Fast Projection Algorithm and Its Step Size Control

Masashi Tanaka, Yutaka Kaneda, Shoji Makino and Junji Kojima

NTT Human Interface Laboratories
3-9-11, Midori-Cho, Musashino-shi, Tokyo 180 Japan
E-mail: tanaka@splab.ntt.jp

Abstract

This paper provides a fast Projection algorithm and a step size control to obtain the same steady-state excess mean squared error (MSE) for various projection orders. Computer simulations for colored noise and speech input signal confirm the effectiveness of the Projection algorithm and the step size control.

1. Introduction

Of the many adaptive filtering algorithms, the Normalized LMS (NLMS) algorithm is generally used in practice because of its simplicity. The computational complexity of the NLMS algorithm is low, however, convergence is very slow and tracking is poor for a colored input signal such as speech.

The RLS (Recursive Least Squares) algorithm, on the other hand, has the same convergence speed for both a colored input signal and a white signal, but its large computational complexity is a drawback.

In recent years, an algorithm called Projection (or Affine Projection) [1] has been drawing attention. This algorithm has properties that lie between those of the NLMS and RLS, i.e. less computational complexity than RLS but much faster convergence than NLMS for an input signal such as speech which can be modeled as a low-order AR process. The Projection algorithm, however, still needed more computation than NLMS, which was a problem.

Recently, efforts to reduce the amount of computation have been made [2] [3] [4], and computational complexity has been significantly reduced. This paper describes the fast Projection algorithm and proposes a control method for so-called step size parameter to obtain the same steady-state excess MSE for various projection orders.

2. Fast Projection Algorithm

2.1 Projection Algorithm

The Projection algorithm was proposed as a generalization of the NLMS algorithm [1]. A block diagram of the conventional Projection algorithm is shown in Fig. 1, where $x(k)$ represents an input signal, $y(k)$ a desired signal, $\hat{y}(k)$ estimation value, $e(k)$ an estimation error, $\hat{\mathbf{h}}(k)$ an FIR filter coefficient vector, and $\mathbf{g}(k)$ decorrelation FIR filter vector.

The projection algorithm consists of four parts: decorrelation of the input data sequence, calculation of the decorrelation filter coefficients, convolution of $x(k)$ and $\hat{\mathbf{h}}(k)$ to generate $\hat{y}(k)$, and adjustment of the filter $\hat{\mathbf{h}}(k)$ using the decorrelated input signal as in the following equation.

$$\hat{\mathbf{h}}(k+1) = \hat{\mathbf{h}}(k) + \mu(k)\mathbf{X}(k)\mathbf{g}(k). \quad (1)$$

Here, $\mathbf{X}(k)$ is a $L \times p$ matrix defined as

$$\mathbf{X}(k) = [\mathbf{x}_L(k), \mathbf{x}_L(k-1), \ldots, \mathbf{x}_L(k-p+1)] \quad (2)$$

$$\mathbf{x}_L(k) = [x(k), x(k-1), \ldots, x(k-L+1)]^T. \quad (3)$$

$\mu(k)$, p, and L denote the step size, the projection order, and the length of the filter $\hat{\mathbf{h}}(k)$. The decorrelation filter vector $\mathbf{g}(k)$ is calculated as

$$\mathbf{g}(k) = (\mathbf{R} + \delta\mathbf{I})^{-1}\mathbf{e}(k). \quad (4)$$

Here,

$$\mathbf{R}(k) = \mathbf{X}(k)^T\mathbf{X}(k) \quad (5)$$

$$\mathbf{e}(k) = [y(k), y(k-1), \ldots, y(k-p+1)]^T - \mathbf{X}(k)^T\hat{\mathbf{h}}(k), \quad (6)$$

and δ denotes a small positive number for initializing and regularizing the covariance matrix $\mathbf{R}(k)$.

Updating the filter coefficients vector $\hat{\mathbf{h}}(k)$ requires $(p-1)L$ computational complexity for decorrelation,

0-7803-2431-5/95 $4.00

$O(p^3)$ for calculation of the decorrelation filter, and $2L$ for filter adjustment and convolution.

2.2 Fast Projection Algorithm

We omit the detailed derivation of the fast algorithm but only list its processes in List 1. The reduction of computational complexity is achieved by the recursive update of the decorrelation filter vector $\mathbf{g}(k)$ [3] [4] (6 and 7 in List 1) and the employment of a filter $\mathbf{z}(k)$ an approximation of $\hat{\mathbf{h}}(k)$ [2] [4] [6](2 and 9 in List 1). In List 1, $\mathbf{a}(k)$ denotes the forward linear prediction filter, $\mathbf{b}(k)$ the backward linear prediction filter, $F(k)$ the minimum value of the sum of forward a posteriori prediction-error squares, and $B(k)$ the minimum value of the sum of backward a posteriori prediction-error squares.

The computational complexity is compared with that of other conventional algorithms in Table 1.

Table 1: comparison of computational complexity
(*L*: filter length, *p*: projection order)

Conventional Projection	$(p+1)L+O(p^3)$
Fast Projection	$2L+20p$
NLMS	$2L$
Fast RLS	$8L$

3. Step Size Control

In the Projection algorithm, the relationship between step size and the steady-state excess MSE depends on the projection order. A step size that brings about some steady-state excess MSE for one projection order does not yield the same steady-state excess MSE for different projection orders. This makes choosing a step size very troublesome. To overcome this problem, we propose a time varying step size, which is controlled using the sum of prediction error squares $F(k)$.

Let us write the optimum value for the estimated impulse response as $\mathbf{h}_o$, then the desired signal $y(k)$ is decomposed into a response of the optimum filter and an additive noise $n(k)$ as follows,

$$y(k) = \mathbf{x}_L(k)^T\mathbf{h}_o + n(k) \quad . \tag{7}$$

By substituting Eqs. (4), (6) and (7) to Eq. (1), we get,

List 1: fast Projection Algorithm

Subscripts of vectors represent the number of elements.
$\mu(k)$ is whether a constant value or controlled as stated in Sec. 3.

0. Initialization

$$\mathbf{r}_{p-1}(0) = \left[\mathbf{x}_L(0)^T\mathbf{x}_L(-1), \mathbf{x}(0)_L^T\mathbf{x}(-2)_L, \ldots, \mathbf{x}(0)_L^T\mathbf{x}_L(-p+1)\right]^T$$

$$\mathbf{e}_p(0) = 0, \mathbf{f}_{p-1}(0) = \mathbf{s}_{p-1}(0) = 0$$

$$F(0) = B(0) = \delta, \mathbf{z}_L(0) \text{:arbitrary}$$

Start with k=1.

1. $\mathbf{r}_{p-1}(k) = \mathbf{r}_{p-1}(k-1) + x(k)\mathbf{x}_{p-1}(k-1) - x(k-L)\mathbf{x}_{p-1}(k-L-1)$

2. $\hat{y}(k) = \mathbf{x}_L(k)^T\mathbf{z}_L(k) + \mathbf{r}_{p-1}(k)^T\mathbf{s}_{p-1}(k-1)$

3. $e(k) = y(k) - \hat{y}(k)$

4. $\begin{bmatrix}\mathbf{e}_p(k)\\ *\end{bmatrix} = \begin{bmatrix}e(k)\\ (1-\mu(k-1))\mathbf{e}_p(k-1)\end{bmatrix}$ $*$: don't care

5. Compute $\mathbf{a}_p(k)$, $\mathbf{b}_p(k)$, $F(k)$, and $B(k)$ by the sliding window version of FTF [5]

6. $\mathbf{g}_p(k) = (1-\mu(k-1))\begin{bmatrix}0\\ \mathbf{f}_{p-1}(k-1)\end{bmatrix} + \dfrac{\mathbf{a}_p(k)^T\mathbf{e}_p(k)}{F(k)}\mathbf{a}_p(k)$

7. $\begin{bmatrix}\mathbf{f}_{p-1}(k)\\ 0\end{bmatrix} = \mathbf{g}_p(k) - \dfrac{\mathbf{b}_p(k)^T\mathbf{e}_p(k)}{B(k)}\mathbf{b}_p(k)$

8. $\begin{bmatrix}\mathbf{s}_{p-1}(k)\\ s(k)\end{bmatrix} = \begin{bmatrix}0\\ \mathbf{s}_{p-1}(k-1)\end{bmatrix} + \mu(k)\mathbf{g}_p(k)$

9. $\mathbf{z}_L(k+1) = \mathbf{z}_L(k) + s(k)\mathbf{x}_L(k-p+1)$

$$\begin{aligned}\Delta(k+1) &= \hat{\mathbf{h}}(k) - \mathbf{h}_o \\ &= (\mathbf{I} - \mu(k)\mathbf{P}(k))\Delta(k) \\ &\quad + \mu(k)\mathbf{X}(k)\mathbf{R}(k)^{-1}\mathbf{n}(k)\end{aligned} \tag{8}$$

where

$$\mathbf{P}(k) = \mathbf{X}(k)\mathbf{R}(k)^{-1}\mathbf{X}(k)^T \tag{9}$$

$$\mathbf{n}(k) = [n(k), n(k-1), \ldots, n(k-p)]^T. \tag{10}$$

The second term on the right side of Eq. (8) disturbs the convergence of $\hat{\mathbf{h}}(k)$ to $\mathbf{h}_o$. Its square norm N_p is estimated as

$$N_p = E\left[\left\|\mu(k)\mathbf{X}(k)\mathbf{R}(k)^{-1}\mathbf{n}(k)\right\|^2\right] \approx \mu(k)^2 pF(k)^{-1}\sigma_n^2 \quad \because (\mathbf{R}(k)^{-1})_{ii} \approx F(k)^{-1} \quad (11)$$

where σ_n^2 represents the variance of $n(k)$ which we assume as white noise. As is known, for NLMS with white noise input, from Eqs. (8) and (11), steady-state excess MSE is estimated as

$$MSE \approx E\left[\|\Delta(k)\|^2\sigma_x^2\right] \approx E\left[\Delta(k)^T\mathbf{P}(k)\Delta(k)\sigma_x^2\right] \approx \frac{N_{p=1}\sigma_x^2}{\mu_1(2-\mu_1)} = \frac{\mu_1\sigma_n^2}{(2-\mu_1)} \quad (12)$$

$$\because E[\|\Delta(k)\|^2] = E[\|\Delta(k+1)\|^2], \quad E[\mathbf{P}(k)] = \mathbf{I},$$

where σ_x^2 represents L times of the variance of $x(k)$, and μ_1 a constant step size for NLMS.

We experimentally found that steady-state excess MSEs for $p > 1$ coincides if step size $\mu(k)$ is controlled so that the following relationship holds.

$$\frac{\mu'(k)p\sigma_n^2}{(2-\mu'(k))} = \frac{\mu_1\sigma_n^2}{(2-\mu_1)} \quad (13)$$

$$\mu'(k) = 2\sqrt{F(k)^{-1}\sigma_x^2}\,\mu(k) \quad (14)$$

The left side of Eq. (13) is equal to $E\left[\Delta(k)^T\mathbf{P}(k)\Delta(k)\sigma_x^2\right]$ when the input signal is $\sqrt{F(k)^{-1}\sigma_x^2}\,x(k)$ and the step size is replaced with $\mu'(k) = 2\sqrt{F(k)^{-1}\sigma_x^2}\,\mu(k)$. Solving Eq. (13), we get a time varying step size $\mu(k)$.

$$\mu(k) = \frac{\mu_1}{\sqrt{F(k)^{-1}\sigma_x^2}((2-\mu_1)p+\mu_1)} \quad (15)$$

With this step size control method, the same steady-state excess MSE can be achieved for various projection orders.

4. Experimental Results

Figures 2 (a), (b), and (c) show convergence curves for the Projection algorithm, and NLMS($p = 1$). The experimental conditions of this computer simulations are: filter length $L = 100$ and the true impulse response is also 100, the power level of additive noise is –40 dB relative to the average power of $y(k)$. The step sizes and input signals are:

(a) $\mu_1 = 0.25$, colored noise generated through an 8-order IIR filter of $x(k) = \sum_{i=1}^{8}(-1)^i(i/10)x(k-9+i)$;

(b) $\mu_1 = 0.5$, colored noise generated through a 4-order IIR filter of $x(k) = 0.95x(k-1) + 0.19x(k-2) + 0.09x(k-3) - 0.5x(k-4)$; and

(c) $\mu_1 = 0.5$, speech signal.

The frequency-amplitude characteristics of the two IIR filters in conditions (a) and (b) are shown in Fig. 3.

From the Figs. 2 (a), (b), and (c), we can see that the Projection algorithm improves the convergence speed as the projection order p increases. This improvement can be achieved with additional computational complexity of $20p$. Also we can see that the steady-state excess MSEs for various projection orders coincide, excluding $p = 1$ in (b) and (c). These results confirm the effectiveness of the Projection algorithm and the step size control.

5. Summary

This paper reported a fast version of a Projection algorithm whose computational complexity is $2L + 20p$ (L is the filter length and p is the projection order), which is much smaller than the $(p+1)L + O(p^3)$ of the conventional algorithm and is comparable to NLMS of $2L$. We also described a step size control method that gives the same steady-state excess MSE for different projection orders. Experimental results showed the effectiveness of the Projection algorithm and the step size control.

References

[1] K.Ozeki and T. Umeda, "A adaptive filtering algorithm

using an orthogonal projection to an affine subspace and its properties," *Trans. ICEIE*, vol. J67-A, no.2, pp.126-132, Feb. 1984 (in Japanese).

[2] Y. Maruyama, "A fast method of projection algorithm," *proceedings of the 1990 IEICE fall conference*, p. B-744, Mar. 1990 (in Japanese).

[3] M. Tanaka, et al., "Reduction of computation for high order projection algorithm," *proceedings of the 1993 IEICE fall conference*, p. 1-103, Aug. 1993 (in Japanese).

[4] S. L. Gay, "A fast converging, low complexity adaptive filtering algorithm", *proceedings of 3rd International Workshop on Acoustic Echo Control*, pp.223-226, Sep. 1993.

[5] J.M Cioffi and Kailath, "Windowed fast transversal adaptive filter algorithms with normalization," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, no.3, pp.607-625, June 1985

[6] M. Tanaka, et al., "A fast projection algorithm for adaptive filtering," submitted to *IEICE Trans. on Fundamentals*

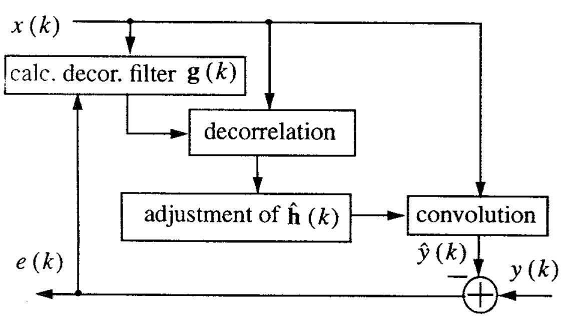


Figure 1. Block diagram of conventional Projection algorithm

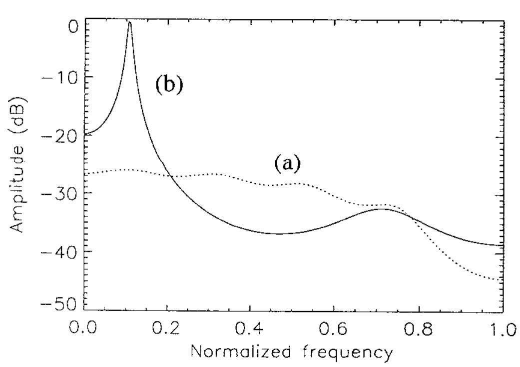
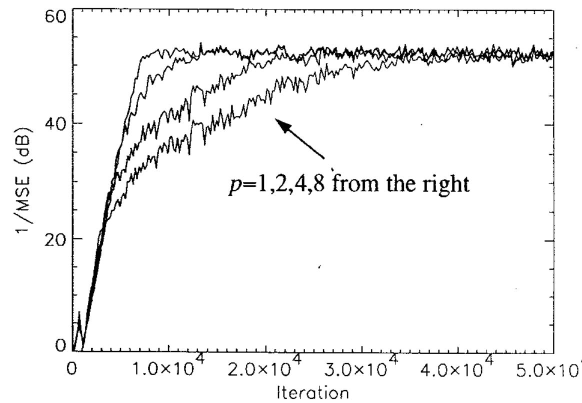
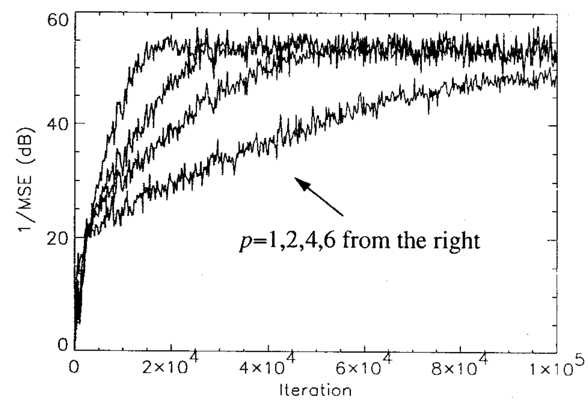


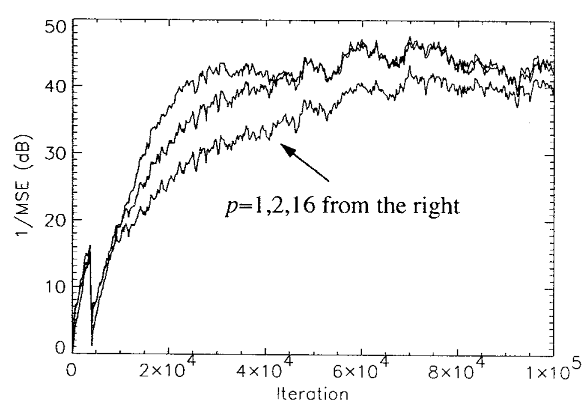
Figure 3 Frequency characteristics of two IIR filters (a) 8-order IIR filter (b) 4-order IIR filter.



(a) for a colored noise generated through an 8-order IIR filter, $\mu_1 = 0.25$.



(b) for a colored noise generated through a 4-order IIR filter, $\mu_1 = 0.5$.



(c) for a speech signal, averaged 50 trials, $\mu_1 = 0.5$

Figure 2. Convergence curves for various projection orders filter length $L = 100$, S/N=−40 dB.