# A General Analysis and Complexity Reduction for the Lattice Transversal Joint Adaptive Filter

Jae Ha Yoo
Digital Media Research Laboratory, LG Electronics Inc.,
16 Woomyeon-Dong, Seocho-Gu, Seoul 137-724, Korea
E-mail : yjh@lge.com

**Abstract**: The necessity of the filter coefficients compensation for the LTJ adaptive filter was explained generally and easily by analyzing it with respect to the time-varying transform domain adaptive filter. And also the reduction method of computational complexity for filter coefficients compensation was proposed and its effectiveness was verified through experiments using artificial and real speech signals. The proposed adaptive filter reduces the computational complexity for filter coefficients compensation by 95%, and when the filter is applied to the acoustic echo canceller with 1000 taps, the total complexity is reduced by 82%

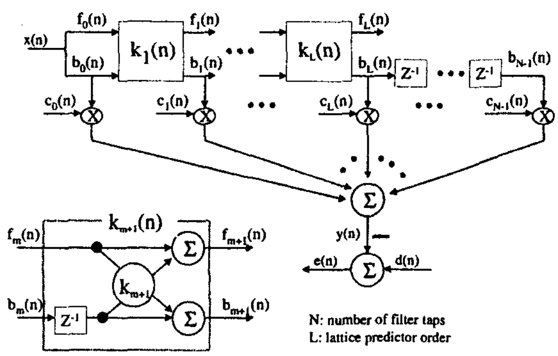


Figure 1. LTJ adaptive filter

## 1. Introduction

The adaptive lattice filter has been used in many applications where the adaptive transversal filter using the least mean square(LMS) algorithm cannot achieve fast convergence[1], and this filter can be implemented with relatively short lattice stages in speech applications. This adaptive lattice filter is called the lattice transversal joint(LTJ) adaptive filter[2].

However, the LTJ adaptive filter has a problem that the error level in the steady state is greater than that of the transversal filter. This problem can be solved if the reflection coefficients are frozen after some initial convergence. But this method is impractical when the input signal is non-stationary. So, this problem limits the application of the LTJ adaptive filter [3] to speech application, since it is non-stationary.

[4] showed that the reason for the instability of the LTJ adaptive filter is that the filter coefficients are updated one sample after the reflection coefficients update, and the problem can be solved through filter coefficients compensation. But this requires heavy additional complexity.

In this paper, the reason for the poor steady state performance of the LTJ adaptive filter is explained more generally than [4] and the complexity reduction method is proposed using the property that speech signals are stationary during a small time period.

## 2. LTJ adaptive filter

Figure 1 shows the LTJ adaptive filter. In this figure, $d(n)$ and $e(n)$ represent the desired signal and error respectively, and $c_m(n)$ is the coefficient of the transversal filter. $f_m(n)$ and $b_m(n)$ represent the forward prediction error and the backward prediction error respectively, and $k_m(n)$ is a reflection coefficient. The relations between signals and updating equations of the LTJ adaptive filter are shown in [2].

The reason for the lattice filtering is that backward prediction errors which are the input signals to the transversal filter, have no correlation through the lattice filtering, and so convergence speed can be improved.

Since speech signal can be modelled as an autoregressive(AR) process, the reflection coefficients after L-th stage can be regarded as zero. So, the reflection coefficients computation after L-th stage is not needed. It is this property that makes the LTJ adaptive filter efficient.

The LTJ adaptive filter is a potential choice in applications such as acoustic echo cancellation and adaptive noise cancellation since these applications require fast convergence because of the frequent change of the impulse response.

## 3. New Analysis

The LTJ adaptive filter can be regarded as a kind of transform domain adaptive filter since it uses backward prediction error signals that are a modified version of filter input signals by lattice filtering. The transform $K$ should be represented as time-varying transform $K(n)$ since reflection coefficient changes continuously in order to follow the statistics of the speech signal.

$$b(n) = K^T(n)x(n) \tag{1}$$

where $b(n)$ and input signal $x(n)$ are defined as follows.

$$b(n) = [b_0(n), b_1(n), \cdots, b_{N-1}(n)]^T \qquad (2)$$

$$x(n) = [x(n), x(n-1), \cdots, x(n-N+1)]^T \qquad (3)$$

$b(n)$ are weighted by filter coefficients $c(n)$ defined as follows.

$$c(n) = [c_0(n), c_1(n), \cdots, c_{N-1}(n)]^T \qquad (4)$$

Such processing makes filter output $y(n)$. By updating filter coefficient $c(n)$ using $b(n)$ and $e(n)$, new filter coefficients vector $c'(n)$ is generated and used the next time.

Namely, $c'(n)$ is an updated coefficient vector in the direction of minimizing error with respect to the signal vector $K^T(n)x$ where $x$ represents the filter input signal at an arbitrary time $t(t > n)$ and is defined as (3). At time $t = n + \Delta$, if the values of the $K(n+\Delta)$ are not the same as those of $K(n)$, $K^T(n+\Delta)x$ is a different signal vector from the signal vector $K^T(n)x$. Therefore, coefficient vector $c'(n)$ should be compensated in order to be suitable to the new signal vector.

In terms of vector space, $K^T(n+\Delta)x$ and $K^T(n)x$ are in the same vector space but have different basis vectors. So, the weight vector should be compensated in the new basis vectors so as to represent the same vector in the same vector space.

If the basis vectors $a$, $b$ and the respective weight vectors $c_a$ , $c_b$ are defined in the same vector space, the following relation has to be fulfilled so as to represent the same vector.

$$c_a^T a = c_b^T b \qquad (5)$$

Substituting $K^T(n+\Delta)x$ and $K^T(n)x$ for the basis vectors $a$ and $b$ makes the following relation.

$$c^T(n+\Delta)K^T(n+\Delta)x = c'^T(n)K^T(n)x \qquad (6)$$

Usually, the adaptive filter calculates filter output and coefficient update at every sample, so $\Delta$ is 1 and (6) becomes the following equation (7), which is the same as the equation (27) in [4].

$$K(n+1)c(n+1) = K(n)c'(n) \qquad (7)$$

Therefore, the compensated filter coefficients are as follows.

$$c(n+1) = K^{-1}(n+1)K(n)c'(n) \qquad (8)$$

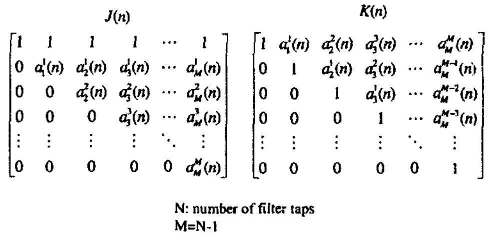


Figure 2. Matrix J(n) and K(n)

Table 1. Equations for elements of $J(n)$ and $K(n)$

*l:row index, m:column index*
*m=1;*
$J_1(n) = K_1(n) = [1, 0, 0, \ldots, 0]^T$
for *m=2:N*
$J_{l,m}(n) = J_{l,m-1}(n) + k_{m-1}(n)K_{l-1,m-1}(n-1)$
$K_{l,m}(n) = k_{m-1}(n)J_{l,m-1}(n) + K_{l-1,m-1}(n-1)$
end

## 4. Reduction Method

As shown in the last section, the LTJ adaptive filter should compensate the coefficient vector. And this compensation requires computation of the backward-prediction-error-generating matrix $K(n)$ and this $K(n)$ needs the forward-prediction-error-generating matrix $J(n)$. Figure 2 shows each matrix' component $J_{l,m}(n)$ and $K_{l,m}(n)$ where $a_i^j$ represents $j$-th filter coefficient of the forward prediction error filter of order $i$.

Components of the $J(n)$ and $K(n)$ are calculated as shown in table 1.

Since the reflection coefficients after L-th stage of the lattice predictor are all zero, the components of the $J(n)$ after L-th column are only delayed versions of the previous column like figure 3 where N and order L are 10 and 3 respectively.

As shown in [4] the filter coefficients compensation requires additional complexity of $2NL$ including that of

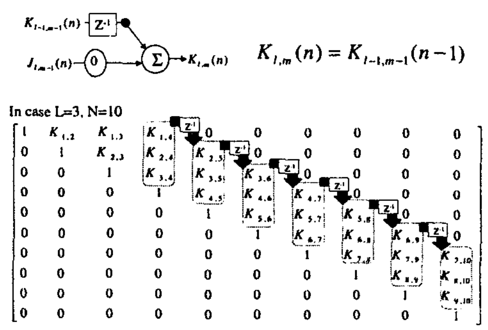


Figure 3. Matrix K(n)

Table 2. Comparison of computational complexities

| | Multiplication |
|---|---|
| LTJ- [4] | 3N+9L+2LN |
| LTJ-proposed | 3N+9L+2LN/D |
| LTJ-proposed (L=10, D=20) | 3N+9L+N |

forward and backward prediction error-generating transform matrix $J(n)$ and $K(n)$. This amount of complexity is a very heavy load for realtime implementation when we consider that the complexity of the adaptive transversal filter with normalized LMS(NLMS) is 3N.

Speech signals are non-stationary. But we can assume they are stationary during a small time period since the characteristics of speech change slowly in time and some parameters can be made fixed during a short time according to the short-time speech analysis[5].

The reflection coefficient is the parameter of the vocal tract since it is a different representation of the LPC filter coefficient. So, we can assume it does not change every sample but changes only certain time intervals. Namely, we can expect it does not disturb the system very much even if calculation for $J(n)$ and $K(n)$ and N-th order linear systems is not done every sample. So, the complexity of $2LN$ can be reduced to $2LN/D$ if the calculation for them is done every $D(D > 1)$ sample. If we find maximum D which can ignore the performance difference in that application, we can save the system resources considerably. Table 2 compares the computational complexity of the proposed method with that of [4]. In the case of L=10 and D=20, the proposed method reduces the complexity by 95%.

## 5. Experiment

To verify the performance of the proposed method, simulations were done with artificial and real speech signals. Simulation environments are same as [3]. The number of the taps for the plant and the adaptive filter are 30, and the power ratio between the plant output and the background noise is 40 dB. Misadjustments of the transversal and the LTJ adaptive filter are 10%.

### 5.1 artificial speech

Simulation using an artificial speech signal shows that filter coefficients compensation of the LTJ adaptive filter can improve the convergence speed and the LTJ filter does not make poor steady state performance even in case the lattice order is smaller than the actual LPC order. In figure 4, ERLE(Echo Return Loss Enhancement) is used as a performance index and we can see that the steady state performance of the LTJ adaptive filter is same as the transversal filter. The filter input signal was made by LPC synthesis filter of order 20 and the order of the LTJ adaptive filter is 10. This simulation shows that the LTJ adaptive filter can have fast convergence as well as same steady state performance even with smaller number of the lattice part.

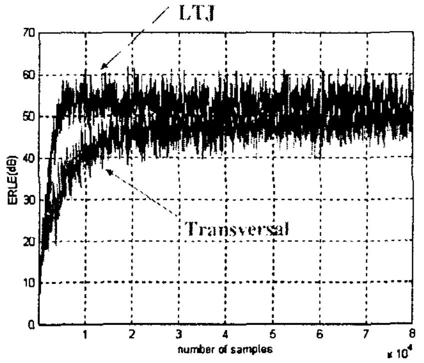


Figure 4. Experiment with an artificial speech signal

### 5.2 real speech

Figure 5 shows the performance with a real speech signal. (a) is the filter input speech signal, and (b) is the echo return loss enhancement(ERLE). The case of D=20 shows faster convergence compared with transversal filter and shows similar performance with D=1. Table 3 shows the convergence of the initial 1 second according to D values. The performance difference between D=20 and D=1 is less than 0.75dB on average, and this amount is less than 1dB and it is the same psychoacoustically.

Table 3. Initial convergence performance according to D values

| D | ERLE(dB) |
|---|---|
| 1 | 24.79 |
| 2 | 24.75 |
| 5 | 24.40 |
| 20 | 24.06 |
| 32 | 23.40 |

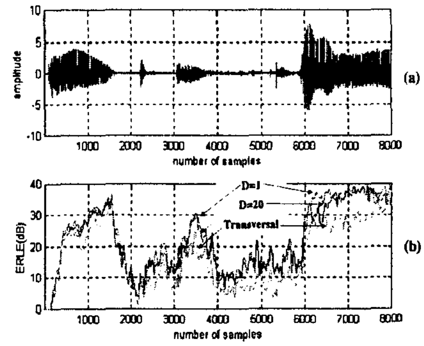


Figure 5. Experiment with a real speech signal

## 6. Conclusion

In this paper, the necessity of the filter coefficients compensation for the LTJ adaptive filter was explained by analyzing it with respect to the transform domain adaptive filter. The LTJ adaptive filter is a transform domain adaptive filter with time-varying transform and the signal vectors in the transformed domain are also time-varying. So, compensation for the filer coefficients is needed.

Using the speech signal's short time stationary property, the reduction method of computational complexity for filter coefficients compensation was proposed and its effectiveness was verified through experiments using artificial and real speech signals. The proposed adaptive filter reduces the computational complexity for filter coefficients compensation by 95%, and when the filter is applied to the acoustic echo canceller with 1000 taps, the total complexity is reduced by 82%

### References


[1] E. H. Satorious and S. T. Alexander, "Channel equalization using adaptive lattice algorithm," *IEEE Transactions on Communication*, vol. 27, pp. 899–905, 1979.

[2] Jae-Ha Yoo, Sung-Ho Cho, and Dae-Hee Youn, "An acoustic echo cancellation based on the adaptive lattice-transversal joint (ltj) filter structure," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E81-A, no. 9, pp. 1951–1954, 1998.

[3] B. Farhang-Boroujeny, *Adaptive Filters Theory and Applications*, chapter 11, John Wiley & Sons, 1998.

[4] Naoki Tokui, Kenji Nakayama, and Akihiro Hirano, "A synchronized learning algorithm for reflection coefficients and tap weight in a joint lattice predictor and transversal filter," in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, 2001, vol. 6, pp. 3741–3744.

[5] Douglas O'Shaughnessy, *Speech Communication Human and Machine*, chapter 6, Addison Wesley, 1987.