telecommunications &
signal processing
laboratory

Measuring Speech Activity

Peter Kabal

Department of Electrical & Computer Engineering
McGill University
Montreal, Canada

August 1997 (revised August 1999)

2000/09/21

Abstract

This report discusses the algorithm described in ITU-T Recommendation P.56 for measuring the active speech level. Method B in P.56 determines a speech activity factor representing the fraction of time that the signal is considered to be active speech (as opposed to background idle noise) and the corresponding active level for the speech part of the signal. The basic algorithm generates an envelope value at each sample time. The envelope values are compared with a discrete set of thresholds. The (approximate) active speech level is determined by interpolating in the log domain between the threshold values. In this report we assess the effects on the speech active level due to interpolation. Recommendation P.56 allows for sampling rates as low as 600 Hz. Results for subsampled data are compared with those calculated at the full speech sampling rate.

Measuring Speech Activity

Speech activity measurement involves determining the fraction of time that a signal contains active speech and the speech level while speech is active. Knowledge of the speech activity is important in speech signal measurements. For speech data bases, it is important to ensure that undue leading and trailing non-speech be excised and that the speech level be properly scaled based on the peak signal level and the active speech level [1]. For testing speech coders with environmental noise, artificial test signals are created by adding recorded background noise to clean speech segments. The signal-to-noise ratio for such speech-plus-noise signals is determined as the ratio of the active level for the speech to the rms level for the recorded noise [1]. In the speech coding community, considerable research effort is being expended on variable rate coders or discontinuous transmission systems that attempt to economize on average bit rate and/or power consumption by exploiting the fact the speech occurs in talk spurts. The efficacy of such techniques can be compared to speech activity measurements.

Specifications for the measurement of the level of speech signals are given in ITU-T (International Telecommunication Union, Telecommunication Standardization Sector) Recommendation P.56 [2] as Method B. The measurement of the active level of speech takes into account the fact that speech may contain embedded pauses. Experiments have shown that listeners will perceive a pause in the speech if there is a gap of 350–400 ms or larger [3]. If such gaps are due to pauses between phrases or pauses to emphasize words, they are termed grammatical pauses. Grammatical pauses and other long gaps with idle noise do not affect the perceived loudness and are not counted as active speech. The smaller gaps inherent in any utterance are termed structural pauses and are counted as part of the active speech segment.

The output of the speech activity algorithm is a *speech activity factor* representing the fraction of the signal that can be considered to be active speech and the corresponding *active speech level* for the speech part of the signal. An implementation of a Speech Voltmeter using the algorithm in Recommendation P.56 is part of the ITU-T Software Tools Library [4][5] referred to here as ITU-T STL.

The algorithm under discussion presents a active level information for an utterance as a whole.

Other speech level measurements rely on an immediate indication of the speech level and are meant for a real-time indication of level (see the discussion of Method A in [2]). An example is the volume unit (VU) meter often seen on both professional and consumer audio equipment.

1 Envelope Calculation

The speech activity algorithm calculates an "envelope" for the speech signal. This is a double exponential filtering of the magnitude of the speech sample values,

$$\begin{aligned} p_i &= g p_{i-1} + (1-g)|x_i|, \\ q_i &= g q_{i-1} + (1-g)|p_i|. \end{aligned} \tag{1}$$

The envelope q_i is calculated starting with zero initial conditions.[1] The parameter g is determined by the time constant of the averaging and is set to

$$g = e^{-t/T}, \tag{2}$$

where T is the time constant (0.03 seconds) and t is the interval between speech samples. The overall effect is that the envelope is derived by a second-order IIR filtering of the absolute value of the input signal.

At each sample time i, the smoothed envelope value q_i is compared to a set of thresholds, $\{c_j\}$. If the value exceeds threshold c_j, the corresponding activity count a_j is incremented. In addition there is a hangover ($H = 0.2$ seconds, or $I = \lceil H/t \rceil$ samples), such that the count a_j is incremented within the hangover time even if the envelope does not exceed the threshold. The effective hangover time is somewhat larger than H due to the smoothing inherent in the filtering of Eq. (1). The activity count is calculated as follows.

```
if $q_i \geq c_j$ then
    increment $a_j$ and set $h_j = 0$
else if $h_j < I$ then
    increment $a_j$ and increment $h_j$
```

An alternate mechanism to implement the hangover is to create a modified smoothed envelope,

$$\tilde{q}_i = \max(q_i, q_{i-1}, \ldots, q_{i-I-1}). \tag{3}$$

With this formulation, activity count a_j is incremented whenever $\tilde{q}_i$ exceeds c_j.

[1] Note that p_i will be non-negative for positive g and an initial non-negative value. As such, the absolute value in the second part of Eq. (1) are not necessary.

There are three types of envelope values: the instantaneous envelope $|x_i|$, the smoothed envelope q_i, and the modified smoothed envelope $\tilde{q}_i$ which includes the effect of hangover. Smoothing of the instantaneous envelope has been deemed to be useful to avoid triggering the hangover by brief noise pulses [3] and the hangover allows for the inclusion of structural pauses as active speech.

2 The Envelope Activity Function

The activity count function for a speech segment with N_s samples can be found as follows. Record the envelope value as each sample arrives. Use these envelope values as the threshold values c_j. For convenience, order the c_j in increasing order. For each envelope level value, accumulate the activity count a_j according to the procedure described above. A fractional activity count function can be formed as a piecewise constant function from the activity counts,

$$a(l) = \begin{cases} 1, & \text{for } l < \min(\{c_j\}) \\ a_j/N_s, & \text{for } c_{j-1} \geq l < c_j \\ 0, & \text{for } l \geq \max(\{c_j\}). \end{cases} \tag{4}$$

The activity count function $a(l)$ is a piecewise-constant non-decreasing function of the level l.[2] In general there are N_s different envelope values, and thus such a procedure to determine the activity counts becomes impractical for a large number of samples. It does however serve as the benchmark with which to compare later results.

3 Active Level

In Recommendation P.56, the level values are calculated as relative dB values. For level l (sample units), the relative dB value is

$$C(l) = 10 \log_{10}(l^2 G_v^2) - R. \tag{5}$$

The value G_v is a scale factor relating volts to sample units. For instance if a full scale value of 32 768 sample units corresponds to a 5 volt input signal, then $G_v = 5/32768$. R is an offset to give the value of $A(c_j)$ relative to a specific reference value r in volts,

$$R = 20 \log_{10} r. \tag{6}$$

[2] The cumulative distribution function (cdf) for the modified smoothed envelope values is $F(l) = 1 - a(l)$.

If r is chosen to be the maximum representable speech value (5 V in the example), $C(l)$ is in units of dB relative to overload (dBov).

The decibel value for the active speech level using level l to distinguish between speech and non-speech is given by

$$A(l) = 10\log_{10}(\frac{\sigma_x^2}{a(l)}G_v^2) - R, \tag{7}$$

where σ_x^2 is the mean-square value given by

$$\sigma_x^2 = \frac{1}{N_s}\sum_{i}^{N_s} x_i^2. \tag{8}$$

The active level is determined by the solution to the parametric equation

$$A(l_o) - C(l_o) = M, \tag{9}$$

where

$$M = 10\log_{10} m. \tag{10}$$

The value M determines the level (in dB) relative to the active level, below which the signal is considered to be idle noise rather than speech. In Recommendation P.56, M is 15.9 dB, corresponding to $m = 38.9$. The solution to this equation gives a value l_o from which the active level of the speech ($A(l_o)$) can be found, and from which the fractional activity factor $a(l_o)$ can be determined.

A threshold level of 15 dB below the rms level of the signal can be used to separate active speech from noise [3]. In Recommendation P.56, the threshold difference is 15.9 dB, with the extra 0.9 dB compensating for the difference between the mean-absolute value and the root mean-square value for a sinusoid ($2/\pi$ versus $1/\sqrt{2}$).

3.1 Interpretation of the active level

The active level is determined from the value of l_o which satisfies $A(l_o) - C(l_o) = M$. This condition is equivalent to the following relationship,

$$ml_o^2 = \frac{\sigma_x^2}{a(l_o)}. \tag{11}$$

The value l_o is the envelope value above which the signal is considered to be speech and below which the signal is considered to be idle noise. This square of this value is a factor m below the mean-square value of the speech part of the signal. The mean-square value of the speech part of the signal is σ_x^2 divided by the fraction of time that the modified smoothed envelope is at or above the level l_o.

The active portion of the signal correspond to $\tilde{q}_i > l_o$. Note that the active speech level is chosen retrospectively; the activity count function for the entire signal must be available before l_o can be determined.

This interpretation of the active speech level depends on the premise that idle noise does not contribute significantly to the mean-square value of the signal. As such, this speech activity measurement is not appropriate for speech contaminated with a significant amount of background noise.

4 Examples

Consider two speech files (16 bit data, sampling rate 8000 Hz). The first file has a male speaker and has a maximum absolute value of 13 550. The second file has a female speaker with a maximum absolute value of 12 132. The envelope values (q_i values) for these signals are strictly positive[3] with maximum values of 2541 and 3278. For both speech files, the mean value is small (less than 20).

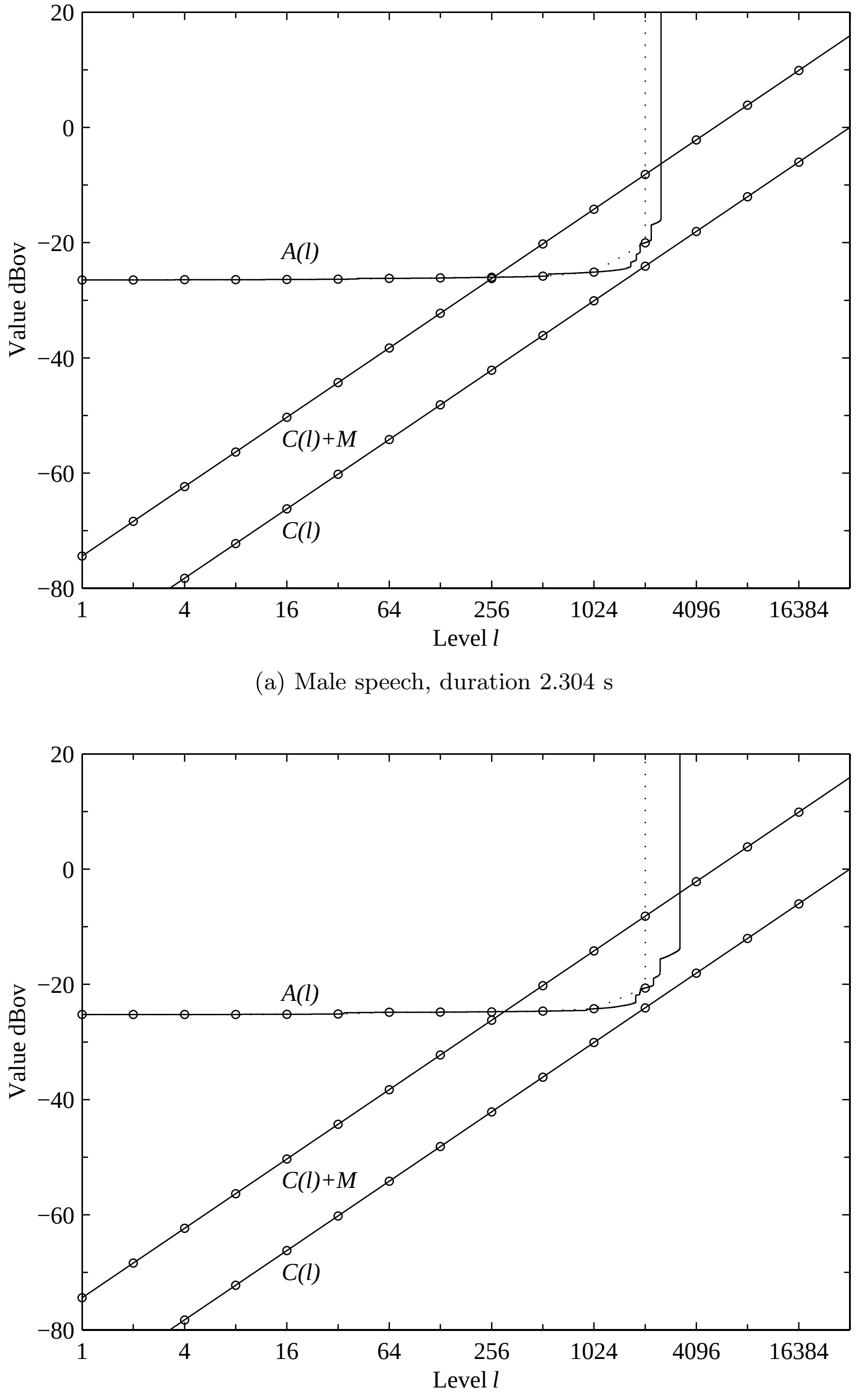
Fig. 1 shows a plot of $A(l)$ and $C(l)$ for the speech files. (The dotted lines joining the circles are discussed later.) The functional form of $A(l)$ is such that as $a(l)$ decreases with increasing l, $A(l)$ increases from its initial value for $l = 0$. Beyond the maximum envelope value, $A(l)$ becomes infinite. For the first signal, $\sigma_x = 1553.0$. The intersection of the curves occurs for $l_o = 263$. The active speech level is a factor $\sqrt{m}$ larger than this value: 1642. The speech activity factor is 89%, i.e., $a(263) = 0.89$. For the second signal, $\sigma_x = 1791.0$. The intersection of the curves occurs for $l_o = 305$. The active speech level is a factor is 1900 and the speech activity factor is 89%.

The envelope is calculated based on the absolute value of the input signal, while the energy of the signal is based on the squared value of the input signal. This leads to somewhat of an incompatibility when envelope values are compared to the signal energy. For the sample speech files, the average envelope values (instantaneous and smoothed) are some 5–6 dB below the rms value. The average modified smoothed envelope value for the sample files is only a fraction of a dB below the rms value.

5 Special Cases

Simple scaling of the input signal causes the $A(l)$ curve to move diagonally, and merely scales the cross-over point by the same scaling factor. Decreasing M decreases the activity factor and

[3] Zero-valued envelope values only occur if the signal contains zeros at the beginning of the segment being measured.



(a) Male speech, duration 2.304 s

(b) Female speech, duration 2.848 s

Fig. 1 Plot of $A(l)$ and $C(l)$ for two speech files. The dotted line shows the active level function interpolated between tabulated values (circles).

eventually results in a situation in which the curve for $A(l)$ and the curve for $C(l)+M$ no longer intersect. The dependence of the activity factor on M can be used to argue for an appropriate interpretation of special cases.

5.1 No intersection

One can create (non-speech) input signals for which there is no intersection of the curve for $A(l)$ and the curve for $C(l)+M$ for the standard value of M (15.9 dB). Such a case will occur for a signal with a large rms value but small envelope values. The rms value depends on x_i^2 while the envelope depends on $|x_i|$. Thus a few isolated large values in a signal will increase the rms value much more than they will increase the envelope.[4] When no intersection occurs for the nominal value of M, consider choosing a larger value for M, one for which there is an intersection of the curves. As the value of M is decreased toward the nominal value, the activity factor corresponding to the crossover point decreases and then becomes undefined as the curves no longer intersect. This argument suggests that the speech activity factor should be considered to be zero for the case of no intersection of the curves.

5.2 Multiple intersections

For the speech example in Fig. 1, a second crossover occurs as $A(l)$ shoots toward infinity at the largest envelope value. One can create signals for which the second crossover occurs below the largest envelope value.[5] If we are to attribute an active level to such a signal, it is the first crossover which gives the activity factor, an activity factor which decreases with decreasing M.

6 Interpolation of Tabulated Counts

Calculating an activity count function as described in Section 2 is overly computationally intensive even for relatively short speech segments. Instead, the algorithm specified in Recommendation P.56 uses samples of the activity count function and interpolates between those values. Fixed values for the threshold levels (the c_j) are used to determine corresponding activity counts. For ITU-T Recommendation P.56, the tabulated values c_j form a geometric progression,

$$c_j = b^j \qquad \text{for } j = 0, 1, \ldots, N_l. \tag{12}$$

[4] A signal (8000 Hz sampling rate) consisting of a sample with value 16 000 followed by zero-valued samples will result in an $A(l)$ curve which does not intersect the curve for $C(l)+M$ for the nominal value of M.

[5] A signal (8000 Hz sampling rate) with 1000 samples with value 4000, followed by 800 zeros, followed by 420 samples with value 8000 will give a situation in which there are two intersections.

The threshold values should not increase by a factor greater than 2:1. The ITU-T STL implementation uses 15 levels with $b = 2$, with levels $1, 2, \ldots, 16384$. These levels are appropriate for 16-bit speech data. For each c_j we get a corresponding log value $C(c_j)$ from Eq. (5), and an activity count a_j and the corresponding log-value $A(c_j)$ from Eq. (7).

Recommendation P.56 mandates that the interpolation between values of $A(c_j)$ be linear on a log-log scale. For $c_j \leq l \leq c_{j+1}$, the interpolated value $A(l)$ is

$$A(l) = A(c_j) + \alpha(A(c_{j+1}) - A(c_j)). \tag{13}$$

where

$$\alpha = \frac{C(l) - C(c_j)}{C(c_{j+1}) - C(c_j)}. \tag{14}$$

If we know that the solution lies between $c_{\hat{\jmath}}$ and $c_{\hat{\jmath}+1}$, using Eq. (9), the cross-over point corresponding to l_o is

$$\alpha = \frac{M - (A(c_{\hat{\jmath}}) - C(c_{\hat{\jmath}}))}{(A(c_{\hat{\jmath}+1}) - C(c_{\hat{\jmath}+1})) - (A(c_{\hat{\jmath}}) - C(c_{\hat{\jmath}}))}. \tag{15}$$

Consider the same speech files used in the previous examples. To calculate the activity counts, thresholds (c_j) of $1, 2, \ldots, 16\,384$ are used. The dotted lines in Fig. 1 are the linear interpolation between the tabulated values (circles). We note that linear interpolation follows the activity count function well except at large envelope values. In the region in which the intersection of the $A(l)$ curve and the $C(l)+M$ curve occurs, the approximation is very good. One can surmise that since for speech data, the crossover point occurs in a region in which the curve for $A(l)$ is relatively flat, the interpolation will give a crossover point which agrees to a high accuracy to the actual solution point.

6.1 Special cases

Consider the case of all of the activity counts being zero. The first sub-case is an all-zero signal—the speech activity factor is zero. The second sub-case is a non-zero signal ($\sigma_x \neq 0$). This sub-case corresponds to a signal with too small an amplitude to be measured using the given set of thresholds.

Since only 15 tabulated values are used, attention must be paid to scaling of the signal relative to the threshold values. Consider the case of non-zero activity counts. If the point corresponding to the first value (smallest level) is such that $A(c_0)$ lies below $C(c_0) + M$, the crossover occurs for a value less than c_0. Note that the fractional activity factor for $l = 0$ is unity. Interpolating on a log-log scale between $l = 0$ and $l = c_0$, to find the crossover is equivalent to choosing the crossover to occur for (see Eq. (11))

$$l_o = \sqrt{\frac{\sigma_x^2}{m\,a(c_0)}}. \tag{16}$$

As discussed earlier, a signal for which there is no crossover, should be considered to have a zero speech activity factor.

7 Sampling Rate

ITU-T Recommendation P.56 allows for sampling rates as low as 600 Hz. For instance for a signal sampled at 8000 Hz, subsampling by an integer factor of up to 13 may be possible to reduce the computational complexity. In this section we explore the impact on accuracy when the sampling rate is reduced.

There are several approaches to reducing the sampling rate. Consider bandlimiting the signal and then subsampling by an integer factor. However, since signal components filtered out before resampling will not contribute to the detection of speech, this approach is not to be recommended. On the other hand, subsampling the signal without prefiltering gives rise to aliasing, but the signal and the aliased components will still contribute to the measured envelope values. A third approach is to calculate the envelope values based on the full sampling rate and then subsample the envelope values.

Tests were conducted on the sample speech files using subsampling by a factor of 10 (giving a sampling rate of 800 Hz). Both direct subsampling of the input signal and subsampling of the envelope values give results that agree closely with the full rate results. These results indicate that simple subsampling of the input signal gives accurate results while substantially reducing the computational load.

More detailed tests were carried out on a number of speech files. The results are shown in Table 1. The left hand column lists the input file. The first two files are the same ones used to generate the plots shown earlier. Three different implementations were used: (1) Operations carried out on the full rate signal (designated as f_s). (2) Operations carried out on the subsampled signal (chosen to give a rate of 1 kHz). (3) The results from the speech activity program in the ITU-T STL package.

The table shows that the active levels calculated using subsampled data are within ±0.02 dB of those calculated using the full sampling rate. The values in the table with subsampling show that the results are reasonably accurate, often more accurate than the ITU-T STL calculations. Subsampling is to be recommended as significantly reducing the computational load, particularly for speech sampled at 16 or 48 kHz.

The ITU-T STL calculates the activity counts at the full sampling rate as does the scheme

Table 1 Comparison of activity factors using subsampling

File	Active Level			Activity Factor		
	f_s	1 kHz	STL	f_s	1 kHz	STL
male (2.304 s) $f_s = 8$ kHz	1642.1	1641.8	1640.8	89.45%	89.48%	89.61%
female (2.848 s) $f_s = 8$ kHz	1901.1	1900.1	1901.1	88.75%	88.85%	88.75%
male (2.434 s) $f_s = 16$ kHz	1353.0	1353.7	1353.0	95.52%	95.42%	95.57%
female (2.538 s) $f_s = 16$ kHz	1791.6	1791.9	1800.3	94.53%	94.50%	93.99%
male (1.728 s) $f_s = 48$ kHz	1274.6	1274.2	1275.3	92.15%	92.20%	92.05%
female (2.002 s) $f_s = 48$ kHz	827.7	826.2	826.6	91.76%	92.09%	92.10%

designated as f_s, with the minor difference that the ITU-T STL algorithm operates on blocks of (default) size 256 samples, ignoring any partial blocks at the end of the file. However, a more substantive difference in results is due to the procedure used to determine the crossover point. In the ITU-T STL, a binary search is used between tabulated values, stopping when the results are within 0.5 dB. However, the ITU-T STL version has a faulty implementation which does not properly halve the search interval at each step of the binary search, potentially resulting in errors which are larger than 0.5 dB.

8 Implementation

Sample implementations in MATLAB [6] for the calculation of the activity counts and to find the activity factor are shown below.

8.1 Activity counts

The routine to accumulate the activity counts takes as input a vector of signal values and set of (ordered) threshold values, and returns a set of activity counts. The requirement that the threshold values be ordered allows for the use of the `break` statement in the innermost loop to reduce the average execution time. Note that this implementation in MATLAB is not to be recommended for general use, since it executes very slowly compared to an implementation in a compiled language, say, C. This implementation includes simple subsampling of the input signal.

```
function an = EnvCount (fs, x, c, Nsub)
% Accumulate envelope activity counts
%
% output: an: vector of fractional activity counts
% input:  fs: sampling rate (Hz)
%         x: input signal
%         c: threshold values (increasing values)
%         Nsub: subsampling ratio

T = 0.03;
H = 0.2;
t = Nsub/fs;
g = exp (-t/T);
I = ceil (H/t);

Nx = length (x);
Nl = length (c);
a = zeros (1, Nl);
h = I * ones (1, Nl);

p = 0;
q = 0;
for i = 1:Nsub:Nx
  p = g * p + (1-g) * abs(x(i));
  q = g * q + (1-g) * p;
  for j = 1:Nl
    if (q >= c(j))
      a(j) = a(j) + 1;
      h(j) = 0;
    elseif (h(j) < I)
      a(j) = a(j) + 1;
      h(j) = h(j) + 1;
    else
      break;
    end
  end
end

Nxs = ceil (Nx/Nsub);
an = a / Nxs;
```

8.2 Crossover point

The crossover point is found in the following MATLAB routine. Note that the calculations are done with natural logarithms, since any scale factors associated with the choice of base for the logarithm cancel in the calculation. The function returns the active speech level $\sqrt{m}l_o$. The active speech level can be converted to appropriate dB units. Note however, that the active speech level is returned as zero for some special cases. The fractional speech activity factor is found as (see Eq. (11))

$$a(l_o) = \frac{\sigma_x^2}{ml_o^2}. \tag{17}$$

```
function ActLev = ActiveLevel (an, c, Ex, MdB)
% Find the active speech level
```

```
%
% output: ActLev: active speech level
% input:  an: fractional activity values (non-increasing values)
%         c:  threshold levels (increasing values)
%         Ex: mean squared value of the signal
%         MdB: offset in dB

% Default return value
ActLev = 0;

Mln = MdB * log(10) / 10;       % log (m), where m = 10^(M/10)
Nl = length (an);

% Find the solution to Aln - Cln = Mln
for j = 1:Nl
  if (an(j) == 0)
    break;
  end
  Alnj = log (Ex / an(j));
  Clnj = 2 * log (c(j));        % log (c(j)^2)
  dACj = Alnj - Clnj;
  if (j == 1)
    if (dACj <= Mln)
      ActLev = exp (Alnj / 2);
      break;
    end
  elseif (dACj <= Mln)
    alpha = (Mln - dACjp) / (dACj - dACjp);
    Alno = Alnjp + alpha * (Alnj - Alnjp);
    ActLev = exp (Alno / 2);
    break;
  end
  Alnjp = Alnj;
  dACjp = dACj;
end
```

9 Summary

This report has considered an algorithm for determining the active speech level. By example, it was shown that for properly scaled signals, interpolation of the tabulated values results in essentially the same results as a full (impractical) evaluation of the envelope activity level function. In addition, subsampling the signal to give an equivalent sampling rate of 1000 Hz gives reliable activity factor results, comparable to those obtained from the full rate speech signal while reducing the computational requirements substantially.

An implementation of the speech activity measurements can be found in the `CompAudio` program which is part of the `AFsp` package of audio file routines.[6]

[6] The `AFsp` routines are available over the Internet from `http://www.TSP.ECE.McGill.CA`.

References

[1] ITU-T, Geneva, *Recommendation P.830, Subjective Performance Assessment of Telephone-Band and Wideband Codecs*, Feb. 1996.

[2] ITU-T, Geneva, *Recommendation P.56, Objective Measurement of Active Speech Level*, Mar. 1993.

[3] R. W. Berry, “Speech-volume measurements on telephone circuits,” *Proc. IEE*, vol. 118, pp. 335–338, Feb. 1971.

[4] ITU-T, Geneva, *Recommendation G.191, Software Tools for Speech and Audio Coding Standards*, Mar. 1993.

[5] ITU-T Users’ Group on Software Tools, Geneva, *ITU-T Software Tool Library Manual*, May 1996.

[6] The MathWorks, Inc., Natick, MA, *Matlab Language Reference Manual, Version 5*, Dec. 1996.