DSP Implementation and Performance Evaluation of a Sparse-Tap Adaptive FIR Filter with Tap-Position Control

タップ位置可変適応フィルタのDSP実現と性能評価

Akihiro Hirano, Michito Washizu† *and* Akihiko Sugiyama
平野晃宏 鷲津導人† 杉山昭彦

Information Technology Research Laboratories, NEC Corporation
日本電気株式会社 情報メディア研究所

Department of Electrical Engineering, Nagaoka University of Technology†
長岡技術科学大学 電気系†

ABSTRACT

This paper presents implementation and performance evaluation of a sparse-tap adaptive FIR filter with tap-position control. For reduction of computational costs, a small number of filter coefficients are assigned to significant regions in the impulse response. Implementation of adaptive filters using two diffrencet floating-point digital signal processors (DSPs), μPD77230 and ADSP-21020, is described. Assuming a typical satellite communication channel, the number of computations can be reduced by almost 90% for both DSP's compared with that of a full-tap normalized LMS algorithm. The implemented echo cancellers reduce echoes by almost 20dB for both white-noise and real speech signals. Tracking performance for phase rolls and flat-delay changes is also examined.

あらまし

タップ位置可変適応フィルタのディジタルシグナルプロセッサ(DSP)を用いた実現と、性能評価について報告する。少数の係数を応答波形部のみに割り当てることによって、演算量を削減している。2種類の浮動小数点DSP、μPD77230とADSP-21020を用いた実現法を検討し、通常の衛星通信回線に適用した場合に、演算量を全タップの学習同定法に比べて約1/10に削減できることを示す。白色雑音および音声を用いた測定で、エコーを約20dB削減できることを示す。また、インパルス応答の極性反転や遅延時間の変動にも追従できることを示す。

1. Introduction

In long distance communincation channels, echo cancellers are widely used to reduce echoes generated by hybrid transformers. Figure 1 shows an example of such a system, a satellite communication.

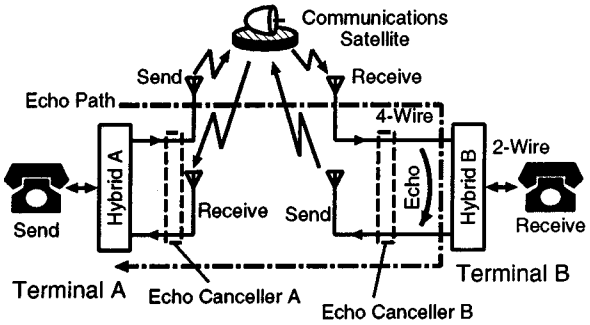


Fig. 1. Echo cancellation in satellite Communications.

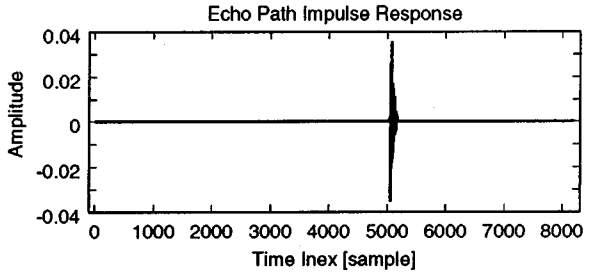


Fig. 2 Echo path impulse response.

Usually, echoes generated by Hybrid B at Terminal B are cancelled by Echo Canceller B at the same terminal. However, in some systems, Terminal B does not have an echo canceller and Echo Canceller A at Terminal A should cancel the echoes. In this case, the echo path consists of a short dispersive region and long flat-delays. Figure 2 demonstrates an impulse response of such an echo path.

Cancellation of such an echo by an FIR filter requires a large number of taps just for spanning the long flat-delays, typically more than 500 milliseconds. To reduce a large number of computations for an FIR filter with a large number of taps, scrub taps waiting in a queue (STWQ) algorithm has been proposed[1]. This algorithm uses a small number of filter coefficients, whose positions are independently controlled.

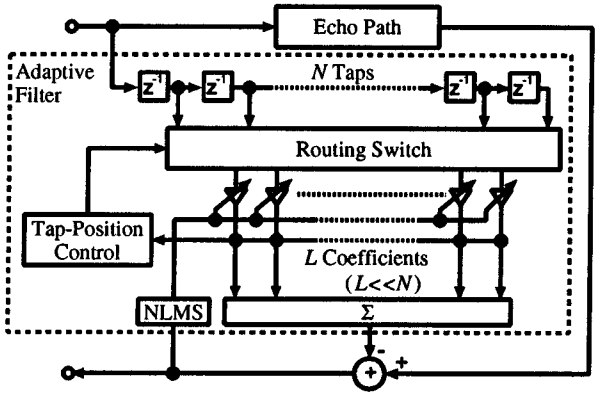


Fig. 3. Adaptive filter with tap-position control.

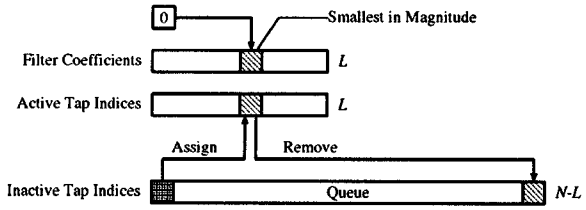


Fig. 4. Tap-position control by STWQ algorithm.

Figure 3 shows the block diagram of an echo canceller equipped with STWQ algorithm. Filter coefficients are connected to a tapped delay-line via a routing switch. STWQ algorithm can reduce the number of filter coefficients at the expense of additional operations for routing switch and tap-position control. However, neither comparison of total computational amount nor implementation of these algorithm has been reported.

This paper presents implementation of an adaptive filter based on STWQ algorithm using digital signal processors (DSP's) and performance evaluation of the implemented adaptive filters. The number of computations is examined for two DSP's with completely different architecture, followed by hardware implementation for both DSP's. Experimental results show the performance of the implemented echo cancellers for a white noise input and a speech signal. Tracking characteristics for phase rolls and flat-delay changes are also examined.

2. Tap-Position Control by STWQ Algorithm

STWQ algorithm[1] selects L active taps with filter coefficients from N taps. The other taps (inactive taps) do not have coefficients. The active taps are selected based on the coefficient values.

Figure 4 demonstrates the tap position control by STWQ algorithm. The indices of the inactive taps are stored in a queue. After Q iterations of coefficient updates by the normalized LMS (NLMS) algorithm[2], an active tap with the smallest coefficient in magnitude is made inactive. Then, an inactive tap index is taken from the queue and an inactive tap corresponding to the index is made active. The filter coefficient for newly activated tap is reset to zero.

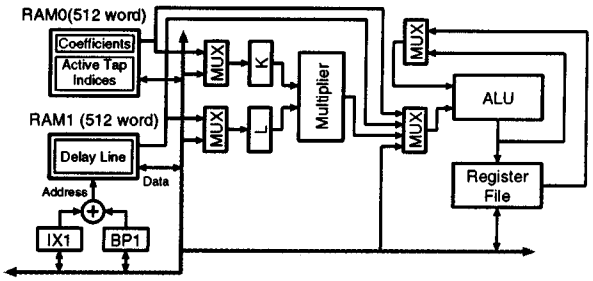


Fig. 5. Block diagram of μPD77230.

3. Considerations for DSP Implementation

In DSP implementation of STWQ algorithm, realization of the routing switch is important. Reading input signals from the tapped delay line through the routing switch takes several instruction cycles for circular buffer operation and address transformation. Finding the smallest coefficient in magnitude may also require several instruction cycles. If a DSP has deep pipeline stages, finding the minimum may be critical. Since the number of computations depends on the DSP architecture, implementation by two floating-point DSP's with completely different architecture, μPD77230[3] by NEC and ADSP-21020[4] by Analog Devices, has been examined. Figures 5 and 6 depict brief block diagrams of these DSP's, in which only important portions are shown.

3.1. Implementation of Routing Switch

The routing switch is realized by two memory read operations: reading the input signal address or the active tap index and then reading the input signal itself. Additional operations may be necessary to cope with a circular buffer overflow.

In implementation using μPD77230, circular buffer operation requires no additional operations. Using whole RAM1 as the tapped delay line makes realization of the circular buffer simple. The index register IX1 holds the memory address which corresponds to the top of the delay line. The active tap index is loaded to the base pointer BP1. The sum IX1+BP1, which is automatically calculated in the "base plus index" addressing mode, is used as the data address. Even if $512 \le IX1+BP1$, the output of the address adder is automatically fixed to IX1+BP1-512.

In implementation using ADSP-21020, the same simplification as for μPD77230 is not applicable because of its wide memory space: 4 gigawords for Data Memory (DM) and 16 megawords for Program

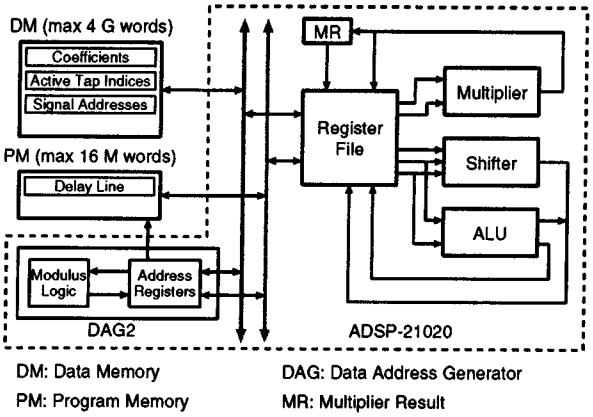


Fig. 6. Block diagram of ADSP-21020

Memory (PM). The circular buffer is implemented by a modulus logic in a data address generator (DAG). The input signal address in the buffer, rather than the active tap index, is loaded to an address register in the DAG. The address is updated and then restored into the DM.

Overhead for these operations is minimized by reducing the number of such address manipulations to only once per active tap in a sampling period. All computations, i. e., coefficient adaptation, convolution, power calculation, and finding the smallest coefficient in magnitude, for an active tap in a sampling period are carried out in a single loop.

This optimization also minimizes the number of memory access for filter coefficients. Only one read and one write are performed for one filter coefficient. Note that minimizing the number of memory access is important for processors with a load-store architecture such as ADSP-21020, in which all calculation units read input data only from a register file and write the results back only to the register file. Such optimization is not always effective for μPD77230, which has dedicated data paths from data memories to calculation units.

3.2. Finding Smallest Coefficient in Magnitude

Finding the smallest coefficient in magnitude may take several instructions. For μPD77230, this is most critical. μPD77230 requires one wait cycle between a compare operation and a conditional branch. This DSP always executes an instruction just after a branch (delayed branch). These features may cause extra no-operation instructions. Two move instructions are required to save both the minimun value and the tap index.

After minimizing the number of instruction cycles, finding the minimum requires 6 instruction cycles per active tap. This is same as the total number of instructions per active tap for FIR filtering, power

Tab. 1. Number of Instructions

μPD77230			ADSP-21020	
NLMS	STWQ		NLMS	STWQ
	FIR+ Adapt	FIR+ Min		
$3.4N$	$6L$	$9L$	$2N$	$6L$

calculation, and coefficient adaptation. To reduce the total number of computations per sampling period, either coefficient adaptation or finding minimum is performed in a single sampling period.

ADSP-21020 also has deep pipeline stages; two instructions after a branch instruction are always executed. However, this DSP needs no wait cycles between a comparison and a branch. Zero-overhead conditional instruction, which means that one computation is executed only if a specified condition is satisfied, may reduce the number of extra no-operations caused by conditional branches.

Thanks to a combination of a conditional instruction and simultaneous data transmissions for both the minimum value and the tap index, no conditional branches are necessary. Execution of all computations for an active tap in a single loop also reduces overhead for finding the minimum. Calculation of squared coefficient, which is an alternative to the magnitude, and comparison of the squared value with the minimum value are simultaneously performed with other operations. Compared with a loop without finding the minimum, the number of additional instruction cycles for finding the minimum is only one cycle per active tap.

3.3. Number of Computations

Table 1 compares the number of instructions for the NLMS algorithm and STWQ algorithm. For both μPD77230 and ADSP-21020, the number of instructions per active tap necesary for STWQ algorithm is almost three times larger than that for the full-tap NLMS algorithm. If the number of total taps N is three times larger than the number of active taps L or more, STWQ requires smaller number of computations than the NLMS. Note that almost same results have been derived for two DSP's with completely different architecture.

Assuming an 8kHz sampling and a typical satellite communication channels, N is about 8000 for 1 second of total echo path delay. For four dispersive regions caused by hybrid transformers, L is about 256. The ratio L/N is about one-thirtieth. Therefore, total number of computations for STWQ algorithm can be reduced by almost one-tenth compared with that for

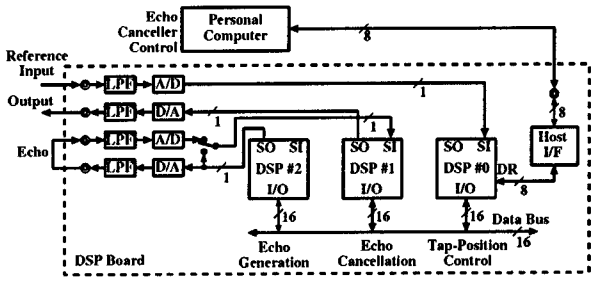


Fig. 7. Block diagram of μPD77230 DSP board.

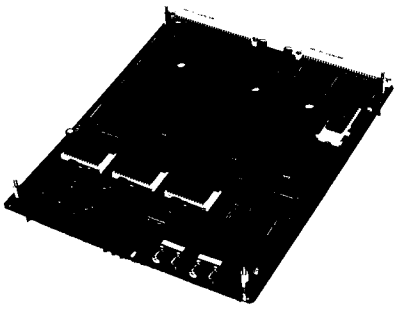


Fig. 8. Implemented echo canceller.

the full-tap NLMS algorithm.

The maximum number of active taps L_{max} is primarily limited by the processor speed. For an 8kHz sampling, L_{max} is about 64 for 6.5MIPS μPD77230 and about 512 for 33MIPS ADSP-21020. The maximum number of total taps N_{max} depends on both memory size and optimization scheme. For optimization, N_{max} for μPD77230 is chosen as 512, which is same as the data RAM size. N_{max} for ADSP-21020 is limited only by the memory size. Since this DSP has no internal memories, no limitations are caused by the internal memory size. Though expensive high-speed memories are required, the DSP can be equipped with any necessary amount of memories.

4. DSP Implementation of STWQ algorithm

4.1. Implementation using μPD77230

A μPD77230-based echo canceller has been implemented using a multiple-DSP system shown in Fig. 7. Three μPD77230s are used. DSP#0 controls active tap position and entire echo canceller. DSP#1 is for adaptive filtering and echo cancellation. Echo generator for performance evaluation is realized by DSP#2. The number of total taps N and the number of active taps L are chosen as 512 and 32. The adaptation step-size is 0.01. Though two DSP's are used for simplicity of programming and shortage of internal memory, only one DSP with external memory is enough. Figure 8 depicts the implemented echo canceller.

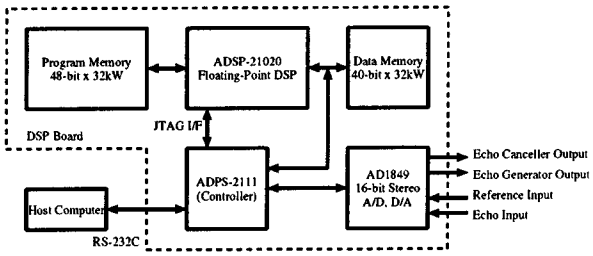


Fig. 9. Block diagram of ADSP-21020 DSP board.

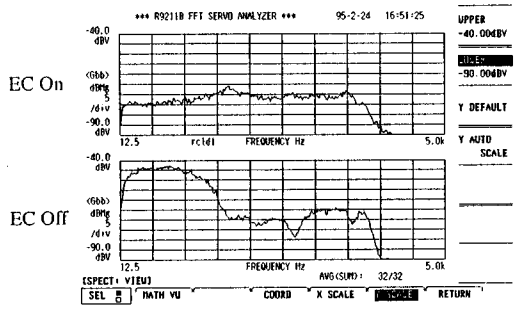


Fig. 10. Power spectrum of residual echo.

4.2. Implementation using ADSP-21020

An echo canceller based on ADSP-21020 has been implemented using EZ-LAB™ evaluation board[5] shown in Fig. 9. To cancel four echoes with reverberation time of eight milliseconds each and one second of end-to-end delay, the number of total taps N and the number of active taps L are chosen as 8192 and 256, respectively. Since this L is half of the maximum number of active taps L_{max}, two echo cancellers can be realized by using one DSP. The adaptation step-size is chosen as 0.25.

5. Experimental Results

The performance of the implemented echo cancellers have been measured for both white-noise input and real speech input. For the μPD77230-based echo canceller, the echo generator on the same DSP board has been used. The residual echo power has been analyzed by Advantest R9211B FFT servo analyzer. Figure 10 shows the power spectrum of the echo and the residual echo by μPD77230 based echo canceller. The echo is reduced by almost 20dB.

Echoes for the echo canceller based on ADSP-21020 have been generated by TASKIT series II telephone network simulator. The echo path consists of two dispersive regions; one is at the near end and the other is at the far end. Except for a flat-delay change test, the round-trip delay has been chosen as 600 milliseconds, which agrees with that of a typical satellite communication channel.

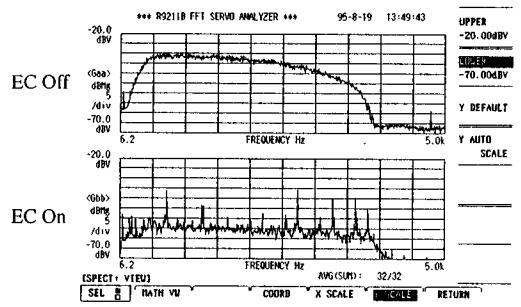


Fig. 11. Power spectrum of residual echo for white noise.

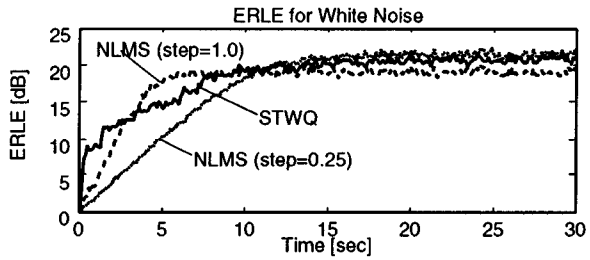
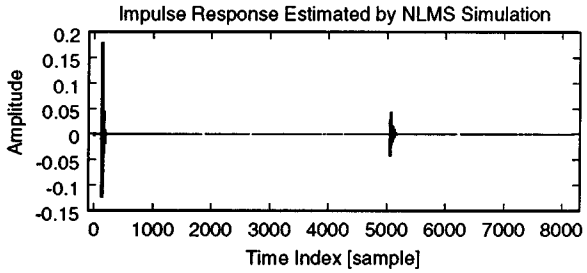


Fig. 12. ERLE for white noise.

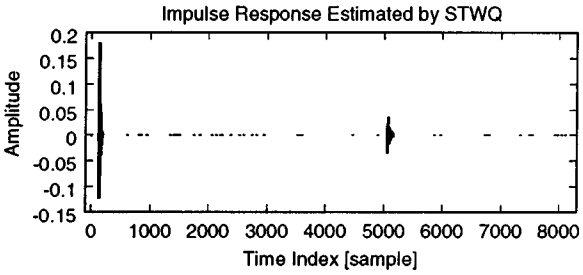
The residual echoes have been analyzed by the same FFT analyzer as that used for the μPD77230-based echo canceller. To examine the estimated impulse response, the active tap indices and the filter coefficients have been uploaded to the host computer. The reference inputs, the echoes and the residual echoes have been digitized for ERLE (echo return loss enhancement) calculation and computer simulations. The performance of the echo canceller using ADSP-21020 has also been compared with the computer simulation results assuming an 8192-tap FIR adaptive filter with the full-tap NLMS algorithm. Computer simulation results have been used because implementation of an 8192-tap FIR adaptive filter based on the full-tap NLMS algorithm is difficult.

The power spectrum of the residual echo is shown in Fig. 11. The echo is reduced by almost 25dB. Figure 12 compares the convergence characteristics of the implemented echo canceller with the simulation results for NLMS algorithm. In the first three seconds, the echo canceller achieves larger ERLE than the NLMS with the step size of 1.0. STWQ congerges faster than the NLMS with the step size of 0.25. Figure 13 shows that the impulse response estimated by the echo canceller agrees with that estimated by the full-tap FIR adaptive filter.

The tracking capability for phase rolls has also been tested. In the phase roll, the polarity of the far-end echo has been inverted. Figure 14 shows the tracking characteristics for phase rolls. The performance degradation of STWQ is smaller than that of the NLMS. The implemented echo canceller also



(a) NLMS



(b) STWQ

Fig. 13. Impulse Responses.

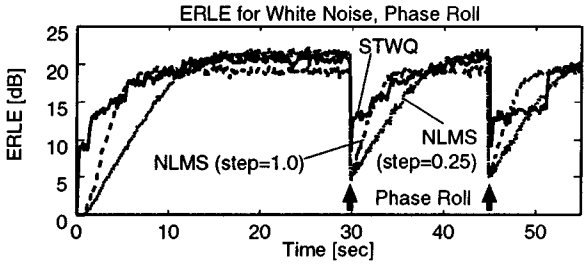


Fig. 14. ERLE for phase roll.

tracks the phase roll faster than the NLMS.

The tracking profile for flat-delay changes and white-noise input is demonstrated by Fig. 15. The round-trip delay has been changed from 400 milliseconds to 600 milliseconds at 30 seconds and then returned to 600 milliseconds again. Though the implemented echo canceller successfully tracks the delay changes, the tracking speed is slower than that of the NLMS. Two impulse responses for diferrent flat-delays estimated by STWQ are shown in Fig. 16. The echo canceller estimates the impulse responses correctly.

The performance for a speech input has also been evaluated. Figure 17 demonstrates the residual echo power for a female speech input. Figure 18 shows the measured power spectrum of the residual echo. The echo is reduced by about 15dB. The estimated impulse response is shown in Fig. 19. Though the near-end echo path is correctly estimated, the estimated impulse response for the far-end echo path is completely different from the optimum response

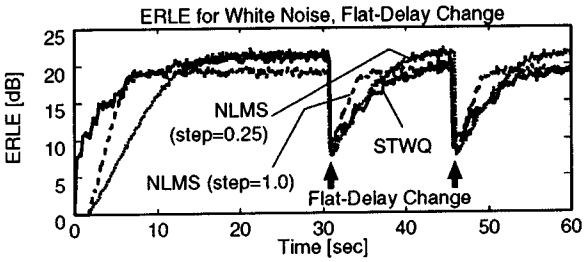


Fig. 15. ERLE for flat-delay change.

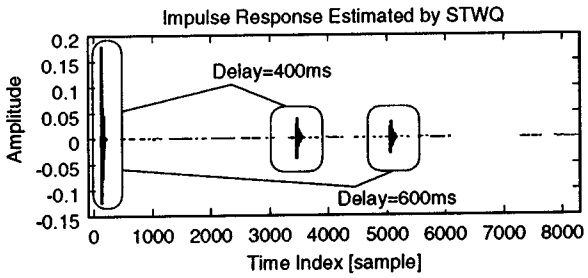


Fig. 16. Impulse responses for different flat-delays.

shown in Fig. 13. Further improvements are necessary in order to estimate the correct impulse response for speech inputs.

6. Conclusion

DSP implementation and performance evaluation of an adaptive FIR filter with tap-position control has been presented. Two STWQ-based echo cancellers, one uses μPD77230 and the other uses ADSP-21020, has been implemented. The number of computations can be reduced by almost 90% for typical satellite communication channels. The implemented echo cancellers reduce echoes by almost 20dB for a white-noise input. The echo canceller successfully tracks phase rolls and flat-delay changes. Though echoes can be reduce for real speech signals, further improvements are necessary for correct estimation of the impulse response.

Acknowledgments

The authors wish to thank Shigeji Ikeda, an assistant manager, and Nobuhiro Abe, both of Radio Application Division, NEC Corporation for providing us with the telephone network simulator and also helping the measurements. Without their help, these evaluation was not possible. They would like to thank Osamu Hoshuyama of Signal Processing Research Laboratory, Information Technology Research Laboratories, NEC Corporation for helpful comments and discussions. The authors are also indebted to Dr. Takao Nishitani, the senior manager of Signal Processing Research Laboratory, Information Technology Research Laboratories, NEC Corporation for his guidance and continuous encouragement.

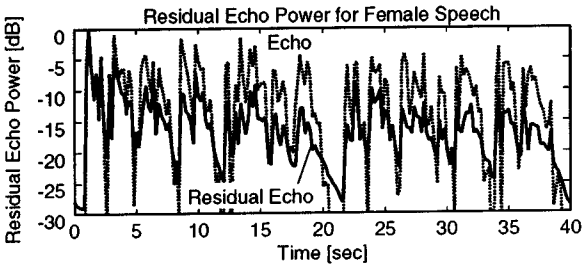


Fig. 17. Residual echo power.

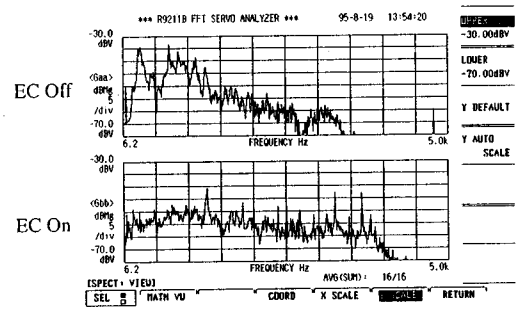


Fig. 18. Power spectrum of residual echo for female speech.

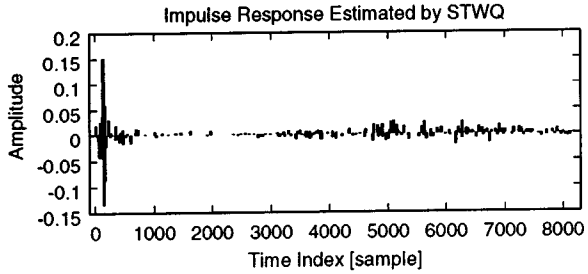


Fig. 19. Impulse response for female speech.

References

[1] S. Kawamura and M. Hatori, "A Tap Selection Algorithm For Adaptive Filters," Proc. of ICASSP '86, pp. 2979-2982, 1986

[2] J. Nagumo, et. al, "A Learning Method for System Identification," IEEE Trans. on Automatic Control, Vol. AC-12, No. 3, pp. 282-287, 1967.

[3] T.Nishitani, et. al."Advanced Single-chip Signal Processor," Proc. of ICASSP 86, pp.409-412, 1986.

[4] "ADSP-21020/21010 User's Manual," Analog Devices Inc., 1993

[5] "ADSP-21020 EZ-LAB Evaluation Board Manual," Analog Devices, Inc., 1992