Optimal Threshold Policies for Admission Control in Communication Networks via Discrete Parameter Stochastic Approximation

Shalabh Bhatnagar [*]
and
I. Bala Bhaskar Reddy

Department of Computer Science and Automation,
Indian Institute of Science,
Bangalore 560 012, India.

E-Mail: shalabh@csa.iisc.ernet.in,
ibbr@rediffmail.com

Fax: +91 (80) 2360 2911

Submitted February 2003; Revised July 2004

[*] Author for correspondence. E-Mail for correspondence: shalabh@csa.iisc.ernet.in

Abstract

The problem of admission control of packets in communication networks is studied in the continuous time queueing framework under different classes of service and delayed information feedback. We develop and use a variant of a simulation based two timescale simultaneous perturbation stochastic approximation (SPSA) algorithm for finding an optimal feedback policy within the class of threshold type policies. Even though SPSA has originally been designed for continuous parameter optimization, its variant for the discrete parameter case is seen to work well. We give a proof of the hypothesis needed to show convergence of the algorithm on our setting along with a sketch of the convergence analysis. Extensive numerical experiments with the algorithm are illustrated for different parameter specifications. In particular, we study the effect of feedback delays on the system performance.

Key Words Admission control, communication networks, regularized semi-Markov modulated Poisson process, two timescale stochastic approximation, simultaneous perturbation stochastic approximation, threshold type policies.

The problem of finding optimal policies for admission control / resource allocation [2], [12], [17], [20], [22], [23], [21], [30] is a very important problem particularly in high speed networks. The key objectives here are to maximize network revenues or profits and at the same time provide good quality of service (QoS) to the customers. For instance, in asynchronous transfer mode (ATM) networks, customers can choose between higher priority services like constant bit rate (CBR) and variable bit rate (VBR) on the one hand, and a lower priority, best effort, service like available bit rate (ABR) on the other. A new connection is typically admitted in the network only if there are sufficient resources available for it, taking into account the resource requirements of the existing connections. Moreover, the switches need to perform admission control over packets (in particular) from low priority connections, in the case of networks that provide service guarantees like low delay and high throughput to the high priority services.

Admission control in communication networks has been a well studied research area with various (stochastic and deterministic) models and settings considered by researchers over the last several years. For instance, in [22], a fluid flow model of call admission control on a link in ATM networks is considered. In [23], certain schedulability conditions that ensure a bounded delay in packet switched networks are presented in a deterministic setting. A fuzzy logic neural network based controller is used for this problem in [12]. A probabilistic framework for packet based networks where admission control decisions are made by end systems instead of network resources is considered and analyzed in [20]. In [24], the setting of integrated services networks is considered and a neuro-dynamic programming (NDP) [3] based temporal difference algorithm is used for admission control.

In this paper, we consider a stochastic setting with a continuous time queueing model of the system under quite general assumptions on system / process dynamics. We provide a simulation based stochastic approximation algorithm to compute an optimal policy within the class of threshold type feedback policies. Markov decision processes (MDP) [1], [26] is a general methodology for finding optimal control policies for stochastic dynamic systems. However, finding optimal policies using MDP is computationally expensive in most cases of interest. Moreover, in order to use an MDP based approach, one needs complete knowledge of system dynamics or transition probabilities which may be hard to obtain in practice. We present in this paper an efficient, simulation based optimization approach for finding an optimal threshold type policy in a general setting that does not require any knowledge of state transition probabilities.

The algorithm that we consider is motivated by recent advances in perturbation analysis type approaches [13], [14], [15] [19] for parameter optimization. Approaches based on infinitesimal perturbation analysis (IPA) typically use only one simulation of the system for computing the optimum parameter value. This is however achieved through an interchange between the gradient and expectation of the associated cost, for which certain constraining regularity requirements are typically imposed on the system parameters and cost function. In [4] and [5], two stochastic approximation algorithms were developed as alternatives to traditional IPA based schemes. The idea in these is to use two timescales or step-size schedules, with aggregation / averaging of data done on the faster timescale and parameter updated on the slower one. The advantage in doing so is that one directly estimates the gradient of average cost and thus the (above) interchange between gradient and expectation operators as in IPA based schemes is not required. The disadvantage in the above schemes, however, is that both of them use finite difference gradient estimates and hence need $(N+1)$ parallel simulations to update an N-dimensional parameter vector. In a related work [28], a one timescale simultaneous perturbation stochastic approximation (SPSA) algorithm for parameter optimization was developed that uses only two parallel simulations for any N-dimensional parameter vector by simultaneously perturbing all parameter components randomly, most commonly by using independent, symmetric, Bernoulli distributed random variables. In [7], the SPSA based analogs of the two timescale algorithms in [4] and [5] were developed for parameter tuning in the framework of hidden Markov models. In [8], the SPSA variant of the algorithm in [4] was used for finding an optimal feedback policy within a parameterized class of policies for rate based flow control in ABR service in ATM networks.

We use in this paper the discrete parameter variant of one of the algorithms in [7] for finding an optimal policy amongst threshold type policies for admission control. Our main model is described in Section 1. The search space is a finite, discrete set and not a continuously valued one. In [16], a variant of the one-timescale SPSA algorithm [28] is used for finding the optimal policy in a resource allocation problem by projecting the algorithm after each update on to the finite search grid so that the algorithm moves only on the grid. We use a similar idea except for the difference that even though we project the parameter after each update on to the finite grid, we use the projected updates only in the feedback policies. Our algorithm essentially updates the parameter in the convex hull of points on the finite grid and is hence more efficient. In particular, we can work with diminishing step-size sequences as with regular stochastic approximation algorithms, unlike [16] that requires constant step-sizes for proper convergence behaviour. The main contribution of this paper lies in the application of a computationally efficient, simulation based, two-timescale stochastic approximation algorithm adapted to a discrete parameter setting that uses SPSA type gradient estimates towards finding an optimal admission control policy within a prespecified class of feedback policies. The framework of Markov decision processes (MDP) that is normally used for solving stochastic control problems is not considered here because of the amount of computation that would otherwise be required. Our algorithm yields an optimal multilevel threshold type feedback policy and has low computational requirements since it uses SPSA gradient estimates, and performs a search in a continuously valued set.

The rest of the paper is organized as follows: We present the model and problem formulation in Section 1. We also state there the basic results that are required for proving convergence of the algorithm. The algorithm itself is described in Section 2. The main convergence theorem is also stated there. Numerical results are shown in Section 3. The concluding remarks are presented in Section 4. Finally, proofs of all the results described in this paper are given in an appendix at the end.

1 Model and Problem Formulation

By a regularized semi-Markov modulated Poisson process (SMMPP) stream, we mean a Poisson process with rate λ_t at time t, where λ_t is determined by an underlying regularized semi-Markov process $\{X_t\}$. We assume that state transitions of this process take place every $T > 0$ units of time for some T fixed. Further, transitions from any state to itself are allowed. Let $\{X_t\}$ take values in a finite set S_u of cardinality N, which we simply assume has the form $S_u = \{1, \ldots, N\}$. The transitions of $\{X_t\}$ at instants nT, $n > 0$, starting from a given initial value are Markovian in nature, i.e., are independent of past values of X_t, given the current value. Let $X_n \overset{\triangle}{=} X_{nT}$ be the state at time nT of $\{X_t\}$. Then $\{X_n\}$ is an embedded Markov chain for the regularized semi-Markov process $\{X_t\}$ that we assume is time homogeneous i.e., its transition probabilities are time invariant. In addition, we assume $\{X_n\}$ is ergodic. Let $\bar{p}(x_1; x_2)$ denote the probability of transition from $X_n = x_1$ to $X_{n+1} = x_2$, $n \geq 0$ that is assumed independent of n. The rate λ_t of the SMMPP is determined as follows: For $t \in [nT, (n+1)T]$, $\lambda_t = \lambda(i)$ if $X_n = i$, $i = 1, \ldots, N$. For simplicity, we can write the above as $\lambda_t = \lambda(X_t)$ to indicate that the rate of the SMMPP at any time t depends on its state at that instant.

Our model is shown in Figure 1. We consider two streams of packets, one controlled and the other uncontrolled. The uncontrolled stream corresponds to higher priority traffic, while the controlled stream characterizes lower priority traffic that is bursty in nature and receives best effort service. For instance, in the context of an ATM network, the uncontrolled stream may correspond to traffic from CBR or VBR sources while ABR traffic would constitute the controlled stream. The controlled stream is assumed to be an SMMPP while packets from the uncontrolled stream arrive as a Poisson process with rate λ_u. The controlled traffic could largely comprise of data as it can tolerate delays due to packet drops and subsequent retransmissions (for instance, as for ABR sources in ATM networks) while uncontrolled traffic could comprise of real time voice and video that have higher quality of service (QoS) requirements. An SMMPP stream suitably models the bursty nature of the controlled traffic as it also takes into account the fact that connections can come in and leave the system at arbitrary instants of time leading to variability in traffic flows. For simplicity of analysis, we have assumed the uncontrolled stream to be a Poisson process. However, with little extra effort, the uncontrolled stream can also be modelled as another SMMPP.

Packets from both streams are collected at an interim control node (or switch) CN during time intervals $[(n-1)T, nT)$, $n \geq 1$, in course of which they are stored in different buffers. (These correspond to the input buffers at node CN.) We assume both buffers have infinite capacity. At instants nT, the queue length q_n, $n \geq 1$, in the main queue is observed and this information is fed back to node CN. On the basis of this information, a decision on the number of packets to accept from either stream (that are stored in the control buffers) is instantly made. Packets that are not admitted to the main queue from either stream are immediately dropped and they leave the system. Thus the interim buffers at the control node CN are emptied every T units of time with packets from these buffers either joining the main queue or getting dropped. The main queue thus corresponds to a bottleneck node that receives packets from both uncontrolled and controlled streams.

We assume that packets from the uncontrolled source have higher priority and are admitted to the main queue first (from node CN), whenever there are vacant slots available in the main queue buffer which we assume is of size B. Thus if $q_n = i$ and number of uncontrolled packets at node CN are j, then all j uncontrolled packets are admitted to the main queue if $j \leq B - i$, otherwise only the first $B - i$ of these packets are admitted. Admission control (in the regular sense) is performed (at node CN) only on packets from the SMMPP stream that are stored at node CN (which is also the reason that we refer to the latter stream as the controlled stream). Packets from this stream are admitted to the main queue at instants nT based on the queue length q_n of the main queue, the number of uncontrolled stream packets admitted at time nT and also the state X_{n-1} (during interval $[(n-1)T, nT)$) of the underlying Markov process of the SMMPP. Note that node CN plays the role of a multiplexor or scheduler that gives higher priority to the uncontrolled traffic for admitting packets to the main queue. We assume that packets (from either stream) that are accepted in the main queue cannot be dropped subsequently. Moreover, the scheduling policy followed in the main queue is a non-preemptive FCFS (first come, first serve) policy, irrespective of the stream from which the packets arrive (to the main queue).

We assume, however, that information about queue length at the main queue that is fed back to the control node CN every T time instants, reaches node CN with a delay D_b assumed to be a finite and nonnegative constant. Thus the decision on number of packets from either stream to be admitted to the main queue at instants nT, $n \geq 1$, is made based on the latest available queue length information. The packets released from node CN are assumed to reach the main queue instantaneously. We thus assume that transmission delays from node CN to the main queue are negligible. Note, however, that because of the delay D_b in information feedback, we do not have precise instantaneous information on queue length at the main queue i.e., we do not know q_n at instant nT but only its value at some previous instant. It may thus happen that packets released from the control node CN get dropped at the main queue because of unavailable vacant slots at the latter. Packets that are dropped from the main queue because of buffer overflow also immediately leave the system. For ease of exposition, we assume D_b to be a multiple of T, i.e., $D_b = MT$ for some given integer $M > 0$. Simple modifications to the analysis can however take care of the case $D_b \neq MT$ for any integer M.

For the controlled stream, we assume that the feedback policies are of the threshold type. Suppose $\bar{L}_1$, ..., $\bar{L}_N$, are given integers satisfying $0 \leq \bar{L}_j \leq B$, $\forall\, j \in \{1, \ldots, N\}$. Here each $\bar{L}_j$ serves as a threshold for the controlled SMMPP stream when the state of the underlying Markov chain is $X_{n-1} = j$, $j = 1, \ldots, N$, $n \geq 1$. Let A^u_{n-1} and A^c_{n-1} denote the number of arrivals from the uncontrolled and controlled streams, respectively, at node CN during the time interval $[(n-1)T, nT)$. Let r_n denote the residual service time of the customer in service at time nT at the main queue. Also let D_n be the number of departures from the main queue in the interval $[nT, (n+1)T)$. Note that since the controlled packets corresponding to A^c_{n-1} are transmitted to the control node CN during the time interval $[(n-1)T, nT)$, these depend on state X_{n-1} of the underlying Markov chain of the SMMPP in that interval. Let $\bar{\theta} \stackrel{\triangle}{=} (\bar{L}_1, \ldots, \bar{L}_N)^T$ represent the vector of all thresholds. Thus $\bar{\theta}$ takes values in the set $\bar{C}$ $\stackrel{\triangle}{=} \{0, 1, \ldots, B\}^N$. Suppose for $D_b = 0$, let us denote the form of the feedback policies using the function $F_{\bar{\theta}}(q_n, X_{n-1}, A^u_{n-1}, A^c_{n-1})$. The precise form of this function (that we use in this paper) is given below. For any given $\bar{\theta}$, it is easy to see that in the absence of feedback delay (i.e., $D_b = 0$), the joint process $\{(q_n, X_{n-1}, r_n)\}$ is Markov. When $D_b > 0$ ($D_b = MT$), $\{(q_n,$ r_n, X_{n-1}, q_{n-1}, r_{n-1}, X_{n-2}, $\ldots$, q_{n-M}, r_{n-M}, $X_{n-M-1})\}$ is Markov. One can also see that for fixed $\bar{\theta}$, the above processes are in addition ergodic. Let $\bar{h} : \{0, 1, \ldots, B\} \times \{1, \ldots, N\}$ $\to \mathcal{R}$ be a given cost function and let

$$\bar{J}(\bar{\theta}) = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} \bar{h}(q_i, X_{i-1}) \tag{1.1}$$

denote the long-run average cost. The limit in (1.1) exists because of ergodicity of the associated Markov process. Our aim is to find a θ^* on the grid $\{0, 1, \ldots, B\}^N$ that minimizes $\bar{J}(\bar{\theta})$. Note that this is a discrete optimization problem. However, in what follows, we shall use an SPSA based two timescale algorithm that is originally designed for continuous parameter optimization, for the above problem along some what similar lines as in [16]. Broadly the idea in [16] is that for purposes of analysis, a smooth approximation of the cost function is considered for parameter values that lie in between two neighbouring grid points. However, after each iteration the parameter updates are projected onto the original grid (cf. pp.1793 of [16]). We consider a slight modification of this procedure. Basically, even though we project the parameter iterates after each iteration onto the grid, we use these projections only in the feedback policies below. We consider the constraint set for the continuously valued approximation of the parameter vector to be $C \stackrel{\triangle}{=} [0, B]^N$ which is the convex hull of $\bar{C}$. Further, for any $x \in \{1, \ldots, N\}$, let $h(\cdot, x)$ denote a continuous interpolation of $\bar{h}(\cdot, x)$ on the interval $[0, B]$. Also, let $J(\theta)$ be defined analogously as in (1.1) with $h(\cdot,x)$ in place of $\bar{h}(\cdot,x)$ and $\theta \overset{\triangle}{=} (L_1,\ldots,L_N)^T \in C$.

We now describe the form of the feedback policies. For simplicity in exposition, we assume $D_b = 0$ in the description below. Let us fix values for the various quantities of interest as $q_n = i$, $X_{n-1} = j$, $A^u_{n-1} = a^u$ and $A^c_{n-1} = a^c$, respectively, for some $0 \leq i \leq B$, $1 \leq j \leq N$, $0 \leq a^u$, $a^c < \infty$. We then have

Feedback Policies $F_{\bar{\theta}}(i,j,a^u,a^c)$

```
• If $a^u \geq B - i$
  {
  Accept first $(B-i)$ uncontrolled packets and no controlled packets.
  }

• If $i < \bar{L}_j$ and $\bar{L}_j - i \leq a^u < B - i$
  {
  Accept all uncontrolled packets and no controlled packets.
  }

• If $i < \bar{L}_j$ and $\bar{L}_j - i > a^u$
  {
  Accept all uncontrolled packets and

    – If $a^c < \bar{L}_j - i - a^u$
      {
      Accept all controlled packets
      }

    – If $a^c \geq \bar{L}_j - i - a^u$
      {
      Accept first $\bar{L}_j - i - a^u$ controlled packets
      }
}

• If $i \geq \bar{L}_j$ and $B - i > a^u$
{
Accept all uncontrolled packets and no controlled packets.
}
```

In Theorem 1.1, we show that $J(\theta)$ is continuously differentiable in θ under the assumption that service times are exponentially distributed. This result is needed in order to push through a Taylor series argument on the function $J(\cdot)$, that in turn is required to prove convergence of the algorithm to the desired limit points (see proof of Theorem 2.1 given in Appendix for the details). We give the proof of Theorem 1.1 and also of the other results in Appendix.

Theorem 1.1 Under the above feedback policies and under exponential service times, $J(\theta)$ is continuously differentiable in θ.

From Theorem 1.1, we obtain

Corollary 1.1 Under the assumptions of Theorem 1.1, for any given q, q', x, x', the transition probabilities $p_\theta(q, x; q', x')$ are continuous in θ.

Note that we require in Theorem 1.1 - Corollary 1.1 that service times in the main queue have exponential distribution. This is mainly needed because the result from [27] that is used in the proof (see Appendix) is valid only for finite state Markov chains. In [29], certain sufficient conditions for differentiablity of the parameterized stationary distribution w.r.t. the parameter for general state chains are given. These are however difficult to verify in practice. If on the other hand, we assume that $J(\theta)$ is continuously differentiable in θ, then the analysis of the algorithm that follows carries through even when service times have the general distribution. Next, we present the adaptation (on our setting) of the two timescale SPSA algorithm of [7] for finding the optimum parameter and sketch its convergence analysis in Appendix.

2 The Algorithm

For ease of exposition, we describe the algorithm for the case of zero feedback delay, viz., $D_b = 0$. A similar analysis (of the algorithm) as given in Appendix also works for the case when $D_b > 0$. We use the discrete parameter version of the two timescale SPSA algorithm of [7] for computing the optimum parameter. The two timescales correspond to the two step-size schedules $\{a(n)\}$ and $\{b(n)\}$ that satisfy:

$$\sum_{n=1}^{\infty} a(n) = \sum_{n=1}^{\infty} b(n) = \infty, \ \sum_{n=1}^{\infty} a(n)^2, \sum_{n=1}^{\infty} b(n)^2 < \infty, \tag{2.1}$$

$$a(n) = o(b(n)). \tag{2.2}$$

Note that (2.1) are standard requirements on step-size sequences in stochastic approximation algorithms. Note also that as a consequence of (2.1), $a(n), b(n) \to 0$ as $n \to \infty$. The requirement $a(n) = o(b(n))$ given in (2.2) basically means that $a(n)$ goes to zero faster than $b(n)$ does. In other words, recursion (2.6) of the algorithm (below) proceeds on the slower timescale (specified by $a(n)$), while recursions (2.4)-(2.5) proceed on the faster scale (as given by $b(n)$). Note that the L-epoch additional averaging of iterates in (2.4)-(2.5) results in an equivalent 'even faster' timescale. For our numerical experiments, we choose the following as our parameter sequences $\{a(n)\}$ and $\{b(n)\}$, respectively.

$$a(0) = \hat{a}, \ b(0) = \hat{b}, \ a(n) = \hat{a}/n, \ b(n) = \hat{b}/n^{\alpha}, \ n \geq 1, \ \frac{1}{2} < \alpha < 1, \tag{2.3}$$

with $\hat{a} = \hat{b} = 1$. One can in general consider sequences with different values of $\hat{a}$ and $\hat{b}$, and also any other step-size sequences that satisfy (2.1) and (2.2), respectively.

Let $\triangle_l(i)$, $l \geq 0$, $i = 1, \ldots, N$, be mutually independent and mean zero random variables taking values in a compact set $E \subset \mathcal{R}^N$ and having a common distribution. Also, let $\triangle(l)$ denote the vector $\triangle(l) \overset{\triangle}{=} (\triangle_l(1), \ldots, \triangle_l(N))^T$. We assume that random variables $\triangle_l(i)$ satisfy Condition (B) below.

Condition (B) There exists a constant $\bar{K} < \infty$, such that for any $l \geq 0$, and $i \in \{1, \ldots, N\}$, $E\left[(\triangle_l(i))^{-2}\right] \leq \bar{K}$.

Further, each $\triangle(i)$ is assumed to be independent of the σ-field $\sigma(\theta(l), l \leq i)$. Condition (B) is a some what standard condition in SPSA algorithms, see for instance, [28], [11] for similar conditions. Most often, as we do in our numerical experiments, one simply assumes $\triangle_l(i)$ as independent, symmetric, Bernoulli distributed random variables, say, $\triangle_l(i) = \pm 1$, w.p. 1/2 for all $l \geq 0$, $i = 1, \ldots, N$.

Suppose $\delta > 0$ is a given constant. Consider two parallel, independent simulations $\{(q_l^1, X_{l-1}^1, r_l^1)\}$ and $\{(q_l^2, X_{l-1}^2, r_l^2)\}$ that are governed by parameter sequences $\{\bar{\theta}^1(n)\}$ and $\{\bar{\theta}^2(n)\}$, respectively, where $\bar{\theta}^1(n)$ (resp. $\bar{\theta}^2(n)$) is the projection of $\theta(n)$ $-\delta\triangle(n)$ (resp. $\theta(n)$ $+\delta\triangle(n)$) on the grid $\bar{C}$. Here $n = \left[\dfrac{l}{L}\right]$, for some given integer $L \geq 1$. The idea here is that the algorithm updates the parameter $\theta(\cdot)$ once every L epochs, while cost corresponding to the two simulations is aggregated and averaged at each instant on the faster scale. Here an epoch corresponds to T time instants in the basic model. Both $\theta(n)$ and $\triangle(n)$ are held fixed between updates of $\theta(n)$ (for L epochs). Alternatively, writing $l = nL + m$, for $n \geq 0$ and $m \in \{0, \ldots, L-1\}$, $\{(q_l^1, X_{l-1}^1, r_l^1)\}$ and $\{(q_l^2, X_{l-1}^2, r_l^2)\}$ are governed by $\{\bar{\theta}^1(n)\}$ and $\{\bar{\theta}^2(n)\}$, respectively.

The parameter sequence $\{\theta(n)\}$ is updated according to the following algorithm: Let $\{Z^1(l)\}$ and $\{Z^2(l)\}$ be sequences that are defined according to $Z^1(0) = Z^2(0) = 0$ and for $n \geq 0$, $m \in \{0, \ldots, L-1\}$,

$$Z^1(nL+m+1) = Z^1(nL+m) + b(n)(h(q_{nL+m}^1) - Z^1(nL+m)), \tag{2.4}$$

$$Z^2(nL+m+1) = Z^2(nL+m) + b(n)(h(q_{nL+m}^2) - Z^2(nL+m)). \tag{2.5}$$

Now for $i = 1, \ldots, N$, $n \geq 0$,

$$L_i(n+1) = \pi\left(L_i(n) + a(n)\left[\frac{Z^1(nL) - Z^2(nL)}{2\delta\triangle_n(i)}\right]\right). \tag{2.6}$$

Note that $\{Z^1(l)\}$ and $\{Z^2(l)\}$ are used to average the cost function in the two simulations. Also (2.6) corresponds to the parameter update recursion. Here $\pi(\cdot)$ is the projection operator such that $\pi : \mathcal{R} \rightarrow [0, B]$ and is defined by $\pi(x) = \max(\min(x, B), 0)$. The operator π

forces $\theta(n) = (L_1(n), \ldots, L_N(n))^T$ to evolve within the constraint set C. Note that in the feedback policies described in the previous section, we use the parameter $\bar{\theta}(n) = (\bar{L}_1(n), \ldots, \bar{L}_N(n))^T$ which is obtained from $\theta(n)$ by further projecting $\theta(n)$ on the grid $\bar{C}$. This is done by setting each value of $L_i(n)$, $i = 1, \ldots, N$, to the nearest integer in the set $\{0, 1, \ldots, B\}$.

Suppose $K \stackrel{\triangle}{=} \{\theta \in C \mid \nabla J(\theta) = 0\}$ denotes the set of all stable fixed points for the ordinary differential equation (ODE): For $i = 1, \ldots, N$,

$$\dot{L}_i(t) = \hat{\pi}\left(-\nabla_i J(\theta)\right), \tag{2.7}$$

where, $\dot{L}_i(t)$ is the derivative of $L_i(t)$ and $\theta = (L_1, \ldots, L_N)^T$. Also, $\hat{\pi}(\cdot)$ is defined according to:

$$\hat{\pi}(v(\theta(t))) = \lim_{\eta \downarrow 0} \left(\frac{\pi(\theta(t) + \eta v(\theta(t))) - \theta(t)}{\eta} \right),$$

for any bounded and continuous $v(\cdot)$. The operator $\hat{\pi}(\cdot)$ forces the ODE (2.7) to evolve within the constraint set C. Suppose also that for $\epsilon > 0$, K^ϵ denotes the set $K^\epsilon \stackrel{\triangle}{=} \{\theta \in C \mid \parallel \theta - \theta' \parallel < \epsilon \ \forall \ \theta' \in K\}$. Thus K^ϵ is the set of all points that are within an ϵ-neighborhood of the set K. In particular, K^ϵ includes all points in the set K. The following theorem (proof of which is given in Appendix) gives the convergence of the algorithm to a point in the set K^ϵ.

Theorem 2.1 Given $\epsilon > 0$, there exists $\delta_0 > 0$ such that for any $\delta \in (0, \delta_0]$, the algorithm (2.4)-(2.6) converges to some $\theta^* \in K^\epsilon$.

Note above that K is the set of all Kuhn-Tucker points and not just those that correspond to local minima. However, points in K that do not correspond to local minima will be unstable equilibria. In principle, any stochastic approximation scheme may get trapped in an unstable equilibrium. In [25], with noise assumed to be sufficiently 'omnidirectional' in addition, it is shown that convergence to unstable fixed points is not possible; see also [10] for conditions on avoidance of unstable equilibria that lie in certain *compact connected chain recurrent sets.* However, in most cases (even without extra noise conditions) due to the inherent randomness, stochastic approximation algorithms converge to stable equilibria.

Our algorithm would then converge to an ϵ-neighborhood of a local minimum. Note also that Theorem 2.1 merely gives the existence of a δ_0, given $\epsilon > 0$, such that if one uses $\delta \leq \delta_0$ in the algorithm, one is guaranteed convergence to an ϵ-local minimum. However, it does not give any guidance as to how such a δ_0 can a priori be chosen. We observe in our numerical experiments (as also with settings studied in other papers [7], [8], [9]) that a small enough value of δ chosen arbitrarily works well in most cases.

Suppose now that the algorithm converges to a point θ^* which is in an ϵ-neighborhood of the above minimum. Note that the minimum is however one of $J(\theta)$ and not $\bar{J}(\bar{\theta})$. Also, the computed policy would correspond to $\bar{\theta^*}$, or the projection of θ^* on to the finite grid of points $\bar{C}$. However, since $J(\bar{\theta}) = \bar{J}(\bar{\theta})$ for all $\bar{\theta} \in \bar{C}$, and $J(\cdot)$ is continuous (in fact, it is continuously differentiable), one expects that the difference between $J(\theta^*)$ and $J(\bar{\theta^*})$ would in most cases be negligible, by using a suitable choice of δ. We now present our numerical results.

3 Numerical Results

In the numerical experiments, we consider a regularized SMMPP stream with four states, numbered, $1, \ldots, 4$. The transition probability matrix of the underlying Markov chain of this process is chosen to be

$$P = \begin{bmatrix} 0 & 0.6 & 0.4 & 0 \\ 0.4 & 0 & 0.6 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0.7 & 0 & 0.3 \end{bmatrix}.$$

The corresponding rates of the associated Poisson processes in each of these states are chosen as $\lambda(1) = 0.5$, $\lambda(2) = 1.0$, $\lambda(3) = 1.5$ and $\lambda(4) = 2.5$, respectively. The service times in the main queue are considered to be exponentially distributed with rate $\mu = 1.3$. The values of δ and L in the algorithm are chosen to be 0.1 and 100, respectively. The step-sizes $\{a(n)\}$ and $\{b(n)\}$ in the algorithm are taken as $a(0) = b(0) = 1$ and for $n \geq 1$, $a(n) = \dfrac{1}{n}$ and $b(n) = \dfrac{1}{n^{2/3}}$, respectively. We show here the results of experiments with different values of parameters T, D_b and the uncontrolled rate λ_u.

In Figures 2–5, we show the convergence behaviour for the four threshold values, respectively, for the case $T = 3$, $D_b = 6$ and $\lambda_u = 0.4$. The convergence behaviour for the other cases is similar and is hence not shown. Here we refer to the threshold values as $N1$, $N2$, $N3$ and $N4$, respectively, for convenience. Also, we refer to the optimal values of these thresholds (as given by our algorithm) by $N1^*$, $N2^*$, $N3^*$ and $N4^*$, respectively. Thus the parameter θ^* (after convergence of our algorithm) is given by $\theta^* = (N1^*, N2^*, N3^*, N4^*)^T$. We assume that if any packet is accepted in the main queue, the cost incurred is the amount of time that the packet spends in the main queue. In Tables 1–3, we assume that the cost of rejection (which we denote by RC) of a packet is 50. In Table 4, we vary the rejection cost (RC) over a range of values keeping the cost of accepting a packet the same as before (viz., the time that the given packet spends in the main queue). We assume that the size of the buffer in the main queue is 60. Thus the projection operator $\pi(\cdot)$ in algorithm (2.4)-(2.6) projects all threshold update values on to the interval $[0, 60]$. In the following tables, we present the values of the average cost $J(\theta^*)$ and the average fraction (fr^*) of packets rejected under θ^*. These values are obtained after convergence of the algorithm. Initially, we run the system for 20,000 updates of the parameter in all cases. We observe that the parameter converges in all cases in the above number of runs. Further, for computing the performance metrics $J(\theta^*)$ and fr^*, we run the system with the converged values of the parameter (θ^*) for another 20,000 iterations. The initial values of all the four thresholds in all cases are chosen to be 40. Note that even though we uniformly use the notation θ^* to denote the value of the parameter to which the algorithm converges, however, this value in general is a function of the system parameters T, D_b and λ_u, for fixed SMMPP rates and service rate μ (as considered here).

It can be seen that the results obtained are along expected lines. Note from Tables 1 and 2, that as the value of T is increased keeping all other system parameters fixed, both the values of the average cost and average fraction of rejected packets under corresponding values of θ^* increase. This happens because when T is increased, the average number of packets that arrive into the control node buffers in any given interval increases and as a result the system has lesser control over its performance in terms of average cost; more so, since the threshold values are themselves updated after longer time intervals. Because of the above, a small increase in fr^* is observed as well. Also from Table 2 (as expected), note that as the uncontrolled rate λ_u is increased to 1.0, the values of both the average cost and the average fraction of packets rejected increase significantly over corresponding values of these for $\lambda_u = 0.1$ (in Table 1). This is again expected since the higher priority uncontrolled packets now have a significant presence in the main queue buffer leading to an over all increase in values of $J(\theta^*)$ and fr^*, respectively, as packets from the control stream are more frequently rejected. In Table 3, we keep T and λ_u fixed and vary the feedback delay D_b. Note that as D_b increases, both $J(\theta^*)$ and fr^* increase as well, since node CN gets increasingly delayed information on queue length (at the main queue) which results in the controller having lesser control over system performance. In Table 4, we keep T, D_b and λ_u fixed but vary the rejection cost (RC). In this case, the average cost $J(\theta^*)$ increases while the average fraction of rejected packets decreases as RC is increased, which is also expected. Note, however, that in all cases studied, there is no particular order in which the various threshold values converge.

4 Conclusions

We studied the problem of admission control in communication networks in the continuous time queueing framework under delayed information feedback. The system model we considered consists of a single queue that is fed with arrival streams from a control node every T instants of time (for given $T > 0$). The control node in turn has two infinite buffers that are fed with a regularized SMMPP (corresponding to controlled traffic) and an uncontrolled Poisson stream, respectively. Traffic from the uncontrolled stream is assumed to have higher priority and hence is admitted to the main queue first, followed by that from the lower priority controlled source.

We used a discrete parameter variant of a recently developed two timescale SPSA algorithm for computing an optimal feedback policy within the class of threshold type policies. The convergence analysis for our algorithm has been briefly presented in Appendix. Finally, we presented numerical experiments using our algorithm for various values of observation instants, uncontrolled source rates, and feedback delays. We also studied the system behaviour when one of the cost components namely the cost of rejection is increased. All our results are along expected lines.

Recently, two timescale SPSA algorithms that use certain deterministic perturbation sequences in place of randomized, have been developed in [9] in a simulation based parameter optimization setting. The use of deterministic perturbation sequences is seen to considerably improve performance over randomized perturbations in this setting. Further, in [6], the use of certain chaotic iterative sequences for random number generation have also been found to improve performance in SPSA type algorithms. The algorithms of [6] and [9] have primarily been designed for continuous parameter optimization. Variants of these for the discrete parameter case can also be tried on the model considered in this paper.

Acknowledgements

The first author was supported in part through a faculty start-up grant from the Indian Institute of Science and by Grant Number SR/S3/EE/43/2002-SERC-Engg from the Department of Science and Technology, Government of India.

References

[1] Bertsekas, D.P. (2001) *Dynamic Programming and Optimal Control*, second edition, Athena Scientific, Belmont, MA.

[2] Bertsekas, D.P. and Gallager, R. (1992) *Data Networks*, Prentice Hall, Englewood Cliffs, NJ.

[3] Bertsekas, D.P. and Tsitsiklis J.N. (1996) *Neuro-Dynamic Programming*, Athena Scientific, Belmont, MA.

[4] Bhatnagar, S. and Borkar, V.S. (1997) “Multiscale stochastic approximation for parametric optimization of hidden Markov models”, *Probability in the Engineering and Informational Sciences*, 11:509-522.

[5] Bhatnagar, S. and Borkar, V.S. (1998) “A two time scale stochastic approximation scheme for simulation based parametric optimization”, *Probability in the Engineering and Informational Sciences*, 12:519-531.

[6] Bhatnagar, S. and Borkar, V.S. (2003) “Multiscale chaotic SPSA and smoothed functional algorithms for simulation optimization”, *Simulation : Transactions of the Society for Modeling and Simulation International*, 79(10):568-580.

[7] Bhatnagar, S., Fu, M.C., Marcus, S.I. and Bhatnagar, S. (2001) “Two timescale algorithms for simulation optimization of hidden Markov models”, *IIE Transactions*, 33(3):245-258.

[8] Bhatnagar, S., Fu, M.C., Marcus, S.I. and Fard, P.J. (2001) “Optimal structured feedback policies for ABR flow control using two-timescale SPSA”, *IEEE/ACM Transactions on Networking*, 9(4): 479-491.

[9] Bhatnagar, S., Fu, M.C., Marcus, S.I. and Wang, I-J. (2003) “Two-timescale simultaneous perturbation stochastic approximation using deterministic perturbation sequences”, *ACM Transactions on Modelling and Computer Simulation*, 13(2):180-209.

[10] Brandiere, O. (1998) “Some pathological traps for stochastic approximation”, *SIAM J. Contr. and Optim.*, 36:1293-1314.

[11] Chen, H. F. and Duncan, T. E. and P.-Duncan, B. (1999) “A Kiefer-Wolfowitz algorithm with randomized differences”, *IEEE Trans. Autom. Cont.*, 44(3):442-453.

[12] Cheng, R.-G., Chang, C.-J. and Lin, L.-F. (1999) "A QoS provisioning neural fuzzy connection admission controller for multimedia high speed networks", *IEEE/ACM Trans. on Network.*, 7(1):111-121.

[13] Chong, E.K.P. and Ramadge, P.J. (1993) "Optimization of queues using an infinitesimal perturbation analysis-based stochastic algorithm with general update times", *SIAM J. Contr. and Optim.*, 31(3):698-732.

[14] Chong, E.K.P. and Ramadge, P.J. (1994) "Stochastic optimization of regenerative systems using infinitesimal perturbation analysis", *IEEE Trans. on Autom. Contr.*, 39(7):1400-1410.

[15] Fu, M.C. (1990) "Convergence of a stochastic approximation algorithm for the $GI/G/1$ queue using infinitesimal perturbation analysis", *J. Optim. Theo. Appl.*, 65:149-160.

[16] Gerencsér, L., Hill, S.D. and Vágó, Z. (1999), "Optimization over discrete sets via SPSA", *Proceedings of the IEEE Conference on Decision and Control*, 1791-1795.

[17] Grossglauser, M., Keshav, S. and Tse, D.N.C. (1997) "RCBR: A simple and efficient service for multiple time-scale traffic", *IEEE Trans. on Network.*, 5(6):741-755.

[18] Hirsch, M.W. (1989) "Convergent activation dynamics in continuous time networks", *Neural Networks*, 2:331-349.

[19] Ho, Y.-C. and Cao, X.-R. (1991) *Perturbation Analysis of Discrete Event Dynamical Systems*, Kluwer, Boston.

[20] Kelly, F.P., Key, P.B. and Zachary, S. (2000) "Distributed admission control", *IEEE Journal on Selected Areas in Communications*, 18:2617-2628.

[21] Keshav, S. (1997) *An Engineering Approach to Computer Networking*, Addison-Wesley, New York.

[22] Lee, T.-H., Lai, K.-C. and Duann, S.-T. (1996) "Design of a real-time admission controller for ATM Networks", *IEEE/ACM Trans. on Network.*, 4(5):758-765.

[23] Liebeherr, J., Wrege, D.E. and Ferrari, D. (1996) "Exact admission control for networks with a bounded delay service", *IEEE/ACM Trans. on Network.*, 4(6):885-901.

[24] Marbach, P., Mihatsch, O. and Tsitsiklis, J. N. (2000), "Call Admission Control and Routing in Integrated Service Networks Using Neuro-Dynamic Programming", *IEEE Journal on Selected Areas in Communications*, 18(2):197-208.

[25] Pemantle, R. (1990) "Nonconvergence to unstable points in urn models and stochastic approximations", *Annals of Prob.*, 18:698-712.

[26] Puterman, M. L. (1994) *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, John Wiley, New York.

[27] Schweitzer, P. J. (1968), "Perturbation theory and finite Markov chains", *J. Appl. Prob.*, 5:401-413.

[28] Spall, J.C. (1992) "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation", *IEEE Trans. Autom. Contr.*, 37(3):332-341.

[29] Vazquez-Abad, F.J. and Kushner, H.J. (1992) "Estimation of the derivative of a stationary measure with respect to a control parameter", *J. Appl. Prob.*, 29:343-352.

[30] Walrand, J. and Varaiya, P. (2000) *High-Performance Computer Networks*, Morgan Kauffman, San Mateo, CA.

Table 1: $\lambda_u = 0.1,\ D_b = 0$

T	$N1^*$	$N2^*$	$N3^*$	$N4^*$	$J(\theta^*)$	fr^*
3	12	43	36	8	13.56	0.14
5	31	14	33	19	14.14	0.16
10	13	20	15	51	17.98	0.18

Table 2: $\lambda_u = 1.0,\ D_b = 0$

T	$N1^*$	$N2^*$	$N3^*$	$N4^*$	$J(\theta^*)$	fr^*
3	58	31	21	17	23.46	0.42
5	10	16	11	2	25.89	0.45
10	51	10	27	7	28.22	0.47

Table 3: $\lambda_u = 0.4$, $T = 3$

D_b	$N1^*$	$N2^*$	$N3^*$	$N4^*$	$J(\theta^*)$	fr^*
0	38	42	29	11	16.96	0.25
3	51	32	59	39	17.61	0.26
6	3	44	53	13	18.35	0.27
9	42	59	27	7	18.76	0.29

Table 4: $T = 3,\ D_b = 0,\ \lambda_u = 0.4$

RC	$N1^*$	$N2^*$	$N3^*$	$N4^*$	$J(\theta^*)$	fr^*
40	21	26	31	7	14.90	0.28
50	38	42	29	11	16.96	0.25
60	23	36	15	23	21.40	0.24
70	27	25	9	10	23.00	0.23
80	23	41	44	59	26.65	0.22
90	17	40	41	16	28.56	0.20
100	4	53	58	7	32.09	0.19

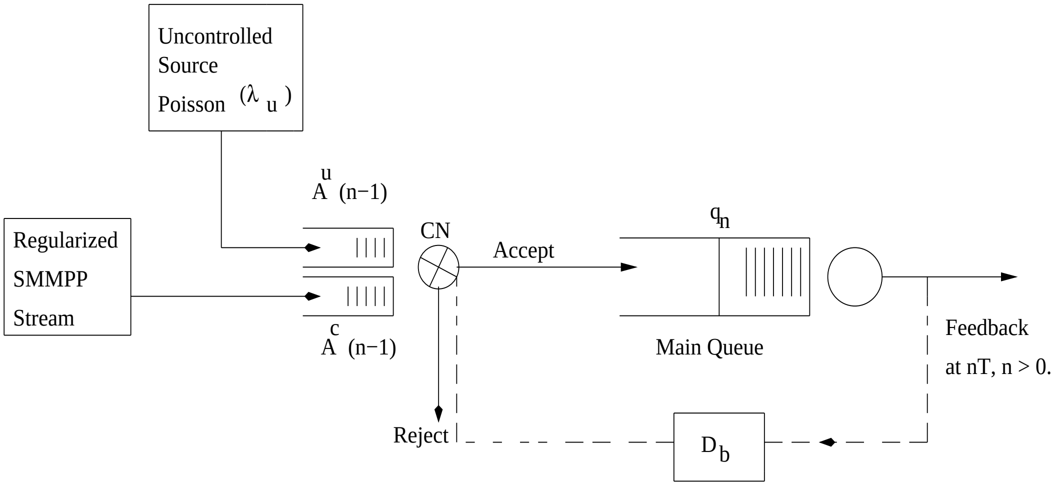

Figure 1: The Model

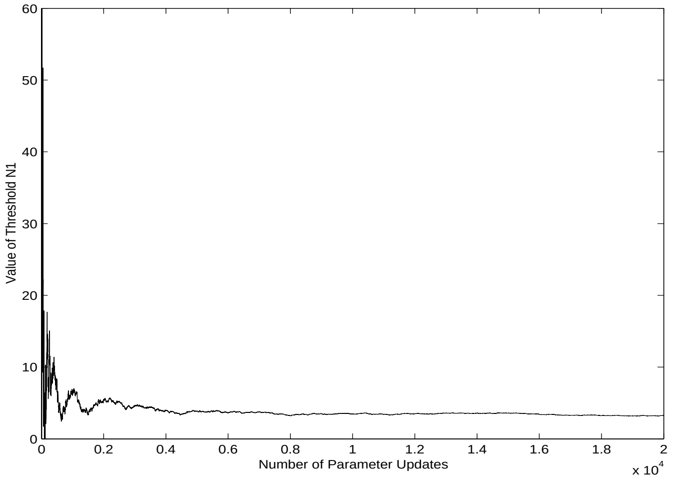

Figure 2: Convergence of the Values of Threshold $N1$ for $T = 3$, $D_b = 6$ and $\lambda_u = 0.4$

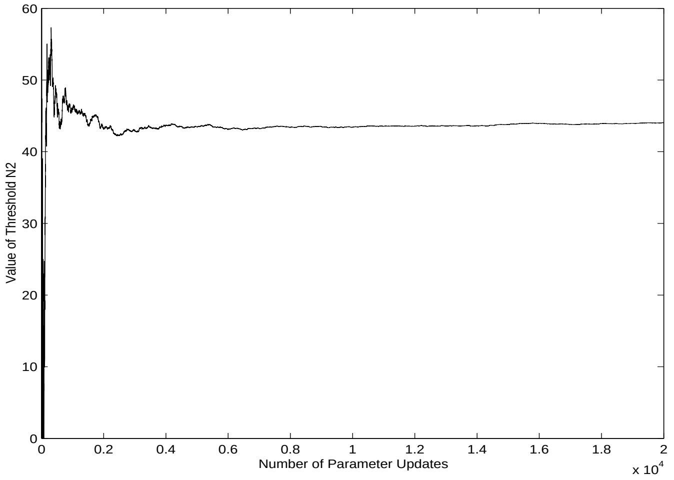

Figure 3: Convergence of the Values of Threshold $N2$ for $T = 3$, $D_b = 6$ and $\lambda_u = 0.4$

Appendix

We provide here proofs of the various results presented in this paper.

Proof of Theorem 1.1 Let us first consider the case $D_b = 0$. We consider here service times to be exponentially distributed according to $\exp(\mu)$ for some $\mu > 0$. For this case, the process $\{(q_n, X_{n-1})\}$ itself is Markov and is ergodic for every θ fixed, under the feedback policies considered here. Let $\mu_\theta(q, x)$ denote the stationary distribution of the process $\{(q_n, X_{n-1})\}$. Suppose for given θ, $p_\theta(q, x; q', x')$ represent the transition probabilities of this Markov process. Writing in vector - matrix notation, let $\mu(\theta) := [\mu_\theta(q, x)]$ and $P(\theta) :=$

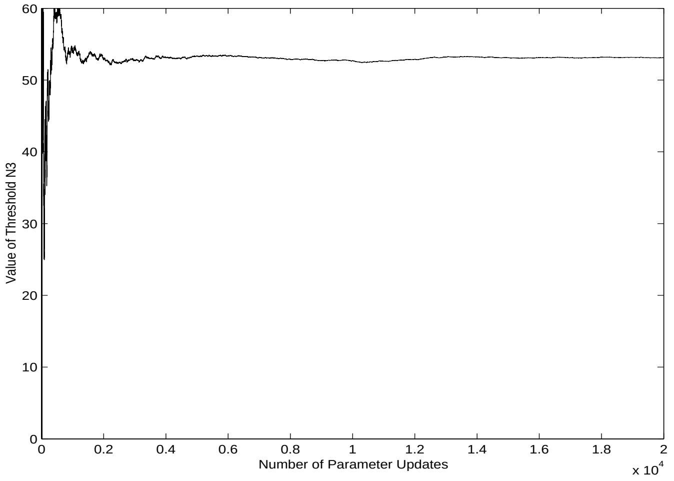

Figure 4: Convergence of the Values of Threshold $N3$ for $T = 3$, $D_b = 6$ and $\lambda_u = 0.4$

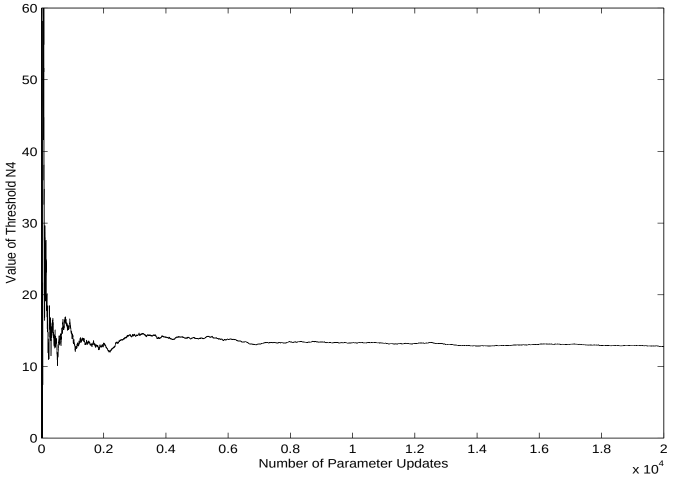

Figure 5: Convergence of the Values of Threshold $N4$ for $T = 3$, $D_b = 6$ and $\lambda_u = 0.4$

$[[p_\theta(q, x; q', x')]]$, respectively, denote the vector and matrix of stationary distribution and transition probabilities. Note that for $J(\theta)$ to be continuously differentiable, it is sufficient to show that $\mu_\theta(\cdot, \cdot)$ is continuously differentiable in θ. Let $Z(\theta) = [I - P(\theta) \; -P^\infty(\theta)]^{-1}$, where I is the identity matrix and $P^\infty(\theta) = \lim_{m\to\infty} (P(\theta) + P^2(\theta) + \cdots P^m(\theta))/m$. Then from Theorem 2, pp.402-403 of [27], we can write

$$\mu(\theta + h) = \mu(\theta)(I + (P(\theta + h) - P(\theta))Z(\theta)) + o(h)). \tag{A.1}$$

Thus,

$$\mu'(\theta) = \mu(\theta)P'(\theta)Z(\theta), \tag{A.2}$$

where $\mu'(\theta)$ and $P'(\theta)$ are the derivatives of $\mu(\theta)$ and $P(\theta)$, respectively. From (A.2) it is clear that $\mu'(\theta)$ exists if $P'(\theta)$ does. Let us assume for the moment that $P'(\theta)$ exists and is continuous. Then

$$|\mu'(\theta+h)-\mu'(\theta)| \le |\mu(\theta+h)P'(\theta+h)Z(\theta+h)-\mu(\theta)P'(\theta+h)Z(\theta+h)|$$

$$+|\mu(\theta)P'(\theta+h)Z(\theta+h)-\mu(\theta)P'(\theta)Z(\theta+h)|+|\mu(\theta)P'(\theta)Z(\theta+h)-\mu(\theta)P'(\theta)Z(\theta)|.$$

Now from Theorem 2, pp.402-403 of [27], one can write $Z(\theta+h)$ as

$$Z(\theta+h)=Z(\theta)H(\theta,\theta+h)-P^{\infty}(\theta)H(\theta,\theta+h)U(\theta,\theta+h)Z(\theta)H(\theta,\theta+h),$$

where

$$H(\theta,\theta+h)=[I-(P(\theta+h)-P(\theta))]^{-1}\to I \text{ as } |h|\to 0,$$

and

$$U(\theta,\theta+h)=(P(\theta+h)-P(\theta))Z(\theta)\to [0] \text{ as } |h|\to 0.$$

In the above, [0] is the matrix (of appropriate dimension) with all elements zero. It is now clear that $Z(\theta+h)\to Z(\theta)$ as $|h|\to 0$. Moreover from (A.1), $\mu(\theta)$ is continuous. Thus, $\mu'(\theta)$ is continuous in θ as well and the claim follows.

We finally need to show now that $P'(\theta)$ exists and is continuous in order to show that $\mu'(\theta)$ indeed exists. Note that

$$p_\theta(i,y;j,x)=Pr(q_{n+1}=j, X_n=x \mid q_n=i, X_{n-1}=y,\theta)$$

$$=Pr(q_{n+1}=j \mid q_n=i, X_n=x, X_{n-1}=y,\theta)Pr(X_n=x \mid q_n=i, X_{n-1}=y,\theta)$$

$$=Pr(q_{n+1}=j \mid q_n=i, X_n=x,\theta)\bar{p}(y;x).$$

In the above, $\bar{p}(y;x)$ is independent of θ. We now evaluate $Pr(q_{n+1}=j \mid q_n=i,\ X_n=x,\ \theta)$. For this, we consider various cases based on the form of the feedback policies. Let us fix $q_n=i$, $X_n=x$, $A^u_{n-1}=a^u$ and $A^c_{n-1}=a^c$, for some arbitrary $0\le i\le B$, $1\le j\le N$, $0\le A^u_{n-1}$, $A^c_{n-1}<\infty$. We now have the following cases:

1. $a^u \geq B - i$.

   Here

$$
\begin{aligned}
& Pr(q_{n+1} = j \mid q_n = i, X_n = x, \theta) \\
&= \sum_{a^u = B-i}^{\infty} Pr(D_n = B - j, A_{n-1}^u = a^u \mid q_n = i, X_n = x, \theta) \\
&= \sum_{a^u = B-i}^{\infty} Pr(D_n = B - j \mid q_n = i, X_n = x, A_{n-1}^u = a^u, \theta) \times \\
& \quad Pr(A_{n-1}^u = a^u \mid q_n = i, X_n = x, \theta) \\
&= \sum_{a^u = B-i}^{\infty} \frac{\exp(-\mu T)(\mu T)^{B-j}}{(B-j)} \frac{\exp(-\lambda_u T)(\lambda_u T)^{a^u}}{a^u}.
\end{aligned}
$$

2. $i < L_x$, $(B - i) > a^u \geq L_x - i$.

   There are two cases that follow here:

   - $j \leq i + a^u$.

$$
\begin{aligned}
& Pr(q_{n+1} = j \mid q_n = i, X_n = x, \theta) \\
&= \sum_{a^u = L_x - i}^{B-i-1} Pr(D_n = i + a^u - j, A_{n-1}^u = a^u \mid q_n = i, X_n = x, \theta) \\
&= \sum_{a^u = L_x - i}^{B-i-1} Pr(D_n = i + a^u - j \mid q_n = i, X_n = x, A_{n-1}^u = a^u, \theta) \times \\
& \quad Pr(A_{n-1}^u = a^u \mid q_n = i, X_n = x, \theta) \\
&= \sum_{a^u = L_x - i}^{B-i-1} \frac{\exp(-\mu T)(\mu T)^{i+a^u-j}}{(i + a^u - j)} \frac{\exp(-\lambda_u T)(\lambda_u T)^{a^u}}{a^u}.
\end{aligned}
$$

   - $j > i + a^u$.

$$
Pr(q_{n+1} = j \mid q_n = i, X_n = x, \theta) = 0.
$$

3. $i < L_x$, $L_x - i > a^u$.

   Two cases follow from this:

   - $a^c < L_x - i - a^u$.

     Two cases follow again from here:

- $j \leq i + a^u + a^c$.

$$
\begin{aligned}
& Pr(q_{n+1} = j \mid q_n = i, X_n = x, \theta) \\
&= \sum_{a^u=0}^{L_x-i-1} \sum_{a^c=0}^{L_x-i-a^u-1} Pr(A^c_{n-1} = a^c, A^u_{n-1} = a^u, D_n = i{+}a^u{+}a^c{-}j \mid q_n = i, X_n = x, \theta) \\
&= \sum_{a^u=0}^{L_x-i-1} \sum_{a^c=0}^{L_x-i-a^u-1} Pr(D_n = i{+}a^u{+}a^c{-}j \mid q_n = i, A^c_{n-1} = a^c, A^u_{n-1} = a^u, X_n = x, \theta) \\
&\qquad \times Pr(A^c_{n-1} = a^c \mid q_n = i, A^u_{n-1} = a^u, X_n = x, \theta) \times \\
&\qquad Pr(A^u_{n-1} = a^u \mid q_n = i, X_n = x, \theta) \\
&= \sum_{a^u=0}^{L_x-i-1} \sum_{a^c=0}^{L_x-i-a^u-1} \frac{\exp(-\mu T)(\mu T)^{i+a^u+a^c-j}}{(i + a^u + a^c - j)} \frac{\exp(-\lambda(x)T)(\lambda(x)T)^{a^c}}{a^c} \times \\
&\qquad \frac{\exp(-\lambda_u T)(\lambda_u T)^{a^u}}{a^u}.
\end{aligned}
$$

- $j > i + a^u + a^c$.

$$
Pr(q_{n+1} = j \mid q_n = i, X_n = x, \theta) = 0
$$

- $a^c \geq L_x - i - a^u$.

  Two cases follow again from here:

  - $j \leq L_x$.

$$
\begin{aligned}
& Pr(q_{n+1} = j \mid q_n = i, X_n = x, \theta) \\
&= \sum_{a^u=0}^{L_x-i-1} \sum_{a^c=L_x-i-a^u}^{\infty} Pr(A^c_{n-1} = a^c, A^u_{n-1} = a^u, D_n = L_x{-}j \mid q_n = i, X_n = x, \theta) \\
&= \sum_{a^u=0}^{L_x-i-1} \sum_{a^c=L_x-i-a^u}^{\infty} Pr(D_n = L_x{-}j \mid q_n = i, X_n = x, A^c_{n-1} = a^c, A^u_{n-1} = a^u, \theta) \\
&\qquad \times Pr(A^c_{n-1} = a^c \mid q_n = i, X_n = x, A^u_{n-1} = a^u, \theta) \times \\
&\qquad Pr(A^u_{n-1} = a^u \mid q_n = i, X_n = x, \theta) \\
&= \sum_{a^u=0}^{L_x-i-1} \sum_{a^c=L_x-i-a^u}^{\infty} \frac{\exp(-\mu T)(\mu T)^{L_x-j}}{(L_x - j)} \frac{\exp(-\lambda(x)T)(\lambda(x)T)^{a^c}}{a^c} \times \\
&\qquad \frac{\exp(-\lambda_u T)(\lambda_u T)^{a^u}}{a^u}.
\end{aligned}
$$

- $j > L_x$.

$$Pr(q_{n+1} = j \mid q_n = i, X_n = x, \theta) = 0$$

4. $i \geq L_x$ and $B - i > a^u$.

Two cases follow from here:

- $j \leq i + a^u$.

$$\begin{aligned} & Pr(q_{n+1} = j \mid q_n = i, X_n = x, \theta) \\ &= \sum_{a^u=0}^{B-i-1} Pr(A^u_{n-1} = a^u, D_n = i + a^u - j \mid q_n = i, X_n = x, \theta) \\ &= \sum_{a^u=0}^{B-i-1} Pr(D_n = i + a^u - j \mid q_n = i, X_n = x, A^u_{n-1} = a^u, \theta)\times \\ & Pr(A^u_{n-1} = a^u \mid q_n = i, X_n = x, \theta) \\ &= \sum_{a^u=0}^{B-i-1} \frac{\exp(-\mu T)(\mu T)^{i+a^u-j}}{(i + a^u - j)} \frac{\exp(-\lambda_u T)(\lambda_u T)^{a^u}}{a^u}. \end{aligned}$$

- $j > i + a^u$.

$$Pr(q_{n+1} = j \mid q_n = i, X_n = x, \theta) = 0.$$

It is now easy to see that for any given $q_n = i$, $X_{n-1} = y$, $q_{n+1} = j$ and $X_n = x$, the transition probabilities $p_\theta(i, y; j, x)$ are continuously differentiable in θ. The claim follows. Finally, an exactly similar argument as above works for the delayed case with $D_b > 0$, $D_b = MT$, for some integer $M > 0$. □

Proof of Corollary 1.1 We showed in Theorem 1.1 that the above quantities are continuously differentiable in θ. Hence they are continuous as well. □

We finally have

Proof of Theorem 2.1 The proof proceeds through a series of approximation steps as in [7]. We briefly sketch it below. Let us define a new sequence $\{\tilde{b}(n)\}$ of step-sizes according to $\tilde{b}(n) = b\left(\left[\frac{n}{L}\right]\right)$, where, $\left[\frac{n}{L}\right]$ denotes the integer part of $\frac{n}{L}$. It is then easy to see that $\tilde{b}(n)$ satisfies

$$\sum_n \tilde{b}(n) = \infty, \;\; \sum_n \tilde{b}(n)^2 < \infty, \;\; a(n) = o(\tilde{b}(n)).$$

In fact $b(n)$ goes to zero faster than $\tilde{b}(n)$ does and thus $\tilde{b}(n)$ corresponds to an even faster timescale than $b(n)$. Now define $\{t(n)\}$ according to $t(0) = 0$ and $t(n) = \sum_{m=1}^{n} \tilde{b}(m)$. It is then easy to see that the algorithm (2.4)-(2.6) on this timescale tracks the trajectories of the system of ODEs:

$$\dot{\theta}(t) = 0, \tag{A.3}$$

$$\dot{Z}^1(t) = J(\theta(t) - \delta\triangle(t)) - Z^1(t), \tag{A.4}$$

$$\dot{Z}^2(t) = J(\theta(t) + \delta\triangle(t)) - Z^2(t), \tag{A.5}$$

in the following manner: Suppose we define continuous time processes $\{X^1(t)\}$ and $\{X^2(t)\}$ according to $X^1(t(n)) = Z^1(nL)$, $X^2(t(n)) = Z^2(nL)$ and for $t \in [t(n), t(n+1)]$, $X^1(t)$ and $X^2(t)$ are continuously interpolated from the values they take at the boundaries of these intervals. Further, let us define a real-valued sequence $\{T_n\}$ as follows: Suppose $T > 0$ is a given constant. Then $T_0 = 0$ and for $n \geq 1$,

$$T_n = \min\{t(m) \mid t(m) \geq T_{n-1} + T\}.$$

Thus $T_n - T_{n-1} \approx T$, $\forall n \geq 1$. Also, for any n, there exists some integer m_n such that $T_n = t(m_n)$. Now define processes $\{X^{1,n}(t)\}$ and $\{X^{2,n}(t)\}$ according to $X^{1,n}(T_n) = X^1(t(m_n))$ $= Z^1(nL)$, $X^{2,n}(T_n) = X^2(t(m_n)) = Z^2(nL)$ and for $t \in [T_n, T_{n+1})$, $X^{1,n}(t)$ and $X^{2,n}(t)$ evolve according to the ODEs (A.4)-(A.5). Using Gronwall's inequality, it is easy to see that

$$\sup_{t \in [T_n, T_{n+1})} \parallel X^{1,n}(t) - X^1(t) \parallel, \sup_{t \in [T_n, T_{n+1})} \parallel X^{2,n}(t) - X^2(t) \parallel \rightarrow 0 \; n \rightarrow \infty.$$

Now note that iteration (2.6) of the algorithm can be written as: For $i = 1, \ldots, N$,

$$L_i(n+1) = \pi(L_i(n) + \tilde{b}(n)\eta_i(n)),$$

where $\eta_i(n) = o(1)$, $\forall i = 1, \ldots, N$, since $a(n) = o(\tilde{b}(n))$. Let us now define two continuous time processes $\{\theta(t)\}$ and $\{\hat{\theta}(t)\}$ as follows: $\theta(t(n)) = \theta(n) = (L_1(n), \ldots, L_N(n))^T$, $n \geq 1$. For $t \in [t(n), t(n+1)]$, $\theta(t)$ is continuously interpolated from the values it takes at the boundaries of these intervals. Also $\hat{\theta}(T_n) = \hat{\theta}(t(m_n)) = \theta(n) = (L_1(n), \ldots, L_N(n))^T$ and for

$t \in [T_n, T_{n+1})$, $\hat{\theta}(t)$ evolves according to the ODE (A.3). Thus given $T, \eta > 0$, $\exists M$ such that $\forall n \geq M$, $\sup_{t \in [T_n, T_{n+1})} \parallel X^{i,n}(t) - X^i(t) \parallel < \eta$, $i = 1, 2$. Also $\sup_{t \in [T_n, T_{n+1})} \parallel \hat{\theta}(t) - \theta(t) \parallel < \eta$. It is easy to see (cf. [18]) that $\parallel Z_n^1 - J(\theta(n) - \delta\triangle(n)) \parallel$ and $\parallel Z_n^2 - J(\theta(n) + \delta\triangle(n)) \parallel \rightarrow 0$ as $n \rightarrow \infty$.

Now note that because of the above, the iteration (2.6) of the algorithm can be written as follows: For $i = 1, \ldots, N$,

$$L_i(n+1) = \pi\left(L_i(n) + a(n)\left(\frac{J(\theta(n) - \delta\triangle(n)) - J(\theta(n) + \delta\triangle(n))}{2\delta\triangle_i(n)}\right) + a(n)\xi(n)\right),$$

where $\xi(n) = o(1)$. Using Taylor series expansions of $J(\theta(n) - \delta\triangle(n))$ and $J(\theta(n) + \delta\triangle(n))$ around the point $\theta(n)$, one obtains

$$J(\theta(n) - \delta\triangle(n)) = J(\theta(n)) - \delta\sum_{j=1}^{N} \triangle_j(n)\nabla_j J(\theta(n)) + o(\delta)$$

and

$$J(\theta(n) + \delta\triangle(n)) = J(\theta(n)) + \delta\sum_{j=1}^{N} \triangle_j(n)\nabla_j J(\theta(n)) + o(\delta),$$

respectively. Thus,

$$\frac{J(\theta(n) - \delta\triangle(n)) - J(\theta(n) + \delta\triangle(n))}{2\delta\triangle_i(n)} = -\nabla_i J(\theta(n))$$

$$- \sum_{j=1, j\neq i}^{N} \frac{\triangle_j(n)}{\triangle_i(n)} \nabla_j J(\theta(n)) + o(\delta). \tag{A.6}$$

Now let $\{\mathcal{F}_n,\ n \geq 0\}$ denote a sequence of σ-fields defined by $\mathcal{F}_n = \sigma(\theta(m),\ m \leq n,\ \triangle(m),\ m < n)$, with $\triangle(-1) = 0$. It is now easy to see that the processes $\{M_1(n)\}, \ldots, \{M_N(n)\}$ defined by

$$M_i(n) = \sum_{m=0}^{n-1} a(m)(\frac{J(\theta(m) - \delta\triangle(m)) - J(\theta(m) + \delta\triangle(m))}{2\delta\triangle_i(m)}$$

$$-E[\frac{J(\theta(m) - \delta\triangle(m)) - J(\theta(m) + \delta\triangle(m))}{2\delta\triangle_i(m)} \mid \mathcal{F}_m]),$$

$i = 1, \ldots, N$, form convergent martingale sequences because of (2.1). From (A.6), we have

$$E[\frac{J(\theta(n) - \delta\triangle(n)) - J(\theta(n) + \delta\triangle(n))}{2\delta\triangle_i(n)} \mid \mathcal{F}_n]$$

$$= -\nabla_i J(\theta(n)) - \sum_{j=1, j\neq i}^{N} E[\frac{\triangle_j(n)}{\triangle_i(n)}] \nabla_j J(\theta(n)) + o(\delta).$$

By Condition (B), $E[\frac{\triangle_j(n)}{\triangle_i(n)}] = 0 \ \forall j \neq i,\ n \geq 0$. The iteration (2.6) of the algorithm can now be written as follows: For $i = 1, \ldots, N$,

$$L_i(n+1) = \pi(L_i(n) - a(n)\nabla_i J(\theta(n)) + a(n)\beta(n)), \tag{A.7}$$

where $\beta(n) = o(1)$ by the above. Thus (A.7) can be seen to be a discretization of the ODE (2.7) except for some additional terms which however vanish asymptotically. The claim follows. □