AN IMPROVED VARIABLE TAP-LENGTH ALGORITHM FOR STRUCTURE ADAPTATION

Yu Gong and Colin F. N. Cowan

Institute of Electronics, Communications and Information Technology
Queen's University of Belfast
Northern Ireland Science Park, Queen's Road, Belfast BT3 9DT
United Kingdom, phone: +44 (0) 28 9097 1800, fax: +44 (0) 28 9097 1802
email: y.gong, c.f.n.cowan@ee.qub.ac.uk

ABSTRACT

This paper investigates the variable tap-length algorithm for structure adaptation. Among existing algorithms, the *Segmented Filter* (SF) and *Gradient Descent* (GD) algorithms are of interest as both can track the tap-length variations quickly. In this paper, we first compare the SF and GD algorithms and show that each has advantages/disadvanges relative to the other. Then we propose an improved variable tap-length algorithm which not only has better performance, but also has less complexity, than existing algorithms. The proposed algorithm has great significance in both theory and applications.

1. INTRODUCTION

It is known that the minimum mean square error (MMSE) is a monotonic non-increasing function of the tap-length, but the decrease of the MMSE performance due to the tap-length increase always becomes trivial when the tap-length is long enough. Obviously it is not suitable to have a "too" long filter, as it not only unnecessarily increases the complexity, but also introduces more adaptation noise. It is therefore desirable to search for the *optimum* tap-length that best balances the steady-state performance and complexity. This is how the term *structure adaptation* is derived.

Among existing variable tap-length algorithms, three of them are of interest in this paper as others either aim at improving the convergence behavior (e.g [1, 2, 3]) or can only be implemented in a limit of applications (e.g. [4]). The first is called *Segmented Filter* (SF) algorithm which was described in [5], where the filter is partitioned into several segments and the tap-length is adjusted by one segment being added to, or removed from, the filter according to the difference of the output error levels from the last two segments. In [6], a variable tap-length algorithm based on gradient descent search (GD) was proposed. Expressing the tap-length adaptation in an explicit adaptation rule, the GD algorithm is more flexible to implement than the SF algorithm. Recently, the optimum tap-length has been defined quantitatively in [7], in which a novel variable tap-length algorithm that can converge to the optimum tap-length was also proposed. The proposed algorithm, however, suffers from slow tap-length convergence under some scenarios.

In section 2 of this paper, we will first describe two cost functions that can be used to search for the optimum tap-length, where the first arises directly from the optimum tap-length definition in [7], and the second cost function may give a biased solution to the optimum tap-length but is easier to handle. We will further point out that the algorithm proposed in [7] is based on the first cost function, and both SF and GD algorithms are based on the second cost function. In section 3, we will compare the SF and GD algorithm to show that these two algorithm are equivalent under specific constraints, and each has advantages/disadvantages relative to the other. Finally an improved variable tap-length algorithm will be proposed in section 4. Overcoming the problems of previous algorithms, the new proposed algorithm not only has better performance, but also has less complexity, than existing algorithms. In section 5, numerical simulations will be given to verify the analysis.

This research was sponsored by the UK Engineering and Physical Sciences Research Council under grant number GR/S00217/01.

2. OPTIMUM TAP-LENGTH

In [7], the optimum tap-length is defined as the smallest integer N_o that satisfies:

$$ {}_{N-1} - {}_{N} \leqslant \mathscr{E} \qquad \text{for all } N \geqslant N_o, \tag{1}$$

where ${}_{N}$ is the steady-state MSE when the tap-length is N, $\mathscr{E}$ is a small predetermined value. Note ${}_{N-1} - {}_{N}$ may be negative due to adaptation noise. This definition means that, when the tap-length is larger than N_o, any two steady-state MSE-s corresponding to two succussive tap-lengths can be regarded as same. The sub-optimum tap-length, which is also defined in [7], equals an integer M which is smaller than N_o but satisfies ${}_{M-1} - {}_{M} \leqslant \mathscr{E}$. The *width* of the suboptimum is defined as the number of succussive suboptimum tap-lengths.

From (1), a cost function for searching N_o may be obtained as:

$$\min \{N \mid {}_{N-} - {}_{N} \leqslant \mathscr{E}\}, \tag{2}$$

which is the minimum N that satisfies ${}_{N-} - {}_{N} \leqslant \mathscr{E}$, where is a positive integer. If is larger than the maximum *width* of the suboptimum, the solution of (2) can escape from the suboptima and gives the optimum tap-length N_o.

In [7], a variable tap-length algorithm based on cost function (2) was proposed. Although it can converge to the optimum tap-length in the mean, the proposed algorithm suffers from slow convergence under some scenarios. This is because, on average, the algorithm adjusts the tap-length by comparing ${}_{N(n)}$ and ${}_{N(n-1)}$, which are the steady-state MSE corresponding to the tap-length at time instant n and $n-1$ respectively. At the start (i.e. $n = 0$), however, ${}_{N(0)}$ is not available and must be initialized to an arbitrary value. If this initialization is not appropriate, the tap-length adaptation may be initially driven away from the optimum tap-length due to the transient behavior of the tap-coefficients adaptation.

Below we introduce a *memoryless* cost function for the optimum tap-length searching that can circumvent the initialization problem.

Denoting $\mathbf{w}_N$ and $\mathbf{x}_N$ as the steady state tap-vector and input-vector corresponding to the tap-length N respectively, we define the segmented steady error as:

$$e_M^{(N)} \triangleq d(n) - \mathbf{w}_N^{\mathrm{T}}(1:M) \cdot \mathbf{x}_N(1:M), \quad (3)$$

where $1 \leqslant M \leqslant N$, $\mathbf{w}_N(1:M)$ and $\mathbf{x}_N(1:M)$ are vectors consisting of the first M coefficients of $\mathbf{w}_N$ and $\mathbf{x}_N$ respectively. Then further defining the segmented steady MSE as ${}_M^{(N)} \triangleq \mathrm{E}|e_M^{(N)}|^2$, we construct a modified cost function as:

$$\min \{N \mid {}_{N-}^{(N)} - {}_N^{(N)} \leqslant \mathscr{E}'\}. \quad (4)$$

For clarity of exposition, (2) and (4) are called *cost function 1* and *cost function 2* respectively.

If without the adaptation noise, it can be easily verify that ${}_N^{(N)} = {}_N$ and ${}_{N-}^{(N)} \geqslant {}_{N-}$, which means:

$${}_{N-}^{(N)} - {}_N^{(N)} \geqslant {}_{N-} - {}_N, \quad (5)$$

where "=" holds if and only if ${}_{N-} = {}_N$. It is implied in (5) that, if we let $\mathscr{E}' = \mathscr{E}$, the optimum tap-length from cost function 2 may be overestimated. On another front, if ${}_N$ and ${}_M^{(N)}$ are known, we can have same solutions for cost function 1 and 2 by choosing particular values of $\mathscr{E}'$ and $\mathscr{E}$. In practice, however, ${}_N$ and ${}_M^{(N)}$ are a priori unknown. Thus cost function 2 always gives a bias solution of the optimum tap-length obtained from cost function 1, but it leads to *memoryless* variable tap-length algorithms which require no information about the steady MSE for the previous tap-length $N(n-1)$. In the following of this paper, we only consider the cost function 2.

3. THE SF AND GD ALGORITHMS

In the SF algorithm [5], the filter is divided into L segments, each with coefficients. The tap-length is thus given by $N = L$. An exponential smoothing window with forgetting factor is used to track the *accumulated squared error* (ASE) of the last two segments:

$$\mathrm{ASE_k}(n) \triangleq \sum_{i=1}^{n} {}^{n-i} \left(e_k^{(N)}(i)\right)^2, \quad (6)$$

where $k = L$ or $L-1$, $e_k^{(N)}(i)$ is defined in (3) which is the output error from the kth segment at time instant i. Suppose at time instant n, there are $L(n)$ segments. At instant $n+1$, if $\mathrm{ASE_L}(n) \leqslant {}_{\mathrm{up}}\mathrm{ASE_{L-1}}(n)$, then $L(n+1) = L(n)+1$; else if $\mathrm{ASE_L}(n) \geqslant {}_{\mathrm{dw}}\mathrm{ASE_{L-1}}(n)$, then $L(n+1) = L(n)-1$. It is clear that, in the average meaning, the SF algorithm adjusts the tap-length by comparing the segmented steady MSE of ${}_L^{(N)}$ and ${}_{L-}^{(N)}$, which is obviously based on the cost function 2.

If we let ${}_{\mathrm{up}} = {}_{\mathrm{dw}} = 1$, then the tap-length adaptation rule for the SF algorithm can be expressed as:

$$N(n+1) = N(n) - \cdot \mathrm{sign}(\mathrm{ASE_L}(n) - \mathrm{ASE_{L-1}}(n)), \quad (7)$$

where $N(n) = L(n)$. Substituting (6) into $\mathrm{ASE_L}(n)$ gives:

$$\begin{aligned} &\mathrm{ASE_L}(n) - \mathrm{ASE_{L-1}}(n) \\ &= -\sum_{i=1}^{n} {}^{n-i} \left(e_{N(n)}^{(N(n))}(i) + e_{N(n)-}^{(N(n))}(i)\right) \cdot \mathbf{x}'^{\mathrm{T}}(i)\mathbf{w}'(i) \\ &\approx -\sum_{i=1}^{n} 2 \cdot {}^{n-i} e_{N(n)-\lfloor /2 \rfloor}^{(N(n))}(i) \cdot \mathbf{x}'^{\mathrm{T}}(i)\mathbf{w}'(i) \end{aligned} \quad (8)$$

where we use the approximation that $e_{N(n)}^{(N(n))}(i) + e_{N(n)-}^{(N(n))}(i) \approx 2 \cdot e_{N(n)-\lfloor /2 \rfloor}^{(N(n))}(i)$ which equals twice of the output error from the middle tap of the last segment, and $\mathbf{x}'(i)$ and $\mathbf{w}'(i)$ consist of coefficients of the last segment of the input vector $\mathbf{x}_{N(n)}(i)$ and tap-vector $\mathbf{w}_{N(n)}(i)$ respectively. Note that $\lfloor . \rfloor$ rounds the embraced value to the nearest integer. Substituting (8) into (7) gives the tap-length adaptation rule for the SF algorithm:

$$N(n+1) = N(n) + \cdot \mathrm{sign}\left(\sum_{i=1}^{n} {}^{n-i} \cdot e_{N(n)-\lfloor /2 \rfloor}^{(N(n))}(i) \cdot \mathbf{x}'^{\mathrm{T}}(i)\mathbf{w}'(i)\right). \quad (9)$$

With slight re-arrangements, the adaptation rule of the GD algorithm (see [6]) can be expressed as:

$$N(n+1) = N(n) + \cdot \mathrm{sign}\left(\frac{1}{T}\sum_{i=n-T+1}^{n} e_{N(n)-\lfloor /2 \rfloor}^{(N(n))}(i) \cdot \mathbf{x}'^{\mathrm{T}}(i)\mathbf{w}'(i)\right), \quad (10)$$

where is the step-size parameter for tap-length adaptation, T is the size of the rectangular window used to obtained the smoothed gradient, and both and T are positive integers. Note that (10) only applies at every T intervals.

It is clear that the only difference between (9) and (10) is that the two approaches use different smoothing method, which however can be replaced by each other, to estimate the gradient. Thus if we let ${}_{\mathrm{up}} = {}_{\mathrm{dw}} = 1$ for the SF algorithm and let $=$ for the GD algorithm, the two algorithms become identical.

The main advantage of the GD algorithm over the SF algorithm is that the former can freely choose , while the latter must have $\equiv$ which implies the tap-length in the SF algorithm has to be changed by each time. With this additional degree of freedom, the GD algorithm is more flexible in handling the local minima, and can more *smoothly* adjust the tap-length, than its SF counterpart.

However, compared with the SF algorithm, the GD algorithm imposes a new constraint by implying ${}_{\mathrm{up}} = {}_{\mathrm{dw}} = 1$. With this constraint, the cost function on which the GD algorithm is based becomes:

$$\min \{N \mid {}_{N-}^{(N)} - {}_N^{(N)} \leqslant 0\}. \quad (11)$$

It is clear by comparing (11) with (2) and (4) that the GD algorithm may overestimate the optimum tap-length. Moreover, we observe that the difference between ${}_{N-}^{(N)}$ and ${}_N^{(N)}$ usually becomes very small when $N \gg N_o$. With inaccurate estimate of the steady MSE, this may cause the tap-length "wandering" about in the range which is larger than N_o. Thus if the initial tap-length $N(0)$ is much larger than N_o, or if the optimum tap-length decreases due to channel variation, it may take a long time for the GD algorithm to converge.

This problem is more serious when the adaptation noise is low which may caused by choosing a small step-size for the LMS algorithm, since then $^{(N)}_{N-}$ and $^{(N)}_{N}$ are almost identical for $N \gg N_o$. On the contrary, the SF algorithm can overcome this problem by choosing the parameters of $_{\text{up}}$ and $_{\text{dw}}$ appropriately.

4. FRACTIONAL TAP-LENGTH ALGORITHM

The “wandering” problem of the GD algorithm is similar to that with the LMS algorithm when it is implemented using fixed point parameters and data. A classical solution is to implement the leaky LMS algorithm where a leaky factor is introduced in the adaptation rule. In this section, we show that, if we relax the constraint that the tap-length must be integer and introduce a concept of pseudo fractional tap-length, we can use a similar approach as that for the leaky LMS algorithm to overcome the “wandering” problem of the tap-length adaption. The concept of the fractional tap-length was first proposed in [7] but based on the cost function 1.

To be specific, we define n_f as the pseudo fractional tap-length which can take real values, and construct the following adaption rule:

$$n_f(n+1) = (n_f(n) - \quad) - \quad \left[\left(e^{(N(n))}_{N(n)}(n)\right)^2 - \left(e^{(N(n))}_{N(n)-}\ (n)\right)^2\right], \tag{12}$$

where   is the leaky factor which satisfies $\ll$, and both   and   are small positive numbers. Initially we have $n_f(0) = N_f(0)$. The true tap-length $N_f(n)$ is adjusted according to:

$$N(n+1) = \begin{cases} \lfloor n_f(n) \rfloor, & |N(n) - n_f(n)| \geqslant \\ N(n), & \text{otherwise} \end{cases} \tag{13}$$

where $\lfloor . \rfloor$ rounds the embraced value to the nearest integer. To make sense of (12), we should ensure $n_f(n) \geqslant \quad +1$. Note that unlike (9) and (10), it is not necessary to have a “sign” operator in (12).

In the FT algorithm, we can freely choose the step-size parameter   and the parameters of   and   which have similar effects as $_{\text{up}}$ and $_{\text{dw}}$ in the SF algorithm. Thus the FT algorithm retains all the advantages of the SF and GD algorithms. Moreover, not like the algorithms proposed in [5, 6, 7], the FT algorithm uses instantaneous errors rather than the averaged errors for the tap-length adaptation. Thus it has significantly less complexity than the existing algorithms.

5. NUMERICAL SIMULATIONS

In this section, as an example, we will compare the FT algorithm with the SF and GD algorithms in the application of adaptive system modelling.

5.1 Simulation setup

For comparison, we use the same system as that in [7]. To be specific, two unknown systems are tested, each with transfer functions of $W_o = W_1$ and $W_o = W_2$ respectively, where:

$$W_1(z) = \frac{1+0.2z^{-8}}{1-0.7z^{-1}}, \qquad W_2(z) = \frac{1}{1-0.3z^{-1}}. \tag{14}$$

The impulse response of $W_o(z)$ has an infinite length, and thus any special filter length has not been privileged. The input signal is a white Gaussian noise passing through a spectrum shaping filter with transfer function of $H(z) = 0.35 + z^{-1} + 0.35z^{-2}$. The Gaussian noise added to the unknown system provides a signal-noise-ratio (SNR) of 20dB.

In all simulations, normalized LMS (NLMS) algorithm is used for tap coefficients adaptation, because as was pointed out in [7], the NLMS algorithm is more robust to implement than the LMS algorithm when a variable tap-length algorithm is used. The step-size of the NLMS algorithm is set as 0.2 for all experiments unless otherwise specified.

For fair comparison, the rectangular window with size of $T = 10$ is used to obtain both the ASE in the SF algorithm and the smoothed gradient in the GD algorithm. The parameters used in the tested variable tap-length algorithms are shown in Table 1.

			T	up	dw		
SF	4	–	10	0.8	1	–	–
GD	4	1	10	–	–	–	–
FT	4	1	–	–	–	0.03	1

Table 1: Parameters for the variable tap-length algorithms.

5.2 Simulation results and discussions

Fig. 1 shows the curves of the steady-state MSE $_N$ with respect to the tap-length N. It is clearly shown that, when $W_o(z) = W_1(z)$, the optimum tap-length is around 15, the sub-optimum tap-lengths are $\{6, 7, 8\}$; but when $W_o(z) = W_2(z)$, the optimum tap-length is around 4 and no sub-optimum tap-lengths exist.

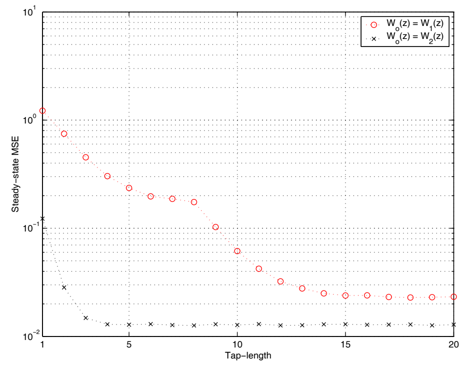
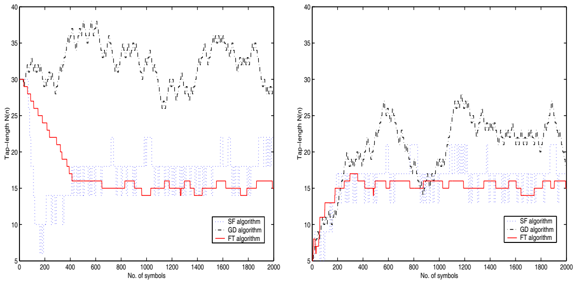


Figure 1: The curves of the steady-state MSE with respect to the tap-length.

Fig. 2 shows the tap-length learning curves with different initializations based on one typical simulation *run* for the SF, GD and FT algorithms respectively. In this experiment, to magnify the ”wandering” effect of the GD algorithm, we deliberately choose a small step-size of 0.05 for the NLMS algorithm to give a small adaptation noise. As expected, the GD algorithm overestimates the optimum tap-length, and tap-length is “wandering” around in the high value areas especially when the initial tap-length $N(0) = 30$. On the contrary, both SF and FT algorithms converge fast to around the optimum tap-length for either $N(0) = 30$ or $N(0) = 5$, but the FT algorithm clearly has a much smoother learning curve than the SF algorithm because the former can freely choose the step-size parameter of .



(a) $N(0) = 30$. (b) $N(0) = 5$.

Figure 2: The learning curves of $N(n)$ based on one typical simulation run when $W_o(z) = W_1(z)$, where the step-size for the NLMS algorithm is deliberately set as small as 0.05.

Fig. 3 shows the averaged tap-length learning curve over 10 independent runs in a time varying scenario, where $W_o(z) = W_1(z)$ when $n < 2000$ or $n \geqslant 4000$, and $W_o(z) = W_2(z)$ when $2000 \leqslant n < 4000$. It is clearly shown that the SF and FT algorithms have similar transient behaviors, and as expected, both converge faster than the GD algorithm especially when $W_o(z)$ is changed from $W_1(z)$ to $W_2(z)$ at time instant 2000. On another front, the GD algorithm has smoother learning curve than the SF algorithm, though the FT algorithm has the smoothest learning curve among all. Moreover, it can also be observed that both SF and GD algorithm overestimate the optimum tap-length. Note that unlike the GD algorithm, the SF can overcome the overestimate problem by adjusting the values of parameters $_{up}$ and $_{dw}$ appropriately.

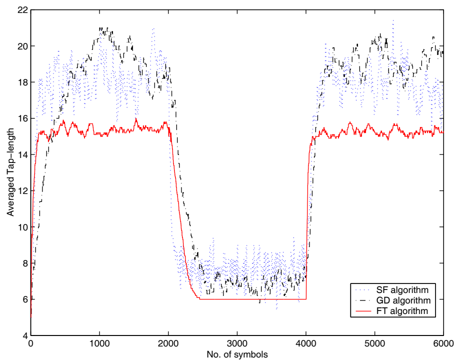


Figure 3: The averaged tap-length learning curves over 10 independent simulation runs, where $W_o(z) = W_1(z)$ when $n < 2000$ or $n \geqslant 4000$, and $W_o(z) = W_2(z)$ when $2000 \leqslant n < 4000$.

Fig. 4 shows the MSE learning curves corresponding to Fig. 3. For better exposition, the MSE learning curve, which is obtained over 10 independent runs, is further averaged by a rectangular window with size of 50. It is clearly shown that the SF and FT algorithms have similar transient behaviors in MSE leaning curves, though the latter seems to be slightly better than the former, while both are better than the GD algorithm. This corresponds to their transient behaviors of the tap-length adaptation respectively. On another front, all 3 algorithms have similar steady-state MSE performance because all algorithm can converge to the optimum tap-length.

In general, the FT algorithm has much better transient and steady-state behaviors for the length adaptation than its SF and GD counterparts.

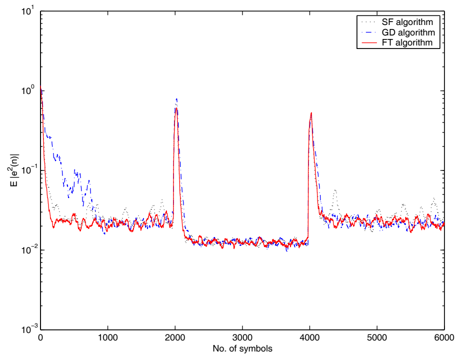


Figure 4: The MSE learning curves corresponding to Fig. 3.

6. CONCLUSION

This paper compared the SF and GD algorithms to show that each has advantages/dis-advantages relative to the other, and proposed an improved variable tap-length algorithm using the concept of the *fractional* tap-length. The proposed FT algorithm not only has much better performance but also has less complexity than existing algorithms. This provides a significant advance towards the practical implementation of flexible, structurally adaptive filters for real-time application in a number of scenarios.

REFERENCES

[1] Z. Pritzker and A. Feuer, "Variable length stochastic gradient algorithm," *IEEE Trans. on Signal Processing*, vol. 39, no. 4, pp. 997 – 1001, April 1991.

[2] Y. K. Won, R. H. Park, J. H. Park, and B. U. Lee, "Variable LMS algorithm using the time constant concept," *IEEE Trans. on Consumer Electronics*, vol. 40, no. 3, pp. 655 – 661, Aug. 1994.

[3] C. Rusu and C. F. N. Cowan, "Novel stochastic gradient adaptive algorithm with variable length," in *European Conference on Circuit Theory and Design (ECCTD'01)*, Espoo, Finland, Aug. 2001.

[4] Y. Gu, K. Tang, H. Cui, and W. Du, "Convergence analysis of a deficient-length LMS filter and optimal-length to model exponential decay impulse response," *IEEE Signal Processing letters*, vol. 10, no. 1, pp. 4 – 7, Jan. 2003.

[5] F. Riera-Palou, J. M. Noras, and D. G. M. Cruickshank, "Linear equalisers with dynamic and automatic length selection," *Electronic Letters*, vol. 37, no. 25, pp. 1553 – 1554, Dec. 2001.

[6] Y. Gu, K. Tang, H. Cui, and W. Du, "LMS algorithm with gradient descent filter length," *IEEE Signal Processing letters*, vol. 11, no. 3, pp. 305 – 307, March 2004.

[7] Y. Gong and C. F. N. Cowan, "Structure adaptation of linear MMSE adaptive filters," *IEE Proceedings - Vision, Image and Signal Processing*, vol. 151, no. 4, pp. 271 – 277, Aug. 2004.