

A Comparative Study of Five LMS-Based Adaptive Time Delay Estimators

H. C. So and P. C. Ching

Department of Electronic Engineering, The Chinese University of Hong Kong
Shatin, N.T., Hong Kong

June 23, 2000

Abstract

A number of adaptive algorithms for estimating time difference between signals received at two spatially separated sensors have been proposed to model the delay using an FIR filter. Among them, there are the LMSTDE, CTDE, ETDE, SETDE and ETDGE, which are computationally efficient because of the LMS implementation. In this paper, these five methods are compared in terms of estimation accuracy and computational complexity. It is proved that the LMSTDE and ETDGE attain identical performance for sufficiently long filter lengths, although the ETDE and SETDE perform similarly to the ETDGE at high signal-to-noise ratio (SNR) and low SNR, respectively. The CTDE involves minimum computational load but it is the worst estimator in the presence of noise. In addition, optimum realizations of the LMSTDE as well as the ETDE and its variants are derived and their delay variances are compared with the Cramér-Rao lower bound. Simulation results show that ETDGE outperforms the other four methods for a wide range of filter lengths at different SNRs.

indexing terms : *time delay estimation, adaptive signal processing*

Corresponding Author:

P. C. Ching

Department of Electronic Engineering

The Chinese University of Hong Kong

Shatin, N.T., Hong Kong

Tel : (852) 2609 8275

Fax: (852) 2603 5701

email: pcching@ee.cuhk.edu.hk

I. Introduction

Estimation and tracking of time delay between two noisy versions of the same signal received at spatially separated sensors have important applications such as direction finding, source location and speed sensing [1]. The mathematical model of the discrete-time sensor outputs is given by

$$x(k) = s(k) + n_1(k) \quad \text{and} \quad y(k) = s(k - D) + n_2(k) \quad (1)$$

where $s(k)$ is the unknown source signal, and $n_1(k)$ and $n_2(k)$ are the additive noises. Without loss of generality, the sampling interval is taken to be unity second. It is assumed that the source signal is stationary Gaussian and bandlimited between -0.5 Hz and 0.5 Hz, while the corrupting noises are uncorrelated white Gaussian processes which are independent of $s(k)$. The objective is to estimate the time delay, D , from the received signals $x(k)$ and $y(k)$.

When the time delay is nonstationary due to relative source/receiver motion, adaptive tracking of D is necessary [1]-[18]. In this paper, the estimation performance of five adaptive time delay estimators, namely, least mean square time delay estimator (LMSTDE) [2]-[3], constrained time delay estimator (CTDE) [7], explicit time delay estimator (ETDE) [10], simplified explicit time delay estimator (SETDE) [16] and explicit time delay and gain estimator (ETDGE) [12]-[13], are compared. We limit our investigation to these LMS-type algorithms because they do allow real-time implementation and no *a priori* spectral contents of the received signals are needed. The LMSTDE uses an adaptive FIR filter to model the time difference and the filter weights are interpolated to obtain the delay estimate. By constraining the FIR filter coefficients to be samples of a sinc function, the CTDE simplifies the LMSTDE algorithm considerably because only the largest filter weight is updated and no interpolation is necessary. The ETDE can be considered as an alternative realization of the CTDE but with explicit adjustment of the delay estimate. Both ETDGE and SETDE are variants of the ETDE and they can provide more accurate delay estimation than the ETDE at very low signal-to-noise ratio (SNR). In the ETDGE, an adaptive gain factor is added in series with the ETDE for optimum filtering while in the SETDE, the mean product of one of the sensor outputs and the filtered output of the other received signal is maximized iteratively.

In Section II, the convergence behavior and mean square error of the delay estimates of the five methods for static as well as linearly varying delays are analyzed and their computational complexity are discussed. When the delay convergence rates are kept identical, we prove that the LMSTDE and ETDGE provide the same delay variance for sufficiently long filter length. It is also shown that the performance of the ETDE and SETDE approach that of the ETDGE at high and low SNR, respectively, while the CTDE gives biased delay estimates when noise is present. In Section III, optimum performance of the LMSTDE, ETDE, SETDE and ETDGE are derived and compared with the Cramér-Rao lower bound (CRLB). Simulation results are presented in Section IV to compare the delay estimation performance of the five techniques and to corroborate the theoretical analysis. Finally, conclusions are drawn in Section V.

II. Adaptive Algorithms for Delay Estimation

A. LMSTDE

The system block diagram of the LMSTDE is shown in Figure 1 with $\hat{\alpha}$ fixed at unity. The basic idea is to model the time difference by an adaptive noncausal FIR filter given by $W(z) = \sum_{i=-P}^P w_i z^{-i}$. Each $w_i(k)$ is adapted according to Widrow's LMS algorithm [19]:

$$w_i(k+1) = w_i(k) + \mu_w e(k)x(k-i), \quad -P \leq i \leq P \quad (2)$$

where $e(k) = y(k) - \sum_{i=-P}^P w_i(k)x(k-i)$ and μ_w is a positive scalar that controls the convergence rate and stability of the updating rule. Using the interpolation formula [20], $s(k-D)$ can be expressed as

$$s(k-D) = \sum_{i=-\infty}^{\infty} s(k-i)\text{sinc}(i-D) \quad (3)$$

where $\text{sinc}(v) = \sin(\pi v)/(\pi v)$. To simplify the analysis, we further assume that $s(k)$ is also a white Gaussian process [3]-[4] and the channel inputs are uncorrelated with the filter weights [21]. Note that if the source signal is nonwhite and/or when it is not of Gaussian distribution, the delay estimation algorithms can still be employed but we may not be able to obtain a closed-form derivation. Taking the expected value of (2) and applying (3) yields

$$E\{w_i(k)\} = \frac{\sigma_s^2}{\sigma_s^2 + \sigma_n^2} \text{sinc}(i-D)(1 - (1 - \mu_w(\sigma_s^2 + \sigma_n^2))^k) + w_i(0)(1 - \mu_w(\sigma_s^2 + \sigma_n^2))^k \quad (4)$$

provided that $0 < \mu_w < 2/(\sigma_s^2 + \sigma_n^2)$. The parameters σ_s^2 , σ_n^2 and $\{w_i(0)\}$ represent the signal power, noise variance and the initial filter weight values, respectively. Upon convergence, the filter coefficients will approach the Wiener solution for delay estimation, viz., $w_i^* = \text{SNR}/(1 + \text{SNR})\text{sinc}(i-D)$ for $-P \leq i \leq P$, where $\text{SNR} = \sigma_s^2/\sigma_n^2$. The delay estimate of the LMSTDE, $\hat{D}_w(k)$, is obtained indirectly from $\{w_i(k)\}$ by interpolation and it is given by [2]

$$\hat{D}_w(k) = \arg \max_t \left\{ \sum_{i=-P}^P w_i(k)\text{sinc}(t-i) \right\} \quad (5)$$

If $W(z)$ has infinite coefficients, the mean delay estimate is exactly equal to D when it converges but it will be biased for finite P [4]. The computational load of updating $\{w_i(k)\}$ requires $(4P+2)$ additions and $(4P+3)$ multiplications for each iteration. However, if interpolation of $\{w_i(k)\}$ is also considered, a further $20P$ additions and $(20P+10)$ multiplications will be needed assuming that the delay estimate has a resolution of 0.001s.

Using (3)-(5) and assuming that the initial values of the filter weights correspond to a delay of D_I , the mean value of $\hat{D}_w(k)$ can be expressed as

$$E\{\hat{D}_w(k)\} \approx \arg \max_t \left\{ (1 - (1 - \mu_w(\sigma_s^2 + \sigma_n^2))^k)\text{sinc}(t-D) + (1 - \mu_w(\sigma_s^2 + \sigma_n^2))^k\text{sinc}(t-D_I) \right\} \quad (6)$$

for sufficiently large P . Since the first term of (6) peaks at $t = D$ with a value of $(1 - (1 - \mu_w(\sigma_s^2 + \sigma_n^2))^k)$ and the second term has a maximum value of $(1 - \mu_w(\sigma_s^2 + \sigma_n^2))^k$ when $t = D_I$, we can approximate the learning behavior of the delay estimate by the following weighted sum,

$$\begin{aligned} E\{\hat{D}_w(k)\} &\approx (1 - (1 - \mu_w(\sigma_s^2 + \sigma_n^2))^k)D + (1 - \mu_w(\sigma_s^2 + \sigma_n^2))^k D_I \\ &= D + (D_I - D)(1 - \mu_w(\sigma_s^2 + \sigma_n^2))^k \end{aligned} \quad (7)$$

Hence the time constant of the delay estimate is approximately equal to that of $\{w_i(k)\}$, which is equal to $\tau_w = 1/(\mu_w(\sigma_s^2 + \sigma_n^2))$. It is noteworthy that setting $w_i(0) = 0$ for $-P \leq i \leq P$ will give the fastest delay convergence rate when the delay has a static value, because $E\{\hat{D}_w(k)\} \approx D$ for all k in (6). In this case, an accurate delay estimate can be obtained after a few iterations even when the filter coefficients have not settled to the Wiener solution.

From [8], the variance of the delay estimate, $\text{var}(\hat{D}_w)$, is given by

$$\text{var}(\hat{D}_w) = \lim_{k \rightarrow \infty} E\{(\hat{D}_w(k) - D)^2\} = \frac{(1 + \text{SNR}^2) \sum_{i=-P}^P \text{var}(w_i) \text{sinc}'^2(i - t)}{\text{SNR}^2 (\sum_{i=-P}^P \text{sinc}(i - D) \text{sinc}''(i - t))^2} \Big|_{t=D} \quad (8)$$

where the symbols ' and '' denote the first and the second derivative respectively and $\text{var}(w_i)$ is the variance of the filter weight $w_i(k)$. Using L'Hôpital's rule, it can be shown that $\sum_{i=-\infty}^{\infty} \text{sinc}(i - D) \text{sinc}''(i - t) \Big|_{t=D} = -\pi^2/3$. Moreover, $\sum_{i=-\infty}^{\infty} \text{sinc}'^2(i - t) \Big|_{t=D} = \pi^2/3$ [22] and $\text{var}(w_i) \approx \mu_w \sigma_n^2 (1 + \text{SNR}/(1 + \text{SNR}))$ [19]. Hence, (8) can be simplified to

$$\text{var}(\hat{D}_w) \approx \frac{3\mu_w \sigma_s^2 (1 + 2\text{SNR})(1 + \text{SNR})}{2\pi^2 \text{SNR}^3} \quad (9)$$

When the time delay is linearly varying, that is, $D(k) = D_0 + \lambda k$, where D_0 and λ represent the delay at $k = 0$ and Doppler time compression respectively, the mean value of $\hat{D}_w(k)$ can be approximated as [11]

$$E\{\hat{D}_w(k)\} \approx D_0 + \lambda k - \frac{\lambda}{\mu_w(\sigma_s^2 + \sigma_n^2)} + \frac{\lambda k (1 - \mu_w(\sigma_s^2 + \sigma_n^2))^k}{1 - (1 - (\mu_w(\sigma_s^2 + \sigma_n^2))^k)} \quad (10)$$

Upon convergence, the LMSTDE tracks the linearly varying delay with a time lag of $\lambda/(\mu_w(\sigma_s^2 + \sigma_n^2))$.

B. CTDE

The CTDE constrains the FIR filter coefficients to be samples of a sinc function and it simplifies the LMSTDE algorithm considerably because in this case only the peak weight, say, $w_L(k)$, needs to be adapted according to (2) in each iteration. The delay estimate of this method, denoted by $\hat{D}_c(k)$, is related to $w_L(k)$ by the mapping $w_L(k) = \text{sinc}(L - \hat{D}_c(k))$. The values of the remaining filter weights are determined by $w_i(k) = \text{sinc}(i - \hat{D}_c(k))$ and they can be easily found by a table lookup operation [7]. Although the computational load of the CTDE is small, the algorithm has its shortcomings. Using (2), the updating rule for $w_L(k)$ can be written as

$$\text{sinc}(L - \hat{D}_c(k + 1)) = \text{sinc}(L - \hat{D}_c(k)) + \mu_w x(k - L)(y(k) - \sum_{i=-P}^P \text{sinc}(i - \hat{D}_c(k))x(k - i)) \quad (11)$$

Taking the expected value of (11) gives

$$E\{\text{sinc}(L - \hat{D}_c(k+1))\} = E\{\text{sinc}(L - \hat{D}_c(k))\}(1 - \mu_w(\sigma_s^2 + \sigma_n^2)) + \mu_w\sigma_s^2\text{sinc}(L - D) \quad (12)$$

Examining the steady state condition of (12), we get

$$\lim_{k \rightarrow \infty} E\{\text{sinc}(L - \hat{D}_c(k))\} = \lim_{k \rightarrow \infty} E\{w_L(k)\} = \frac{\sigma_s^2}{\sigma_s^2 + \sigma_n^2}\text{sinc}(L - D) \quad (13)$$

At a very high SNR environment or noise-free condition, accurate delay estimation will be acquired since $\sigma_s^2/(\sigma_s^2 + \sigma_n^2)$ is close to 1. However, large delay errors are introduced through the inaccurate mapping at lower SNR. For sufficiently small μ_w , the mean square delay error of the CTDE, $\text{mse}(\hat{D}_c)$, can be approximated by

$$\text{mse}(\hat{D}_c) \approx \left(L - D - \text{sinc}^{-1} \left(\frac{\sigma_s^2}{\sigma_s^2 + \sigma_n^2} \text{sinc}(L - D) \right) \right)^2 \quad (14)$$

C. ETDE

Similar to the CTDE, the ETDE uses the property that a delayed version of a bandlimited signal can be represented by convolving a sinc function with the signal itself. Basically, the ETDE has the same filter structure with the LMSTDE but the filter coefficients $\{w_i(k)\}$ are replaced by $\{\text{sinc}(i - \hat{D}_e(k))\}$ for $-P \leq i \leq P$, where $\hat{D}_e(k)$ is the estimated delay. The output error function in the ETDE can be computed from $e(k) = y(k) - \tilde{x}(k - \hat{D}_e(k))$ where $\tilde{x}(k - \hat{D}_e(k)) \triangleq \sum_{i=-P}^P \text{sinc}(i - \hat{D}_e(k))x(k - i)$ denotes an approximate version of $x(k)$ with time lag $\hat{D}_e(k)$. The delay estimate is updated directly and iteratively according to [10]

$$\hat{D}_e(k+1) = \hat{D}_e(k) - \mu_e e(k) \sum_{i=-P}^P f(i - \hat{D}_e(k))x(k - i) \quad (15)$$

where $f(v) = (\cos(\pi v) - \text{sinc}(v))/v$ and μ_e is the convergence factor of the ETDE algorithm. To reduce computation, values of the sinc and cosine function are retrieved from pre-stored tables [10]. By examining the ETDE performance surface, it has been proved [12] that the mean delay estimate \bar{D}_e can be obtained from the following equation,

$$\sum_{i=-P}^P \text{sinc}(i - D)f(i - \bar{D}_e) = \left(1 + \frac{1}{\text{SNR}} \right) \sum_{i=-P}^P \text{sinc}(i - \bar{D}_e)f(i - \bar{D}_e) \quad (16)$$

In the absence of noise or if the filter length is infinitely long, then $\bar{D}_e = D$. However, for other circumstances, the unbiased property of the ETDE will not necessarily exist. As a rule of thumb, the delay bias can be reduced by approximately one-tenth by either increasing the SNR by 10 dB or by a ten-fold increase of P . When P is chosen sufficiently large enough, taking the expected value of (15) yields [10]

$$E\{\hat{D}_e(k)\} \approx D + (\hat{D}_e(0) - D)(1 - \frac{1}{3}\mu_e\sigma_s^2\pi^2)^k \quad (17)$$

where $\hat{D}_e(0)$ denotes the initial delay estimate. Provided that $0 < \mu_e < 3/(2\pi^2\sigma_s^2)$, $E\{\hat{D}_e(k)\}$ will converge to D with a time constant $\tau_e = 3/(\pi^2\sigma_s^2)$. Moreover, the delay variance of the ETDE, denoted by $\text{var}(\hat{D}_e)$, is given by [10]

$$\text{var}(\hat{D}_e) \approx \frac{\mu_e \sigma_s^2 (1 + \text{SNR})}{\text{SNR}^2} \quad (18)$$

A closed form expression of $E\{\hat{D}_e(k)\}$ for $D(k) = D_0 + \lambda k$ is given by [11]

$$E\{\hat{D}_e(k)\} \approx D_0 + \lambda k - \frac{3\lambda}{\mu_e \sigma_s^2 \pi^2} + (\hat{D}_e(0) - D_0 + \frac{3\lambda}{\mu_e \sigma_s^2 \pi^2}) \left(1 - \frac{\mu_e \sigma_s^2 \pi^2}{3}\right) \quad (19)$$

We see that the ETDE tracks the moving delay with a time lag of $3\lambda/(\mu_e \sigma_s^2 \pi^2)$ in steady state.

D. SETDE

In the SETDE, the delay estimate $\hat{D}_s(k)$ is derived from maximizing the mean product of $\tilde{x}(k - \hat{D}_s(k))$ and $y(k)$, that is [16],

$$\hat{D}_s(k) = \arg \max_t \{E\{\tilde{x}(k-t)y(k)\}\} \quad (20)$$

It can be revealed from (5) and (20) that the delay modeling error in the SETDE is identical to that of the LMSTDE. The delay estimate is adapted directly to maximize the instantaneous value of $\tilde{x}(k - \hat{D}_s(k))y(k)$ according to [16]:

$$\hat{D}_s(k+1) = \hat{D}_s(k) - \mu_e y(k) \sum_{i=-P}^P f(i - \hat{D}_s(k)) x(k-i) \quad (21)$$

The SETDE and ETDE algorithm are very similar but the former is more computationally efficient because $(2P+1)$ additions and multiplications are saved for each sampling interval.

It has been proved [16] that the learning behavior of the delay estimate is the same as (17) while the delay variance $\text{var}(\hat{D}_s)$ is given by

$$\text{var}(\hat{D}_s) \approx \frac{\mu_e \sigma_s^2 \left(1 + 2\text{SNR} + \text{SNR}^2 \left(1 - \frac{6}{\pi^2} \sum_{i=1}^P f^2(i)\right)\right)}{2\text{SNR}^2} \quad (22)$$

E. ETDGE

Similar to the LMSTDE, the ETDGE aims to reach the Wiener solution for time delay estimation, although their system structures as well as the adaptive algorithms are different. It comprises an ETDE, whose filter weights are given by $\{\text{sinc}(i - \hat{D}_g(k))\}$, in series with a variable gain $\hat{\alpha}(k)$ for tracking the factor $\text{SNR}/(1+\text{SNR})$ so that the minimum mean square error can be attained. The delay estimate $\hat{D}_g(k)$ is adapted according to (15) and $\hat{\alpha}(k)$ is adjusted using Widrow's LMS algorithm [12] while the error function is now modified to $e(k) = y(k) - \hat{\alpha}(k) \sum_{i=-P}^P \text{sinc}(i - \hat{D}_g(k))x(k-i)$.

Again, the convergence behavior of the delay estimate for the ETDGE is the same as that of the ETDE [12] while the delay variance of the ETDGE, $\text{var}(\hat{D}_g)$, is given by

$$\text{var}(\hat{D}_g) \approx \frac{\mu_e \sigma_s^2 (1 + 2\text{SNR})}{2\text{SNR}^2} \quad (23)$$

Investigation of the global minimum of the ETDGE performance surface reveals [12] that $\hat{D}_g(k)$ is unbiased for any finite P and is independent of SNR. This is an important merit of the ETDGE because a small P can be used to save computation for both high and low SNR conditions, without sacrificing the unbiasedness of the estimated delay.

F. Comparative Performance

The computation requirements of the five adaptive delay estimation algorithms are summarized in Table 1. It can be seen that for the same filter length, the overall computational load of the LMSTDE is much greater than the other four methods due to the extra operations involved in interpolation. Note that if fractional delay estimates are not required at each iteration, then the LMSTDE may require less computation than the ETDE and ETDGE because (5) will only be applied whenever we need to update an estimate of D . On the other hand, CTDE is the most computationally efficient estimator, although its complexity is comparable to those of the ETDE and its variants. However, one or two table lookup operations and additional memory for storing the tables are necessary in these interpolation-free techniques.

In the following, the estimation accuracy of LMSTDE, ETDE, SETDE and ETDGE will be contrasted, with the assumption that the filter length is long enough so that their delay estimates are approximately unbiased. To do so, we first equate the time constants of the delay estimates and then compare their delay variances. Notice that the CTDE is not considered because it is a biased estimator even at high SNR. Equating τ_w and τ_e gives the subsequent relationship between μ_w and μ_e ,

$$\mu_w = \frac{\mu_e \pi^2 \text{SNR}}{3(1 + \text{SNR})} \quad (24)$$

Substituting (24) into (9) yields (23) which implies that both LMSTDE and ETDGE have identical performance at all SNRs. Comparing (18) and (23), it can be seen that the ETDE and ETDGE have equal delay variance of $\mu_e \sigma_n^2$ when $\text{SNR} \gg 1$ but $\text{var}(\hat{D}_g)$ equals $\mu_e \sigma_s^2 / (2\text{SNR}^2)$, which is only half of the ETDE variance, at $\text{SNR} \ll 1$. While comparing (22) and (23), we can observe that the SETDE performs equally well with the ETDGE at low SNR but it is inferior under high SNR conditions. It is worthy to note that when μ_w and μ_e are related by (24), the delay estimates of the four methods will have the same steady state time lag in tracking a linearly varying delay. Due to page limit, interested readers may refer to [11], [18] for Doppler estimation and performance comparison of the delay estimators.

III. Optimum Realizations and Comparison with the CRLB

An optimum realization of the LMSTDE is to use the least squares estimate of the filter weights to compute the delay estimate via (5). The least squares filter coefficients, denoted by $\mathbf{w}^o = \{w_i^o\}$ for $-P \leq i \leq P$, are given by

$$\mathbf{w}^o = \arg \min_{\mathbf{w}} \{J_w(\mathbf{w})\} \quad (25)$$

where $J_w(\mathbf{w}) = \sum_{k=1}^N \left(y(k) - \sum_{i=-P}^P w_i x(k-i) \right)^2$ and N denotes the observation time. The variances of the filter weights are expressed as [2]

$$\text{var}(w_i^o) = \frac{E \left\{ \left(\frac{\partial J_w(\mathbf{w})}{\partial w_i} \right)^2 \right\}}{\left(E \left\{ \frac{\partial^2 J_w(\mathbf{w})}{\partial^2 w_i} \right\} \right)^2} \bigg|_{\mathbf{w}=\mathbf{w}^*}, \quad -P \leq i \leq P \quad (26)$$

and they can be simplified to (Appendix I)

$$\text{var}(w_i^o) \approx \frac{1 + 2\text{SNR}}{N(1 + \text{SNR})^2}, \quad -P \leq i \leq P \quad (27)$$

which are independent of the filter coefficient index i . Replacing $\text{var}(w_i)$ by $\text{var}(w_i^o)$ in (8), we get the delay variance using the optimum LMSTDE filter weights as follows,

$$\text{var}(\hat{D}_w^o) \approx \frac{3(1 + 2\text{SNR})}{\pi^2 N \text{SNR}^2} \quad (28)$$

which is essentially identical to the CRLB for passive time delay estimation and this result also agrees with [23].

Similarly, the ‘‘largest products’’ delay estimate of the SETDE can be calculated as

$$\hat{D}_s^o = \arg \max_{\hat{D}} \{J_s(\hat{D})\} \quad (29)$$

where

$$J_s(\hat{D}) = \sum_{k=1}^N \left(y(k) \sum_{i=-P}^P \text{sinc}(i - \hat{D}) x(k-i) \right) \quad (30)$$

while the least squares ETDGE delay estimate can be computed by minimizing the cost function

$$J_g(\hat{\alpha}, \hat{D}) = \sum_{k=1}^N \left(y(k) - \hat{\alpha} \sum_{i=-P}^P \text{sinc}(i - \hat{D}) x(k-i) \right)^2 \quad (31)$$

The delay variance of the optimum implementation of the SETDE, denoted by $\text{var}(\hat{D}_s^o)$, is evaluated as

$$\text{var}(\hat{D}_s^o) \approx \frac{3 \left(1 + 2\text{SNR} + \text{SNR}^2 \left(1 - \frac{6}{\pi^2} \sum_{i=1}^P f^2(i) \right) \right)}{N \pi^2 \text{SNR}^2} \quad (32)$$

and that of the ETDGE, denoted by $\text{var}(\hat{D}_g^o)$, equals (28) [13] which is the CRLB. Equation (32) indicates that at $\text{SNR} \gg 1$, $\text{var}(\hat{D}_s^o)$ is greater than $\text{var}(\hat{D}_e^o)$ but it approaches the CRLB for small SNR.

Following [13], the variance of the least squares ETDE delay estimate is derived as

$$\text{var}(\hat{D}_e^o) \approx \frac{6(1 + \text{SNR})}{\pi^2 N \text{SNR}^2} \quad (33)$$

which approaches the CRLB for $\text{SNR} \gg 1$. However, the variance is larger than the CRLB by approximately 3 dB at low SNR. In summary, we have proved that the optimum delay variances of both LMSTDE and ETDGE are equal to the CRLB while those of the ETDE and SETDE approach the CRLB at high and low SNR, respectively.

IV. Simulation Results

Simulation tests had been conducted to compare the performance of the LMSTDE, CTDE, ETDE, SETDE and ETDGE for nonstationary delay estimation. The sequences $s(k)$, $n_1(k)$ and $n_2(k)$ were independent white Gaussian processes. The signal power was fixed to unity and different SNRs were obtained by proper scaling of the random noise sequences. The delayed signal $s(k - D)$ was produced by passing $s(k)$ through a 61-tap FIR filter whose transfer function was given by $\sum_{i=-30}^{30} \text{sinc}(i - D)z^{-i}$. In the LMSTDE, $w_i(0) = 0$ for $-P \leq i \leq P$ while the initial delay estimates of the CTDE as well as the ETDE and its variants were set to 0s. For simplicity, $\hat{\alpha}(0)$ of the ETDGE was chosen to be 1 and the step size for $\hat{\alpha}(k)$ was equal to μ_e . The delay estimate of each method had a resolution of 0.001s. To fulfill this requirement, 10 bisections were performed to search the delay estimate of the LMSTDE using (5) while the cosine and sinc table were constructed with sizes 512 and 512×31 , respectively, for the remaining four methods. The results provided were averages of 200 independent runs.

Figure 2 shows the trajectories for the delay estimates of the five algorithms for a step change in D at $\text{SNR} = 10$ dB. The actual delay had a value of 0.3s at the first 4000 iterations and then changed instantaneously to 0.7s afterwards. The step sizes μ_w and μ_e were selected as 2×10^{-3} and 6.687×10^{-4} , respectively, so that all methods had the same convergence speed. It can be seen that the learning curves of the ETDE, SETDE and ETDGE were almost identical and their delay estimates converged to the desired values of 0.3s and 0.7s at approximately the 2000th and 6000th iteration, respectively. The LMSTDE also provided unbiased delay estimation and it had the fastest initial learning rate because an optimum setting of $\{w_i(0)\}$ was used, although its convergence behavior at $D = 0.7$ s was fairly similar to those of the ETDE and its variants. On the other hand, the delay estimate of the CTDE converged to 0.38s and 0.62s upon reaching the steady state which clearly indicated a large delay bias. The discrepancy was due to the inaccurate mapping. At lower SNR, the delay estimation error was much larger. As a result, the CTDE is only suitable to operate at a noise-free or very high SNR environment.

In order to investigate the comparative performance for a short filter length, the previous test was repeated for $P = 3$ and the results are depicted in Figure 3. Again, the ETDE and ETDGE

provided unbiased delay estimates while the estimates of the CTDE were biased and they had nearly equal trajectories as in Figure 2. On the other hand, the delay estimates of both LMSTDE and SETDE converged approximately to 0.27s and 0.73s for $D = 0.3\text{s}$ and $D = 0.7\text{s}$, respectively, although the LMSTDE provided the fastest static behavior. The bias was resulted from the delay modeling error in (5) or (20), which became apparent for a short filter length.

The above tests for $2 \leq P \leq 15$ were performed and the average steady state mean square delay errors of the five methods are shown in Figure 4. It can be seen the CTDE was the poorest delay estimator while the ETDGE provided the minimum delay variance. The mean square delay errors of the CTDE and ETDGE were equal to $6.4 \times 10^{-3}\text{s}^2$ and $8.0 \times 10^{-5}\text{s}^2$, respectively, for the whole range of P , and these values were close to the theoretical calculations in (14) and (23). For $P \geq 5$, the ETDE delay variance converged to a value of approximately $8.3 \times 10^{-5}\text{s}^2$ which agreed with (18). However, the mean square delay errors of the LMSTDE and SETDE were larger than their expected values in (9) and (22), respectively, although the deviations diminished as P increased. In particular when P was small, their delay biases dominated the mean square delay errors which made the measured values much larger than the theoretical delay variances.

Figure 5 compares the LMSTDE, ETDE, ETDGE and SETDE in estimating a linearly varying delay given by $D(k) = 0.0005ks$. The convergence curve of the CTDE was not shown because of its inaccurate delay estimation in the presence of noise. In this trial, μ_w and μ_e were increased to 5×10^{-3} and 1.672×10^{-4} , respectively, and $P = 15$ was used in all methods. It can be seen that the four methods tracked the delay satisfactorily with a time lag close to 0.1s, which agreed with (10) and (19). Moreover, the measured values of their mean square delay errors were approximately 0.01s^2 . It is noteworthy that the delay estimation accuracy of the four algorithms can be further increased if we estimate the Doppler time compression as well using the slopes of their delay trajectories [11].

The convergence characteristics of the LMSTDE, ETDE and its variants for a step-changed delay at $\text{SNR} = -10$ dB for $P = 15$ and $P = 3$ are shown in Figures 6 and 7, respectively, while their mean square delay errors for $2 \leq P \leq 15$ are plotted in Figure 8. The nonstationary delay had a value of 0.3s in the first 40000 iterations and $D = 0.7\text{s}$ afterwards. Again, in order to maintain an identical convergence speed, the step size parameters μ_w and μ_e were assigned values of 2×10^{-5} and 6.687×10^{-5} , respectively. Notice that the step sizes were chosen small so that occurrence of false peak weights in the LMSTDE, which has a high probability at low SNR [24], could be eliminated. We observe in Figure 6 that the LMSTDE, SETDE and ETDGE accurately tracked the delay and had similar learning behavior except for the LMSTDE at $D = 0.3\text{s}$. However, the delay estimate of the ETDE was biased and it had values of 0.34s and 0.64s for $D = 0.3\text{s}$ and $D = 0.7\text{s}$, respectively, upon convergence. As a result, it is not appropriate to use ETDE at very small SNR because it does not provide an accurate delay estimation even with a long filter length. In Figure 7, we see that only the ETDGE provided unbiased delay estimation as in Figure 6. While

the delay estimates of the other three methods were inaccurate due to delay modeling errors given by (5), (16) and (20), although the estimation errors of the LMSTDE and SETDE were smaller than that of the ETDE. From Figure 8, it is observed that the delay variances of the ETDGE for the whole range of P as well as those of the LMSTDE, SETDE and ETDE for $P \geq 9$ conformed to the theoretical calculation. Whilst the measured mean square delay errors for the remaining cases were larger than their expected values because the biases in the delay estimates governed the errors. Again, the ETDGE gave the smallest measured delay variances for $2 \leq P \leq 15$.

V. Conclusions

Five LMS-based adaptive delay estimators, namely, the LMSTDE, CTDE, ETDE, SETDE and ETDGE, for estimating time delay between signals received at separated sensors are compared in terms of computational complexity and estimation accuracy. Although all these methods use an FIR filter to model the time difference of arrival, they have different structures and adaptive algorithms. When the filter length is kept identical, the CTDE involves minimum computation but it cannot give unbiased delay estimation even at high SNR. We have shown that the LMSTDE and ETDGE attain the same delay variance when the learning speed of their delay estimates are equal and of sufficiently long filter length. While the estimation accuracy of the ETDE and SETDE approach that of the ETDGE at $\text{SNR} \gg 1$ and $\text{SNR} \ll 1$, respectively. The delay variances of optimum realizations of the LMSTDE as well as the ETDE and its variants are derived which show that those of the LMSTDE and ETDGE attain the CRLB for all SNRs. It is demonstrated that the ETDGE provides unbiased delay estimates for a wide range of filter lengths and different SNRs. As a result, the ETDGE is superior to the other four methods because it can be made more computationally efficient by using less filter taps but without degradation in delay estimation performance.

Appendix I

The derivation of (27) is given as follows. Partial differentiating $J_w(\mathbf{w})$ with respect to w_j , $-P \leq j \leq P$, yields

$$\frac{\partial J_w(\mathbf{w})}{\partial w_j} = -2 \sum_{k=1}^N (y(k) - \sum_{i=-P}^P w_i x(k-i)) x(k-j) \quad (\text{A.1})$$

Squaring both sides of (A.1) and taking the expected value at $\mathbf{w} = \mathbf{w}^*$, we get

$$\begin{aligned} E \left\{ \left(\frac{\partial J_w(\mathbf{w})}{\partial w_j} \right)^2 \right\} \Big|_{\mathbf{w}=\mathbf{w}^*} &\approx 4E \left\{ \left(\sum_{k=1}^N (y(k) - \frac{\text{SNR}}{1+\text{SNR}} x(k-D)) x(k-j) \right)^2 \right\} \\ &= 4N\sigma_n^4(1+2\text{SNR}) \end{aligned} \quad (\text{A.2})$$

On the other hand, it is easy to show that

$$E \left\{ \frac{\partial^2 J_w(\mathbf{w})}{\partial^2 w_j} \right\} \Big|_{\mathbf{w}=\mathbf{w}^*} = 2N(1+\text{SNR})\sigma_n^2 \quad (\text{A.3})$$

Using (A.2) and (A.3), we obtain (27).

References

- [1] CARTER, G.C.: *Coherence and Time Delay Estimation: An Applied Tutorial for Research, Development, Test, and Evaluation Engineers*, IEEE Press, 1993.
- [2] REED, F.A., FEINTUCH, P.L. and BERSHAD, N.J.: "Time delay estimation using the LMS adaptive filter - static behaviour," *IEEE Trans. Acoust., Speech, Signal Processing*, 1981, **29**, (3), pp.561-571.
- [3] YOUN, D.H., AHMED, N. and CARTER, G.C.: "On using the LMS algorithm for time delay estimation," *IEEE Trans. Acoust., Speech, Signal Processing*, 1982, **30**, (5), pp.798-801.
- [4] CHAN Y.T., RILEY, J.M.F. and PLANT, J.B.: "Modeling of time-delay and its application to estimation of nonstationary delays," *IEEE Trans. Acoust, Speech, Signal Processing*, 1981, **29**, (3), pp.577-581.
- [5] ETTER, D.M. and STEARNS, S.D.: "Adaptive estimation of time delays in sampled data systems," *IEEE Trans. Acoust. Speech, Signal Processing*, 1981, **29**, (3), pp.582-586.
- [6] SMITH, J.O. and FRIEDLANDER, B.: "Adaptive interpolated time-delay estimation," *IEEE Trans. Aerosp. Electron.*, 1985, **21**, (2), pp.180-199.
- [7] CHING, P.C. and CHAN, Y.T.: "Adaptive time delay estimation with constraints," *IEEE Trans. Acoust., Speech, Signal Processing*, 1988, **36**, (4), pp.599-602.
- [8] HO, K.C., CHAN, Y.T. and CHING, P.C.: "Adaptive time-delay estimation in nonstationary signal and/or noise power environments," *IEEE Trans. Signal Processing*, 1993, **41**, (7), pp.2289-2299.
- [9] BOUDREAU, D. and KABAL, P.: "Joint time-delay estimation and adaptive recursive least squares filtering," *IEEE Trans. Signal Processing*, 1993, **41**, (2), pp.592-601.
- [10] SO, H.C., CHING, P.C. and CHAN, Y.T.: "A new algorithm for explicit adaptation of time delay," *IEEE Trans. Signal Processing*, 1994, **42**, (7), pp.1816-1820.
- [11] SO, H.C. and CHING, P.C.: "Comparative performances of LMSTDE and ETDE for delay and Doppler estimation," *Proc. of the 28th Annual Asilomar Conference on Signals, Systems, and Computers.*, 1994, **2**, pp.1501-1505, Monterey, USA.
- [12] SO, H.C., CHING, P.C. and CHAN, Y.T.: "An improvement to the explicit time delay estimator," *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 1995, **5**, pp.3151-3154, Detroit, Michigan, USA.
- [13] SO, H.C. and CHING, P.C.: "Performance analysis of ETDGE - an efficient and unbiased TDOA estimator," *IEE Proc. - Radar, Sonar Navig.*, 1998, **6**, pp.325-330.

- [14] QIU, T. and WANG, H. : “An Eckart-weighted adaptive time delay estimation method,” *IEEE Trans. Signal Processing*, 1996, **44**, (9), pp.2332-2335.
- [15] KUO, S.M. and CHUNG, K.M.: “Narrowband active noise control using adaptive delay filter,” *IEEE Signal Proc. Let.*, 1998, **5**, (12), pp.309-311.
- [16] SO, H.C.: “Fast algorithm for nonstationary delay estimation,” *IEE Electronics Letters*, 1998, **34**, (8), pp.722-723.
- [17] HANDEL, P.: “Frequency selective adaptive time delay estimation,” *IEEE Trans. Signal Processing*, 1999, **47**, (2), pp.532-535.
- [18] DOOLEY, S.R. and NANDI, A.K.: “Adaptive time delay and Doppler shift estimation for narrowband signals,” *IEE Proc. - Radar, Sonar Navig.*, 1999, **146**, (5), pp.243-250.
- [19] WIDROW, B. and STEARNS, S.D.: *Adaptive Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1985.
- [20] OPPENHEIM, A.V. and SCHAFER, R.W.: *Digital Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1975, p.29.
- [21] TREICHLER, J.R.: “Transient and convergent behaviour of the adaptive line enhancer,” *IEEE Trans. Acoust., Speech, Signal Processing*, 1979, **27**, (1), pp.53-62.
- [22] GRADSHTEGN, I.S. and RYZHIK, I.M.: *Tables of Integrals, Series and Products*, New York: Academic Press, 1980, p.7.
- [23] BERSHAD, N.J. and QU. L.Z.: “On the probability density function of the LMS adaptive filter weights,” *IEEE Trans. Acoust., Speech, Signal Processing*, 1989, **37**, (1), pp.43-56.
- [24] KROLIK, J., JOU, M., PASUPATHY, S. and EIZENMAN, M: “A comparative study of the LMS adaptive filter versus generalized correlation methods for time delay estimation,” *Proc. ICASSP'84*, 1984, **1**, pp.15.11/1-4.

Table and Figure Captions

Table 1. Computational complexity comparison of the five adaptive delay estimation methods

Figure 1. System block diagram of an adaptive time delay estimator

Figure 2. Delay estimates for a step change in delay at SNR = 10 dB with $P = 15$

Figure 3. Delay estimates for a step change in delay at SNR = 10 dB with $P = 3$

Figure 4. Steady state mean square delay errors at SNR = 10 dB

Figure 5. Delay estimates for a linearly varying delay at SNR = 10 dB with $P = 15$

Figure 6. Delay estimates for a step change in delay at SNR = -10 dB with $P = 15$

Figure 7. Delay estimates for a step change in delay at SNR = -10 dB with $P = 3$

Figure 8. Steady state mean square delay errors at SNR = -10 dB

	operations involved in algorithm		additional requirements
	no. of additions	no. of multiplications	
LMSTDE	$4P + 2$	$4P + 3$	$20P$ additions & $(20P + 10)$ multiplications
CTDE	$2P + 2$	$2P + 3$	one sinc look-up operation
ETDE	$6P + 3$	$6P + 5$	one sinc and one cosine look-up operation
SETDE	$4P + 2$	$4P + 4$	one sinc and one cosine look-up operation
ETDGE	$6P + 4$	$6P + 8$	one sinc and one cosine look-up operation

Table 1:

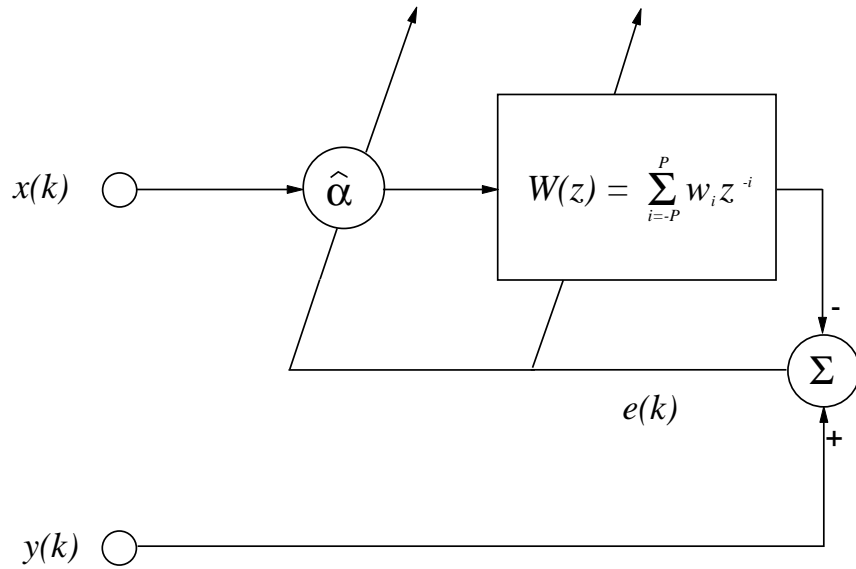


Figure 1:

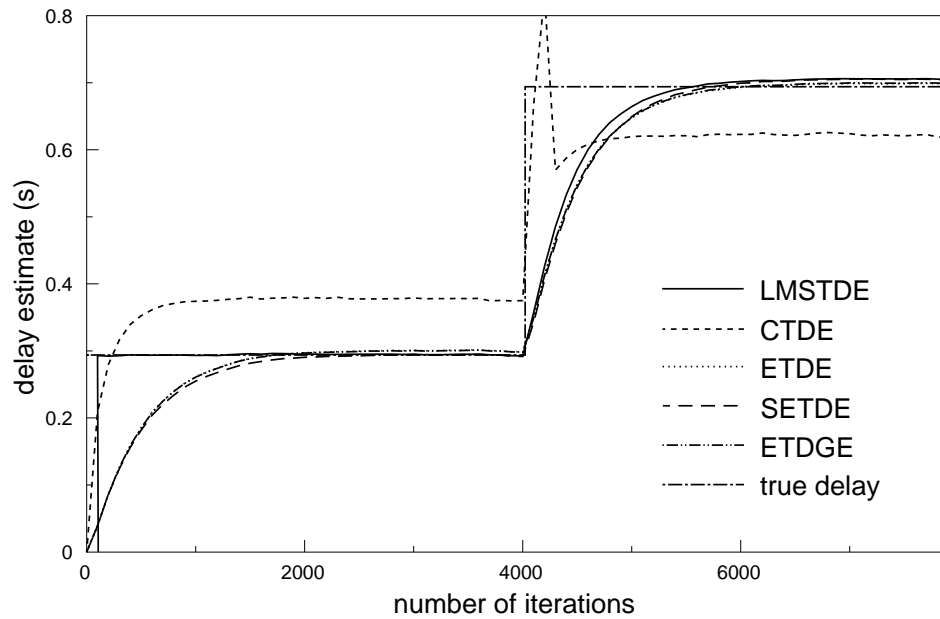


Figure 2:

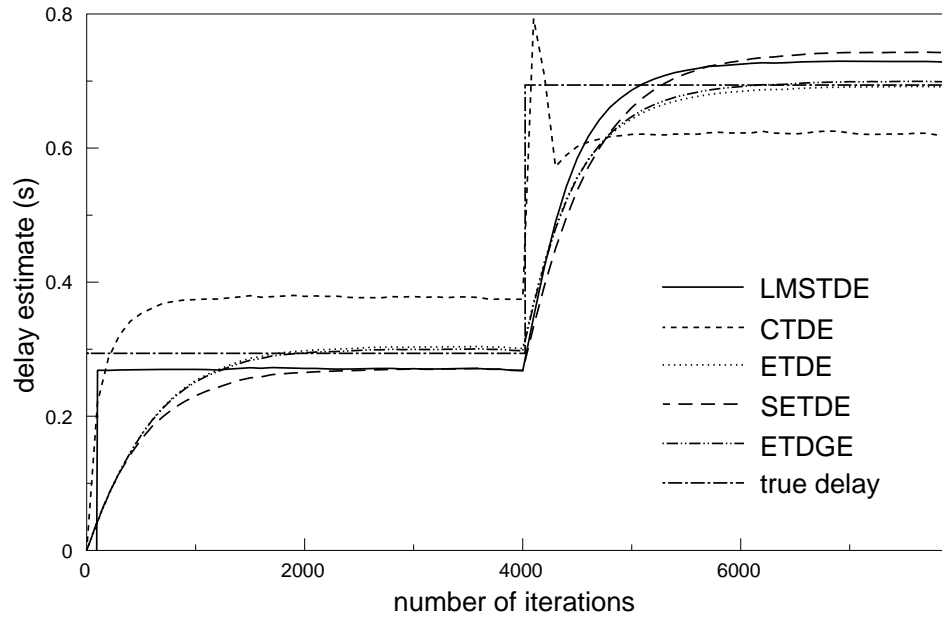


Figure 3:

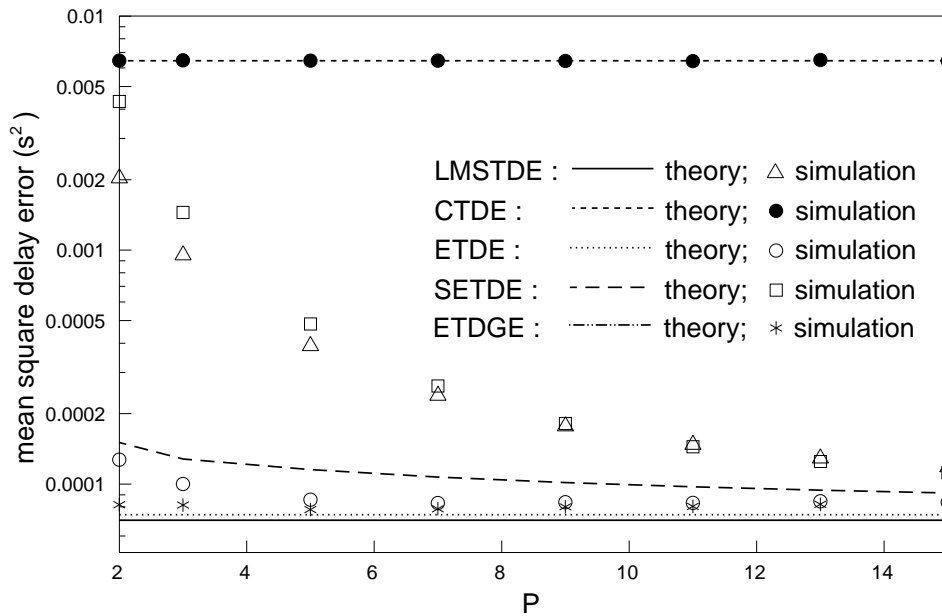


Figure 4:

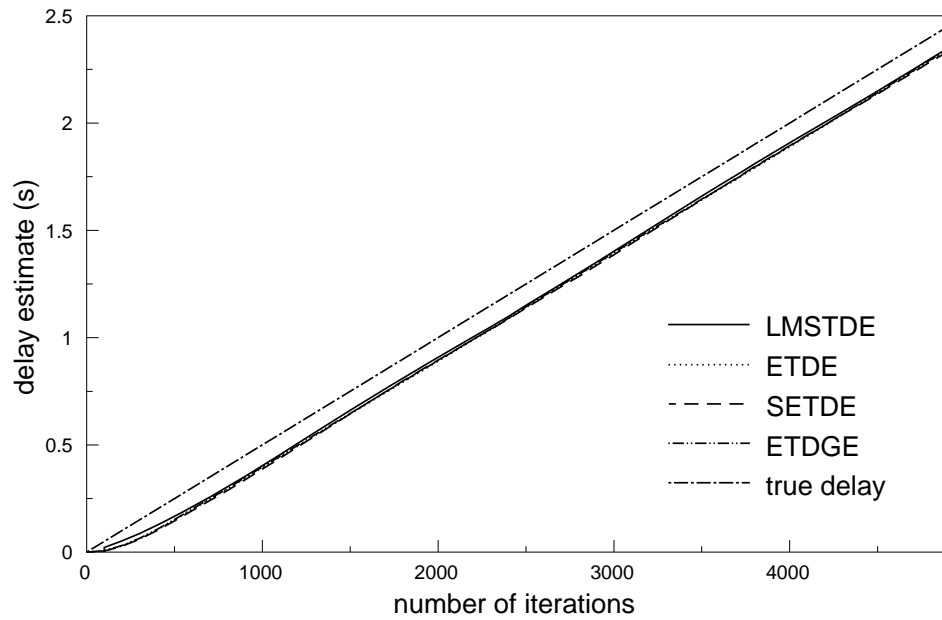


Figure 5:

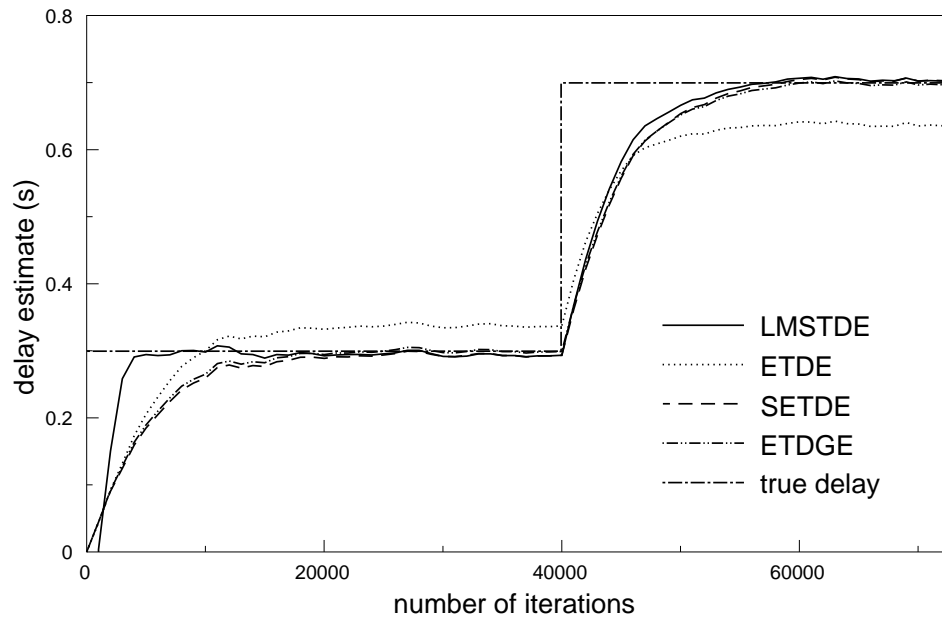


Figure 6:

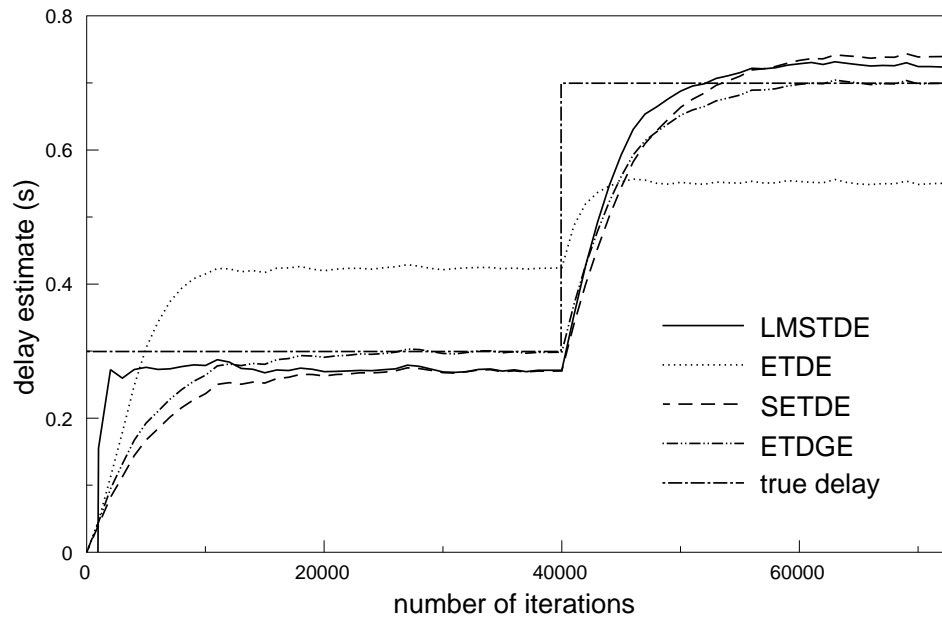


Figure 7:

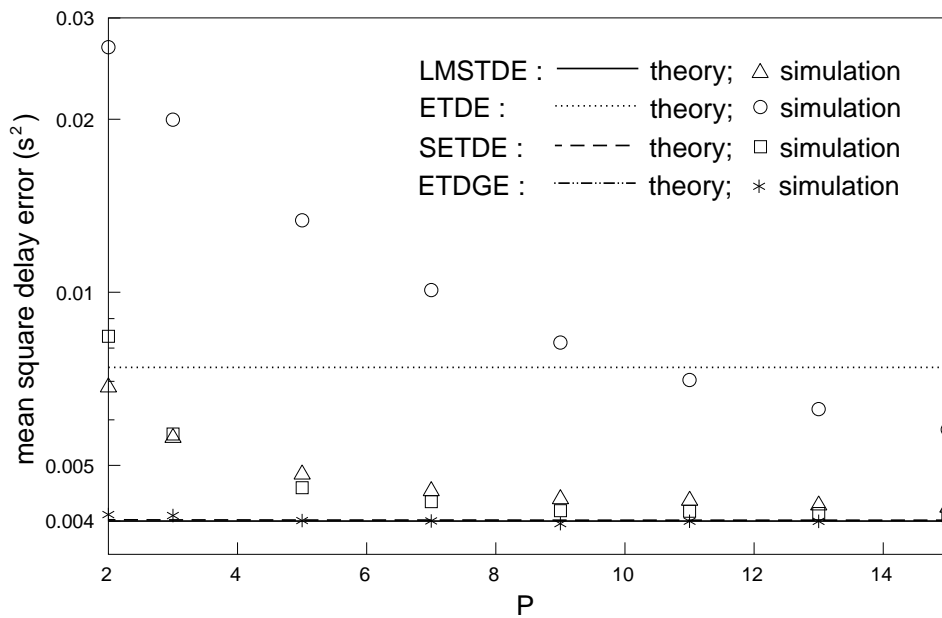


Figure 8: