Fast & Efficient Delay Estimation Using Local All-Pass & Kalman Filters

Beth Jelfs and Christopher Gilliam
RMIT University, Melbourne, Australia
E-mail: {beth.jelfs,christopher.gilliam}@rmit.edu.au

***Abstract*—Delay estimation is a common problem in signal processing which becomes particularly challenging when the delay is time-varying and the recorded signals are non-stationary. While methods for time-varying delay (TVD) estimation exist many of these are based on maximum likelihood estimation and thus are not well suited to real-time implementation. In this paper we present a method for TVD estimation which is suitable for real-time non-stationary applications. The proposed method combines local all-pass (LAP) filters with a Kalman filter. By using measurement fusion to combine the outputs of several LAP filters in the Kalman filter we can accurately track TVDs whilst allowing for fast and efficient parallel computation. Illustrative simulations demonstrate the effectiveness of the proposed approach.**

I. Introduction

Situations where two or more time-dependent signals are recorded from different sensors occur in numerous applications, such as communications [1], radar [2], sonar [3] or the monitoring of complex processes such as biological systems [4]. Often this results in the need to estimate the delay either in receiving the transmitted signal or between receiving signals at spatially separated sensors. For example, delay estimation can occur as a pre-processing step that allows the signals to be aligned or synchronised for further analysis. Alternatively, the estimation of the delay itself can be key to the application as it sheds light on the underlying physical process. Thus, the estimation of a constant delay between two signals is a well studied problem in signal processing [5]. However, as many applications comprise measuring dynamical systems, the delay in question can be time-varying in nature [6], [7]. Accordingly, in this paper, we are interested in the estimation of a time-varying delay (TVD) between two signals. The TVD estimation problem can be described by the delay between two spatially separated sensors (or transmitter and receiver) such that

$$\begin{aligned} x_1(t) &= f(t) + e_1(t) \\ x_2(t) &= f\big(t - \tau(t)\big) + e_2(t) \end{aligned} \tag{1}$$

where $x_1(t)$ and $x_2(t)$ are the signals at each sensor at time t, $f(t)$ is the signal of interest and $\tau(t)$ is the TVD signal to be estimated. Each signal is corrupted by additive noises $e(t)$ that are assumed to be i.i.d Gaussian processes.

For many delay estimation problems, such as in radar and acoustic applications, we require real-time estimates of the delay. Previously we have presented a multiscale Local All-Pass (LAP) framework for TVD estimation which was shown to be both accurate and robust [8], [9]. This framework combines multiple LAP filters with different supports and is thus able to track both small and large delays as well as both fast and slowly varying delays. However, although this approach has a low overall computational cost, the multiple filters are combined sequentially, with post-processing at each stage. Thus, the entire input signals need to be obtained before processing. Importantly, this characteristic is a product of the multiscale architecture not the actual LAP algorithm; the LAP filters themselves require only a short time window from which to estimate the delay at each sample. This short time window combined with the ability to efficiently implement the computation using convolution and pointwise multiplication [10] make the LAP algorithm an ideal choice for applications that require real-time embedded system solutions. Accordingly, there is a need for an implementation of the LAP algorithm that takes advantage of the local nature of the LAP filters whilst maintaining the advantages of the previously proposed multiscale framework.

In this paper we propose to combine LAP filters for TVD estimation with a Kalman filter to provide accurate tracking of the delay. The proposed LAP + Kalman filter combines the benefits of both filters: by using a single LAP filter, we can take full advantage of the speed of computation of the LAP algorithm; whilst the use of a Kalman filter on the output of the LAP filter can improve the accuracy of the estimation. Furthermore the proposed LAP + Kalman filter can use multiple LAP filters, each with different support, to provide different scales, hence preserving the strengths of the multiscale framework. Unlike the multiscale LAP framework these single scale LAP filters can be implemented in parallel to

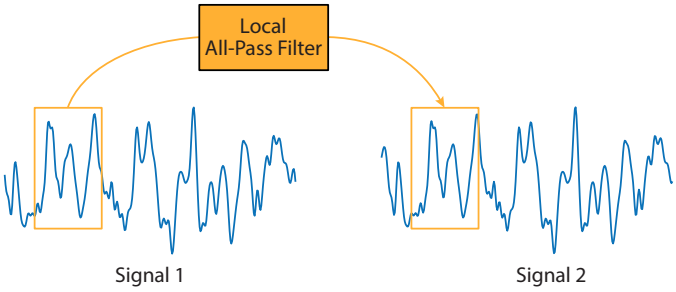

Fig. 1. Diagram illustrating the principle of the LAP algorithm. The local region on the left-hand side is related to the corresponding location on the right-hand side via an all-pass filter.

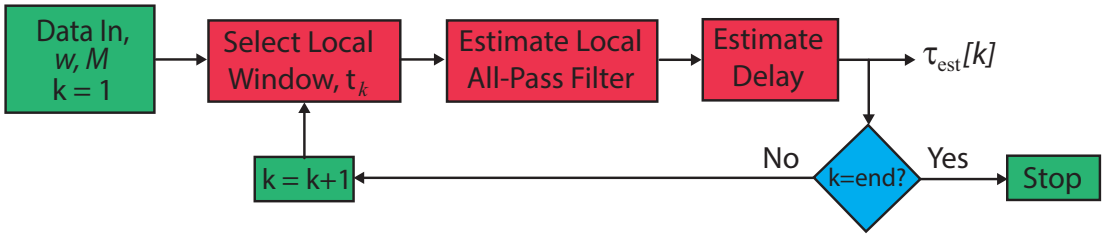

Fig. 2. Flowchart of the single scale LAP algorithm. Note that the 'Estimate Local All-Pass Filter' block equates to solving (4) and the 'Estimate Delay' block equates to solving (6).

maintain the computational speed. The single scale LAP filters can then be combined using a measurement fusion Kalman filter to provide a better estimate of the delay than any of the individual LAP + Kalman filters. Simulations illustrate the efficiency and speed of this proposed combination.

II. Local All-Pass Filters for Delay Estimation

The multiscale LAP framework for TVD estimation was introduced in [8], here we will briefly outline the general LAP algorithm [10] and its use for delay estimation. The main aim of the algorithm is to relate a local region in one signal to the corresponding region in another signal using an all-pass filter; an illustration of which is shown in Fig. 1. This aim follows from the observation that if the TVD signal $\tau(t)$ is constant within a local region then, due to the Fourier shift theorem, a constant delay is equivalent to filtering with an all-pass filter. Thus, the LAP algorithm seeks to estimate an all-pass filter, h, within the local region with a frequency response $H(\omega) = \mathrm{e}^{-j\tau\omega}$. An estimate of the delay is then extracted from the all-pass filter. This process is repeated for every sample, using a sliding-window principle, to obtain a per sample estimate of the TVD signal.

Formally, the estimation of a single filter, h, is as follows. First, for any digital all-pass filter, its 2π-periodic frequency response, $H(\omega)$, can be expressed as

$$H(\omega) = \frac{P\left(\mathrm{e}^{j\omega}\right)}{P\left(\mathrm{e}^{-j\omega}\right)}, \tag{2}$$

where $P\left(\mathrm{e}^{j\omega}\right)$ is the forward and $P\left(\mathrm{e}^{-j\omega}\right)$ the backward version of a real digital filter p. Application of this property results in the all-pass filtering relationship presented in [10] whereby the all-pass filtering operation performed by h can be expressed linearly as a function of p:

$$x_2[k] = h[k] * x_1[k] \iff p[-k] * x_2[k] = p[k] * x_1[k], \tag{3}$$

where $*$ is the convolution operator and k denotes discrete time. The estimation problem is then further reduced by approximating p as a linear combination of a few fixed, known, real filters p_n, i.e. a filter basis. For this paper, we use the filter basis proposed in [8], [10] that comprises a Gaussian function and its first derivative. Such a basis is both compact and scalable, and offers good approximation properties, see the analysis presented in [11] for more details. Accordingly, using (3) and the basis approximation, the LAP algorithm solves the following minimisation:

$$\min_{c} \sum_{k\in\mathcal{W}} \left| p_{\text{app}}[k] * x_1[k] - p_{\text{app}}[-k] * x_2[k] \right|^2, \tag{4}$$

where $\mathcal{W}$ is the local region, c is the coefficient of the filter basis and p_{app} is the filter basis approximation of p defined as:

$$p_{\text{app}}[k] = \mathrm{e}^{-k^2/2\sigma^2} + c\,k\,\mathrm{e}^{-k^2/2\sigma^2}. \tag{5}$$

The variable σ controls the shape of the filters and is defined as $\sigma = M/2 - 0.2$ where M is the integer half support of the filters.

The resulting filter obtained from (4) corresponds to the central sample of the local region. Using a sliding-window principle and solving (4) in each window, the LAP estimates a local all-pass filter per sample. Note that, although such an operation may seem costly, the minimisation in (4) is equivalent to solving a linear system of equations with one unknown and the shifting operation can be implemented using convolution, thus the algorithm can be implemented very efficiently using convolution and pointwise multiplication [10]. The final step in the algorithm is to obtain the estimate of the TVD signal from the all-pass filters. Using the all-pass structure of the filters, the delay estimate can be expressed in terms of the impulse response p_{app} [10]:

$$\tau_{\text{est}} = 2\frac{\sum_k k p_{\text{app}}[k]}{\sum_k p_{\text{app}}[k]}. \tag{6}$$

The complete LAP algorithm is shown in Fig. 2, the algorithm is controlled by two input parameters, the size of the filter basis, M and the size of the window, w, which defines the local region. The size of the delay this algorithm can estimate is dependent upon the size of the filter basis; logically this parameter is the upper bound on the size of the delay the LAP filter can estimate. Thus, to estimate large delays the LAP algorithm requires large filters and as the filter size increases so too does the size of the window, w. The window size defines the region for which the TVD is assumed to be locally constant. This is equivalent to assuming a large delay is slowly varying. However, the window size also affects the accuracy of the delay estimate; solving (4) is equivalent to solving a linear system of equations thus using a larger window size is equivalent to having an over-determined system, which is useful in noisy conditions.

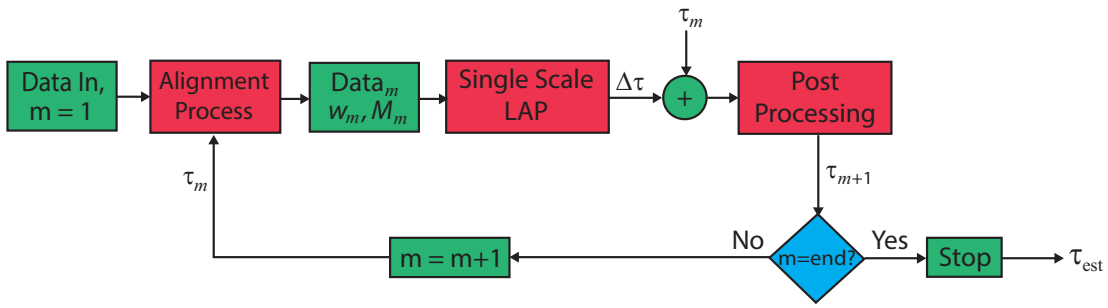

Fig. 3. Flowchart of the multiscale LAP algorithm. The 'Single Scale LAP' block equates to the flowchart illustrated in Fig. 2. Note that m is the scale index for the algorithm and $\Delta\tau$ is the refinement of the delay from the previous estimate.

The minimum size of window is dictated by the size of the LAP filter, for a given half support, M, the minimum window size is

$$w_{min} = 2M + 1. \tag{7}$$

However, there is no theoretical limit to the maximum size of the window. Thus a trade-off exists: larger windows enable more accurate delay estimation but at the cost of restricting the amount of time-variation in the delay. In contrast, smaller windows allow faster time-variation but can result in noisy delay estimates. Smaller windows also restrict the size of the delay which can be estimated by restricting the LAP filter size. Previously, to enable the estimation of both quickly and slowly varying delays, we proposed a multiscale framework for TVD estimation [8] which implements several different values of M sequentially, as illustrated in Fig. 3.

III. Kalman Filter Fusion

The multiscale framework for TVD estimation proposed in [8] operates in an iterative manner. First using the largest value of M to estimate the delay and then using this estimate to warp the delayed signal closer to the original signal. The process is then repeated with the next value of M and so on for each of the desired scales. While this provides accurate results, it also requires the whole signal in advance to perform the post-processing and alignment at each stage of the algorithm, as shown in Fig. 3.

In contrast, using single scales of the LAP filter, as shown in Fig. 2, does not have the same restriction and outputs a per sample estimate of the delay as only the samples used in the local region, $\mathcal{W}$ are required. By specifying the support of the LAP we define the maximum size of the delay which can be estimated. Note that the choice of M can be informed by our prior knowledge of the application at hand. Thus, the remaining parameter is the choice of w, the size of the local region. As previously highlighted, this has a trade-off: the accuracy of the estimate versus the amount of variation permissible in the delay. Ideally we want to allow the maximum possible variation whilst maintaining accuracy.

To enable the use of shorter window sizes, in this paper we combine the LAP filter with a Kalman filter [12] as illustrated in Fig. 4. We assume the delays estimated by the LAP algorithm are noisy measurements of the true TVD and use the Kalman filter's ability to model both this noise and structure of the TVD to obtain a more accurate estimate of the true delay. Hence estimating the TVD using a LAP + Kalman filter allows us to reduce the size of the local region in the LAP algorithm to that of the minimum possible for the specified size of delay (7) without significant loss in accuracy.

To implement the Kalman filter the state vector, based on the LAP estimate of the delay, τ_{LAP}, is assumed to be

$$\boldsymbol{\tau}_k = \begin{pmatrix} \tau_k \\ \dot{\tau}_k \\ \ddot{\tau}_k \end{pmatrix}, \tag{8}$$

with the process governed by the following equations:

$$\boldsymbol{\tau}_k = A_k \boldsymbol{\tau}_{k-1} + u_k \tag{9}$$

$$\tau_{\text{LAP}_k} = C_k \boldsymbol{\tau}_k + v_k, \tag{10}$$

where u and v are independent Gaussian noise processes. For a given sampling period Δt, the transition and measurement matrices are given by

$$A_k = \begin{pmatrix} 1 & \Delta t & \Delta t^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{pmatrix} \tag{11}$$

$$C_k = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \tag{12}$$

respectively. Using these equations we can define the Kalman filter updates as [13]

$$\boldsymbol{\tau}_{k|k-1} = A_k \boldsymbol{\tau}_{k-1} \tag{13}$$

$$P_{k|k-1} = A_k P_k A_k^T + Q_k \tag{14}$$

$$K = P_{k|k-1} C_k^T \left(C_k P_{k|k-1} C_k^T + R_k \right)^{-1} \tag{15}$$

$$\boldsymbol{\tau}_k = \boldsymbol{\tau}_{k|k-1} + K \left(\tau_{\text{LAP}_k} - C_k \boldsymbol{\tau}_{k|k-1} \right) \tag{16}$$

$$P_k = \left(I - K C_k \right) P_{k|k-1}, \tag{17}$$

where P is the state estimate covariance matrix, K is the Kalman gain, and Q and R are the process and measurement noise covriances respectively.

The LAP + Kalman filter combination provides an accurate estimate of the delay but is still limited by the size of the half support of the LAP filter. To overcome this and gain the benefits of the smaller window size whilst also maintaining

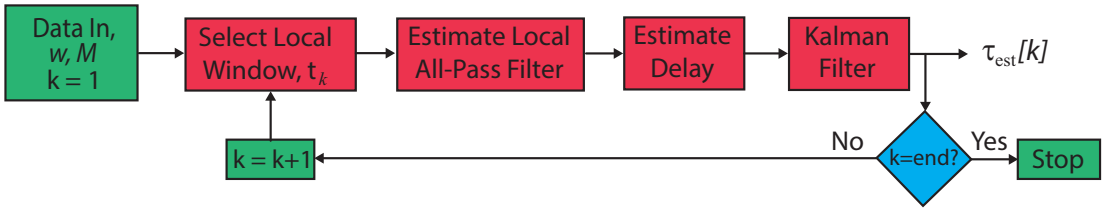

Fig. 4. Flowchart of the LAP + Kalman algorithm.

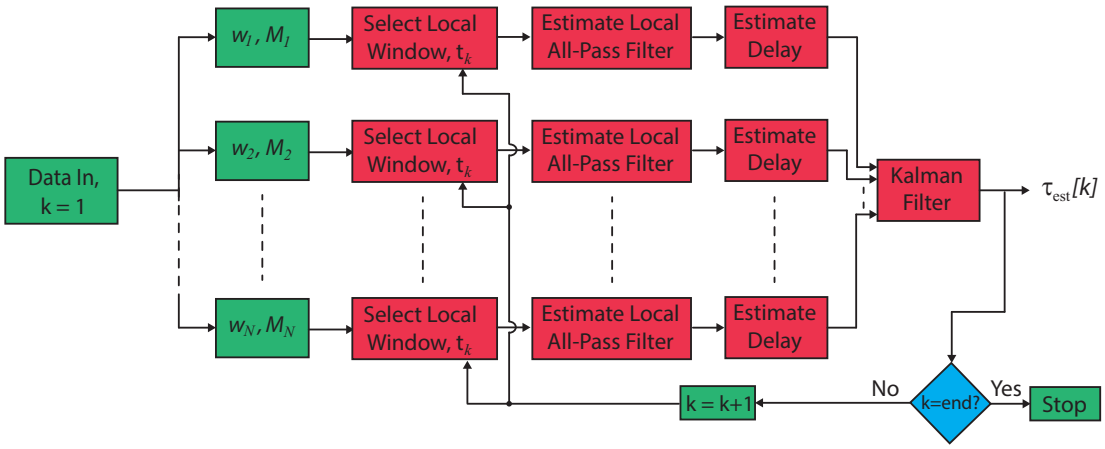

Fig. 5. Flowchart of the LAP + Kalman filter fusion algorithm.

the advantages of the multiscale LAP, we propose to fuse multiple LAP filters using a Kalman filter. In this way different values of M can be implemented separately as demonstrated in Fig. 5, and crucially, unlike the multiscale LAP these filters can be implemented in parallel providing fast and efficient computation.

Several approaches to fusion using Kalman filters have been proposed, primarily state vector fusion which produces filtered state vectors for each of the measurements and then combines to give an updated estimate [14] and measurement fusion which first combines the measurements and then updates the state vector [15]. Despite modifications to state vector fusion which take into account the correlation of the process noise, measurement fusion has been shown to be preferable to state vector fusion [16]. Measurement fusion can be obtained either by augmenting the observation vector [15] or by weighting the observations [16]. In [15] the two were shown to be functionally equivalent for identical measurement matrices (as is the case here). For simplicity, we have implemented an augmented observation as follows:

$$\tau_{\mathrm{LAP}_k} = \left[\tau^1_{\mathrm{LAP}_k} \cdots \tau^N_{\mathrm{LAP}_k}\right]^T \tag{18}$$

$$C_k = \left[C^1_k \ldots C^N_k\right]^T \tag{19}$$

$$R_k = \mathrm{diag}\left[R^1_k \ldots R^N_k\right], \tag{20}$$

where N is the number of LAP filters to be fused. Having formed the augmented measurements the Kalman filter can then be implemented as before (14)–(17).

IV. Simulation Results

A. Comparison with Multiscale LAP

To test the efficacy of the LAP + Kalman combination we first compared the results with those obtained from the multiscale LAP. In [8] the multiscale framework was tested on synthetic signals which have previously been used in the investigation of muscle fibre conduction velocity from surface electromyography signals [17], [18]. In brief, the first channel is generated by filtering white Gaussian noise through a FIR filter with a known spectrum and the delayed version obtained by interpolating the first channel with the desired delay, $\tau(t)$. In this case the delay was represented as the reciprocal of a sinusoid. The signals are then corrupted by additive white Gaussian noise and low-pass filtered to simulate the response of the acquisition device. Previously, the multiscale framework has been shown to provide a more accurate and robust estimate of the TVD for this type of signal compared to alternative methods [8].

Figure 6 provides an example of the delay estimates obtained from 5 seconds of synthetic data with a sampling rate $F_s = 2048$Hz using a single scale LAP algorithm with $M = 8$ ($w = 2M + 1$ in all simulations) and the LAP + Kalman filter. As can be seen the addition of the Kalman filters gives a smoother estimate of the delay than the LAP alone whilst also providing accurate tracking of the TVD. Comparing the LAP and the LAP + Kalman now with the multiscale LAP with $M = \{2, 4, 8\}$ and $w = 512$ we can observe that the multiscale LAP provides a more accurate estimate of the delay, as is to be expected. Figure 7 shows the average mean absolute error

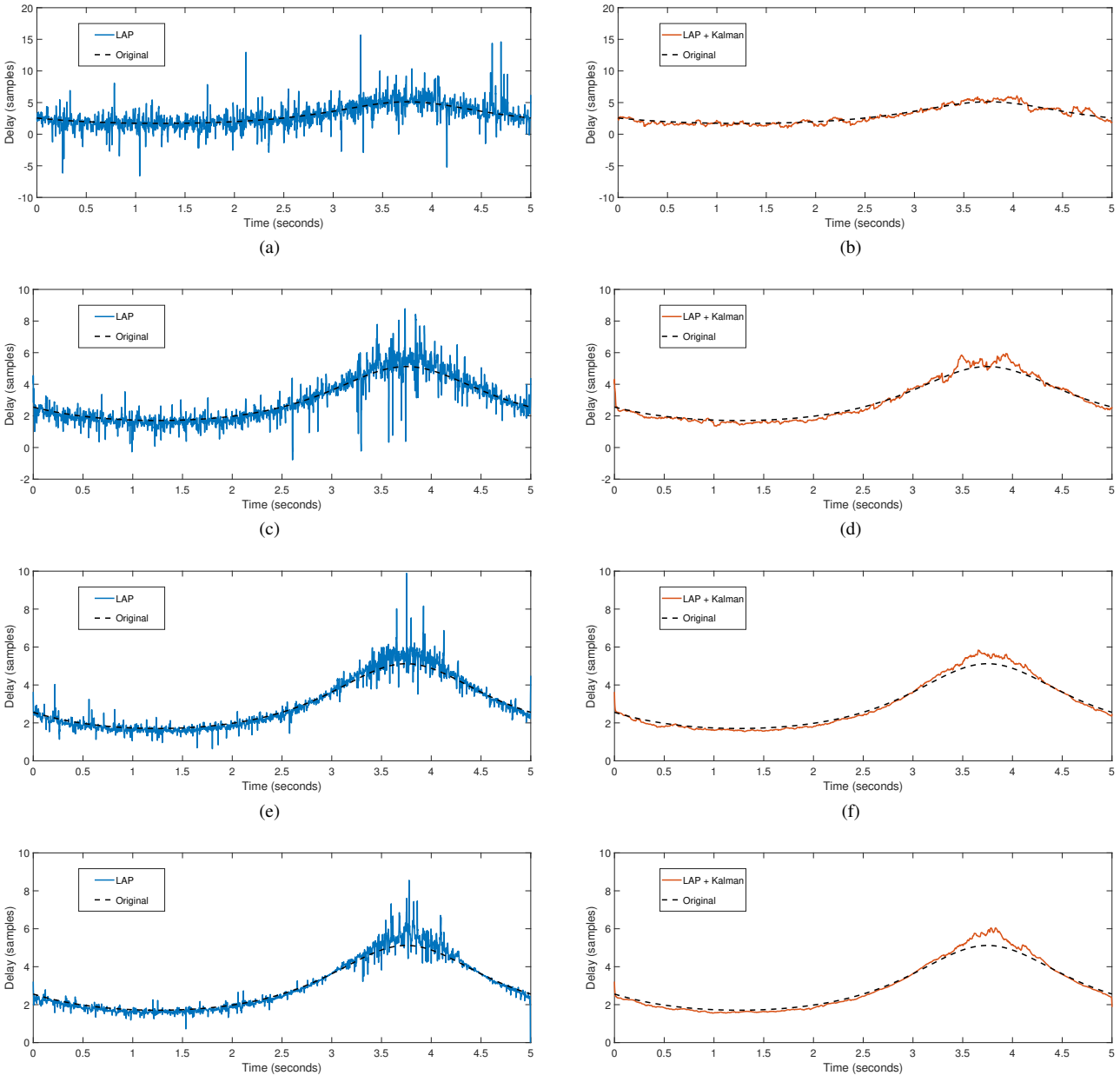

Fig. 6. Example delay estimates from the LAP and LAP + Kalman for different noise levels. Row 1 SNR = 10dB, row 2 SNR = 20dB, row 3 SNR = 30dB, row 4 noiseless data. Left-hand column LAP and right-hand column LAP + Kalman.

obtained from 100 realisations of the synthetic data. Although the multiscale LAP is more accurate than the LAP + Kalman filter, the results from Fig. 6 and Fig. 7 indicate that the LAP + Kalman is still a good estimator of the TVD.

The notable advantage of the LAP + Kalman filter arises in the computation, Table I provides the average computation time to process 5 seconds of data and the latency of the algorithms. Note that we consider latency to be the time taken to provide an estimate of the current delay; it is not a product of hardware speed rather it dependents on the algorithm's architecture. While the LAP + Kalman and multiscale LAP have very similar computation time, because of the alignment and post-processing performed at each iteration of the multi-scale LAP the algorithm requires the full 5 seconds of data to perform the processing. On the other hand the LAP, and hence the LAP + Kalman filter, can start outputting per sample delay estimates as soon as they have the first window of data, which results in a latency of only 8.301ms ($w = 17$ samples at $F_s = 2048$Hz).

B. *Speech*

To provide a more realistic example we next constructed an example using a real world signal and introducing a known

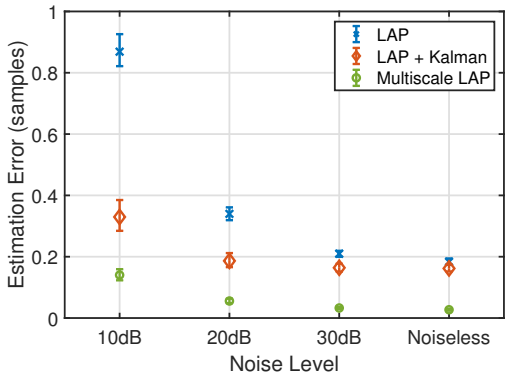

Fig. 7. Average errors in the delay estimates for different noise levels. Error bars indicate the 5th and 95th quantiles.

TABLE I
COMPARISON OF TOTAL COMPUTATION TIME TO PROCESS 5 SECONDS OF DATA AND ALGORITHM LATENCY FOR THE LAP, LAP + KALMAN AND MULTISCALE LAP.

	Computation Time (ms)	Latency (ms)
LAP	2.9	8.3
LAP + Kalman	34.5	8.3
Multiscale LAP	35.9	5000.0

All calculations performed in MATLAB (Mathworks inc.) on a HP Workstation Intel Xeon W2133 3.6 6 Core 64GB RAM.

delay. To illustrate the advantage of using the measurement fusion with the Kalman filter combined with the speed of computation and the short latency of the LAP we used a short 610.4ms speech signal (Fig. 8a) with 5000 samples (sampling rate of 8192Hz). Similar to the approach in [6] we introduce a linearly decreasing delay in this case decreasing from 8 samples to 1.5 samples until 3500 samples and then constant for the remaining samples (Fig. 8b). The speech signal itself is non-stationary and comprised of several different frequency components and by introducing a non-constant delay the estimation problem is not straightforward as can be seen from the illustrative segments of the original and delayed signals shown in Fig. 9.

Due to the complexity of the estimation problem in this case we implemented two LAP filters one with a value of $M = 8$ and the other with $M = 16$, the outputs of which were then fused using the Kalman filter. Figure 10 shows the estimates obtained by using the individual LAP + Kalman filters and the output of Kalman filter performing measurement fusion on the two LAP filters. All three algorithms successfully track the changes in the delay with Table II giving the associated mean absolute errors. Table II demonstrates the improvement in the delay estimation obtained by using the LAP + Kalman compared to the LAP alone and shows the fused algorithm provides smaller errors than either of the individual LAP + Kalman filters. However, as can be seen from Fig. 10 the errors in the estimation from the fused algorithm are still relatively larger at the start of the signal and around the 350-400ms. If

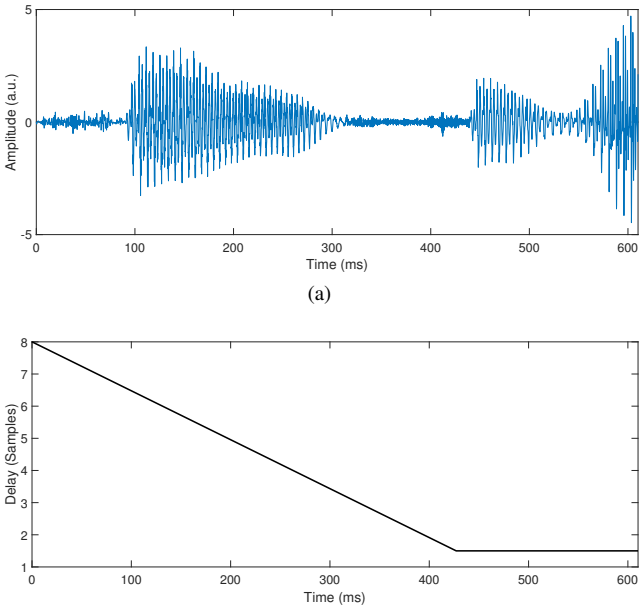

Fig. 8. (a) Original speech signal. (b) The introduced delay.

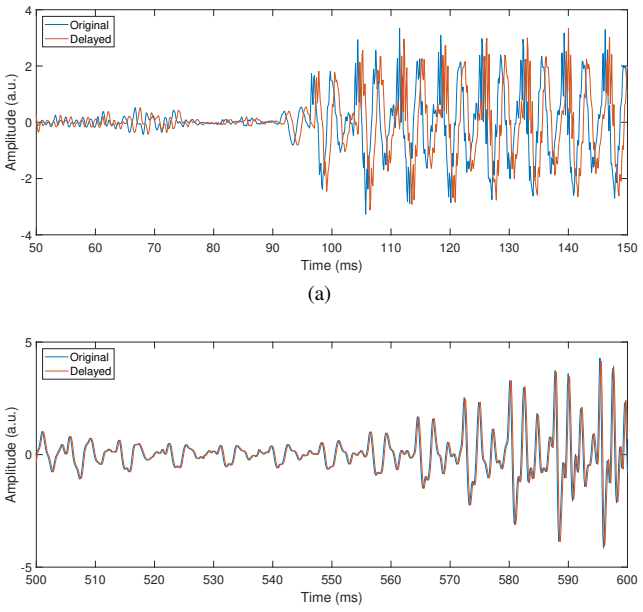

Fig. 9. Segments of the original and delayed speech signals. (a) 50–150ms corresponding to delays of 7.2–5.7 samples. (b) 500–600ms corresponding to a constant delay of 1.5 samples.

we consider the same regions in the speech signal, shown in Fig. 8a, it can be noted that these correspond to the smaller low amplitude sections of the signal where the difficulty in the estimation of the delay increases. Despite this the overall error obtained from the fused algorithm is still comparatively small.

V. CONCLUSIONS

In this paper we have presented an algorithm for delay estimation which combines local all-pass filters with a Kalman

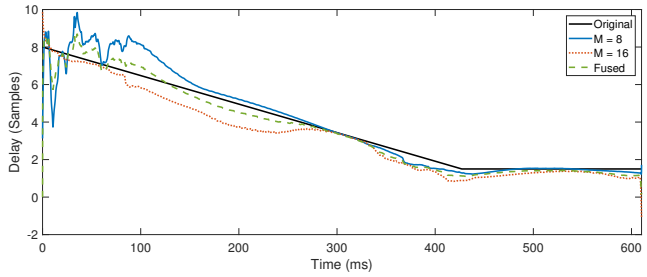

Fig. 10. The original delay and the estimates obtained from the LAP + Kalman filter with $M = 8$ and $M = 16$, and the fused combination of the two.

TABLE II
MEAN ABSOLUTE ERRORS OBTAINED FROM THE LAP AND LAP + KALMAN FILTERS WITH $M = 8$ AND $M = 16$, AND THE FUSED COMBINATION OF THE TWO.

	LAP	LAP + Kalman
$M = 8$	1.001	0.408
$M = 16$	0.620	0.509
fused	–	0.310

filter to track time-varying delays. While the results presented here do not provide the same level of accuracy as our previously proposed multiscale framework they illustrate that the proposed algorithm can still obtain acceptable errors with a much lower latency. The proposed LAP + Kalman filter can also take advantage of measurement fusion using the Kalman filter to combine the estimates of multiple single scale LAP filters to provide a better estimate than any of the constituent filters. The key advantage of the proposed approach is in providing an alternative implementation which can take advantage parallel processing to give a fast and efficient implementation for real-time applications.

REFERENCES

[1] J. Chen, Y. Huang, and J. Benesty, "Time delay estimation," in *Audio Signal Processing for Next-Generation Multimedia Communication Systems*, Y. Huang and J. Benesty, Eds. Kluwer Academic Publishers, 2004, pp. 197–227.

[2] K. V. Mishra and Y. C. Eldar, "Performance of time delay estimation in a cognitive radar," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, USA, 2017, pp. 3141–3145.

[3] G. Carter, "Time delay estimation for passive sonar signal processing," *IEEE Trans. Acoust., Speech, and Signal Process.*, vol. 29, no. 3, pp. 463–470, 1981.

[4] T. Müller, M. Lauk, M. Reinhard, A. Hetzel, C. H. Lücking, and J. Timmer, "Estimation of delay times in biological systems," *Ann. Biomed. Eng.*, vol. 31, no. 11, pp. 1423–1439, 2003.

[5] G. C. Carter, "Coherence and time delay estimation," *Proc. IEEE*, vol. 75, no. 2, pp. 236–255, 1987.

[6] J.-F. Stumper and A. Rosich, "Online estimation of a time-varying delay based on a univariate cross-ambiguity function analysis," in *Proc. 23rd European Signal Processing Conf. (EUSIPCO)*, Nice, France, 2015, pp. 1471–1475.

[7] F. Viola and W. F. Walker, "A spline-based algorithm for continuous time-delay estimation using sampled data," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 52, no. 1, pp. 80–93, 2005.

[8] C. Gilliam, A. Bingham, T. Blu, and B. Jelfs, "Time-varying delay estimation using common local all-pass fitlers with application to surface electromyography," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, Canada, 2018, pp. 841–845.

[9] C. Gilliam and B. Jelfs, "Estimating muscle fibre conduction velocity in the presence of array misalignment," in *Proc. Asia-Pacific Signal and Information Processing Assoc. Annu. Summit and Conf. (APSIPA-ASC)*, Honolulu, HI, USA, 2018, pp. 853–860.

[10] C. Gilliam and T. Blu, "Local all-pass geometric deformations," *IEEE Trans. Image Process.*, vol. 27, no. 2, pp. 1010–1025, 2018.

[11] T. Blu, P. Moulin, and C. Gilliam, "Approximation order of the LAP optical flow algorithm," in *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, Québec City, Canada, 2015, pp. 48–52.

[12] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME–J. Basic Eng.*, vol. 82, no. Series D, pp. 35–45, 1960.

[13] M. S. Grewal and A. P. Andrews, *Kalman filtering: theory and practice with MATLAB*, 4th ed., ser. Wiley – IEEE. Hoboken, New Jersey: Wiley, 2015.

[14] Y. Bar-Shalom, "On the track-to-track correlation problem," *IEEE Trans. Automat. Contr.*, vol. 26, no. 2, pp. 571–572, 1981.

[15] Q. Gan and C. J. Harris, "Comparison of two measurement fusion methods for Kalman-filter-based multisensor data fusion," *IEEE Trans. on Aerosp. Electron. Syst.*, vol. 37, no. 1, pp. 273–279, 2001.

[16] J. Roecker and C. McGillem, "Comparison of two-sensor tracking methods based on state vector fusion and measurement fusion," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 24, no. 4, pp. 447–449, 1988.

[17] P. Ravier, D. Farina, and O. Buttelli, "Time-varying delay estimators for measuring muscle fiber conduction velocity from the surface electromyogram," *Biomed. Signal Process. Control*, vol. 22, pp. 126–134, 2015.

[18] A. Boualem, M. Jabloun, P. Ravier, and O. Buttelli, "Legendre polynomial modeling of time-varying delay applied to surface EMG signals—derivation of the appropriate time-dependent CRBs," *Signal Process.*, vol. 114, pp. 34–44, 2015.