Copyright © 1978 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 57, No. 5, May–June 1978
Printed in U.S.A.

Idle Channel Noise Suppression by Relaxation of Binary ADM-Encoded Speech

By S. V. AHAMED

(Manuscript received November 3, 1977)

Techniques of identifying the location of silence periods in the binary data of prerecorded telephone messages from Adaptive Delta Modulation (ADM) encoders are discussed. Two algorithms for detecting and replacing such periods by absolute silence are investigated. In the first method one or two words (each 16 bits long) spanning one or two msecs, respectively, at a 16 kHz ADM-sampling frequency, are searched. They are either replaced or not in their entirety by a perfect silence sequence based upon a computed decision. In the second method, blocks of any prespecified number (1 to 512) of words spanning 1 to 512 msec at 16 kHz are searched to detect the authenticity of a silence period within a smaller block (typically between 1 to 31 words) spanning 1 to 31 msec embedded within the larger block. The first method has yielded unsatisfactory results, and the second method with larger window scan of 80–200 msecs and a smaller window duration of 1 to 5 msec has yielded nearly perfect results by eliminating all of the idle channel noise without degrading the quality of the message. This technique of locating the silence periods in ADM encoded speech is compared with that for ADPCM encoded speech. Also the optimized parameters (for making the computed decision) are shown to be satisfactory for preventing false silence clues for genuine speech data over the wide variations of frequencies and amplitudes. Further, these parameters may be converted for different clock rates, and some are traced back to the specifics of the encoder and the others to the character of speech.

I. INTRODUCTION

The source of idle channel noise in most ADM (Adaptive Delta Modulation) speech encoders is the nonuniformity of the data stream generated by the encoder during silence periods. From a relaxation* study

* In the context of this paper, relaxation implies changing the binary data to a state such that any further change will not yield an improvement in the quality of the speech or of silence.

on the computer it becomes obvious that if the decoder is forced into a repetitive input bit pattern synchronized with the main ADM clock, the decoder idle channel noise can be suppressed. The decoder, responding only to the incoming data stream, cannot generate any noise on its own, and a perfect encoder, to assure silence, would provide a repetitive pattern to the decoder. In the absence of such an encoder, it is possible to determine the silence periods during speech by a computed decision and force the bit pattern to be repetitive during such periods.

The silence periods are particularly conspicuous in exposing the imperfection of the encoder. Being devoid of any meaningful information, they can become annoying if the encoder does not produce a bit pattern which forces a complete silence for the decoder. For instance, the most commonly used bit pattern is 0101. . ., even though other patterns such as 001100. . ., 01001101. . ., etc., all yield a type of semisilence or humming silence with intertwined frequencies. The nature of decoder silence is also influenced by the companding algorithm and the output filter characteristics. In this application we have obtained the best silence by a sequence of 0101. . . which offers two distinct advantages:

(*i*) The compandor having been rendered inoperative by this sequence, the step size decays to its minimum value.

(*ii*) The frequency generated by this sequence, being half the clock rate, is beyond the bandwidth of the audio filter.

II. SILENCE CLUES

2.1 Cluster and transitions clues for ADM data

When the only available data is the encoded binary stream from an ADM encoder, three clues to judge the authenticity of the silence period may be used: (*i*) the absence of at least one cluster of ones whose width exceeds that of a prechosen minimum threshold cluster, (*ii*) the absence of at least one cluster of zeros whose width exceeds that of a prechosen minimum threshold cluster, and (*iii*) the excess of transitions between zeros and ones and vice versa over a prechosen number of transitions during a prechosen interval of time. The first two clues together have been termed "cluster clues," and the last one is termed "transitions clue." There are thus five variables necessary to implement the scheme: (*a*) the number of ones in the minimum threshold cluster of ones, (*b*) the number of zeros in the minimum threshold clusters of zeros, (*c*) the number of transitions in the duration to seek the clues, (*d*) the duration of time to seek the clues and finally, (*e*) the duration for substituting the ideal silence sequence 0101. . . instead of the imperfect silence data from the encoder.

2.2 Code word energy clue for ADPCM data

In Ref. 1, the concept of "code word energy" which is defined as an integrated energy for 16 msec (8 msec forward, 8 msec backward) around a preselected code word of ADPCM data is used. The concept has been used to locate the beginning and end of utterances. Individual discrete code words which convey one of the sixteen levels of amplitude information of the speech wave are scanned for amplitude deviation from a mean value. If the code word energy consistently exceeds that obtained during silence periods at the ADPCM encoder for 50 msec, then the beginning of the utterance is traced back to the instant at which the code word energy first started to exceed the threshold. Conversely, if the code word energy falls below that recorded during speech consistently for 160 msec, the end of the utterance is traced back to the instant at which the code energy first receded from the threshold.

2.3 Differences between clues for ADPCM and ADM data

Code word energy clue is well suited for ADPCM data where amplitude information is discretely coded in each word. For ADM data, only the sign information is conveyed to the decoder and there is no distinguishable boundary between any neighboring bits. For this reason, the concept of code word energy is inapplicable for ADM-encoded data. The cluster clues contain the information that not once during the search interval did the encoder experience an unidirectional change in amplitude lasting for a minimum number of clock cycles. The transitions clue contains the information that the frequency of change of direction of the output from the decoder exceeds what one would expect it to be during low level speech signals, and very close to what one would expect during the silence periods. The cluster clue is based on the fact that speech contains the dominant portion of energy in lower formant frequencies and the transition clue is based on the fact that speech energy concentration tapers off at higher formant frequencies.

Further, the computed decision for ADPCM data is a process of only looking forward in time to determine the increase of energy over the silence threshold energy (for locating the beginning of an utterance) or the decrease of energy below that of the speech threshold energy (for locating the end of the utterance). In ADM data we have found it neessary to look forward and backward for clues to make a computed decision regarding the authenticity of silence during the intermediate interval. As shown in the following sections, if the five variables [(a) through (e) in Section 2.1] on which the concept is implemented, are carefully chosen, the cluster and transitions clues work very dependably.

III. IMPLEMENTATIONAL DETAILS

Implementing the concept has been attempted by two distinct techniques: (*i*) word relaxation schemes and (*ii*) block relaxation schemes. In word relaxation schemes the duration for seeking the cluster and transitions clues [i.e., (*d*) in Section 2.1] is made the same as the duration for replacing the ideal silence sequence 0101 . . . instead of the imperfect silence data [i.e., (*e*) in Section 2.1]. In block relaxation schemes the duration for replacement has been made 2.3 to 13 percent of the duration for searching for the clues to judge the authenticity of the silence.

3.1 Word relaxation schemes

In this method the durations of search and replacement have both been made typically 1 or 2 msec. The width of the minimum threshold cluster for ones (hereafter referred to as "ones cluster threshold") and the width of the minimum threshold cluster for zeros (hereafter referred to as "zeros cluster threshold") are both typically set at two. The threshold for the transitions (hereafter referred to as "transitions threshold") is typically set at ten (16 kHz clock rate) for a one msec search and replacement window. The concept has been programmed in assembly language on a minicomputer in a conversational mode of input and output. Large blocks of data have been processed. The results have been analyzed both by subjective tests and oscillographic studies. Whenever the computer does replace the imperfect silence by a perfect silence the replaced data generates a limit cycle with respect to the input frequency and it appears as a stationary pattern on an oscilloscope. The scheme, even though successful about 90 percent of the time, fails to satisfy a perceptive listener during the silence period. When the three threshold parameters are made (3,3,8) respectively,* the quality of the silence becomes perfect but the speech tends to become slightly choppy and a happy compromise of the three threshold parameters has not proved possible.

Similar results have been encountered with two msec duration for seeking the clues and replacing the silence bit-pattern. A slight improvement gained in the quality of silence with perfect speech still does not satisfy a perceptive listener, and for this reason the block relaxation procedure has been developed.

3.2 Block relaxation schemes

The limited performance of the word relaxation scheme is due to a very narrow slice of time used to determine whether a word contains silence

* These parameters are always written in the same order: ONES, ZEROS, TRANSITIONS.

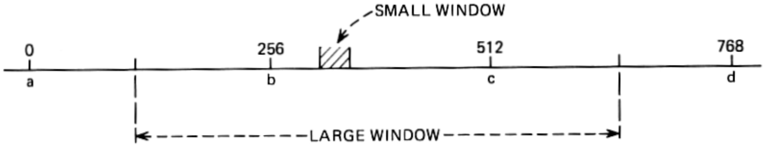

Fig. 1—Data management during the detection and replacement of silence periods. *a* to *d*: Memory buffer refilled from the disk after processing the middle 256 words such that *b* is reread at *a*, *c* at *b*, etc. *b* to *c*: Limits of excursion of the small window and storage of processed data.

or genuine data. The block relaxation scheme expands the search for clues over a much larger interval (large window) to compute the decision for a much shorter interval (small window, see Fig. 1). Typical durations for the large and small windows are from 39 to 131 msec and from 1 to 5 msecs, respectively. The clusters and transitions clues are verified by the same method discussed in Section 2.1. The large window is defined during program execution by two parameters: (*i*) the duration for which the search for the clues should start prior to the beginning of the small window and (*ii*) the duration for which the search for the clues should continue after the end of the small window. Parameters (*i*) and (*ii*) have been made distinct because the choppiness of plosives, fricatives, siblants and nasals at the ends of the utterances can be eliminated by making (*i*) longer than (*ii*) while simultaneously eliminating the slightest crackle before an utterance. The effects of changing the durations of the large window, the small window and the threshold parameters are tabulated in Table I.

IV. DISCUSSION OF RESULTS

Dependable determination of the silence periods and complete suppression of idle channel noise is possible with the right value of the threshold parameters and window sizes. On one hand if the threshold parameters are (1,1,16) for ADM data at 16 kHz with a large window and small window of 1 msec each, then the block relaxation scheme becomes identical to the word relaxation scheme and no change is effected between the original data and the processed data. On the other hand, if the threshold parameters are made (16,16,0) with one msec small and large window, all the data is changed to that of a perfect silence. Other variations of threshold parameters and window widths, if not *all* optimal, may fail one way (by retaining slight choppiness in the words) or the other (by imperfect idle channel noise suppression). Hence the search for optimal threshold parameters and window widths has led us to a minimum cluster threshold parameter of three* and to a transition

* This defines that the presence of an "instantaneous" frequency generated with a cluster whose width exceeds three (i.e., 2 kHz and below at 16 kHz clock rate) within the large window qualifies the small window to be classified as genuine speech data.

Table I — Control parameters and quality

File Name	Durations in msec Before	Small window	After	Threshold parameters Ones	Zeros	Transitions	Remarks, subjective
A	3	3	3	3	3	80	Noisy silence, message OK.
B	32	17	32	2	2	896	Noisy silence, message OK.
C	3	3	3	3	3	88	Slightly noisy silence, slightly choppy words.
D	3	3	3	3	3	72	Perfect silence, choppy words.
T	64	3	64	3	3	1856	Noisy silence, message OK.
U	64	3	64	3	3	1664	Perfect silence, message OK.
V	64	3	64	3	3	1536	Perfect silence, very slight choppiness of words.
P	64	3	64	3	3	1728	Trace of crackle in the silence periods, message OK.
Y	18	3	18	3	3	480	Very slight crackle, very slight choppiness.
Z	18	3	18	3	3	512	One crackle in 8 sec of silence, message OK.
P*	18	3	18	3	3	480	One crackle in 16.2 sec of silence.
O*	18	3	18	3	3	544	Three crackles in 16.2 sec of silence.

Notes: 1. The clock frequency is 16 kHz.
2. The message is a typical standard telephone announcement.
3. Large window width is sum of three columns Before, Small window and After.
* The imperfect silence of the encoder is processed here.

threshold parameter of about 0.783[†] times the maximum number of transitions that can occur in the large window; the large and small windows themselves being 100 to 131 and 1 to 5 msecs respectively. This transition threshold parameter is consistent with our estimation of the frequency of 01 or 10 transitions and 00 or 11 and 000 or 111 cluster formations during silence periods. Sixty percent of the time the encoder generates a 01 or 10 transition, thirty percent of the time the 00 or 11 cluster appears and ten percent of the time the 000 or 111 cluster is generated. These statistics yield the transitions threshold parameter as 0.783[‡] times the maximum transitions one may expect.

Computationally optimized values of the transitions threshold verify the result. For instance, in case of File U (Table I) processed with a large window width of 131 msec, the maximum number of transitions possible are 2096 at a clock rate of 16 kHz and the estimated transitions for the silence period are thus 1634. The computationally optimized value is 1664. A similar assertion of the threshold parameters may be made for File P (Table I) with an estimated value at 489 transitions and a computationally optimized value of 480 transitions.

[†] This defines that if the average frequency generated in the large window is above 6.26 kHz (i.e., 0.783 × 8 kHz), then in the absence of the cluster clues, the data in the small window is the imperfect silence of the encoder.

[‡] $0.783 = 0.6 + 0.3/2 + 0.1/3$.

V. RELATION BETWEEN OPTIMIZED PARAMETERS AND DATA ON SPEECH-SILENCE STATISTICS

The computationally optimized control parameters are consistent with published speech statistics.[2]

Low level consonant sounds (some plosives "p", "t" and fricative "θ"), which are short (30–50 msec) are not mistaken as silence because of neighboring higher level signals which precede or follow them. The cluster clues in the large window (up to 131 msecs wide) fail to be triggered. Longer duration (up to 230 msecs) low level consonant sounds especially sibilant "s" and fricative "z" may tend to trigger the cluster clues but fail to trigger the transitions clue because of the larger value of the transitions threshold parameter. Other plosives, fricatives, sibilant-fricatives, nasals and semivowels lasting between 80 to 200 msec falling between the two earlier cases fail to trigger either the cluster clue (because of neighboring clusters in the large window) at the lower limit of 80 msec or the transitions clue (because of high value of transitions expected in the large window) at the higher limit. The fricative "f" and the semivowel "r" which have some spectral energy at about 3500 Hz and also last between 200 and 160 msecs, still fail the transitions clue since the average limit for the frequency of the imperfect encoder silence is about 6.26 kHz. Higher level consonants and key word vowels fail to trigger either cluster clue or the transitions clue because of their higher amplitude. As has been determined by the variation of the large window width on the computer, the results obtained by longer larger windows are more satisfactory than those from much shorter ones.

The nature of the compandor (syllabic or instantaneous) also plays a role here. Consider a syllabic compandor whose encoded data is being scanned by a large window width of about 70 msec and a long sibilant or fricative that is at the end of a word. If the step size is large due to earlier letters in the word, then a false trigger of both the cluster and transitions clues is possible. Such a process yields a choppy ending of the word. However, a larger window of about 300 msec will eliminate this condition. Conversely, data from an instantaneous compandor would fail to trigger either of the two clues even with a 70 msec large window.

Further, we have noticed that an exact central spacing of the smaller window in the larger window is less desirable than spacing the smaller window towards the end (shifted 10–15 percent). Such a spacing has two advantages: (*i*) false clues by long sibilants and fricatives at the end of words (when the step size of the ADM encoder is already large due to earlier letters in the word) are eliminated and (*ii*) crackles in the silence periods located just at the beginning of words are correctly eliminated (because the step size is low due to the preceding silence, a cluster is most likely formed at the beginning of the utterance). A similar observation has also been made in Reference 1 while detecting the end of an utterance. The critical duration after which code word energy falls below the threshold is 160 msec as against 50 msec for the energy to be above the threshold while detecting the beginning of the utterance. Large windows, inconsistently long in relation to the type of companding of the encoder face the risk of imperfectly suppressing the idle channel noise, and conversely large windows too narrow for the type of companding in the encoder face the risk of leaving behind traces of choppiness in the words.

From the study presented in the paper for the data from an ADM encoder* with an attack time constant of 3 msec and a decay time constant of about 9 msec, the large window should be about 120–150 msec encompassing a small window of 3–5 msec located at 60 percent from the start of the large window duration. The cluster threshold constants are each 3 and the transitions threshold is approximately 0.783 times the maximum number of transitions possible in the large window.

VI. CONCLUSIONS

Idle channel noise may be virtually eliminated from stored ADM messages by the block relaxation techniques presented in this paper. With a proper selection of the control parameters, the silence periods can be accurately located and replaced by a perfect silence. At very low clock rates (about 16 kHz) where the compromise between idle channel noise and intelligibility is very real, the intelligibility can be enhanced by recording with a low step size, which implies a disturbing component of the idle channel noise. This can be completely suppressed, however, by block relaxation techniques.

The flexibility offered by this technique also permits the detection of silence over a wide range of clock frequencies and amplitudes of the recording message. Typically it is necessary to scan about 70–200† milliseconds before judging the authenticity of a silence period and to base the judgment on the number of transitions and consecutive ones and zeros in that interval of time.

REFERENCES

1. L. H. Rosenthal, R. W. Schafer, and L. R. Rabiner, "An Algorithm for Locating the Beginning and End of an Utterance Using ADPCM coded Speech," B.S.T.J., *53,* No. 6 (July–August 1974), pp 1127–1135.
2. D. L. Richards, "Telecommunications by Speech," New York: John Wiley, 1973, Sec. 2.1.3.3.

* With syllabic companding.

† H. Seidel and C. H. Bricker have implemented the concepts in real time hardware, arriving at similar values.