A Training Method of Adaptive Filters for Inverse Filter Estimation in Noise Cancellation Problem

Asadul Huq† Zhiqiang MA‡ Kenji NAKAYAMA‡
†Graduate School of Natural Sci.& Tech., Kanazawa Univ.
‡Faculty of Technology.,Kanazawa Univ.
2-40-20 Kodatsuno,Kanazawa 920,Japan
Tel (0762)61-2101(372) Fax (0762)61-2509
E-mail: nakayamahaspnn1.ec.t.kanazawa-u.ac.jp

Abstract

In a noise canceler system, generally, adaptive filters have to approximate a pole zero transfer function. As the transfer function contains numerator polynomials, the inverse filtering in the noise cancelling problem, becomes more severe. In the case of an FIR adaptive filter, it is possible to approximate the pole zero transfer function using a sufficiently large number of taps. However, the convergence time depends on the step parameter of the algorithm, used in the adaptive process. Also, when zeros are close to the unit circle, the FIR requires a very long learning time to converge, in the case of inverse filtering. This paper proposes a learning method to overcome this slow rate of convergence. This method is mainly based on the fact that the NLMS algorithm converges very quickly if the initial tap vector is close to the optimum value. The proposed method works as follows. The problem is divided into two phases. In the first phase, the system identification is performed which generally requires a shorter learning time than that of the inverse filtering when the noise path contains zeros away from the center of the unit unit circle. The parameters estimated in this phase is then copied to the initial tap vector of the second phase. Now, in the second phase inverse filtering is performed. However, before, copying the tap vector to the second phase its inverse impulse response should be calculated out. This inversion requires no matrix inversion, since a sample by sample inversion is employed. As said earlier, the NLMS converges quickly when the initial tap vector is close to the optimum, the adaptive filter converges quickly in this phase.

So, in brief, this paper proposes a Two Phase Adaptation Process (TPAP) method for quicken the learning time of an adaptive process in an inverse filtering problem.

1 Introduction

Adaptive noise cancelling is needed where no prior knowledge of signal or noise characteristics are available. An adaptive filter uses an input as its reference data signal which is derived from one or more sensors placed in the noise field. The noisy signal works as the desired data signal of the adaptive filter. In this case of noise cancellation, the function of the adaptive filter is to improve the signal to noise ratio at its output in comparison with the same at the desired input. In noise cancellation problem, the adaptive filter try to estimate a transfer function, output of which is an estimation of the noise at the desired input. This generally, corresponds to inverse filtering. Inverse filtering itself is a slow process when the transfer function contains zeros away from the center of the unit circle. Specially, when the algorithm, used in the adaptive process, has a small step size, it takes a very long time to converge. In this paper, a two phase adaptive process (TPAP) has been proposed to cope with the this slow rate learning process and also a comparison between the proposed and the conventional method is provided. Here, we study the following subjects:

- Inverse filtering takes longer learning time compared to direct filtering when the zeros are away from the unit circle center.
- NLMS algorithm is sensitive to initial tap vector, i.e., it can converge faster if sarts from a tap vector which is closer to the optimum value.

Exploiting the above facts, we built the proposed method (TPAP). The proposed method has following advantages

- Faster than conventional method.
- Less sensitive to step parameter size.

The effectivity of the proposed method has been confirmed through computer simulations.

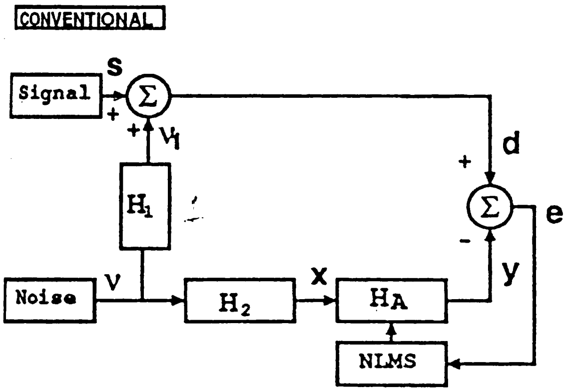

Figure 1: Conventional adaptive noise canceler

2 Formulation of the Problem

Figure 1 shows a general noise cancel circuit, where H_A is the FIR adaptive filter transfer function. According to the figure, d,x and e are the desired, input and error data sequences respectively. H_1 and H_2 are two path transfer functions. Now, we write corresponding z-domain equations in case of no signal condition

$$X(z)[H_1(z) - H_2(z)H_A(z)] = E(z) \quad (1)$$

Now, our goal is to make the right side of the equation equals to zero, because it is nothing but the error itself. So, we can write

$$H_A(z) = \frac{H_1(z)}{H_2(z)} \quad (2)$$

So, in a noise canceler system, an adaptive filter $H_A(z)$ must approximate a transfer function $\frac{H_1(z)}{H_2(z)}$ as expressed in above equation. Now, as the term $1/H_2(z)$ exists, the problem can be regarded as an inverse filter estimation problem. $H_A(z)$ is assumed to be an adaptive FIR filter adapted by the NLMS algorithm in this paper. poles of the filters $H_1(z)$ and zeros $H_2(z)$ are located inside the unit circle. Therefore, the theoretical optimum solution can exist, which has a rational function of z^{-1},that is an IIR filter. However, even $H_A(z)$ is an FIR filter, it is possible to approximate $\frac{H_1(z)}{H_2(z)}$ by $H_A(z)$ using a sufficient number of taps.

The time domain equation for the general adaptive noise canceler can be written as follow

$$e(n) = s(n) + \nu_1(n) - \mathbf{W}^T(n)\Phi(n) \quad (3)$$

where, $\mathbf{W}^T(n)$ is the adaptive filter coefficients and $\Phi(n)$ is the filter input vector. ν_1 and s are signal and noise respectively. After convergence, $\mathbf{W}^T(n)$ should be equal to the impulse response of $\frac{H_1(z)}{H_2(z)}$ ideally. i.e., the e(n) at the ouput of the adaptive filter equals to zero. It should be noted here that, the canceler output

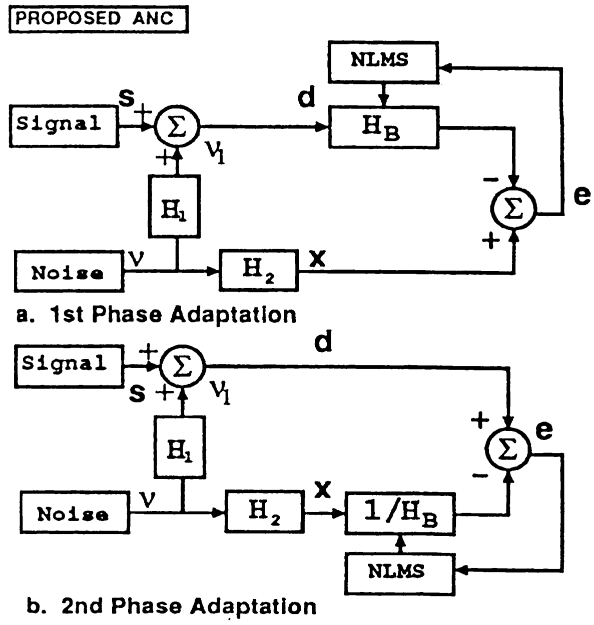

Figure 2: Two phase adaptive noise canceler

e(n) not only contain the $\nu_1(n)$, the noise term but also the signal s(n). The adaptation process reduces the error term e(n) at the filter output without any reduction of the signal itself i.e., s(n). However, this is only true in only ideal case, where no *leaking* of the signal component into the input data is confirmed.

In our problem, as we said earlier, the problem is divided into two phases. Figure 2 shows the method. Figure 2a shows first phase where desired signal is applied at the adaptive filter input and the ouput of H_2 is used as the desired data. So, in this condition, a direct system identification is performed. After convergence, the tap vector is stored and the adaptive filter is switched to the second phase as shown in Fig.2b. In this phase, the initialization of the tap vector is done by copying the tap vector stored un the previous phase. However, before using this vector for initialization in this stage, proper inversion should be carried out. This inversion requires no matrix inversion; a sample by sample inversing can be employed. The inverse impulse response of can be is calculated out as follows.

$$\begin{aligned} H_B(z) &= \frac{1}{h_0 + h_1 z^{-1} + h_2 z^{-2} + \ldots} \\ &= w_0 + w_1 z^{-1} + w_2 z^{-2} + .. \end{aligned} \quad (4)$$

Here,h_n and w_n are the tap vectors of the adaptive filters in the first and second phase respectively. Furthermore, it can be rewritten

$$[h_0 + h_1 z^{-1} + h_2 z^{-2} + \ldots][w_0 + w_1 z^{-1} + w_2 z^{-2} + ..] = 1$$

Comparing the polynomials in both sides, the following formula can be obtained

$$w_0 = 1/h_0$$

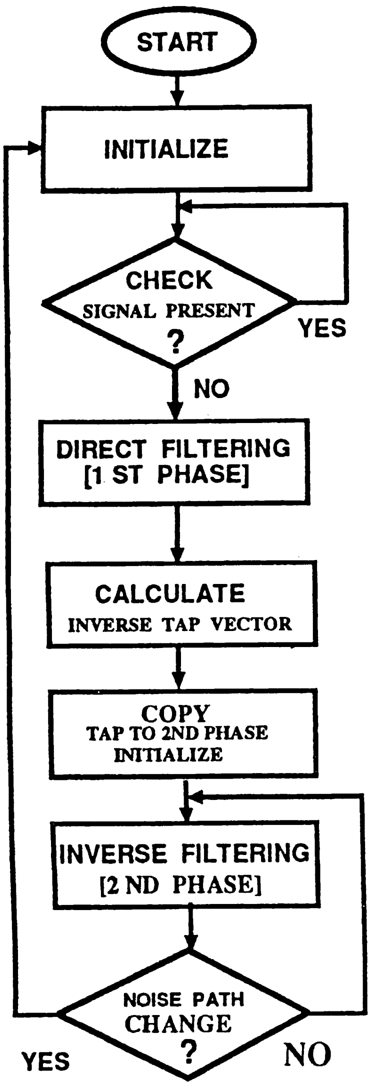

Figure 3: A flow chart for the proposed method

$$w_n = -\frac{1}{h_0}\sum_{i=0}^{n-1} w_i h_{n-i}, \quad n \geq 1 \tag{5}$$

Thus the inverse impulse response w_n can be calculated from w_0 to w_{N-1} sample by sample.

Now, a flow chart for the operations of the proposed method is shown in Fig.3. It is be noted here that the real application should require a no signal period to be detected. When, a no signal is detected, only then the operations for the first phase should be performed, otherwise the signal will interfere with the identification process. This no signal detection act has been included in the flow chart though this was not actually employed in the simulation.

3 Algorithm

Many adaptive algorithms have been proposed till now to solve these types of adaptive filtering problems, but each of them has its own drawbacks considering different actual conditions.

Here, in our case, we used the NLMS algorithm [1] in our simulations as we stated earlier. In the process of adaptation, the error sequence e(n) is obtained by substructing $\hat{d}(n)$, an estimate of the desired sequence d(n), from d(n) itself. The tap weights **W** are updated iteratively untill error term becomes significantly low. The algorithm can be summarized as follows

$$\begin{aligned} \mathbf{W}(n+1) &= \mathbf{W}(n) + 2\mu e(n)\mathbf{x}(n) \\ \mu(n) &= \frac{\bar{\mu}}{a + \sum_{k=1}^{M} \mathbf{x}^2(n-k+1)} \\ e(n) &= d(n) - \hat{d}(n) \\ \mathbf{x}^T(n) &= [x(n), \cdots, x(n-N+1)] \end{aligned} \tag{6}$$

4 Effects of the parameters of the Algorithm

4.1 Effect of Step Parameter

Step parameter is one of the prime factors that affect the LMS algorithm. When a small step parameter is used, convergence rate is slow. On the otherhand, when this parameter is large the convergence speed is relatively fast but in the expense of an increase in the residual error [2]. In this case, less data enter the estimation, hence, a degraded performance. So, in conventional method a small size step parameter is desirable. However, in the proposed method, for same step size, the learning time can be reduced significantly. This is possible because a two phase method is employed as explained before. Also, it is clear from the simulation results shown in the following section.

4.2 Effect of Initial Tap Value

Generally adaptive filtering algorithms starts with an initial tap value which equals to a null vector. However, from simulation results it has been found that the NLMS algorithm is very sensitive to the initial tap weights. The convergence rate is faster if the adaptation sarts from a tap vector which is closer to the optimum value. In a different problem, a pre-estimation method has been suggested when a minimum is far away from the initial tap vector [3]. However, in our proposed method, the convergence time has been reduced significantly by using a different initial tap value other than zero. This fact would be clear by inspecting the simulation results shown in the following section.

5 Simulation Results

Computer simulations were performed based on the general noise canceler configuration as shown in Fig.1. Conditions used in the simulation are summarized as follow.

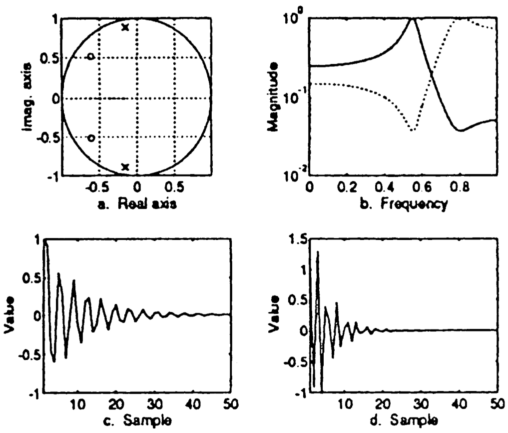

Figure 4: Noise path characteristics.
(a) Positions of zeros and poles.
(b) Frequency and inverse frequency response.
(c) Impulse response.
(d) Inverse impulse response.

- $H_1(z) = 1$
- $H_2(z)$ is a second order transfer function.
- No signal condition.
- Algorithm: NLMS
- Step parameters: 0.1 and 1.0

The transfer function of a second order system can be expressed as follows.

$$H(z) = c[\frac{1 - 2r_z cos\theta_z z^{-1} + r_z^2 z^{-2}}{1 - 2r_p cos\theta_p z^{-1} + r_p^2 z^{-2}}] \quad (7)$$

Here c is a scaling constant. The following table shows the pole-zero locations of the noise path filter.

TABLE 1

pole-zero locations of H_2

$r_p = 0.9$	$\theta_p = 100°$
$r_z = 0.8$	$\theta_z = 140°$

Now, in Fig.3 we see the noise path characteristics. Figure 3a shows the pole-zero position on the unit circle. Figure 3b shows the calculated frequency response of the transfer function and its inverse. The solid and the dotted plots show the frequency and the inverse frequency plots respectively. Figurs 3c and 3d show calculated impulse and inverse impulse responses respectively upto 50 samples.

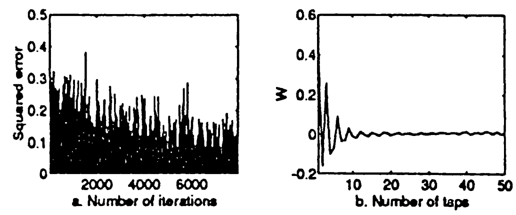

Figure 5: Simulation using conventional result.
(a) Squared error.
(b) Tap weights.

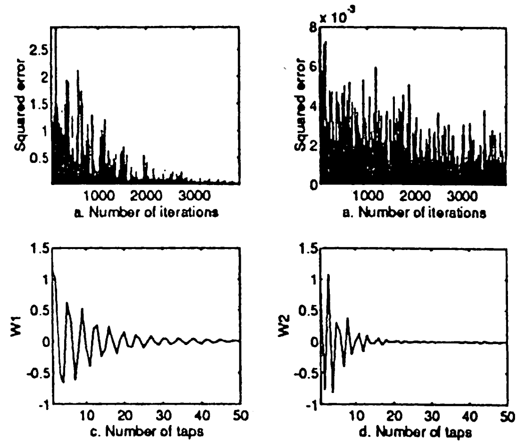

Figure 6: Simulation using proposed method.
for step parameter 0.10.
(a) Squared error at first phase.
(b) Squared error at second phase.
(c) Tap weights after first phase.
(d) Tap weights after second phase.

Now, we investigate the conventional inverse filtering used in the noise cancel problem. In this case, the value of step parameter was 0.1. Figure 5a shows the learning curve for the case. 8000 iterations were used in the simulation. Figure 5b shows the adaptive filter tap value after 8000 iterations. By, inspecting Fig.5a, we can guess that the full convergence is not yet obtained. Now, we investigate Fig.6, where the simulation result for the proposed method has been shown. Figure 6a shows the learning curve for the first phase whereas, Fig.6b shows the same for the second phase. We can notice that the error in the second phase is extremely small compared with the conventional method. Figures 6c and d shows the estimated tap values for first and second phase respectively. w_1 and w_2 corresponds to h and h_i respectively.

In brief, using same number of iterations, the proposed method can attain a smaller error compared to the conventional method. In other words, the learning time has been reduced in proposed method in comparison with the conventional method.

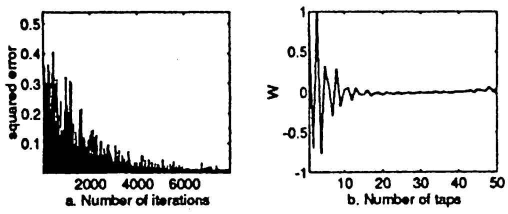

Figure 7: Simulation using conventional method.
for step parameter 1.0.
(a) Squared error.
(b) Tap weights.

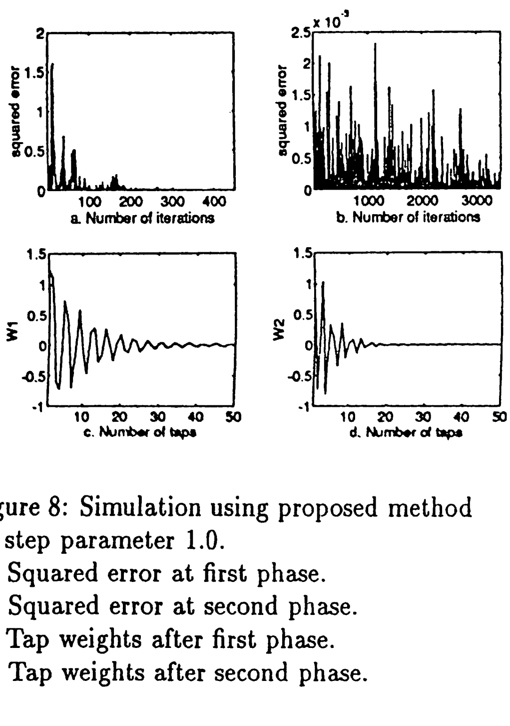

Figure 8: Simulation using proposed method
for step parameter 1.0.
(a) Squared error at first phase.
(b) Squared error at second phase.
(c) Tap weights after first phase.
(d) Tap weights after second phase.

Now, we investigate the second case, where the value of step parameter was 1.0. In this case, the result of conventional method has been shown in Fig.7. In this case, the conventional method approached convergence but the error after 8000 iteration is higher in comparison with the proposed method, shown Fig.8. Figure 7b shows the tap value for the conventional method after 8000 iterations. It is interesting to note that in this large step parameter case, the first phase of the proposed method took only about 450 iterations to converge, which is shown in Fig.8a. Figure 8b shows the learning curve for the second phase. We can notice that the error is very small in the the proposed method after completing two phases. Two phases aggregately took only 450+4000=4,4500 iterations to reach a very small error in comparison with the proposed method. Figures 8a and b show tap values for first phase and second phase respectively.

Hence, this example also shows that the proposed method is faster than the conventional one.

Now, also it has been found by computer simulations that a 10 percent variation from the optimum initial tap weights is tolerated by the NLMS in inverse identification problem. So, the possibility of a slow rate tracking is also expected.

6 Conclusion

In this paper, we investigated the inverse filtering problem in connection with the adaptive noise cancellation. In conventional noise caneling problem, a long convergence time is needed when zeros of the noise path locate away from the center of the of the unit circle. Also, NLMS algorithm is sensitive to initial tap vector. However, when the initial tap vector is close to the optimum, the convergence rate becomes quicker. Based on these facts, a Two Phase Adaptation Process (TPAP) method was proposed to cope with the slow convergence rate. This method, works as follows. In the first phase, a direct system identification is employed and the tap weights obtained after the convergence is stored. This tap vector is inversed sample by sample and used as the initial tap vector in the second phase. Now, in the second phase, an inverse filtering is performed. As the initial tap vector is calculated out from the estimated tap weights of the first phase, a quick convergence is obtained in this phase. The total time of convergence in two phases is less than the same in the conventional method, keeping the error constant. Simulations were performed using two different step parameters. In both cases similar results were obtained.

References

[1] B. Widrow and S. D. Stearms, Adaptive signal Processing, Prentice-Hall, Inc., Englewood Cliffs, N.J. 07632, 1985.

[2] Symon Haykin, Adaptive Filter Theory. 2nd edition, Prentice-Hall, 1992.

[3] L. Ljung, System Identification Theory for the User. Prentice Hall Information and System Sciences Series, 1987.