ISSN 2047-3338

Implementation of Spectral Subtraction Noise Suppressor Using DSP Processor

Anil Chokkarapu, Sarath C. Uppalapati and Abhiram Chintakuntla

***Abstract*— Surrounding noise and interference will effects the quality of speech during communication. To remove this effect and to improve the quality of speech signal, speech enhancement is one of the most used branches of signal processing. For reduction of noise in speech signals, spectral subtraction can be used and it also requires Voice Activity Detection (VAD) to detect the speech components for a given particular instinct of time. This paper deals with real-time implementation of Spectral Subtraction using Weighted OverLap Add (WOLA) filter bank to suppress noise. From the results, it can be analyzed that the noise is efficiently suppressed using spectral subtraction method and Power spectral density (PSD) of noise suppressed signal obtained from MATLAB and Digital Signal Processor (DSP) kit are studied and compared.**

***Index Terms* — WOLA, DSK6713, Spectral Subtraction and VAD**

I. INTRODUCTION

IN the past decades, research in the field of speech enhancement has focused on the suppression of additive background noise. The presence of background noise in speech significantly reduces the intelligibility of speech. Noise reduction or speech enhancement algorithms are used to suppress such background noise and improve the perceptual quality and intelligibility of speech. Several methods have been developed as a result of research efforts. So we implemented a noise reduction algorithm called Spectral Subtraction (SS) which is a popular method of reducing the effect of background noise efficiently.

Here the method used requires signal to be analyzed in time-frequency domain; in order to carry out implementation we make use of filter banks. Filter banks are used in most of the real time digital signal processing applications. We implemented this method by making use of WOLA filter bank. And in order to estimate spectral contents of noise we use a special technique called VAD. Here filter bank analysis and synthesis and VAD techniques used in this paper are implemented in MATLAB for a speech corrupted and then validated on DSP processor kit TMS320C6713.

In this paper, Section 2 which describes about WOLA filter bank, VAD and SS for noise suppression in noisy speech signal. Section 3 describes implementation and results.

II. PROJECT DESCRIPTION

A. Filter bank

A filter bank provides a natural decomposition of the input signal into different frequency bands. These filter banks can be used to transform the time-domain input signal in to certain number of uniform frequency bands. The transformed filter bank signals are denoted as subband signals since each of them describe a subband of original signal, which can be processed independently so that processing load can be implemented in parallel for every sub band.

The part of the filter bank that transforms a time-domain input signal into a corresponding frequency domain representation is referred to as an Analysis filter bank and the corresponding reconstruction that transforms a frequency signal into a time signal is referred to as a Synthesis filter bank. An analysis-synthesis filter bank is illustrated in Fig. 1.

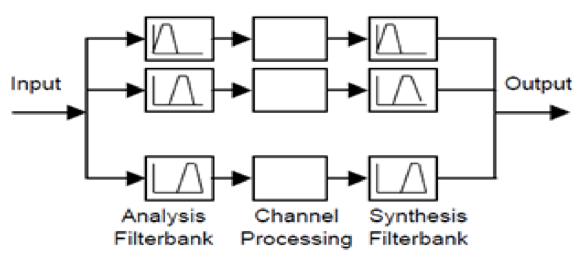

Fig. 1. Analysis-Synthesis filter bank

The WOLA [2] is a highly efficient implementation of an over-sampled generalized DFT filter bank, offering a low-delay, computationally cost effective, perfect/near-perfect reconstruction system. The WOLA structure is based on the block transform interpretation which provides highly flexible time-frequency representation.

WOLA filter bank is defined by four variables namely, L the length of the analysis window, D the decimation rate, K the number of sub bands, and DF the synthesis window decimation rate along with an analysis window function $w[n]$. The choice of WOLA filter bank parameters has effect on aliasing, group delay, frequency resolution and time resolution so appropriate selection of these parameters is important to obtain the best performance. WOLA filter bank has two stages; analysis stage and synthesis stage.

A simplified block diagram of the WOLA analysis stage is shown in Fig. 2. Here, the input signal is shifted *D* samples at a time into the input First In First Out (FIFO) buffer $u[n]$, of length *L* samples. Input FIFO is then windowed with a prototype Finite Impulse Response (FIR) filter (analysis window) of length *L* and stored into a temporary buffer of length *L* samples $t_1[n] = u[n] * w[n]$. The resulting vector is time folded into another temporary vector $t_2[n]$ of *K* samples length. The temporary buffer is circularly shifted by *K/*2 samples in order to produce a zero-phase signal for the Fast Fourier Transform (FFT). This means that the upper half of $t_2[n]$ is swapped place with the lower half of $t_2[n]$. The circularly shifted buffer $t_2[n]$ is then fed into a *K*-sized FFT to compute FFT to subband signals $x_k[n]$.Then, the outputs from the analysis filter bank provide both magnitude and phase information.

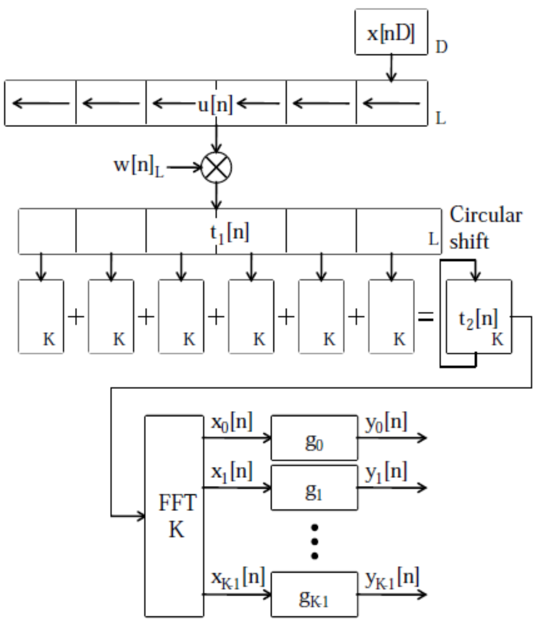

Fig. 2. Analysis stage of WOLA filter bank

To generate a modified time-domain signal, the channel gains are applied to the *K*-sized FFT outputs (channel signals) and an inverse FFT is computed

The synthesis stage of the WOLA filter bank implements the WOLA procedure. The synthesis stage, as shown in Fig. 3, starts by applying a size-*K* IFFT to the processed subband signals $y_k[n]$. The IFFT output is circularly shifted *K/*2 samples, to counter-act the circular shift used in the analysis stage, and the circularly shifted data is stored in a temporary buffer $t_3[n]$, of size *K* samples. The buffer $t_3[n]$ is then stacked, by repetition, in the buffer $t_4[n]$ of length *L/DF*, where *L* is the analysis window length, and *DF* is the synthesis window decimation factor. The buffer $t_4[n]$ is weighted by a synthesis window function $z[n]$ of size *L/DF*, defined as $z[n] = w[nDF]$, i.e. a factor *DF* decimated analysis window function. The weighted data is summed with the data in the output FIFO, $t_5[n]$ of length *L/DF*, and the output FIFO data is over-written with the summation results into $t_5[n] = t_5[n] + z[n] * t_4[n]$. The output FIFO is then shifted left by D samples i.e. FIFO's rear was filled with D zeros and out shifted data is actual output data block *y[nD]*.

B. Voice Activity Detection

For audio processing applications such coding, enhancement, and recognition of speech, whose performance mainly depends upon Voice Activity Detection (VAD), since it is necessary to distinguish speech and noise components in the noisy speech data. The main primary function of voice activity detection is to produce an indication of speech presence in order to facilitate speech processing techniques /methods. The variety and the varying nature of speech and background noise make it challenging in designing. This VAD should be independent of application area and noisy condition i.e. may be varying or stationery. The main primary function of voice activity detection is to produce an indication of speech presence in order to facilitate speech processing techniques /methods. The variety and the varying nature of speech and background noise make it challenging in designing.

Here VAD works on the principle of extracting measured features from the incoming audio signal which is divided into frames of 10-40 ms duration. These extracted features from the audio signal are then compared to a threshold limit usually estimated from the noise only periods of the input signal and a VAD decision is computed. If the feature of the input frame exceed the estimated threshold value, a VAD decision (*VADout* = 1) is computed which declare that speech is present. Otherwise, a VAD decision (*VADout* = 0) is

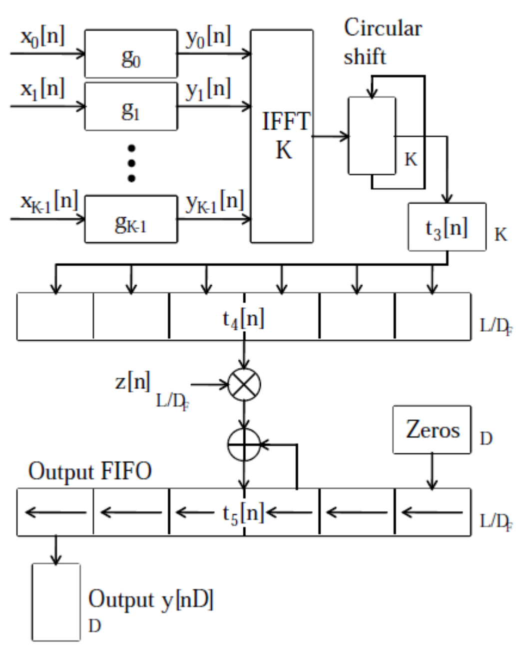

Fig. 3. Synthesis stage of WOLA filter bank

computed which declares the absence of speech in the input frame. The block diagram of a basic VAD is shown in Fig. 4.

Here the features are extracted by computing the subband power spectrum of signals. The subband energy compute is given by:

$$E_{final} = (1 - t_A) * E_{intial} + t_A * {x_k}^2 \quad (1)$$

here x_k subband sample, t_A is time constant which is learning parameter, depends on the subband framing time(T_f) and sampling frequency (f_s) of signal given by:

$$t_A = 1/(f_s * T_f) \quad (2)$$

Thus, by using this VAD we achieved in distinguishing between speech and variety types of acoustic background noise components with low Signal to Noise Ratios (SNRs).

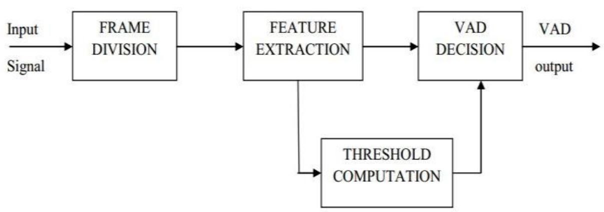

Fig. 4. Block diagram of VAD

C. Spectral Subtraction

Noise reduction or speech enhancement algorithms/methods are used to suppress background acoustic noise and improve the perceptual quality and intelligibility of speech. One of the well-known noise reduction techniques is Spectral Subtraction [3]. This is frequency- dependent method that offers better quality of speech in the resulting enhanced speech with reduced residual noise

Now let us assume that the received subband signals comprise a mixture of clean speech plus background noise is

$$X_k[n] = S_k[n] + V_k[n] \quad (3)$$

here $S_k[n]$ denotes clean subband speech and $V_k[n]$ denote sub band noise.

The basic principle of the spectral subtraction method is to subtract the magnitude spectrum of noise from that of the noisy speech. And this method assumes that spectral properties of noise do not change much just before speech is active until the speech is active. An estimate of the noise signal is measured during silence or non-speech activity in the signal i.e. $X_k[n] = V_k[n]$.

The spectrum of input noisy speech signal is:

$$P_{X;K}[n] = P_{S;K}[n] + P_{V;K}[n] \quad (4)$$

And it is assumed that speech and noise are uncorrelated, i.e., $P_{SV;K}[n] = 0$, so we may write above relation as:

$$P_{S;K}[n] = P_{X;K}[n] - P_{V;K}[n] \quad (5)$$

Hence, the spectral subtraction method continuously subtracts the noise spectrum estimate from the input signal spectrum to acquire an approximate speech spectrum, i.e.,

$$\hat{P}_{S;K}[n] = max\left(\hat{P}_{X;K}[n] - \hat{P}_{V;K}[n]\,, 0\right) \quad (6)$$

the max-function is to ensure a constantly positive spectrum estimate. The enhanced speech signal is

$$Y_K[n] = \hat{P}_{S;K}[n]^{\frac{1}{p}}\, e^{j\angle X_K[n]} \quad (7)$$

In above equation (7), $\angle X_K[n]$ is the phase property of noisy speech and p determines magnitude if p=1 and power spectrum if p=2.

Power spectrum estimation for real time processing applications can be done using recursive averages as

$$\hat{P}_{X;K}[n] = (1 - \lambda_K)\hat{P}_{X;k}[n-1] + \lambda_k |X_K(n)|^p \quad (8)$$

here λ_k learning parameter is defined as $\lambda_k = 1/F_s T_{\lambda;k}$ here F_s is the sampling frequency in Hz and $T_{\lambda;k}$ is the average memory length in seconds. And the background noise spectrum is estimated during non speech activity when VAD=0 i.e. $X_k[n] = V_k[n]$ then spectrum is estimated as:

$$\hat{P}_{v;K}[n] = \begin{cases} (1 - \lambda_K)\hat{P}_{x;k}[n-1] + \lambda_k |X_K(n)|^p & if\ VADout = 0 \\ \hat{P}_{v;k}[n-1] & else \end{cases} \quad (9)$$

D. TMS320C6713 Digital Signal Processor

The TMS320C6713 DSP Starter Kit (DSK) [4] device is based on the high-performance advanced VelociTI very-long-instruction-word (VLIW) architecture developed by Texas Instruments (TI). The VelociTI architecture provides ample performance to decode a variety of existing digital audio formats and the flexibility to add future formats. The kit uses USB communications for true plug-and-play functionality. The C6713 is based on the TMS320C6000 DSP platform designed to needs of high-performing high-precision applications such as pro-audio, medical and diagnostic.

The block diagram of C6713 DSK is shown in Fig. 5**,** it consists of internal program memory 256KB of L2 cache memory shared between the programmer and data spaces, which is structured such a way that it fetches eight instructions for every cycles futures of C6713. The DSK kit package is powerful tool with hardware and software support for real time processing application, it includes 32-bit stereo codec, and board uses sigma-delta technology for analog to digital conversion and also digital to analog conversion. It is connected to 12 MHz system clock, the on board kit includes SDRAM, flash memory and four connections MIC IN, LINE IN, LINE OUT and HEADPHONE OUT.

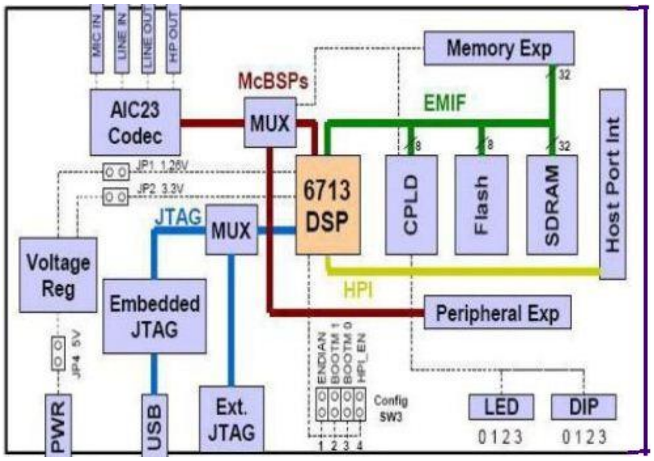

Fig. 5. Block diagram of TMS320C6713 DSP kit

III. IMPLEMENTATION AND RESULTS

Here for implementation a speech signal is degraded with a background noise of restaurant noise which is recorded at sampling frequency of 8 KHz. Then a filter bank is implemented with following parameters chosen are L=128, K=32, D=2, DF=1. Initially filter bank implemented in Matlab and then evaluated it on DSP kit. We tested the implementation of perfect filter bank by analyzing the result of the output in comparison with its input clean speech signal by using Perceptual Evaluation of Speech Quality tool (PESQ) and then evaluated filter bank on DSP kit by exporting WOLA filter coefficients into CC Studio to implement filter bank on the DSP processor (TMS320C6713) then finally verified by observing spectral densities of input and output signals of filter bank.

Calculation of the VAD data is done before the spectral subtraction, the noisy speech signal and VAD output is shown in Fig. 6.

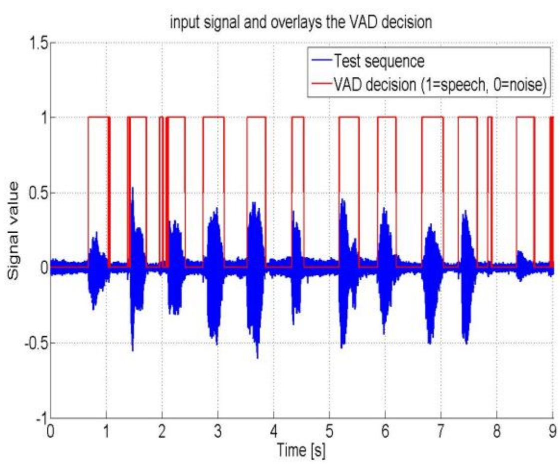

Fig. 6. VAD Decision Data and Input Noisy Signal

From Fig. 6. We can observe clearly that VAD part determines the instance of time where the speech is active. For VAD decision we have chosen the SNR threshold to be equal to 1. And this predefined SNR threshold is used for comparison with signal and noise energy of every new sample and decision is made based on comparison. The VAD decision is denoted as *VADout* and it depends on the energy difference between signal and noise. If *VADout*=1 it indicates the activeness of speech and *VADout*=0 indicates no speech. The noise spectrum is calculated at the instance where speech is inactive and this spectrum can be used in spectral subtraction when speech is active, this process repeats when *VADout*=0.

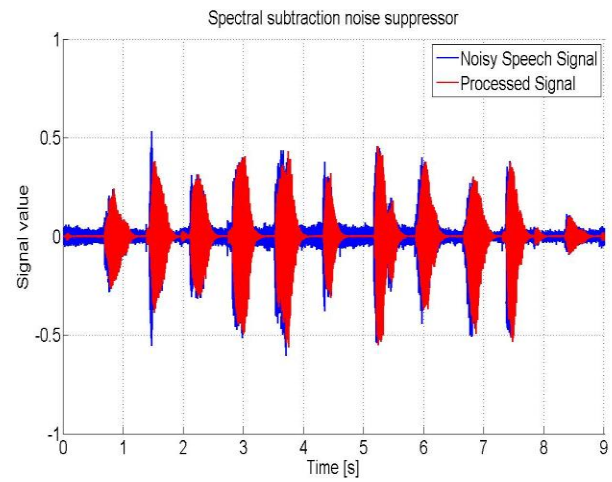

Fig. 7. Comparison of noisy speech signal and processed signal

We perform spectral subtraction depending upon the date obtained from VAD, and whose noisy signal and the processed signals are plotted as shown in fig. 5.

From Fig. 7, it can be inferred that the noise is substantially suppressed. After implementing above noise suppression algorithm in MATLAB, we generated WOLA filter coefficients and exported in to CC Studio to implement this technique on the DSP processor (TMS320C6713). On this processor we efficiently suppressed noise by using spectral subtraction technique with the use of CC Studio software. In order to plot the spectrum of output signal from DSP processor we use 'Wavrecord' command in MATLAB to record the signal and then we calculated its Power Spectral Density (PSD) in MATLAB which is represented in Fig. 8.

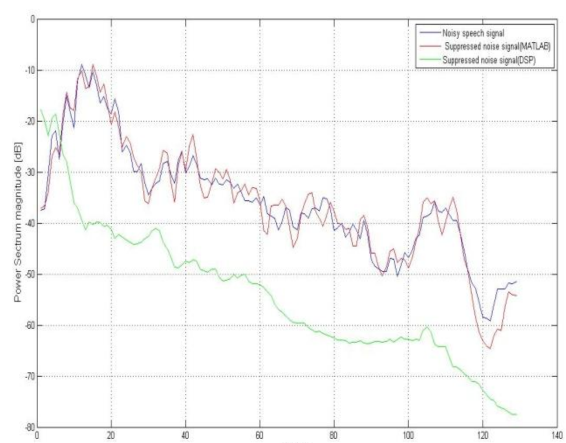

Fig. 8. Power spectrum of noise suppressed signal recorded from DSP and MATLAB

IV. Conclusion

The obtained results show that the unwanted background Additive Noise is suppressed from noisy speech signal by using spectral subtraction algorithm. And the results are identical when we perform this technique on MATLAB and DSP processor in suppressing unwanted noise. In obtaining efficient result we add challenges in designing the VAD technique part in order to obtain speech and noise components separately from noisy speech signal.

References

[1] Benny Sällberg. (2010 Nov.), "Digital Signal Processors ET1304 projects", course material.

[2] R. E. Crochiere. A weighted overlap-add method of short-time fourier analysis/synthesis. *IEEE Trans. Acoust. Speech and Sig. Proc.*, ASSP-28:99–102, February 1980.

[3] S. F. Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Trans. Acoust. Speech and Sig. Proc.*, ASSP-27:113–120, April 1979. S. F. Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Trans. Acoust. Speech and Sig. Proc.*, ASSP-27:113–120, April 1979.

[4] Rulph Chassaing, Donald Reay in Digital Signal Processing and Applications with the TMS320C6713 and TMS320C6416DSK, second addition, John Wiley & Sons, INC., Publication, 2008, pp. 210-254.

Anil Chokkarapu was born in huzarabad, India in 1987. He completed his Master's (M.Sc & M.Tech) programme in Electrical Engineering with emphasis on Signal Processing at Blekinge Institute of Technology, Karlskrona, Sweden. He completed his Bachelor degree in Electronics and Communication Engineering from Jawaharlal Nehru Technical University (JNTU) Hyderabad, India. His areas of research interests are in Audio/Speech Processing, hearing aids Biomedical Signal Processing.

Sarath Chandra Uppalapati was born in 1987 at Hyderabad, India. He did his Master of Science (M.Sc) in Electrical Engineering with emphasis on Signal Processing at Blekinge Institute of Technology, Sweden and Master of Technology (M.Tech) in Signal Processing at JNTU University, India. He completed his Bachelor degree in Electronics and Communication Engineering, from Jawaharlal Nehru Technical University (JNTU), India. He worked as a Trainee Engineer at Metox Resistor Industries from July 2008 till august 2009. His current research interests are Speech Processing, Image Processing and Neural Networks.

Abhiram Chinthakuntla was born in Warangal, India in 1988. He completed his Bachelor degree in Electronics and Communication Engineering from Jawaharlal Technological University (JNTU) Hyderabad. He completed Master's programme in Electrical Engineering, with emphasis on Signal Processing at Blekinge Institute of Technology, Karlskrona, Sweden. His areas of interests are Digital signal processing, image/audio processing, Adaptive systems and Neural Networks.