GENERAL PARAMETER-BASED ADAPTIVE EXTENSION TO FIR FILTERS

Olli Vainio

Tampere University of Technology
Department of Information Technology
P. O. Box 553, FIN-33101 Tampere, Finland

Seppo J. Ovaska

Helsinki University of Technology
Institute of Intelligent Power Electronics
P. O. Box 3000, FIN-02015 HUT, Finland

ABSTRACT

A class of computationally efficient adaptive algorithms for transversal filters is discussed. The algorithms, which are based on the so-called general parameter method, use typically one or a few dynamically adjusted parameters, each to be added to a block of coefficients of a fixed basis FIR filter. Thus the overall filter is adapted so that the output error is minimized. The adaptive extension can be constructed as an 'add-on' element to be used in parallel with fixed-coefficient filters. An efficient implementation structure is proposed, and the stability and convergence properties of the multiple-parameter algorithm are analyzed.

1. INTRODUCTION

Adaptive filters are widely used in signal processing when the filtering task cannot be completely specified in advance. For example, the characteristics of the signal and noise may vary, or the system parameters may change in time. If the possible variations are large, a fully adaptive filter is needed, in order to achieve satisfactory system performance. Such a filter often has a high computational complexity, especially in adaptive transversal filters where a large number of coefficients may be needed due to the length of the impulse response of the system.

Among the most popular adaptation methods are the LMS and RLS algorithms [1],[2]. The RLS algorithm has excellent convergence characteristics, while the LMS algorithm is relatively simple to implement. The coefficient update equation in LMS is given by

$$\mathbf{w}(n+1) = \mathbf{w}(n) + 2\mu e(n)\mathbf{x}(n), \tag{1}$$

where $\mathbf{w}(n)$ is the coefficient vector, $\mathbf{x}(n)$ is the data vector within the filter window, and $e(n) = r(n) - y(n)$ is the difference between the desired value $r(n)$ and the filter output $y(n)$. When the processes $x(n)$ and $r(n)$ are jointly stationary, this algorithm converges to coefficients which, in average, follow the Wiener-Hopf solution. Coefficient updates in the LMS algorithm require $N+1$ multiplications for an N-tap transversal filter.

Simplified versions of the basic LMS algorithm have been developed by replacing in (1) either $e(n)$ or $\mathbf{x}(n)$ or both by their signs [2]. If the gain parameter μ is then chosen as a power of two, the coefficients can be updated without multiplications. However, even those simplified algorithms require an arithmetic operation to be done on all of the N coefficients. Furthermore, it has been proposed to filter the gradient estimate, to arrive at a higher order algorithm for faster convergence [3].

In this paper, we discuss an adaptive filtering approach, where the filter coefficients consist of a fixed part and an adjustable part. Typically the fixed basis filter is designed assuming some nominal signal characteristics, such as the frequency of a sinusoidal signal [4]. The adjustable part is adapted to tune the overall filter so that the output error is minimized. The proposed adaptation approach has a low computational complexity, as the number of arithmetic operations depends only on the number of adaptive parameters but not on the overall filter length.

2. GENERAL PARAMETER-BASED ADAPTIVE FILTERS

Ashimov and Syzdykov [5] have proposed a class of algorithms, called the *general parameter method,* for system identification. Digital signal processing applications of the method have been considered recently in [6],[7], where a single general parameter was used, implementing the filtering algorithm

$$y(n) = \sum_{k=0}^{N-1} [g(n) + h(k)]x(n-k), \tag{2}$$

where the $h(k)$'s are the coefficients of a fixed basis filter.

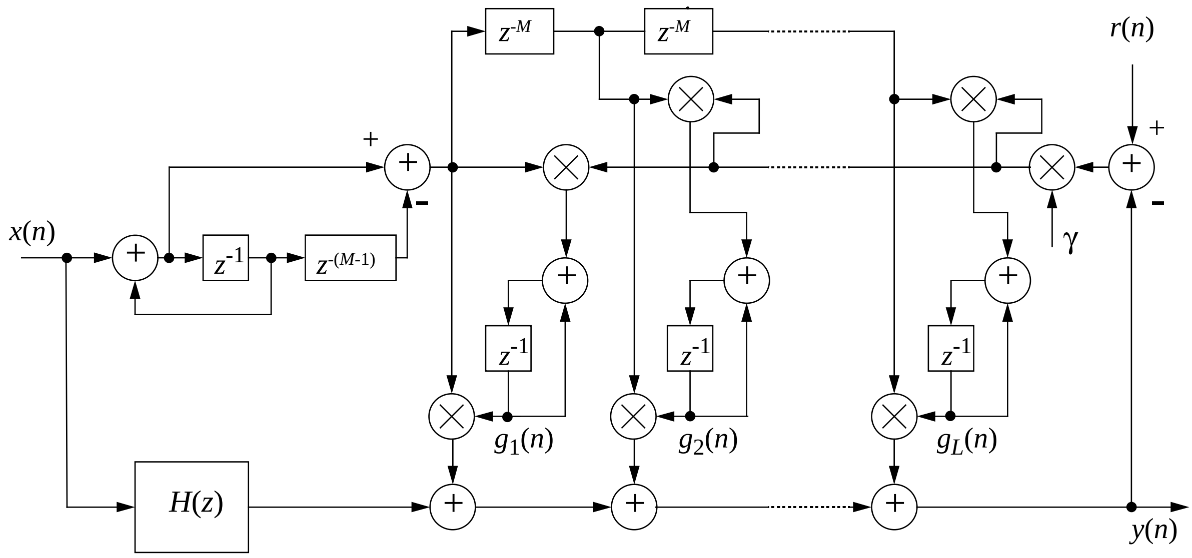

Fig. 1. Implementation of the adaptive extension with multiple general parameters.

The general parameter $g(n)$ is updated as

$$g(n+1) = g(n) + \gamma[r(n) - y(n)] \sum_{k=0}^{N-1} x(n-k), \quad (3)$$

where $r(n)$ is the reference input against which the output is compared, and γ is a gain factor. The use of several general parameters was also discussed in [5].

In this work, we consider the case of L general parameters $g_1(n)$, $g_2(n)$, $\cdots$, $g_L(n)$, each to be related to a group of filter coefficients. Assuming that the overall filter length, N, is divisible by L, and $M = N/L$, the filtering algorithm is of the form:

$$y(n) = \sum_{j=0}^{L-1} \sum_{k=0}^{M-1} [h(k+Mj) + g_{j+1}(n)] x(n-k-Mj), \quad (4)$$

where the parameters $g_{j+1}(n)$, $j = 0, 1, \cdots, L-1$, are updated according to

$$g_{j+1}(n+1) = g_{j+1}(n) + \gamma[r(n) - y(n)] \sum_{k=0}^{M-1} x(n-k-Mj). \quad (5)$$

The standard LMS algorithm is seen to be a special case of this approach, where $L = N$, and all the $h(k) = 0$ for $k = 0, 1, \cdots, N-1$.

In order to arrive at an efficient implementation, we can write (4) in the form:

$$y(n) = \sum_{k=0}^{N-1} h(k) x(n-k) + \sum_{j=0}^{L-1} g_{j+1}(n) \sum_{k=0}^{M-1} x(n-k-Mj). \quad (6)$$

The L running sums of the data samples can be efficiently computed using the recursive running sum structure [8], where the arithmetic complexity does not depend on the number of samples inside of the running window. We further notice that the subsequent running sums are simply delayed versions of the first sum. Therefore, this computation only needs to be implemented once, which is a reason for using equally sized coefficient blocks. The resulting implementation structure of the filter with multiple adaptive parameters is shown in Fig. 1. Altogether, implementation of the adaptive extension requires $2L + 1$ multiplications.

3. STABILITY

The algorithm can be analyzed using a similar approach as for the LMS algorithm [7],[9]. Let us assume that the reference data sequence is generated by the linear time varying model

$$r(n) = \mathbf{x}(n)\mathbf{\Theta}_0(n) + v(n), \quad (7)$$

where $\mathbf{x}(n) = [x(n) \; x(n-1) \; \cdots \; x(n-N+1)]$. The true parameter $\mathbf{\Theta}_0(n)$ has a model of the form

$$\mathbf{\Theta}_0(n+1) = \mathbf{\Theta}_0(n) + \xi(n). \quad (8)$$

The variables $v(n)$ and $\xi(n)$ are considered as noise or disturbances. The adaptation error is

$$\hat{\mathbf{\Theta}}(n) = \mathbf{\Theta}_0(n) - \mathbf{\Theta}(n), \quad (9)$$

where

$$\mathbf{\Theta}(n) = \mathbf{h} + \mathbf{g}(n) \quad (10)$$

is the composite coefficient vector with

$$\mathbf{h} = [h(0)\ h(1)\ \cdots\ h(N-1)]^{\mathrm{T}}$$

and

$$\mathbf{g}(n) = [\overbrace{g_1(n)\ \cdots\ g_1(n)}^{\text{multiplicity } M}\ \cdots\ \overbrace{g_L(n)\ \cdots\ g_L(n)}^{\text{multiplicity } M}]^{\mathrm{T}}.$$

Therefore,

$$\hat{\mathbf{\Theta}}(n+1) = \hat{\mathbf{\Theta}}(n)+\xi(n)-\mathbf{h}-\mathbf{g}(n)-\gamma[r(n)-\mathbf{x}(n)\mathbf{\Theta}(n)]\mathbf{S}, \quad (11)$$

where

$$\mathbf{S} = [\overbrace{\sum_{j=0}^{M-1} x(n-j)\ \cdots\ \sum_{j=0}^{M-1} x(n-j)}^{\text{multiplicity } M}\ \cdots$$

$$\overbrace{\sum_{j=(L-1)M}^{N-1} x(n-j)\ \cdots\ \sum_{j=(L-1)M}^{N-1} x(n-j)}^{\text{multiplicity } M}]^{\mathrm{T}}.$$

Substituting (7) into (11) we obtain

$$\hat{\mathbf{\Theta}}(n+1) = \hat{\mathbf{\Theta}}(n) + \xi(n) - \gamma[\mathbf{x}(n)\hat{\mathbf{\Theta}}(n) + v(n)]\mathbf{S}. \quad (12)$$

This can be written in the form

$$\hat{\mathbf{\Theta}}(n+1) = [\mathbf{I} - \gamma\mathbf{S}\mathbf{x}(n)]\hat{\mathbf{\Theta}}(n) + \xi(n) - \gamma\mathbf{S}v(n), \quad (13)$$

where $\mathbf{I}$ is an $N \times N$ unit matrix. The stability of the algorithm therefore depends on the eigenvalues of the matrix $\mathbf{I} - \gamma\mathbf{S}\mathbf{x}(n)$ which are equal to one except that given by

$$\lambda = 1 - \gamma\mathbf{x}(n)\mathbf{S}. \quad (14)$$

For stability it is required that the gain factor γ is limited by

$$0 < \gamma < \frac{2}{\mathrm{E}[\mathbf{x}(n)\mathbf{S}]}, \quad (15)$$

where E[.] denotes the expectation value.

4. CONVERGENCE PROPERTIES

Like any adaptive algorithm, the general parameter-based approach is capable of adaptating to zero output error only if there are equally many or more adaptive parameters than there are degrees of freedom in the underlying signal model. Another characteristic feature of the algorithm is the fact that summation of samples in (5) corresponds to averaging, which causes different gain for different frequencies. This is reflected as different adaptation speed for different sinusoidal inputs when the algorithm gain γ is kept constant. These properties are illustrated by the following example.

In this experiment, we set $r(n) = x(n)$ and the input signal consists of multiple sinusoids. The filter $H(z)$ is a notch type FIR filter of length 12 with $\omega_0 = 0.1\pi$ as the blocking notch frequency. The amplitude response of $H(z)$ is shown in Fig. 2. There are four adaptive parameters, $g_0(n)$ to $g_3(n)$, initialized to zero, and $M = 3$. The summation of the three samples within each coefficient group therefore corresponds to filtering with the frequency response shown with the dashed line in Fig. 2. The output error and the values of the four parameters are shown in Fig. 3 and Fig. 4, respectively, when the input signal varies as follows. From 0 to 300 samples, the input is a single sinusoid of the angular frequency 0.1π. As it is initially blocked by the fixed filter, the adaptive extension adapts to pass this signal with unity gain and zero phase shift. At sample number 300, an additional sinusoid of $\omega = 0.2\pi$ is switched on and the system again adapts to zero output error. At sample 600, the input becomes a single sinusoid with $\omega = 0.55\pi$. At sample 900, the input is switched to three separate sinusoids ($\omega = 0.1\pi, 0.2\pi$, and 0.3π). Full adaptation is no longer possible with this number of parameters, but the general parameters follow a periodical pattern.

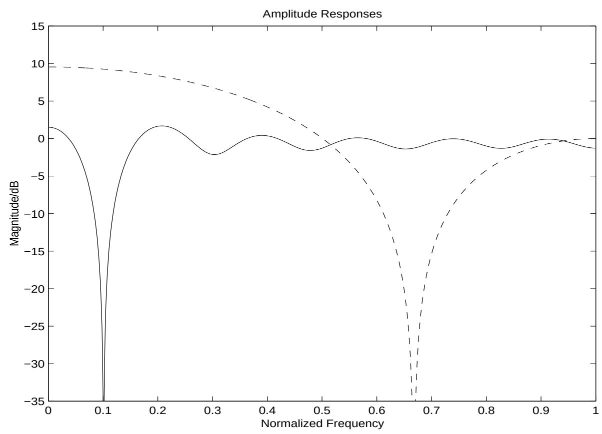

Fig. 2. Amplitude response of $H(z)$ (solid) and the running sum of length three (dashed).

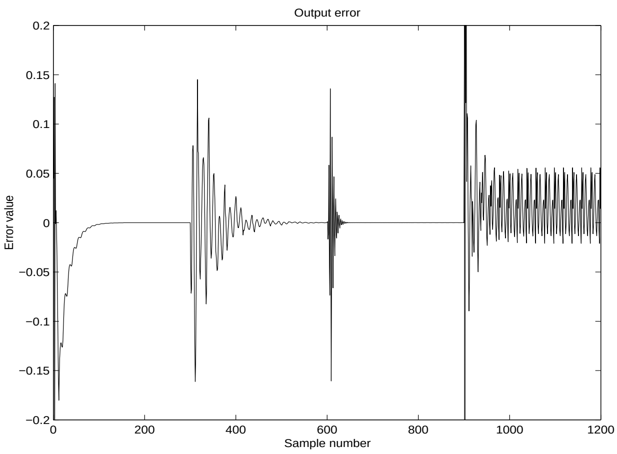

Fig. 3. Behavior of the output error for multiple sinusoid input.

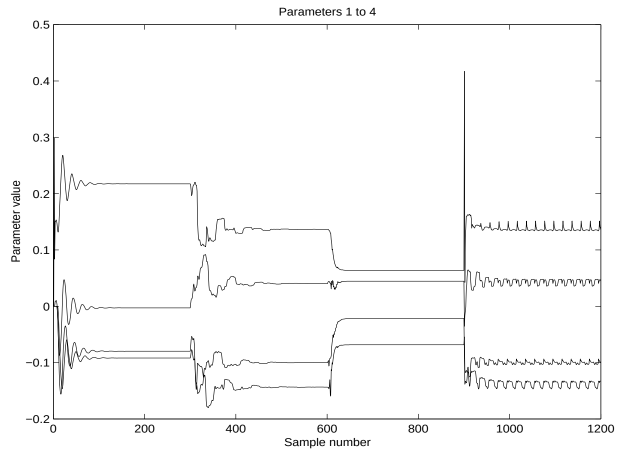

Fig. 4. Trace of the four adaptive parameters.

As with LMS, a *normalized* algorithm can be introduced, adjusting the gain factor according to

$$\gamma(n) = \alpha\gamma(n-1) + (1-\alpha)\frac{\beta}{1+\beta\mathbf{x}(n)\mathbf{S}}, \qquad (16)$$

where α and β are fixed. The output error shown in Fig. 5 results from repeating the previous example with the normalized algorithm using $\alpha = 0.95$ and $\beta = 0.1$. The settling times are seen to be more equal after changes in the input signal.

5. CONCLUSIONS

In a typical application, the general parameter algorithm allows computationally simple adaptation to be used in parallel with a fixed basis filter. The algorithm with several adaptive parameters can be considered in certain cases as a simplified version of the conventional LMS adaptive filter algorithm.

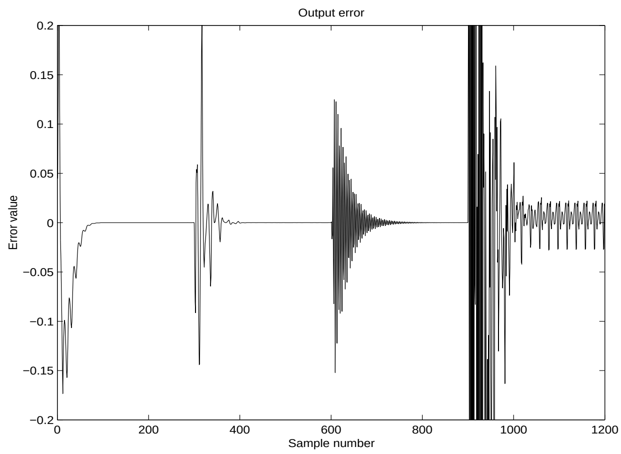

Fig. 5. Output error from the normalized algorithm.

6. REFERENCES

[1] B. Farhang-Boroujeny, *Adaptive Filters Theory and Applications.* Chichester, England: John Wiley & Sons, 1998.

[2] O. Macchi, *Adaptive Processing The Least Mean Squares Approach with Applications in Transmission.* Chichester, England: John Wiley & Sons, 1995.

[3] J. R. Glover, "High order algorithms for adaptive filters," *IEEE Trans. Commun.,* vol. COM-27, pp. 216-221, 1979.

[4] O. Vainio and S. J. Ovaska, "Noise reduction in zero crossing detection by predictive digital filtering," *IEEE Trans. Ind. Electron.,* vol. 42, pp. 58-62, Feb. 1995.

[5] A. A. Ashimov and D. J. Syzdykov, "Identification of high dimensional system by the general parameter method," in *Preprints 8th Triennial World Congress of Int. Federation of Automatic Control,* Kyoto, Japan, Aug. 1981, pp. 32-37.

[6] J. M. A. Tanskanen, S. J. Ovaska, and O. Vainio, "Adaptive general parameter extension to FIR predictors," in *Proc. 10th European Signal Processing Conference,* Tampere, Finland, Sept. 2000, pp. 1005-1008.

[7] J. M. A. Tanskanen, O. Vainio, and S. J. Ovaska, "Adaptive general parameter extension for tuning FIR predictors," in *Proc. IFAC Workshop on Linear Time Delay Systems,* Ancona, Italy, Sept. 2000, pp. 42-47.

[8] T. Raita-aho, T. Saramäki, and O. Vainio, "A digital filter chip for ECG signal processing," *IEEE Trans. Instrum. Meas.,* vol. 43, no. 4, pp. 644-649, Aug. 1994.

[9] N. Kalouptsidis, *Signal Processing Systems Theory and Design.* New York, NY: John Wiley & Sons, 1997, pp. 601-604.