# Adaptive Graph Filter Based on LMS Algorithm

Chien-Cheng Tseng[1], *Senior Member, IEEE* and Su-Ling Lee[2]
[1]National Kaohsiung University of Science and Technology, Kaohsiung, Taiwan
[2]Chang Jung Christian University, Tainan, Taiwan

***Abstract*-In this paper, an adaptive graph filter based on least mean squared (LMS) algorithm is presented. First, the basics of graph filter are briefly reviewed. Then, conventional LMS algorithm is extended to the graph LMS algorithm to update the graph filter coefficients. Finally, numerical examples of adaptive graph filters for identifying and tracking unknown graph system is demonstrated to show the effectiveness of the proposed algorithm.**

## I. INTRODUCTION

Classical adaptive filtering has received a great deal of attenuations in the area of communication, radar, control and biomedical electronics [1]. The growth is mainly promoted by the factor that the real-world signal processing environment is usually time-varying, so the use of adaptive filter provides performance improvement over the use of a fixed filter. So far, various adaptive algorithms have been developed to adjust the digital filter parameters by minimizing the mean squared error. The main reason of choosing least mean squared (LMS) error criterion is its mathematical simplicity. Due to the success of classical adaptive filters, it is an interesting research topic to develop adaptive graph filters for identifying and tracking unknown graph systems used in the processing of graph signals. Although an adaptive graph filtering approach has been developed to semi-supervised classification in [2], it is not a direct generalization of classical adaptive filter in textbook [1]. In this paper, the conventional LMS algorithm is extended to the graph LMS algorithm to update the adaptive graph filter coefficients. Based on the general block diagram of the adaptive filter in the textbook [1], the proposed adaptive graph filter is shown in Fig.1. The purpose of this paper is to develop an adaptive graph filter based on LMS algorithm.

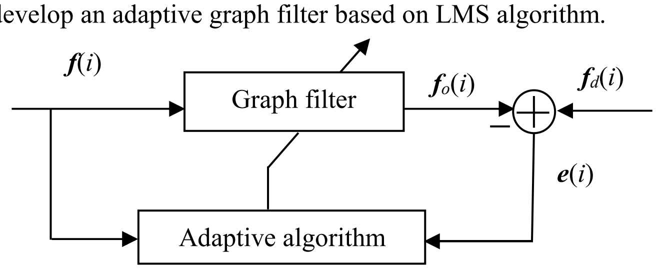


Fig.1 The block diagram of adaptive graph filter.

## II. ADAPTIVE GRAPH FILTER

Let $G = (V, \boldsymbol{A})$ be a finite undirected graph, where $V = \{v_1, v_2, \cdots, v_N\}$ is the vertex set and $\boldsymbol{A}$ is the adjacent matrix. The elements $a_{n,m}$ of adjacent matrix $\boldsymbol{A}$ give the weight along the edge from vertices $v_n$ to $v_m$. Let $d_n$ be the sum of all edge weights connected to vertex $v_n$, then $\boldsymbol{D} = diag\,[d_1, d_2, \cdots, d_N]$ is the diagonal degree matrix and the graph Laplacian matrix is defined as $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$ [3]. In Fig.1, the $\boldsymbol{f}(i)$ is the input graph signal defined by $\boldsymbol{f}(i) = [f_i(1), f_i(2), \cdots, f_i(N)]^T$ and $\boldsymbol{f}_d(i)$ is the desired graph signal. The notation $i$ is the time index of graph signals. The output signal of graph filter is given by

$$\boldsymbol{f}_o(i) = \left(\sum_{k=0}^{K} b_k(i)\boldsymbol{L}^k\right)\boldsymbol{f}(i) \tag{1}$$

where $K$ is filter order and the notation $b_k(i)$ denotes the filter coefficient $b_k$ at time index $i$ because the graph filter is time-varying. Defining the signals as

$$\boldsymbol{f}_k(i) = \boldsymbol{L}^k \boldsymbol{f}(i) \quad k = 0,1,\cdots,K \tag{2}$$

then (1) can be written as the following form

$$\boldsymbol{f}_o(i) = \boldsymbol{\Psi}(i)\boldsymbol{b}(i) \tag{3}$$

where matrix $\boldsymbol{\Psi}(i)$ and vector $\boldsymbol{b}(i)$ are defined by

$$\boldsymbol{\Psi}(i) = [\boldsymbol{f}_0(i) \quad \boldsymbol{f}_1(i) \quad \cdots \quad \boldsymbol{f}_K(i)] \tag{4a}$$

$$\boldsymbol{b}(i) = [b_0(i) \quad b_1(i) \quad \cdots \quad b_K(i)]^T \tag{4b}$$

The error graph signal between desired signal $\boldsymbol{f}_d(i)$ and output signal $\boldsymbol{f}_o(i)$ is then given by

$$\begin{aligned}\boldsymbol{e}(i) &= \boldsymbol{f}_d(i) - \boldsymbol{f}_o(i) \\ &= \boldsymbol{f}_d(i) - \boldsymbol{\Psi}(i)\boldsymbol{b}(i)\end{aligned} \tag{5}$$

Thus, the instantaneous energy of error signal at time index $i$ can be computed as

$$\begin{aligned}J(i) &= \boldsymbol{e}(i)^T\boldsymbol{e}(i) \\ &= \left(\boldsymbol{f}_d(i) - \boldsymbol{\Psi}(i)\boldsymbol{b}(i)\right)^T\left(\boldsymbol{f}_d(i) - \boldsymbol{\Psi}(i)\boldsymbol{b}(i)\right) \\ &= \boldsymbol{b}(i)^T\boldsymbol{\Psi}(i)^T\boldsymbol{\Psi}(i)\boldsymbol{b}(i) - 2\boldsymbol{b}(i)^T\boldsymbol{\Psi}(i)^T\boldsymbol{f}_d(i) + \boldsymbol{f}_d(i)^T\boldsymbol{f}_d(i)\end{aligned} \tag{6}$$

The gradient of $J(i)$ with respect to $\boldsymbol{b}(i)$ is then given by

$$\nabla J(i) = 2\boldsymbol{\Psi}(i)^T\boldsymbol{\Psi}(i)\boldsymbol{b}(i) - 2\boldsymbol{\Psi}(i)^T\boldsymbol{f}_d(i) \tag{7}$$

Substituting (3) and (5) into (7), we have

$$\begin{aligned}\nabla J(i) &= 2\boldsymbol{\Psi}(i)^T\boldsymbol{f}_o(i) - 2\boldsymbol{\Psi}(i)^T\boldsymbol{f}_d(i) \\ &= -2\boldsymbol{\Psi}(i)^T(\boldsymbol{f}_d(i) - \boldsymbol{f}_o(i)) \\ &= -2\boldsymbol{\Psi}(i)^T\boldsymbol{e}(i)\end{aligned} \tag{8}$$

In this paper, the filter coefficient vector $\boldsymbol{b}(i)$ is adjusted by the steepest descent algorithm so that the error energy $J(i)$ is minimized. The updating equation is given by

$$\boldsymbol{b}(i+1) = \boldsymbol{b}(i) - \mu\nabla J(i) \tag{9}$$

where $\mu$ is the step size. Substituting (8) in to (9), it yields

$$\boldsymbol{b}(i+1) = \boldsymbol{b}(i) + 2\mu\boldsymbol{\Psi}(i)^T\boldsymbol{e}(i) \tag{10}$$

Given the input graph signal $\boldsymbol{f}(i)$, desired signal $\boldsymbol{f}_d(i)$, Laplacian matrix $\boldsymbol{L}$, step size $\mu$, initial coefficient vector $\boldsymbol{b}(0)$, filter order $K$, training data length $N_x$ and time index $i = 0$, the procedure of the proposed LMS graph adaptive filter is listed below:

(p1): Use (2) to compute the graph signals $\boldsymbol{f}_k(i) = \boldsymbol{L}^k \boldsymbol{f}(i)$ for $k = 0,1,2,\cdots,K$. (p2): Use (4a) to construct the matrix $\boldsymbol{\Psi}(i) = [\boldsymbol{f}_0(i) \quad \boldsymbol{f}_1(i) \quad \cdots \quad \boldsymbol{f}_K(i)]$. (p3): Use (3) to compute the output graph signal $\boldsymbol{f}_o(i) = \boldsymbol{\Psi}(i)\boldsymbol{b}(i)$. (p4): Use (5) to calculate the error graph signal $\boldsymbol{e}(i) = \boldsymbol{f}_d(i) - \boldsymbol{f}_o(i)$. (p5): Use (10) to update the vector $\boldsymbol{b}(i+1) = \boldsymbol{b}(i) + 2\mu\boldsymbol{\Psi}(i)^T \boldsymbol{e}(i)$. (p6): If $i + 1 > N_x$, stop the iterative loop. Otherwise, set $i \leftarrow i+1$ and go to (p1).

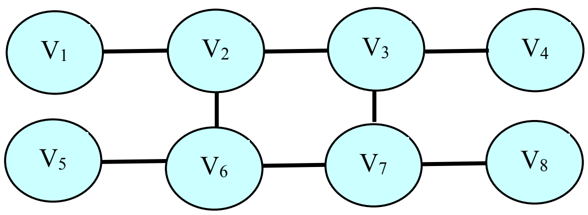


Fig.2 A undirected graph whose orders of vertices are labeled from left to right and top to bottom. The weights at all edges are equal to one.

## III. NUMERICAL EXAMPLES

Now, two examples are used to show the effectiveness of the proposed adaptive graph filtering algorithm.

**Example 1**: This simulation concerns the tracking capability of adaptive graph filter in system identification problem. Here, the following time-varying unknown system is considered:

$$\boldsymbol{f}_d(i) = \begin{cases} \boldsymbol{f}(i) + 2\boldsymbol{L}\boldsymbol{f}(i) + 3\boldsymbol{L}^2\boldsymbol{f}(i) & i \le 1000 \\ \boldsymbol{f}(i) + 3\boldsymbol{L}\boldsymbol{f}(i) + 4\boldsymbol{L}^2\boldsymbol{f}(i) & i > 1000 \end{cases} \quad (11)$$

where $\boldsymbol{L}$ is the Laplacian matrix of the undirected graph in Fig.2 and the elements of input graph signal $\boldsymbol{f}(i)$ are uniform random variables in the interval [0,1]. In (11), the coefficients of system are suddenly changed from (1,2,3) to (1,3,4) at time index $i = 1000$, so it is a time-varying system. The parameters of adaptive graph filter are chosen as $\mu = 0.04$, $N_x = 2000$, $K = 2$ and zero initial $\boldsymbol{b}(0) = \boldsymbol{0}$. Fig.3 depicts the learning curves of three filter coefficients $b_0(i)$, $b_1(i)$ and $b_2(i)$. Obviously, three curves converge to the true system parameters (1,2,3) for the first 1000 input signals. When the index $i > 1000$, filter parameters change gradually from (1,2,3) to (1,3,4) for minimizing the errors. So, time-varying unknown system has been successfully tracked by the adaptive graph filter.

**Example 2**: Consider a noisy graph signal measurement

$$\boldsymbol{y} = \boldsymbol{f}_t + \boldsymbol{\eta} \quad (12)$$

where $\boldsymbol{f}_t$ is a true signal and $\boldsymbol{\eta}$ is uncorrected additive Gaussian noise. In [3], it has been shown that true signal $\boldsymbol{f}_t$ can be estimated from noisy signal $\boldsymbol{y}$ by

$$\hat{\boldsymbol{f}}_t = (\boldsymbol{I} + \gamma\boldsymbol{L})^{-1}\boldsymbol{y} \quad (13)$$

The advantage of (13) is a closed-form solution, but it needs to compute the matrix inversion. In order to avoid the computation of matrix inverse, it is wanted to use the graph filter to approximate the de-noising operator $(\boldsymbol{I} + \gamma\boldsymbol{L})^{-1}$. This problem can be solved by the adaptive graph filtering algorithm if the desired signal $\boldsymbol{f}_d(i)$ in Fig.1 is chosen as

$$\boldsymbol{f}_d(i) = (\boldsymbol{I} + \gamma\boldsymbol{L})^{-1}\boldsymbol{f}(i) \quad (14)$$

In this example, The parameters of adaptive graph filter are chosen as $\mu = 0.04$, $N_x = 1000$, $K = 3$, $\gamma = 0.2$ and initial $\boldsymbol{b}(0) = \boldsymbol{0}$. The $\boldsymbol{L}$ is the Laplacian matrix of the graph in Fig.2 and the input graph signal $\boldsymbol{f}(i)$ are uniform random variables in [0,1]. Fig.4 depicts the learning curves of four filter coefficients. After adaptive algorithm converges, the error norm $\left\| \sum_{k=0}^{K} b_k(i)\boldsymbol{L}^k - (\boldsymbol{I} + \gamma\boldsymbol{L})^{-1} \right\|_2$ is equal to $0.012$ at $i = 1000$. Because the error is small, the de-noising operator is well approximated.

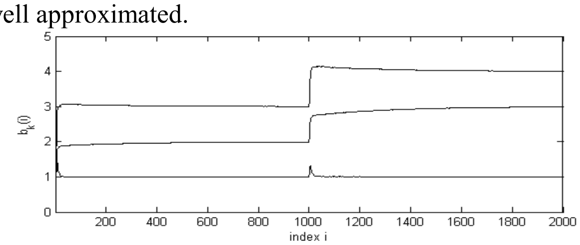


Fig.3 Results of time-varying system identification by adaptive graph filter. The learning curves of three filter coefficients $b_0(i)$, $b_1(i)$ and $b_2(i)$.

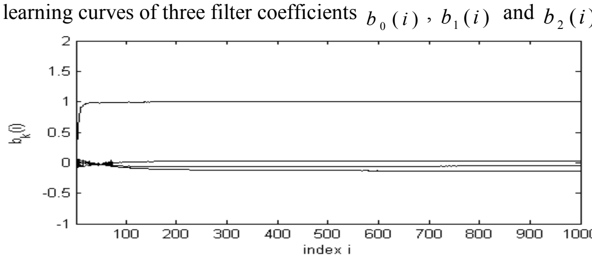


Fig.4 Results of de-noising operator approximation by adaptive graph filter. The learning curves of filter coefficients $b_0(i)$, $b_1(i)$, $b_2(i)$ and $b_3(i)$.

## IV. CONCLUSIONS

In this paper, an adaptive graph filter based on LMS algorithm has been presented. However, only LMS criterion is studied here. Thus, it is interesting to use other learning criterion to develop adaptive graph filter in the future.

## REFERENCES


[1] S. Haykin, *Adaptive Filter Theory*, Fifth Edition, New York: Pearson, 2014.

[2] S. Chen, A. Sandryhaila, J.M.F. Moura and J. Kovacevic, "Adaptive graph filtering: Multiresolution classification on graphs," in *Proc. of IEEE Global Conf. on Signal and Information Processing*, pp.427-430, Dec. 2013.

[3] D.I. Shuman, S.K. Narang, P. Frossard, A. Ortega and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol.30, no.3, pp.83-98, May 2013.