Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computer Communications 28 (2005) 1938–1946

computer communications

www.elsevier.com/locate/comcom

Robust delay estimator for playout buffering in Internet audio applications

Kyung-Joon Park*, Eun-Chan Park, Chong-Ho Choi

School of Electrical Engineering and Computer Science ENG420-047, Seoul National University, Kwanak-Gu, Seoul 151-742, South Korea

Received 25 March 2003; revised 1 March 2005; accepted 16 March 2005
Available online 19 April 2005

Abstract

In this paper, we propose a robust delay estimation scheme for Internet multimedia applications. The proposed scheme adopts an autoregressive (AR) model for the delay process. The adopted AR model is a natural generalization of the Jacobson's algorithm in TCP retransmission timeout (RTO) estimation. However, unlike fixed parameter values in the Jacobson's algorithm, the parameters of the proposed scheme adapt themselves to the estimation error via a robust identification algorithm. This robust identification algorithm usually leads to better performance when the noise is correlated and/or non-stationary, and it is also more robust to modeling uncertainties. As an application of the proposed algorithm, we consider the problem of delay prediction for audio playout buffering. We rigorously formulate the proposed scheme in the realm of multimedia audio mechanisms and give simulation result which shows its effectiveness.

Keywords: Delay estimation; Playout buffering; Autoregressive model; Robust control

1. Introduction

Recently, there has been a significant increase of interest in Internet multimedia applications such as video conferencing, Internet telephony, video-on-demand, etc. These kinds of applications require a certain level of quality of service (QoS), such as throughput, packet loss, delay, and delay jitter. However, the current Internet multimedia applications commonly employ the UDP transport mechanism, which is not capable of congestion control. Consequently, QoS is usually provided by application-level end-user adaptation [1–3].

In the application-level, the problem of traffic modeling and estimation at the end users is crucial for providing QoS [4]. One of the prevalent applications is the prediction of packet delay for audio playout mechanisms [5]. In Internet audio applications such as real-time voice communication and packet radio service, delay and delay jitter are the most critical factors for QoS. In order to alleviate the problem caused by unpredictability of delay and delay jitter in the Internet, the receiver usually needs to buffer some amount of packets before it actually plays them.

Fig. 1 shows a sample sequence of operations on a sender and a receiver of an audio application. Two solid lines represent the sequence numbers of audio packets sent and received, respectively, and two dotted lines represent the sequence number of packets played by the receiver. Delay jitters during transmission cause uneven arrivals of packets at the receiver. In order to reduce such delay jitters, a receiver delays the initiation of playout of received packets. Here, t_1 and t_2 represent two different playout times. In case of t_1, some of the packets are delivered to the receiver after their playout time, therefore, they are dropped without playing. In case of t_2, the receiver has larger delayed startup time (i.e., t_2-t_0) than that in case of t_1 (i.e., t_1-t_0). Consequently, the main objective of playout algorithms is to choose a small startup time which can also keep acceptable packet loss rate. Small buffering delay cannot tolerate severe delay jitter and also leads to significant packet loss, whereas large buffering delay causes large startup delay. Therefore, the amount of buffered packets and timing of playout are very important for the performance of audio applications.

The basic algorithm of delay prediction used in audio conferencing tools such as NeVot 1.4 [6] has been

* Corresponding author. Tel.: +82 11 9736 6040; fax: +82 2 885 4459.
E-mail addresses: kjpark@csl.snu.ac.kr (K.-J. Park), ecpark@csl.snu.ac.kr (E.-C. Park), chchoi@csl.snu.ac.kr (C.-H. Choi).

0140-3664/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2005.03.001

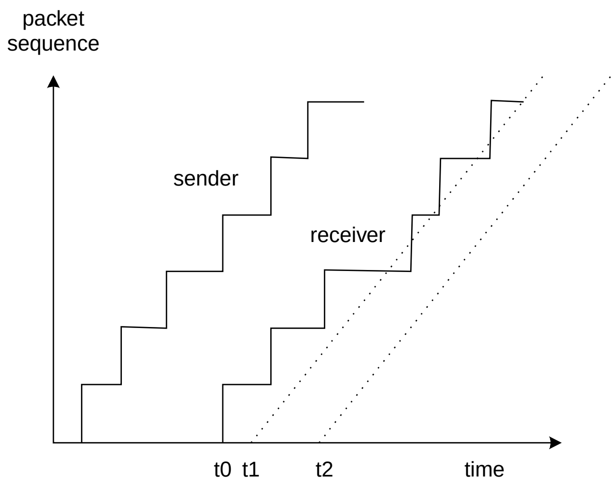

Fig. 1. Playout jitter problem.

influenced by RFC 793 TCP retransmission timeout (RTO) mechanism [7]. The estimate of average packet delay $\hat{d}_i$ is as follows:

$$\hat{d}_i = \alpha\hat{d}_{i-1} + (1-\alpha)d_i \tag{1}$$

where d_i is the delay suffered by the ith packet in the network, and $\boldsymbol{\alpha}$ is a weighting factor which controls the rate of convergence of the algorithm. The variation in this delay, $\hat{v}_i$ is estimated by

$$\hat{v}_i = \alpha\hat{v}_{i-1} + (1-\alpha)|d_i - \hat{d}_i|. \tag{2}$$

This is used to fix the total end-to-end delay (ted) for playing out the next packet as follows:

$$\text{ted} = \hat{d}_i + \beta\hat{v}_i \tag{3}$$

where β is called a safety factor used to guarantee that the estimated delay is larger than the actual delay with a high probability. The most significant shortcoming of this basic algorithm is that it does not adapt to network traffic. Once the value of $\boldsymbol{\alpha}$ is given, the model is fixed and this can degrade estimation performance. It does not adapt itself efficiently, especially when there is an abrupt change in network environment.

Many researchers have suggested modified algorithms for better performance [1,5,8–11]. However, most of the proposed schemes were derived not from theoretical basis, but from some kind of heuristics. Recently, DeLeon and Sreenan [12] have proposed a scheme which used a simple normalized least-mean-square (NLMS) algorithm in adaptive filter theory [13]. In [11], the authors proposed a heuristic algorithm for delay boundary prediction, which considered the second-order statistics of the delay process.

In this paper, we introduce a robust delay estimation algorithm based on the development in robust control [14]. Preliminary result was presented as a position paper in [15]. Here, we give a more detailed and rigorous treatment on the estimation methodology we adopted and also present more comprehensive simulation results for performance evaluation. The scheme is not only adaptive in its nature, but also robust to non-stationary noise and modeling uncertainties: the proposed scheme is adaptive in the sense that the parameters of the process is adaptively updated according to the estimation error. Furthermore, it is robust because of the criterion we used, which is of a H^∞ type [16]. Qualitatively speaking, the objective of the H^∞ criterion is to find the optimal estimation policy for any unknown noise with finite energy. Hence, even in case we do not have knowledge on noise and uncertainties, still we can guarantee a pre-specified performance bound of the algorithm.

We will also show that most of the existing schemes correspond to special cases of the proposed scheme. Note that our aim here is not to propose an application-specific algorithm, rather to introduce a more general estimation algorithm, which is optimal without knowledge of the noise and can be used efficiently for various problems. We expect that the proposed scheme can be applied to many estimation problems in the Internet applications. The rest of the paper is organized as follows. In Section 2, we will formulate the delay process as an autoregressive (AR) model and give a design criterion which is of H^∞ type. In Section 3, we derive the update rules for the estimate of $\boldsymbol{\alpha}$ and the estimate of the variance σ^2. We give simulation results in Section 4. Finally, conclusion and future work follow in Section 5.

2. Problem formulation

In this section, we rigorously formulate the problem of delay estimation in Internet applications. Here, the delay could be either round-trip-time (RTT) as in TCP retransmission timeout (RTO) estimation or end-to-end delay as in playout buffering problem of Internet audio/video applications. We adopt an autoregressive (AR) model for the delay process. AR model is a natural generalization of the model used in TCP RTO algorithm [7].

Here, we adopt the methodology introduced in [17]. This methodology is a discrete-time version of the algorithm in [14]. The AR model we adopt is as follows:

$$d_{n+1} = \sum_{i=1}^{p} \alpha_i d_{n+1-i} + \phi_n \tag{4}$$

where d_i is the ith packet delay, $\boldsymbol{\alpha}_i$s are parameters of the AR process to be identified, and ϕ_i is an unknown noise sequence. Here, p is the order of the AR process. Note that the complexity and the computational burden of the algorithm is determined by p. The AR model (4) is a natural generalization of (1). Furthermore, while the basic algorithm (1) uses a fixed weighting factor $\boldsymbol{\alpha}$, the proposed algorithm updates the parameters $\boldsymbol{\alpha}_i$s via the robust identification method as presented in Section 3.

We can express (4) with the following vector notation:

$$\boldsymbol{d}_{n+1} = \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{d}_n + \phi_n \tag{5}$$

where $\boldsymbol{\alpha} := (\alpha_1,\ldots,\alpha_p)^{\mathrm{T}}$ and $\boldsymbol{d}_n := (d_n,\ d_{n-1},\ldots,d_{n-p+1})^{\mathrm{T}}$. What we are going to do is to identify $\boldsymbol{\alpha}$ based on the previous data, i.e., $\boldsymbol{d}_i$, $i=1,2,\ldots,n$.

We wish to obtain a sequence of estimates for $\boldsymbol{\alpha}$, denoted by $\hat{\boldsymbol{\alpha}}_n$ at step n, so that $\hat{\boldsymbol{\alpha}}_n$ would depend on all the past and the present values of d_i, i.e., $\hat{\boldsymbol{\alpha}}_n = \hat{\boldsymbol{\alpha}}(d_n, d_{n-1}, \ldots, d_0)$. The criterion J to be minimized is of the H^∞ type [16], which is the gain from the energy of the unknowns to a weighted quadratic identification error as follows:

$$J(\{\hat{\boldsymbol{\alpha}}_n\}_{n=0}^{\infty}) := \sup_{\{\phi_n\}_{n=0}^{\infty},\bar{\boldsymbol{\alpha}}_0} \frac{\sum_{n=0}^{\infty}(\boldsymbol{\alpha}-\hat{\boldsymbol{\alpha}}_n)^{\mathrm{T}} Q_n (\boldsymbol{\alpha}-\hat{\boldsymbol{\alpha}}_n)}{\sum_{n=0}^{\infty}|\phi_n|^2 + (\boldsymbol{\alpha}-\bar{\boldsymbol{\alpha}}_0)^{\mathrm{T}}\bar{Q}_0(\boldsymbol{\alpha}-\bar{\boldsymbol{\alpha}}_0)} \tag{6}$$

where $\bar{\boldsymbol{\alpha}}_0$ is some initial estimate for $\boldsymbol{\alpha}$, $\bar{Q}_0 > 0$ is a fixed weighting matrix, and $Q_n > 0$, $n=0,1,\ldots$ is a sequence of weighting matrices which will be specified later.

An interpretation of the H^∞ criterion J in (6) is as follows: the denominator of the ratio in (6) can be regarded as the energy of disturbances, i.e. the unknown noise sequence $\{\phi_n\}_{n=0}^{\infty}$ and the initial estimation error of $\boldsymbol{\alpha}$, and the numerator can be regarded as the energy of the estimation error sequence. Thus, the ratio in (6) is the energy gain from the disturbances to the estimation errors. Note that a desirable estimator is one for which this energy gain is small so that the disturbances are attenuated, and a non-desirable estimator is one for which the energy gain is large since the disturbances are amplified. The H^∞ framework [16] proposes to choose an estimator such that the worst-case energy gain is bounded by the prescribed value γ^2, i.e., H^∞ estimators yield an energy gain less than γ^2 for all energy-bounded disturbances, no matter what they are. The robustness of H^∞ estimators comes from this fact. Consequently, our objective is to find a sequence $\hat{\boldsymbol{\alpha}}^* := \{\hat{\boldsymbol{\alpha}}_n^*\}_{n=0}^{\infty}$ which minimizes J in (6), i.e.

$$J(\hat{\boldsymbol{\alpha}}^*) = \inf_{\hat{\boldsymbol{\alpha}}=\{\alpha_n\}_{n=0}^{\infty}} J(\hat{\boldsymbol{\alpha}}) =: (\gamma^*)^2 \tag{7}$$

and $\lim_{n\uparrow\infty}\hat{\boldsymbol{\alpha}}_n^* = \boldsymbol{\alpha}$.

3. Robust identification algorithm

3.1. Update rule for estimation of $\boldsymbol{\alpha}$

Now, from [17], we derive update rules for the estimate of $\boldsymbol{\alpha}$. By letting the right-hand side of (6) as γ^2, we can have the following soft-constrained objective function parameterized by γ:

$$L_\gamma = \sum_{i=0}^{n}|\alpha-\hat{\alpha}_i|_{Q_i}^2 - \gamma^2\sum_{i=0}^{n}|\phi_i|^2 - \gamma^2|\alpha-\hat{\alpha}_0|_{\bar{Q}_0}^2. \tag{8}$$

Now, we obtain a parameterized family of identifiers $\hat{\boldsymbol{\alpha}}^\gamma := \{\hat{\boldsymbol{\alpha}}_n^\gamma\}_{n=0}^{\infty}$ such that $\hat{\boldsymbol{\alpha}}^\gamma$ achieves the upper bound for L_γ. Here, the maximizer is $\{\phi_n\}_{n=0}^{\infty}$ and the minimizer is $\{\hat{\boldsymbol{\alpha}}_n\}_{n=0}^{\infty}$. What we want to do is to minimize L_γ by adjusting $\{\hat{\boldsymbol{\alpha}}_n\}_{n=0}^{\infty}$ when there is no information on $\{\phi_n\}_{n=0}^{\infty}$. If, for a given value of γ, the upper value of L_γ for an identifier $\hat{\boldsymbol{\alpha}}^\gamma := \{\hat{\boldsymbol{\alpha}}_n^\gamma\}_{n=0}^{\infty}$ can achieve zero, then we have $J(\hat{\boldsymbol{\alpha}}^\gamma) \le \gamma$. Intuitively, this implies that the total estimation error will be less than or equal to the noise energy multiplied by γ. Robustness of the proposed algorithm can be explained in this sense. We get the following update rule for $\hat{\boldsymbol{\alpha}}_n^\gamma$, denoted by $\hat{\boldsymbol{\alpha}}_n$ for simplicity.

$$\hat{\boldsymbol{\alpha}}_{n+1} = \hat{\boldsymbol{\alpha}}_n + (\Sigma_{n+1} + \boldsymbol{d}_n\boldsymbol{d}_n^{\mathrm{T}})^{-1}(d_{n+1} - \hat{\boldsymbol{\alpha}}_n^{\mathrm{T}}\boldsymbol{d}_n)\boldsymbol{d}_n, \quad \hat{\boldsymbol{\alpha}}_0 = \boldsymbol{\alpha}_0 \tag{9}$$

where Σ_n is a sequence of $p\times p$-dimensional positive-definite matrices, which is updated as follows:

$$\Sigma_{n+1} = \Sigma_n + \boldsymbol{d}_{n-1}\boldsymbol{d}_{n-1}^{\mathrm{T}} - \frac{1}{\gamma^2}Q_n, \quad \Sigma_1 = \bar{Q}_0 - \frac{1}{\gamma^2}Q_0. \tag{10}$$

Further, we need the following technical condition for the convergence of the algorithm.

Condition 1. Persistency of excitation: $\lim_{n\uparrow\infty}\lambda_{\min}\left(\sum_{i=0}^{n}Q_i\right) = \infty$ where $\lambda_{\min}$ denotes the minimum eigenvalue of a matrix. The above persistency of excitation (PE) condition makes it possible to identify the uncertain parameter $\boldsymbol{\alpha}$ with bounded disturbances [18].

More rigorously, we have the following theorem for our algorithm.

Theorem 1. *Consider the identification problem of* (7), *and let* Condition 1 *be satisfied, then* γ^* *is finite, and for all* $\gamma > \gamma^*$ *we have the following*:

(1) *The matrices* Σ_n, $n=1,2,\ldots$ *in* (10) *are positive definite.*
(2) *There exists an estimator that achieves the upper value (zero) in* (7), *which is given by* $\hat{\boldsymbol{\alpha}}_n^\gamma = \hat{\boldsymbol{\alpha}}_n$, $n=1,2,\ldots$ *where the sequence* $\{\hat{\boldsymbol{\alpha}}_n\}$ *is generated by* (9).
(3) *For all noise sequences which belong to* l_2, *if* Condition 1 *is satisfied, then the sequence* $\{\hat{\boldsymbol{\alpha}}_n^\gamma\}$ *converges to the true value of* $\boldsymbol{\alpha}$, *i.e.,* $\lim_{n\uparrow\infty}\hat{\alpha}_n^\gamma = \alpha$.

Proof. See Appendix A.

We further simplify the estimator by choosing appropriate weight matrices Q_i, $i=0,1,2,\ldots$. If we let $\bar{Q}_0 = Q_0 = I_p$, where I_p is the $p\times p$ identity matrix, and $Q_n = \boldsymbol{d}_{n-1}\boldsymbol{d}_{n-1}^{\mathrm{T}}$, then $\gamma^* = 1$ in (7) and we get the following update rule for Σ_n:

$$\Sigma_{n+1} = \Sigma_n + (1-\gamma^{-2})\boldsymbol{d}_{n-1}\boldsymbol{d}_{n-1}^{\mathrm{T}}, \quad \Sigma_1 = (1-\gamma^{-2})I_p. \tag{11}$$

Here, γ should be larger than 1 from Theorem 1.

In this case, we have the following update rule for $\hat{\boldsymbol{\alpha}}_n$:

$$\hat{\boldsymbol{\alpha}}_{n+1} = \hat{\boldsymbol{\alpha}}_n + (\Sigma_{n+1} + \boldsymbol{d}_n\boldsymbol{d}_n^{\mathrm{T}})^{-1}(d_{n+1} - \hat{\boldsymbol{\alpha}}_n^{\mathrm{T}}\boldsymbol{d}_n)\boldsymbol{d}_n, \quad \hat{\boldsymbol{\alpha}}_0 = \boldsymbol{\alpha}_0. \tag{12}$$

Remark 1. The estimator (12) is a generalized LMS filter [14]. Hence, the NLMS estimator in [12] corresponds to a special case of the proposed algorithm. Furthermore, if we let $\gamma \uparrow \infty$, the proposed algorithm will be precisely the least-squares (LS) estimator [14,17]. In general, the proposed estimator ranges from the LS estimator to the generalized LMS estimator for certain choices of γ.

Remark 2. The proposed algorithm has been derived based on H^∞ optimal control theory [16]. It usually shows better performance when the noise is correlated and/or non-stationary, and it is also more robust to modeling uncertainties. Consequently, unlike the NLMS estimator, the proposed algorithm does not require pre-filtering methods such as Discrete Wavelet Transform (DWT) in [9] for decorrelation.

3.2. Update rule for estimation of the variance

Here, we obtain an estimate of the variance σ^2 of ϕ_n. We need the variance σ^2 when we calculate the playout time as in (3). First, we define the autocorrelation function estimator at step n as follows:

$$\hat{R}^n[k] = \frac{1}{n-k+1}\sum_{i=k}^{n} d_i d_{i-k} \tag{13}$$

where $k=0,1,\ldots,p$. Then we have the following recursive relation:

$$\hat{R}^{n+1}[k] = \frac{n-k+1}{n-k+2}\hat{R}^n[k] + \frac{1}{n+2} d_{n+1}d_{n+1-k}. \tag{14}$$

Now we have the following equation for the estimate of variance σ_n^2:

$$\hat{\sigma}_n^2 = \hat{R}^n[0] - \sum_{i=1}^{p} \hat{l}_n^i \hat{R}^n[i]. \tag{15}$$

Here, $\hat{l}_n^i$ is the ith element of $\hat{l}_n$, where $\hat{l}_n$ is the LS estimate of $\boldsymbol{\alpha}$ at time n. Note that $\hat{l}_n$ is a p-dimensional vector. $\hat{l}_n$ can be obtained as follows:

$$\hat{l}_{n+1} = \hat{l}_n + (d_{n+1} - \hat{l}_n^{\mathrm{T}}\boldsymbol{d}_n)\left(\sum_{i=0}^{n}\boldsymbol{d}_i\boldsymbol{d}_i^{\mathrm{T}} + I_p\right)^{-1}\boldsymbol{d}_n. \tag{16}$$

Though (15) is useful for calculating the variance, we do not need to use (15) if we already have estimated $\boldsymbol{\alpha}$. Instead, we can simply obtain an estimate of σ^2 as follows:

$$\hat{\sigma}_n^2 = \frac{1}{n+1}\sum_{i=0}^{n}(d_{i+1} - \hat{\boldsymbol{\alpha}}_i^{\mathrm{T}}\boldsymbol{d}_i)^2. \tag{17}$$

Hence, we have the following recursive equation for $\hat{\sigma}_n^2$:

$$\hat{\sigma}_{n+1}^2 = \frac{n+1}{n+2}\hat{\sigma}_n^2 + \frac{1}{n+2}(d_{n+1} - \hat{\boldsymbol{\alpha}}_n^{\mathrm{T}}\boldsymbol{d}_n)^2. \tag{18}$$

4. Simulation

In this section, we give simulation results of the proposed algorithm applied to the Internet audio playout buffering problem. Note that, even though we apply the algorithm to the audio playout buffering, the proposed algorithm will work for many kind of delay estimation/prediction problems where the delay can be modeled as an AR process.

Here, for performance comparison of the proposed algorithm with existing schemes for the playout buffering, we have adopted the notion of the variation as in (2) instead of the variance in (18). Note that the variation of the proposed scheme is different from those of other algorithms since the estimate of $\boldsymbol{\alpha}$ is different. γ in (11) is set to 1.5.

The proposed robust playout algorithm can be summarized as follows:

For each talkspurt,
  For every i-th packet,
    Calculate $\hat{\alpha}_i$ and $\hat{v}_i$ by (12) and (2), respectively
  End
  Next playout time $t_p = \hat{d}_{n+1} + \beta\hat{v}_n$, where $\hat{d}_{n+1} = \hat{\boldsymbol{\alpha}}_n^{\mathrm{T}}\boldsymbol{d}_n$.
End

We compare the proposed algorithm with the basic algorithm of (1), the spike-detecting algorithm [5], and the NLMS-filter based algorithm [12] for six traces in [5,8]. We used $\boldsymbol{\alpha}=0.99802$ for the basic algorithm and $\beta=4.0$ for the variation as in [5] and we set $p=2$ for the AR model in (4).

4.1. Comparison of the delay estimation

First, we compare the delay estimation of the algorithms. Figs. 2–5 show the estimation error of the basic algorithm, the spike-detecting algorithm, the NLMS algorithm,

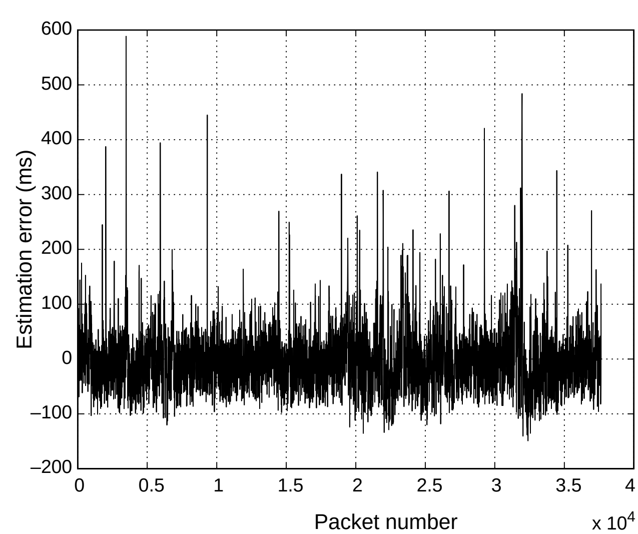

Fig. 2. Estimation error of the basic algorithm.

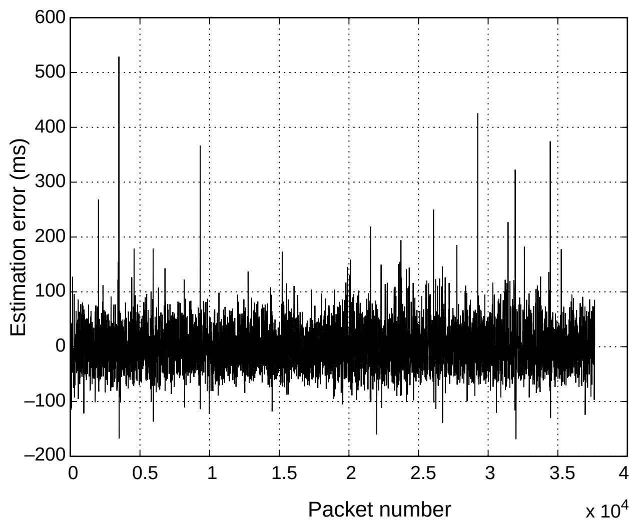

Fig. 3. Estimation error of the spike-detecting algorithm.

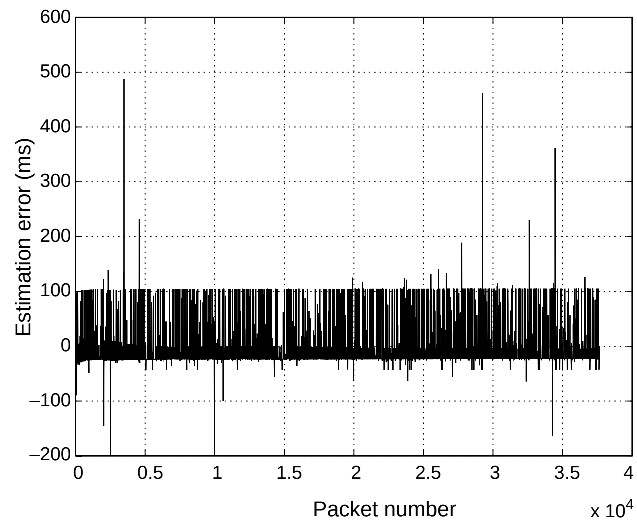

Fig. 5. Estimation error of the proposed algorithm.

and the proposed algorithm, respectively. From the figures, we can see that the spike-detecting, NLMS, and the proposed algorithms show better performance than the basic algorithm. However, the spike-detecting algorithm shows a biased behavior as we can see from Table 1. The spike-detecting algorithm can be regarded as a hybrid algorithm, which consists of two basic algorithms with different parameter values. The key idea of the algorithm is to choose one adaptively between these two different algorithms according to the difference between the current delay and the previous delay. If the difference between the current and the previous delay is larger than a pre-specified value, then this difference will be reflected fully instead of using a low-pass filter and hence, the estimation error will not exceed this pre-specified value. Consequently, the estimation error of the spike-detecting algorithm will remain in a pre-specified region. In this way, the algorithm can detect the spike, i.e., abrupt change of delay and adapt itself more efficiently to delay variation. This algorithm provides a very competitive performance with regard to the estimation error as can be seen from Fig. 3. In Table 1, the mean errors of Traces 1, 2, and 4 of the spike-detecting algorithm are very large compared to those of other algorithms. This indicates that the spike-detecting algorithm is a biased estimator due to the heuristic combination of two algorithms. Also, in the spike-detecting algorithm, there are several parameters that should be tuned by a trial-and-error procedure. In short, even though the spike-detecting algorithm is competitive in estimation of delay, it has several factors which should be carefully tuned beforehand according to the network environment.

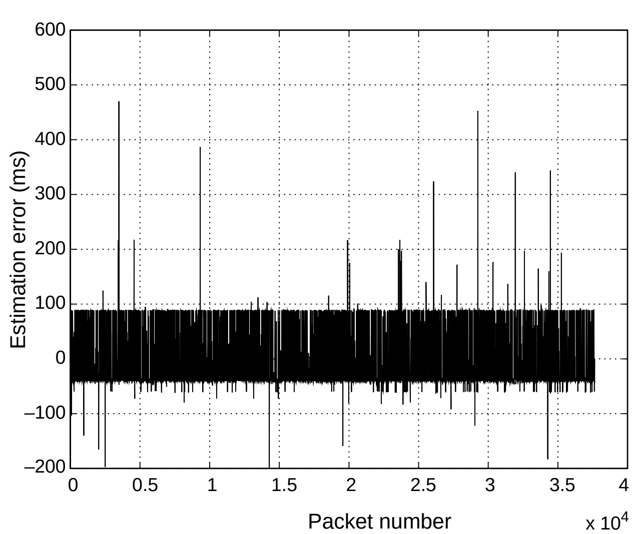

Fig. 4. Estimation error of the NLMS algorithm.

Now let us investigate the performance of NLMS and the proposed algorithm in Figs. 4 and 5. As we can see from the figures, these two algorithms show very competitive performance compared to the basic algorithm. To compare the algorithms more quantitatively, we calculate the mean and the variance of the estimation error for each algorithm. In Table 1, we can see that the variance of the proposed algorithm is the smallest among those of three algorithms in most cases. This is evident if we think of (6). Also, in all cases, the proposed algorithm outperforms NLMS with the same value of β. Our algorithm is an optimal estimator for any unknown noises in the average sense that it minimizes the infinite-time-horizon energy of the estimation error and consequently, the variance of the estimation error. Note that the optimality of the proposed algorithm is of the infinite-time-horizon sense, and hence the performance could be not optimal for some specific finite traces, which can be actually seen for Trace 5 in Table 1. However, still we can expect that the proposed algorithm shows a very competitive performance in almost all cases and further its robustness is guaranteed via the H^{∞} criterion.

Table 1
Mean and standard deviation of estimation errors

Trace #	Basic		Spike		NLMS		Robust	
1	−0.826	230.6	18.65	66.34	0.037	91.54	−0.009	66.34
2	−24.84	229.9	−53.33	73.88	−0.002	89.11	0.050	67.11
3	−1.464	61.83	−0.012	49.11	0.018	59.95	−0.009	45.32
4	−3.387	67.60	−18.74	26.01	0.008	33.29	0.033	25.49
5	0.139	16.43	0.001	15.59	−0.002	19.65	0.0002	15.76
6	12.96	111.1	0.315	53.46	0.123	46.20	0.086	36.47

mean (ms)|std (ms).

4.2. Delay-loss performance

Here, we compare the performance of the proposed algorithm with that of the spike-detecting algorithm more thoroughly. We performed simulations with β as a control parameter as in [5,8,9] and calculated the average playout delay and the loss percentage. The range of values of β varies from 1 to 20 in simulations. The spike-detecting algorithm was proposed in [5] and showed better performance than the basic algorithm [5,8]. Since, the proposed algorithm outperforms NLMS with the same value of β, we did not include NLMS in the simulations. We concentrated on the region of low loss percentage below 10% because the meaningful value of loss percentage is around 5% and below. Figs. 6–11 show the delay-loss performance for each trace, respectively. As we can see from the figures, the proposed algorithm outperforms the spike-detecting algorithm in most cases. In the region of very low loss percentage, say, below 1%, performances of the two algorithms are not so much different. However, the objective of a playout buffering algorithm is to reduce the startup delay with guaranteed loss percentage of 5% or below. Hence, from the figures, we can conclude that the proposed algorithm would be more appropriate for the playout buffering problem.

4.3. Effect of the parameter γ

Now we consider the effect of the parameter γ. As we have already mentioned, our estimator $\{\hat{\boldsymbol{\alpha}}_n\}_{n=0}^{\infty}$ in (12) can guarantee a performance bound of γ, i.e., the objective function J in (6) satisfies the following:

$$J(\{\hat{\boldsymbol{\alpha}}_n\}_{n=0}^{\infty}) = \sup_{\{\phi_n\}_{n=0}^{\infty},\bar{\boldsymbol{\alpha}}_0} \frac{\sum_{n=0}^{\infty}(\boldsymbol{\alpha}-\hat{\boldsymbol{\alpha}}_n)^{\mathrm{T}} Q_n (\boldsymbol{\alpha}-\hat{\boldsymbol{\alpha}}_n)}{\sum_{n=0}^{\infty}|\phi_n|^2 + (\boldsymbol{\alpha}-\bar{\boldsymbol{\alpha}}_0)^{\mathrm{T}} \bar{Q}_0 (\boldsymbol{\alpha}-\bar{\boldsymbol{\alpha}}_0)} \le \gamma.$$

If we consider the initial estimation error as an initial noise, then the following inequality holds for any noise sequence $\{\phi_n\}_{n=0}^{\infty}$:

$$\sum_{n=0}^{\infty}(\boldsymbol{\alpha}-\hat{\boldsymbol{\alpha}}_n)^{\mathrm{T}} Q_n (\boldsymbol{\alpha}-\hat{\boldsymbol{\alpha}}_n) \le \gamma \sum_{n=0}^{\infty}|\phi_n|^2. \tag{19}$$

The left-hand side is the total estimation error, of which the mean is the variance of the estimate. Hence, from (19), we know that the total estimation error will not exceed the noise energy multiplied by γ. Consequently, a small value of γ gives a more tight performance bound for the estimator. Note that, since γ should be larger than 1 from the construction of the estimator, a value of γ which is very close to 1 would result in a numerically unstable algorithm.

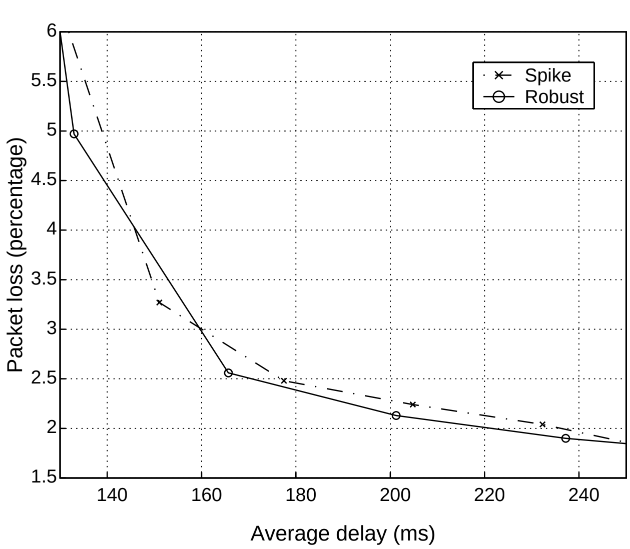

Fig. 6. Delay-loss comparison with the spike algorithm (Trace 1).

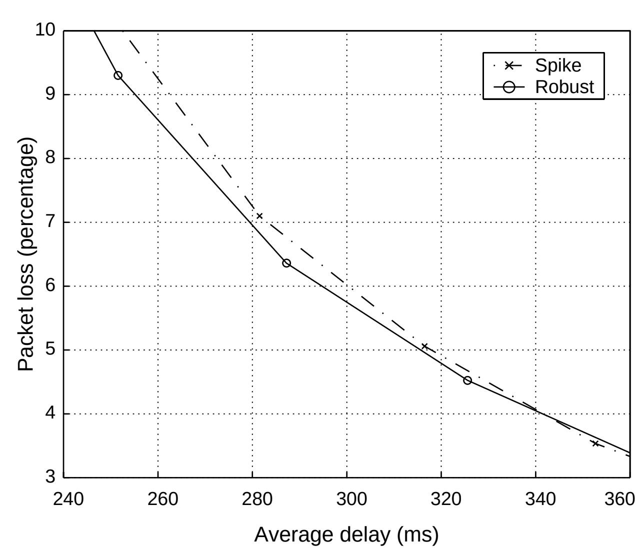

Fig. 7. Delay-loss comparison with the spike algorithm (Trace 2).

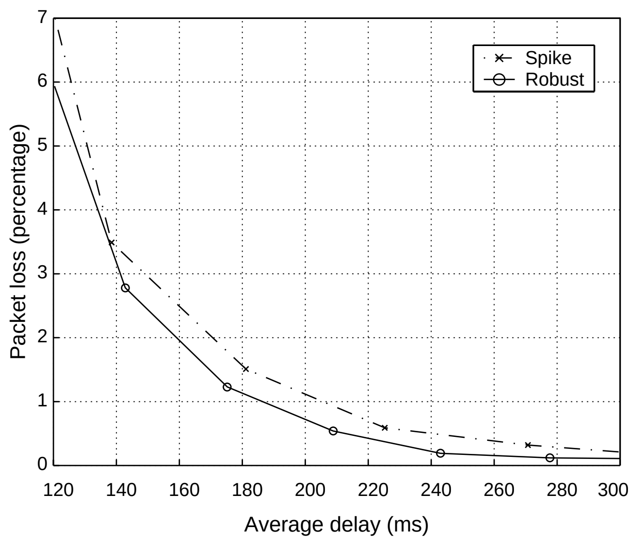

Fig. 8. Delay-loss comparison with the spike algorithm (Trace 3).

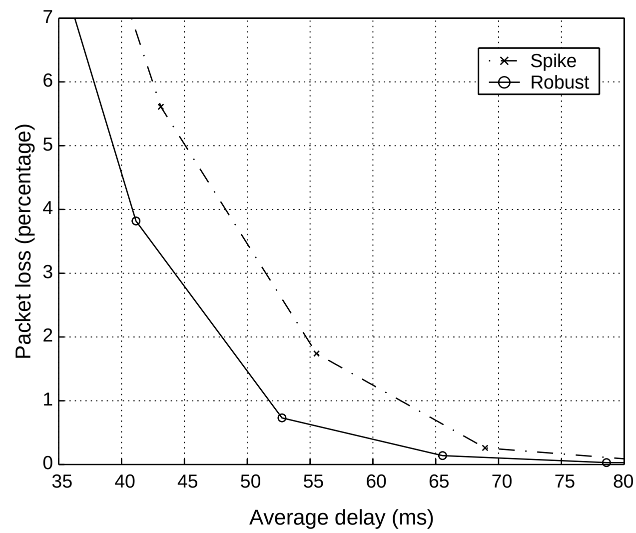

Fig. 10. Delay-loss comparison with the spike algorithm (Trace 5).

Here, we compare performance of the algorithm applied to Trace 2 with different values of $\gamma = 1.1$, 1.5, 3.0, 10, and ∞, respectively. We can expect from (19) that the estimation performance degrades as γ increases in an average sense. Table 2 shows the simulation result with different values of γ. We can see that the estimation performance degrades a little as γ increases. Hence, a small value of γ which is not too close to 1 would be appropriate.

5. Conclusion

We have proposed a robust estimation algorithm for the Internet delay estimation problem. The most salient feature of the proposed algorithm is that the parameters of an AR model are identified by a robust identification algorithm which is based on recent development in robust control [14]. Hence, the overall algorithm is adaptive to an abrupt change in network environment. The basic algorithm, i.e., the Jacobson's algorithm does not have this kind of adaptiveness. Furthermore, this identification algorithm is not only more effective when noise is correlated and/or non-stationary, but is also more robust to model uncertainties than the usual estimation schemes. The objective of the H^∞ criterion is to minimize the total estimation error when we do not have any knowledge on the noise process. Hence, the resulting estimator in (12) is an optimal policy for any unknown noise.

We have applied the proposed scheme to the problem of Internet audio playout mechanism. Our simulation result shows the effectiveness of the method. Furthermore,

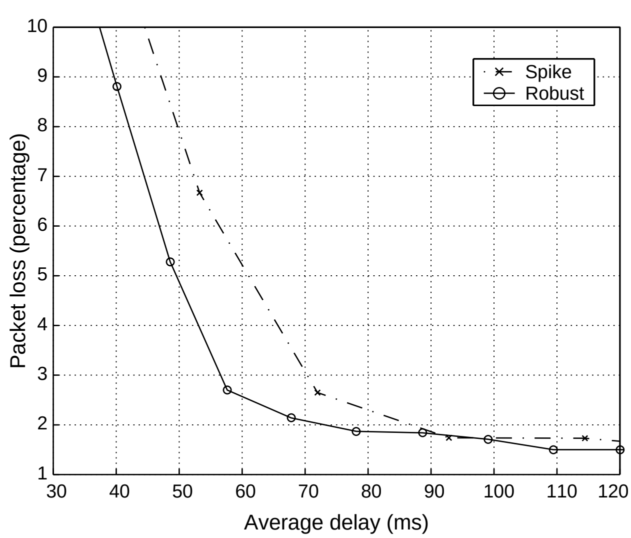

Fig. 9. Delay-loss comparison with the spike algorithm (Trace 4).

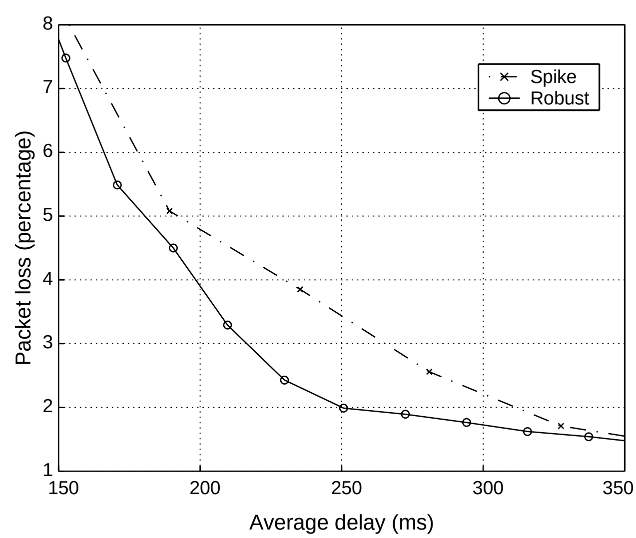

Fig. 11. Delay-loss comparison with the spike algorithm (Trace 6).

Table 2
Effect of the parameter γ

γ	1.1	1.5	3.0	10	∞
std (ms)	36.43	36.47	36.54	36.56	36.57

the proposed robust playout scheme covers most of the existing algorithms and hence can be regarded as a general algorithm, which has been obtained from concrete theoretical derivation. The benefit of the proposed algorithm over some intuitive and heuristic algorithms is that we do not have any parameters to tune by a trial-and-error. We only have one parameter γ, which can be adjusted systemically. Another potential feature of the proposed scheme would be robustness that may overcome the self-similarity of Internet traffic [19]. Since robustness of the proposed algorithm is guaranteed for any kind of noise, we can expect that the estimation performance will not degrade with self-similar traffic. We expect that the proposed algorithm can be applied to many estimation problems in Internet multimedia applications.

Acknowledgements

The first author would like to appreciate Won J. Jeon for his initial stimulation on this work and also appreciate Prof Tamer Başar and Prof. Klara Nahrstedt for their helpful comments. This work has been partly supported by Posco and the BK21 Program of the Korean Ministry of Education and Human Resources.

Appendix A

Theorem here is a modified version of Theorem 1 in [17]. Hence, we give a proof by following the result in [14]. Since, the proof of the positive definiteness of Σ_n is the same as that in [14], we begin with the proof of Theorem part (2). First, we introduce a time-truncated version of (8) as follows:

$$L_\gamma^n = \sum_{i=0}^{n} |\alpha - \hat{\alpha}_i|^2_{Q_i} - \gamma^2 \sum_{i=0}^{n} |\phi_i|^2 - \gamma^2 |\alpha - \hat{\alpha}_i|^2_{Q_0}. \quad \text{(A1)}$$

If we substitute (5) for ϕ_n and complete the squares in $\boldsymbol{\alpha}$, then we get the following for L_γ^n:

$$L_\gamma^n = -\gamma^2 |\boldsymbol{\alpha} - \boldsymbol{\beta}_n|^2_{\Sigma_{n+1}} + m_{n+1}, \quad n \geq 0 \quad \text{(A2)}$$

where $\boldsymbol{\beta}_i$ and m_i are generated by

$$\boldsymbol{\beta}_i = \Sigma_{i+1}^{-1}[\Sigma_i \boldsymbol{\beta}_{i-1} + d_i \boldsymbol{d}_{i-1} - \gamma^{-2} Q_i \hat{\boldsymbol{\alpha}}_i],$$

$$\boldsymbol{\beta}_0 = \Sigma_1^{-1}[Q_0 \boldsymbol{\alpha}_0 - \gamma^{-2} Q_0 \hat{\boldsymbol{\alpha}}_0]$$

and

$$m_{i+1} = m_i - \gamma^2 s_i + [\hat{\boldsymbol{\alpha}}_i - \bar{\boldsymbol{\alpha}}_i]^2_{(\Sigma_i + \boldsymbol{d}_{i-1}\boldsymbol{d}_{i-1}^T)\Sigma_{i+1}^{-1} Q_i},$$
$$m_1 = |\hat{\boldsymbol{\alpha}}_0 - \bar{\boldsymbol{\alpha}}_0|_{Q_0 \Sigma_1^{-1} Q_0} \quad \text{(A3)}$$

where

$$\bar{\boldsymbol{\alpha}}_i = \boldsymbol{\beta}_{i-1} + (\Sigma_i + \boldsymbol{d}_{i-1}\boldsymbol{d}_{i-1}^{\mathrm{T}})^{-1}[d_i - \boldsymbol{d}_{i-1}^{\mathrm{T}}\boldsymbol{\beta}_{i-1}]\boldsymbol{d}_{i-1},$$

$$s_{i+1} = \boldsymbol{\beta}_i^{\mathrm{T}} \Sigma_{i+1} \boldsymbol{\beta}_i + d_{i+1}^2 - (\Sigma_{i+1}\boldsymbol{\beta}_i + d_{i+1}\boldsymbol{d}_i)^{\mathrm{T}}(\Sigma_{i+1} + \boldsymbol{d}_i\boldsymbol{d}_i^{\mathrm{T}})^{-1}(\Sigma_{i+1}\boldsymbol{\beta}_i + d_{i+1}\boldsymbol{d}_i).$$

Since, $\Sigma_i > 0$, the first term in (8) is negative and the third term in (A3) is positive. Hence, if we choose $\hat{\boldsymbol{\alpha}}_i$ as

$$\hat{\boldsymbol{\alpha}}_i = \bar{\boldsymbol{\alpha}}_i, \quad \text{(A4)}$$

then a non-positive value for L_γ can be guaranteed. Since, the upper value cannot be negative by the construction of L_γ, it follows that (A4) is the unique optimal policy. By inserting $\hat{\boldsymbol{\alpha}}_i = \bar{\boldsymbol{\alpha}}_i$ into (A3) and comparing the result with (A4), we know that $\boldsymbol{\beta}_i = \bar{\boldsymbol{\alpha}}_i$. Again by using this in (A3), we get $\boldsymbol{\beta}_i = \hat{\boldsymbol{\alpha}}_i$ where $\hat{\boldsymbol{\alpha}}_i$ is generated by (12). This completes the proof of Theorem 2.

Now we consider Theorem 3. First, we introduce a candidate Lyapunov function $V(n; \tilde{\boldsymbol{\alpha}}_n)$ as follows: $V(n; \tilde{\boldsymbol{\alpha}}_n) = \tilde{\boldsymbol{\alpha}}_n^{\mathrm{T}} \Sigma_{n+1} \tilde{\boldsymbol{\alpha}}_n$ where $\tilde{\boldsymbol{\alpha}}_n = \hat{\boldsymbol{\alpha}}_n - \boldsymbol{\alpha}$. After some algebraic calculation, we have

$$V(n+1; \tilde{\boldsymbol{\alpha}}_{n+1}) - V(n; \tilde{\boldsymbol{\alpha}}_n) = A_n - \gamma^{-2} B_n \quad \text{(A5)}$$

where

$$A_n := (\Sigma_{n+1}\tilde{\boldsymbol{\alpha}}_n + \phi_n \boldsymbol{d}_n)^{\mathrm{T}}(\Sigma_{n+1} + \boldsymbol{d}_n\boldsymbol{d}_n^{\mathrm{T}})^{-1}(\Sigma_{n+1}\tilde{\boldsymbol{\alpha}}_n + \phi_n \boldsymbol{d}_n) - \tilde{\boldsymbol{\alpha}}_n^{\mathrm{T}} \Sigma_{n+1} \tilde{\boldsymbol{\alpha}}_n$$

$$B_n := (\Sigma_{n+1}\tilde{\boldsymbol{\alpha}}_n + \phi_n \boldsymbol{d}_n)^{\mathrm{T}}(\Sigma_{n+1} + \boldsymbol{d}_n\boldsymbol{d}_n^{\mathrm{T}})^{-\mathrm{T}} Q_n (\Sigma_{n+1} + \boldsymbol{d}_n\boldsymbol{d}_n^{\mathrm{T}})^{-1}(\Sigma_{n+1}\tilde{\boldsymbol{\alpha}}_n + \phi_n \boldsymbol{d}_n).$$

We further have $A_n \leq |\phi_n|^2$ and $B_n \geq 0$. Hence the following inequality holds from (A5):

$$V(n+1; \tilde{\boldsymbol{\alpha}}_{n+1}) - V(n; \tilde{\boldsymbol{\alpha}}_n) \leq |\phi_n|^2. \quad \text{(A6)}$$

Finally, we have

$$V(N; \tilde{\boldsymbol{\alpha}}_N) = V(0; \tilde{\boldsymbol{\alpha}}_0) + \sum_{n=0}^{N-1}(A_n - \gamma^{-2} B_n) \quad \text{(A7)}$$

$$\leq V(0; \tilde{\boldsymbol{\alpha}}_0) + \sum_{n=0}^{N-1} |\phi_n|^2. \quad \text{(A8)}$$

Since, $\{\phi_n\}$ is l_2, i.e. $\sum_{n=0}^{N-1} |\phi_n|^2 < \infty$, the right-hand side of (A8) is bounded. From Condition 1, $\lim_{n\uparrow\infty} \lambda_{\min}(\Sigma_n) = \infty$, and consequently $\lim_{n\uparrow\infty} \tilde{\boldsymbol{\alpha}}_n = 0$, i.e., $\lim_{n\uparrow\infty} \tilde{\boldsymbol{\alpha}}_n = \boldsymbol{\alpha}$.

References

[1] X. Wang, H Schulzrinne, Comparison of adaptive Internet multimedia applications, IEICE Transactions on Communications E82-B (6) (1999).

[2] B. Li, K. Nahrstedt, A control theoretical model for quality of service adaptations, Proceedings of the IEEE International Workshop on Quality of Service (IWQoS), Napa, CA, 1998.

[3] B. Li, Agilos: a middleware control architecture for application-aware quality of service adaptations, PhD Dissertation, Department of Computer Science, University of Illinois at Urbana-Champaign, May 2000.

[4] V. Paxon, Measurements and analysis of end-to-end Internet dynamics, PhD Dissertation, Computer Science Department, University of California, Berkeley, 1997.

[5] R. Ramjee, J. Kurose, D. Towsley, H. Schulzerine, Adaptive playout mechanisms for packeteized audio applications in wide area networks, in: Proceedings of IEEE INFOCOM, 1994, pp. 680–686.

[6] H. Schulzrinne, Voice communication across the Internet: a network voice terminal, Technical Report, Department of Computer Science, University of Massachusetts, Amherst, MA, July 1992.

[7] V. Jacobson, Congestion avoidance and control, in: Proceedings of ACM SIGCOMM, 1988.

[8] S.B. Moon, J. Kurose, D. Towsley, Packet audio playout delay adjustment: performance bounds and algorithms, Multimedia Systems 6 (1998) 17–28.

[9] A. Kansal, A. Karandikar, Adaptive delay estimation for low jitter audio over Internet, in: Proceedings of IEEE GLOBECOM, 2001.

[10] C.J. Sreenan, J.C. Chen, P. Agrawal, B. Narendran, Delay reduction techniques for playout buffering, IEEE Transactions on Multimedia 2 (2) (2000) 88–100.

[11] Q. Li, D.L. Mills, Jitter based delay boundary prediction of wide-area networks, IEEE/ACM Transactions on Networking 9 (5) (2001).

[12] P. DeLeon, C.J. Sreenan, An adaptive predictor for media playout buffering, in: Proceedings of IEEE ICASSP, Phoenix, AZ, March 1999, pp. 3097–3100.

[13] S. Haykin, Adaptive Filter Theory, third ed., Prentice-Hall, Englewood Cliffs, NJ, 1998.

[14] G. Didinsky, Z. Pan, T. Başar, Parameter identification for uncertain plants using H^∞ methods, Automatica 31 (9) (1995) 1227–1250.

[15] K.-J. Park, W.J. Jeon, K. Nahrstedt, T. Başar, C.-H. Choi, Robust delay estimation for Internet multimedia applications, in: Proceedings of Joint International Workshop on Interactive Distributed Multimedia Systems and Protocols for Multimedia Systems (IMDS/PROMS) 2002, Coimbra, Portugal, November 2002.

[16] T. Başar, P. Bernhard, H^∞-Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach, Birkhäuser, Boston, MA, 1995.

[17] O. Ait-Hellal, E. Altman, T. Başar, A robust identification algorithm for traffic models in telecommunications, in: Proceedings of IEEE Conference on Decision and Control, 1999, pp. 3071–3076.

[18] G.C. Goodwin, K.W. Sin, Adaptive Filtering, Prediction, and Control, Prentice-Hall, Englewood Cliffs, NJ, 1984.

[19] K. Park, W. Willinger (Eds.), Self-Similar Network Traffic and Performance Evaluation, Wiley–Interscience, New York, 2000.