REAL-TIME IMPLEMENTATION OF ACOUSTIC NOISE SUPPRESSION

P.R. Holmes and T.J. Moulsley *

ABSTRACT

The addition of environmental background noise to speech signals can be a serious problem in speech processing. It can make tasks such as voice recognition substantially more difficult and also significantly reduces the intelligibility obtained with low bit-rate speech coding techniques (ref 1). The technique of spectral subtraction has been shown to be useful in reducing the level of narrowband stationary noise from speech (ref 2). This paper outlines the implementation and performance of an FFT based spectral subtraction noise reduction algorithm using a single TMS 32020 digital signal processing chip.

INTRODUCTION

Several spectral subtraction laws for the enhancement of noisy speech have been suggested (ref 1,2,3) and evaluated. Here, the removal of the noise from the magnitude spectrum of the input is used because (as in ref 4) this gave the best subjective performance.

The following assumptions are made in the development of the magnitude subtraction law and in the other parts of the algorithm. The noise is assumed to be additive and locally stationary to the extent that the magnitude spectrum of the noise signal is the same just prior to a speech sound as during it. Finally, the assumption is made that significant noise suppression is possible by removing the effect of noise from the magnitude spectrum only.

THE NOISE SUPPRESSION ALGORITHM

Assume a windowed sampled noise signal \( n(k) \) has been added to a windowed sampled speech signal \( s(k) \) to give a sum \( x(k) \). The suppression law used here is derived by Boll (ref 1) and gives the following equation for the enhanced speech spectrum

\[
\hat{S}(m) = |X(m)| \exp(j\Theta_m) \left[ 1 - \frac{|N(m)|^2}{|X(m)|^2} \right] = X(m)K_1(m) \quad (1)
\]

where \( \hat{S}(m) \) is the DFT of the enhanced speech signal, \( X(m) \) is the DFT of the noisy input signal, \( \Theta_m \) is the phase of the input signal, \( |N(m)|^2 \) is the noise power spectrum and

\[
K_1(m) = \left[ 1 - \frac{|N(m)|^2}{|X(m)|^2} \right] \quad (2)
\]

\( K_1(m) \) is the suppression coefficient at frequency component \( m \). The noise power spectrum in equation (2) can only be estimated, however, and so \( K_1(m) \) is given a lower bound of zero to ensure that the enhanced speech magnitude spectrum does not become negative.

* Philips Research Laboratories, Crossoak Lane, Redhill, Surrey RH1 5HA.
ESTIMATION OF $|N(m)|^2$

Under the assumptions made in the development of the noise suppression algorithm, the noise power spectrum can be estimated during non-speech activity periods and $|N(m)|^2$ can be replaced by its average value. The whole frequency spectrum is equally divided into 8 frequency bands and in each of these a speech activity decision is made on the basis of the logarithm of the average power in each band. For each band a speech/non-speech threshold is calculated from the power in the current frame plus an exponentially weighted sum from previous frames. If the power exceeds the threshold in any band then a speech activity decision is made for that band, otherwise a non-speech activity decision is made. The exponential weighting factor determines the threshold update rate and tracking ability of the speech/non-speech detector. A time constant of .8 secs gives good results. In each frequency band which is classified as non-speech, the average noise power spectrum in that band is updated using an exponentially weighted sum of previous frames' noise power spectra together with a contribution from the current frame. An exponential weighting factor giving a time constant of .4 secs produces a good estimate of the average noise power spectrum.

RESIDUAL NOISE SUPPRESSION

One of the problems with the noise suppression law given in equation (1) is the residual noise that remains subjectively disturbing due to its musical quality (ref 5). An additional suppression law has been added to reduce the residual noise in regions of the spectrum with low signal to noise ratio. The suppression coefficients for this additional suppression are

$$K_{sp}(m) = 1 - \sqrt{c \frac{|N(m)|^2}{|X_p(m)|^2}}$$

(3)

where $|X_p(m)|^2$ is the peak input power spectrum which is allowed to decay exponentially with a time constant of .8 secs. The constant $c$ is required to achieve suppression since $|N(m)|^2$ is usually quite a bit smaller than $|X_p(m)|^2$.

A reduction in the residual noise can also be achieved by the use of frequency domain smoothing on the estimated noise power spectrum and input power spectrum used in the suppression law given by equation (1). This gives a small subjective improvement in the enhanced speech.

IMPLEMENTATION DETAILS

Fig.1 is a block diagram of the noise suppression algorithm described in this paper.

The noisy analogue speech input is low pass filtered to 3.4 kHz and sampled at 8 kHz. Digital pre-emphasis is used which eases processing the spectrum in finite word length arithmetic. A frame of input samples together with overlapping samples from the previous frame is windowed using the square root of a Tukey window (a flat top with cosine tapering at the edges) prior to taking the Fast Fourier transform. This windowing is repeated just prior to inverse transforming so that the sum of the overlapping output sequences is identical to the original input signal (assuming no spectral modification). The reason for windowing twice is to smooth the effect of fluctuations in suppression coefficients from frame to frame. In this implementation a window length of 256 samples
is used with a frame size of 200 samples. The 256 point real FFT for each data window is calculated using a 128 point complex transform followed by conversion to a 256 point real transform (see ref 6). This achieves a reduction by a factor of about two in memory storage requirements and a decrease in execution time over a 256 point complex transform with complex data set to zero.

HARDWARE

The noise suppression algorithm described has been implemented in real time using a TMS 32020. The algorithm was developed using an IBM PC with a Loughborough Sound Images TMS 32020 development board with analogue i/o. This development system allows the creation of source code on the IBM PC which can then be assembled, linked and downloaded onto the board and then executed in real time.

A frame size of 200 samples with a sampling rate of 8 kHz gives 25 msecs of available processing time per frame. The algorithm is completed in about 22 msecs. The FFT and IFFT with windowing require 14 msecs and the noise suppression requires about 8 msecs. A total of approximately 1.6 K words of data memory are used.

PERFORMANCE

The algorithm described in this paper achieves good noise suppression for speech signals with S/N ratio down to about 0 dB. Quantitatively, the performance is difficult to assess without listening tests such as DRT’s. These have not been conducted here. However, the algorithm does significantly reduce the level of noise although the residual musical noise remains slightly disturbing between speech sounds. This can be reduced by increasing the amount of suppression at low S/N ratios but a careful balance between noise suppression and speech quality must be achieved for the best subjective performance. It is also important that the algorithm does not degrade non-noisy speech signals. When clean speech is processed by the algorithm described here the processed speech is virtually indistinguishable from the unprocessed.

The algorithm is capable of suppressing different types of background noise, for example white noise (see fig.2), coloured noise and tones. The optimum performance for each type of noise can be obtained by adjusting the time constants described earlier. For continuous tones, fast time constants yield better performance in terms of speed of adaption, whereas for white and coloured noise slower time constants give better noise suppression.

CONCLUSIONS

A spectral subtraction noise suppression algorithm has been successfully implemented in real time on a TMS 32020 DSP. The algorithm is simple, modular, variable and may be simplified if required. Although the processing parameters (eg. FFT size, time constants, number of frequency bands) are not critical, they could be varied to optimise performance for particular types of background noise. Speech enhancement techniques based on frequency domain processing are likely to be useful in both speech recognition and speech coding. Implementation requirements are relatively modest particularly if the speech application already incorporates the necessary Fourier transform operation.
REFERENCES


Fig.1 Noise Suppression Algorithm

Fig.2 Enhancement of Noisy Speech