caltechCSTR:2002.009

Analysis and Design of AQM for stabilizing TCP *

Ki Baek Kim and Steven H. Low

Departments of CS and EE
California Institute of Technology
kkb@cisl.snu.ac.kr, slow@caltech.edu

Abstract

In this paper, we formulate the AQM (Active Queue Management) design problem for stabilizing a given TCP (Transmission Control Protocol) as state-space models. First, we propose a PD-type (Proportional-Derivative) control structure and by applying integral control action, a PID-type (Proportional-Integral-Derivative) control structure that is a unified AQM framework. Second, we compensate for delays in congestion measure explicitly by using a memory control structure. Third, we propose stabilizing optimal AQMs for linearized systems of the given TCP by minimizing linear quadratic costs of the transients in queue length, aggregate rate, jitter in the aggregate rate, and the congestion measure, which is called RHA (Receding Horizon AQM) in this paper. We also show that any AQM with an appropriate structure solves the same stabilizing optimal control problem with appropriate weighting matrices. We interpret existing AQMs, including a simplified RED (Random Early Detection) without a low-pass filter and saturation functions, REM (Random Exponential Marking), PI (Proportional-Integral) and a simplified AVQ without an adaptive virtual-queue dynamics, as different approximations of the unified AQM structure. Finally, we discuss the impact of each structure on performance from the results of the stabilizing optimal AQMs. We illustrate our results through simulation examples for the linearized system of the given TCP and queue dynamics.

*We submitted this paper to IEEE Conference on Decision and Control, March 1, 2002 (Paper ID: CDC02-REG1058). This paper updated the technical report [1]. The corresponding author, Ki Baek Kim is visiting Caltech from Engineering Research Center, Seoul National University, Korea

1 Introduction

Congestion control is a distributed iterative procedure to maximize utilization of given network resources and share them efficiently among competing sources. It consists of local algorithms executed dynamically at sources (Transmission Control Protocol, or TCP) and at links (active queue management, or AQM). Links update, implicitly or explicitly, a measure of congestion, and feed it back to sources by dropping or marking arrival packets. In response, sources adjust their rates based on the feedback information from links in their paths. Popular TCP algorithms include Reno (and its variants) [2] and Vegas [3], and popular AQM algorithms include DropTail, RED (Random Early Detection) and its variants [4], [5].

There are many ways to formulate congestion control problems. In this paper, as in [6], we formulate TCP as a dynamical system consisting of sources, queues, and feedback control from AQM, and AQM as a control action, where the queue itself is a dynamical system. Thus, the equilibrium points of a TCP/AQM pair and dynamic behavior generally depend on both the TCP/AQM algorithms and the queue dynamics. The TCP algorithm determines the underlying utility functions that are implicitly optimized by the TCP, while AQM stabilizes this equilibrium. While the primal-dual model in [7, 8, 9, 10, 11] provides a unified framework to compare different TCP algorithms in terms of their utility functions and equilibrium allocation, a similar model is lacking to compare and understand various AQMs in terms of structures and performance. As a result, AQM algorithms are only compared in the literature through simulations. Thus, it is important to develop a unified control structure for analysis and design of AQMs.

To our knowledge, existing AQM algorithms only use the current dynamic information and thus do not compensate for large delays explicitly. Since they only use the current dynamic information, they may have difficulty to stabilize the given TCP algorithm and thus get maximum throughput in the presence of large delays. Thus, we need to investigate how to compensate for large delays explicitly.

Maximizing utilization of given network resources corresponds to minimizing the stabilizing cost of the given dynamical system. In general, it is almost impossible to find the best AQM that stabilizes the given nonlinear TCP and queue dynamics with minimum cost. Thus, it will be interesting to study a stabilizing optimal AQM even for a linearized system near the equilibrium point.

In this paper, we formulate the AQM design problem for stabilizing a given TCP as state-space models. First, we propose a PD-type (Proportional-Derivative) control structure, which the primal-dual models in [7, 8, 9, 10, 11, 12, 13, 14, 15] and existing AQM algorithms in [4, 5, 6, 16, 17] did not capture. By applying integral control action, we derive a PID-type (Proportional-Integral-Derivative) control structure that is a unified mathematical framework for analysis and design of AQM. Second, we propose a memory control structure to compensate for delays in congestion measure explicitly. Third, we propose stabilizing optimal AQMs for linearized systems of the given TCP by minimizing linear quadratic costs of the transients in queue length, aggregate rate, jitter in the aggregate rate, and the congestion measure under the

assumption that the global information is given. We also show that any AQM with an appropriate structure solves the same optimal control problem with appropriate weighting matrices. We interpret existing AQMs including a simplified RED without a low-pass filter and saturation functions, REM/PI, and a simplified AVQ without an adaptive virtual-queue dynamics, as different approximations of the unified AQM structures. Finally, we discuss the impact of each structure on performance from the results of the stabilizing optimal AQMs.

Although we consider TCP Reno as an exercise for the analysis and design of AQM, the proposed procedures and results in this paper can be applied directly to the AQM design problem for any TCP algorithms. Also, although we consider the AQM design problem at routers in order to support TCP algorithms, the proposed control structures in this paper can also be implemented at sources (end-users). As a first step to analyze and design AQM, we don't analyze stability of the nonlinear stochastic system in this paper as in [4, 5, 6, 16, 17].

In Section 2, we review a primal-dual model of TCP/AQM. In Section 3, we formulate the AQM design problem based on real-queue dynamics for the given TCP as a state-space model. Then, we derive a PD-type AQM and by applying integral control action, a PID-type AQM. In Section 4, we compensate for delays in congestion measure explicitly by using a memory control structure. In Section 5, we propose stabilizing optimal AQMs, which is called receding horizon AQM (RHA) in this paper. In Section 6, we illustrate our results via simulation examples. Finally, we present our conclusions in Section 7.

2 Primal-Dual model of TCP/AQM

In this section, we describe a general model for congestion control that allows us to study the equilibrium and dynamics of general TCP/AQM for arbitrary network topology, routing, and delays [7, 8, 9, 10, 11, 12, 13, 14, 15]. We then argue that the equilibrium structure is defined largely by the TCP algorithm, in the sense that the TCP algorithm alone determines the underlying constrained optimization problem for which the equilibrium of TCP/AQM solves. In the following sections, we show that the primal-dual models in [7, 8, 9, 10, 11, 12, 13, 14, 15] are not enough to compare and design existing and new AQMs, although it provides a unified framework for the TCP algorithm in terms of bandwidth allocation (i.e., fairness).

A network is modeled as a set L of links (scarce resources) with finite capacities $c = (c_l, l \in L)$. They are shared by a set S of sources indexed by s. Each source s uses a set $L_s \subseteq L$ of links. The sets L_s define an $L \times S$ routing matrix[1]

$$R_{ls} = \begin{cases} 1 & \text{if } l \in L_s \\ 0 & \text{otherwise} \end{cases}$$

Associated with each source s is its transmission rate $x_s(t)$ at time t, in packets/sec. Associated with each link l is a scalar congestion measure $p_l(t) \geq 0$ at time t. Following the notation of

[1]We abuse notation to use L and S to denote sets and their cardinalities.

[13], let

$$y_l(t) \quad = \quad \sum_s R_{ls} x_s(t - \tau_{ls}^f)$$

be the aggregate rate at link l at time t, where τ_{ls}^f is the (equilibrium) forward delays from sources s to link l, which are assumed constant. Let

$$q_s(t) \quad = \quad \sum_l R_{ls} p_l(t - \tau_{ls}^b)$$

be the end-to-end congestion measure for source s, where τ_{ls}^b is the (equilibrium) backward delays from links l to source s, assumed constant.

TCP is modeled by a function F_s that specifies how the source rate $x_s(t)$ is adjusted in response to end-to-end congestion measure $q_s(t)$:

$$\dot{x}_s(t) \quad = \quad F_s(x_s(t), q_s(t)). \tag{1}$$

F_s is coupled at bottleneck links by queue dynamics

$$\dot{b}_l(t) \quad = \quad E_l(y_l(t), c). \tag{2}$$

Different TCP algorithms are modeled by different F_s functions. AQM is modeled by functions (G_l, H_l) that describe how congestion measure $p_l(t)$ is updated, implicitly or explicitly, based on the aggregate rate $y_l(t)$ and possibly some internal variables $v_l(t)$ [15]:

$$\dot{p}_l(t) \quad = \quad G_l(y_l(t), v_l(t)) \tag{3}$$

$$\dot{v}_l(t) \quad = \quad H_l(y_l(t), v_l(t)). \tag{4}$$

In this section, as in [15], we refer to an AQM by G_l, *without explicit reference* to the internal variables $v_l(t)$ and their adaptation H_l.

In summary, a TCP/AQM protocol pair is modeled by a certain $(F, G) = (F_s, G_l, s \in S, l \in L)$.

It is shown in [7, 8, 9, 10, 11, 12, 18] that the equilibrium structure of (1–4) depends largely on the TCP functions F_s in (1) in the following sense. Consider an equilibrium (x, p) of (1–3). The fixed point of (1) defines an implicit relation between equilibrium rate x_s and end-to-end congestion measure q_s:

$$q_s \quad = \quad f_s(x_s) \ > \ 0$$

Define the utility function of each source s as

$$U_s(x_s) \quad = \quad \int f_s(x_s) dx_s, \qquad x_s \geq 0 \tag{5}$$

and consider the problem of maximizing aggregate utility:

$$\max_{x \geq 0} \qquad \sum_s U_s(x_s) \ \text{ subject to } Rx \leq c \tag{6}$$

The constraint says that, at each link l, the aggregate rate y_l does not exceed the capacity c_l. The key to understanding the equilibrium of (1–4) is to regard $x(t)$ as a primal variable, $p(t)$ as a dual variable, and $(F, G) = (F_s, G_l, s \in S, l \in L)$ as a distributed primal-dual algorithm carried out by sources and links over the Internet in the form of congestion control to solve the primal problem (6) and its Lagrangian dual. Various TCP/AQM protocols can be modeled as different distributed primal-dual algorithms (F, G, H) to solve the same global optimization problem (6), with different utility functions U_s given by (5).

Note that the utility function, and hence the underlying primal problem, depends solely on the TCP algorithm F_s. As long as the AQM functions G_l matches the aggregate rate y_l^* to link capacity c_l at every bottleneck link with $p_l^* > 0$, an equilibrium (x^*, p^*) will be primal-dual optimal (see [10]). This property is satisfied by all AQMs that stabilize queues, e.g., RED, REM, PI, and AVQ, etc. *Hence, we can interpret the design of TCP functions F_s as choosing an equilibrium point (e.g., bandwidth allocation and fairness), and the role of AQM functions G_l as stabilizing the equilibrium point of given TCP and queue dynamics.*

This view is taken by [19] and extended in [15, 16, 20]. It prompts the questions of how the stabilizing cost can be compared for a given TCP function F_s and queue dynamics E_l, what AQM G_l minimizes the given cost function, and how different AQM functions G_l can be compared.

The purpose of this paper is to propose a unified framework model within which these questions can be rigorously studied.

Existing AQMs including RED, REM/PI, and AVQ can be roughly summarized by

$$\text{RED:} \quad p^r(t) = H_2^r \hat{b}(t), \qquad \dot{\hat{b}}(t) = -P_1 \hat{b}(t) + P_1 b(t) \tag{7}$$

$$\text{AVQ:} \quad \dot{p}^a(t) = p(y(t-\tau),\ \tilde{b}(t-\tau)), \qquad \dot{\tilde{b}}(t) = \gamma_1 \left[y(t) - \gamma_2 c\right] \tag{8}$$

$$\text{REM/PI:} \quad \dot{p}^m(t) = H_1^m (b(t) - b^*) + H_2^m \dot{b}(t) \tag{9}$$

for some nonnegative constants H_2^r, H_2^a, H_1^m, H_2^m, $P_1 > 0$, $\gamma_1 < 0$, and $1 > \gamma_2 \geq 0$, where $\hat{b}$, $\tilde{b}$ and b are average-, virtual-, and real- queue lengths, respectively.

Here, we ignore the nonlinear function of the loss probability in RED. The average queue length $\hat{b}$ is called a low-pass filter in this paper. If we ignore the low-pass filter structure, then RED has the form $\dot{p}^r(t) = H_2^r \dot{b}(t) = H_2^r (y(t) - c)$. REM has PI-type control structure and exponential marking method for loss-probability. In this paper, we don't consider the exponential marking method in REM. Thus, RED without a low-pass filter, REM/PI, and AVQ are captured by the primal-dual models (1)-(4) in [7, 8, 9, 10, 11, 12, 13, 14, 15]. *However, the primal-dual models (F, G, H) in [7, 8, 9, 10, 11, 12, 13, 14, 15] do not capture very important structures as we argue in the following section.*

The basic idea is to treat TCP F_s as a dynamical system with congestion measure $p(t)$ as its control input as in [6]. The problem of optimal AQM design is to choose an input that stabilizes TCP and minimizes the linear quadratic cost function. In this paper, we study a simplified version of this problem, simplified in five regards. First, we consider a deterministic

TCP and queuing model, not a stochastic model. Second, we consider the linearized version of the TCP function F_i in (1), so the variables denote perturbations around an equilibrium and the cost measures the deviation from the equilibrium point. For example, a slower transient will incur a higher cost. Third, when we try to get stabilizing optimal AQMs in Section 5, we assume that we know the global information of the given networks and forward delays are zero ($\tau^f_{ls}(\cdot) = 0$). Fourth, we do not use virtual-queue dynamics in AVQ and in [21] and internal variables $v_l(t)$ such as low-pass filter in RED. Finally, as an exercise for analysis and design of AQM, we consider TCP Reno.

3 PD-type and PID-type AQM structures

Consider the simple case of a single link of capacity c shared by N TCP Reno sources, modeled by

$$\begin{aligned}
\dot{w}_s(t) &= \frac{w_s(t-\tau_s(t))}{d_s + b(t-\tau_s(t))/c}(1 - p(t-\tau^b(t)))\frac{1}{w_s(t)} \\
&\quad - \frac{w_s(t-\tau_s(t))}{d_s + b(t-\tau_s(t))/c} p(t-\tau^b(t))\frac{w_s(t)}{2} \\
\dot{b}(t) &= -c + \sum_{s=1}^{N} \frac{w_s(t-\tau^f_s(t))}{d_s + b(t-\tau^f_s(t))/c} \\
&= -c + y(t-\tau^f(t)),
\end{aligned} \tag{10}$$

where $w_s(t)$ is (expected) TCP window size, in packets, of source s at time t, τ_s is the round trip time, $p(t)$ is the loss probability at time t, $b(t)$ is the queue length at time t, $y(t)$ is aggregate rate, and c is the link capacity, in packets/sec. We define the source rate by $x_s(t) = w_s(t)/(d_s + b(t)/c)$, where d_s is the round trip propagation delay of source s. As in [19], we assume sources are identical $d_s = d$ and all have the common window $w_s(t) \equiv w(t)$; we assume delays take their equilibrium values and are constant, and forward delays are zero, $\tau^f_s(\cdot) = 0$, so that $\tau_s(t) = \tau^b_s(t) = \tau$. Let (w^*, b^*, p^*) be the equilibrium point. Then τ is related to b^* by $\tau = d + b^*/c$.

The first key step to proposing a new AQM structure is to convert the system (10) to the equivalent form

$$\begin{aligned}
\ddot{b}(t) &= \frac{N(\dot{b}(t-\tau)+c)}{(d_s + b(t)/c)^2(\dot{b}(t)+c)} - \frac{(\dot{b}(t)+c)\dot{b}(t)}{(d_s + b(t)/c)c} \\
&\quad - \left\{ \frac{N(\dot{b}(t-\tau)+c)}{(d_s + b(t)/c)^2(\dot{b}(t)+c)} + \frac{(\dot{b}(t-\tau)+c)(\dot{b}(t)+c)}{2N} \right\} p(t-\tau) \\
&= f(b(t), \dot{b}(t), \dot{b}(t-\tau), p(t-\tau)).
\end{aligned} \tag{11}$$

Note that minimizing a stabilizing cost for the given system (11) corresponds to utilizing the given network resources fully for the given TCP function and real-queue dynamics. In general, it is very difficult to find the best nonlinear function $p(t)$ that minimizes stabilizing cost for the

given nonlinear dynamical system. Thus, we consider the linearized version of the TCP function in this paper. For simplicity, we linearize the TCP function (11) only on $b(t)$, $\dot{b}(t)$, $\dot{b}(t-\tau)$ and $p(t-\tau)$, while we also need to linearize it on $b(\sigma)$ and $\dot{b}(\sigma)$ for $\sigma \in [t-\tau, t)$.

From (11), we derive the following state-space model of the linearized TCP and queue dynamics:

$$\dot{z}(t) \;=\; Az(t) + Bu(t-\tau), \tag{12}$$

where $z(0)$ and $\{u(\sigma), \sigma \in [-\tau, 0]\}$ are given,

$$\begin{aligned} z(t) &= \begin{bmatrix} \delta b(t) \\ \delta \dot{b}(t) \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 \\ A_1 & A_2 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ B_1 \end{bmatrix}, \qquad u(t) = \delta p(t) \\ \delta b(t) &= b(t) - b^*, \qquad \delta \dot{b}(t) = \dot{b}(t), \qquad \delta p(t) = p(t) - p^* \\ A_1 &= -\frac{2cN}{\tau(2N^2 + c^2\tau^2)}, \quad A_2 = -\frac{2cN\tau + 2N^2 + c^2\tau^2}{\tau(2N^2 + c^2\tau^2)}, \quad B_1 = -\frac{2N^2 + c^2\tau^2}{2\tau^2 N}. \end{aligned} \tag{13}$$

Refer to Appendix A for derivation of (12). Since $A_2^2 + 4A_1 > 0$ for system matrices of (16), we have $a_1 \neq a_2 \neq 0$, $e_1 < 0$ and $\hat{B}_1 < 0$ for $\tau > 0$. Note that $a_1 = a_2$ when $2N = \tau c$ even for $\tau > 0$ in the linearized model of [19]. Note that the pair $(A, \hat{B})$ is stabilizable.

From the above state-space model, we can naturally get a PD-type (Proportional-Derivative) state-feedback AQM

$$\delta p(t) \;=\; H\ z(t) \;=\; H_P \delta b(t) + H_D \delta \dot{b}(t) \tag{14}$$

if we use only the current dynamic information $b(t)$ and $\dot{b}(t)$ for $\delta p(t)$. Note that this PD-type structure (14) is not captured by the primal-dual models (1)-(4) in [7, 8, 9, 10, 11, 12, 13, 14, 15].

Remark 1 *It is interesting to see that if we ignore the delay term, the transfer function from $\delta b(t)$ to $\delta w(t)$ is equal to the lead-lag compensator with the form*

$$\frac{\delta w(s)}{\delta b(s)} \;=\; \frac{k_2 s + k_3}{s + k_1},$$

where k_1, k_2, and k_3 are some constants, and $\delta w(t)$ is a variation of $w(t)$ near the equilibrium point. Here, we emphasize that the primal-dual models in [7, 8, 9, 10, 11, 12, 13, 14, 15] and existing AQM algorithms in [4, 5, 6, 16, 17] did not capture the proposed PD-type control structures. We also emphasize that the proposed PD-type structure can also be implemented either at sources or at routers with congestion measures different from the buffer $\delta b(t)$ (for example, Vegas in [3] uses the round-trip delay as congestion measure).

For implementation, (14) can be rewritten as

$$p(t) \;=\; p^* \;+\; H_P(b(t) - b^*) + H_D \dot{b}(t).$$

If b^* is too small or too large, it is not easy to stabilize the system (10) with (14) since real-queue length cannot be negative and has maximum buffer size. For the same reason, p^* should not

be close to zero or to one in order not to make $p(t)$ saturated. b^* should not be large for the following two reasons. First, a large queuing delay, which comes from the large queue, makes the nominal-stable system unstable. Second, congestion avoidance phase at TCP does not control small packets. Thus, we can consider small packets noise or disturbance for the system (10). In order for small packets to go through networks without causing congestion, we need to make the queue-length small if possible. p^* can be large if ECN (Explicit Congestion Notification) is used to express (14), but it should be small if packet dropping is used to express (14).

If we use the above PD-type (or P-type) AQM structure (14) for the original nonlinear system (10), the tracking error $\delta b(t)$ goes to zero very slowly at the steady-state (i.e., $b(t)$ is close to b^*) since $\delta p(t)$ is almost zero when $\delta b(t)$ is close to zero. We overcome this problem in the following.

In order to make the steady-state tracking error go to zero fast, we apply integral control action that augments the original system by differentiating the control [22]. *The key step to applying the technique is to have another derivative of the system (10) as follows*:

$$\begin{aligned} \dddot{b}(t) &= \frac{\partial f}{\partial b(t)}\dot{b}(t) + \frac{\partial f}{\partial \dot{b}(t)}\ddot{b}(t) + \frac{\partial f}{\partial \dot{b}(t-\tau)}\ddot{b}(t-\tau) + \frac{\partial f}{\partial p(t-\tau)}\dot{p}(t-\tau) \\ &= g(b(t), \dot{b}(t), \ddot{b}(t), \dot{b}(t-\tau), \ddot{b}(t-\tau), p(t-\tau), \dot{p}(t-\tau)). \end{aligned} \tag{15}$$

From (15), we can derive the following third order linearized TCP model:

$$\dot{z}_e(t) = A_e z_e(t) + B_e \dot{u}(t-\tau), \tag{16}$$

where $z_e(0)$ and $\{\dot{u}(\sigma), \sigma \in [-\tau, 0]\}$ are given,

$$z_e(t) = \begin{bmatrix} z_0(t) \\ \dot{z}(t) \end{bmatrix}, \quad A_e = \begin{bmatrix} 0 & I_e \\ 0 & A \end{bmatrix}, \quad B_e = \begin{bmatrix} 0 \\ B \end{bmatrix}, \quad \delta\ddot{b}(t) = \ddot{b}(t), \quad \dot{u}(t) = \delta\dot{p}(t) = \dot{p}(t),$$

with $z(t)$, A and B in (13), $z_0(t) = I_e z(t)$, and $I_e = [1,\ 0]$. Refer to Appendix A for derivation of (16).

Note that the linearized model of (15) has the same A and B as those of the linearized model of (11), while they are different in general. Also note that the linearized model of (15) has only $\delta\dot{p}(\cdot)$ as controllers, while it has both $\delta p(\cdot)$ and $\delta\dot{p}(\cdot)$ in general. It is easy to check that the pair (A_e, B_e) is stabilizable if $B_1 \neq 0$.

From the above state-space model, we can naturally get a PID-type (Proportional-Integral-Derivative) state-feedback AQM

$$\delta\dot{p}(t) = H_e\ z_e(t) = H_I \delta b(t) + H_P \delta\dot{b}(t) + H_D \delta\ddot{b}(t), \tag{17}$$

if we consider only the current dynamic information $b(t)$, $\dot{b}(t)$ (or $y(t)$), and $\ddot{b}(t)$ (or $\dot{y}(t)$) for $\delta\dot{p}(t)$. This kind of control is called a memoryless control in the control literature.

The proposed AQM (17) is a unified mathematical framework for analysis and design of a memoryless AQM since it includes P-type and PI-type control structures of RED and REM/PI, respectively. In addition, the proposed PID-type AQM does not require the equilibrium point p^*

but requires the initial loss probability $p(t_0)$, while PD-type AQM needs p^*. In general, it is not easy to know p^* at each source or each router, although we can assume that $p(t_0)$ is zero. Thus, the proposed unified framework also has an advantage in implementation. Since we can handle P-type, PI-type, and PD-type AQMs as special cases of PID-type AQM, we mainly consider PID-type AQM from now on in this paper.

If we use a memoryless control, we have performance limit in the presence of large delays τ, i.e., we cannot also fully utilize the given network resources in the presence of large delays τ. We overcome this problem in the next section.

4 AQMs with a memory control structure

In order to compensate for large delays in congestion measure, we need a memory control that uses not only the current dynamic information but also the previous dynamic information for $\delta p(t)$ or $\delta \dot{p}(t)$. In order to derive a memory control explicitly for the delayed system, throughout the rest of this paper, we define

$$\begin{aligned}
e_1 &= e^{-a_1\tau} - e^{-a_2\tau}, \quad e_2 = a_1 e^{-a_1\tau} - a_2 e^{-a_2\tau} \\
e_3 &= a_2 e^{-a_1\tau} - a_1 e^{-a_2\tau} \\
e_4 &= \frac{-1}{e_1}\left[\frac{a_1}{a_2}e^{-a_1\tau} - \frac{a_2}{a_1}e^{-a_2\tau} + \frac{a_2^2 - a_1^2}{a_1 a_2}e^{-A_2\tau}\right] \\
e_5 &= \frac{1}{e_1}\left[\frac{1}{a_2}e^{-a_1\tau} - \frac{1}{a_1}e^{-a_2\tau} + \frac{a_2 - a_1}{a_1 a_2}e^{-A_2\tau}\right] \\
a_1 &= \frac{A_2 + \sqrt{A_2^2 + 4A_1}}{2}, \quad a_2 = \frac{A_2 - \sqrt{A_2^2 + 4A_1}}{2} \\
\hat{B}_1 &= \frac{B_1(a_2 - a_1)e^{-A_2\tau}}{e_1}.
\end{aligned} \tag{18}$$

The key to deriving an explicit memory control for the delayed system (12) is to transform the delayed system (12) to the equivalent nominal system

$$\dot{s}(t) = As(t) + \hat{B}u(t), \tag{19}$$

where

$$\begin{aligned}
s(t) &= [s_1,\ s_2]^T, \quad \hat{B} = [0,\ \hat{B}_1]^T \\
s_1(t) &= -\frac{e_2}{e_1}(\delta b(t) + u_{1\tau}(t)) + \delta\dot{b}(t) + u_{2\tau}(t) && (20) \\
s_2(t) &= A_1(\delta b(t) + u_{1\tau}(t)) + \frac{e_3}{e_1}(\delta\dot{b}(t) + u_{2\tau}(t)) && (21) \\
\begin{bmatrix} u_{1\tau}(t) \\ u_{2\tau}(t) \end{bmatrix} &= \frac{B_1}{a_1 - a_2}\int_{-\tau}^{0}\begin{bmatrix} e^{-(\sigma+\tau)a_1} - e^{-(\sigma+\tau)a_2} \\ a_1 e^{-(\sigma+\tau)a_1} - a_2 e^{-(\sigma+\tau)a_2} \end{bmatrix} u(\sigma + t)d\sigma. && (22)
\end{aligned}$$

Refer to Appendix B for derivation of (19). Note that the pair $(A, \hat{B})$ is stabilizable and the closed-loop system of (19) is asymptotically stable if and only if the transformed system (12) is asymptotically stable. If $\tau = 0$, then $s(t) = z(t)$ and $\hat{B} = B$.

Thus, by adding the memory control structure $u_{i\tau}(t)$, we can remove the delay in congestion measure from the original linearized system. From the above state-space model, we can naturally get a PD-type (Proportional-Derivative) memory AQM

$$\delta p(t) \;=\; H_P^\tau(\delta b(t) + u_{1\tau}(t)) \;+\; H_D^\tau(\delta \dot{b}(t) + u_{2\tau}(t)). \tag{23}$$

Similarly, for derivation of an explicit memory control for the delayed system (16), we transform the delayed system (16) to the equivalent nominal system

$$\dot{s}_e(t) \;=\; A_e s_e(t) + \hat{B}_e \dot{u}(t), \tag{24}$$

where

$$\begin{aligned}
s_e(t) &= [s_1,\ s_2,\ s_3]^T,\quad \hat{B}_e \;=\; [0,\ \hat{B}^T]^T \\
s_1(t) &= \frac{(a_2-a_1)e^{-A_2\tau}}{e_1}(\delta b(t) + u_{1\tau}(t)) + e_4(\delta \dot{b}(t) + u_{2\tau}(t)) + e_5(\delta \ddot{b}(t) + u_{3\tau}(t)) && (25)\\
s_2(t) &= -\frac{e_2}{e_1}(\delta \dot{b}(t) + u_{2\tau}(t)) + \delta \ddot{b}(t) + u_{3\tau}(t) && (26)\\
s_3(t) &= A_1(\delta \dot{b}(t) + u_{2\tau}(t)) + \frac{e_3}{e_1}(\delta \ddot{b}(t) + u_{3\tau}(t)) && (27)\\
\begin{bmatrix} u_{1\tau}(t) \\ u_{2\tau}(t) \\ u_{3\tau}(t) \end{bmatrix} &= \frac{B_1}{a_1-a_2}\int_{-\tau}^{0} \begin{bmatrix} \frac{a_1-a_2}{a_1 a_2} + \frac{e^{-(\sigma+\tau)a_1}}{a_1} - \frac{e^{-(\sigma+\tau)a_2}}{a_2} \\ e^{-(\sigma+\tau)a_1} - e^{-(\sigma+\tau)a_2} \\ a_1 e^{-(\sigma+\tau)a_1} - a_2 e^{-(\sigma+\tau)a_2} \end{bmatrix} \dot{u}(\sigma+t)d\sigma. && (28)
\end{aligned}$$

Refer to Appendix B for derivation of (19).

From the above state-space model, we can naturally get a PID-type (Proportional-Integral-Derivative) memory state-feedback AQM

$$\delta \dot{p}(t) \;=\; +\, H_I^\tau\big(\delta b(t) + u_{1\tau}(t)\big) \;+\; H_P^\tau\big(\delta \dot{b}(t) + u_{2\tau}(t)\big) \;+\; H_D^\tau\big(\delta \ddot{b}(t) + u_{3\tau}(t)\big). \tag{29}$$

The proposed AQM (29) is a unified mathematical framework for analysis and design of a memory AQM since it includes P-type and PI-type control structures.

Also note that the closed-loop system of (24) is also asymptotically stable if and only if the transformed system (16) is asymptotically stable. If $\tau = 0$, then $s_e(t) = z_e(t)$ and $\hat{B}_e = B_e$. Also note that the pair (A_e, B_e) is stabilizable (or controllable) if the pair (A, B) is stabilizable (or controllable).

5 Stabilizing Optimal AQM: RHA

In this subsection, we derive stabilizing optimal AQMs for the linearized systems (19) and (24).

5.1 PD-type Stabilizing Optimal AQM: RHA

As a performance measure for (19), we consider the following optimization problems:

$$\min_{u(\cdot)} \;\; J(s(t), u(\cdot)) \;=\; \int_t^{t+\infty} \big(s^T(\sigma)Qs(\sigma) + u^2(\sigma)\big)\, d\sigma, \tag{30}$$

for the linearized system (19), where $Q = Q^T \geq 0$ and the pair $(A, Q^{\frac{1}{2}})$ is observable. Even if Q is negative, we can get a stabilizing control if the system is stabilizable. However, for simplicity, we don't consider a negative Q in this paper.

Note that problem (30) with $u(\sigma) = -\hat{B}^T K s(\sigma)$ for all σ is equal to

$$\min_{u(\cdot)} \quad J(s(t), u(\cdot)) \;=\; \int_t^{t+T} \left(s^T(\sigma) Q s(\sigma) + u^2(\sigma)\right) d\sigma + s^T(t+T) K s(t+T),$$

where $0 = A^T K + KA - K\hat{B}\hat{B}^T K + Q$ for any $T > 0$ as shown in [23].

We then propose a stabilizing optimal AQM (Receding Horizon AQM: RHA) for the linearized system (19), which is obtained from the following procedure.

(a) At the present time t, the solutions for the optimal closed-loop control, $u(\tau)$ are obtained for $\forall\ \tau \in [t, t+\infty)$, which minimizes Eq. (30).

(b) Among these controls, only the first control $u(\tau)_{\tau=t}$ is used.

(c) At the next t, the procedures (a) and (b) are repeated.

The above procedure is well known as the receding horizon control scheme in control area. The receding horizon control has received much attention in both academia [24, 25, 26, 27], and industry fields [28, 29] because it has many advantages such as simple computation, good tracking performance, I/O constraint handling, and extension to nonlinear systems, compared with the steady-state Linear Quadratic (LQ) control. The RHC seems to be equal to the steady-state LQ control for unconstrained systems. Strictly speaking, however, the RHC is a closed-loop control strategy, while the LQ control is a open-loop control strategy.

Throughout the rest of this section, for simplicity, we define

$$F_1 \;=\; A_1^2 + \hat{B}_1^2 Q_1, \qquad F_2 \;=\; A_2^2 + \hat{B}_1^2 Q_2, \qquad a_3 \;=\; \frac{1}{a_1 - a_2} \log_e \frac{a_2}{a_1} \tag{31}$$

$$K \;=\; \begin{bmatrix} K_1 & K_2 \\ K_2 & K_3 \end{bmatrix}, \qquad Q \;=\; \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix}. \tag{32}$$

Proposition 1 *If we solve the problem (30), we can get a stabilizing RHA*

$$u^*(t) \;=\; -\hat{B}_1 \left[K_2\ s_1(t) + K_3\ s_2(t)\right] \tag{33}$$

and the optimal cost of (30)

$$J^* \;=\; s^T(0) K s(0), \tag{34}$$

where the control gains satisfy that $K_1 > 0$, $K_3 > 0$, and

$$K_1 \;=\; \frac{\sqrt{(F_2 + 2A_1 + 2)F_1} - A_1 A_2}{\hat{B}_1^2}, \qquad K_2 \;=\; \frac{A_1 + \sqrt{F_1}}{\hat{B}_1^2}, \qquad K_3 \;=\; \frac{A_2 + \sqrt{F_2 + 2A_1 + 2\sqrt{F_1}}}{\hat{B}_1^2}.$$

If the state and input constraints are not violated, then $s_1(t)$ is given by

$$s_1(t) \;=\; \frac{1}{\lambda_2 - \lambda_1}\left[(\lambda_2 s_1(0) - s_2(0))e^{\lambda_1 t} - (\lambda_1 s_1(0) - s_2(0))e^{\lambda_2 t}\right] \quad \textit{when } \lambda_1 \neq \lambda_2 \tag{35}$$

$$\;=\; [s_1(0) + t(s_2(0) - s_1(0)\lambda_1)]e^{\lambda_1 t} \quad \textit{when } \lambda_1 = \lambda_2$$

$$\lambda_1,\ \ \lambda_2 \;=\; \frac{-\sqrt{F_2 + 2(A_1 + \sqrt{F_1})} \pm \sqrt{F_2 + 2(A_1 - \sqrt{F_1})}}{2}. \tag{36}$$

Proof: The optimal control that minimizes (30) and the resulting optimal cost are given by

$$u^*(t) \;=\; -\hat{B}^T K s(t), \qquad J^*(s(t)) \;=\; s^T(t) K s(t) \tag{37}$$

where K satisfies $0 = A^T K + KA + Q - K\hat{B}\hat{B}^T K$ [30]. Note that K is a symmetric positive definite matrix and the resulting closed-loop system is asymptotically stable since the pairs $(A, \hat{B})$ and $(A, Q^{\frac{1}{2}})$ are controllable and observable, respectively [31], [23]. For observability of the system (12), Q_1 should be positive while Q_2 and Q_3 could be zero.

By solving (37), we can get (33) and (34). The closed-loop system with (46) is given by

$$\dot{s}(t) \;=\; \begin{bmatrix} 0 & 1 \\ A_1 - \hat{B}_1^2 K_2 & A_2 - \hat{B}_1^2 K_3 \end{bmatrix} s(t).$$

From this, we can get (35)-(36). □

Note that real parts of λ_1 and λ_2 of the closed-loop system with the proposed RHA are always negative, although the real part of λ_1 can be near zero. In order that λ_1 is not located near zero, $\sqrt{F_1}$ should not be too small. In order that $\delta s_1(t)$ does not oscillate, we should have $F_2 + 2A_1 \;\geq\; 2\sqrt{F_1}$. Thus, an easy way to design Q_1 and Q_2 is to make λ_1 equal to λ_2 and increase the value of λ_1. It can be done by setting $Q_1 = \frac{\alpha^4 - A_1^2}{\hat{B}_1^2}$, $Q_2 = \frac{2\sqrt{A_1^2 + \hat{B}_1^2 Q_1} - A_2^2 - 2A_1}{\hat{B}_1^2}$. Then, $F_1 = \alpha^4$, $F_2 = 2\alpha^2 - 2A_1$.

Proposition 1 implies that the solution of the problem (30) is a stabilizing AQM algorithm, specified by (K_2, K_3).

Proposition 2 *Given a stabilizing AQM $u(t) = [H_1,\ H_2]s(t)$ that satisfies $A_1 + \hat{B}_1 H_1 < 0$ and $A_2 + \hat{B}_1 H_2 < 0$, it solves the problem (30) with weights:*

$$Q_1 \;=\; \frac{H_1^2 \hat{B}_1 + 2H_1 A_1}{\hat{B}_1}, \qquad Q_2 \;=\; \frac{H_2^2 \hat{B}_1 + 2H_2 A_2 + 2H_1}{\hat{B}_1}. \tag{38}$$

Then, K_1, K_2, K_3, λ_1, and λ_2 are given by

$$K_2 \;=\; -\frac{H_1}{\hat{B}_1}, \quad K_3 \;=\; -\frac{H_2}{\hat{B}_1}, \quad K_1 \;=\; \frac{\hat{B}_1 H_1 H_2 + A_1 H_2 + A_2 H_1}{\hat{B}_1} \tag{39}$$

$$\lambda_1,\ \ \lambda_2 \;=\; \frac{(A_2 + \hat{B}_1 H_2) \pm \sqrt{(A_2 + \hat{B}_1 H_2)^2 + 4(A_1 + \hat{B}_1 H_1)}}{2}. \tag{40}$$

Proof: It can be easily proved from Proposition 1. □

In order for an AQM $u(t) = [H_1\ H_2]s(t)$ to be a stabilizing optimal control when the system is controllable and observable for the system (12), it should satisfy $H_1 > -\frac{A_1}{\hat{B}_1}$ and $H_2 > 0$. As shown in Proposition 2, if $H_2 = 0$, then we have $\lambda_1 + \lambda_2 = A_2 + \hat{B}_1 H_2 \leq A_2$ where the last inequality follows from that $A_2 < 0$, $\hat{B}_1 < 0$ and $H_2 \geq 0$. Since all eigenvalues should have negative real parts for the closed-loop stability, the above inequality means that the sum of the real parts of the eigenvalues is less negative when $H_2 = 0$ than when $H_2 > 0$. This suggests that we need to add D-control in order to make the dynamics move faster. The decaying rate our PD-type RHA makes the states go to zero faster than P-type RED ($H_2 = 0$).

Proposition 3 *Given the eigenvalues λ_1 and λ_2 of the closed-loop system (19) with (33), where real parts of λ_1 and λ_2 are negative, it solves the problem (30) with weights:*

$$Q_1 = \frac{(\lambda_1\lambda_2)^2 - A_1^2}{\hat{B}_1^2}, \quad Q_2 = \frac{\lambda_1^2 + \lambda_2^2 - A_2^2 - 2A_1}{\hat{B}_1^2}. \tag{41}$$

When real parts of λ_1 and λ_2 are negative, K_1, K_2, and K_3 can be rewritten as

$$K_2 = \frac{A_1 + \lambda_1\lambda_2}{\hat{B}_1^2}, \quad K_3 = \frac{A_2 - (\lambda_1 + \lambda_2)}{\hat{B}_1^2}, \quad K_1 = \frac{-\lambda_1\lambda_2(\lambda_1 + \lambda_2) - A_1A_2}{\hat{B}_1^2}. \tag{42}$$

Remark 2 *Actually, $u(t)$ is constrained as $-p^* \le u(t)(= \delta u(t)) \le 1 - p^*$. Thus, it is necessary to check the extremum of $u^*(t)$ in (33). Define*

$$\begin{aligned}
t_a^* &= \frac{1}{\lambda_1 - \lambda_2} log_e \frac{-\lambda_2\beta_2}{\lambda_1\beta_1}, \quad t_b^* = \frac{\beta_3}{s_2(0) - \lambda_1 s_1(0)} \\
\beta_1 &= (A_1 + A_2\lambda_1 - \lambda_1^2)(\lambda_2 s_1(0) - s_2(0)), \quad \beta_2 = (A_1 + A_2\lambda_2 - \lambda_2^2)(s_2(0) - \lambda_1 s_1(0)) \\
\beta_3 &= \frac{\lambda_1^2(A_2 - 2\lambda_1)s_1(0) - (A_1 + 2A_2\lambda_1 - 3\lambda_1^2)s_2(0)}{\lambda_1[(A_2 - 2\lambda_1)\lambda_1 + A_1 + \lambda_1^2]}.
\end{aligned}$$

From (33) and (35), if $\lambda_1 \neq \lambda_2$, the extremum of $u^*(t)$ is given by

$$\begin{aligned}
u^*(t_a^*) &= \frac{1}{\hat{B}_1(\lambda_1 - \lambda_2)} \left\{ \beta_1 (\frac{-\lambda_2\beta_2}{\lambda_1\beta_1})^{\frac{\lambda_1}{\lambda_1 - \lambda_2}} + \beta_2 (\frac{-\lambda_2\beta_2}{\lambda_1\beta_1})^{\frac{\lambda_2}{\lambda_1 - \lambda_2}} \right\} \quad when \;\; t_a^* > 0 \\
u^*(0) &= \frac{-1}{\hat{B}_1}[(A_1 + \lambda_1\lambda_2)s_1(0) + (A_2 - \lambda_1 - \lambda_2)s_2(0)] \quad when \;\; t_a^* \le 0.
\end{aligned}$$

If $\lambda_1 = \lambda_2$, the extremum of $u^*(t)$ is given by

$$\begin{aligned}
u^*(t_b^*) &= -\left\{ \frac{A_1 + \lambda_1^2}{\hat{B}_1}(s_1(0) + t\beta_3) + \frac{A_2 - 2\lambda_1}{\hat{B}_1}(s_2(0) + \lambda_1 t\beta_3) \right\} e^{\lambda_1 t^*} \quad when \;\; t_b^* > 0 \\
u^*(0) &= -\frac{1}{\hat{B}_1}[(A_1 + \lambda_1^2)s_1(0) + (A_2 - 2\lambda_1)s_2(0)] \quad when \;\; t_b^* \le 0.
\end{aligned}$$

For implementation, we need to rewrite the RHA (33) as (14) for a memoryless AQM and (23) for a memory AQM, where

$$\begin{aligned}
K_P &= -\frac{(A_1 + \sqrt{A_1^2 + B_1^2 Q_1})}{B_1}, \quad K_D = -\frac{(A_2 + \sqrt{A_2^2 + B_1^2 Q_2 + 2A_1 + 2\sqrt{A_1^2 + B_1^2 Q_1}})}{B_1} \\
H_P^\tau &= \frac{A_1 e_3 - e_2\sqrt{F_1} + A_1 e_1 \sqrt{F_2 + 2A_1 + 2\sqrt{F_1}}}{(a_1 - a_2)B_1} \\
H_D^\tau &= \frac{e_1(A_1 + \sqrt{F_1}) + e_3(A_2 + \sqrt{F_2 + 2A_1 + 2\sqrt{F_1}})}{(a_1 - a_2)B_1}.
\end{aligned}$$

Next, we analyze the effect of setting H_1^τ (or H_2^τ) the proposed RHA (23) to zero.

Remark 3 *First, we assume that $H_2^\tau = 0$, i.e., $K_2 = -K_3\frac{e_3}{e_1}$. Then, (23), λ_1, and λ_2 can be converted to*

$$\begin{aligned}
u^*(t) &= \frac{(a_2 - a_1)^3 B_1 e^{-2A_2\tau}}{e_1^3} K_3(\delta b(t) + u_{1\tau}(t)) \\
\lambda_1 + \lambda_2 &= A_2 - \hat{B}_1^2 K_3, \quad \lambda_1\lambda_2 = -(A_1 + \hat{B}_1^2 K_3 \frac{e_3}{e_1}).
\end{aligned}$$

Thus, $\lambda_1\lambda_2 = -A_1 + \frac{e_3}{e_1}(-A_2 + \lambda_1 + \lambda_2)$. *Since* $e_3 \le 0$ *for* $\tau \le a_3$ *and* $e_1 < 0$, *we have very conservative* λ_1 *and* λ_2 *for the closed-loop stability. When* $\tau = a_3$, $\lambda_1\lambda_2 = -A_1$.

Second, assume that $H_1^\tau = 0$, *i.e,* $K_2 = \frac{e_1}{e_2}A_1K_3$. *Then, (23),* λ_1, *and* λ_2 *can be converted to*

$$\begin{aligned} u^*(t) &= \frac{B_1(a_2-a_1)^3}{e_1^2 e_2} e^{-2A_2\tau} K_3(\delta\dot{b}(t) + u_{2\tau}(t)) \\ \lambda_1 + \lambda_2 &= A_2 - \hat{B}_1^2 K_3, \qquad \lambda_1\lambda_2 = -A_1 + A_1\hat{B}_1^2\frac{e_1}{e_2}K_3. \end{aligned}$$

Thus, $\lambda_1\lambda_2 = A_1\frac{e_1}{e_2}(\lambda_1 + \lambda_2 - A_2) - A_1$. *Since* $e_2 > 0$ *and* $e_1 < 0$, *we have very conservative* λ_1 *and* λ_2 *for the closed-loop stability.*

In the following, we propose another stabilizing optimal AQM (RHA) by applying integral action technique to this section in order to make the steady-state tracking error between the queue length and the target queue length go to zero fast.

5.2 PID-type Stabilizing Optimal AQM: RHA

As a performance measure for (24), we consider the following optimization problems:

$$\min_{\dot{u}(\cdot)} \; J(s_e(t), \dot{u}(\cdot)) = \int_t^{t+\infty} \left(s_e^T(\sigma)Qs_e(\sigma) + \dot{u}^2(\sigma)\right) d\sigma \tag{43}$$

for the linearized system (24), where $Q = Q^T \ge 0$ and the pair $(A_e, Q^{\frac{1}{2}})$ is observable.

For simplicity, throughout the rest of this section, we also define

$$K_e = \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{12} & K_{22} & K_{23} \\ K_{13} & K_{23} & K_{33} \end{bmatrix}, \quad Q = \begin{bmatrix} Q_1 & 0 & 0 \\ 0 & Q_2 & 0 \\ 0 & 0 & Q_3 \end{bmatrix} \tag{44}$$

$$\hat{\lambda}_1 = \lambda_1 + \lambda_2 + \lambda_3, \quad \hat{\lambda}_2 = \lambda_1\lambda_2 + \lambda_2\lambda_3 + \lambda_1\lambda_3, \quad \hat{\lambda}_3 = \lambda_1\lambda_2\lambda_3. \tag{45}$$

Proposition 4 *If we solve the problem (43), then we can get a stabilizing optimal AQM (RHA)*

$$\dot{u}^*(t) = -\hat{B}_1\left[K_{13}\, s_1(t) + K_{23}\, s_2(t) + K_{33}\, s_3(t)\right] \tag{46}$$

and the optimal cost of (43)

$$J^*(s_e(0)) = s_e^T(0)K_e s_e(0), \tag{47}$$

where the control gains satisfy that $K_{13} > 0$, $K_{23} > \frac{A_1}{\hat{B}_1^2}$, $K_{33} > 0$,

$$\begin{aligned} K_{11} &= (A_1 - \hat{B}_1^2K_{23})\frac{\sqrt{Q_1}}{\hat{B}_1}, \quad K_{12} = (A_1 - \hat{B}_1^2K_{33})\frac{\sqrt{Q_1}}{\hat{B}_1}, \quad K_{13} = -\frac{\sqrt{Q_1}}{\hat{B}_1} \\ K_{22} &= \frac{\sqrt{Q_1}}{\hat{B}_1} - A_1K_{33} - (A_2 - \hat{B}_1^2K_{33})K_{23}, \quad K_{23} = \frac{-2A_2K_{33} + \hat{B}_1^2K_{33}^2 - Q_3}{2}, \end{aligned} \tag{48}$$

and K_{33} is the positive solution of the following fourth order polynomial, that makes K_{23} greater than $\frac{A_1}{\hat{B}_1^2}$ and makes K_{22} positive:

$$-\hat{B}_1^7 K_{33}^4 + 4A_2\hat{B}_1^5 K_{33}^3 + (4A_1\hat{B}_1^3 - 4A_2^2\hat{B}_1^3 + 2\hat{B}_1^5 Q_3)K_{33}^2 + (-8\hat{B}_1^2\sqrt{Q_1} - 8A_1A_2\hat{B}_1 \\ -4A_2\hat{B}_1^3 Q_3)K_{33} + 8A_2\sqrt{Q_1} - 4A_1\hat{B}_1 Q_3 + 4\hat{B}_1 Q_2 - \hat{B}_1^3 Q_3^2 \;=\; 0. \tag{49}$$

If the state and input constraints are not violated, then $s_1(t)$ is given by

$$\begin{aligned} s_1(t) &= b_{11}e^{\lambda_1 t} + b_{12}e^{\lambda_2 t} + b_{13}e^{\lambda_3 t} \quad when \quad \lambda_1 \neq \lambda_2 \neq \lambda_3 && (50)\\ &= b_{21}e^{\lambda_1 t} + b_{22}e^{\lambda_2 t} + b_{23}te^{\lambda_2 t} \quad when \quad \lambda_1 \neq \lambda_2 = \lambda_3 && (51)\\ &= b_{31}e^{\lambda_1 t} + b_{32}te^{\lambda_1 t} + b_{33}t^2e^{\lambda_1 t} \quad when \quad \lambda_1 = \lambda_2 = \lambda_3, && (52)\end{aligned}$$

where

$$\begin{aligned} b_{11} &= \frac{\lambda_2\lambda_3 s_1(0) - (\lambda_2+\lambda_3)s_2(0) + s_3(0)}{(\lambda_1-\lambda_3)(\lambda_1-\lambda_2)}\\ b_{12} &= \frac{\lambda_1\lambda_3 s_1(0) - (\lambda_1+\lambda_3)s_2(0) + s_3(0)}{(\lambda_2-\lambda_1)(\lambda_2-\lambda_3)}\\ b_{13} &= \frac{\lambda_1\lambda_2 s_1(0) - (\lambda_1+\lambda_2)s_2(0) + s_3(0)}{(\lambda_3-\lambda_1)(\lambda_3-\lambda_2)} && (53)\\ b_{21} &= b_{11}|_{\lambda_2=\lambda_3}, \quad b_{22} = \frac{(\lambda_1^2 - 2\lambda_1\lambda_2)s_1(0) + 2\lambda_2 s_2(0) - s_3(0)}{(\lambda_2-\lambda_1)^2}\\ b_{23} &= \frac{\lambda_1\lambda_2 s_1(0) - (\lambda_1+\lambda_2)s_2(0) + s_3(0)}{(\lambda_2-\lambda_1)} && (54)\\ b_{31} &= s_1(0), \quad b_{32} = s_2(0) - \lambda_1 s_1(0), \quad b_{33} = \frac{\lambda_1^2 s_1(0) - 2\lambda_1 s_2(0) + s_3(0)}{2} && (55)\end{aligned}$$

and

$$\hat{\lambda}_1 = A_2 - \hat{B}_1^2 K_{33}, \quad \hat{\lambda}_2 = -A_1 + \hat{B}_1^2 K_{23}, \quad \hat{\lambda}_3 = \hat{B}_1\sqrt{Q_1}. \tag{56}$$

Proof: Refer to the proof of Proposition 4. □

Note that this paper starts from PD-type control and then derives PID-type control by applying integral control action in this section, while REM/PI start from P-type control and then add I-type control [17], [6] in order to separate the congestion measure from the performance.

We should design Q_1, Q_2, and Q_3 so that none of λ_1, λ_2, and λ_3 is near zero, and $s_1(t)$ does not oscillate. An easy way is to make λ_1, λ_2, and λ_3 equal, and increase the value of λ_1. It can be done by setting $Q_1 = (\frac{\alpha^3}{\hat{B}_1})^2$, $Q_2 = \frac{-A_1^2+3\alpha^4}{\hat{B}_1^2}$, $Q_3 = \frac{-A_2^2-2A_1+3\alpha^2}{\hat{B}_1^2}$, and increasing the value of α. Since $\lambda_1 = \lambda_2 = \lambda_3 = \alpha$, α decides the decaying rate of the closed-loop system.

Proposition 4 implies that the solution of the problem (43) is an AQM algorithm, specified by (K_{13}, K_{23}, K_{33}). Conversely, given any AQM of this structure, it solves the problem (43) with appropriate weights Q_i, as the next result says. It can be easily proved from Proposition 4.

Proposition 5 *Given a stabilizing AQM $\dot{u}(t) = [H_1\ H_2\ H_3]s_e(t)$, it solves the problem (43) with weights*

$$Q_1 = H_1^2, \quad Q_2 = H_2^2 - 2\frac{A_2H_1 + \hat{B}_1H_1H_3 - A_1H_2}{\hat{B}_1}, \quad Q_3 = H_3^2 + 2\frac{A_2H_3 + H_2}{\hat{B}_1}. \tag{57}$$

Then, K_{11}, K_{12}, K_{22}, K_{13}, K_{23}, and K_{33} are given by

$$\begin{aligned} K_{11} &= \frac{(A_1 + \hat{B}_1H_2)H_1}{\hat{B}_1}, \quad K_{12} = \frac{(A_2 + \hat{B}_1H_3)H_1}{\hat{B}_1}, \quad K_{22} = H_2H_3 + \frac{A_1H_3 + A_2H_2 + H_1}{\hat{B}_1} \\ K_{13} &= -\frac{H_1}{\hat{B}_1}, \quad K_{23} = -\frac{H_2}{\hat{B}_1}, \quad K_{33} = -\frac{H_3}{\hat{B}_1} \end{aligned} \tag{58}$$

and $\hat{\lambda}_1$, $\hat{\lambda}_2$, $\hat{\lambda}_3$ are given by

$$\hat{\lambda}_1 = A_2 + \hat{B}_1H_3, \qquad \hat{\lambda}_2 = -(A_1 + \hat{B}_1H_2), \qquad \hat{\lambda}_3 = \hat{B}_1H_1. \tag{59}$$

From (59), we can see decaying rate of the closed-loop system for the given H_1, H_2, and H_3. In order for an AQM $\dot{u}(t) = [H_1\ H_2\ H_3]s_e(t)$ to be a stabilizing optimal control when the system is controllable and observable for the system (16), it should satisfy $H_1 > 0$, $H_2 > -\frac{A_1}{\hat{B}_1}$, and $H_3 > 0$. Note that some stabilizing $\dot{u}(t) = [H_1\ H_2\ H_3]s(t)$ may not be expressed with diagQ_i for $i = 1, 2, 3$.

Proposition 6 *Given the eigenvalues λ_1, λ_2, and λ_3 of the closed-loop system (24) with (46), where real parts of λ_1, λ_2, and λ_3 are negative, $\dot{u}(t)$ solves the problem (43) with weights*

$$Q_1 = \frac{\hat{\lambda}_3^2}{\hat{B}_1^2}, \quad Q_2 = \frac{-A_1^2 + \hat{\lambda}_2^2 - 2\hat{\lambda}_1\hat{\lambda}_3}{\hat{B}_1^2}, \quad Q_3 = \frac{-A_2^2 - 2A_1 + \hat{\lambda}_1^2 - 2\hat{\lambda}_2}{\hat{B}_1^2}. \tag{60}$$

Moreover, K_{13}, K_{23}, K_{33}, K_{11}, K_{12}, and K_{22} are given by

$$\begin{aligned} K_{11} &= -\frac{\hat{\lambda}_2\hat{\lambda}_3}{\hat{B}_1^2}, \quad K_{12} = \frac{\hat{\lambda}_1\hat{\lambda}_3}{\hat{B}_1^2}, \quad K_{22} = \frac{-A_1A_2 - \hat{\lambda}_1\hat{\lambda}_2 + \hat{\lambda}_3}{\hat{B}_1^2} \\ K_{13} &= -\frac{\hat{\lambda}_3}{\hat{B}_1^2}, \quad K_{23} = \frac{A_1 + \hat{\lambda}_2}{\hat{B}_1^2}, \quad K_{33} = \frac{A_2 - \hat{\lambda}_1}{\hat{B}_1^2}. \end{aligned} \tag{61}$$

For implementation, we rewrite RHA (46) as (17) for a memoryless AQM and (29) for a memory AQM, where

$$\begin{aligned} K_I &= -B_1K_{13}, \qquad K_P = -B_1K_{23}, \qquad K_D = -B_1K_{33} \\ H_I^\tau &= -\hat{B}_1K_{13}\frac{(a_2 - a_1)e^{-A_2\tau}}{e_1} \\ H_P^\tau &= -\hat{B}_1[K_{13}e_4 - K_{23}\frac{e_2}{e_1} + K_{33}A_1] \\ H_D^\tau &= -\hat{B}_1[K_{13}e_5 + K_{23} + K_{33}\frac{e_3}{e_1}]. \end{aligned}$$

It can be derived from (25)-(28).

From (50)-(55), the input (46) can also be rewritten as

$$\begin{aligned}
\dot{u}^*(t) &= -\frac{1}{\hat{B}_1}\{b_{11}[A_1\lambda_1 + (A_2-\lambda_1)\lambda_1^2]e^{\lambda_1 t} + b_{12}[A_1\lambda_2 + (A_2-\lambda_2)\lambda_2^2]e^{\lambda_2 t} \\
&\quad + b_{13}[A_1\lambda_3 + (A_2-\lambda_3)\lambda_3^2]e^{\lambda_3 t}\} \quad \text{when} \quad \lambda_1 \neq \lambda_2 \neq \lambda_3 \qquad (62) \\
&= -\frac{1}{\hat{B}_1}\{b_{21}[A_1\lambda_1 + (A_2-\lambda_1)\lambda_1^2]e^{\lambda_1 t} + [b_{22}(A_1\lambda_2 + (A_2-\lambda_2)\lambda_2^2) + b_{23}(A_1 + \\
&\quad 2A_2\lambda_2 - 3\lambda_2^2) + b_{23}t(A_1\lambda_2 + (A_2-\lambda_2)\lambda_2^2)]e^{\lambda_2 t}\} \quad \text{when} \quad \lambda_1 \neq \lambda_2 = \lambda_3 \qquad (63) \\
&= -\frac{1}{\hat{B}_1}\{b_{31}(A_1\lambda_1 + (A_1-\lambda_1)\lambda_1^2) + b_{32}(A_1 + 2A_1\lambda_1 - 3\lambda_1^2) + 2b_{33}(A_1 - 3\lambda_1) \\
&\quad + t[b_{32}(A_1\lambda_1 + (A_1-\lambda_1)\lambda_2^2) + 2b_{33}(A_1 + 2A_1\lambda_1 - 3\lambda_1^2)] \\
&\quad + b_{33}t^2[A_1\lambda_1 + (A_1-\lambda_1)\lambda_1^2]\}e^{\lambda_1 t} \quad \text{when} \quad \lambda_1 = \lambda_2 = \lambda_3. \qquad (64)
\end{aligned}$$

5.3 Approximation

We now interpret AQMs including a simplified RED without a low-pass filter and saturations functions, REM, PI, and a simplified AVQ without an adaptive virtual-queue dynamics as various approximations of RHA. For easy comparison, we assume that $\tau = 0$ (i.e., $s_e(t) = z_e(t)$, $\hat{B}_e = B_e$) for the linearized model. Here, we emphasize again that the primal-dual algorithms do not capture the proposed PD-type and PID-type structures in this paper.

The models we use for these schemes are highly simplified and ignore many important characteristics. For RED, we make two simplifying assumptions. First, we remove averaging and assume the marking probability depends on the instantaneous queue. Second, we assume the marking probability is $p(t) = \rho_1(b(t) - \underline{b})$. For AVQ, we ignore an adaptive virtual-queue dynamics that is an essential structure in AVQ, and use the linearized model in [16]. We emphasize that the goal is to give a general structure of AQM as a performance limit that sheds light on the behavior of practical AQMs.

The linear models motivated by these AQMs are:

$$\begin{aligned}
\text{simplified RED/AVQ:} \quad \delta\dot{p}^r(t) &= H_2^r\delta\dot{b}(t) \\
\text{REM/PI:} \quad \delta\dot{p}^m(t) &= H_1^m\delta b(t) + H_2^m\delta\dot{b}(t)
\end{aligned}$$

for some nonnegative constants $H_2^r, H_3^a, H_1^m, H_2^m$. The linear models of RED, REM and PI are caricatures of the models in the original papers [4, 17, 6, 16].

By Proposition 4, the stabilizing optimal AQM has strictly positive gain $K_{33} > 0$. Since this condition is satisfied by none of RED, REM, PI and AVQ, none of them can be made optimal, in the sense of minimizing (43), by tuning its parameters. Moreover, their structure implies a limitation to rate of convergence to equilibrium.[2]

Specifically, a simplified RED and AVQ have $H_1^r = 0$ and $H_3^r = 0$. From Proposition 5, the sum of eigenvalues of the closed-loop system is given by

$$\lambda_1 + \lambda_2 + \lambda_3 = A_2 + B_1H_3^r \leq A_2,$$

[2]We caution however that the analysis applies only when the linear models we use are reasonable approximations of these AQMs.

where the last inequality follows from that $A_2 < 0$, $B_1 < 0$, and $H_3^r \geq 0$. Since all eigenvalues have nonpositive real parts, the above inequality means that the sum of the real parts of the eigenvalues is less negative when $H_3^r = 0$ than when $H_3^r > 0$. This suggests that the decay rate is smaller with RED ($H_3^r = 0$). The implication of $H_1^r = 0$ is that at least one of the eigenvalues λ_i is zero, implying that the convergence rate is very slow and the original nonlinear system could be unstable according to the center manifold theorem [32]. In case of AVQ, the decay rate of the virtual-queue dynamics is small, but it has a good performance by protecting real queue from building up instead of sacrificing throughput slightly.

Since $H_3^m = 0$ for REM and PI, they suffer from similar structural limitation on decay rate and stability to RED. This means that the proposed PID-type control structure is easier to stabilize the given TCP than P-type and PI-type control structures of RED and REM/PI, respectively.

As shown in (47) and (58), the cost of RED/AVQ, and REM/PI can be obtained from (47) by setting some elements of K to zero, with $H_1 = H_3 = 0$ (i.e., $K_{11} = K_{12} = K_{13} = K_{33} = 0$) and with $H_3 = 0$ (i.e., $K_{33} = 0$), respectively. Note that the costs of RED, REM, and AVQ are always greater than that of the RHA since (47) is the optimal cost for the given system and weighting matrices.

6 Simulation Examples

In the following, we illustrate the performance of RHA (46) via simulation for the linearized TCP/AQM model (16).

We simulate a single link of round-trip time $\tau = 0.25$sec, capacity $c = 4000$ packets/sec shared by $N = 100$ TCP sources. In the simulations, we impose a constraint of 800 packets on $b(t)$:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -3.1373 & -4.7843 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ -81600 \end{bmatrix}, \quad x_0 = \begin{bmatrix} 0 \\ 10 \\ 0 \end{bmatrix}. \tag{65}$$

Sampling time is 2 msec and the total simulation time is 4 sec (2000 steps).

6.1 Delay-free third-order linearized system

In this subsection, we present some simulation results to illustrate our discussions about AQM design without delay compensation. We simulate the linearized system (16) with $\tau = 0$ in $\delta\dot{u}(t-\tau)$ in order to illustrate performance limitation of existing AQMs. In the next subsection, we present simulations that illustrate the effect of delay for the linearized system. We compare the proposed RHA with a simplified RED, REM/PI, and D-type (Derivative) AQM. Since AVQ is based on an adaptive virtual-queue dynamics, we do not compare it with the RHA in this paper.

Using Proposition 6, we choose the eigenvalues of the linearized closed-loop system to be $\lambda_1 = \lambda_2 = \lambda_3 = \alpha < 0$. This sets the weighting matrix in the cost function J:

$$Q_1 = \left(\frac{\alpha^3}{B_1}\right)^2, \qquad Q_2 = \frac{-A_1^2 + 3\alpha^4}{B_1^2}, \qquad Q_3 = \frac{-A_2^2 - 2A_1 + 3\alpha^2}{B_1^2}.$$

We set $\alpha = -10$, giving $H_1 = -B_1 K_{13} = 0.0123$, $H_2 = -B_1 K_{23} = 0.0036$, $H_3 = -B_1 K_{33} = 0.000309$. For RED, we use `max_p` $= 0.1$, `min_th` $= 50$ pkts, `max_th` $= 650$ pkts, giving a control gain of $H_2^r = 1/600$. For REM/PI, we use $H_1^m = 0.001$ and $H_2^m = 0.01$. For D-type control, we simply take $H_3 = \frac{-A2-2\sqrt{-A1}}{B1}$ to be the control gains for AVQ. The closed-loop system has two repeated negative eigenvalues and one zero eigenvalue. We compare these AQMs both in terms of their costs $J(u(\cdot))$ and in terms of the transients in queue error.

Figure 1 shows the queue trajectory $\delta b(t)$ of each AQM. The queue length of RHA is stable and converges to its equilibrium rapidly. The convergence of RED, REM/PI/AVQ is much slower because of the limitation in decay rate resulting from setting $H_i = 0$ for some i. Queue length of a simplified RED and D-type AQM converges to a nonzero constant value; that of REM/PI exhibits oscillation. Using larger values for H_1 and H_2 will increase the rate of convergence for REM/PI/D-type AQM. However, large H_1 and H_2 can lead to instability if $H_3 = 0$, as in these AQMs.

Figure 2 shows the cost J for each of the AQMs. As expected, RHA has the least cost (0.06 unit). RED, REM/PI, and D-type AQM have higher but finite costs (about 0.12, 0.93, and 200, respectively), indicating that $\delta b(t)$ converge to its equilibrium, slowly. The costs of a simplified RED and D-type AQM grow linearly because of steady-state error that results from setting $H_1^r = H_1^a = 0$ (i.e., one of eigenvalues is zero). Note that this does not mean that the original nonlinear system has constant queue length.

Figure 3 shows the values of each AQM.

6.2 Delayed third-order linearized system

In this subsection, we illustrate the performance of RHA (46) with delay compensation for the system (16).

For implementation of the RHA, we set $Q_1 = (\frac{\alpha^3}{\hat{B}_1})^2$, $Q_2 = \frac{-A_1^2+3\alpha^4}{\hat{B}_1^2}$, and $Q_3 = \frac{-A_2^2-2A_1+3\alpha^2}{\hat{B}_1^2}$, and approximate (28) with L steps (for this example, $L = 125$):

$$\begin{bmatrix} u_{1\tau}(t) \\ u_{2\tau}(t) \\ u_{3\tau}(t) \end{bmatrix} = \frac{B_1}{(a_1 - a_2)} \frac{\tau}{L} \left\{ \begin{bmatrix} 0 \\ 0 \\ a_1 - a_2 \end{bmatrix} u(t-\tau) + \begin{bmatrix} \frac{a_1-a_2}{a_1 a_2} + \frac{e^{-\frac{\tau}{L}a_1}}{a_1} - \frac{e^{-\frac{\tau}{L}a_2}}{a_2} \\ e^{-\frac{\tau}{L}a_1} - e^{-\frac{\tau}{L}a_2} \\ a_1 e^{-\frac{\tau}{L}a_1} - a_2 e^{-\frac{\tau}{L}a_2} \end{bmatrix} u\left(t + \frac{\tau}{L}(1-L)\right) \right.$$
$$\left. + \cdots + \begin{bmatrix} \frac{a_1-a_2}{a_1 a_2} + \frac{e^{-\frac{\tau}{L}(L-1)a_1}}{a_1} - \frac{e^{-\frac{\tau}{L}(L-1)a_2}}{a_2} \\ e^{-\frac{\tau}{L}(L-1)a_1} - e^{-\frac{\tau}{L}(L-1)a_2} \\ a_1 e^{-\frac{\tau}{L}(L-1)a_1} - a_2 e^{-\frac{\tau}{L}(L-1)a_2} \end{bmatrix} u\left(t - \frac{\tau}{L}\right) \right\}. \tag{66}$$

Figure 4 shows that the RHA (46) with delay compensation makes the queue length (or,

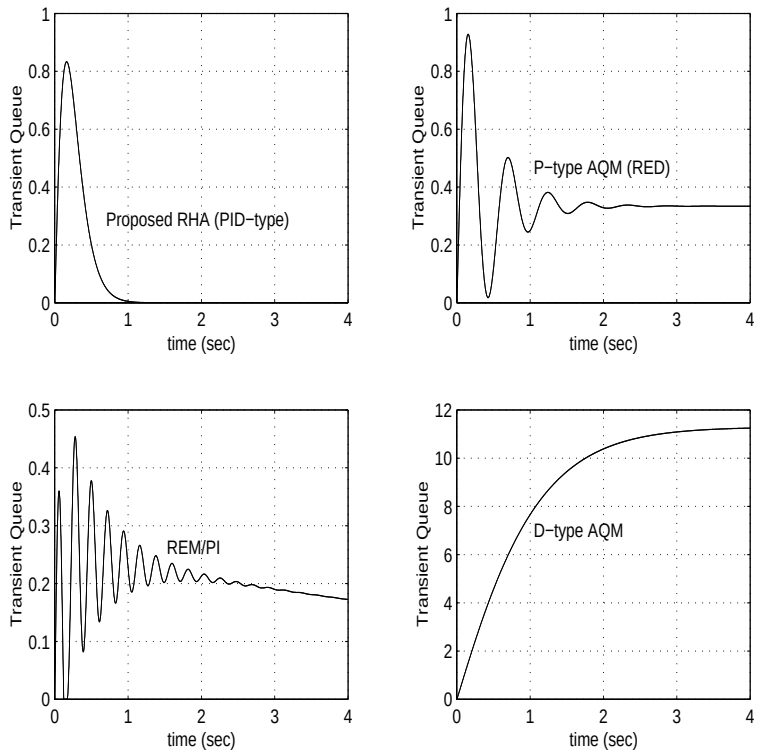

Figure 1: Queue trajectory $\delta b(t)$ for a linearized model

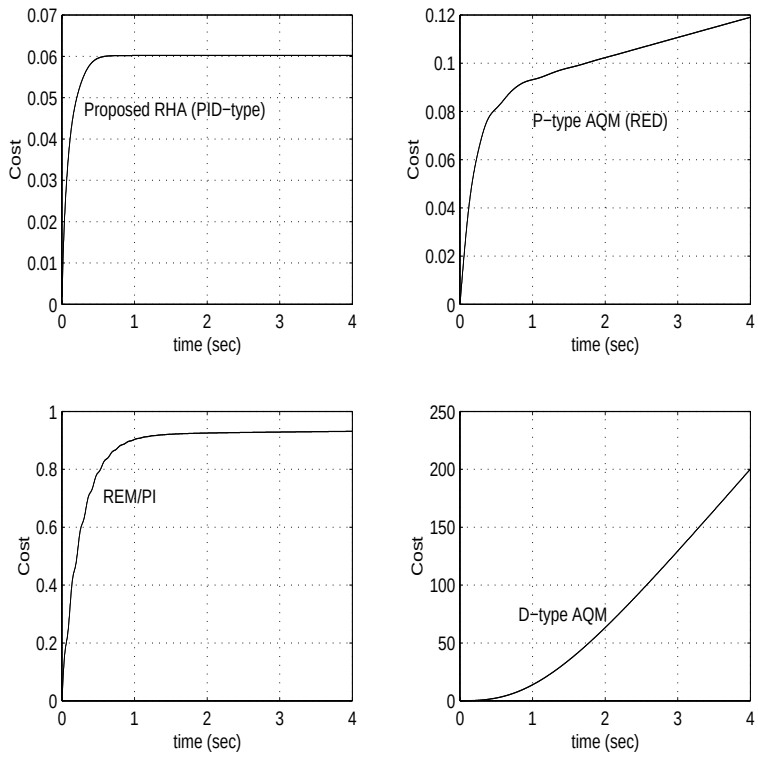

Figure 2: Cost J of each AQM for a linearized model

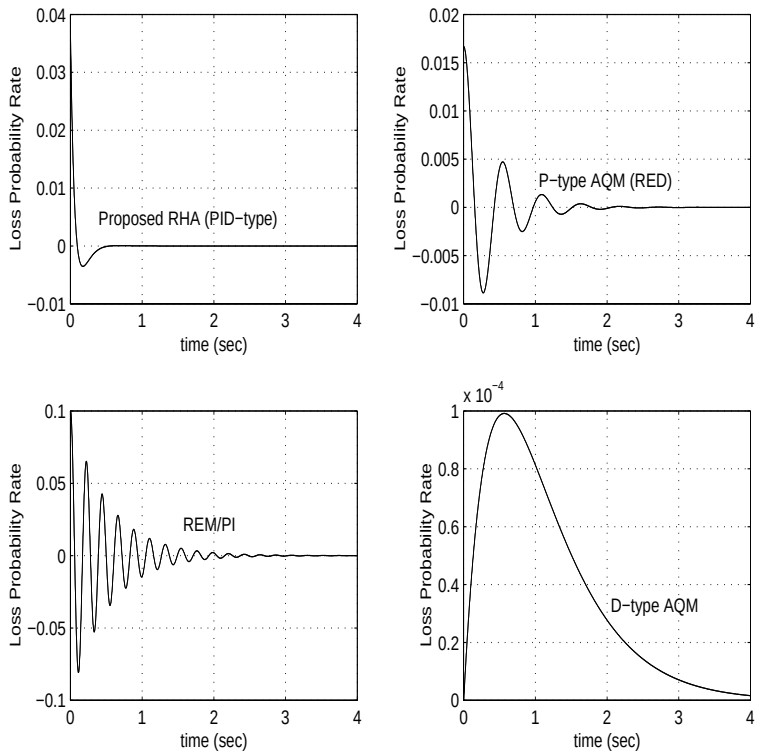

Figure 3: Input ($\delta\dot{p}$) trajectory for a linearized model

source rate) go to zero, while the other AQMs cause instability. As mentioned in Remark 2, having large Q_i violates the state constraint and thus, makes the closed-loop system unstable.

Figure 5 shows the values of each AQM.

7 Conclusion

In this paper, we proposed a PD-type (Proportional-Derivative) AQM structure and by applying integral control action, a PID-type (Proportional-Integral-Derivative) AQM structure that is a unified mathematical framework. We showed that we can compensate for delays in congestion measure explicitly by using a memory control structure. We also proposed stabilizing optimal AQMs for linearized systems of the given TCP by minimizing linear quadratic costs of the transients in queue length, aggregate rate, jitter in the aggregate rate, and the congestion measure, which is called RHA (Receding Horizon AQM) in this paper. We also showed that any AQM with an appropriate structure solves the same stabilizing optimal control problem with appropriate weighting matrices. We interpreted existing AQMs, including a simplified RED (Random Early Detection) without a low-pass filter and saturation functions, REM (Random Exponential Marking), PI (Proportional-Integral), and a simplified AVQ (Adaptive Virtual Queue) without an adaptive virtual-queue dynamics, as different approximations of the proposed unified AQM structure. Finally, we discussed the impact of each structure on performance from the results

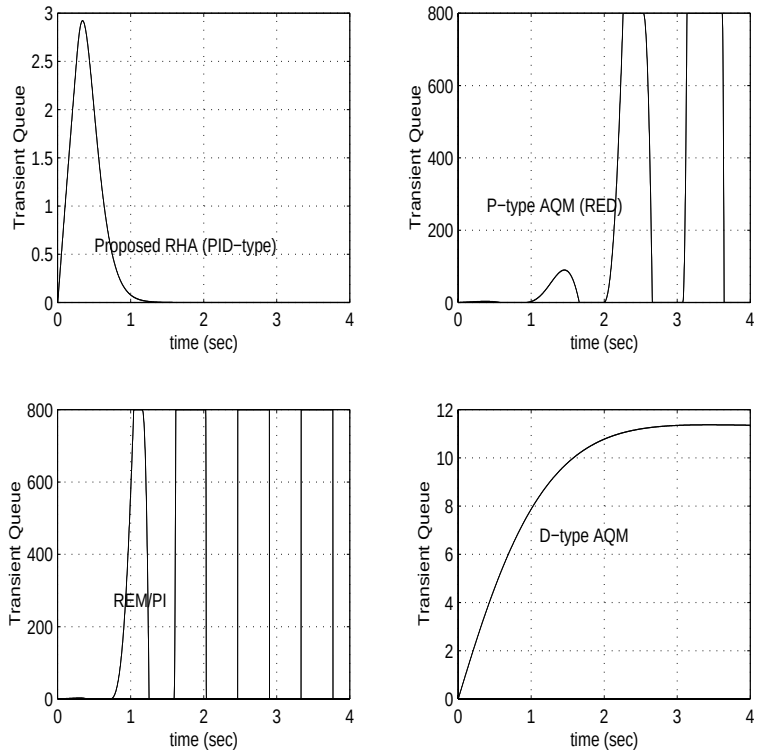

Figure 4: Queue trajectory $\delta b(t)$ for a linearized model

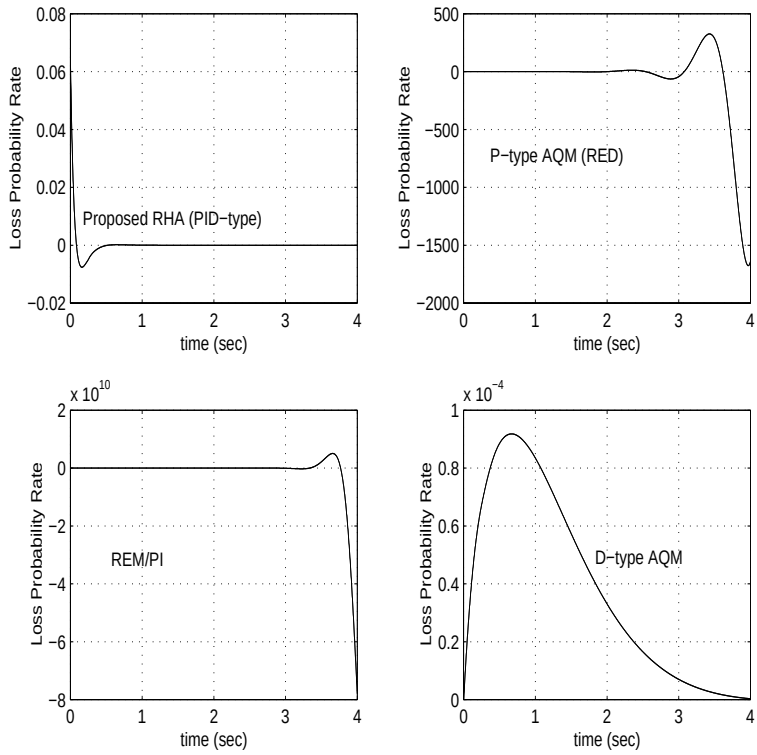

Figure 5: Input ($\delta\dot{p}$) trajectory for a linearized model

of the stabilizing optimal AQMs.

Since we formulated the AQM design problem for the given TCP as state-space models, we got three important features. First, we started from a PD-type control and then derived a PID-type control, while most existing AQMs based on transfer function models start from P-type control and then add I-type control in order to make the equilibrium queue length go to the target queue length [6, 17], or add virtual-queue dynamics in order to reduce the effect of real-queue dynamics in [11, 12, 21]. Note that the PD-type AQM can be considered as a lead-lag type compensator when we think about the transfer function model, and PD-type and PID-type AQMs was not captured by the primal-dual models in [7, 8, 9, 10, 11, 12, 13, 14, 15] as mentioned in Remark 1. Second, this paper compensated for input delays in congestion measure explicitly by using simple extensions of the reduction and transformation techniques, where the reduction technique based on a state-space model corresponds to the well-known Smith Predictor technique based on a transfer function model. Third, this paper proved relationships between each AQM structure and stabilizing optimal AQMs near the equilibrium point in terms of appropriate weighting matrices and eigenvalues of the closed-loop system.

We believe that PD-type, PID-type, and memory AQM structures are necessary to not only stabilize the given TCP and real-queue dynamics with minimum cost but also maximize throughput. This work represents a first step in developing a unified mathematical framework to analyze, and synthesize, AQMs that can stabilize different TCP algorithms and queue dynamics. Two directions of future research seem worthwhile pursuing. First, we need to conduct more extensive simulations to verify merits of the proposed AQM structures in this paper and extend our understanding of each structure. Second, it would be interesting to develop practical AQMs that are decentralized and can better approximate the optimal AQM than existing proposals for original nonlinear systems with heterogeneous delays, various constraints, multiple links, and random variables. Especially, we will try to enjoy an adaptive virtual-queue dynamics that is an essential structure in AVQ and a low-pass filter in RED. We believe that this work will make old and new TCP algorithms achieve their maximum performance, and will also be useful for analysis and design of other dynamical systems with the same structure.

Acknowledgement: The first author acknowledges the support of the Post-doctoral Fellowship Program of Korean Science & Engineering Foundation (KOSEF).

References

[1] K. B. Kim and S. H. Low, "Cost of AQM in stabilizing TCP." Caltech Technical Report caltechCSTR:2002.008, February 2002. `http://cisl.snu.ac.kr/~kkb,http://caltechcstr.library.caltech.edu/`.

[2] V. Jacobson, "Congestion avoidance and control," in *Proceedings of ACM/SIGCOMM*, 1988. `ftp://ftp.ee.lbl.gov/papers/congavoid.ps.Z`.

[3] L. S. Brakmo and L. L. Peterson, "TCP vegas: end to end congestion avoidance on a global internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 8, 1995. `http://netweb.usc.edu/yaxu/Vegas/Reference/brakmo.ps`.

[4] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. on Networking*, vol. 1, no. 4, pp. 397–413, 1993. `ftp://ftp.ee.lbl.gov/papers/early.ps.gz`.

[5] S. Floyd, "TCP and explicit congestion notification," *ACM Computer Communications Review*, vol. 24, pp. 10–23, 1994. `ftp://ftp.ee.lbl.gov/papers/tcp_ecn_4.ps.gz`.

[6] C. Hollot, V. Misra, D. Towsley, and W. B. Gong, "On designing improved controllers for AQM routers supporting TCP flows," in *Proceedings of IEEE/INFOCOM*, 2001. `http://www-net.cs.umass.edu/networks/publications.html`.

[7] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *Journal of Operations Research Society*, vol. 49, no. 3, pp. 237–252, 1998. `http://www.statslab.cam.ac.uk/~frank/mmi.html`.

[8] F. P. Kelly, "Mathematical modelling of the Internet," in *Proc. 4th International Congress on Industrial and Applied Mathematics*, July 1999. `http://www.statslab.cam.ac.uk/~frank/mmi.html`.

[9] S. H. Low and D. E. Lapsley, "Optimization flow control, I: basic algorithm and convergence," *IEEE/ACM Transactions on Networking*, vol. 7, pp. 861–874, December 1999. `http://netlab.caltech.edu`.

[10] S. H. Low, "A duality model of TCP flow controls," in *Proceedings of ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management*, September 18-20 2000. `http://netlab.caltech.edu`.

[11] S. Kunniyur and R. Srikant, "End–to–end congestion control schemes: utility functions, random losses and ECN marks," in *Proceedings of IEEE/INFOCOM*, 2000. `http://www.ieee-infocom.org/2000/papers/401.ps`.

[12] S. Kunniyur and R. Srikant, "A time–scale decomposition approach to adaptive ECN marking," in *Proceedings of IEEE/INFOCOM*, 2001. `http://comm.csl.uiuc.edu:80/~srikant/pub.html`.

[13] F. Paganini, J. C. Doyle, and S. H. Low, "Scalable laws for stable network congestion control," in *Proceedings of Conference on Decision and Control*, December 2001. `http://www.ee.ucla.edu/~paganini`.

[14] G. Vinnicombe, "A new TCP with guaranteed network stability," tech. rep., Cambridge Univiversity, preprint, June 2001.

[15] S. H. Low, F. Paganini, and J. C. Doyle, "Internet congestion control," *IEEE Control Systems Magazine*, February 2002.

[16] S. Kunniyur and R. Srikant, "Analysis and design of an adaptive virtual queue (AVQ) algorithm for active queue management," in *Proceedings of ACM/SIGCOMM*, 2001. `http://comm.csl.uiuc.edu/~srikant/pub.html`.

[17] S. Athuraliya, V. H. Li, S. H. Low, and Q. Yin, "REM: active queue management," *Submitted for publication*, 2000. `http://netlab.caltech.edu/pub.html`.

[18] S. H. Low, L. Peterson, and L. Wang, "Understanding Vegas: a duality model," in *Proceedings of ACM Sigmetrics*, 2001. `http://netlab.caltech.edu/pub.html`.

[19] C. Hollot, V. Misra, D. Towsley, and W.-B. Gong, "A control theoretic analysis of RED," in *Proceedings of IEEE Infocom*, April 2001. `http://www-net.cs.umass.edu/papers/papers.html`.

[20] S. H. Low, F. Paganini, J. Wang, S. A. Adlakha, and J. C. Doyle, "Dynamics of TCP/AQM and a scalable control," in *Proceedings of IEEE Infocom*, June 2002.

[21] R. J. Gibbens and F. P. Kelly, "Distributed connection acceptance control for a connectionless network," in *In proc. of the 16th Intl. Teletraffic Congress*, 1999. Edinburgh, Scotland.

[22] K. B. Kim and W. H. Kwon, "Intervalwise receding horizon H_∞ tracking controls for linear continuous time-varying systems," in *American Control Conference*, (Alaska, USA), May, 2002.

[23] K. B. Kim, T. W. Yoon, and W. H. Kwon, "On stabilizing receding horizon H_∞ controls for linear continuous time-varying systems," *IEEE Transactions on Automatic Control*, vol. 46, no. 8, pp. 1273 – 1279, 2001.

[24] W. H. Kwon and A. E. Pearson, "A modified quadratic cost problem and feedback stabilization of a linear system," *IEEE Trans. Automat. Contr.*, vol. 22, no. 5, pp. 838 – 842, 1977.

[25] H. Chen and F. Allgower, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," *Automatica*, vol. 34, pp. 1205–1217, 1998.

[26] G. De Nicolao and L. Magni and R. Scattolini, "Stabilizing receding horizon control of nonlinear time-varying systems," *IEEE Trans. Automat. Contr.*, vol. 43, no. 7, pp. 1030 – 1036, 1998.

[27] K. B. Kim, "Generalized receding horizon control scheme for constrained linear discrete-time systems," in *Proc. of 15st IFAC World Congress on Automatic Control*, (Barcelona, Spain), 2002.

[28] C. E. García, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice – a survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.

[29] K. B. Kim, M. J. Kim, and W. H. Kwon, "Receding Horizon Guidance Laws with no information on the Time-to-Go," *Journal of Guidance, Control, and Dynamics*, vol. 23, no. 2, pp. 193 – 199, 2000.

[30] D. E. Kirk, *Optimal Control Theory.* Englewood Cliffs, New Jersey: Prentice-Hall, 1970.

[31] D. O. Anderson and J. B. Moore, *Optimal Control : Linear Quadratic Methods.* Englewood Cliffs, NJ 07632: Prentice-Hall International Editions, 1989.

[32] H. K. Khalil, *Nonlinear systems.* Prentice Hall, 1996.

[33] W. H. Kwon and A. E. Pearson, "Feedback stabilization of linear systems with delayed control," *IEEE Trans. Automat. Contr.*, vol. 25, no. 2, pp. 266 – 269, 1980.

A Derivation of (16)

Let (w^*, b^*, p^*) be the equilibrium point. Then the linearized model of TCP Reno (11) (or its variants such NewReno and SACK) is

$$\delta\ddot{b}(t) = \frac{\partial f}{\partial b(t)}|_*\delta b(t) + \frac{\partial f}{\partial \dot{b}(t)}|_*\delta\dot{b}(t) + \frac{\partial f}{\partial \dot{b}(t-\tau)}|_*\delta\dot{b}(t-\tau) + \frac{\partial f}{\partial p(t-\tau)}|_*\delta p(t-\tau),$$

where

$$\begin{aligned} \frac{\partial f}{\partial b(t)}|_* &= -\frac{2N}{\tau^3 c}(1-p^*), & \frac{\partial f}{\partial \dot{b}(t)}|_* &= -\frac{Nc}{\tau^2 c^2}(1-p^*) - \frac{1}{\tau} - \frac{p^* c}{2N} \\ \frac{\partial f}{\partial \dot{b}(t-\tau)}|_* &= 0, & \frac{\partial f}{\partial p(t-\tau)}|_* &= -\frac{N}{\tau^2} - \frac{c^2}{2N}, \qquad p^* = \frac{2N^2}{2N^2 + c^2\tau^2}. \end{aligned} \tag{A.67}$$

From (A.67), the linearized model of TCP Reno (11) (or its variants such NewReno and SACK) can be converted to (12).

Similarly, the linearized model of TCP Reno (15) is

$$\begin{aligned} \delta\dddot{b}(t) &= \frac{\partial g}{\partial b(t)}|_*\delta b(t) + \frac{\partial g}{\partial \dot{b}(t)}|_*\delta\dot{b}(t) + \frac{\partial g}{\partial \ddot{b}(t)}|_*\delta\ddot{b}(t) + \frac{\partial g}{\partial \dot{b}(t-\tau)}|_*\delta\dot{b}(t-\tau) \\ &\quad + \frac{\partial g}{\partial \ddot{b}(t-\tau)}|_*\delta\ddot{b}(t-\tau) + \frac{\partial g}{\partial p(t-\tau)}|_*\delta p(t-\tau) + \frac{\partial g}{\partial \dot{p}(t-\tau)}|_*\delta\dot{p}(t-\tau) \\ &= \frac{\partial f}{\partial b(t)}|_*\delta\dot{b}(t) + \frac{\partial f}{\partial \dot{b}(t)}|_*\delta\ddot{b}(t) + \frac{\partial f}{\partial p(t-\tau)}|_*\delta\dot{p}(t-\tau). \end{aligned}$$

From (A.67), the linearized model of TCP Reno (15) (or its variants such NewReno and SACK) can be converted to (16).

B Derivation of (12) and (24)

We have only to derive (24) since we can handle (19) as a special case of (24).

Employing the reduction method [33], we can transform $z_e(t)$ and $\{\dot{u}(t+s), s \in [-\tau, 0]\}$ to $y_e(t)$ as follows. Note that the system (16) can be written as

$$\begin{aligned} z_e(t+\tau) &= e^{A_e\tau} z_e(t) + \int_t^{t+\tau} e^{A_e(t+\tau-\sigma)} B_e \dot{u}(\sigma-\tau) d\sigma \\ &= e^{A_e\tau}[z_e(t) + \int_{-\tau}^{0} e^{-A_e(\sigma+\tau)} B_e \dot{u}(\sigma+t) d\sigma], \end{aligned} \tag{B.68}$$

where

$$e^{A_e t} = \frac{1}{a_2 - a_1}\begin{bmatrix} a_2 - a_1 & \frac{a_1^2 - a_2^2}{a_1 a_2} + \frac{a_2}{a_1}e^{a_1 t} - \frac{a_1}{a_2}e^{a_2 t} & \frac{a_2 - a_1}{a_1 a_2} - \frac{1}{a_1}e^{a_1 t} + \frac{1}{a_2}e^{a_2 t} \\ 0 & a_2 e^{a_1 t} - a_1 e^{a_2 t} & -e^{a_1 t} + e^{a_2 t} \\ 0 & a_1 a_2 (e^{a_1 t} - e^{a_2 t}) & -a_1 e^{a_1 t} + a_2 e^{a_2 t} \end{bmatrix}. \tag{B.69}$$

Define

$$y_e(t) \;=\; z_e(t) + \int_{-\tau}^{0} e^{-A_e(\sigma+\tau)} B_e \dot{u}(t+\sigma) d\sigma \;=\; z_e(t) + \begin{bmatrix} u_{1\tau}(t) \\ u_{2\tau}(t) \\ u_{3\tau}(t) \end{bmatrix}. \tag{B.70}$$

Using (B.68) and (B.70), the system (16) can be rewritten as

$$\dot{y}_e(t) \;=\; A_e y_e(t) + \tilde{B}_e \dot{u}(t), \tag{B.71}$$

where

$$\tilde{B}_e \;=\; \begin{bmatrix} \tilde{B}_1 \\ \tilde{B}_2 \\ \tilde{B}_3 \end{bmatrix} \;=\; e^{-A_e\tau} B_e \;=\; \frac{B_1}{a_1 - a_2} \begin{bmatrix} (\frac{a_1-a_2}{a_1 a_2} + \frac{1}{a_1} e^{-a_1\tau} - \frac{1}{a_2} e^{-a_2\tau}) \\ (e^{-a_1\tau} - e^{-a_2\tau}) \\ (a_1 e^{-a_1\tau} - a_2 e^{-a_2\tau}) \end{bmatrix}. \tag{B.72}$$

Let $s_e(t) = T_e y_e(t)$, where $T_e = \begin{bmatrix} -\frac{A_1\tilde{B}_2^2 + A_2\tilde{B}_2\tilde{B}_3 - \tilde{B}_3^2}{\tilde{B}_2(A_1\tilde{B}_1 + A_2\tilde{B}_2 - \tilde{B}_3)} & \frac{A_1\tilde{B}_1\tilde{B}_2 + A_2\tilde{B}_1\tilde{B}_3 - \tilde{B}_2\tilde{B}_3}{\tilde{B}_2(A_1\tilde{B}_1 + A_2\tilde{B}_2 - \tilde{B}_3)} & \frac{\tilde{B}_2^2 - \tilde{B}_1\tilde{B}_3}{\tilde{B}_2(A_1\tilde{B}_1 + A_2\tilde{B}_2 - \tilde{B}_3)} \\ 0 & -\frac{\tilde{B}_3}{\tilde{B}_2} & 1 \\ 0 & A_1 & \frac{A_2\tilde{B}_2 - \tilde{B}_3}{\tilde{B}_2} \end{bmatrix}$.

T_e can be rewritten as $T_e = \begin{bmatrix} \frac{(a_2-a_1)e^{-(a_1+a_2)\tau}}{e_1} & e_4 & e_5 \\ 0 & -\frac{e_2}{e_1} & 1 \\ 0 & A_1 & \frac{e_3}{e_1} \end{bmatrix}$ since

$$\frac{A_1\tilde{B}_2^2 + A_2\tilde{B}_2\tilde{B}_3 - \tilde{B}_3^2}{\tilde{B}_2} \;=\; -\frac{B_1^2}{\tilde{B}_2} e^{-(a_1+a_2)\tau} = \frac{B_1(a_2-a_1)}{e_1} e^{-(a_1+a_2)\tau}, \quad A_1\tilde{B}_1 + A_2\tilde{B}_2 - \tilde{B}_3 \;=\; -B_1$$

$$\frac{A_1\tilde{B}_1\tilde{B}_2 + A_2\tilde{B}_1\tilde{B}_3 - \tilde{B}_2\tilde{B}_3}{\tilde{B}_2} = \frac{B_1}{e_1}\left[\frac{a_1}{a_2} e^{-a_1\tau} - \frac{a_2}{a_1} e^{-a_2\tau} + \frac{a_2^2 - a_1^2}{a_1 a_2} e^{-(a_1+a_2)\tau}\right]$$

$$\frac{\tilde{B}_2^2 - \tilde{B}_1\tilde{B}_3}{\tilde{B}_2} = -\frac{B_1}{e_1}\left[\frac{1}{a_2} e^{-a_1\tau} - \frac{1}{a_1} e^{-a_2\tau} + \frac{a_2 - a_1}{a_1 a_2} e^{-(a_1+a_2)\tau}\right]$$

Since $a_2 < a_1 < 0$ for system matrices of (16), $\tilde{B}_1 < 0$, $\tilde{B}_2 > 0$, and $\tilde{B}_3 < 0$ for $\tau > 0$. Then, the system (B.71) can be rewritten as

$$\dot{s}_e(t) \;=\; S A_e S^{-1} s(t) + S\tilde{B}_e \dot{u}(t) \;=\; A_e s_e(t) + \begin{bmatrix} 0 \\ 0 \\ \frac{A_1\tilde{B}_2^2 + A_2\tilde{B}_2\tilde{B}_3 - \tilde{B}_3^2}{\tilde{B}_2} \end{bmatrix} \dot{u}(t).$$

Since $\det(S) = \frac{B_1^3}{\tilde{B}_2^3} \neq 0$, there exists S^{-1}. Thus, (B.71) can be rewritten as (24).

Similarly, we can get (19).