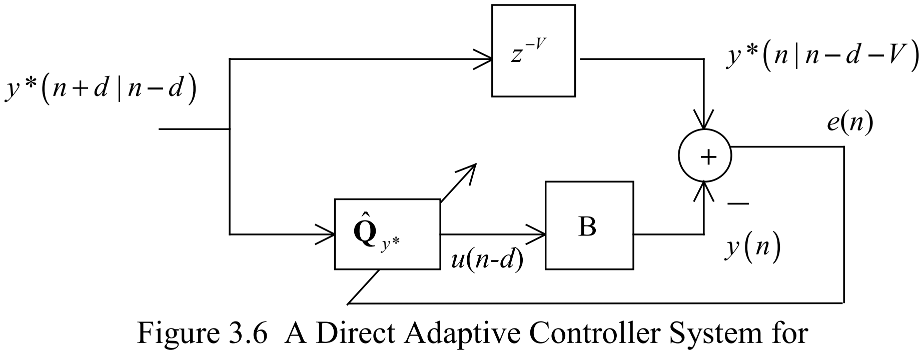

Figure 3.6 A Direct Adaptive Controller System for Controlling MA Plant That MAY NOT CONVERGE.

A careful study of Figure 3.6 shows that convergence cannot be ensured. 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 $\mathbf{y}^*(n+V\,|\,n-d)$, 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 [63]. Therefore, the controller of Figure 3.6 is disqualified as a viable explicit rate congestion controller.

3.1.3.2 An Unrealizable Controller

Consider a second control method shown in Figure 3.7 which inverts the order of $\hat{\mathbf{Q}}$ and **B** in Figure 3.6. The auxiliary signal $t(n)$ is introduced. The issue of filtering the coefficient error vector is overcome.

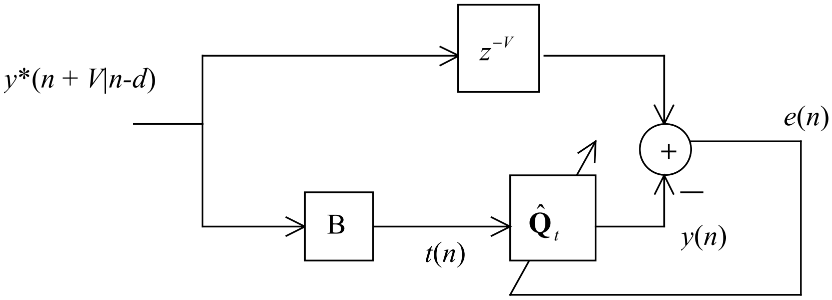

Figure 3.7 Inverting Plant and Controller, AN UNREALIZABLE CONTROLLER ($t(n)$ is not available).

Comparing Figure 3.7 with Figure 3.6, 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 [64], is

$$\mathbf{Q}_{t,0} = \mathbf{Q}_{y*,0} = \left(\mathbf{B}\mathbf{R}_{y^*}\mathbf{B}^T\right)^{-1}\mathbf{B}\mathbf{R}_{y^*}\mathbf{e}_V \qquad (3.16)$$

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 (3.16) is equivalent to (3.15).

However, there is a problem. Since $B\left(z^{-1}\right)$ is unknown, $t\left(n\right)$ cannot be created. The formulation of Figure 3.7 provides insight and intuition, but cannot be implemented.

3.1.4 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 3.7.

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 [59]. The approach here distinguishes itself from the approach of [59] 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 3.2.

3.1.4.1 A Convergent, Realizable Adaptive Control Strategy

An attractive adaptive control strategy must converge to desirable parameters and be realizable; the controllers of Sections 3.1.3.1 and 3.1.3.2 each fail in one of these respects. Yet both controllers 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 3.6). Figure 3.7 achieves this. Unfortunately $t(n)$ is not available. However, if in Figure 3.7, 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 3.8, all necessary signals are available.

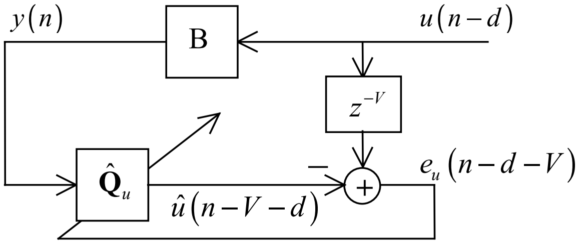

Figure 3.8 Direct Inverse Plant Modeling

Figure 3.8 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

$$\mathbf{Q}_{u,0} = \left(\mathbf{B}\mathbf{R}_u\mathbf{B}^T\right)^{-1}\mathbf{B}\mathbf{R}_u\mathbf{e}_V. \tag{3.17}$$

Although (3.17) and (3.16) are not equivalent, except for the case of $B\left(z^{-1}\right) = b_0$, both provide an approximate inverse of **B**. To better compare $\mathbf{Q}_{y^*,0}$ and $\mathbf{Q}_{u,0}$, consider the formulation of Figure 3.9.

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 $\{x\}$.

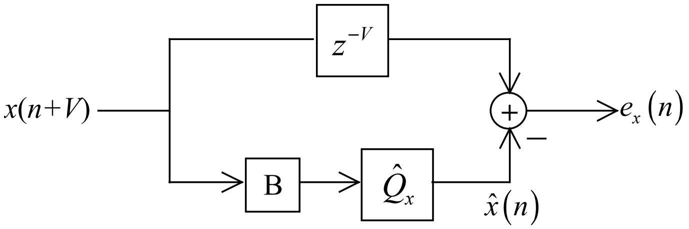

Figure 3.9 A System Where $\mathbf{Q}_{x,0}$ is a Function of $\{x\}$

For $\mathbf{Q}_{y^*,0}$, the driving signal is $\{y^*\}$, while the driving signal of $\mathbf{Q}_{u,0}$ is $\{u\}$. When $\{y^*\}$ and $\{u\}$ have similar spectral characteristics, then by (3.16) and (3.17), $\mathbf{Q}_{y^*,0} \approx \mathbf{Q}_{u,0}$. Further, if both $\mathbf{Q}_{y^*,0}$ and $\mathbf{Q}_{u,0}$ have enough taps to well match the inverse of **B** at all frequencies, assuming sufficient excitation, then $\mathbf{Q}_{y^*,0} \approx \mathbf{B}^{-1} \approx \mathbf{Q}_{u,0}$.

3.1.4.2 Removing DC Offsets With a DC Tap

For Sections 3.1.2 through 3.1.4.1, 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

$$\mathbf{y}(n) \equiv \left[y(n), y(n-1), ..., y(n-dQ), y_{DC}\right]^T \quad (3.18)$$

$$\mathbf{y}^*(n+d+V \mid n) \equiv \left[y^*(n+d+V \mid n), ..., y^*(n+d+V-dQ \mid n-dQ), y_{DC}\right]^T . \quad (3.19)$$

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 3.2.2.2.

3.1.4.3 Normalized Least Mean Square Adaptive Mechanism

Unlike $\mathbf{Q}_{y^*,0}$, $\mathbf{Q}_{u,0}$ can be estimated using the Normalize Least Mean Square algorithm [64]. At time *n*, calculate[14]

$$\hat{u}(n-d-V)=\hat{\mathbf{Q}}(n)^T\,\mathbf{y}(n) \tag{3.20}$$

$$e_u(n-d-V)=u(n-d-V)-\hat{u}(n-d-V) \tag{3.21}$$

$$\hat{\mathbf{Q}}(n+1)=\hat{\mathbf{Q}}(n)+\frac{\mu\mathbf{y}(n)}{\mathbf{y}(n)^T\,\mathbf{y}(n)}e_u(n-d-V) \tag{3.22}$$

$$u(n)=\hat{\mathbf{Q}}(n+1)^T\,\mathbf{y}^*(n+d+V\,|\,n) \tag{3.23}$$

Defining the error parameter vector $\tilde{\mathbf{Q}}(n)\equiv\hat{\mathbf{Q}}(n)-\mathbf{Q}_0$, Section 3.2 shows that if $0<\mu<2$ and certain other assumptions are made, then $\tilde{\mathbf{Q}}(n)$ converges to the zero vector. Section 3.2 also addresses global stability.

3.1.4.4 Complete Control Architecture

Figure 3.10 shows the complete control architecture. The Identification section uses NLMS adaptation to determine $\hat{\mathbf{Q}}(n+1)$ (shown with $\hat{q}_{DC}$ separated from the remaining linear taps, $\hat{\mathbf{Q}}_{lin}$, with $y_{DC}=1$, and without time index) by creating estimate $\hat{u}(n-V-d)$ using (3.20). $\hat{\mathbf{Q}}(n+1)$ is copied into the Controller, which produces $u(n)$

[14] There will be no further consideration of $\mathbf{Q}_{y^*}$ and $\mathbf{Q}_t$ introduced in Sections 3.1.3.1 and 3.1.3.2. From this point forward, the explicit subscript u in $\mathbf{Q}_{u,0}$ and $\hat{\mathbf{Q}}_u(n)$ is dropped, i.e. $\mathbf{Q}_{u,0}=\mathbf{Q}_0$ and $\hat{\mathbf{Q}}_u(n)=\hat{\mathbf{Q}}(n)$.

from the set point $y^*(n+V+d\,|\,n)$ ((3.19) using (3.11)). The Plant is represented by (3.3).

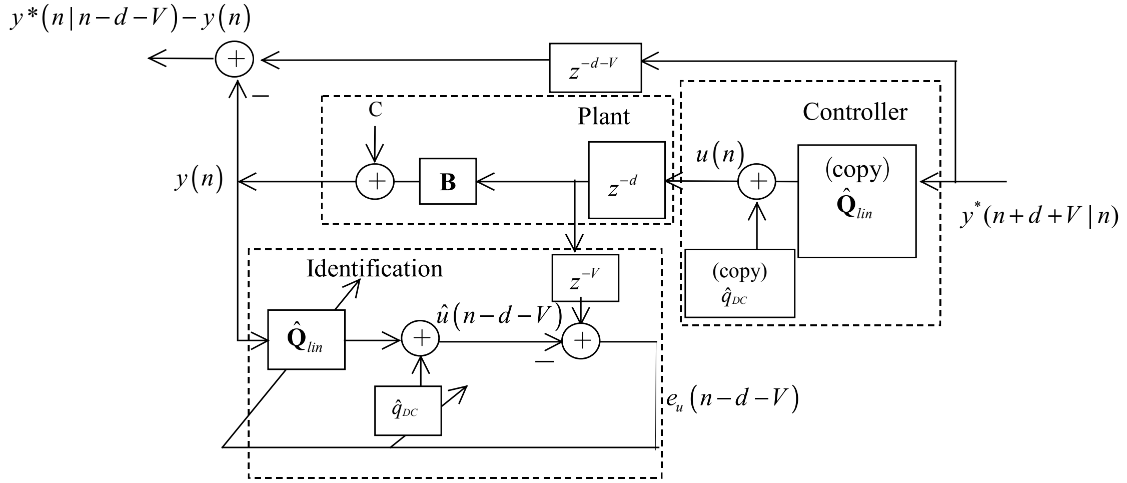

Figure 3.10 Complete Control Architecture ($y_{DC}=1$)

3.1.5 Summary

After reviewing the ATM ABR congestion control plant in Section 3.1.1, five control mechanisms based on adaptive linear control theory are presented. The first, described in Section 3.1.2.1, requires use of a polynomial root-finding algorithm and is therefore impractical. The second, described in Section 3.1.2.3, is a previously published controller that approximates the inverse of the Moving Average (MA) plant with a BIBO stable FIR filter. This indirect controller approach 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 3.1.3, but provide intuition to the proposed controller in Section 3.1.4. The selected controller can be viewed as a direct adaptive controller based on the controller of Section 3.1.2.3. This controller can employ an NLMS adaptation mechanism.

Convergence and other issues pertaining to this control architecture are discussed in Section 3.2. Other algorithm modifications to improve performance are given in Chapter 4.

3.2 Convergence and Stability

In this section, the convergence and stability properties of the controller proposed in Section 3.1.4 are examined extensively. Section 3.2 contains two separate yet complimentary proofs. The proof in Section 3.2.2 is summarized by Theorem 3.1 and Theorem 3.2. The proof in Section 3.2.3 is summarized by Theorem 3.3. Each of these two proofs demonstrates desirable qualities of the controller presented in Section 3.1.4. Each proof has its own set of assumptions.

The first proof's assumptions relate primarily to the signals associated with the controlled system. This proof focuses on the convergence of the controller parameters. Its claims on global stability are weaker than that of the second proof. The second proof's assumptions relate primarily to the plant. These assumptions take the discussion in Section 3.1.2.2 to its logical extension–a finite impulse response (FIR) filter's approximate inversion of a second FIR filter is assumed to be exact. This approximation allows a very compact yet rigorous proof of convergence and global stability.

The two proofs are necessitated by lack of the perfect modeling that attempting to invert one FIR filter with another affords. When perfect modeling is assumed, as it is in the second proof, strong results are possible. Intuitively, these results well approximate practical experience when a plant can be nearly inverted by an FIR filter, 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.

3.2.1 Plant and Controller (Review)

This section briefly reviews the plant and controller under consideration. The plant is given by (3.1) to (3.3), repeated here:

$$y(n) = b_0 u(n-d) + \cdots + b_{dB} u(n-d-dB) + C \tag{3.24}$$

$$y(n) = B(z^{-1}) u(n-d) + C \tag{3.25}$$

$$y(n) = \mathbf{B}^T \mathbf{u}(n-d) + C \tag{3.26}$$

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

Identification of the controller $\hat{\mathbf{Q}}$ employs Normalized Least Mean Square (NLMS) [64]:

$$\hat{u}(n-d-V) = \hat{\mathbf{Q}}(n)^T \mathbf{y}(n) \tag{3.27}$$

$$\hat{\mathbf{Q}}(n)=\left[\hat{q}_0(n),\hat{q}_1(n),\ldots,\hat{q}_{dQ}(n),\hat{q}_{DC}(n)\right]^T$$

$$\mathbf{y}(n)=\left[y(n),y(n-1),\ldots,y(n-dQ),y_{DC}\right]^T \tag{3.28}$$

$$e(n)\equiv e_u(n-d-V)\equiv u(n-d-V)-\hat{u}(n-d-V) \tag{3.29}$$

$$\hat{\mathbf{Q}}(n+1)=\hat{\mathbf{Q}}(n)+\frac{\mu\,\mathbf{y}(n)}{\mathbf{y}(n)^T\,\mathbf{y}(n)}e(n). \tag{3.30}$$

The controller $\hat{\mathbf{Q}}(n+1)$, as described in Section 3.1.4, comprises an adaptive FIR filter with a DC tap. Its input is the desired future input rate $\mathbf{y}^*(n\,|\,n-d-V)$ (this notation is introduced at the end of Section 2.1) and its output is the explicit rate $u(n)$.

$$u(n)=\hat{\mathbf{Q}}(n+1)^T\,\mathbf{y}^*(n+d+V\,|\,n) \tag{3.31}$$

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

The scalar d is the minimum plant delay, V is an operator chosen (non-negative) inversion polynomial delay (discussed at length in Section 3.1.2.2), 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 (3.28) so that the final tap of $\hat{\mathbf{Q}}$ becomes a DC tap $\hat{q}_{DC}$ (discussed further in Section 3.2.2.2),

$$\hat{\mathbf{Q}}(n)\equiv\left[\hat{\mathbf{Q}}_{lin}(n)^T,\hat{q}_{DC}(n)\right]^T. \tag{3.32}$$

Quantify the amount of convergence at time n by defining the parameter error vector $\tilde{\mathbf{Q}}(n)\equiv\hat{\mathbf{Q}}(n)-\mathbf{Q}_0$. For notational convenience, define another vector, identical to (3.28) except for the DC term:

$$\mathbb{Y}(n)\equiv\left[y(n),y(n-1),\ldots,y(n-dQ)\right]^T \tag{3.33}$$

Figure 3.10 shows the complete system under consideration. Plant (3.25) is controlled by Controller (3.31). The Controller is identified with (3.27) to (3.30).

The system will operate optimally if $\hat{\mathbf{Q}}(n)$ produces, at time n, 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 (3.34)). This section shows that $\hat{\mathbf{Q}}(n)$ converges to $\mathbf{Q}_0$, and that $y(n)$ appropriately emulates $y^*(n\,|\,n-d-V)$.

3.2.2 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. These results are published as [31].

The convergence analysis in this section is based on a proof of convergence for the NLMS algorithm by Tarrab and Feuer [62]. However different assumptions are made; see Section 3.2.2.1. Most notably, this proof does not require zero-mean signals, which are required by [62]. 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 moved to the Appendix. Lemmas with numbers proceeded by the letter "A", e.g. Lemma A.1, are found in the Appendix. Reference [33] contains a full, continuous presentation.

3.2.2.1 Assumptions

The following assumptions are made throughout Section 3.2.2:

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 (3.30) will occur. In implementation, it is common to impose Assumption 4 by simply skipping the adaptation of (3.30) 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 Lemma A.1), which ensures that the plant will be fully identified, allowing the discovery of a unique $\mathbf{Q}_0$; see (3.34).

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 [62]. The proof of [62] assumes that the excitation signal is Gaussian and further, if $\mathbf{y}(n)$ is a vector of the excitation signal at time *n*, $E[\mathbf{y}(n)\mathbf{y}(m)] = \mathbf{0}$ for $n \neq m$, even if $m = n+1$. This assumption tends not to be even approximately true if $\{y\}$ is the input 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 [62] further requires $\mathbf{y}(n)$ to be zero-mean. No such assumption is made here. Broadening [62] beyond the zero-mean case is the primary contribution of this proof.

3.2.2.2 DC Identification

Consider the Identification section of Figure 3.10 ((3.27) through (3.30)), redrawn as Figure 3.11.

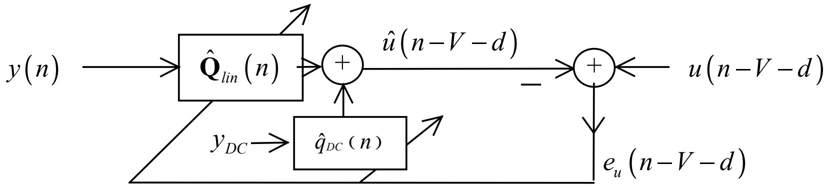

Figure 3.11 Adaptive System for Calculating $\hat{\mathbf{Q}}(n)$

From Figure 3.11, 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 [64]

$$\mathbf{Q}_0 = \mathbf{R}^{-1} E\left[\mathbf{y}(n) u(n-V-d)\right]. \tag{3.34}$$

This solution exists and is unique since $\mathbf{R}$ is full rank (Lemma A.1).

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 3.12.

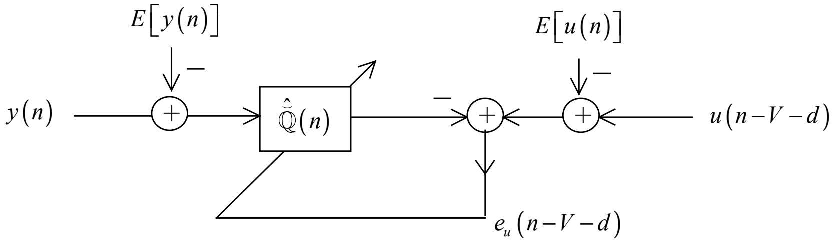

Figure 3.12 Adaptive System with Means Explicitly Removed

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

$$E\left[\breve{\mathbb{Y}}(n)\breve{\mathbb{Y}}(n)^T\right]\breve{\mathbb{Q}}_0 = E\left[\breve{\mathbb{Y}}(n)\breve{u}(n-V-d)\right]. \tag{3.35}$$

Lemma 3.1 shows that the solutions $\breve{\mathbb{Q}}_0$ and $\mathbf{Q}_0$ are closely related.

Lemma 3.1 The unique solution of (3.34) is

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

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

$$\left[\breve{\mathbb{Q}}_0{}^T, \frac{-E\left[y(n)\right]\sum_i \breve{\mathbb{Q}}_{0,i} + E\left[u(n-V-d)\right]}{y_{DC}}\right]^T \tag{3.36}$$

as the proposed solution to (3.34). Directly substituting the proposed solution into (3.34) verifies that it is indeed a solution (Lemma A.2), and thus the unique solution, completing the proof.

Lemma 3.1 demonstrates that by using a DC tap in the adaptive estimator, as in Figure 3.11, 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, the DC tap 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 3.11 and Figure 3.12– a zero-mean optimal error. By defining the optimal error

$$e*\left(n\right) \equiv u\left(n-d-V\right)-\mathbf{Q}_0\left(n\right)^T \mathbf{y}\left(n\right),$$

then with (3.36), it is easy to show (Lemma A.3) that

$$E\left[e*\left(n\right)\right]=0. \tag{3.37}$$

3.2.2.3 Other Notation

Now that the control and estimation methods have been described, what remains is to show convergence. Before proceeding with the proofs, some 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 $\mathbf{\Lambda}=\mathrm{diag}\left(\lambda_i\right)$, $i=1, \cdots, \left(dQ+1\right)$,

$$\mathbf{W}^T\mathbf{W}=\mathbf{I},\ \mathbf{W}\,\mathbf{R}\,\mathbf{W}^T=\mathbf{\Lambda}. \tag{3.38}$$

Because **R** is full-rank, **W** is full-rank. A linear transformation of the random vector $\mathbf{y}\left(n\right)$ is defined as follows.

$$\boldsymbol{\psi}(n) \equiv \mathbf{W}\mathbf{y}(n),\ \mathbf{W}^T\boldsymbol{\psi}(n) = \mathbf{W}^T\mathbf{W}\mathbf{y}(n) = \mathbf{y}(n) \tag{3.39}$$

$$\mathbf{L}(n) \equiv \mathbf{W}\tilde{\mathbf{Q}}(n), \mathbf{W}^T\mathbf{L}(n) = \tilde{\mathbf{Q}}(n) \tag{3.40}$$

$$E\left[\boldsymbol{\psi}(n)\boldsymbol{\psi}(n)^T\right] = \boldsymbol{\Lambda}\,. \tag{3.41}$$

Substituting (3.27) and (3.29) into (3.30), pre-multiplying by **W,** adding and subtracting a term, then subtracting $\mathbf{WQ}_0$ from both sides produces

$$\mathbf{L}(n+1) = \left(I - \mu\frac{\boldsymbol{\psi}(n)\boldsymbol{\psi}(n)^T}{\boldsymbol{\psi}(n)^T\,\boldsymbol{\psi}(n)}\right)\mathbf{L}(n) + \frac{\mu\;\boldsymbol{\psi}(n)e^*(n)}{\boldsymbol{\psi}(n)^T\,\boldsymbol{\psi}(n)} \tag{3.42}$$

and

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

$$\tag{3.43}$$

The following notations are used extensively:

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

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

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

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

3.2.2.4 Parameter Convergence in the Mean

In this section shows that $\lim_{n\to\infty} E[\mathbf{L}(n)] = \mathbf{0}$, and in turn, by (3.40), $\lim_{n\to\infty} E[\tilde{\mathbf{Q}}(n)] = \mathbf{0}$.

Note a few key independencies. From Assumption 2, and the fact that **W** provides one-to-one mapping, Section 5.4 of [61] 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 ((3.38) 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 3.2 $\mathbf{A}$ and $\mathbf{H}$ are diagonal matrices.

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

$$\left|\mathbf{A}_{ij}\right| = \left| E\left[\frac{\boldsymbol{\psi}(n)\boldsymbol{\psi}(n)^T}{\boldsymbol{\psi}(n)^T\boldsymbol{\psi}(n)}\right]_{ij}\right| = \left|\int_{\mathbf{X}} \frac{x_i x_j}{\|\mathbf{X}\|^2} f_{\boldsymbol{\psi}}(\mathbf{X})\, d\mathbf{X}\right|.$$

From Assumption 4,

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

From (3.41),

$$\left|A_{ij}\right| \le \begin{cases} 0 \text{ if } i \ne j \\ \dfrac{\lambda_i}{\alpha_0} \text{ if } i = j \end{cases} \tag{3.44}$$

Thus all non-diagonal elements of $\mathbf{A}$ equal zero, completing the proof (note that $\mathbf{A}$ is positive definite and thus all of its eigenvalues λ_i are positive).

Lemma 3.3 $0 < \alpha_1 < A_{ii} \le 1$

Proof:

$$A_{ii} = \int_{\mathbf{X}} \frac{x_i(n)^2}{\sum_{j=1}^{dQ+1} x_j(n)^2} f_{\boldsymbol{\psi}(n)}(\mathbf{X})\, d\mathbf{X}.$$

$$0 \le \frac{x_i(n)^2}{\sum_{j=1}^{dQ+1} x_j(n)^2} \le 1 \text{ and } 0 \le f_{\boldsymbol{\psi}(n)}(\mathbf{X}) \le 1 \text{ for any } \mathbf{X}.$$

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. $0 < f_{\boldsymbol{\psi}(n)}(\mathbf{X})$, and

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.

The main result of Section 3.2.2.4 is as follows:

Theorem 3.1 Given Assumption 1 through Assumption 4 and $0 < \mu < 2$, $\lim_{n\to\infty} E\left[\tilde{\mathbf{Q}}(n)\right] = \mathbf{0}$.

Proof: From (3.37),

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

From (3.42), 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 3.2, 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 3.3 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 (3.40) gives $\lim_{n\to\infty} E\left[\tilde{\mathbf{Q}}(n)\right] = \mathbf{0}$, thus completing the proof.

Theorem 3.1 states that given Assumption 1 through Assumption 4 (if μ is bounded as $0 < \mu < 2$) then our NLMS adaptive system of estimating $\hat{\mathbf{Q}}$ (given by (3.27) through (3.30)) converges in the mean to the ideal $\mathbf{Q}_0$ (given by Lemma 3.1). Theorem

3.1 is distinctive from the related proof of [62] in that it uses (3.37), instead of a zero-mean assumption for $\mathbf{y}(n)$, to eliminate the expectation of the second term of (3.42).

3.2.2.5 Parameter Convergence in the Mean Square

Given Theorem 3.1, a statement bounding the variance of $\tilde{\mathbf{Q}}(n)$ would give additional credibility to the proposed controller, and is the goal of this section.

The proof in [62] relies heavily on a zero-mean assumption on $\mathbf{y}(n)$, an assumption not made here. However, Lemma 3.4 shows that almost all of the terms of $\boldsymbol{\psi}(n)$ are zero-mean. Because of this, a strategy similar to that of [62] is adopted.

Lemma 3.4 Given the independence of the terms of $\boldsymbol{\psi}(n)$, (3.41) is the necessary and sufficient condition that no less than dQ of the $dQ+1$ elements of $\boldsymbol{\psi}(n)$ are zero mean.

Proof: (Sufficiency) By contradiction. For $i \neq j$,

$$E\left[\psi_i(n)\psi_j(n)\right] = E\left[\psi_i(n)\right]E\left[\psi_j(n)\right].$$

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 (3.41).

(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 (3.41), concluding the proof.

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

$$E\left[\psi_i(n)\right] = 0, i = 1, \ldots, dQ+1,\ i \neq \zeta, \tag{3.45}$$

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

Lemma 3.5 The expectation of the fifth through eighth term of (3.43) is zero.

Proof: Examine the expectation of the seventh term on the right side of (3.43), specifically the term at row *i*, column *j*:

$$\mu E\left[\frac{\sum_{k=1}^{dQ+1} L_k(n) e^*(n)\psi_i(n)\psi_j(n)\psi_k(n)}{\left(\boldsymbol{\psi}(n)^T\boldsymbol{\psi}(n)\right)^2}\right] \tag{3.46}$$

For the *k*th term of the summation, since $L_k(n)$ is independent of $\boldsymbol{\psi}(n)$, and $e^*(n)$ is independent from $\boldsymbol{\psi}(n)$, $E\left[e^*(n)L_k(n)\right]$ can be separated from the remaining terms inside the expectation of (3.46). Now, from (3.40), Assumption 2, and (3.37):

$$E\left[e^*(n)L_k(n)\right] = \mathbf{W}_k E\left[e^*(n)\right]E\left[\tilde{\mathbf{Q}}(n)\right] = 0\text{, for all } k,$$

which shows that the expectation of row *i*, column *j* of the seventh term equals zero. Since this is true for every (i,j), the expectation of the seventh term results in a matrix of zeros. By a similar argument, the expectations of the fifth, sixth, and eighth terms of (3.43) produce matrices of zeros, thus completing the proof.

With Lemma 3.5, (3.43) is equivalent to

$$\mathbf{C}(n+1) = \mathbf{C}(n) - \mu\left(\mathbf{AC}(n) + \mathbf{C}(n)\mathbf{A}\right) + \mu^2\mathbf{D}(n) + \mu^2\varepsilon^*\mathbf{H} \tag{3.47}$$

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

Since (3.47) is a discrete, linear, time-invariant difference equation, its convergence is guaranteed if its homogeneous part is asymptotically stable (this implies BIBO stability) and if its forcing term $\mu^2\varepsilon^*\mathbf{H}$ is bounded. These are shown below.

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

$$D_{ij}(n) = 2G_{ij}C_{ij},\ i \neq j \tag{3.48}$$

$$D_{ii}(n) = \sum_{J=1}^{dQ+1} G_{iJ} C_{JJ} \tag{3.49}$$

where **G** is defined as

$$G_{ij} \equiv E\left[\frac{\left(\psi_i(n)\right)^2 \left(\psi_j(n)\right)^2}{\left(\boldsymbol{\psi}(n)^T \boldsymbol{\psi}(n)\right)^2} \right]. \tag{3.50}$$

By direct substitution,

$$D_{ij}(n) = \sum_{K=1}^{dQ+1} \sum_{J=1}^{dQ+1} E\left[\frac{\psi_i(n)\psi_j(n)\psi_J(n)\psi_K(n)}{\left(\boldsymbol{\psi}(n)^T \boldsymbol{\psi}(n)\right)^2} \right] C_{JK}. \tag{3.51}$$

Since the elements of $\boldsymbol{\psi}(n)$ are independent and all but one are zero-mean, the numerator of the expectation in (3.51) equals zero in many cases. To show which terms of the double summation of (3.51) equal zero, note that for any (i, j, J, K) the numerator can always be expressed as

$$\begin{gathered} E\left[\psi_i(n)\psi_j(n)\psi_J(n)\psi_K(n)\right] = \\ E\left[\left(\psi_{\tau_1}(n)\right)^{\rho_1}\right] E\left[\left(\psi_{\tau_2}(n)\right)^{\rho_2}\right] E\left[\left(\psi_{\tau_3}(n)\right)^{\rho_3}\right] E\left[\left(\psi_{\tau_4}(n)\right)^{\rho_4}\right], \end{gathered} \tag{3.52}$$

such that

$$\rho_1, \rho_2, \rho_3, \rho_4 \in \{0,1,2,3,4\}, \ \rho_1 + \rho_2 + \rho_3 + \rho_4 = 4, \ \rho_1 \ge \rho_2 \ge \rho_3 \ge \rho_4,$$

$$\tau_1, \tau_2, \tau_3, \tau_4 \in \{1, ..., dQ+1\}, \text{ and } \tau_\beta \ne \tau_\delta \text{ for } \beta \ne \delta.$$

Then each (i, j, J, K) maps to exactly one of the following five cases:

Case 1: $\rho_1 = 1, \rho_2 = 1, \rho_3 = 1, \rho_4 = 1$

Case 2: $\rho_1 = 2, \rho_2 = 1, \rho_3 = 1, \rho_4 = 0$

Case 3: $\rho_1 = 2, \rho_2 = 2, \rho_3 = 0, \rho_4 = 0$

Case 4: $\rho_1 = 3, \rho_2 = 1, \rho_3 = 0, \rho_4 = 0$

Case 5: $\rho_1 = 4, \rho_2 = 0, \rho_3 = 0, \rho_4 = 0$

Remembering that the element not assured to be zero-mean is denoted as $\psi_\zeta(n)$, (see (3.45)), then for Case 1, at most one member of the set $\{\tau_1, \tau_2, \tau_3, \tau_4\}$ equals ζ, thus (3.52) equals zero. Similarly for Case 2, at most one member of the set $\{\tau_1, \tau_2, \tau_3\}$ equals ζ, thus (3.52) equals zero.

For Case 4, if $\tau_1 = \zeta$, then from (3.45), (3.52) equals zero. If $\tau_2 = \zeta$, then by Lemma A.4, (3.52) equals zero. If $\tau_1, \tau_2 \neq \zeta$, then by either (3.45) or Lemma A.4, (3.52) equals zero, thus (3.52) equals zero for the entirety of Case 4.

For Cases 3 and 5, (3.52) could be non-zero.

Using an argument similar to that found in the proof of Lemma 3.2, the instances of (i, j, J, K) that fall into Cases 1, 2, and 4 can be shown to have a zero contribution to the double summation of (3.51). For the remaining cases, consider first the instances where $i \neq j$, which eliminates Case 5 in addition to Cases 1, 2, and 4, thus (3.51) equals (3.48). For $i = j$, after eliminating Cases 1, 2 and 4, Cases 3 and 5 make (3.51) equal to (3.49).

Having shown (3.47), (3.48), and (3.49), the techniques used for remainder of the mean-square proof are nearly identical to those presented in [62], and thus will only be outlined here (see the Appendix and [33] for details). Off-diagonal elements of $\mathbf{C}(n)$ are treated separately from the diagonal elements. The off-diagonal term of (3.47) is

$$C_{ij}(n+1) = \gamma_{ij} C_{ij}(n), i \neq j \tag{3.53}$$

$$\gamma_{ij} = 1 - \mu\left(A_{ii} + A_{jj}\right) + 2\mu^2 G_{ij}, i \neq j, \tag{3.54}$$

with $\left|\gamma_{ij}\right| < 1$ (Lemma A.5), and thus (3.53) 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 $\mathbf{C}(n)$, $\mathbf{\Omega}(n) \equiv \left[C_{11}(n), C_{22}(n), \ldots, C_{(dQ+1)(dQ+1)}(n)\right]^T$. From (3.47),

$$\mathbf{\Omega}(n+1) = \mathbf{F}\,\mathbf{\Omega}(n) + \mu^2 \varepsilon^* \mathbf{h}, \tag{3.55}$$

$$\mathbf{F} = \text{diag}\left\{\left(1 - 2\mu \mathbf{A}_{ii}\right)\right\} + \mu^2 \mathbf{G}, \ \mathbf{h} \equiv \left[\mathbf{H}_{11}, \mathbf{H}_{22}, \ldots, \mathbf{H}_{(dQ+1)}\right]^T. \tag{3.56}$$

It can be shown ([62], [33], Lemma A.8) that (3.55) is BIBO stable. Assuming $0 < \mu < 2$, the forcing term of (3.55) 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 $\left|H_{ii}\right| \leq 1/\left(dQ-2\right)\lambda_{\min}$ (Lemma A.6). Equations (3.53) and (3.55) 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,

$$\lim_{n \to \infty} \mathbf{\Omega}(n) = \mu^2 \varepsilon^* \left(\mathbf{I} - \mathbf{F}\right)^{-1} \mathbf{h}.$$

This is formalized in Theorem 3.2, the main result of Section 3.2.2.5:

Theorem 3.2 Given Assumption 1 through Assumption 4 and $0 < \mu < 2$,

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

Proof: Linear time-invariant system (3.55) is BIBO stable ([62], [33], Lemma A.8). Its input signal is bounded, thus $\lim_{n \to \infty} \mathbf{\Omega}(n) = \left(\mathbf{I} - \mathbf{F}\right)^{-1} \mu^2 \varepsilon^* \mathbf{h}$. Then with (3.53) and $\left|\gamma_{ij}\right| < 1$, $\lim_{n \to \infty} \mathbf{C}(n) = diag\left\{\left(\mathbf{I} - \mathbf{F}\right)^{-1} \mu^2 \varepsilon^* \mathbf{h}\right\}$. The definitions of $\mathbf{C}(n)$ and $\mathbf{L}(n)$ give the final result, concluding the proof.

The key contribution of Section 3.2.2.5 comes from showing (3.47) through (3.49) without requiring $E\left[\mathbf{y}(n)\right]=\mathbf{0}$. Expressions (3.47) through (3.49) exist in [62], but required $E\left[\mathbf{y}(n)\right]=\mathbf{0}$. By demonstrating (3.47) through (3.49) without requiring $E\left[\mathbf{y}(n)\right]=\mathbf{0}$, the results of [62] are significantly extended.

3.2.2.6 Global Stability

Global Stability has been built into the control structure. Both plant (3.24) and controller (3.31) are FIR filters. The controller simply conditions the set point $\left\{y^*\right\}$; 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 (3.30). The FIR structure of the controller then guarantees BIBO stability for the modified system.

3.2.2.7 Discussion

This concludes the first proof of Section 3.2. Theorem 3.1 and Theorem 3.2 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 3.2.3, no assumption about the FIR invertibility of the plant is made in Section 3.2.2.

3.2.3 A Proof of Controller Convergence and Global Stability– The Accurate Plant Inversion Case

Section 3.2.3 contains the second of the proofs of Section 3.2. The plant and controller used in Section 3.2.3 are identical to that described in Section 3.2.1 except the NLMS update equation (3.30) is replaced by

$$\hat{\mathbf{Q}}(n+1)=\hat{\mathbf{Q}}(n)+\frac{\mu\mathbf{y}(n)}{\varsigma+\mathbf{y}(n)^T\mathbf{y}(n)}e(n),\ \varsigma>0. \tag{3.57}$$

3.2.3.1 All-Pole Plant Approximation and other Assumptions

As discussed in Section 3.1.2.2, the possibility of a Non-Minimum Phase (NMP) $B\left(z^{-1}\right)$ encourages the use of an all-pole plant approximation (with a corresponding all-zero controller). Assume that $B\left(z^{-1}\right)$ has no zeros on the unit circle and that $B\left(z^{-1}\right)=B^{+}\left(z^{-1}\right)B^{-}\left(z^{-1}\right)$, with $B^{+}\left(z^{-1}\right)$ having zeros exclusively inside the unit circle and $B^{-}\left(z^{-1}\right)$ having zeros exclusively outside the unit circle. By long division, $N^{+}\left(z^{-1}\right)=\left(B^{+}\left(z^{-1}\right)\right)^{-1}=n_0^{+}+n_1^{+}z^{-1}+\ldots$, with $\left|n_i^{+}\right|\geq\xi_1\left|n_{i+1}^{+}\right|$, $i\geq 0$, $0\leq\xi_1<1$ (a causal filter) and $N^{-}\left(z^{-1}\right)=\left(B^{-}\left(z^{-1}\right)\right)^{-1}=n_\gamma^{-}+n_{\gamma+1}^{-}z^{\gamma+1}+\ldots$, with $\left|n_i^{-}\right|\geq\xi_2\left|n_{i+1}^{-}\right|$, $i\geq\gamma$, $\gamma>0$, $0\leq\xi_2<1$ (a non-causal filter). Since the coefficients are decreasing exponentially, with enough taps, N^{+} and N^{-} can be approximated by truncated FIR filters, $\bar{N}^{+}\left(z^{-1}\right)=n_0^{+}+n_1^{+}z^{-1}+\ldots+n_{dN^{+}}^{+}z^{-dN^{+}}$ and $\bar{N}^{-}\left(z^{-1}\right)=n_\gamma^{-}+n_{\gamma+1}^{-}z^{\gamma+1}+\ldots+n_V^{-}z^{V}$. Therefore, an FIR

$$Q\left(z^{-1}\right)\equiv z^{-V}\bar{N}^{+}\left(z^{-1}\right)\bar{N}^{-}\left(z^{-1}\right)+C' \tag{3.58}$$

can well approximate $z^{-V}\left(B\left(z^{-1}\right)+C\right)^{-1}$. This approximation is formalized by the following Assumption, used throughout Section 3.2.3.

Assumption 5 $B\left(z^{-1}\right)$ has no zero on $|z|=1$ and the plant (3.25) is equivalently expressed as

$$Q\left(z^{-1}\right)y(n)=u(n-d-V). \tag{3.59}$$

In addition to the all-pole assumption given by Assumption 5, here are the other assumptions made throughout Section 3.2.3:

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

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

Assumption 6 requires at least one tap of $\hat{\mathbf{Q}}(n)$ to be non-zero.

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.

3.2.3.2 Convergence and Global Stability

In Section 3.2.2, the all-pole plant approximation discussed in Section 3.2.3.1 is considered inexact. As a result, considerable effort and some additional assumptions are required. These assumptions included a Gaussian excitation signal that is sufficiently exciting [63] and an assumption of independence between the current tap estimates $\hat{\mathbf{Q}}(n)$ and previous values of $\mathbf{y}(n)$ and $u(n-d-V)$. The main result of Section 3.2.2 is that, under these assumptions, the NLMS adaptation process produces a controller that converges in the mean and mean square to the minimum mean square error solution, $\mathbf{Q}_0$. Global stability claims are made, although they are looser than those made below.

In this section, some of the more limiting assumptions of Section 3.2.2 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.

3.2.3.2.1 Proof of Convergence and Global Stability

The update equation (3.59) is identical to (3.3.19) of [63]. From (3.29), (3.59) and (3.27), $e(n) = -\tilde{\mathbf{Q}}(n)^T \mathbf{y}(n)$, and from Lemma 3.3.2 of [63],

$$\lim_{n\to\infty} \frac{e(n)}{\left(\varsigma + \mathbf{y}(n)^T \mathbf{y}(n)\right)^{1/2}} = 0, \tag{3.60}$$

$$\lim_{n\to\infty} \left\| \hat{\mathbf{Q}}(n-k) - \hat{\mathbf{Q}}(n) \right\| = 0 \text{ for any finite } k. \tag{3.61}$$

From (3.29), (3.31), and (3.27),

$$\begin{aligned} e(n) &= \hat{\mathbf{Q}}(n-d-V)^T \mathbf{y}^*(n \mid n-d-V) - \hat{\mathbf{Q}}(n)^T \mathbf{y}(n) \\ &= \left(\hat{\mathbf{Q}}(n-d-V) - \hat{\mathbf{Q}}(n)\right)^T \mathbf{y}^*(n \mid n-d-V) \\ &\quad + \hat{\mathbf{Q}}(n)^T \left(\mathbf{y}^*(n \mid n-d-V) - \mathbf{y}(n)\right) \end{aligned} \tag{3.62}$$

Then from (3.60) and (3.61)

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

$$\lim_{n\to\infty} \frac{\left(\hat{\mathbf{Q}}(n)^T \boldsymbol{\chi}(n)\right)^2}{\varsigma + \mathbf{y}(n)^T \mathbf{y}(n)} = 0 \tag{3.63}$$

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

To show that (3.63) implies that $\lim_{n\to\infty}\left(\hat{\mathbf{Q}}(n)^T \boldsymbol{\chi}(n)\right)^2 = 0$, note that

$$\left\|\mathbf{y}(n)\right\| \le \kappa_1 + \kappa_2 \max_{0\le\tau\le n}\left\|\mathbf{y}(\tau)\right\|,\ 0<\kappa_1,\kappa_2<\infty$$

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,

$$\left\|\mathbf{y}(n)\right\| \le \kappa_3 + \kappa_4 \max_{0\le\tau\le n}\left|\hat{\mathbf{Q}}(\tau)^T \boldsymbol{\chi}(\tau)\right|,\ 0<\kappa_3,\kappa_4<\infty . \tag{3.64}$$

With (3.63) and (3.64), the Key Technical Lemma [63] asserts that

$$\left\|\mathbf{y}(n)\right\| \text{ is bounded, and} \tag{3.65}$$

$$\lim_{n\to\infty}\left(\hat{\mathbf{Q}}(n)^T \boldsymbol{\chi}(n)\right)^2 = 0 . \tag{3.66}$$

Note that (3.65) and (3.66) do 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 $\hat{\mathbf{Q}}(n)$; see (3.61). Assumption 7 prevents $\hat{\mathbf{Q}}(n)$ from nulling frequencies present in $\boldsymbol{\chi}(n)$.

The main result of Section 3.2.3 is as follows:

Theorem 3.3 Given Assumption 5 through Assumption 7, the plant (3.24), which is equivalent to (3.59), controlled by (3.31) through (3.29) and (3.57), gives

$$\lim_{n\to\infty}\chi(n) = 0 .$$

Proof: Equations (3.60) and (3.61) give (3.63). The Key Technical Lemma gives (3.66), which with Assumption 7, gives the result. This completes the proof.

3.2.3.2.2 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 3.2.2.1'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 3.1 and Theorem 3.2 as well as simulation experiments suggest that the control structure behaves stably. Thus the results of this section and the convergence results of Section 3.2.2 should be viewed as complimentary. Both examine the control system of Section 3.2.1, each start with different assumptions, and both produce desirable results.

3.2.4 Summary

In Section 3.2, the convergence and stability properties of the controller proposed in Section 3.1.4 are examined extensively. Section 3.2 contains two separate yet complimentary proofs. Theorem 3.1 and Theorem 3.2 summarizes the first proof, in Section 3.2.2. The second proof, in Section 3.2.3, is summarized by Theorem 3.3. Each of these proofs demonstrates desirable qualities of the controller presented in Section 3.1.4. 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 3.2 make a convincing case that Adaptive Approximate Inverse Control has attractive convergence and stability properties.

3.3 Chapter Summary

This chapter takes up the challenge of finding an effective control strategy for the explicit rate congestion controller. Section 3.1 recognizes that the plant developed in Chapter 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 what appears to be attractive convergence properties. However, formal convergence analysis is postponed to Section 3.2.

Section 3.2 contains two complimentary proofs of convergence for the control structure selected in Section 3.1. Each of these proofs begin with different assumptions; both suggest attractive analytical convergence properties.

Further comments are made in Sections 3.1.5 and 3.2.4. Most of the material of this chapter has been published as [23] and [31].

CHAPTER 4 ALGORITHM ENHANCEMENTS

In this chapter, three additions to the congestion control mechanism are introduced and discussed. 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.

Before these modifications are introduced, a simulation framework is presented, wherein the parameters of the plant are assigned values consistent with actual ATM networks. This framework is described in Section 4.2.

The first algorithm enhancement addresses the convergence rate of the controller. The results of Section 3.2 ensure that the originally proposed congestion controller eventually converges. However, without the modifications presented in Section 4.3, convergence rates are unnecessarily, and possibly unacceptably, slow. Significant speedup is obtained with the modifications of Section 4.3.

The second algorithm enhancement, described in Section 4.4, 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 here and elsewhere 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 enhancement offered here controls queue size in an elegantly simple way.

The third algorithm enhancement, described in Section 4.5, 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 as a noise source in the plant model. This noise causes biasing in the parameter estimates used for the controller. A novel method to minimize the bias is introduced. 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.

The remainder of this chapter begins with a brief review of the original congestion control structure. Section 4.2 discusses how certain plant and controller parameters are chosen to emulate a realistic congestion control scenario, thereby establishing a framework for the subsequent simulations. Convergence rates are addressed in Section 4.3. Adding queue sizes to the model and controller occurs in Section 4.4. Section 4.5 augments the original plant to include non-responsive ABR sources with varying rates and then attacks the resulting bias issues. Brief concluding remarks are made in Section 4.6.

4.1 System Definition (Review)

The system under consideration is

$$y(n) = \mathbf{B}^T \mathbf{u}(n-d) + C \tag{4.1}$$

$$u(n) = \hat{\mathbf{Q}}(n)^T \mathbf{y}^*(n+d+V \mid n) \tag{4.2}$$

$$\hat{\mathbf{Q}}(n) = \left[\hat{q}_0(n), \hat{q}_1(n), \ldots, \hat{q}_{dQ}(n), \hat{q}_{DC}(n)\right]^T$$

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

$$\hat{u}(n-d-V) = \hat{\mathbf{Q}}(n)^T \mathbf{y}(n) \tag{4.3}$$

$$\mathbf{y}(n) = \left[y(n), y(n-1), \ldots, y(n-dQ), y_{DC}\right]^T \tag{4.4}$$

$$e(n) \equiv e_u(n-d-V) \equiv u(n-d-V) - \hat{u}(n-d-V) \tag{4.5}$$

$$\hat{\mathbf{Q}}(n+1) = \hat{\mathbf{Q}}(n) + \frac{\mu \mathbf{y}(n)}{\mathbf{y}(n)^T \mathbf{y}(n)} e(n),\ 0 < \mu < 2 \tag{4.6}$$

This system is introduced and discussed in Sections 2.4 and 3.1.4.

4.2 Simulation Framework

To demonstrate the various design issues covered in this chapter, a common simulation framework, using the Matlab [67] simulation tool, is now defined.

The plant as defined in Section 2.4 and reviewed in Section 4.1 envisions a switch *SW* having an output port *j* containing a congestion controller. The output port has a buffer and output link that carries traffic of various service categories. The amount of bandwidth assigned to the explicit rate controlled Available Bit Rate (ABR) traffic is designated $y^*(n)$ cells per second. Outgoing ABR traffic arrives to port *j* from the various input ports of *SW* at rate $y(n)$. The congestion controller must determine an explicit rate $u(n)$. Resource management (RM) cells deliver these explicit rates to the ABR sources. Each responsive ABR source changes its sending rate to the explicit rate communicated by the most recently arrived RM cell. Port *j* sees each source responding to its explicit rates with a potentially different (but non-changing) delay. This plant is shown in Figure 2.1, repeated here.

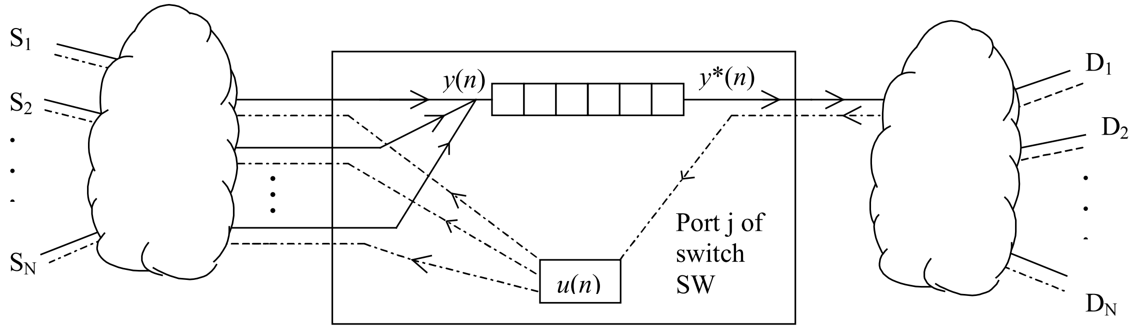

Figure 4.1 Plant from perspective of Switch Output Port

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. Of that, 1 Mcps (on average) is allocated to ABR traffic. 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 $E\left[y^*\right]$=1 Mcps and a standard deviation σ_{y^*} of 22 kcps[15].

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.

[15] 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.

The minimum response delay d =10 msec. The distribution of the delays of the 22 sources is given by $B\left(z^{-1}\right)=z^{-10}\left(2+9z^{-1}+8z^{-2}+3z^{-3}\right)$. This corresponds to a plant with one non-minimum phase zero and a pair of complex minimum phase zeros (See Figure 3.3). 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 inspection reveals that this rarely occurs after an initial transient.

4.3 Convergence Rate Improvements

The results of Chapter 3 assure convergence as time goes to infinity, but say little about the rate of convergence.

4.3.1 Unmodified Convergence

Figure 4.2 shows the results of simulating the system without any modifications to improve the rate of convergence. After 8 seconds, the convergence of the controller is so poor that it appears to be admitting over twice the desired rate of traffic[16]. This is clearly unacceptable performance.

[16] Note that the results from Chapter 3 ensures that $y(n)$ will eventually coincide with $y^*(n)$.

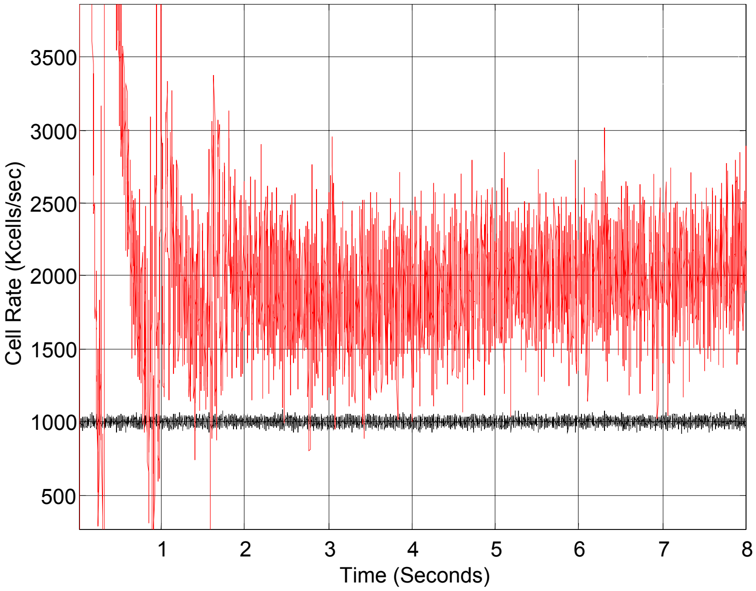

Figure 4.2 Comparing the Set Point (lower curve centered at 1000) and Port Input Rates (higher curve approximately centered at 2000)–Unmodified Case. The lower set point y^* plot remains around 1000 kcps while the port input rate y plot has a mean value around 2000 kcps.

4.3.2 Managing the Eigenvalue Spread

The Least Mean Square (LMS) algorithm has the property that the mean of the coefficient error vector, $E[\tilde{\mathbf{Q}}(n)]$, converges to zero at a rate inversely proportional to the eigenvalue spread $\lambda_{\max} / \lambda_{\min}$ of $\mathbf{R} = E\left[\mathbf{y}(n)\mathbf{y}(n)^T\right]$ [64]. Note that the eigenvalue spread is a measure of the conditionality of a matrix. Often the term *condition number* is used to quantitatively describe the ill-condition of a matrix; in fact, the eigenvalue spread is equivalent to one definition of the condition number [64][17]. It is more difficult to

[17] This is true if the condition number of a matrix $\mathbf{A}$ is $\|\mathbf{A}\|\|\mathbf{A}^{-1}\|$ and the norm is $\|\mathbf{A}\| \equiv \sqrt{\text{largest eigenvalue of } \mathbf{A}^H\mathbf{A}}$. Reference [64] describes this relationship in its concise review of the properties of eigenvalues and their relationship to adaptive filtering.

specify the convergence trajectory of $E[\tilde{\mathbf{Q}}(n)]$ for Normalized Least Mean Square (NLMS) adaptation in all but the simplest cases [62], although practical experience shows that speed of convergence is still a strong function of eigenvalue spread.

Generally adaptive filters will converge relatively quickly in the frequency bands associated with the strongest parts of excitation signal. In many situations, the excitation signal used to identify the filter is also the input used to produce the desired adaptive filter output. However, this is not the case in the current control strategy. Therefore, the congestion control system under study is unlikely to perform well until $\hat{\mathbf{Q}}(n)$ well approximates $\mathbf{Q}_0$ at all frequencies.

4.3.3 Reducing Means Via Constant Estimates

In the course of the study leading to this dissertation, several strategies to improve convergence time were proposed and evaluated. The most promising strategy is as follows: provide the identification algorithm with zero-mean signals by estimating and removing the signal means. Then, perform DC correction in the controller by an additive term.

The basic concept is illustrated by Figure 4.3. Let α and β be fixed estimates of $E\left[u(n)\right]$ and $E\left[y(n)\right]$ respectively. Subtract α and β from their corresponding signals to perform identification. Constants α and β are then added to the controller to perform DC correction.

It is now shown that for the architecture of Figure 4.3, once $\hat{\mathbf{Q}}(n)$ converges to its optimal $\mathbf{Q}_0$, $E\left[y(n)\right]=E\left[y^*(n\,|\,n-V-d)\right]$. As shown in Section 3.2.2.2, the DC tap allows $\hat{\mathbf{Q}}_{lin}(n)$ to converge as if the signals $u_1(n)$ and $y_1(n)$ are zero-mean. In other words, inclusion of the DC tap allows $\hat{\mathbf{Q}}_{lin}(n)$ to converge to the same $\mathbf{Q}_{0,lin}$, regardless of the chosen values of α and β. Further, inclusion of a DC tap ensures

$$E\left[u_1(n)\right]=E\left[\hat{u}_1(n)\right] \tag{4.7}$$

once $\hat{\mathbf{Q}}(n)$ fully converges to $\mathbf{Q}_0$. Therefore, when $\hat{\mathbf{Q}}=\mathbf{Q}_0$, from (4.7),

$$E\left[u(n)-\alpha\right]=E\left[y(n)-\beta\right]\sum_{i\in lin} q_{0,i}+q_{0,DC}\,. \tag{4.8}$$

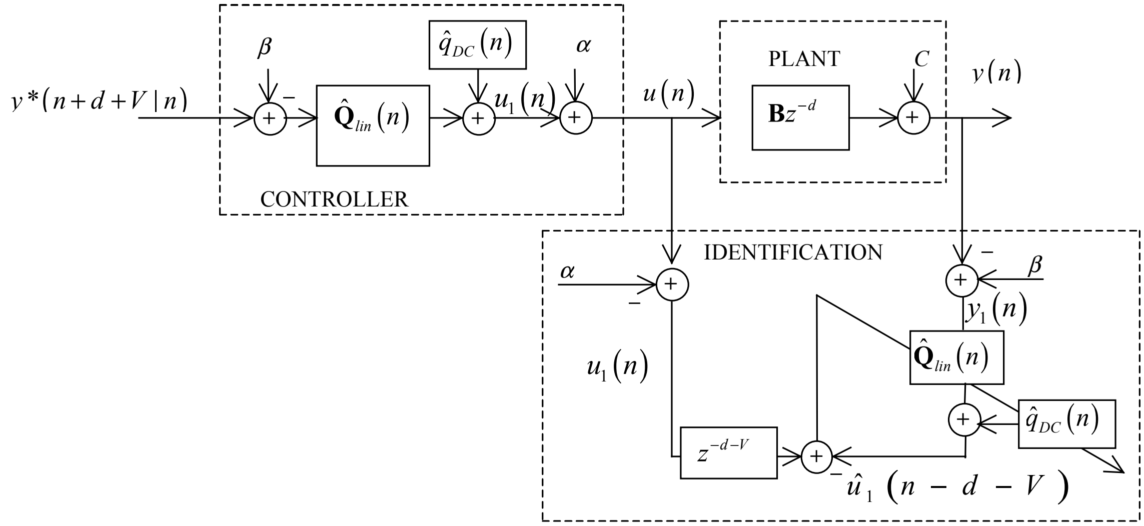

Figure 4.3 Architecture for Adding and Subtracting Fixed Estimates of the Means

Also, if $\hat{\mathbf{Q}}(n)$ is set equal to its converged value $\mathbf{Q}_0$, the relationship between the controller's mean input and mean output is

$$E\left[u(n)\right]=E\left[y^*(n+d+V\,|\,n)-\beta\right]\sum_{i\in lin} q_{0,i}+q_{0,DC}+\alpha\,. \tag{4.9}$$

Eliminating $E[u(n)]$ from (4.8) and (4.9) reveals the desired result: when $\hat{\mathbf{Q}}$ in Figure 4.3 is replaced with the filter to which it is converging, $\mathbf{Q}_0$, then

$$E\left[y(n)\right]=E\left[y^*(n\,|\,n-V-d)\right]. \tag{4.10}$$

Methods for finding α and β have not been completely explored. One possibility is to set β equal to the sample mean of $y^*(n+d+V\,|\,n)$. Then, if N is the total number of ABR flows supported by the port (including bottle-necked flows), set $\alpha=\beta/N$. Other methods are also possible. However, simply using sample means of y and u leads to instability, as will be shown in Section 4.3.4.

Simulations show the method depicted in Figure 4.3 has the potential to make a significant improvement in convergence rate, but that performance is quite sensitive to the accuracy of the mean estimates. For example, Figure 4.4 shows the case when $a=0.99E[u(n)]$ and $\beta=1.01E[y(n)]$. The measured eigenvalue spread of $\mathbf{R}$ is 50. The convergence is very fast.

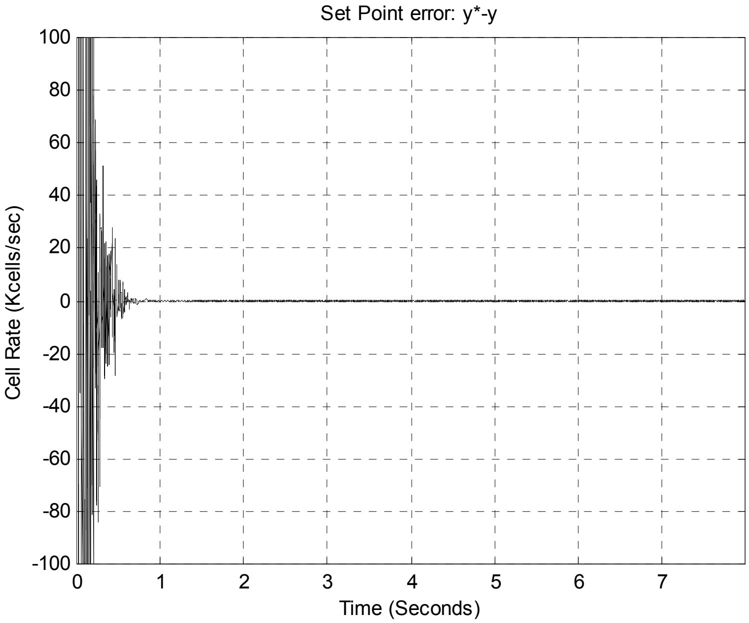

Figure 4.4 Set-Point Error When Estimates a and β are Within 1% of Their Correct Values

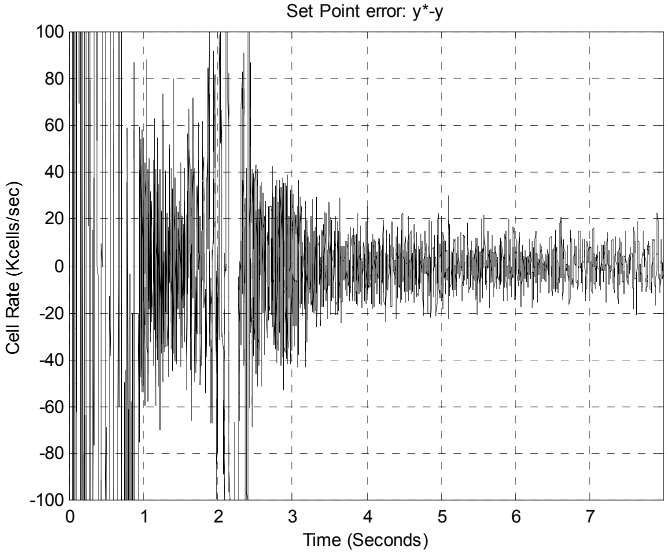

Figure 4.5 Set-Point error When Estimates a and β are Within 10% of Their Correct Values

However, when $a = 0.9E[u(n)]$ and $\beta = 1.1E[y(n)]$, as shown in Figure 4.5, the performance is noticeably slower, albeit much better than shown in Figure 4.2 (where essentially $a = 0$ and $\beta = 0$). The measured eigenvalue spread is 1.7×10^5.

In summary, if an off-line method can be found to estimate $E[u(n)]$ and $E[y(n)]$ accurately, this method holds promise, but its effectiveness decreases rapidly as the estimates a and β become less accurate.

4.3.4 Reducing Means Via Constantly Updating Estimates

One obvious method for estimating $E[u(n)]$ and $E[y(n)]$ is by directly calculating sample means. The most common method is using a single-pole filter. If the sample means of $u(n)$ and $y(n)$ are notated $u_{SM}(n)$ and $y_{SM}(n)$ respectively, then

$$u_{SM}(n) = u_{SM}(n-1)(1-\delta) + u(n) \tag{4.11}$$

$$y_{SM}(n) = y_{SM}(n-1)\,(1-\delta) + y(n) \tag{4.12}$$

where $0 < \delta < 1$.

The sample means $u_{SM}(n)$ and $y_{SM}(n)$ then replace α and β in Figure 4.3, as shown in Figure 4.6. Note that no DC tap is employed in this architecture.

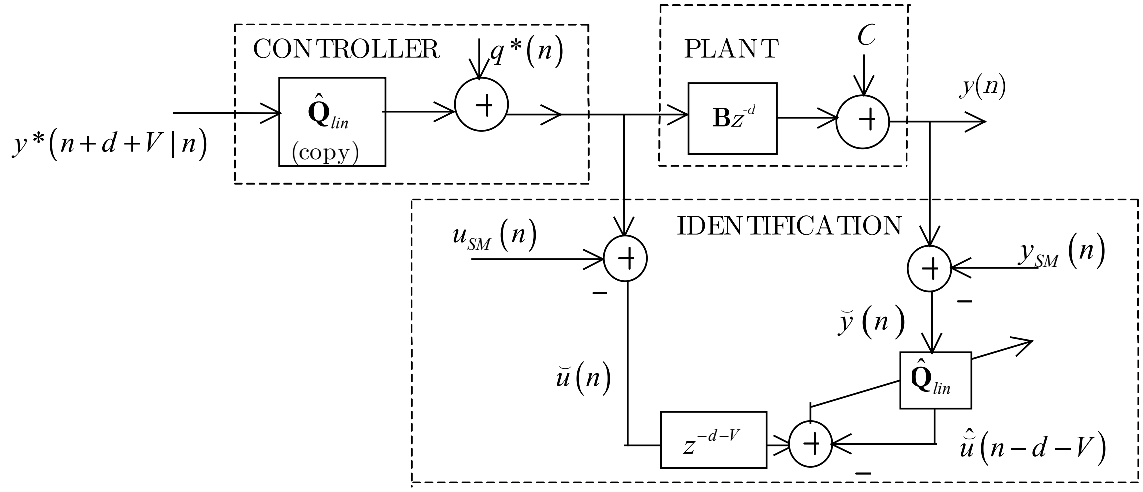

Figure 4.6 Architecture for Subtracting Sample Mean Estimates

From Figure 4.6,

$$y(n) = \sum_k u(n-k)\,b_k + C \tag{4.13}$$

$$u(n) = \sum_i y^*(n+d+V-i\,|\,n-i)\,\hat{q}_{lin,i}(n) \tag{4.14}$$

$$y(n) = \sum_k \sum_i y^*(n+d+V-k-i\,|\,n-k-i)\hat{q}_{lin,i}b_k + q^*(n)\sum_k b_k + C \tag{4.15}$$

The optimal DC correcting value for $q^*(n)$ is

$$q^*(n) = E\left[u(n)\right] - E\left[y(n)\right]\sum_{i=0}^{dQ}\hat{q}_i(n), \tag{4.16}$$

which is now shown. Lemma 4.1 relies on Assumption 6, introduced on page 83, repeated here for convenience:

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

Lemma 4.1 For the architecture shown in Figure 4.6, given Assumption 6, the necessary and sufficient condition that $E\left[y^*(n|n-V-d)\right] = E\left[y(n)\right]$ is that $q^*(n)$ is given by (4.16).

Proof: (Sufficiency) Note that

$$\sum_{j=0}^{dQ} \hat{q}_j(n) = \sum_{j=0}^{dQ} \hat{q}_{lin,j}$$

is the DC gain of FIR filter $\hat{\mathbf{Q}}_{lin}$, since there is no DC tap. Taking expectations of both sides of (4.15), and then substitute for $E\left[u(n)\right]$ from (4.13) gives

$$E\left[y(n)\right] = E\left[y^*(n+d+V|n)\right]\sum_k\sum_i \hat{q}_{lin,i}b_k + E\left[y(n)\right] - C - E\left[y(n)\right]\sum_k\sum_j \hat{q}_{lin,j}b_k + C$$

$$0 = \left(E\left[y^*(n+d+V|n)\right] - E\left[y(n)\right]\right)\sum_k\sum_j \hat{q}_{lin,j}b_k.$$

Thus from Assumption 6,

$$E\left[y^*(n+d+V|n)\right] = E\left[y(n)\right].$$

(Necessity) Given: $E\left[y^*(n+d+V|n)\right] = E\left[y(n)\right]$, substitute (4.14) into (4.13), take the expectation of both sides of the result, and also (4.13), then

$$E\left[u(n)\right]\sum_k b_k = E\left[y(n)\right]\sum_i\sum_k \hat{q}_i(n)b_k + q^*(n)\sum_k b_k$$

$$q^*(n) = E\left[u(n)\right] - E\left[y(n)\right]\sum_i \hat{q}_{lin,i},$$

thus completing the proof.

However, there is a problem. Signal $q^*(n)$ creates feedback paths not readily observable in Figure 4.6. Express (4.11) as a filter in the delay operator z^{-1},

$$u_{SM}(n) = \frac{z\delta}{z-(1-\delta)}u(n) \tag{4.17}$$

$$\breve{u}(n) = u(n) - u_{SM}(n) = \frac{(z-1)}{\frac{z}{1-\delta}-1}u(n), \tag{4.18}$$

with $y_{SM}(n)$ and $\breve{y}(n)$ similarly defined.

Redrawing Figure 4.6 using expressions (4.17) and (4.18) gives Figure 4.7, where the feedback path is plainly shown.

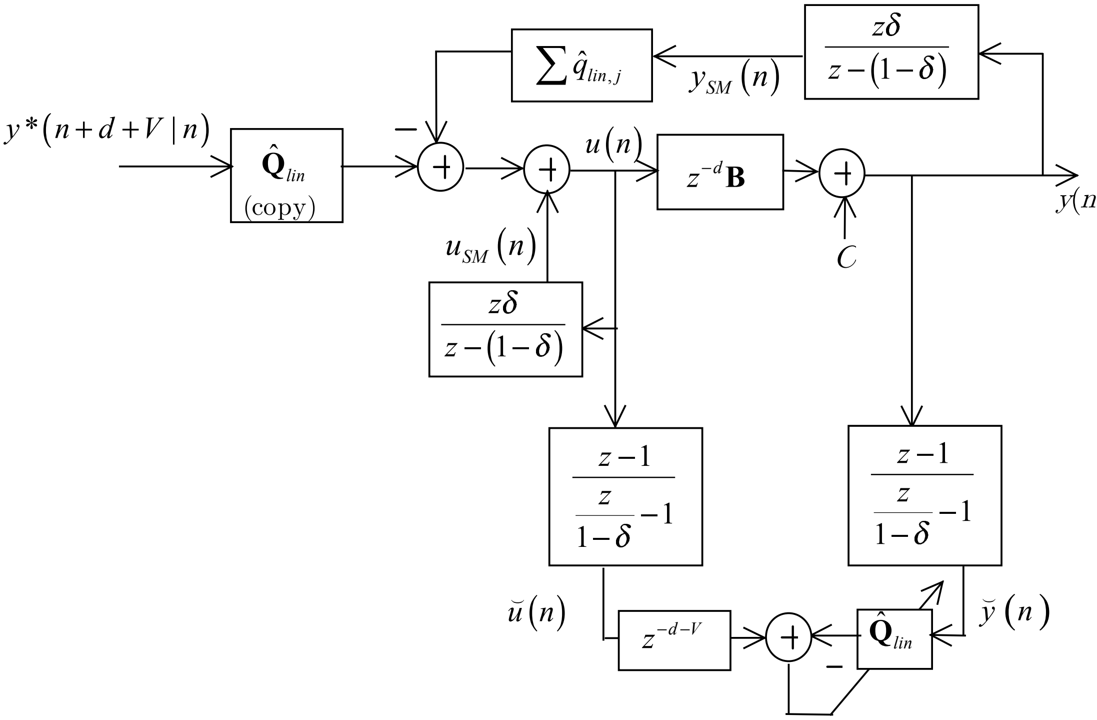

Figure 4.7 Architecture for Subtracting Sample Mean Estimates (Shown in Figure 4.6) with Feedback Explicitly Shown

Several simulations expose the unstable behavior suggested by Figure 4.7. When the closed-loop poles and zeros of this system are periodically plotted during system convergence, it is clear that unstable performance occurs when the closed-loop poles fall outside the unit-circle during the convergence interval. Unsurprisingly, stable performance is more likely as δ is decreased, e.g. below .001. This has the effect of nearly breaking the feedback path shown in Figure 4.7. However, as δ is decreased, the sample-mean estimates $u_{SM}(n)$ and $y_{SM}(n)$ take much longer to converge to good estimates of $E[u(n)]$ and $E[y(n)]$. As a result $\breve{u}(n)$ and $\breve{y}(n)$ take longer to become approximately zero-mean signals, thus the eigenvalue spread of $E\left[\breve{\mathbf{y}}(n)\breve{\mathbf{y}}(n)^T\right]$ remains large and $\hat{\mathbf{Q}}_{lin}(n)$ converges very slowly.

4.3.5 Reducing Means Via Downsampled Estimates

To break the feedback path shown in Figure 4.7 and thus avoid instability, use significantly down-sampled versions of $u_{SM}(n)$ and $y_{SM}(n)$ for DC Correction. Specifically, run the identification process as shown in Figure 4.7, but update $q*(n)$ at a down-sampled rate.

$$q*(n) = u_{SM,q*}(n) - y_{SM,q*}(n)\sum_{i=0}^{dQ}\hat{q}_i\left(\left\lfloor\frac{n}{dsInterval}\right\rfloor dsInterval\right), \tag{4.19}$$

$$u_{SM,q*}(n) \equiv u_{SM}\left(\left\lfloor\frac{n}{dsInterval}\right\rfloor dsInterval\right),$$

$$y_{SM,q*}(n) \equiv y_{SM}\left(\left\lfloor\frac{n}{dsInterval}\right\rfloor dsInterval\right),$$

where $\lfloor x \rfloor$ is the integer part of x and *dsInterval* is an integer down-sample interval.

By infrequently latching the values of $u_{SM,q*}(n)$ and $y_{SM,q*}(n)$ used for determining $q*(n)$, the feedback paths of Figure 4.7 are essentially broken. For example, Figure 4.8 shows the case when $dsInterval = 500$, i.e. $q*(n)$ is updated once per 500 msec. The final measured eigenvalue spread is 6. The convergence rate is satisfactorily fast.

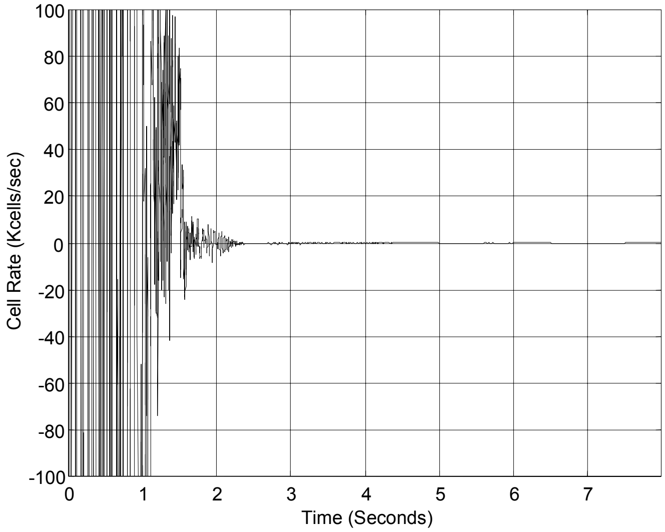

Figure 4.8 Set-Point Error When $Q*(n)$ is Updated Twice a Second.

4.3.6 Discussion

Convergence rate is a serious issue for the proposed explicit rate congestion controller. Without modifications, performance is unacceptable (Figure 4.2). If accurate estimates of $E[u(n)]$ and $E[y(n)]$ can be obtained a-priori, fixed estimates provide excellent performance (Figure 4.4), but if these fixed estimates are less accurate, performance degrades severely (Figure 4.5). An online sample mean calculation works quite well, as long as the feedback path of Figure 4.7 is broken by down-sampling the DC correction update (Figure 4.8).

4.4 Control of Queue Size

Most of the congestion control work by control theorists presented in Section 1.3.2 explicitly include queue matching in addition to rate matching in their cost functions, no doubt in part a response to [11]. In contrast, this dissertation, up to this point, has focussed on a pure rate-matching controller. This strategy, supported by [12], requires that the bandwidth available for ABR traffic be slightly under-utilized, thus creating extremely short (or zero) steady state queue lengths. While this has advantages, e.g. shorter end-to-end delay and smaller memory requirements, it may be more desirable to have, on average, longer buffers. Since ABR is not designed for delay-sensitive traffic, it may be preferable to add a small, known delay by targeting a non-zero buffer size in order to ensure network efficiency. The scheme presented thus far does not allow for a desired queue depth greater than zero.

Queue control is fairly easily incorporated into rate-matching schemes. The basic idea, suggested by [21], uses any preferred rate-matching scheme to determine an explicit rate. This explicit rate is then increased if the present queue depth is below its target, or decrease the explicit rate if the present queue depth is above the target.

The proposal of this section is distinct from [21] in that it scales the set point, $y^*\left(n+d+V \mid n\right)$, not the explicit rate $u\left(n\right)$ directly. Specifically, decide at time *n* the target input rate for time $n+d+V$ (see Section 2.1), but notate this as $\Theta\left(n+d+V \mid n\right)$

instead of $y^*(n+d+V\,|\,n)$. The target input rate $\Theta(n+d+V\,|\,n)$ is chosen without regard of the queue size. Further, for simplicity of presentation, assume that $\Theta(n+d+V\,|\,n)$ is the actual service capacity for ABR traffic at $n+d+V$.

Define a scalar $\eta(n)$ that is monotonically decreasing function of the queue size $queue\,(n)$. Control of this queue size is accomplished by multiplying $\Theta(n+d+V\,|\,n)$ by $\eta(n)$ to form $y^*(n+d+V\,|\,n)$, i.e.

$$y^*(n+d+V\,|\,n)=\eta(n)\Theta(n+d+V\,|\,n). \tag{4.20}$$

This queue-aware set-point $y^*(n+d+V\,|\,n)$ is used in exactly the same way as outlined in Sections 4.3.3 and 4.3.5. The plant model now includes the queue-depth $queue\,(n)$, which progresses as

$$queue(n+1)=queue(n)+y(n)-\Theta(n\,|\,n-d-V). \tag{4.21}$$

Taking the constant mean estimate method of Section 4.3.3 (shown by Figure 4.3) and incorporating (4.20) and (4.21) produces Figure 4.9.

To illustrate the queue control provided by (4.20), reconsider the example discussed in Section 4.3.3, where accurate constant estimates $a=0.99E[u(n)]$ and $\beta=1.01E[y(n)]$ are used to reduce the convergence rate. The set-point error is shown in Figure 4.4. In one example, with no attempt to control the size of the queue, i.e $\eta(n)=1$, the queue grows to just over 5000 cells, as shown in Figure 4.10.

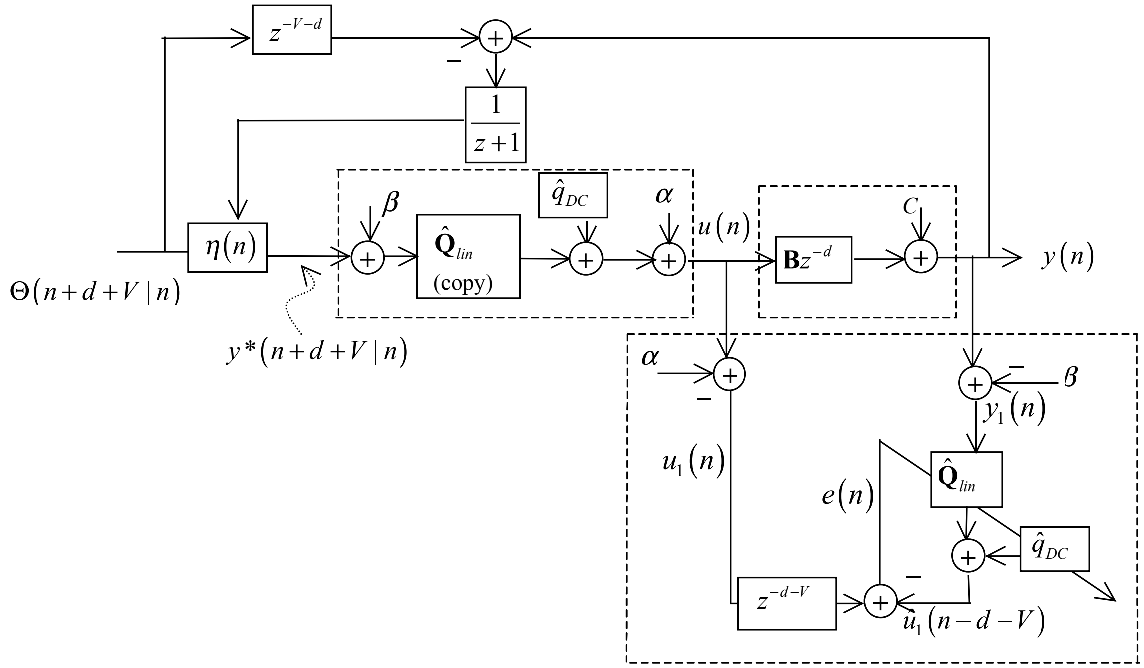

Figure 4.9 Queue Control Added to Controller of 4.3.3 (Compare to Figure 4.3)

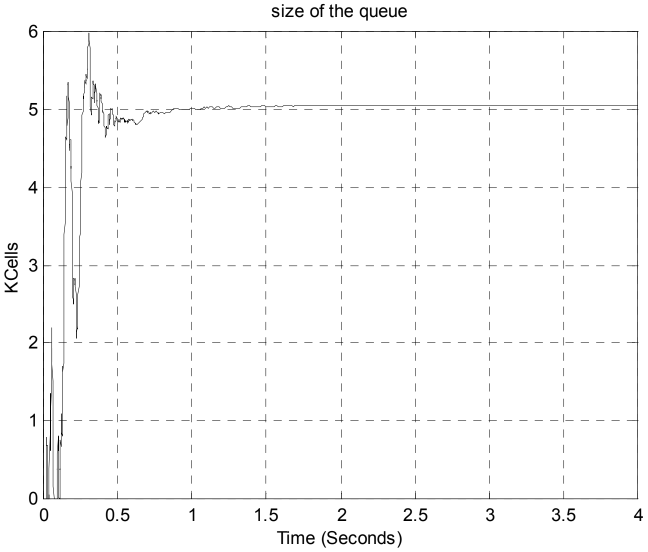

Figure 4.10 Queue Size for Example in Figure 4.4 Using No Queue-Depth Control

If instead, $\eta(n) = 0.99$ for all n, corresponding to a fixed policy of using only 99% of the ABR capacity, then the target queue depth is zero. Figure 4.11 shows the result. The rate at which the queue re-converges to zero increases if $\eta(n) < 0.99$. In itself, allowing 1% of the available explicit rate bandwidth to go unused in steady state may be acceptable. However, if the available explicit rate bandwidth were to increase or if the number of responsive flows were to decrease, the port would remain under utilized until the control system could respond or reconverge.

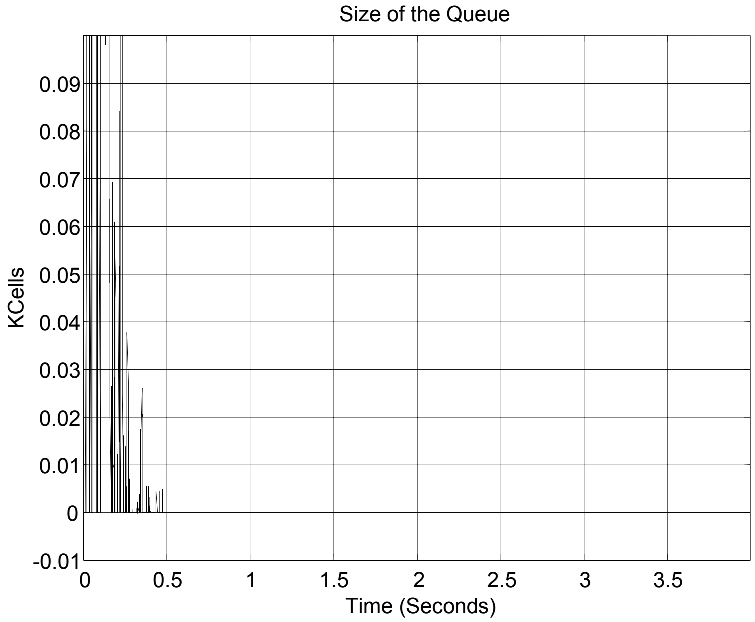

Figure 4.11 Queue Depth When Requested Bandwidth Remains 99% of Available Bandwidth, i.e. Target Queue Depth is Zero

To target a non-zero queue-depth, use a $\eta(n)$ function that decreases monotonically with *queue*(*n*). A simple function is shown in Figure 4.12.

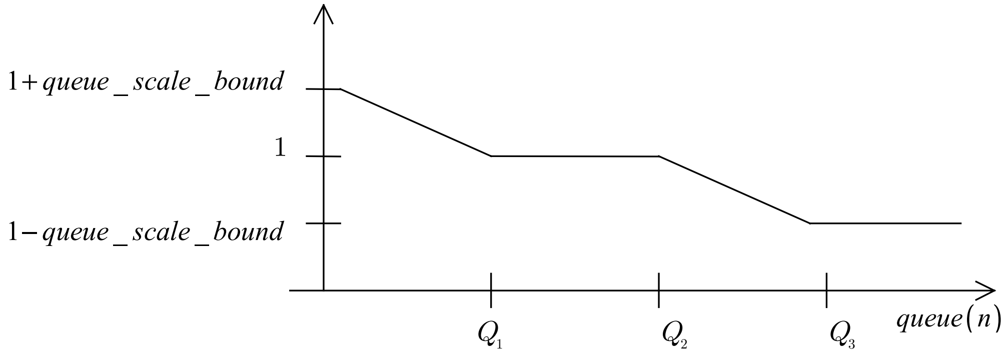

Figure 4.12 Sample $\eta(n)$ Function

Using the $\eta(n)$ function shown in Figure 4.12, with *queue_scale_bound* = 0.01, $Q_1 = 100$ cells, $Q_2 = 200$ cells, $Q_3 = 300$ cells, achieves the target queue-depth without perceptibly affecting the convergence rate, as shown in Figure 4.13 and Figure 4.14.

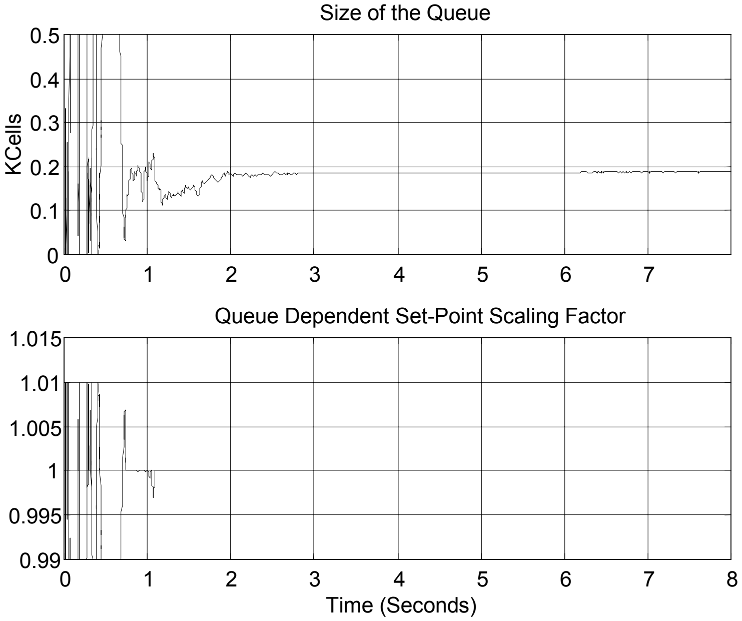

Figure 4.13 Queue depth and Set-point Scaling Factor $\eta(n)$ when Queue Target is 100-200 Cells

Comparing Figure 4.3 to Figure 4.9, clearly potentially destabilizing feedback is created by performing queue control with (4.20). Intuition suggests, and simulations confirm, that stability is only in jeopardy when the scaling of $\eta(n)$ is aggressive. Stability is maintained, using the $\eta(n)$ shown in Figure 4.12, with *queue_scale_bound* equal to 0.01. However, if *queue_scale_bound* changes from 0.01 to 0.1, the oscillations introduced significantly impact overall performance, as demonstrated by Figure 4.15 and Figure 4.16. It seems intuitive that using a small *queue_scale_bound* can make the impact of $\eta(n)$ on $y^*(n+d+V\,|\,n)$ nearly negligible, yet still effect the desired behavior.

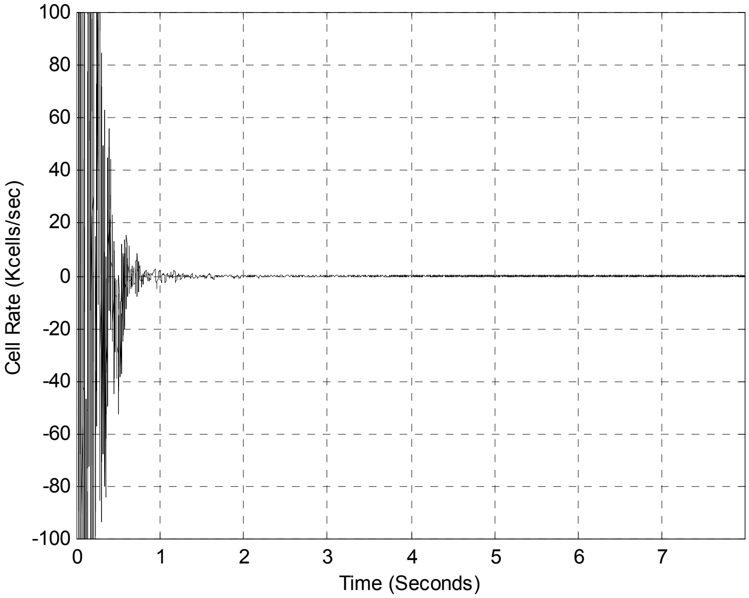

Figure 4.14 Set-Point Error when Queue Depth is Actively Controlled. Estimates a and β are within 1% of their correct values. Note that this is comparable to Figure 4.4

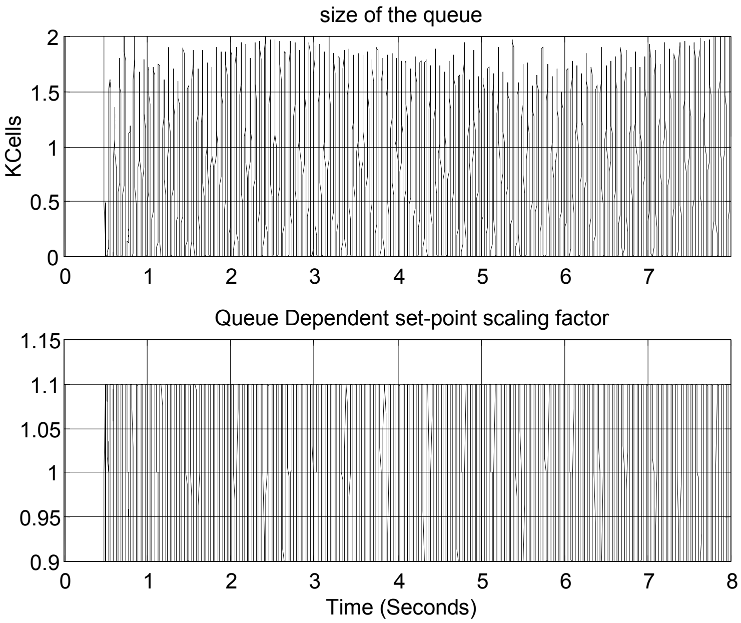

Figure 4.15 Queue Depth and Set-Point Scaling Factor $\eta(n)$ when Queue Target is 100-200 Cells. Too aggressive queue depth control can lead to instability.

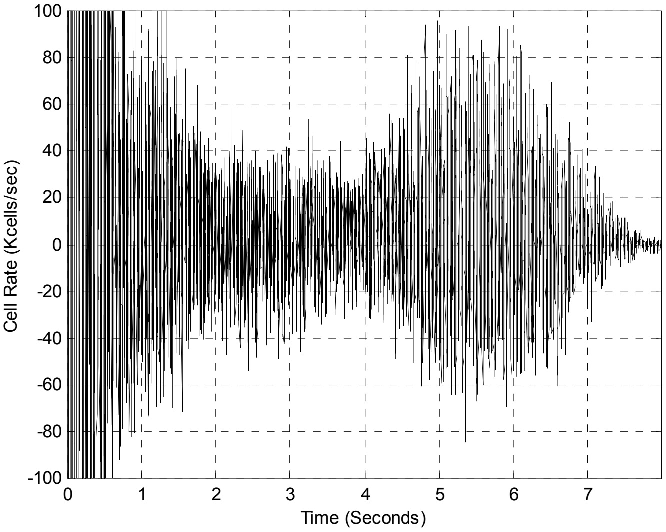

Figure 4.16 Set-Point error, $y^*(n \mid n-d-V)-y(n)$. Using too aggressive queue depth control, poor performance can result.

4.5 Biasing Issues

4.5.1 Generalizing the Plant by Incorporating Noise

Until this point, ABR traffic that is non-responsive to the explicit rate $u(n)$ of port *j* has been characterized as a constant C (see (4.1)). This characterization is plausible if the non-responsiveness is due to a set of characteristics of the source. For example, a source may be entitled to a minimum cell rate (MCR) that exceeds the explicit rates proposed by port *j*, or the source provides data at a fixed rate below the offered explicit rate of port *j*. However, an ABR source may be non-responsive to port *j* because it is responsive to another port $i(\neq j)$ of another switch. The explicit rates of port *j* are no more likely to be constant than those of port *i*. Therefore a more realistic traffic model for port *j* includes non-zero variance in its non-responsive traffic. Specifically, a zero-mean, white Gaussian noise signal $\varpi(n)$*,* which is uncorrelated to $u(n)$, is added to the plant (4.1).

$$y(n) = \mathbf{B}^T \mathbf{u}(n-d) + C + \varpi(n) \tag{4.22}$$

The signal $(C + \varpi(n))$ can be viewed as the non-responsive traffic having mean C and variance σ_{ϖ}^2. Let $y_{\varpi}(n) \equiv y(n) - \varpi(n)$ be the plant output without noise. Figure 4.17 shows the modified identification process incorporating the plant noise.

A complete study of the implications of plant noise on convergence and stability does not appear here. However, it appears feasible to extend the results from the noiseless case of Section 3.2 to the $\varpi(n) \neq 0$ case using techniques similar to those of Chapter 8 of [63].

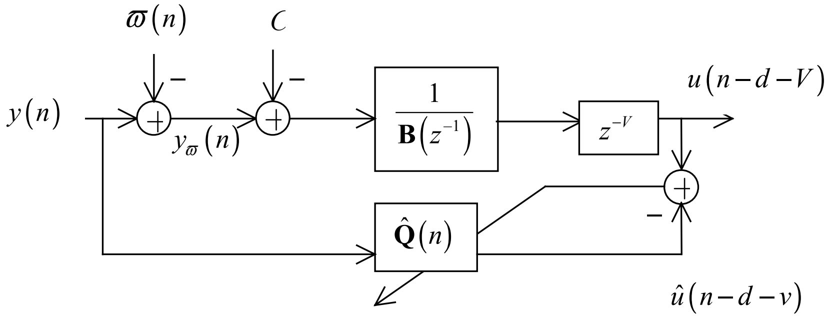

Figure 4.17 Identification Process Incorporating Plant Noise $\varpi(n)$

A parameter estimation process is said to be biased if the mean of the estimates are not equal to the parameters being estimated. In Section 3.2, the controller identification process of (4.3) through (4.6) is shown to converge to its Weiner solution. For the noiseless case $\varpi(n)=0$ and $y(n)=y_\varpi(n)$, the unbiased Weiner solution $\mathbf{Q}_{UB}$ is

$$\begin{aligned}\mathbf{Q}_{UB} \equiv \mathbf{Q}_0 &= \left\{E\left[\mathbf{y}(n)\mathbf{y}(n)^T\right]\right\}^{-1} E\left[\mathbf{y}(n)u(n-d-V)\right] \\ &= \left\{E\left[\mathbf{y}_\varpi(n)\mathbf{y}_\varpi(n)^T\right]\right\}^{-1} E\left[\mathbf{y}_\varpi(n)u(n-d-V)\right].\end{aligned} \tag{4.23}$$

When $\varpi(n)\neq 0$, the biased Weiner solution $\mathbf{Q}_B$ is

$$\begin{aligned}\mathbf{Q}_B &= \left\{E\left[\mathbf{y}(n)\mathbf{y}(n)^T\right]\right\}^{-1} E\left[\mathbf{y}(n)u(n-d-V)\right] \\ &= \left\{E\left[\mathbf{y}_\varpi(n)\mathbf{y}_\varpi(n)^T\right]+\sigma_\varpi^2\mathbf{I}\right\}^{-1} E\left[\mathbf{y}_\varpi(n)u(n-d-V)\right].\end{aligned} \tag{4.24}$$

Clearly $\mathbf{Q}_B \neq \mathbf{Q}_{UB}$ when $\varpi(n)\neq 0$.

4.5.2 Related Work

The biasing effect of $\varpi(n)\neq 0$ on Adaptive Approximate Inverse Control was previously reported ([59], [60], [65], [66]).

The accompanying recommendations focus on adding a second adaptive filter $\hat{\mathbf{B}}(n)$, which includes a DC tap, to estimate the plant. This estimate will be unbiased, as the noise $\varpi(n)$ occurs on the output of the estimated plant (**B**).

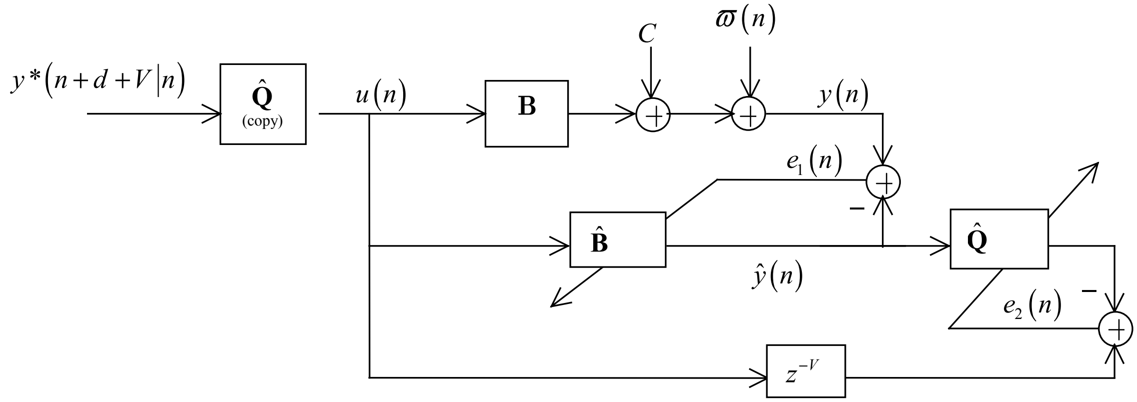

Figure 4.18 A First Method for Removing Bias from $\hat{\mathbf{Q}}(n)$ [59]

Figure 4.18 shows a controller estimation process that identifies $\hat{\mathbf{Q}}(n)$ with $\hat{\mathbf{B}}(n)$ in place of the true plant **B**. Widrow [59] argues that since the filter $\hat{\mathbf{B}}(n)$ is free of output noise, and if $\hat{\mathbf{B}}(n)$ has a sufficient number of parameters, then $\hat{\mathbf{B}}(n)$ converges to **B** in the mean, and $\hat{\mathbf{Q}}(n)$ will converge without bias to $\mathbf{Q}_{UB}$. The second scheme, shown in Figure 4.19, identifies $\hat{\mathbf{Q}}(n)$ with an offline process. The modeling signal $model(n)$ can be chosen by the operator to produce fast convergence of $\hat{\mathbf{Q}}(n)$. This second scheme is fundamentally an indirect adaptive controller – the plant **B** is estimated by $\hat{\mathbf{B}}(n)$ and the control parameters $\hat{\mathbf{Q}}(n)$ are determined using $\hat{\mathbf{B}}(n)$. As such, this second method is similar to that of Yahagi (discussed in Section 3.1.2.3), where $\hat{\mathbf{Q}}(n)$ is determined from $\hat{\mathbf{B}}(n)$ using matrix calculations. All indirect methods, including those shown in

Figure 4.18 and Figure 4.19, require extra calculations as compared to direct schemes since both $\hat{\mathbf{B}}(n)$ and $\hat{\mathbf{Q}}(n)$ are calculated.

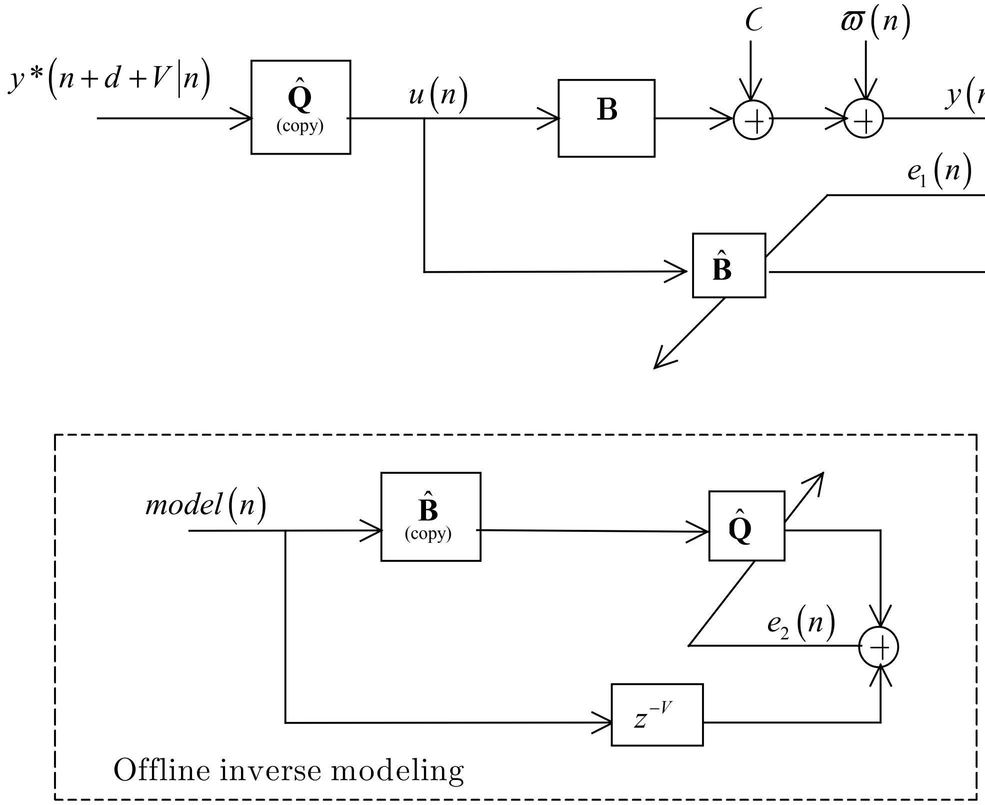

Figure 4.19 A Second Method for Removing Bias from $\hat{\mathbf{Q}}(n)$ [59]. The top figure is used to estimate $\hat{\mathbf{B}}$. The bottom figure depicts and off-line process to estimate $\hat{\mathbf{Q}}$ using the estimate $\hat{\mathbf{B}}$ from the top figure.

The schemes of Figure 4.18 and Figure 4.19 have intuitive merit, yet both lack complete analysis. Preliminary, often heuristic, results are presented in [59]. The possibility of poor estimates of $\hat{\mathbf{B}}(n)$ motivates yet another architecture (see Chapter 7 of [59]) to reduce the sensitivity of $\hat{\mathbf{Q}}(n)$ to the parameter errors in $\hat{\mathbf{B}}(n)$. However, this new architecture filters its adaptation error, and thus, like the rejected controller of Section 3.1.3.1, cannot be assured to converge.

Another possible solution to the biasing problem is to extend the estimated plant model to explicitly characterize the noise. One popular algorithm is the Pseudo Linear Regression Algorithm [63], also known as the Extended Least Squares Algorithm. In its general form, this algorithm uses an ARMAX model

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

where $w\left(n\right)$ is white noise. The estimates of $C\left(z^{-1}\right)$ are estimated along with the other plant parameters $A\left(z^{-1}\right)$ and $B\left(z^{-1}\right)$. For each time *n*, both a priori and a posteriori estimates are created, the former to estimate $\hat{y}\left(n\right)$, and the latter to find a posteriori estimates of $w\left(n\right)$. This method works well if the noise is well modeled by $C\left(z^{-1}\right)w\left(n\right)$. However, care must be taken to ensure convergence. Specifically, the estimate $1/\hat{C}\left(n\right)$ must remain positive real since this expression filters the adaptation error. If a good a priori estimate of $C\left(z^{-1}\right)$ exists, this estimate can be used to improve the likelihood of convergence. In short, Pseudo Linear Regression finds bias-free plant estimates if the model noise is well approximated by $C\left(z^{-1}\right)w\left(n\right)$. However, there is a higher computational cost – estimates are required for $C\left(z^{-1}\right)$ in addition to $A\left(z^{-1}\right)$ and $B\left(z^{-1}\right)$, and convergence is no longer assured unless extra steps are performed.

The Simple Hyperstable Adaptive Recursive Filter (SHARF) algorithm [43] offers another method to find bias-free estimates. Like the Pseudo Linear Regression algorithm, it requires added computational complexity and requires care to ensure convergence.

4.5.3 Reducing Estimation Bias

This section presents a novel method for reducing the biasing effect of plant noise described in Section 4.5.1. Unlike the previous suggestions of Section 4.5.2, this strategy does not require additional adaptive filter coefficients, e.g. $\hat{\mathbf{B}}(n)$ in Figure 4.18 and Figure 4.19 or $\hat{\mathbf{C}}(n)$ of the Pseudo Linear Regression Algorithm, and is thereby computationally less expensive. Further, this bias-reducing strategy poses no threat to global stability, as was the case with the methods of Section 4.5.2.

The strategy employed is reparameterization. Instead of adaptively finding $\hat{\mathbf{Q}}(n)$ by estimating $\hat{u}(n-d-V)$ as in (4.3), repeated here:

$$\begin{aligned}\hat{u}(n-d-V) &= \mathbf{y}(n)^T\hat{\mathbf{Q}}(n)\\ &= \hat{q}_0(n)y(n)+\hat{q}_1(n)y(n-1)+\cdots+\hat{q}_{dQ}(n)y(n-dQ),\end{aligned}\tag{4.25}$$

use the following reparameterized adaptive model to estimate $y(n-\pi)$:

$$\begin{aligned}\hat{y}(n-\pi) &= \hat{\theta}_0(n)u(n-d-V)scale_{yu}+\hat{\theta}_1(n)y(n)+\hat{\theta}_2(n)y(n-1)+\cdots\\ &\quad+\theta_\pi(n)y(n-(\pi-1))+\hat{\theta}_{\pi+1}(n)y(n-(\pi+1))+\cdots+\hat{\theta}_{dQ}(n)y(n-dQ)\\ &= \boldsymbol{\varphi}(n)^T\hat{\boldsymbol{\theta}}(n)\end{aligned}\tag{4.26}$$

for some appropriately chosen integer π, $0\le\pi\le dQ$, and

$$\boldsymbol{\varphi}(n)=\left[u(n-d-V)scale_{yu},y(n),\ldots y(n-(\pi-1)),y(n-(\pi+1)),\ldots,y(n-dQ)\right]^T$$

where $scale_{yu}$ is an operator chosen constant. Normalized Least Mean Square (NLMS) adaptation is performed using

$$e_{y_\pi}(n)=y(n-\pi)-\hat{y}(n-\pi),\tag{4.27}$$

$$\hat{\boldsymbol{\theta}}(n+1)=\hat{\boldsymbol{\theta}}(n)+\frac{\mu\boldsymbol{\varphi}(n)e_{y_\pi}(n)}{\boldsymbol{\varphi}(n)^T\boldsymbol{\varphi}(n)}.\tag{4.28}$$

For each n, $\hat{\boldsymbol{\theta}}(n+1)$ is translated into the controller FIR $\hat{\mathbf{Q}}^{\theta}(n+1)$ using

$$\begin{aligned}\hat{\mathbf{Q}}^{\theta}(n+1) = \frac{1}{\hat{\theta}_0(n+1)scale_{yu}} \Big[& -\hat{\theta}_1(n+1), -\hat{\theta}_2(n+1), \ldots, \\ & -\hat{\theta}_{\pi}(n+1), 1, -\hat{\theta}_{\pi+1}(n+1), \ldots, -\hat{\theta}_{dQ}(n+1)\Big]^T\end{aligned} \tag{4.29}$$

Note that (4.26) through (4.28) do not attempt to include a characterization of the noise, nor attempt to otherwise filter the adaptation error. Such techniques, including those of [63] and [59], often require strictly positive real (SPR) assumptions on the "noise filter" or some other plant aspect. Violation of such an assumption compromises convergence, both theoretically and practically. By avoiding any adaptation error filtering, the reparameterized adaptation of (4.26) through (4.28) causes $\hat{\boldsymbol{\theta}}(n)$ to converge to its Weiner solution. This Weiner solution will be biased, but as shown below, the biasing is decreased for the reparameterized case as compared to the non-reparameterized case.

Frequently when a plant's pole polynomial is estimated, this pole polynomial is assumed to be monic and the remaining plant parameters are scaled accordingly. However, in the present case, the magnitude of the first term of $Q_0(z^{-1})$ is ideally close to zero. Numerical difficulties arise in any estimation scheme that treats $Q_0(z^{-1})$ as a monic polynomial. This is the reason for choosing a non-zero π for the purpose of estimating $Q_0(z^{-1})$. Ideally π is chosen as

$$\pi = \arg\max_i \left|q_{0,i}\right|,$$

although any π such that $\left|q_{0,\pi}\right|$ is "relatively large" will do.

For the noiseless case, i.e. $\varpi(n)=0$, both the original non-reparameterized adaptation scheme ((4.3) through (4.6)) and the reparameterized scheme ((4.26) through (4.28)) have unbiased Weiner solutions. Let the unbiased Weiner solution for the non-reparameterized case and reparameterized case be $\mathbf{Q}_{UB}$ and $\boldsymbol{\theta}_{UB}$ respectively:

$$\begin{aligned}\mathbf{Q}_{UB} &= \left\{E\left[\mathbf{y}(n)\mathbf{y}(n)^T\right]\right\}^{-1} E\left[\mathbf{y}(n)u(n-d-V)\right] \\ &= \left\{E\left[\mathbf{y}_\varpi(n)\mathbf{y}_\varpi(n)^T\right]\right\}^{-1} E\left[\mathbf{y}_\varpi(n)u(n-d-V)\right]\end{aligned} \tag{4.30}$$

$$\begin{aligned}\boldsymbol{\theta}_{UB} &= \left\{E\left[\boldsymbol{\varphi}(n)\boldsymbol{\varphi}(n)^T\right]\right\}^{-1} E\left[\boldsymbol{\varphi}(n)y(n-\pi)\right] \\ &= \left\{E\left[\boldsymbol{\varphi}_\varpi(n)\boldsymbol{\varphi}_\varpi(n)^T\right]\right\}^{-1} E\left[\boldsymbol{\varphi}_\varpi(n)y_\varpi(n-\pi)\right].\end{aligned} \tag{4.31}$$

Further, define the transformation of $\boldsymbol{\theta}_{UB}$ to $\mathbf{Q}_{UB}^{\theta}$ as

$$\mathbf{Q}_{UB}^{\theta} \equiv \frac{1}{scale_{yu}\theta_{UB,0}}\left[-\theta_{UB,1},-\theta_{UB,2},\ldots-\theta_{UB,\pi},1,-\theta_{UB,\pi+1},\ldots,-\theta_{UB,dQ}\right]^T. \tag{4.32}$$

Note that $\mathbf{Q}_{UB}^{\theta}=\mathbf{Q}_{UB}$ if perfect inversion is assumed (Assumption 5 on page 83).

When $\varpi(n)\neq 0$, the Weiner solutions for both the non-reparameterized case $\mathbf{Q}_B$ and reparameterized case $\boldsymbol{\theta}_B$ are biased.

$$\begin{aligned}\mathbf{Q}_{B} &= \left\{E\left[\mathbf{y}(n)\mathbf{y}(n)^T\right]\right\}^{-1} E\left[\mathbf{y}(n)u(n-d-V)\right] \\ &= \left\{E\left[\mathbf{y}_\varpi(n)\mathbf{y}_\varpi(n)^T\right]+\sigma_\varpi^2\mathbf{I}\right\}^{-1} E\left[\mathbf{y}_\varpi(n)u(n-d-V)\right]\end{aligned} \tag{4.33}$$

$$\begin{aligned}\boldsymbol{\theta}_{B} &= \left\{E\left[\boldsymbol{\varphi}(n)\boldsymbol{\varphi}(n)^T\right]\right\}^{-1} E\left[\boldsymbol{\varphi}(n)y(n-\pi)\right] \\ &= \left\{E\left[\boldsymbol{\varphi}_\varpi(n)\boldsymbol{\varphi}_\varpi(n)^T\right]+\sigma_\varpi^2\operatorname{diag}\{0,1,1,\ldots,1\}\right\}^{-1} E\left[\boldsymbol{\varphi}_\varpi(n)y_\varpi(n-\pi)\right],\end{aligned} \tag{4.34}$$

where $\boldsymbol{\varphi}_\varpi(n)$ is defined

$$\boldsymbol{\varphi}_{\varpi}(n) \equiv \Big[u(n-d-V)\,scale_{yu}, y_{\varpi}(n), \\ \ldots, y_{\varpi}\big(n-(\pi-1)\big), y_{\varpi}\big(n-(\pi+1)\big), \ldots, y_{\varpi}(n-dQ)\Big]^{T}.$$

Noting the bias error in $\mathbf{Q}_B$ and $\boldsymbol{\theta}_B$ as vectors $\mathbf{Q}_{BE} \equiv \mathbf{Q}_B - \mathbf{Q}_{UB}$ and $\boldsymbol{\theta}_{BE} \equiv \boldsymbol{\theta}_B - \boldsymbol{\theta}_{UB}$, then from (4.33) and (4.34)

$$\left\{E\left[\mathbf{y}_{\varpi}(n)\mathbf{y}_{\varpi}(n)^{T}\right] + \sigma_{\varpi}^{2}\mathbf{I}\right\}(\mathbf{Q}_{UB} + \mathbf{Q}_{BE}) = E\left[\mathbf{y}_{\varpi}(n)u(n-d-V)\right]$$

$$\mathbf{Q}_{BE} = -\left\{E\left[\mathbf{y}_{\varpi}(n)\mathbf{y}_{\varpi}(n)^{T}\right] + \sigma_{\varpi}^{2}\mathbf{I}\right\}^{-1}\sigma_{\varpi}^{2}\begin{bmatrix} q_{UB,0} \\ q_{UB,1} \\ \vdots \\ q_{UB,\pi} \\ \vdots \\ q_{UB,dQ} \end{bmatrix} \tag{4.35}$$

and

$$\left\{E\left[\boldsymbol{\varphi}_{\varpi}(n)\boldsymbol{\varphi}_{\varpi}(n)^{T}\right] + \sigma_{\varpi}^{2}\,\mathrm{diag}\{0,1,1,\ldots,1\}\right\}(\boldsymbol{\theta}_{UB} + \boldsymbol{\theta}_{BE}) = E\left[\boldsymbol{\varphi}_{\varpi}(n)\,y_{\varpi}(n-\pi)\right]$$

$$\boldsymbol{\theta}_{BE} = \left\{E\left[\boldsymbol{\varphi}_{\varpi}(n)\boldsymbol{\varphi}_{\varpi}(n)^{T}\right] + \sigma_{\varpi}^{2}\begin{bmatrix} 0 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}\right\}^{-1}\frac{\sigma_{\varpi}^{2}}{q_{UB,\pi}^{\theta}}\begin{bmatrix} 0 \\ q_{UB,0}^{\theta} \\ q_{UB,1}^{\theta} \\ \vdots \\ q_{UB,\pi-1}^{\theta} \\ q_{UB,\pi+1}^{\theta} \\ \vdots \\ q_{UB,dQ}^{\theta} \end{bmatrix} \tag{4.36}$$

It is possible to translate $\boldsymbol{\theta}_{BE}$ into an analytical expression for the bias error $\mathbf{Q}_{BE}^{\theta}$. However, the non-linearity of the translation (4.29) and (4.32) obscures any added intuition provided by such an analytical expression. Instead, what follows are heuristic arguments claiming that σ_{ϖ}^{2} has a larger biasing effect on $\mathbf{Q}_B$ than on $\boldsymbol{\theta}_B$, and thus $\mathbf{Q}_{B}^{\theta}$.

Consider the large σ_{ϖ}^{2} case. As σ_{ϖ}^{2} increases, (4.35) indicates that $\mathbf{Q}_{BE} \to -\mathbf{Q}_{UB}$, or $\mathbf{Q}_{B} \to \mathbf{0}$. Such a controller produces an all-zero control signal, i.e. doing nothing is better than attempting any non-trivial control, the biasing effect is so great. In contrast, as σ_{ϖ}^{2} becomes large in (4.36), the matrix

$$\left\{E\left[\boldsymbol{\varphi}_{\varpi}(n)\boldsymbol{\varphi}_{\varpi}(n)^{T}\right]+\sigma_{\varpi}^{2}\operatorname{diag}\{0,1,1,\ldots,1\}\right\} \tag{4.37}$$

becomes increasingly diagonal. (It also becomes increasingly ill-conditioned, but avoids singularity since $E\left[u(n-d-V)^{2}\right]\neq 0$.) As (4.37) becomes more diagonal, from (4.36) the first term of $\boldsymbol{\theta}_{BE}$, $\theta_{BE,0}$, becomes close to zero. Surprisingly, as the noise increases, $\theta_{B,0} \approx \theta_{UB,0}$ and thus $Q_{B,\pi}^{\theta}$ is only slightly biased (and not equal to zero, as in the non-reparameterized case). By construction, the πth tap of the controller is one of its most significant taps.

Even when σ_{ϖ}^{2} is not excessively large, the reparameterized case seems to have an advantage, although the explanation is somewhat more heuristic. If $scale_{yu}$ could be chosen so that

$$\left\|\left\{E\left[\mathbf{y}_{\varpi}(n)\mathbf{y}_{\varpi}(n)^{T}\right]+\sigma_{\varpi}^{2}\mathbf{I}\right\}^{-1}\right\| \approx \left\|\left\{E\left[\boldsymbol{\varphi}_{\varpi}(n)\boldsymbol{\varphi}_{\varpi}(n)^{T}\right]+\sigma_{\varpi}^{2}\operatorname{diag}\{0,1,1,\ldots,1\}\right\}^{-1}\right\|, \tag{4.38}$$

and if (4.36) is multiplied by $-q_{UB,\pi}^{\theta}$, the result compares favorably to (4.35). The expression $\mathbf{Q}_{BE}$ of (4.35) includes $q_{UB,\pi}$ in its right-most vector. However, the expression of $\boldsymbol{\theta}_{BE}$ has a zero instead of $q_{UB,\pi}^{\theta}$ in its right-most vector. The same is true for $-q_{UB,\pi}^{\theta}\boldsymbol{\theta}_{BE}$. Recall that $q_{UB,\pi}$ is one of largest magnitude taps of $\mathbf{Q}_{B}^{\theta}$. It is therefore quite plausible that $\left\|-q_{UB,\pi}^{\theta}\boldsymbol{\theta}_{BE}\right\| < \left\|\mathbf{Q}_{BE}\right\|$, i.e. the reparameterized case is less biased than the non-parameterized case.

The key advantage of reparameterizing can be seen by comparing (4.33) and (4.34). By making $y(n-\pi)$ the value to be estimated and including $u(n-d-V)$ in the regressor vector, there is an "advantage" in the auto-correlation matrix, yet there is no "disadvantage" in the cross-correlation matrix. The structure of the Weiner solution shows that if white noise corrupts only the value being estimated, with no noise corrupting the regressor vector, no bias results. If there is noise only on the regressor vector and not the signal being estimated, large bias results. The reparameterization suggested above creates an amount of bias somewhere between these two extremes.

Before presenting the simulation results, a few comments on $scale_{yu}$ are in order. As briefly implied above, $scale_{yu}$ should be chosen to reduce eigenvalue spread of the auto-correlation matrix $E\left[\boldsymbol{\varphi}(n)\boldsymbol{\varphi}(n)^{T}\right]$. As discussed in 4.3.2, reducing the eigenvalue spread of the auto-correlation matrix causes a desirable reduction in the convergence time. An appropriate choice of $scale_{yu}$, or even a reasonable guess, can significantly reduce convergence time when the reparameterized scheme is used.

There are several ways to determine a helpful $scale_{yu}$, including

$$scale_{yu} = \sqrt{\frac{\tilde{\sigma}_y^2}{\tilde{\sigma}_u^2}} \tag{4.39}$$

where $\tilde{\sigma}_y^2$ and $\tilde{\sigma}_u^2$ are sample-mean estimates of the variance of y and u respectively. The scaler $scale_{yu}$ is treated as a constant, but in practice could be occasionally updated using on-line measurements.

The simulation experiments presented below demonstrate the reduction of bias that occurs with reparameterization. As in Section 4.2,

$B\left(z^{-1}\right)=z^{-10}\left(2+9z^{-1}+8z^{-2}+3z^{+3}\right)$, $C=200$, $dQ=30$, and $V=10$. The sample time is $T_s=1$ msec. The bandwidth available for explicit rate traffic, $y^*\left(n\,|\,n-d-V\right)$, is modeled as a Gaussian random process with $E\left[y^*\left(n\,|\,n-d-V\right)\right]=1$ Mcps, ${\sigma_{y^*}}^2=484$ kcps. When reparameterization is performed, $\pi=9$, as this is the largest magnitude tap of $\mathbf{Q}_{UB}$ (Figure 4.20). To reduce the eigenvalue spread of the autocorrelation matrix, the method of *Reducing Means Via Downsampled Estimates* is used, as described in Section 4.3.5.

When the plant output noise $\varpi\left(n\right)$ is a zero-mean, Gaussian random process with variance $\sigma_{\varpi}^2=120$ kcps, without reparameterization, biasing is pronounced. Figure 4.20 shows the impulse response the parameter estimate $\hat{\mathbf{Q}}$ and the optimal, unbiased $\mathbf{Q}_{UB}$ after 8 seconds (8000 samples) of convergence. The estimate $\hat{\mathbf{Q}}$ bears a poor resemblance to $\mathbf{Q}_{UB}$.

When $\hat{\mathbf{Q}}$ is convolved with **B**, instead of the expected impulse at $V=10$ (see (3.10)), Figure 4.21 demonstrates that $\hat{\mathbf{Q}}$ poorly inverts **B**.

Comparing bode plots of $\mathbf{Q}_{UB}$, $\mathbf{Q}_B$, and $\hat{\mathbf{Q}}$ in Figure 4.22 shows that $\hat{\mathbf{Q}}$ does indeed closely resemble $\mathbf{Q}_B$ and poorly resembles $\mathbf{Q}_{UB}$. The set-point error, $y^*\left(n\,|\,n-d-V\right)-y_{\varpi}\left(n\right)$, is shown in Figure 4.23. In contrast, the reparameterized case, with $\pi=9$, shows much less bias. The impulse response of $\hat{\mathbf{Q}}$ is much closer to $\mathbf{Q}_{UB}^{\theta}$, as shown in Figure 4.24. Convolution of $\hat{\mathbf{Q}}$ and **B** reveals an impulse at delay $V=10$, as shown in Figure 4.25.

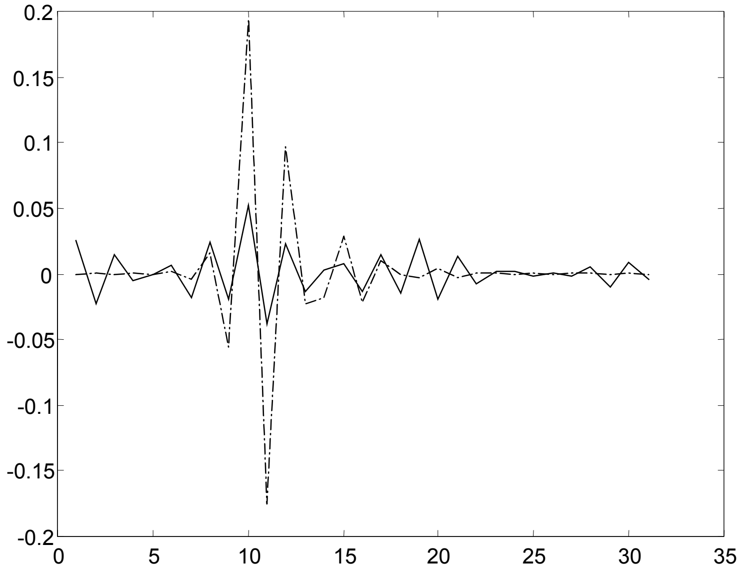

Figure 4.20 Impulse Response of $\hat{\mathbf{Q}}$ (solid line) and $\mathbf{Q}_{UB}$ (dash-dot line) with $\sigma_{\varpi}^{2} = 120$ kcps. No Reparameterization.

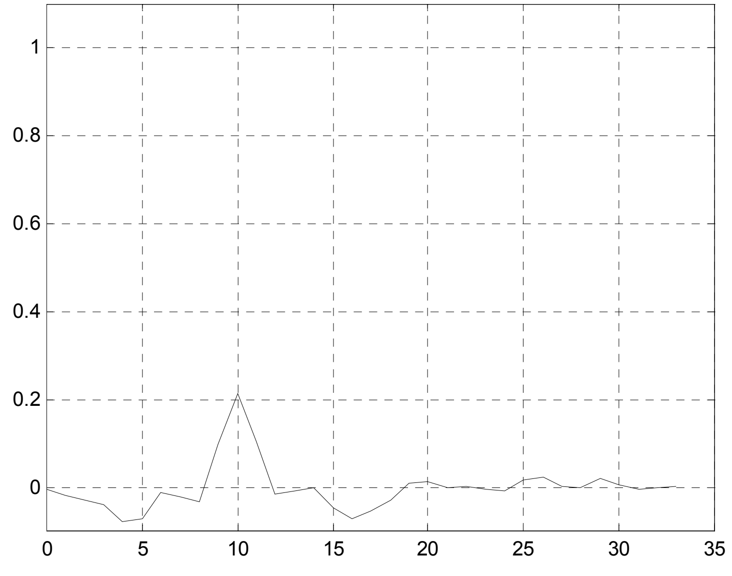

Figure 4.21 Convolution of $\mathbf{B}$ and $\hat{\mathbf{Q}}$, with $\sigma_{\varpi}^{2} = 120$ kcps. No Reparameterization.

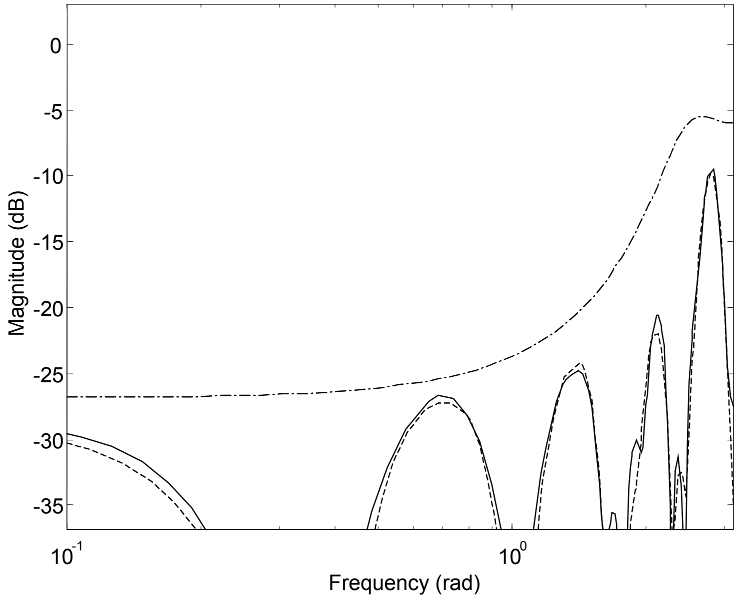

Figure 4.22 Bode Plot of $\mathbf{Q}_{UB}$ (dash-dot line), $\mathbf{Q}_{B}$ (dashed line), and $\hat{\mathbf{Q}}$ (solid line) with $\sigma_{\varpi}^{2} = 120$ kcps. No Reparameterization

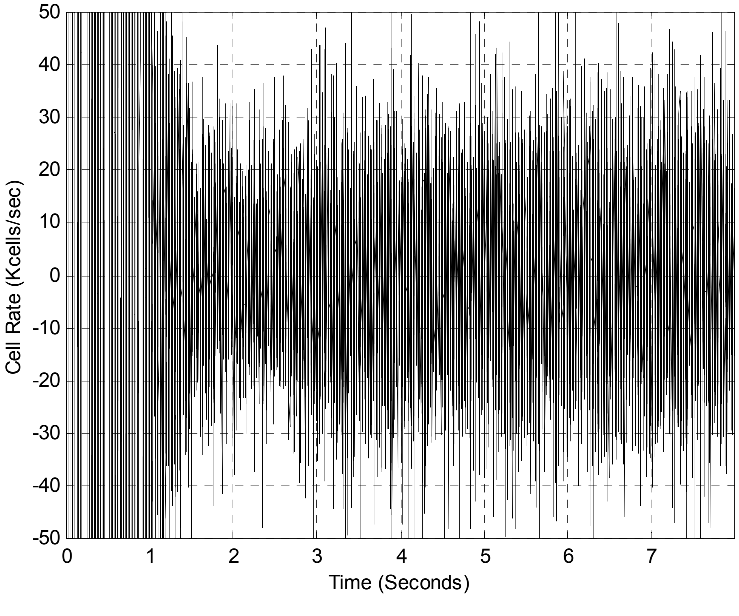

Figure 4.23 Set-Point error, $y^{*}(n \mid n-d-V) - y_{\varpi}(n)$, with $\sigma_{\varpi}^{2} = 120$ kcps. No Reparameterization.

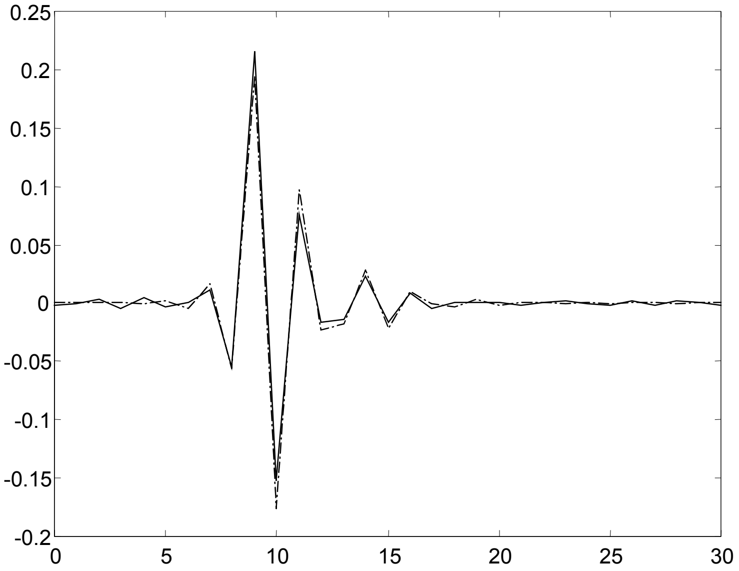

Figure 4.24 Impulse response of $\hat{\mathbf{Q}}$ (solid line) and $\mathbf{Q}_{UB}$ (dashed line) with $\sigma_{\varpi}^2 = 120$ kcps. Using Reparameterization. $\pi = 9$, $scale_{yu} = 12.5$.

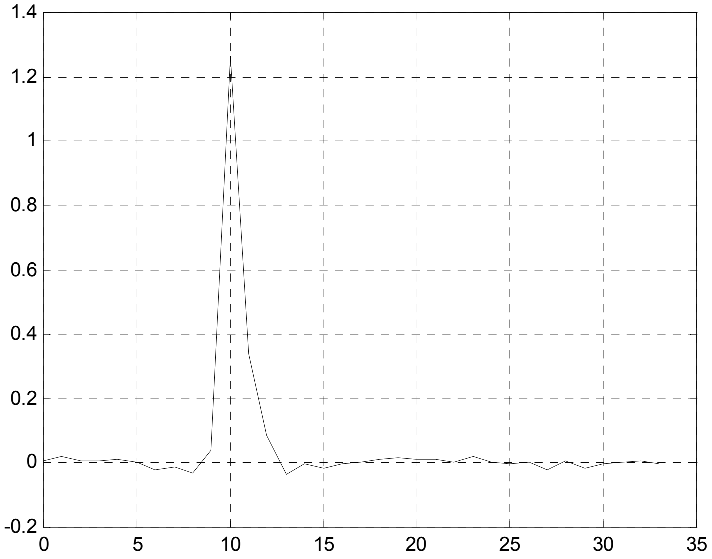

Figure 4.25 Convolution of **B** and $\hat{\mathbf{Q}}$, with $\sigma_{\varpi}^2 = 120$ kcps. Using Reparameterization. $\pi = 9$, $scale_{yu} = 12.5$.

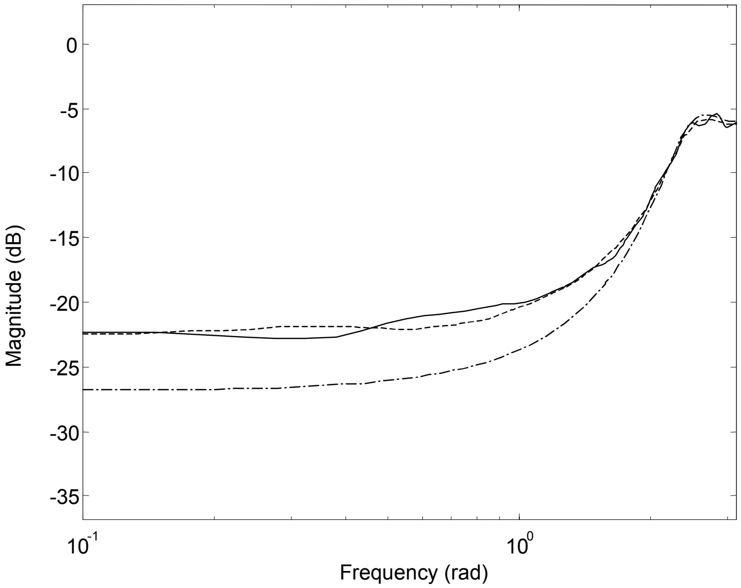

Figure 4.26 Bode Plot of $\mathbf{Q}_{UB}$ (dash-dot line), $\mathbf{Q}_B$ (dashed line), and $\hat{\mathbf{Q}}$ (solid line) with $\sigma_{\varpi}^2 = 120$ kcps. Using Reparameterization. $\pi = 9$, $scale_{yu} = 12.5$.

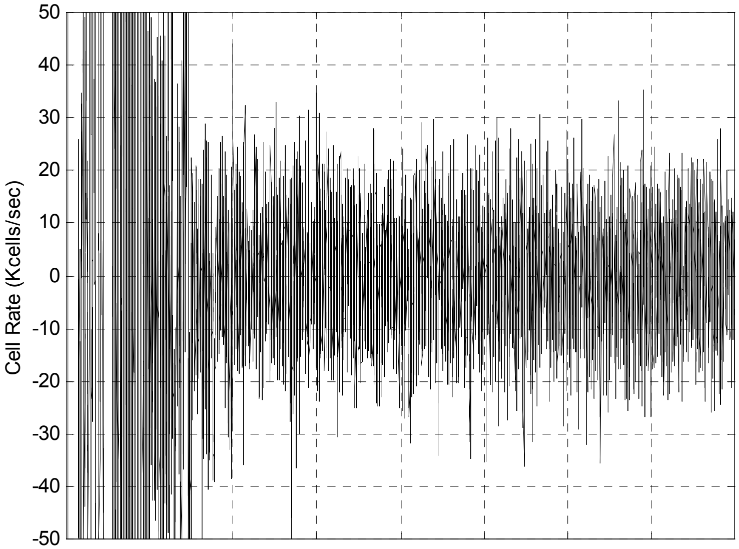

Figure 4.27 Set-Point error, $y^*(n \mid n-d-V) - y_{\varpi}(n)$, with $\sigma_{\varpi}^2 = 120$ kcps. Using Reparameterization. $\pi = 9$, $scale_{yu} = 12.5$.

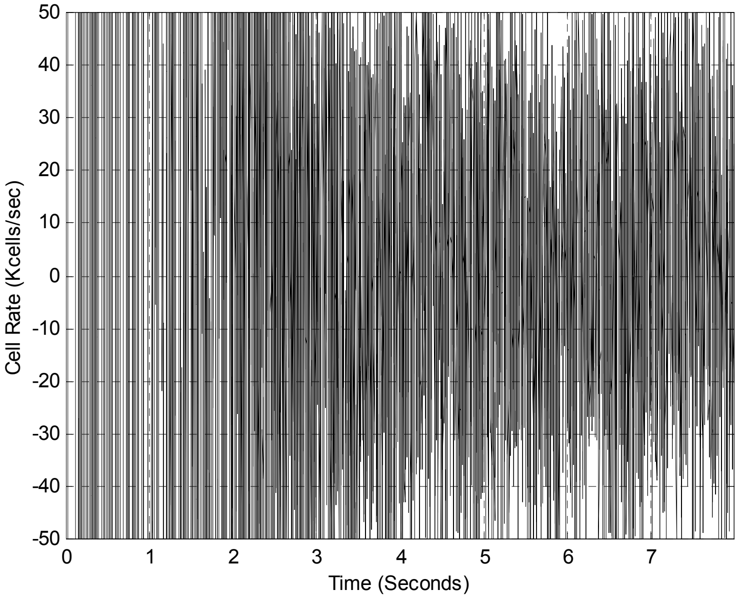

Figure 4.28 Set Point error, $y^*(n \mid n-d-V) - y_{\varpi}(n)$, with $\sigma_{\varpi}^2 = 120$ kcps. Using Reparameterization, $\pi = 9$, $scale_{yu} = 1$. Note that convergence is much slower than in Figure 4.27.

In Figure 4.26, the bode plots of $\mathbf{Q}_{UB}^{\theta}$, $\mathbf{Q}_{B}^{\theta}$, and $\hat{\mathbf{Q}}$ show that $\hat{\mathbf{Q}}$ well approximates $\mathbf{Q}_{B}^{\theta}$ and nearly well approximates $\mathbf{Q}_{UB}^{\theta}$. The upward shift of $\hat{\mathbf{Q}}$ as compared to $\mathbf{Q}_{UB}^{\theta}$ is consistent with the slight overshoot observed in the delayed impulse of Figure 4.25.

The set-point error, as shown in Figure 4.27, is noticeably superior as compared to the non-parameterized case shown in Figure 4.23.

Note that the results of Figure 4.24 through Figure 4.27 use $scale_{yu} = 12.5$ as calculated by (4.39). To demonstrate the improvement in convergence rate from this reasonable estimate of $scale_{yu}$, the experiment shown in Figure 4.27 is repeated with $scale_{yu} = 1$. The results are shown in Figure 4.28. Note that the measured eigenvalue spread of the auto-correlation matrix increased from 15 to 145 for $scale_{yu} = 12.5$ and 1 respectively.

Consider now the large σ_{ϖ}^2 case. As σ_{ϖ}^2 is increased from 120 to 300 kcps, the non-reparameterized case produces $\mathbf{Q} = \mathbf{0}$, as predicted by (4.35). The impulse response and set-point error are shown in Figure 4.29 and Figure 4.30 respectively.

In contrast this large increase in σ_{ϖ}^2 degrades the performance of the reparameterized case only slightly, as shown in Figure 4.31 through Figure 4.34 (compare to Figure 4.24 through Figure 4.27).

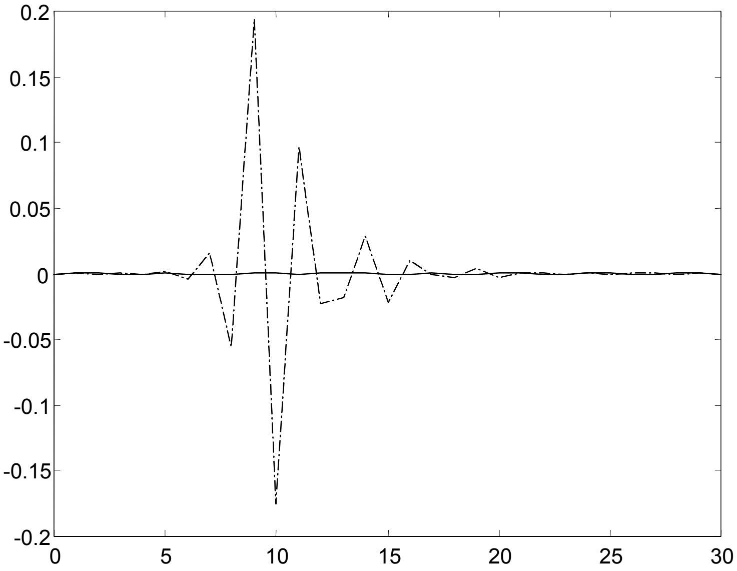

Figure 4.29 Impulse response of $\hat{\mathbf{Q}}$ (solid line) and $\mathbf{Q}_{UB}$ (dashed line) with σ_{ϖ}^2=300 kcps. No Reparameterization.

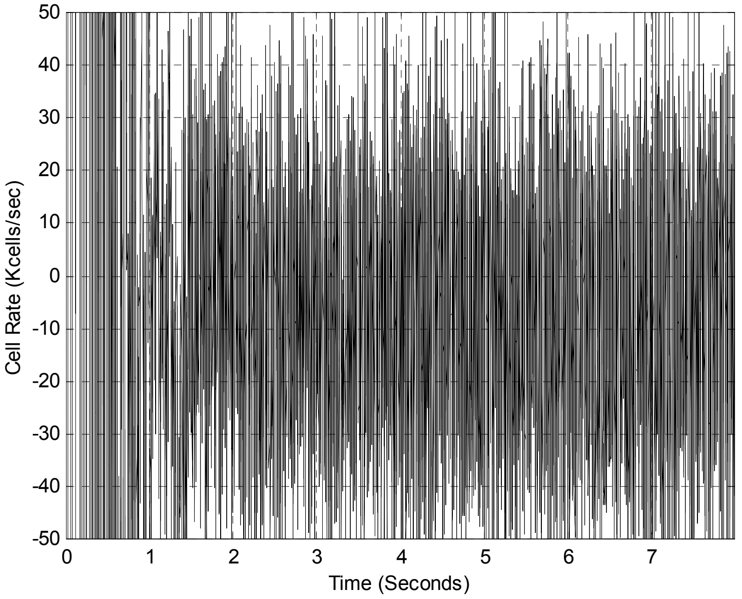

Figure 4.30 Set Point error, $y^*(n \mid n-d-V) - y_{\varpi}(n)$, with σ_{ϖ}^2=300 kcps. No Reparameterization.

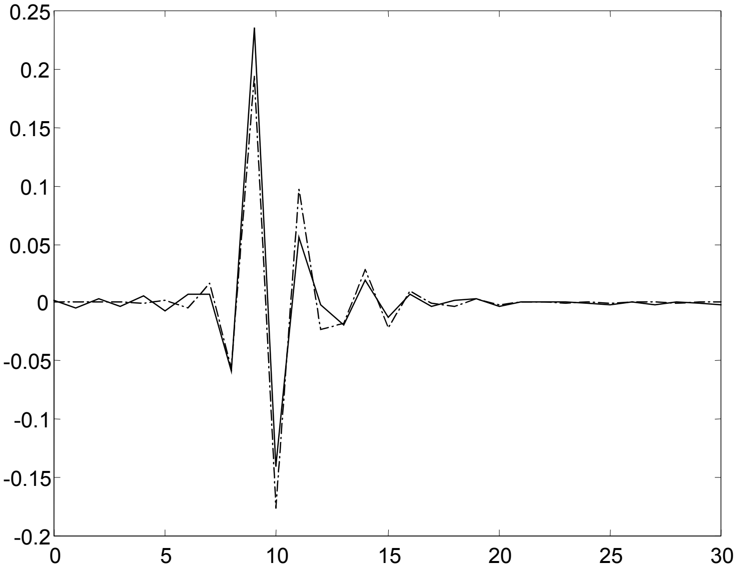

Figure 4.31 Impulse response of $\hat{\mathbf{Q}}$ (solid line) and $\mathbf{Q}_{UB}$ (dashed line) with σ_{ϖ}^2=300 kcps. Using Reparameterization. $\pi = 9$, $scale_{yu} = 12.5$.

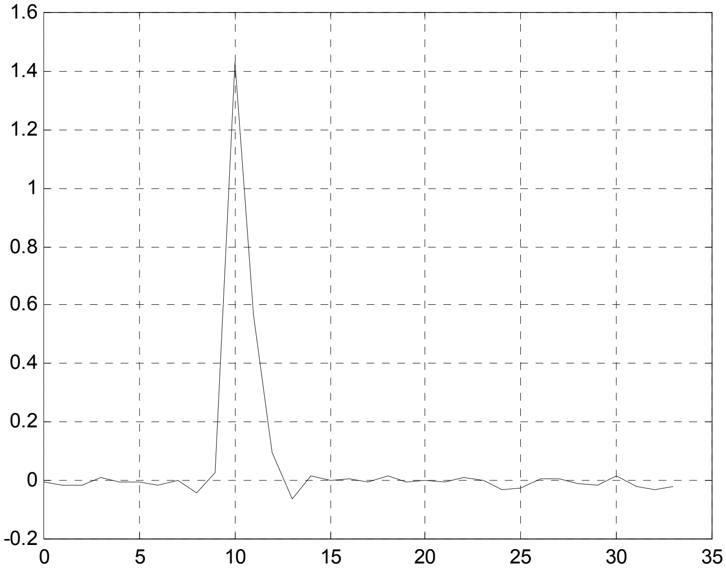

Figure 4.32 Convolution of **B** and $\hat{\mathbf{Q}}$, with σ_{ϖ}^2=300 kcps. Using Reparameterization. $\pi = 9$, $scale_{yu} = 12.5$.

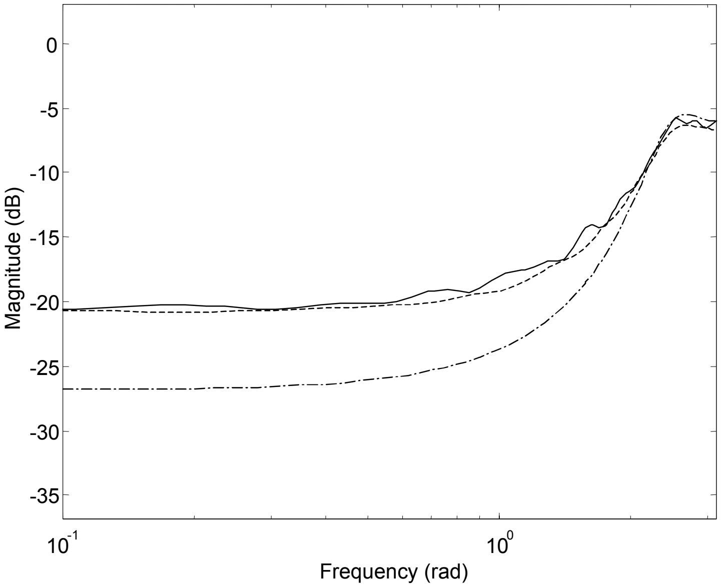

Figure 4.33 Bode Plot of $\mathbf{Q}_{UB}$ (dash-dot line), $\mathbf{Q}_B$ (dashed line), and $\hat{\mathbf{Q}}$ (solid line) with σ_{ϖ}^2=300 kcps. Using Reparameterization. $\pi = 9$, $scale_{yu} = 12.5$.

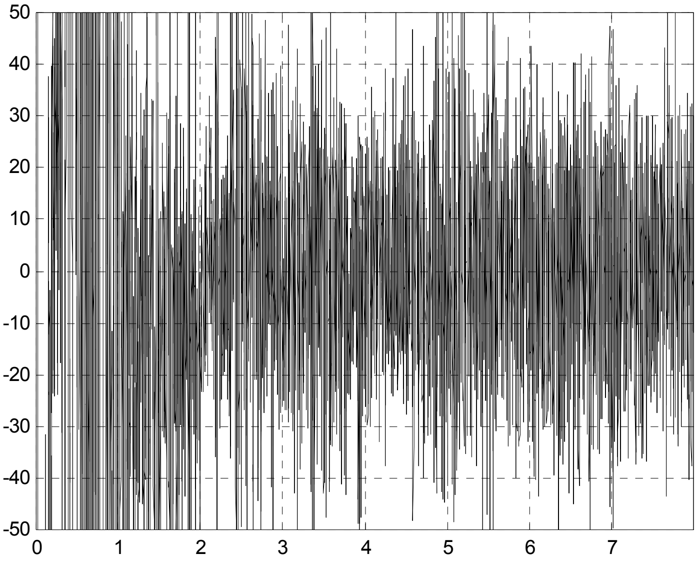

Figure 4.34 Set-Point error, $y^*(n \mid n-d-V)-y_{\varpi}(n)$, with σ_{ϖ}^2=300 kcps. Using Reparameterization. $\pi=9$, $scale_{yu}=12.5$.

4.6 Chapter Summary

This chapter addresses three important practical issues that arise in the application of Adaptive Control theory to problem of congestion control.

Section 4.3 discovers that the bandwidth desired by a congestion controller is likely to be a large mean, small variance signal. Such a set-point is problematic for the convergence rate of the congestion controller originally presented and studied in Chapter 3. The offered solution “subtracts and adds” the large signal means and reduces convergence speed dramatically.

Section 4.4 integrates queue depth management into the original congestion controller. This is achieved by slightly modifying the original set-point up or down, depending on the current queue's size compared to its target.

Section 4.5 recognizes that port j will likely serve both responsive and non-responsive ABR sources. These non-responsive sources were originally assumed to offer a constant bandwidth load to port j. Assuming a non-constant load for the non-responsive sources makes the model more realistic, but exposes the original control effort to biasing problems. The offered solution can dramatically reduce the biasing effect without significantly increasing the computational cost over the original controller. Further, unlike previously published solutions to the biasing issue, the solution of Section 4.5.3 does not compromise convergence of the controller.

CHAPTER 5 DISSERTATION SUMMARY AND CONCLUSIONS

This chapter contains a summary of the contributions of this dissertation as well as possible future research directions.

5.1 Summary

This dissertation addresses congestion control for explicit rate controlled data networks.

Chapter 1 introduces the concept of congestion control for data networks, with a focus on explicit rate controlled networks. The Available Bit Rate (ABR) service category of Asynchronous Transfer Mode (ATM) networks, or *ATM ABR*, is an example of an explicit rate controlled network, and is the example studied in this dissertation. The characteristics of ATM ABR are reviewed. Previous contributions in this field of study are summarized.

Chapter 2 addresses the modeling of an ATM ABR explicit rate controlled network. One simple model is shown to be similar to the implicit model of the Uniform Tracking [24] algorithm, thereby explicitly connecting the Uniform Tracking algorithm to the theory of adaptive control. As an extension of this simple model, a more detailed model is presented, which affords analytically tractable control schemes. Simple control strategies are suggested. Chapter 2 also introduces the Blending Effect, a property of multiple switch rate-controlled plant models. The Blending Effect makes high fidelity plant modeling intractable. Further, it can be minimized through other means and is therefore ignored for the remainder of the dissertation.

Chapter 3 explores control strategies for the explicit rate controlled network plant proposed in Chapter 2. The plant of Chapter 2 is potentially non-minimum phase. Therefore the control strategies explored in Chapter 3 are all capable of controlling non-minimum phase plants. After evaluation, one scheme, Adaptive Approximate Inverse Control, is selected for its relatively low computational cost, realizability, and attractive convergence and stability properties. This control scheme was independently developed for the present congestion control application. However, it shares many characteristics with the previously proposed Adaptive Inverse Control [59]. Upon selecting Adaptive Approximate Inverse Control as the preferred control strategy, convergence and stability properties are explored. Theorem 3.1 through Theorem 3.3 result. The convergence and stability analysis presented in Section 3.2 significantly contribute to the understanding of this control paradigm.

Chapter 4 extends the control algorithm proposed in Chapter 3. Application of a control strategy, such as Adaptive Approximate Inverse Control, to a real-word application, such as ATM ABR congestion control, often benefits from application-specific tailoring. The three modifications suggested in Chapter 4 receive particular attention due to the generality of the problems they address, problems that are likely to arise in other applications. The first modification significantly reduces convergence times of the control parameters. The second modification extends the system to explicitly model and control the buffer queue depth. The third modification extends the plant to allow a disturbance due to rate-varying non-responsive sources. Unlike other solutions for this issue, this third modification addresses the bias on controller parameters without significantly increasing the computational load or compromising controller convergence. Simulations are shown throughout Chapter 4 to validate each solution, as well as the overall control strategy.

5.2 Future Research Directions

There are several potential directions for future research. One path examines real-world protocols and networks in an attempt to improve the fidelity of the plant model. This almost certainly creates a more complex plant model. Modeling the Blending Effect introduced in Chapter 2 is but one possibility. Other modeling extensions include delayed or lost data (e.g. resource management cells), non-linearities due to rate and buffer saturations, bursty sources, and other phenomena.

Another direction for research is to examine different control methodologies outside of linear adaptive control, the primary tool of this dissertation. However, the dynamic nature of data networks suggests that any successful control strategy maintain an adaptive quality.

Yet another direction, albeit further afield from this work, is application of adaptive control principles to TCP/IP congestion control. TCP/IP is a set of protocols widely employed in today's data networks, including the Internet. TCP/IP presently employs coarse congestion control only at the end-points of a connection. Current efforts are underway to improve TCP/IP congestion control. This likely requires delivering additional information from internal nodes to TCP/IP end-points. Researchers must determine what information should be delivered, how it should be delivered, and how best to use that information to effect improved congestion control. However, any suggested change must heavily consider backwards compatibility with the large installed base.

APPENDEX

This Appendix contains supporting lemmas for Section 3.2.2.

Lemma A.1 $\mathbf{R}=E\left[\mathbf{y}(n)\mathbf{y}(n)^T\right]$ is positive definite, thus full rank.

Proof: Consider an arbitrary, real, non-zero vector $\mathbf{x}_+ \equiv [\mathbf{x}^T, x_{DC}]^T$ that is length $dQ+1$, i.e. $\mathbf{x}$ is length dQ and x_{DC} is a scalar.

$$\mathbf{x}_+{}^T\mathbf{R}\mathbf{x}_+ = E\left[\left(\mathbf{x}^T\mathbf{y}(n)+x_{DC}y_{DC}\right)^2\right]=E\left[\left(\mathbf{x}^T\mathbf{y}(n)\right)^2+2\mathbf{x}^T\mathbf{y}(n)x_{DC}y_{DC}+\left(x_{DC}y_{DC}\right)^2\right]\geq 0$$

Since $\mathbf{x}_+{}^T\mathbf{R}\mathbf{x}_+$ cannot be negative, determine if it can equal zero. Define $\eta \equiv x_{DC}y_{DC}$ and find the roots η_{root} of $\mathbf{x}_+{}^T\mathbf{R}\mathbf{x}_+=0$.

$$\begin{aligned}\eta_{root} &= \frac{-2\mathbf{x}^T E\left[\mathbf{y}(n)\right]\pm\sqrt{4\mathbf{x}^T E\left[\mathbf{y}(n)\right]E\left[\mathbf{y}(n)^T\right]\mathbf{x}-4\mathbf{x}^T E\left[\mathbf{y}(n)\mathbf{y}(n)^T\right]\mathbf{x}}}{2}\\ &= \mathbf{x}^T E\left[\mathbf{y}(n)\right]\pm\sqrt{-\mathbf{x}^T E\left[\left(\mathbf{y}(n)-E\left[\mathbf{y}(n)\right]\right)\left(\mathbf{y}(n)^T-E\left[\mathbf{y}(n)^T\right]\right)\right]\mathbf{x}}\\ &= \mathbf{x}^T E\left[\mathbf{y}(n)\right]\pm\sqrt{-\mathbf{x}^T\boldsymbol{\sigma}^2\mathbf{x}}\end{aligned}$$

However, $\mathbf{x}^T\boldsymbol{\sigma}^2\mathbf{x}>0$, thus η_{root} must be complex, contradicting the definition of x_{DC}. Therefore, for any real $\mathbf{x}_+$, $\mathbf{x}_+{}^T\mathbf{R}\mathbf{x}_+>0$ and $\mathbf{R}$ is full rank, completing the proof.

Lemma A.2 A solution of (3.34) is $\mathbf{Q}_0=\left[\breve{\mathbb{Q}}_0{}^T, \dfrac{-E\left[y(n)\right]\sum_i\breve{\mathbb{Q}}_{0,i}+E\left[u(n-V-d)\right]}{y_{DC}}\right]^T$.

Proof: Describe

$$\left[\breve{\mathbb{Q}}_0{}^T, \frac{-E\left[y(n)\right]\sum_i\breve{\mathbb{Q}}_{0,i}+E\left[u(n-V-d)\right]}{y_{DC}}\right]^T \tag{A.1}$$

as the *proposed solution* of (3.34). Equation (3.34) defines $dQ+1$ linear equations. The proposed solution is a solution of the equation defined by the first row of (3.34), as shown by

$$\begin{aligned}
&\sum_{i=1}^{dQ} E\left[y(n)y(n-i-1)\right]\breve{\mathbb{Q}}_{0,i} + y_{DC}E\left[y(n)\right]\frac{-E\left[y(n)\right]\sum_{i}\breve{\mathbb{Q}}_{0,i} + E\left[u(n-V-d)\right]}{y_{DC}} \\
&= \sum_{i=1}^{dQ} E\left[y(n)y(n-i-1) - \left(E\left[y(n)\right]\right)^2\right]\breve{\mathbb{Q}}_{0,i} + E\left[y(n)\right]E\left[u(n-V-d)\right] \\
&= \sum_{i=1}^{dQ} E\left[\breve{y}(n)\breve{y}(n-i-1)\right]\breve{\mathbb{Q}}_{0,i} + E\left[y(n)\right]E\left[u(n-V-d)\right],
\end{aligned} \tag{A.2}$$

where the last line uses the fact that

$$\begin{aligned}
E\left[y(n)y(n-i)\right] - \left(E\left[y(n)\right]\right)^2 &= E\left[\left(y(n)-E\left[y(n)\right]\right)\left(y(n-i)-E\left[y(n-i)\right]\right)\right] \\
&= E\left[\breve{y}(n)\breve{y}(n-i)\right].
\end{aligned}$$

From the equation defined by the first row of (3.35)

$$\sum_{i=1}^{dQ} E\left[\breve{y}(n)\breve{y}(n-i-1)\right]\breve{\mathbb{Q}}_{0,i} = E\left[\breve{y}(n)\breve{u}(n-V-d)\right]. \tag{A.3}$$

Substituting (A.3) into the last line of (A.2) gives

$$\begin{aligned}
&\sum_{i=1}^{dQ} E\left[y(n)y(n-i-1)\right]\breve{\mathbb{Q}}_{0,i} + y_{DC}E\left[y(n)\right]\frac{-E\left[y(n)\right]\sum_{i}\breve{\mathbb{Q}}_{0,i} + E\left[u(n-V-d)\right]}{y_{DC}} \\
&= E\left[\breve{y}(n)\breve{u}(n-V-d)\right] + E\left[y(n)\right]E\left[u(n-V-d)\right] \\
&= E\left[y(n)u(n-V-d)\right] - E\left[y(n)\right]E\left[u(n-V-d)\right] + E\left[y(n)\right]E\left[u(n-V-d)\right] \\
&= E\left[y(n)u(n-V-d)\right],
\end{aligned} \tag{A.4}$$

since $\breve{y}(n)$ and $\breve{u}(n-V-d)$ are both zero mean. Thus from (A.4), the proposed solution does in fact solve the first row of (3.34). In a similar manner, the proposed solution can be shown to solve rows 2 through dQ of (3.34).

It is trivial to show that the proposed solution solves the last row of (3.34):

$$\begin{aligned}E\left[y(n)\right]y_{DC}\sum_{i=1}^{dQ}\breve{\mathbb{Q}}_{0,i}+\left(y_{DC}\right)^2\frac{-E\left[y(n)\right]\sum_{i}\breve{\mathbb{Q}}_{0,i}+E\left[u(n-V-d)\right]}{y_{DC}}\\ =y_{DC}E\left[u(n-V-d)\right]\end{aligned}\tag{A.5}$$

Thus the proposed solution is indeed a solution to (3.34), concluding the proof.

Lemma A.3 $E\left[e^*(n)\right]=0$

Proof:

$$\begin{aligned}&E\left[e^*(n)\right]=E\left[u(n-V-d)\right]-E\left[\mathbf{Q}_0(n)^T\mathbf{y}(n)\right]\\ &=E\left[u(n-V-d)\right]-E\left[y(n)\right]\sum_{i}\breve{\mathbb{Q}}_{0,i}+y_{DC}\left[\frac{-E\left[y(n)\right]\sum_{i}\breve{\mathbb{Q}}_{0,i}+E\left[u(n-V-d)\right]}{y_{DC}}\right]\\ &=0\end{aligned}$$

completing the proof.

Lemma A.4 Since the elements of $\boldsymbol{\psi}(n)$ are Gaussian and independent, given (3.45), the expectation of an Odd Function of $\boldsymbol{\psi}(n)$ around $\psi_i(n)$ for $i\neq\zeta$ is zero.

Proof: A function of $\boldsymbol{\psi}(n)$, $\Gamma_i\left(\boldsymbol{\psi}(n)\right)$, is defined as an *Odd Function around* $\psi_i(n)$ if

$$\begin{aligned}&\Gamma_i\left(\psi_1(n),\psi_2(n),\ldots,\psi_i(n),\ldots,\psi_{dQ+1}(n)\right)\\ &\qquad=-\Gamma_i\left(\psi_1(n),\psi_2(n),\ldots,-\psi_i(n),\ldots,\psi_{dQ+1}(n)\right).\end{aligned}\tag{A.6}$$

Also define $\hat{\boldsymbol{\psi}}(n)$ as $\boldsymbol{\psi}(n)$ without the *i*th element, i.e.

$$\hat{\boldsymbol{\psi}}(n)\equiv\left[\psi_1(n),\psi_2(n),\ldots,\psi_{i-1}(n),\psi_{i+1}(n),\ldots,\psi_{dQ+1}(n)\right].\tag{A.7}$$

Because the elements of $\boldsymbol{\psi}(n)$ are independent

$$f_{\boldsymbol{\psi}(n)}\left(\boldsymbol{\psi}\right)=f_{\psi_i(n)}\left(\psi_i\right)f_{\hat{\boldsymbol{\psi}}(n)}\left(\hat{\boldsymbol{\psi}}\right),\tag{A.8}$$

and since $E\left[\psi_i(n)\right]=0$ and $\psi_i(n)$ is Gaussian,

$$f_{\psi_i(n)}\left(\psi_i\right) = f_{\psi_i(n)}\left(-\psi_i\right). \tag{A.9}$$

Then,

$$\begin{aligned}
E\left[\Gamma_i\left(\boldsymbol{\psi}(n)\right)\right] &= \int\limits_{\hat{\boldsymbol{\psi}}=-\infty}^{\infty} \int\limits_{\psi_i=-\infty}^{\infty} \Gamma_i\left(\boldsymbol{\psi}\right) f_{\psi_i(n)}\left(\psi_i\right) f_{\hat{\boldsymbol{\psi}}(n)}\left(\hat{\boldsymbol{\psi}}\right) d\psi_i d\hat{\boldsymbol{\psi}} \\
&= \int\limits_{\hat{\boldsymbol{\psi}}=-\infty}^{\infty} \left[\int\limits_{\psi_i=-\infty}^{\infty} \Gamma_i\left(\boldsymbol{\psi}\right) f_{\psi_i(n)}\left(\psi_i\right) d\psi_i \right] f_{\hat{\boldsymbol{\psi}}(n)}\left(\hat{\boldsymbol{\psi}}\right) d\hat{\boldsymbol{\psi}} \\
&= \int\limits_{\hat{\boldsymbol{\psi}}=-\infty}^{\infty} \left[\int\limits_{\psi_i=-\infty}^{0} \Gamma_i\left(\boldsymbol{\psi}\right) f_{\psi_i(n)}\left(\psi_i\right) d\psi_i + \int\limits_{\psi_i=0}^{\infty} \Gamma_i\left(\boldsymbol{\psi}\right) f_{\psi_i(n)}\left(\psi_i\right) d\psi_i \right] f_{\hat{\boldsymbol{\psi}}(n)}\left(\hat{\boldsymbol{\psi}}\right) d\hat{\boldsymbol{\psi}}.
\end{aligned}$$

Performing a change a variable on the first integral inside the brackets,

$$\begin{aligned}
E\left[\Gamma_i\left(\boldsymbol{\psi}(n)\right)\right] &= \int\limits_{\hat{\boldsymbol{\psi}}=-\infty}^{\infty} \left[\int\limits_{t=\infty}^{0} \Gamma_i\left(\psi_1,\psi_2,\ldots,-t,\ldots,\psi_{dQ+1}\right) f_{\psi_i(n)}\left(-t\right)\left(-1\right) dt \right] f_{\hat{\boldsymbol{\psi}}(n)}\left(\hat{\boldsymbol{\psi}}\right) d\hat{\boldsymbol{\psi}} \\
&\quad + \int\limits_{\hat{\boldsymbol{\psi}}=-\infty}^{\infty} \left[\int\limits_{\psi_i=0}^{\infty} \Gamma_i\left(\psi_1,\psi_2,\ldots,\psi_i,\ldots,\psi_{dQ+1}\right) f_{\psi_i(n)}\left(\psi_i\right) d\psi_i \right] f_{\hat{\boldsymbol{\psi}}(n)}\left(\hat{\boldsymbol{\psi}}\right) d\hat{\boldsymbol{\psi}} \\
&= \int\limits_{\hat{\boldsymbol{\psi}}=-\infty}^{\infty} \left[\int\limits_{t=0}^{\infty} -\Gamma_i\left(\psi_1,\psi_2,\ldots,t,\ldots,\psi_{dQ+1}\right) f_{\psi_i(n)}\left(t\right) dt \right] f_{\hat{\boldsymbol{\psi}}(n)}\left(\hat{\boldsymbol{\psi}}\right) d\hat{\boldsymbol{\psi}} \qquad \text{(A.10)} \\
&\quad + \int\limits_{\hat{\boldsymbol{\psi}}=-\infty}^{\infty} \left[\int\limits_{\psi_i=0}^{\infty} \Gamma_i\left(\psi_1,\psi_2,\ldots,\psi_i,\ldots,\psi_{dQ+1}\right) f_{\psi_i(n)}\left(\psi_i\right) d\psi_i \right] f_{\hat{\boldsymbol{\psi}}(n)}\left(\hat{\boldsymbol{\psi}}\right) d\hat{\boldsymbol{\psi}} \\
&= 0,
\end{aligned}$$

where the third line of (A.10) uses (A.6) and (A.9), thus completing the proof. This lemma was mentioned without proof in [62].

Lemma A.5 $\left|\gamma_{ij}\right| < 1$

Proof: Since $\left(\psi_i(n) - \psi_j(n)\right)^2 \geq 0$

$$\left(\psi_i(n)\right)^2 + \left(\psi_j(n)\right)^2 \geq 2\left(\psi_i(n)\right)\left(\psi_j(n)\right). \tag{A.11}$$

With $0 < \mu < 2$, (3.54) becomes

$$\begin{aligned}1-\gamma_{ij} &= \mu\left[\left(\mathbf{A}_{ii}+\mathbf{A}_{jj}\right)-2\mu\mathbf{G}_{ij}\right]\\ &= \mu\ E\left[\frac{\left(\psi_i(n)\right)^2+\left(\psi_j(n)\right)^2}{\boldsymbol{\psi}(n)^T\ \boldsymbol{\psi}(n)}-\frac{2\mu\left(\psi_i(n)\right)^2\left(\psi_j(n)\right)^2}{\left(\boldsymbol{\psi}(n)^T\ \boldsymbol{\psi}(n)\right)^2}\right].\end{aligned}$$

Squaring both sides of (A.11) produces

$$\begin{aligned}1-\gamma_{ij} &\geq \mu\ E\left[\frac{\left(\psi_i(n)\right)^2+\left(\psi_j(n)\right)^2}{\boldsymbol{\psi}(n)^T\ \boldsymbol{\psi}(n)}\ \left\{1-\frac{\mu}{2}\cdot\frac{\left(\psi_i(n)\right)^2+\left(\psi_j(n)\right)^2}{\boldsymbol{\psi}(n)^T\ \boldsymbol{\psi}(n)}\right\}\right]\\ &> 0\end{aligned} \tag{A.12}$$

where the strict inequality is due to Lemma 3.3. Conversely

$$1-\gamma_{ij} < \mu\left(\mathbf{A}_{ii}+\mathbf{A}_{jj}\right)=\mu\ E\left[\frac{\left(\psi_i(n)\right)^2+\left(\psi_j(n)\right)^2}{\boldsymbol{\psi}(n)^T\ \boldsymbol{\psi}(n)}\right]\leq\mu<2 \tag{A.13}$$

Taking (A.12) and (A.13) together, $0<1-\gamma_{ij}<2$, or, $\left|\gamma_{ij}\right|<1$, completing the proof. This was originally proven in [62].

Lemma A.6 $\left|H_{ii}\right|\leq 1/\left(dQ-2\right)\lambda_{\min}$

Proof: $\left|\mathbf{H}_{ii}\right|\leq tr\left[\mathbf{H}\right]=E\left[\dfrac{1}{\boldsymbol{\psi}(n)^T\ \boldsymbol{\psi}(n)}\right]=E\left[\dfrac{1}{\boldsymbol{\psi}(n)^T\ \boldsymbol{\Lambda}^{1/2}\boldsymbol{\Lambda}^{-1}\ \boldsymbol{\Lambda}^{1/2}\boldsymbol{\psi}(n)}\right]$

$$\left|\mathbf{H}_{ii}\right|\leq\frac{1}{\lambda_{\min}}E\left[\frac{1}{\boldsymbol{\psi}(n)^T\ \boldsymbol{\Lambda}^{-1/2}\boldsymbol{\Lambda}^{-1/2}\ \boldsymbol{\psi}(n)}\right]$$

$\boldsymbol{\upsilon}(n)\equiv\boldsymbol{\Lambda}^{-1/2}\ \boldsymbol{\psi}(n)$ is a vector of independent Gaussian random variables with and unit variance with the further property that all but the ζth term is zero mean, i.e. $E\left[\upsilon_i(n)\right]=0, i=1,\ldots,dQ+1, i\neq\zeta$ (see (3.45)). [62] shows that

$$E\left[\left(\sum_{\substack{i=1,\\ i\neq\zeta}}^{dQ+1}\upsilon_i(n)^2\right)^{-1}\right]=\frac{1}{dQ-2}, \tag{A.14}$$

and since

$$E\left[\left(\sum_{i=1}^{dQ+1}\upsilon_i(n)^2\right)^{-1}\right]\leq E\left[\left(\sum_{\substack{i=1,\\ i\neq\zeta}}^{dQ+1}\upsilon_i(n)^2\right)^{-1}\right]$$

then $|H_{ii}|\leq 1/(dQ-2)\lambda_{\min}$, thus completing the proof.

Lemma A.7 All entries of $\mathbf{F}$ are non-negative.

Proof: Consider off-diagonal and diagonal entries separately. For $i\neq j$

$$\mathbf{F}_{ij}=\mu^2\mathbf{G}_{ij}=\mu^2 E\left[\frac{(\psi_i(n))^2(\psi_j(n))^2}{\left(\boldsymbol{\psi}(n)^T\boldsymbol{\psi}(n)\right)^2}\right]\geq 0\,. \tag{A.15}$$

For diagonal entries, consider the fact that

$$E\left[\left(1-\frac{\mu(\psi_i(n))^2}{\boldsymbol{\psi}(n)^T\boldsymbol{\psi}(n)}\right)^2\right]\geq 0\,. \tag{A.16}$$

Expanding (A.16)

$$\begin{aligned}0&\leq E\left[1-\frac{2\mu(\psi_i(n))^2}{\boldsymbol{\psi}(n)^T\boldsymbol{\psi}(n)}+\frac{\mu^2(\psi_i(n))^4}{\left(\boldsymbol{\psi}(n)^T\boldsymbol{\psi}(n)\right)^2}\right]\\ &\leq 1-2\mu\mathbf{A}_{ii}+\mu^2\mathbf{G}_{ii} &\text{(A.17)}\\ &\leq \mathbf{F}_{ii}\end{aligned}$$

completes the proof. This lemma was mentioned in [62] with a different, longer proof.

Lemma A.8 For $0<\mu<2$, $\lim_{n\to 0} v(n)=0$ where $v(n)$ is defined by (A.21).

Proof: Consider the autonomous part of (3.55)

$$\tilde{\boldsymbol{\Omega}}(n+1)=\mathbf{F}\,\tilde{\boldsymbol{\Omega}}(n) \tag{A.18}$$

where $\tilde{\mathbf{\Omega}}(0) = \mathbf{\Omega}(0)$, noting that for $i = 1, \ldots, dQ+1$,

$$\tilde{\mathbf{\Omega}}_i(0) = \mathbf{C}_{ii}(0) = \left(E\left[\mathbf{L}(0)\mathbf{L}(0)^T\right]\right)_{ii} \geq 0. \tag{A.19}$$

From (A.18) and Lemma A.7, clearly

$$\tilde{\mathbf{\Omega}}_i(n) \geq 0,\ i = 1, \ldots, dQ+1,\ n \geq 0. \tag{A.20}$$

Now define

$$v(n) \equiv \sum_{i=1}^{dQ+1} \tilde{\mathbf{\Omega}}_i(n). \tag{A.21}$$

Substituting (3.56) into (A.18), pre-multiplying the result by a row vector of ones,

$$\begin{aligned}[1,1,1,\ldots,1]\tilde{\mathbf{\Omega}}(n+1) = [1,1,1,\ldots,1]\tilde{\mathbf{\Omega}}(n) - 2\mu[1,1,1,\ldots,1]\operatorname{diag}\{A_{ii}\}\tilde{\mathbf{\Omega}}(n) \\ +\mu^2[1,1,1,\ldots,1]\mathbf{G}\tilde{\mathbf{\Omega}}(n).\end{aligned} \tag{A.22}$$

Using the fact that $\sum_{j=1}^{dQ+1} \mathbf{G}_{ij} = \mathbf{A}_{ii}$,

$$\begin{aligned} v(n+1) &= v(n) - 2\mu\left[A_{11}, A_{22}, \ldots, A_{(dQ+1)(dQ+1)}\right]\tilde{\mathbf{\Omega}}(n) \\ &\quad +\mu^2\left[A_{11}, A_{22}, \ldots, A_{(dQ+1)(dQ+1)}\right]\tilde{\mathbf{\Omega}}(n) \\ &= v(n) - \mu(2-\mu)\sum_{j=1}^{dQ+1} \mathbf{A}_{jj}\tilde{\mathbf{\Omega}}_j(n). \end{aligned} \tag{A.23}$$

From (A.21) and Lemma 3.3

$$v(n+1) \leq \left(1 - \mu(2-\mu)\alpha_1\right)v(n).$$

By constraining μ such that $0 < \mu < 2$, the lemma is proven. This lemma was mentioned without complete proof in [62].

BIBLIOGRAPHY

[1] J. Kenney, Editor, *Traffic Management Specification Version 4.1*, available from [3], March 1999.

[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, pp. 1723-1738, October 1996.

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

[4] F. Bonomi, and K. Fendick, “Rate-based flow control framework for the available bit rate ATM service,” *IEEE Network*, 9 2 Mar-Apr 1995, pp. 25-39.

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

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

[7] J. Bennett and G. Tom Des Jardins, “Comments on the July PRCA Rate Control Baseline,” AF-TM 94-0682, available from [3], July 1994.

[8] 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.

[9] 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, pp. 227-246, Sep-Oct 1997.

[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*, Raleigh, NC, pp. 121-140, Oct. 1993.

[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*, New Orleans, December 1995.

[14] Z. Pan, E. Altman and T. Basar, “Robust Adaptive Flow Control in High Speed Telecommunication Networks,” *Proceedings of the 35th IEEE Conference on Decision and Control* , 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, pp. 2916-2921, Dec. 1997.

[16] E. Altman, Eitan, T. Basar, and R. Srikant, “Robust rate control for ABR sources,” *IEEE INFOCOM*, pp. 166-173, 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 1998, Atlanta, GA, pp. 1485-1491, June 1998.

[23] K. Laberteaux and C. Rohrs, “Application Of Adaptive Control To ATM ABR Congestion Control,” *Proceedings of Globecom ’98*, Sydney, Australia, 1998.

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

[25] 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.

[26] O. Smith, "A Controller to Overcome Dead Time," *ISA J.*, Vol. 6, No. 2, pp. 28-33, 1959.

[27] 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.

[28] S. Mascolo, D. Cavendish, and M. Gerla, "ATM Rate Based Congestion Control Using A Smith Prediction: An EPRCA Implementation," *IEEE Infocom '96*, San Francisco, CA, March 1996.

[29] S. Mascolo, "Classical Control Theory For congestion Avoidance in High-Speed Internet," *Proc. of Conf. Decision Control*, Phoenix, Dec. 1999.

[30] S. O'Halek, *Extending a Control System Model for Rate-based Congestion Control*, thesis, Mass. Inst. Tech. Department of Electrical Enginering and Computer Science, 1996.

[31] K. Laberteaux and C. Rohrs, "On the Convergence of a Direct Adaptive Controller for ATM ABR Congestion Control," *Proceedings of Int Conf Comm 2000*, June 2000.

[32] K. Laberteaux and C. Rohrs, "A Direct Adaptive Controller for ATM ABR Congestion Control," *Proceedings of Amer Control Conf 2000*, Chicago, IL, June 2000.

[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, September 2000.

[34] C. Rohrs, J. Melsa, D. Schultz, *Linear Control Systems*, McGraw-Hill, 1993.

[35] M. J. Grimble, "A control weighed minimum-variance controller for non-minimum phase systems," *Int. J. Control*, Vol. 33, No.4, p.751-762, 1981.

[36] M. J. Grimble, "Weighted minimum-variance self-tuning control," *Int. J. Control*, Vol. 36, No. 4, p. 597-609, 1982.

[37] A. Niederlinski and J. Moscinski, "Robust Implicit Adaptive Control for Nonminimumphase Plants," *IFAC Robust Adaptive Control*, Newcastle, Australia, p. 163-169, 1988.

[38] W. Press, S. Teukolsky, W. Vetteriling, B. Flannery, *Numerical Recipies in C, 2nd ed.,* Cambridge University Press, 1992.

[39] 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.

[40] A. Tanenbaum, *Computer Networks*, Prentice-Hall, Upper Saddle River, NJ, 1996.

[41] William Stallings, *High-Speed Networks – TCP/IP and ATM Design Principles,* Prentice-Hall, Upper Saddle River, NJ, 1998.

[42] Internet Engineering Task Force (IETF) web site, http://www.ietf.org.

[43] L. Larimore, J. Treichler, and C. Johnson "SHARF:An Algorithm for Adaptive IIR Digital Filters," *IEEE Trans. on Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 428-440, Aug. 1980.

[44] J. Postel, "Transmission Control Protocol," Internet RFC 793, available from [42], September 1981.

[45] R. T. Bradenl, "Requirements for Internet hosts - communication layers," Internet RFC 1122, available from [42], October 1989.

[46] V. Jacobson, "Congestion avoidance and control," *Proceedings of SIGCOMM '88 (ACM)*, Stanford, CA, August 1988.

[47] S. Blake et al., "An Architecture for Differentiated Services," Internet RFC 2475, available from [42], Dec. 1998.

[48] 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.

[49] S. Floyd, The RED Queue Management web site, http://www.aciri.org/floyd/red.html.

[50] K. Ramakrishnan and S. Floyd, "A Proposal to add Explicit Congestion Notification (ECN) to IP," Internet RFC 2481, available from [42], January 1999.

[51] D. Awduche, "Requirements for Traffic Engineering Over MPLS," Internet RFC 2702, available from [42], September 1999.

[52] R. Cochran, “ATM: Sales Finally Match the Hype,” Business Communications Review, pp. 40-44, January 1999.

[53] E. Krapf, “VOIP Services Emerging,” Business Communications Review, pp. 17, January 2000.

[54] N. Testi, “Handicapping the QOS Race,” Business Communications Review, pp. 46-52, May 2000.

[55] United States Internet Counsel, “State of the Internet: USIC's Report on Use & Threats in 1999,” available from http://www.usic.org, 2000.

[56] T. Dwight, ed., “Frame-Based ATM Over Sonet/SDH (FAST),” ATM Forum, af-fbatm-0151.000, available from [3], July 2000.

[57] A. Bragg, ed., “Differentiated UBR,” ATM Forum, addendum to Traffic Management Version 4.1, af-tm-0149.000, available from [3], July 2000.

[58] A. Oppenheim and R. Schafer, *Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1975.

[59] B. Widrow and E. Walach, *Adaptive Inverse Control*, Prentice-Hall, Englewood Cliffs, NJ, 1996.

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

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

[62] 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, July 1988, p. 680-691.

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

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

[65] B. Widrow and G. Plett, “’Intelligent’ Adaptive Inverse Control,” *IFAC 96*, San Francisco, CA, pp. 104-105, July 1996.

[66] B. Widrow and G. Plett, “Adaptive inverse control based on linear and nonlinear adaptive filtering,” *Proc. World Congress on Neural Networks 1996*, San Diego, CA, pp. 620-27, vol. 2., December 1997.

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