An Adaptive All-Pass Filter for Time-Varying Delay Estimation

Beth Jelfs, Shuai Sun, Kamran Ghorbani, and Christopher Gilliam

School of Engineering, RMIT University, Australia

email: {beth.jelfs, shuai.sun2, kamran.ghorbani}@rmit.edu.au, dr.christopher.gilliam@ieee.org

RMIT UNIVERSITY

Summary

Estimation of a delay between two signals is a common problem and particularly challenging when the delay is non-stationary. Our proposed solution is based on an all-pass filter framework and an adaptive filtering algorithm with an LMS style update that estimates the delay from the filter coefficients. We validate the filter on synthetic data demonstrating that it is both accurate and capable of tracking time-varying delays.

Delay Estimation

1. Fourier Shift Theorem

Constant delay $\Longrightarrow$ Multiplication by complex exponential in frequency domain

$\looparrowright$ Equivalent to filtering with filter h with frequency response [1]:

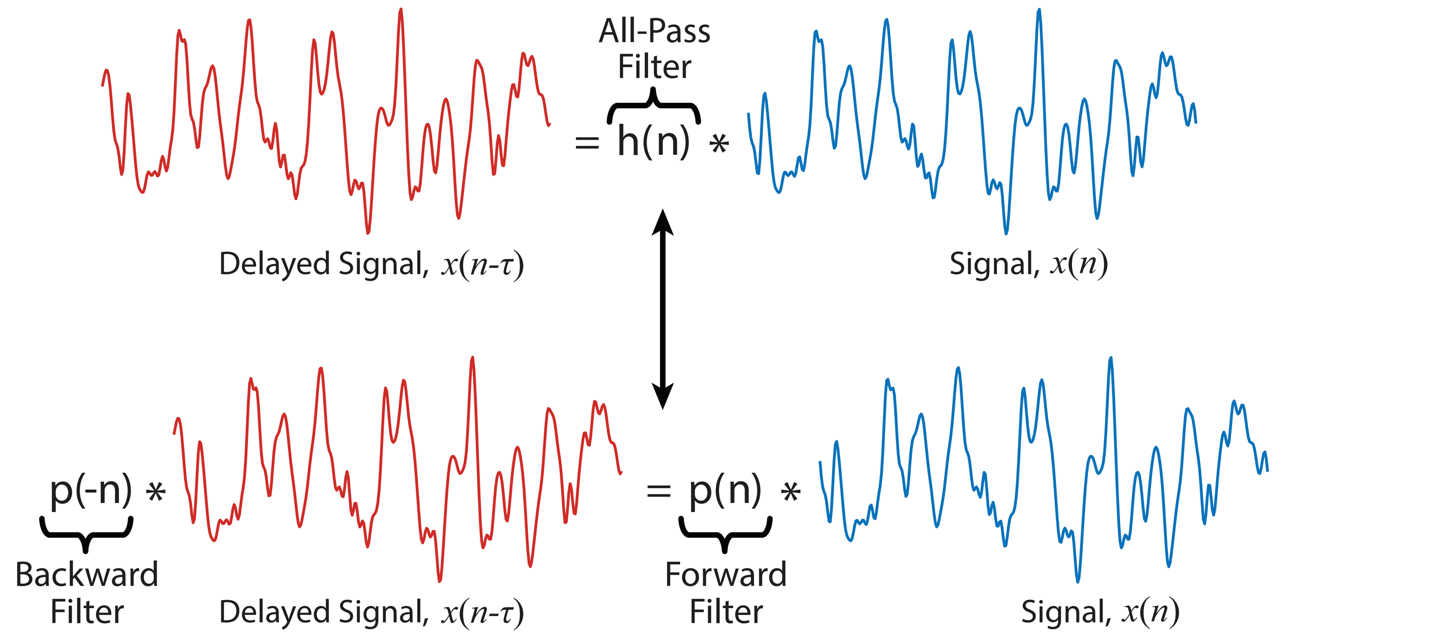
$$H(\omega) = \mathrm{e}^{-j\tau\omega} \quad \longleftrightarrow \quad \text{All-Pass Filter}$$

2. Rational Representation of All-Pass Filter

The (2π)-periodic frequency response of any digital all-pass filter can be expressed as:

$$H(\omega) = \frac{P\left(\mathrm{e}^{j\omega}\right)}{P\left(\mathrm{e}^{-j\omega}\right)} \quad \begin{array}{l} \longleftrightarrow \ \text{Forward Filter} \\ \longleftrightarrow \ \text{Backward Filter} \end{array}$$

Linearise filtering: $x(n-\tau) = h(n) * x(n) \quad \Longleftrightarrow \quad p(-n) * x(n-\tau) = p(n) * x(n)$



3. Filter with Finite Support

Filter response can be described by:

$$\text{FIR filter with finite support } k \in [0, K] \quad \Longrightarrow \quad p(k) = \begin{cases} a_k, & 0 \le k \le K \\ 0, & \text{otherwise} \end{cases}$$

$$p(n) * x(n) = \sum_{k=0}^{K} a_k x(n-k) \quad \looparrowright \quad \sum_{k=0}^{K} a_k x(n+k-\tau) = \sum_{k=0}^{K} a_k x(n-k)$$

Problem becomes estimating the filter coefficients $\{a_k\}_{k=0,\dots,K}$

4. Extracting the Delay

K defines the maximum delay which can be estimated by the filter

$$\text{Evaluate} \quad \left.\frac{\mathrm{d}\, H(\omega)}{\mathrm{d}\,\omega}\right|_{\omega=0} \quad \Longrightarrow \quad \hat{\tau} = 2\frac{\sum_k k a_k}{\sum_k a_k}$$

References

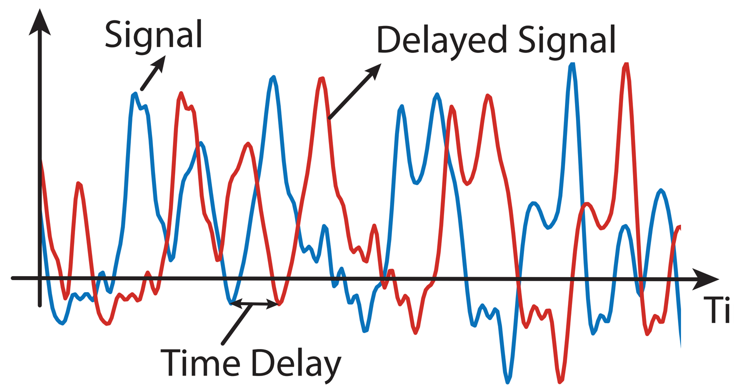
[1] C. Gilliam, A. Bingham, T. Blu, and B. Jelfs, "Time-varying delay estimation using common local all-pass filters with application to surface electromyography," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, Canada, pp. 841–845, 2018.

[2] H. So, P. Ching, and Y. Chan, "A new algorithm for explicit adaptation of time delay," *IEEE Trans. Signal Process.*, vol. 42, no. 7, pp. 1816–1820, 1994.

[3] X. Sun and S. Douglas, "Adaptive time delay estimation with allpass constraints," in *Proc. Asilomar Conf. Signals, Systems, and Computers*, Pacific Grove, CA, USA, 199, vol. 2, pp. 898–902.

Our Approach

Assume two spatially separated sensors both receiving the same signal but with a delay between them



At sample time n, sensor 1 receives $x(n)$ and sensor 2 receives $x(n-\tau)$

$$\text{Equivalent to: } a_0 = 1 \ \looparrowright \ x(n-\tau) - x(n) = \sum_{k=1}^{K} a_k x(n-k) - \sum_{k=1}^{K} a_k x(n+k-\tau)$$
$$= \mathbf{x}_-^T(n)\mathbf{a} - \mathbf{x}_+^T(n-\tau)\mathbf{a}$$

$$\mathbf{a} = [a_1, \dots, a_K]^T$$
$$\mathbf{x}_-(n) = \left[x(n-1), \dots, x(n-K)\right]^T \quad \longleftrightarrow \text{ backward vector of sensor 1}$$
$$\mathbf{x}_+(n-\tau) = \left[x(n+1-\tau), \dots, x(n+K-\tau)\right]^T \quad \longleftrightarrow \text{ forward vector of sensor 2}$$

Express samples from each sensor as a linear predictor of samples from the other sensor

$$x(n) = \mathbf{x}_+^T(n-\tau)\mathbf{a} \qquad x(n-\tau) = \mathbf{x}_-^T(n)\mathbf{a}$$

Adaptive All-Pass Filter

1. Definition

Difference between the noisy outputs of the optimum filter

$$\text{Desired response:} \quad d(n) = \mathbf{x}_-^T(n)\mathbf{a} + \eta_1(n) - \mathbf{x}_+^T(n-\tau)\mathbf{a} - \eta_2(n)$$

$\eta_1(n)$, $\eta_2(n) \Longrightarrow$ zero mean i.i.d. noise sources with variance σ_η^2

Current estimate of the true filter coefficients $\mathbf{a} \looparrowright \mathbf{w}(n) = [w_1, w_2, \dots, w_K]^T$

$$\text{Filter output:} \quad y(n) = \left[\mathbf{x}_-^T(n) - \mathbf{x}_+^T(n-\tau)\right]\mathbf{w}(n)$$

Our problem $\Longrightarrow$ minimize the error between measured samples, $d(n)$ and estimates obtained from filter coefficients, $y(n)$

$$e(n) = d(n) - y(n) = \left[\mathbf{x}_-^T(n) - \mathbf{x}_+^T(n-\tau)\right]\mathbf{a} - \left[\mathbf{x}_-^T(n) - \mathbf{x}_+^T(n-\tau)\right]\mathbf{w}(n) + \eta_1(n) - \eta_2(n)$$

2. Update

Using steepest descent with learning rate μ and cost function $\mathcal{J}(n) = |e(n)|^2$

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \nabla \mathcal{J}(n)\big|_{w=w(n)} = \mathbf{w}(n) + 2\mu e(n)\left[\mathbf{x}_-^T(n) - \mathbf{x}_+^T(n-\tau)\right]$$

Define residuals $\looparrowright$ $\mathbf{r}(n) = \mathbf{x}_-(n) - \mathbf{x}_+(n-\tau)$ and $\eta(n) = \eta_1(n) - \eta_2(n)$

$$e(n) = \mathbf{r}^T(n)\mathbf{a} - \mathbf{r}^T(n)\mathbf{w}(n) + \eta(n)$$
$$\mathbf{w}(n+1) = \mathbf{w}(n) + 2\mu e(n)\mathbf{r}^T(n)$$

Gives adaptive all-pass filter in the form of the standard LMS algorithm for input $\mathbf{r}(n)$

3. Convergence

For convergence in the mean square the bound on the learning rate is given by:

$$0 < \mu < \frac{1}{3\mathrm{tr}[\mathbf{R}]} \qquad \text{where } \mathrm{tr}[\cdot] \text{ is the trace of the matrix}$$

Note the correlation $\mathbf{R} = \left[\mathbf{x}_-(n) - \mathbf{x}_+(n-\tau)\right]\left[\mathbf{x}_-(n) - \mathbf{x}_+(n-\tau)\right]^T$ is based on the difference between the forward and backward vectors from the two sensors.

Normalised Adaptive All-Pass Filter

$$\mathrm{tr}[\mathbf{R}] = \mathbf{r}^T(n)\mathbf{r}(n) = \sum_{i=1}^{K} r^2(n-i) \quad \text{and} \quad r(n-i) = x(n-i) - x(n+i-\tau)$$

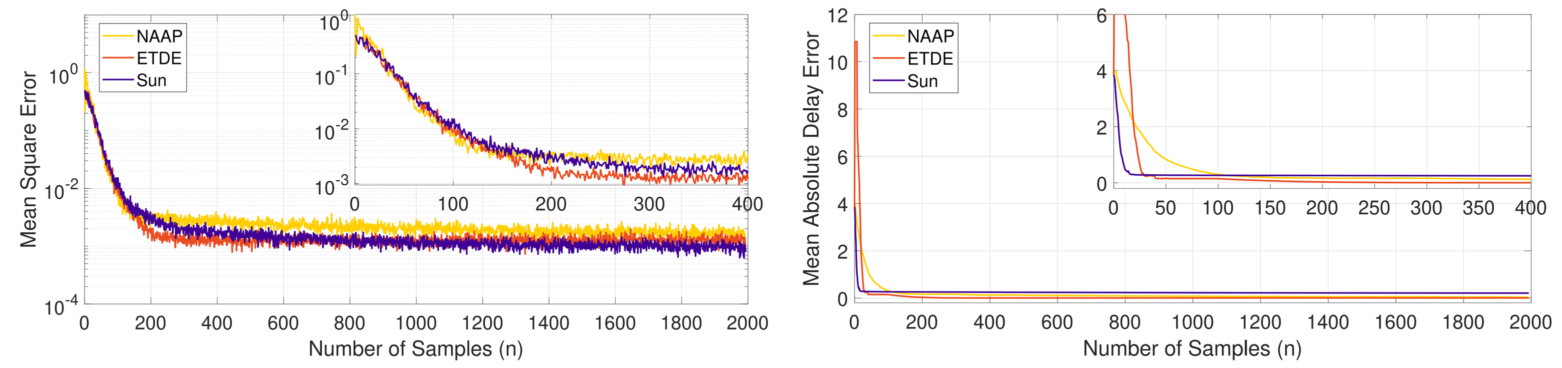
$$\text{Leads to bound on the learning rate:} \quad 0 < \mu < \frac{1}{3\|\mathbf{x}_-(n) - \mathbf{x}_+(n-\tau)\|_2^2}$$

$$\textbf{NAAP filter:} \quad \mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\rho}{\|\mathbf{x}_-(n) - \mathbf{x}_+(n-\tau)\|_2^2 + \varepsilon} e(n)\mathbf{r}(n)$$

Results

1. Constant Delay

Estimation of a constant delay of $\tau(n) = 5.85$ samples



(a) Average evolution of the MSE (b) Evolution of the mean absolute delay error, in samples

Figure 1: Comparison of the performance of NAAP ($\rho = 0.08$), the ETDE [2] ($\mu = 0.04$) and Sun algorithm [3] ($\mu = 0.02$), insets: the first 400 samples

2. Tracking Performance

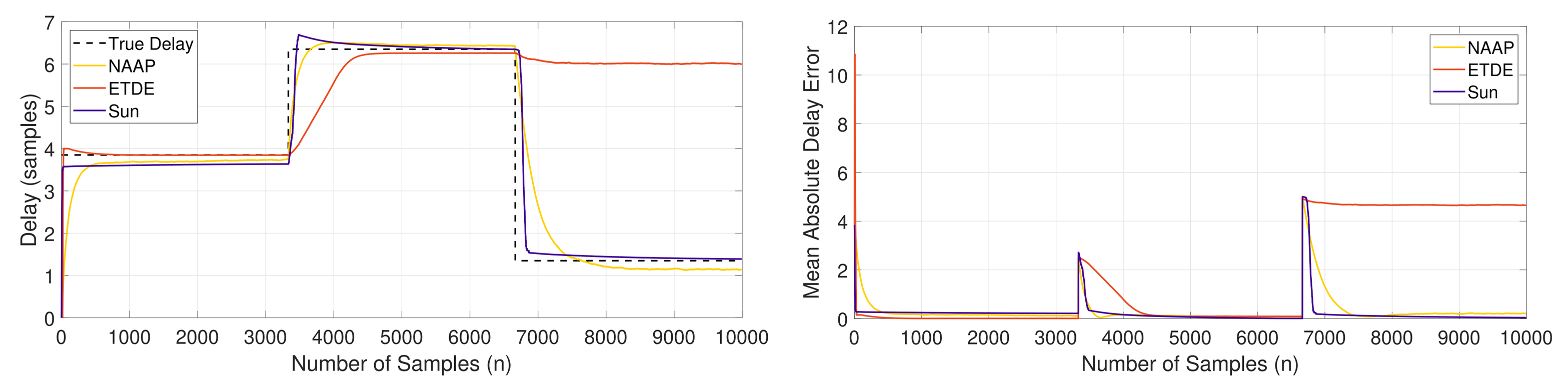
Estimation of a piecewise constant delay signal with two step changes for two scenarios both with an initial delay of $\tau = 3.85$ samples and SNRs of 5dB, 10dB, 20dB and 30dB:

'small' step changes: Changes of $+0.75$ and -1.50 samples

'large' step changes: Changes of $+2.50$ and -5.00 samples

Table 1: Average mean absolute delay errors for different SNR values

	Small Step Change				Large Step Change			
SNR (dB)	5	10	20	30	5	10	20	30
NAAP ($\rho = 0.01$)	0.496	0.313	0.153	0.124	0.528	0.337	0.228	0.219
ETDE ($\mu = 0.02$)	0.112	0.074	0.052	0.047	1.805	1.700	1.661	1.663
Sun ($\mu = 0.008$)	0.249	0.235	0.235	0.234	0.243	0.230	0.233	0.233



(a) Average evolution of the delay (b) Evolution of the mean absolute delay error, in samples

Figure 2: Comparison of the performance of the NAAP ($\rho = 0.01$), the ETDE ($\mu = 0.02$) and Sun algorithm ($\mu = 0.008$) for the large step change scenario with SNR=20dB

Conclusions

- Formulated a LMS style algorithm to estimate the coefficients of the FIR filter
- Time delay estimated using a direct expression based on the filter coefficients
- Algorithm provides a more versatile estimate than alternative methods.