IMPLEMENTATION OF WIENER FILTER ON FPGA USING XILINX SYSTEM GENERATOR FOR SPEECH ENHANCEMENT

Dr Neetu Agrawal,

Associate Professor

AIACT&R, Delhi

ABSTRACT

Implementation of Weiner Filter for Speech Enhancement on FPGA is presented in this paper. We use Xilinx System Generator as a software tool for design and implementation of Weiner Filter. Then, design is implemented and analyzed using the same tool. Xilinx Spartan 3E is used for implementation due to its low cost, and low power consumption. Time-domain analysis and spectrogram of original speech and enhanced speech are presented for comparison. FPGA resource analysis, and power analysis are also presented.

INTRODUCTION

The aim of speech enhancement is to improve the quality of speech which includes clarity, intelligence, and pleasantness. Produced speech signal can be used for human hearing, or machine recognition. The popular methods for enhancing speech are the removal of background noise, echo suppression and the process of artificially bringing certain frequencies into the speech signal. Noise reduction is the most important field of speech enhancement.

Digital Signal Processors or ASICs (Application Specific Integrated Circuits) are used for implementing digital filters on hardware [7,8]. However recent advances in technology have led to implementation of digital filter algorithms on FPGA.Experiments have been performed which indicates FPGAs can outperform Digital Signal Processor and ASICs [9, 13]. FPGA has been implemented in areas such as Digital Image Processing [10] and Digital Signal Processing [11, 12].

Programming FPGA using HDL languages such as VHDL and Verilog is a daunting task. Sometimes, it takes weeks or months to write a synthesizable code. To overcome such difficulties, FPGA manufacturers have developed high-level tools and languages, so that signal processing applications can be easily implemented on FPGA. One such tool is Xilinx System Generator. Xilinx System Generator (XSG) is a state of the art tool which offers high performance and requires very small learning and development time. XSG offers block libraries that plugs into Simulink tool (containing bit-true and cycle-accurate models of their FPGA's particular math, logic, and DSP functions). No knowledge of HDL languages is required to use this tool. Research papers have been proposed, in which many existing digital filter algorithms are implemented on FPGA hardware using XSG [14, 15].

Spectral subtraction is a popular approach for speech enhancement. Implementation of spectral subtraction on FPGA is presented in [1]. But

spectral subtraction suffers from the following problems [17]:

- residual noise
- musicality that results from coming and going of waves over successive frames

The above mentioned problems are solved by the use of Wiener filter.

In this paper Wiener filter is implemented on Xilinx Spartan 3E FPGA for the Speech Enhancement. Results are verified using co-simulation. For comparison, time-domain analysis and spectrogram of enhanced speech are presented along with original speech. FPGA resource estimation and power analysis are also presented.

This paper is organized as follows: In section II, brief introduction to Speech Enhancement using Wiener Filter is presented. In section III and section IV, FPGA Hardware tools, and Design Tools are described, respectively. Hardware Implementation of Wiener Filter is presented in Section V. Experimental Results are presented in Section VI. The conclusions are drawn in Section VII.

SPEECH ENHANCEMENT USING WIENER FILTER

Wiener Filter [2]

To recover speech signal$x[n]$ corrupted by additive noise $b[n]$, from a sequence $y[n] = x[n] + b[n]$, a linear filter $h[n]$ is to be used such that the sequence $\hat{x}[n] = y[n] * h[n]$ minimizes the expected value of $(\hat{x}[n] - x[n])^2$. Under the condition that the signals $x[n]$ and $b[n]$ are uncorrelated and stationary, the frequency-domain solution to this stochastic optimization problem is given by the suppression filter.

$$H_s(\omega) = \frac{S_x(\omega)}{S_x(\omega)+S_b(\omega)} \qquad (1)$$

Above equation is referred as Wiener filter.

For non-stationary and time-varying signal with stationary background noise, Wiener filter can be expressed as

$$H_s(pL, \omega) = \frac{\hat{S}_x(pL,\omega)}{\hat{S}_x(pL,\omega)+S_b(\omega)} \qquad (2)$$

Where $\hat{S}_x(pL, \omega)$ is an estimate of the time-varying power spectrum of $x[n]$, $S_x(n, \omega)$, on each frame, and $S_b(\omega)$ is an estimate of the power spectrum of a stationary background noise.

Smoothing

An approach to reduce annoying fluctuations by slowing down the rapid frame to frame movement of the object power spectrum is by application of temporal smoothing. The estimated smooth power spectrum is given by the equation,

$$\tilde{S}_x(pL, \omega) = \tau\tilde{S}_x\big((p-1)L, \omega\big) + (1-\tau)\hat{S}_x(pL, \omega) \quad (5)$$

Here,τ is the smoothing constant which controls how fast we adapt to non-stationary signal spectrum. If value of τ is low then time resolution is improved, but noise in estimated power spectrum and thus musicality is high. When large value of τ is used, estimated power spectrum improves in regions of stationarity, but smears rapid events.

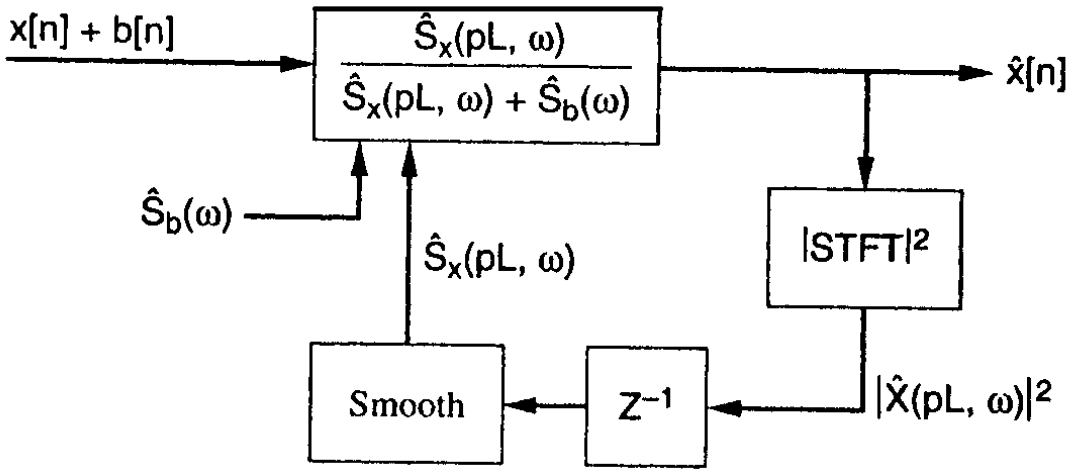


Fig.1. Traditional Wiener Filter with Smoothing of the Object Spectrum Estimate.

SOFTWARE TOOLS

The XSG is the primary tool used for development of the design.System Generator is part of the ISE® (Interactive Software Environment) Design Suite. XSG provides Xilinx DSP Blockset such as adders, multipliers, registers, filters and memories for application specific design for Simulink environment which can be readily compiled into a hardware description language (HDL) and subsequently synthesized for specific Xilinx FPGAs.These blocks leverage the Xilinx IP core generators to deliver optimized results for the selected device. [3].

The main advantage of using XSG is that, knowledge of HDL is not required. Thus, one can concentrate on design rather than coding. Moreover, the development time is greatly reduced.

HARDWARE TOOLS

Xilinx Spartan 3E FPGA is the tool used for hardware implementation of final design. Xilinx Spartan 3E is a part of Spartan 3 generation of FPGAs which are specifically designed to meet the needs of high volume, cost-sensitive, electronic applications, such as consumer products. It is embedded with 90nm technology in its architecture and, can be used to address a wide range of applications like embedded processing, digital processing, etc [16]. For initial hardware implementation Xilinx Spartan 3E starter kit development board is used.

IMPLEMENTATION

Software Perspective

A block diagram of the wiener filter algorithm with smoothing is provided in Fig. 3. Input signals consist of 16-bit speech waveforms sampled at 8 kHz. For this specific implementation, a frame size of 16 samples was used. A description of blocks in diagram is given below:

FFT:

For implementation of the DFT, two XSG forward and inverse FFT block are used. These block provides both real and imaginary data outputs. To generate frequency domain magnitude and phase data from the FFT block output, an XSG cordic arctan block is used.

Noise Estimate:

The initial noise magnitude estimation is calculated using an accumulator, and a XSG cmult block. Each incoming frequency binary data word is added to the previously accumulated value for that frequency. To obtain an average, the final sum must be divided by the number of frames used in the calculation. Restricting the number of frames to a power of two,

greatly simplifies the division operation. Therefore, this implementation uses the 64 frames of data to calculate the noise magnitude spectrum estimation.

Wiener Filter:

Next, power spectrum is determined for noise and signal by squaring both using two multiplier blocks. Wiener filter is implemented using an adder block, and a cordic divide block. The output of cordic divide block is multiplied by DFT of noisy signal to obtain output speech DFT. Final output speech DFT is fed back, squared to estimated power, then multiplied by smoothing factor τ and added to original speech spectrum multiplied by $(1-\tau)$ to determine transfer function. Here $\tau=0.85$ is used.

IFFT:

Before a time-domain frame can be generated, the new magnitude and previously retained phase frames must be combined and converted to real and imaginary cartesian coordinates required for input to the IFFT process (the XSG inverse FFT block discussed earlier). This is performed with an XSG cordic sin-cos block and two multipliers.

Hardware Perspective

After successful implementation of design in Simulink, next step is to implement the same on hardware i.e. Xilinx Spartan 3E FPGA. A design may seem to work fine in Simulink environment, but can show errors while generation of bit code for hardware. Two of the most common errors are; used resources exceed available resources, and timing errors. Thus, some modifications are to be made to design to make it suitable for hardware implementation.

In case of resources exceed error, properties of blocks are adjusted so that each block uses minimal resources. For example, number of stages in cordic blocks isreduced, use of 3-multiplier structure in IFFT block for resource optimization instead of CLB logic, and properties some multiplier blocks are set for area optimization instead of speed.To resolve timing errors, latency of XSG blocks are adjusted.

Testing using Co-Simulation

Noise and speech signal are added in Simulink environment and then sent to Xilinx Spartan 3E starter kit. Noise signal used is Gaussian with zero mean and (0.05)^2 variance. After processing, enhanced speech is sent back to Simulink environment where it can be plotted, played, and further analyzed. Speech signal used is in wav format, and enhanced speech signal sent by FPGA is converted to same wav format.

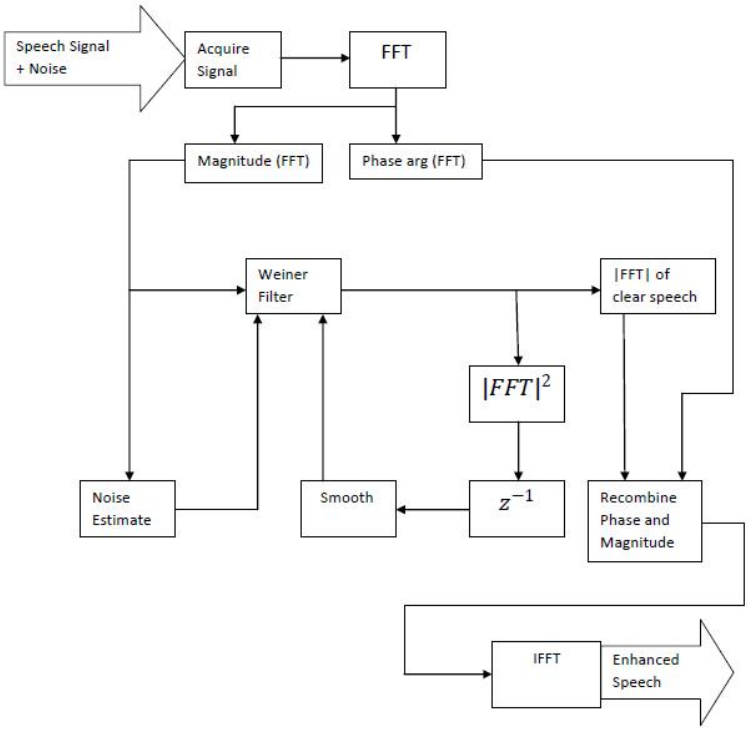


Fig.2. Block Diagram of hardware implementation of Wiener Filter with Smoothing

RESULTS AND DISCUSSION

Speech used to obtain results is "*A quick brown fox jumps over a lazy dog*". Proposed design is compared with spectral subtraction method using time-domain analysis and frequency-domain analysis. Resource usage and power analysis of the design are presented too.

Time Domain Analysis

Fig. 3 shows the time domain analysis of proposed approach and spectral subtraction. Comparison of clean speech (Fig. 3(b)) with Wiener filter (Fig. 3(c)) and spectral subtraction (Fig. 3(d)) enhanced speech, shows that, Wiener filter performed better than spectral subtraction approach.

Increasing number of stages in cordic blocks gives even better results, but need more resources. Thus, a FPGA chip with higher number of resources will be needed.

Frequency Domain Analysis

Fig. 4 and Fig. 5 shows the frequency spectrum and spectrogram analysis, respectively. From frequency-domain analysis it can be seen that higher frequencies are suppressed by both Wiener filter (Fig. 4(c)) and spectral subtraction (Fig. 4(d)), but results obtained from Wiener filter are better.

By the use of pre-emphasis and de-emphasis filters better results can be obtained.

Resource Estimation

From Table 1 it can be seen that design uses about 66% of FPGA resources. Design was optimized to use minimal resources.

Power Analysis

Detailed power analysis was obtained by power analysis tool available in Xilinx ISE 13.1. Results are presented in Table 2. It can be seen that design uses just 0.14 W power.

Time Delay

Time delay was calculated by sending a pulse at 0 sec to the input, and then calculating the time at which output appears. Time delay was calculated to

be 25 ms. Thus, proposed design can be used in real- time.

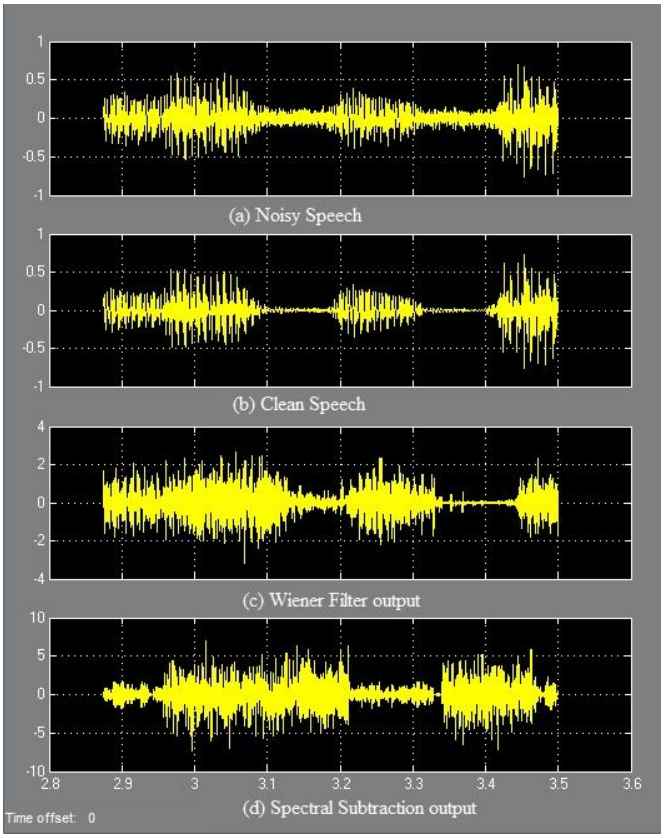
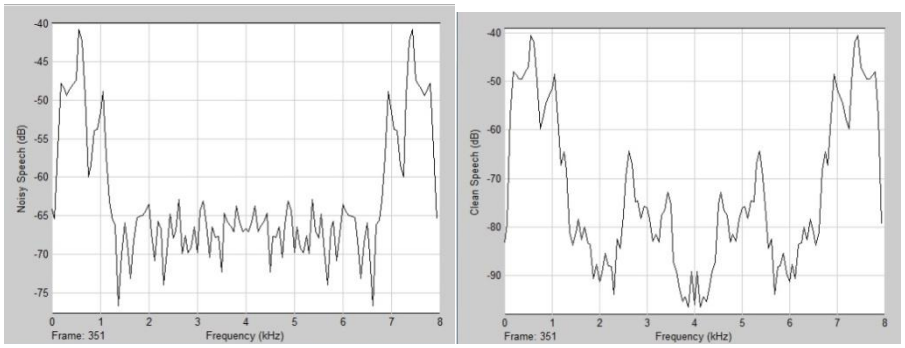
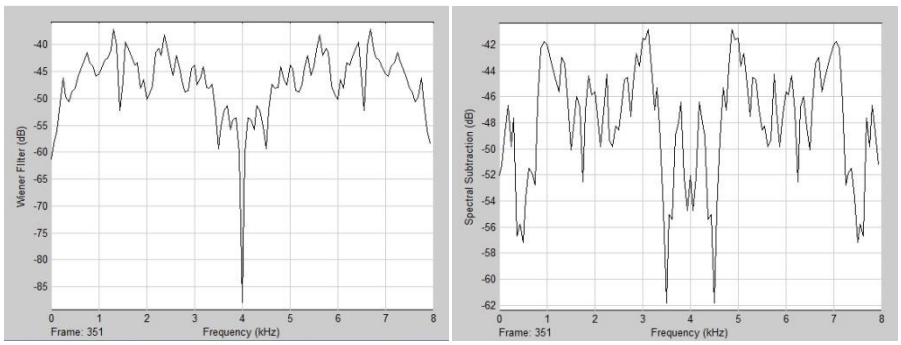


Fig.3. Time-domain analysis (a) Noisy speech (b) Clean speech (c) Wiener filter output (d) Spectral subtraction output

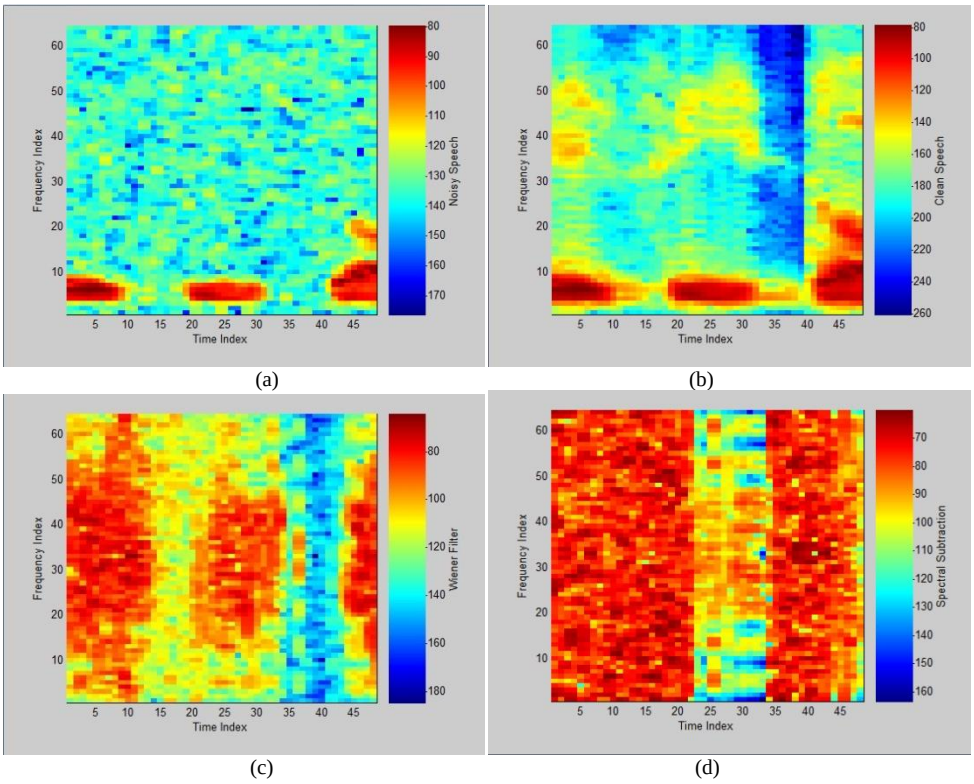


(a) (b)



(c) (d)

Fig.4. Frequency spectrum of (a) Noisy speech (b) Clean speech (c) Wiener filter output (d) Spectral subtraction output



(a) (b)

(c) (d)

Fig.4. Spectrogram of (a) Noisy speech (b) Clean speech (c) Wiener filter output (d) Spectral subtraction output

Table 1

Resource Estimation

Logic Utilization	Used	Available	Utilization
Number of Slice Filp Flops	6186	9312	66%
Number of 4 input LUTs	6368	9312	68%
Number of occupied Slices	3964	4656	85%
Total Number of 4 input LUTs	6721	9312	72%
Number of bonded IOBs	39	232	16%
Number of BUFGMUXs	1	24	4%
Number of MULT 18X 18SIOs	16	20	80%

Table 2

Power Analysis

On-Chip	Power (W)
Clocks	0.010
Logic	0.022
Signals	0.021
MULTs	0.004
IOs	0.000
Leakage	0.082
Total	**0.140**

CONCLUSION

In this paper, Wiener Filteris implemented on Xilinx Spartan 3E FPGA for speech enhancement, which is a low cost FPGA device from Xilinx. Results show that design uses minimal resources and minimum power consumption. Thus, this device can be usedwhere low power consumption is required. Now FPGA chips are available with in-built ADC/DAC, so design can be implemented using a single chip only. Present results are appropriate for machine recognition purposes, but with further improvement it can be used for hearing purposes too. Moreover, design is created using Xilinx System Generator, thus it can be ported to a large variety of Xilinx FPGA devices including high-end FPGA devices like Vertex series.

REFERENCES

1. Whittington, Jim and Deo, Kapeel and Kleinschmidt, Tristan and Mason, Michael W. (2008) *FPGA Implementation of Spectral Subtraction for In-Car Speech Enhancementand Recognition.* In: International Conference on Signal Processing and Communication Systems 2008, 15-17 December 2008, Gold Coast, Australia.
2. Thomas F. Quatieri. Discrete-Time Speech Signal Processing Principles and Practice, Pearson Education.
3. Xilinx System Generator for DSP Reference Guide. Xilinx Inc.

4. System Generator for DSP Performing Hardware-in-the-Loop With the Spartan™-3E Starter Kit. Xilinx Inc.
5. Kiran Kintali and Yongfeng Gu.Model-Based Design with Simulink, HDL Coder, and Xilinx System Generator for DSP. Mathworks Inc.
6. Spartan 3E FPGA User Guide. Xilinx Inc.
7. C.-C. Cheng, W.-H. Liu, C.-H. Yang, and J.-S. Hu, "A robust speech enhancement system for vehicular applications using H*1* adaptive filtering," in *IEEE International Conference on Systems, Man andCybernetics*, vol. 3, 2006, pp. 2541–2546.
8. S. Yu, "Hybrid speech enhancement and speech recognition system for car telematics platform for hands-free control GPS navigator and voice dialer for handphone," ASEAN Virtual Instrument Applications Contest Submission, 2006.
9. D. Halupka, A. Rabi, P. Aarabi, and A. Sheikholeslami, "Low-power dual-microphone speech enhancement using field programmable gate arrays," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3526–3535, 2007.
10. A. E. Nelson "Implementation of image processing algorithms on FPGA hardware." *PhD diss., Vanderbilt University*, 2000.
11. S. Choi, R. Scrofano, V. K. Prasanna, J.W. Jang. "Energy-efficient signal processing using FPGAs." in *Proceedings of the 2003 ACM/SIGDA eleventh international symposium on Field programmable gate arrays*, ACM, pp. 225-234, 2003.
12. S. Mittal, S. Gupta, S. Dasgupta. "System generator: The state-ofart FPGA design tool for dsp applications." in *Third International Innovative Conference On Embedded Systems, Mobile Communication And Computing (ICEMC2)*, 2008.
13. R. Duren, J. Stevenson, M. Thompson. "A comparison of FPGA and DSP development environments and performance for acoustic array processing." in *50th Midwest Symposium on Circuits and Systems MWSCAS, IEEE*, pp. 1177-1180, 2007.
14. Sparsh Mittal, Saket Gupta. "FPGA Implementation of MIMO System using Xilinx System Generator for Efficient Hardware/Software co-design" in Software Engineering : An International Journal (SEIJ), Vol. 3, No. 1, april 2013.
15. Prasit Kumar Bandyopadhyay, Arindam Biswas, Pramit Kumar Bandyopadhyay, Durbadal Mandal, Rajib Kar. "FPGA Based High Frequency Noise Elimination System from Speech Signal using Xilinx System Generator" in International Journal of Scientific Research and Application, ISSN: 2334-6051, Section: F, Volume 1 - 2012, Issue 2, pg. 20-25**.**
16. Spartan-3 generation FPGA user guide. Xilinx Inc.
17. J. S. Lim and A. V. Oppenheim: Enhancement and band width compression of Noisy speech, Proc. of the IEEE, vol. 67, No..12, pp. 1586-1604, Dec. (1979).