The Use of Companding in Conferencing Voice Communications Systems

Jon P. Klages
University of Central Florida

Part of the Engineering Commons

Find similar works at: https://stars.library.ucf.edu/rtd
University of Central Florida Libraries http://library.ucf.edu

This Masters Thesis (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation
https://stars.library.ucf.edu/rtd/694
THE USE OF COMPANDING IN CONFERENCING
VOICE COMMUNICATIONS SYSTEMS

BY

JON PARRY KLAGES
B.S.E., University of Central Florida, 1981

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Engineering in the
Graduate Studies Program of the College of Engineering
University of Central Florida
Orlando, Florida

Summer Term
1983
ABSTRACT

Companded codes are used for representing voice data in digital communication systems. This thesis addresses the use of the Mu-law companding algorithm in a system optimized for conferencing. A procedure for determining the degree of compression for a variable number of conferees and design equations for implementing a table-lookup scheme using read-only-memories are presented.
# TABLE OF CONTENTS

LIST OF FIGURES ........................................ iv

I. INTRODUCTION ........................................ 1

II. PULSE CODE MODULATION ................................. 2

  Sampling ............................................. 3
  Quantizing .......................................... 4
  Coding ............................................. 6

III. COMPANDING ........................................ 8

  Statistical Model of Speech ........................ 8
  Uniform Quantizing .................................. 10
  Non-Uniform Quantizing ............................... 14

IV. COMPANDED SUMMING .................................. 19

V. DIGITAL CONFERENCING ................................ 27

VI. DESIGN EXAMPLE ..................................... 31

VII. CONCLUSION .......................................... 42

  Follow-on Efforts .................................... 45

APPENDIX: HP-41C PROGRAMS .............................. 51

  Find SNR(dB) For a Uniform Quantizer ............... 48
  Find SNR(dB) For a Mu-Law Quantizer ................. 49
  Plot Normalized Mu-Law Characteristic ............... 50
  A Mu-Law Encoder .................................... 51
  A Mu-Law Decoder .................................... 52
  Program the Translate ROM ............................ 53
  Program the Summer ROM .............................. 54

LIST OF REFERENCES .................................... 55
LIST OF FIGURES

1. Sampling theorem ........................................... 5
2. Signal quantization ........................................... 5
3. Probability distribution of speech ......................... 9
4. Normalized output vs. input, uniform signal ............ 11
5. Quantized output vs. input, uniform signal ............. 11
6. SNR(dB) of uniform quantizing ............................ 13
7. Normalized output vs. input, compressed signal ........ 15
8. Quantized output vs. input, compressed signal ........ 15
9. SNR(dB) of mu-law quantizing ............................. 17
10. SNR(dB) vs. mu for mu-law quantizing .................. 18
11. B-bit mu-law compressor (coder) ........................ 21
12. B-bit mu-law expander (decoder) ........................ 21
13. Translator .................................................... 22
14. Summer ....................................................... 24
15. System ....................................................... 25
16. Alternate system ............................................. 25
17. Ram switch .................................................. 28
18. Ram switch with conferencing ............................. 28
19. Dynamic range .............................................. 32
20. Mu = 4110 .................................................. 34
21. Translate mu1=255 to mu2=4110 and back ............... 36
22. Summer with mu=4110 .................. 38
23. Block diagram of design example ........ 40
24. Block diagram of multi-translate version ... 41
25. PCM in tandem ........................... 44
I. INTRODUCTION

Companding voice transmissions has come about because, in the words of K. W. Cattermole, "No rational telephone system would use uniform quantizing" (Cattermole 1969, p.130). He was referring to the fact that the human voice does not have a uniform probability distribution. When it is encoded uniformly, certain parts are more accurately encoded than others. A scheme used to overcome this problem is to pass the voice signal through a compander so that the encoding will be uniform over the signal. It is typical practice in telephony today to use a companding scheme during transmission.

In a typical voice conferencing system, companded signals are decoded into their linear form, summed, then the result is re-companded for transmission. In a system where voice data is companded at the end-instrument, certain benefits might be gained if the voice data could be summed and switched in its companded form.

When voice data is combined in a conferencing system, the voice signals are summed together and the result is a new signal with properties different than that of a single speaker. These properties may influence the realization of a companding scheme.
II. PULSE CODE MODULATION

Common impairments suffered by telecommunications are amplitude distortion, frequency distortion and noise. Amplitude distortion usually results from some non-linear component in the system, therefore, the output is not just a linear function of the input. Frequency distortion results when the system responds differently to different frequencies. Noise is the addition of unwanted signals superimposed on the message signal. These impairments can be lumped together as noise or distortion through the system. If they are made small enough, they can be totally overlooked by the average listener. "Something between 20 and 27 dB signal-to-noise is desirable; anything less than 20 is perceptible to most people and offensive to some, while anything better than 27 is rarely if ever perceptible. Lower values down to about 16 dB, detract little from the intelligibility of speech; though listening to such sounds is fatiguing" (Cattermole 1969, p.234). These specifications are considered "telephone quality" as opposed to high fidelity and can suffer much more degradation before intelligibility is lost.

It is clear, then, that the signal received need not be an exact replica of the transmitted signal and that the
system can still be considered good. The ear detects by correlation since man evolved in all but the best of listening environments.

A discrete representation of a signal, in general, can withstand much greater impairments than the original signal. While the signal, itself, can tolerate quite a bit of impairment, a discrete representation can survive around noise with an amplitude almost as big as itself. Also, if the discrete signal can be interpreted, the result is the same as the unimpaired signal.

A continuous signal can be represented by a plot of time versus amplitude. A discrete version can also be represented this way with the magnitudes provided by sampling and quantizing respectively.

**Sampling**

Sampling is equivalent to reading points from a graph. This amounts to evaluating the continuous signal over a relatively short period of time during which the signal changes only a negligible amount. A relationship between an analog signal and its discrete counterpart is known as the Uniform Sampling Theorem. The word "Uniform" implies identical spacing between samples. If a signal \( x(t) \) contains no frequency components for frequencies above \( f=W \) Hz, then it is completely described by instantaneous sample values uniformly spaced in time with period \( T_s<1/2W \). The
signal can be exactly reconstructed from the sampled waveform by passing it through an ideal lowpass filter with bandwidth $B$, where $W < B < (f_s - W)$ with $f_s = 1/T_s$. The frequency $2W$ is referred to as the Nyquist frequency. It can be seen in Figure 1 that $f_s$ has to be at least twice $W$ or the higher bands would cross over and disturb the baseband. By placing the ideal lowpass filter as shown, it is obvious that an exact replica of the original signal can be obtained. Although ideal filters are not really available, the spectral range of telephony ($< 8$ KHz) allows us to sample at large enough frequencies such that "real" filters are totally adequate.

Quantization

After a signal has been sampled, the analog version may be used to represent the signal or, more commonly, it is divided into levels which are discrete themselves. This process is called quantization and is the major source of impairment in these types of systems. The impairment is an error due to the difference between the actual amplitude and the quantized amplitude and it is referred to as quantization noise even though it does not manifest itself in the same way that we normally associate with the word noise (Catermole 1969, p.233). The quanta do not have to be uniformly spaced and, in general, are not. An example of uniformly spaced quantization is shown in Figure 2.
Sampled waveform = \( x_d(t) = x(t) \sum_{n=-\infty}^{n=\infty} d(t-nT_s) \)

where \( d(t) \) is the impulse function and \( T_s \) is the sampling interval.

The Fourier transform is

\[
X_d(f) = f_s \sum_{n=-\infty}^{n=\infty} X(f - nf_s).
\]

Figure 1. Sampling theorem

baseband = 2W

sampled spectra with low-pass filter

Figure 2. Signal quantization
Coding

Translation of the quantized signal to a form suitable for transmission is called coding. Typically the processes of quantizing and coding are inseparable but it helps to analyze them separately. "The purpose of encoding is to represent one element of high radix in the form of a group elements of low radix: the latter being better suited to the transmission path" (Cattermole 1969, p.15). This is exemplified by using a coded signal of radix of 2 to represent a quantized sample of radix 256. A radix of 2 can tolerate more noise than 256 but the tradeoff is that the lower radix requires more elements and, therefore, a larger bandwidth. Telephone transmission lines are an obvious choice for such coding. Many different codes exist mostly due to the circuitry developed to encode then decode them. A typical code is a symmetrical binary number which represents the numbers, say, from -127 to 127 in an 8 bit word. Also typically, the values will be complemented due to the nature of digital repeaters.

PCM, Pulse Code Modulation, is the combination of sampling, quantizing and coding and is used widely in modern telecommunications.
The advantages of PCM are among the following:

1. Quality is independent of distance and network topology.
2. Compatible to different media.
3. Compatible to different traffic.
4. Increased capacity.
5. Economics.
6. Applicable to difficult transmission paths.

(Cattermole 1969, p. 27)
III. COMPANDING

A coded signal conveys maximum information only if all levels are equiprobable. Since all levels of a uniformly quantized speech signal are not equiprobable, the coded result would not convey maximum information. Particularly, the lower levels, which occur most often, would be coded the same as the higher levels that only occur rarely. Clearly, this is not the most optimal use of these quantization levels (Note, quantization from this point on, implies quantization plus encoding and vice versa).

Statistical Model for Speech

An ergodic random process is one in which the time average equals the ensemble average. This implies that the process is stationary, that is, an arbitrary choice can be made for the time origin. Using this model for speech provides a powerful tool. The Gamma probability distribution function is a good approximation of speech but a simpler more often used model is the Laplacian distribution. Figure 3 shows both in a normalized form, that is, the mean is zero and the variance is unity (Rabiner 1978, p.176). Note that 99.65% of the speech samples fall within the range from minus four sigma x to plus four sigma x. Where sigma x is the square root of the
The gamma distribution function

\[ p(x) = \left( \frac{\sqrt{3}}{8\pi \sigma_x |x|} \right)^{1/2} \exp(-\sqrt{3} |x| / 2\sigma_x) \]

The Laplace distribution function

\[ p(x) = \left( \frac{1}{\sqrt{2} \sigma_x} \right) \exp(-\sqrt{2} |x| / \sigma_x) \]

Figure 3. Probability distribution of speech
variance or the RMS, root mean square, value of the signal assuming the mean is zero (capacitor coupled signal). This model for speech provides us with valuable information pertaining to clipping. Since a large majority of the signal is contained between \( \sigma_x = -4 \) and \( +4 \), there will only be a small chance (0.35%) of a value falling outside this range. If we only allow \( \sigma_x \) to vary between plus and minus 8, then chance of overload or clipping is definitely negligible. This criterion will be used in the formulation of the design procedure.

**Uniform Quantizing**

In the case where the distribution of the quantized data is uniform, like it might be if it were data from a computer, uniform quantizing is optimal. The normalized output versus the normalized input is shown in Figure 4. The normalized output versus the normalized input after it has been quantized into 8 levels is shown in Figure 5. The normalized input is an analog value that varies continuously between -1 and 1. The output is also normalized but is restricted to the discrete values of \(-7/8, -5/8, -3/8, -1/8, 1/8, 3/8, 5/8, \) and \(7/8\). The binary numbers represents one of many 3 bit codes that could be used to define the output. If the input sampled value is between 0 and 0.25, the code 111 is generated which is decoded at the output to be \(1/8\). If the input sampled
Figure 4. Normalized output vs. input, uniform signal.

Figure 5. Quantized output vs. input, uniform signal.
value is between -0.75 and -0.5, a code of 001 is generated which produces an output of -5/8. Since the quantization is uniform, the error is uniformly distributed between two sample intervals. The SNR(dB), Signal-To-Noise Ratio in decibels, is given in Figure 6. Plots of SNR(dB) versus $X_{\text{max}}$ divided by sigma $x$ for various numbers of encoding bits can, also, be seen in Figure 6. Note that for large signals, in the average sense, the SNR is very good, but for small signals (right side of graph), the SNR can be very poor. A reasonable design target will be an SNR greater or equal to 34 dB as per Figure 3.12 of Principles of Pulse Code Modulation (Cattermole 1969, p.146). This value is greater than the 20 through 27 dB range mentioned earlier and will be explained later. Notice that for an 11 bit data word $X_{\text{max}}$ divided by sigma $x$ is approximately 68 (137 for a 12 bit data word) to achieve the target SNR. A reasonable range for $X_{\text{max}}$ divided by sigma $x$ is approximately 100 (Cattermole 1969, p.222). If the input is speech, then 8 is the lowest acceptable value to prevent clipping. Thus, an $X_{\text{max}}$ divided by sigma $x$ of 108 must produce a SNR of at least 34 dB. A minimum of a 12 bit data word is required to produce this, which in turn produces an exorbitant SNR (59 dB) at a left side value of 8. Clearly, uniform quantization is not the optimal method for representing voice signals.
\[ \text{SNR}(\text{dB}) = 6B + 4.77 - 20 \log_{10} \left( \frac{X_{\text{max}}}{\sigma_X} \right) \]

where \( B \) = number of bits

\( X_{\text{max}} \) = maximum value of signal

\( \sigma_X \) = RMS value of signal.

Figure 6. SNR(dB) of uniform quantizing
Non-Uniform Quantizing

Compand is a contraction of the words compress and expand. Compress refers to non-uniform quantizing and expand refers to the inverse operation to obtain the original signal. Many different type of compressions (laws) exist and each has its own equation to describe the distribution of quanta. The mu-law will be discussed here since it is the best trade-off between SNR and background noise (Rabiner 1978, p.195).

The model used in this report for speech predicts that small amplitude signals have a higher probability of occurring than larger ones. It would be beneficial, then, to assign more levels of quantization to the smaller signals and less to the larger ones. Figure 7 shows the normalized mu-law equation along with the input (x) versus output (y) plot. It should be noticed that a value that corresponds to half the input amplitude translates to 88% of the output amplitude. Now if uniform quanta were spread up and down the y axis, many more levels would be assigned to the smaller amplitude signals. The normalized output versus the normalized input after it has been quantized into 8 levels is shown in figure 8. The inputs and outputs can take on all the same values as in the uniform case, but, 3 of the 4 positive levels are used up before the normalized input is 0.1.
\[ Y = \frac{\ln(1+\mu |x|)}{\ln(1+\mu)} \cdot \text{SIGN}(X) \]

\[ \text{SIGN}(X) = \begin{cases} 
1 & \text{if } X \geq 0 \\
-1 & \text{if } X < 0 
\end{cases} \]

Figure 7. Normalized output vs. input, compressed signal

\[ Y_1 = \text{RNDINT}((2^{B-1}-1)Y) \]

\[ \text{RNDINT} = \text{rounded integer} \]

Figure 8. Quantized output vs. input, compressed signal
The SNR in dB is given in Figure 9. Also in the figure are plots of SNR(dB) versus $X_{\text{max}}$ divided by sigma x for mu values of 100, 255 and 500 (Rabiner 1978, p.190). Using an $X_{\text{max}}$ divided by sigma x of 108 and minimum SNR of 34 dB found above, the plots with mu's of 255 and 500, both, meet the criterion. Note that for only a 2.53 dB drop in SNR from mu=100 to mu=500, the 43 dB point shifted from 103 to 240. This more than doubles the dynamic range.

A different perspective is shown in Figure 10 (B = 7). Here, mu is plotted against SNR(dB) for various values of $X_{\text{max}}$ divided by sigma x. This plot is a view looking from the right edge of Figure 9 with with increasing mu going into the page. Notice that at relatively small mu's there is a large difference in SNRs between an $X_{\text{max}}$ divided by sigma x of 1 and 2000, but at large mu's, the difference becomes negligible.

The mu-law companded code, in general, seems well matched to the distribution of single speaker voice data, but can it be used effectively in a conferencing system where multiple speakers data is encoded?
SNR(dB) = 6B + 4.77 - 20\log_{10}[\ln(1 + \mu)]
- 10\log_{10}[1 + (\frac{x_{\text{max}}}{\sigma_x})^2 + \sqrt{2} (\frac{x_{\text{max}}}{\sigma_x})]

where B = number of bits

\(x_{\text{max}}\) = maximum value of signal

\(\sigma_x\) = RMS value of signal

Figure 9. SNR(dB) of mu-law quantizing
Figure 10. SNR(dB) vs. mu for mu-law quantizing
"Single speaker speech has been well investigated and the literature is readily available" (Petrasko 1982, p.4). The SNR(dB) equation of Figure 9 does not suggest that multiple speakers are ruled out. In fact, it suggests just the opposite. The variables of the equation are mostly system parameters and only one deals with the signal itself. That variable is $X_{\text{max}}$ divided by sigma x. The only assumptions are such: The input signal must be expressible as a random variable. The quantization step size must be small enough to remove any possibility of signal correlated patterns in the noise waveform. Quantizer overload must be negligible (Rabiner 1978, p.189).

When two signals are summed together, their composite produces a new maximum value that is twice the individual value, this assumes the individual maximum values were originally the same. Consequently when n signals are summed together the new maximum is n times their original maximum. This leads to calculating a new mu which, in turn, develops the need for a new component to translate from the input mu to the system mu. The new component will be called a translator and will be needed at both the input and output to the summer system.
Typically, the input to the summing system will be a B bit binary word from a compander of some mu. The equation for the compression part of the compander is presented in Figure 11 with Y being a binary word produced by some voice signal X. Figure 12 is the equation for the expansion part of the compander which will transform a B bit binary word into a voice signal. Clearly, if some Y(mu2) in terms of some Y(mu1) is desired, the equation of Figure 12 is used with that of Figure 11. The result is in Figure 13. N is included to translate from a maximum X of the individual speakers to a system maximum and back. So N will be the maximum number of speakers to be summed if it appears at the input to a summer. If it is at the output of a summer, then it will be 1/N, 2/N, 3/N, ..., #/N, where # is the actual number summed and cannot be greater than N. This implies that there might be up to N translators at the output of a summing system. The proper one can be picked if base 2 log(N) bits are carried through the system along with the data to indicate the number of sums. The extra bits could be decoded to 1 out of N and select one of the translators for output. It seems that this is really not needed, though, assuming the SNR in the summer is of an acceptable value. It would tend to regulate the volume since different sums would be relative to their own maximums. If N was just made a 1 on the output, everything
\[ Y = \text{RNDINT} \left( \frac{\ln(1 + \mu |x|/X_{\text{max}})}{\ln(1 + \mu)} \text{SIGN}(X) \right) \]

where RNDINT = rounded integer
\( X/X_{\text{max}} \) = normalized signal input
\( Y \) = digital output
\( B \) = number of bits

\[-1 \leq X/X_{\text{max}} \leq 1\]
\[-(2^{B-1}-1) \leq Y \leq (2^{B-1}-1)\]

Figure 11. B-bit mu-law compressor (coder)

\[ X/X_{\text{max}} = \frac{\exp \left[ \frac{|Y| \ln(1 + \mu)}{(2^{B-1}-1)} \right] - 1}{\mu \text{SIGN}(Y)} \]

where \( Y \) = digital input
\( X/X_{\text{max}} \) = normalized signal output
\( B \) = number of bits

\[-1 \leq X/X_{\text{max}} \leq 1\]
\[-(2^{B-1}-1) \leq Y \leq (2^{B-1}-1)\]

Figure 12. B-bit mu-law expander (decoder)
\[ Y_{\mu 2} = \text{RNDINT} \left[ \frac{\ln \left( 1 + \mu 2 \right) \exp \left[ \frac{Y_{\mu 1}}{N \mu 1} \ln(1 + \mu 1) \right] - 1}{\ln(1 + \mu 2)} \right] \]

where

- \text{RNDINT} = \text{rounded integer}
- N = \text{number of speakers (if translating into a summer)}
  = 1/N through N (if translating out of a summer)
- B = \text{number of bits (input and output have same number)}

\[-(2^{B-1}-1) \leq Y_{\mu 1} \leq (2^{B-1}-1)\]

Figure 13. Translator
would be relative to a maximum of N sums and the volume would grow with the number of speakers.

Summing two companded codes together is essentially the same as translation. The equation from Figure 12 is used twice, once for each input. They are summed together them substituted into the equation of Figure 11. The difference here is that all the mu's have to be in translated form, that is, be proper for summing. This also assumes that all maximum X's are the same. The equation for programming a summer is in Figure 14.

A system using translators and a summer is shown in Figure 15 in block diagram form. It is assumed that each of the blocks will be implemented in a table look-up fashion with a ROM (Read Only Memory). An alternate implementation is shown in Figure 16, in which the table look-up has been extended so that the translators are incorporated in the summer ROM.

Since quantization originates in the compressor, every time a translator is encountered, the system will endure some degradation due to the error inherent in this process. The summer can be modeled in a similar manner. The result, again, is that for each sum, a quantization error incurred. Referring back to Figure 15, each translator introduces a quantization error and the summer adds one for each sum it does.
\[ Y_\Sigma = \text{RNDINT} \left[ \frac{(2^{B-1}-1)}{\ln(1 + \mu)} \ln \left( 1 + \frac{x_1 + x_2}{x_{\text{max}}} \right) \right] \text{SIGN}(x_1 + x_2) \]

\[
\frac{x_1 + x_2}{x_{\text{max}}} = \frac{\exp \left[ \frac{|Y_1| \ln(1 + \mu)}{(2^{B-1})} \right] - 1}{\mu} \text{SIGN}(Y_1) + \frac{\exp \left[ \frac{|Y_2| \ln(1 + \mu)}{(2^{B-1})} \right] - 1}{\mu} \text{SIGN}(Y_2)
\]

\[-(2^{B-1}-1) \leq Y_1 \leq (2^{B-1}-1)\]
\[-(2^{B-1}-1) \leq Y_2 \leq (2^{B-1}-1)\]

Figure 14. Summer
new data \[\xrightarrow{\mu_1} \] XLATE \[\xrightarrow{B} \] SUM \[\xrightarrow{\mu_2} \] XLATE \[\xrightarrow{B} \] current data

XLATE = translator
B = number of bits

Figure 15. System

new data \[\xrightarrow{\mu_1} \] XLATE \[\xrightarrow{B} \] SUM \[\xrightarrow{\mu_2} \] XLATE \[\xrightarrow{B} \] current data

Figure 16. Alternate system
Thus, quantization effects the system in two ways. The first is that associated with quantizing a small signal relative to some large maximum value. The second being that produced by the summing process. In order to achieve acceptable results, a system will be developed that will produce satisfactory SNR for single speaker using an n speaker channel and, again, be satisfactory when n sums are performed.

We have seen that companding requires fewer bits than uniform quantizing to maintain an acceptable SNR, therefore, a system using companding summers would require smaller bus widths. In complex digital switching systems, controlling bus widths is a major consideration.
V. DIGITAL CONFERENCING

There are many digital switching concepts and design ideas, most of which, have a primary function, which is to connect a large number of subscribers in pairs. Extra equipment is usually needed to establish a conferencing mode and, at the least, it is usually an add-on task, and the system is not optimized in any way for it.

A proposed digital switching system like the one in Figure 17, operates as follows. Voice data is presented to the input in a time multiplex fashion. The data is an n bit binary word and coded in some arbitrary way. The input time line shows what we will call a "frame" of information. A, B, C, and D represent the users identification in this four person system. As it can be seen on the input time line, all four users are sampled, in order, every frame time, F. A pair of RAMs (Random Access Memory) act as the switch. During each frame, the input voice data is routed to one RAM while the the other RAM supplies data to the output. Synchronized with the data being presented to the input, another set of RAMs, called the User Maps, address the Switching RAM. Every frame time, the switching RAMS "change place" and the one that was being written into, (input) becomes the output and vice versa. The effect is a switching action, because, if the user time slots are being
Figure 17. RAM switch

Figure 18. RAM switch with conferencing
sampled on the output at approximately the same time they were presented on the input, a different user might be found. Of course, there is a minimum time delay of one frame time through the system. The example shown on the time lines in Figure 17 show a switching configuration of A with B and C with D.

Figure 17 is just a switching system and has no ability to establish a conference. If more than one voice is read into a memory location, the last one put there will be the one that reaches the output.

Figure 18 shows just the input side of the switch with a slight modification, the voice data has to go through a summer before it reaches the Switching RAM. The other input to the summer is connected to the output of the RAM, such that, the word that gets put into the RAM is a sum of the new data and what was already there. The time lines show a situation where B, C, and D are put into A's slot and A is put into C's slot. Note that the memory must be cleared before the first sum of each frame to insure left over data from the previous frame is not included in the sum. Also note, that if the two inputs to the summer are n bits, and the data is represented as binary numbers, the result needs to be n+m bits to insure that the sum is correct. The m is some number that depends on the maximum number of sums the system can handle.
If the number of bits representing the summed voices grows large, the system can become unwieldy. In systems that stress the conferencing mode, it does. Thus, it would be desirable to have a code that uses a compact number of bits and does not grow larger when summed. Companded data seems to provide us with an answer, but it also provides us with a problem. Normal off-the-shelf summers do not sum companded code. An answer comes to us in a form that seems to be able to solve any digital logic problem, a memory device. The inputs are connected to the address lines and the answer is read from the data lines. A ROM (Read Only Memory) can be programmed so that the data in each address represented the sum of, say, the higher and lower order address bits. Its implementation could be exactly the same as that shown in Figure 18 with the summer being represented by a ROM. In a more complicated system, one only need replace the summers with ROMs, reducing the number of bits required for the data bus.
VI. DESIGN EXAMPLE

The goal of this design is to develop a system that can conference up to 16 speakers with a quality comparable to that of a typical companding device that handles a single speaker. The data will be kept in its companded form throughout the system. The basic design concept will be the same as that of a linear digital system, such that, multiplexed data will be brought in, and summed to memory locations. Since the data is to be kept in its companded form, a companded code will have to be developed that will have a dynamic range that includes all from the quiet single speaker to 16 loud speakers. The dynamic range considered here will be those values of $X_{\text{max}}$ divided by sigma $x$ that are greater than 8 and within the area where the SNR remains within 2dB of the maximum SNR (Rabiner 1978, p.191) as in Figure 19.

A typical companding device will be one such as the 2910 mu-law PCM codec made by Intel which has a mu of 255. Using the SNR(dB) equation for mu-law companders, Figure 9, the right-hand limit of the graph is calculated to be $X_{\text{max}}/\sigma_x=86$. The left-hand side of the graph will be 8 so that a negligible portion of the signal will fall outside the range, therefore, not be clipped (Rabiner 1978, p.180).
Using equation from Figure 9 with $B = 8$ and $\mu = 255$.

Maximum SNR = 37.87 dB \ $(X_{\text{max}}/\sigma_x = 1)$.

It is down 2 dB at $X_{\text{max}}/\sigma_x = 86$.

This implies $\sigma_x = X_{\text{max}}/86$ for a single speaker.

In a 16 speaker system $X_{\text{max}}$ becomes 16 $X_{\text{max}}$ so

$$\frac{X_{\text{max}}}{\sigma_x} = \frac{16 X_{\text{max}}}{X_{\text{max}}/86} = 1376.$$ 

Using equation from Figure 9, find $\mu$ that is 2 dB down at $X_{\text{max}}/\sigma_x = 1376$.

This implies $\mu = 4110$. 

Figure 19. Dynamic range
This means the acceptable limits for $\frac{X_{\text{max}}}{\sigma_x}$ will be between 8 and 86 for a single speaker.

If $X_{\text{max}}/\sigma_x = 86$, that implies $\sigma_x = X_{\text{max}}/86$ for the softest acceptable single voice. Applying this to a 16-speaker system where $X_{\text{max}} = 16$, then $X_{\text{max}}/\sigma_x = 16/(1/86) = 1376$ is the new right-hand limit that system must have. The old left-hand limit of 8 still applies to this system since a single speaker does not want to be clipped. Using the SNR(dB) equation, a mu of 4110 is found to be needed to make the dynamic range wide enough to encompass one to 16 voices. A plot of normalized input versus output is shown in Figure 20. The maximum SNR(dB) for a mu of 255 is 3.51dB more than that for mu=4110, but if another bit is added to compensate, the mu=4110 system becomes 6.00dB better than the mu=255 system. Since the goal of the design is to approach the quality of the mu=255 single-speaker system, not better it, we will use the same number of bits that it uses and accept the fact that the quality might be slightly less.

The 2910 mu-law PCM codec uses 8 bits for it's data word and is typical of the kind of input and output the conferencing system will be seeing. This requires that the system have translators to convert the mu=255 data to mu=4110 data on the input and back again on the output. Also, if translating into a summing system, $X_{\text{max}}$ will have to be adjusted to reflect the maximum number of sums but,
For an 8 bit quantized system:

\[ Y = 127 \frac{\ln(1 + 4110|X|/X_{\text{max}})}{\ln(4111)} \frac{\text{SIGN}(X)}{0.5} \]

In the graph, X and Y have been normalized.

*Figure 20. Mu = 4110*
if translating out of the system, the system $X_{\text{max}}$ must be maintained since this relates the actual new $X_{\text{max}}$. The translators, in both cases, need only be ROMs with 8 bit address lines (255 words) and 8 bit data words. This is exemplified in Figure 21 which shows the equations used to program the translators. Note that the same equation is used on both the input and the output to the system, the only differences being the values of $mu$ and $N$.

It should be noted that the output $X_{\text{max}}$ of the system is based on $N \times X_{\text{max}}$ of a single user and that the full range of the output translator will not be used unless $N$ speakers are being summed. This should not present a speech quality problem since the SNR is within 2dB of the maximum but, a scheme to use the full range of the translator (auto volume?) would be to carry an extra 4 bits of data through the system to indicate the number of sums that really occurred. These four bits would then be used on the output to pick one of 16 translators where each has been programmed with the equation of Figure 21 with $N$ replaced by $\# \div N$ where $\#$ is the actual number of speakers that were summed.

The summer will also be a ROM. Since two 8 bit companded words are present, 16 bits of address will be used which presents us with 64k memory locations or "possible answers" which are also 8 bits. It must be stressed that the input to these summers must be translated
Translate \( u_1 = 255 \) to \( u_2 = 4110 \).

\[
Y_{u_2} = \text{RNDINT} \left( \frac{\ln (1 + 4110)}{(127)} \exp \left[ \frac{|Y_{u_1}| \ln (256)}{(127)} \right] - 1 \right) \frac{\ln (4111)}{\ln (256)} \text{SIGN}(Y_{u_1})
\]

\(-127 \leq Y_{u_1} \leq 127\)

Translate \( u_2 = 4110 \) to \( u_1 = 255 \)

\[
Y_{u_1} = \text{RNDINT} \left( \frac{\ln (1 + 255)}{(127)} \exp \left[ \frac{|Y_{u_2}| \ln (4111)}{(127)} \right] - 1 \right) \frac{\ln (M(4110))}{\ln (256)} \text{SIGN}(Y_{u_2})
\]

\(-127 \leq Y_{u_2} \leq 127\)

\(1/N \leq M \leq N/N\)

Figure 21. Translate \( mu_1=255 \) to \( mu_2=4110 \) and back.
data and that the output from the final summer in the system must be translated back. Figure 22 shows the equations used for programming the summing ROM. Notice that 85 summed 16 times results in an $X_{\text{max}}$. This is because 85 is the value the translator returns for a single speaker $X_{\text{max}}$.

The system has been designed with one summer and one set of translators but it should be obvious that the data could be switched around a much more complex system with the only restriction that a maximum of 16 sums can happen to any set of signals. It should be noted that the equations for both the translators and the summer do not have any reference to $X_{\text{max}}$ or $\sigma_x$ and are only functions of their respective mu's, number of bits (B), maximum number of signals to summed (N) and coded values (Y).

Implementation of the system would require connecting the 8 bit signals from the codecs to the address lines of the Input Translate ROM. The 8 bit output of the Input Translate ROM would then address half of the Summer with the other half being addressed by the data resident in memory. The 8 bit output of the Summer ROM would then be loaded into the location of the Memory Switch that had previously been half its address. The 8 bit output of the Memory Switch would then address the Output Translate ROM which would present its 8 bit data back to the codecs. The system data bus would only be 8 bits regardless of how many
\[ Y = \text{RNDINT} \left[ \frac{\ln \left( 1 + 4110 \right) \frac{X_1 + X_2}{X_{\max}}}{\ln(4111)} \right] \text{SIGN}(X_1 + X_2) \]

\[ \frac{X_1 + X_2}{X_{\max}} = \frac{\exp \left[ \frac{|Y_1| \ln(4111)}{(127)} \right] - 1}{4110} \text{SIGN}(Y_1) + \frac{\exp \left[ \frac{|Y_2| \ln(4111)}{(127)} \right] - 1}{4110} \text{SIGN}(Y_2) \]

\[ -(127) \leq Y_1 \leq (127) \]

\[ -(127) \leq Y_2 \leq (127) \]

Figure 22. Summer with \( \mu = 4110 \)
suns have occurred. Refer to Figure 23 for a block diagram of the system.

Figure 16 presented an alternate method for the system. The translate portion was included in the summer. If the new data presented to the input is always voice data encoded in the untranslated mu-law, then it is the way the system should be implemented. However, if the new data is coming in from some other part of the same system and has already been translated, then the system of Figure 15 would be the choice.

In the case where the full output of the codecs was desired, no matter how many sums, the system would be modified as per Figure 24. Notice the 4 extra bits carried through the system to take care of choosing the proper translator. Each ROM would be programmed, in turn, with M=1/16, M=2/16, M=3/16, M=4/16, etc. where the numerator reflects the actual number of sums that took place.
Figure 23. Block diagram of design example.
Figure 24. Block diagram of multi-translate version
Pulse Code Modulation in the form of Companding has been found to be a viable method to implement a conferencing communication system. The quality and implementation seem very favorable. No obvious drawbacks were found, except maybe the cost of ROMs. Surely though, the savings in bus interconnection would overshadow this.

The design philosophy used herein was based on a few assumptions. These assumptions deal with the sensitivity, or lack of it, to certain parameters. Namely, the actual distribution of summed voices and the actual SNR due to repeated quantizing during the summing process.

The central limit theorem predicts that if a large enough number of probability distributions are summed together the result will be a Gaussian or Normal distribution. In the case at hand it was not clear exactly what the distribution was after two voices were summed or how many sums was a large number. The design was not sensitive to this, though, because it just determined the leftmost side of the dynamic range and was very small compared to the size of the dynamic range itself. A value for $X_{\text{max}}$ divided by sigma $x$ of 8 was chosen using the Gamma or Laplacian distributions for a single speaker. If the
Gaussian distribution was used, it would have even been less since 99.9% of the signal is in plus and minus three sigmas. Thus, a value of 8 has a safety factor of at least two since only 4 sigmas were need for a single speaker.

The system was actually just designed for the softest single speaker whose amplitude is relative to a maximum for 16 speakers. The SNR plots apply to this one voice but fall short when other speakers are summed in. The problem is due to the quantization error introduced each time a sum is done. This can be equated to PCM systems in tandem. An example of this is a telephone switching network where the quantized signal might be converted back to the voice signal and then back to a digital signal many times as in Figure 27. A number of these tandem connections that still produces favorable results is 22. Larger numbers can be accommodated; and though there is considerable distortion, the overall effect is tolerable speech (Cattermole 1969, p.229). In the design example, 16 speakers could be summed but this was the worst case and it would be comparable to 18 tandem operations. Typically fewer speakers would be summed and, thus, the overall quality of the system should be rather good and not just tolerable.

One of the early design considerations was to have a set of table look-up ROMs, such that, depending on the number of sums that had occurred, a different ROM might be
PCM = Pulse Code Modulation
PCD = Pulse Code De-modulation

Figure 25. PCM in tandem
used to optimize the SNR by using the best mu for a certain number of sums. Using Figure 10, it can be seen that the loss in maximum SNR by picking a large mu is negligible, and that the gain in dynamic range is dramatic. Thus, there is nothing to justify the addition of the hardware necessary to implement multiple table look-ups.

"It is interesting to note that the first PCM equipment designed for telephone quality, built at the Bell Telephone Laboratories and completed in 1947, used 8 kHz sampling, 128 levels and 25 dB companding advantage" (Cattermole 1969, P.222). The author notes here that the present systems are not much different from the earliest and that the theory that spawned PCM, quite accurately, predicted it's performance. Hopefully, this will be extrapolated into the calculations presented herein.

**Follow on Efforts**

The obvious follow-up to the design effort is to build a working prototype and subject it to rigorous testing. There is a need to determine its actual limits and compare them to theoretical results. While a quantitative approach is desired during the design phase, the true success will be measured by qualitative observations.

The mu-law was studied here, but many other laws exist and each need to be investigated. The design presented here would work just as well with any type of quantization
scheme, since all that need be changed is the formula for programing the ROM.

In the translator, the number of bits, $B$, was kept a constant. Benefits might be gained if it was allowed to be a variable.

Experiments need to be run to find the qualitative difference between one and $N$ Translators on the output.
APPENDIX

HP-41C PROGRAMS
Find SNR(dB) For a Uniform Quantizer

Inputs:  
- B  
- XMAX/SIGX  

number of bits  
normalized R.M.S. signal

Output:  
SNR(dB) Signal-to-Noise-Ratio in dB

01 LBL UNSNR  
02 B=?  
03 AVIEW  
04 STOP  
05 STO 06  
06 LBL 01  
07 XMAX/SIGX=?  
08 AVIEW  
09 STOP  
10 LOG  
11 20

12 *  
13 CHS  
14 4.77  
15 +  
16 RCL 06  
17 6  
18 *  
19 +  
20 STOP  
21 GTO 01  
22 END
Find SNR(dB) For a Uniform Quantizer

Inputs:
- B: number of bits
- MU: desired mu-law
- XMAX/SIGX: normalized R.M.S. signal

Output:
SNR(dB) signal-to-noise ratio in dB

```
01 LBL SNR 25 /
02 B=? 26 +
03 AVIEW 27 LOG
04 STOP 28 10
05 STO 01 29 *
06 MU=? 30 CHS
07 AVIEW 31 RCL 02
08 STOP 32 1
09 STO 02 33 +
10 LBL 01 34 LN
11 X,MAX/SIGX=? 35 LOG
12 AVIEW 36 20
13 STOP 37 *
14 STO 03 38 -
15 RCL 02 39 4.77
16 / 40 +
17 X 2 41 RCL 01
18 1 42 6
19 + 43 *
20 2 44 +
21 SQRT 45 STOP
22 RCL 03 46 GTO 01
23 * 47 END
24 RCL 02
```
Plot Normalized Mu-Law Characteristic

Inputs:  
- MU  desired mu  
- X  normalized input (-1 to 1)

Output:  normalized output (-1 to 1)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>LBL MU-LAW</td>
</tr>
<tr>
<td>02</td>
<td>MU=?</td>
</tr>
<tr>
<td>03</td>
<td>AVIEW</td>
</tr>
<tr>
<td>04</td>
<td>STOP</td>
</tr>
<tr>
<td>05</td>
<td>STO 05</td>
</tr>
<tr>
<td>06</td>
<td>LBL 01</td>
</tr>
<tr>
<td>07</td>
<td>CF 01</td>
</tr>
<tr>
<td>08</td>
<td>X=?</td>
</tr>
<tr>
<td>09</td>
<td>AVIEW</td>
</tr>
<tr>
<td>10</td>
<td>STOP</td>
</tr>
<tr>
<td>11</td>
<td>X&lt;0?</td>
</tr>
<tr>
<td>12</td>
<td>SF 01</td>
</tr>
<tr>
<td>13</td>
<td>ABS</td>
</tr>
<tr>
<td>14</td>
<td>RCL 05</td>
</tr>
<tr>
<td>15</td>
<td>*</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>+</td>
</tr>
<tr>
<td>18</td>
<td>LN</td>
</tr>
<tr>
<td>19</td>
<td>RCL 05</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>21</td>
<td>+</td>
</tr>
<tr>
<td>22</td>
<td>LN</td>
</tr>
<tr>
<td>23</td>
<td>/</td>
</tr>
<tr>
<td>24</td>
<td>FS? 01</td>
</tr>
<tr>
<td>25</td>
<td>CHS</td>
</tr>
<tr>
<td>26</td>
<td>STOP</td>
</tr>
<tr>
<td>27</td>
<td>GTO 01</td>
</tr>
<tr>
<td>28</td>
<td>END</td>
</tr>
</tbody>
</table>
A Mu Law Encoder

**Inputs:**
- **B** number of bits
- **MU** desired mu-law
- **X** normalized analog input (between -1 & 1)

**Output:**
- digital number (between $-\left(2^{B-1}\right) - 1$ & $2^{B-1} - 1$)
- (if $B=8$ between -127 & 127)

```
01 LBL CODE
02 B=? 24 ABS
03 AVIEW 25 RCL 05
04 STOP 26 *
05 1 27 1
06 - 28 +
07 2 29 LN
08 X<>Y 30 RCL 05
09 Y^X 31 1
10 1 32 +
11 - 33 LN
12 STO 06 34 /
13 MU=? 35 RCL 06
14 AVIEW 36 *
15 STOP 37 FS? 01
16 STO 05 38 CHS
17 LBL 01 39 FIX 0
18 CF 01 40 RND
19 X=? 41 STOP
20 AVIEW 42 GTO 01
21 STOP 43 END
22 X<0?
```
A Mu-Law Decoder

**Inputs:**
- **B** number of bits
- **MU** desired mu-law
- **Y** binary number
  (between \(-(2^{(B-1)}-1)\) & \((2^{(B-1)}-1)\)
  (if B=8 between -127 & 127)

**Output:** **X** normalized analog output
  (between -1 & 1)

---

01  LBL DECODE  23  SF 01
02  B=?  24  ABS
03  AVIEW  25  RCL 05
04  STOP  26  1
05  1  27  +
06  -  28  LN
07  2  29  *
08  X<>Y  30  RCL 06
09  Y^X  31  /
10  1  32  E^X
11  -  33  1
12  STO 06  34  -
13  MU=?  35  RCL 05
14  AVIEW  36  /
15  STOP  37  FS? 01
16  STO 05  38  CHS
17  LBL 01  39  STOP
18  CF 01  40  GTO 01
19  Y=?  41  END
20  AVIEW
21  STOP
22  X<0?
Program the Translate ROM

This program takes n-bit mu-law companded data and translates it to a new mu-law with a provision for increasing the maximum input value.

Inputs:

- **NO OF SPKRS**: max. no. of voices to be conferenced
- **INTO=1, OUT=0**: into implies toward the summer, out implies away from the summer
- **B**: number of bits
- **MU1**: mu into translator, if INTO=1
- **MU2**: mu out of translator, if INTO=1
- **Y,MU1**: B bit digital word into translator

Outputs:

- **Y,MU2**: B bit digital word out of translator

---

```
01 LBL XLATE
02 CF 03
03 NO. OF SPKRS= 27 LBL 01
04 AVIEW
05 STOP
06 INTO=1, OUT=0 30 AVIEW
07 AVIEW
08 STOP
09 X=0? 32 X<0? 56 RCL 10
10 SF 03 33 SF 01 57 *
11 RDN 34 ABS 58 1
12 FS? 03 35 RCL 05 59 +
13 1 36 1 60 LN
14 STO 09 37 + 61 RCL 10
15 B=? 38 LN 62 1
16 AVIEW 39 * 63 +
17 STOP 40 RCL 06 64 LN
18 STO 06 41 1 65 /
19 MU1=? 42 X<>Y 66 RCL 06
20 AVIEW 43 2 67 1
21 STOP 44 X<>Y 68 -
22 STO 05 45 Y^X 69 2
23 MU2=? 46 1 70 X<>Y
24 AVIEW 47 - 71 Y^X
25 STOP 48 / 72 1
26 STO 10
27 E^X 49 1 73 -
50 1 50 1 74 *
51 - 52 RCL 05 75 FS? 01
53 / 76 CHS 77 FIX 0
54 RCL 09 78 CF 29
55 / 79 RND
56 X=0? 80 Y,MU2=
57 * 81 ARCL X 82 AVIEW
58 1 83 STOP
59 + 84 GTO 0
60 LN 85 END
```
Program the Summer ROM

Inputs:  
- B: number of bits  
- MU: desired mu-law  
- Y,1: one of the digital words to be summed  
- Y,2: the other digital word to be summed  

Output: digital companded sum  

<table>
<thead>
<tr>
<th>Line</th>
<th>Instruction(s)</th>
<th>Line</th>
<th>Instruction(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>LBL SUM</td>
<td>31</td>
<td>+</td>
</tr>
<tr>
<td>02</td>
<td>B=?</td>
<td>32</td>
<td>X&lt;0?</td>
</tr>
<tr>
<td>03</td>
<td>AVIEW</td>
<td>33</td>
<td>SF 01</td>
</tr>
<tr>
<td>04</td>
<td>STOP</td>
<td>34</td>
<td>ABS</td>
</tr>
<tr>
<td>05</td>
<td>1</td>
<td>35</td>
<td>RCL 05</td>
</tr>
<tr>
<td>06</td>
<td>-</td>
<td>36</td>
<td>*</td>
</tr>
<tr>
<td>07</td>
<td>2</td>
<td>37</td>
<td>1</td>
</tr>
<tr>
<td>08</td>
<td>X&lt;&gt;Y</td>
<td>38</td>
<td>+</td>
</tr>
<tr>
<td>09</td>
<td>Y^X</td>
<td>39</td>
<td>LN</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>40</td>
<td>RCL 05</td>
</tr>
<tr>
<td>11</td>
<td>-</td>
<td>41</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>STO 06</td>
<td>42</td>
<td>+</td>
</tr>
<tr>
<td>13</td>
<td>MU=?</td>
<td>43</td>
<td>LN</td>
</tr>
<tr>
<td>14</td>
<td>AVIEW</td>
<td>44</td>
<td>/</td>
</tr>
<tr>
<td>15</td>
<td>STOP</td>
<td>45</td>
<td>RCL 06</td>
</tr>
<tr>
<td>16</td>
<td>STO 05</td>
<td>46</td>
<td>*</td>
</tr>
<tr>
<td>17</td>
<td>LBL 01</td>
<td>47</td>
<td>FS? 01</td>
</tr>
<tr>
<td>18</td>
<td>CF 01</td>
<td>48</td>
<td>CHS</td>
</tr>
<tr>
<td>19</td>
<td>Y,1=?</td>
<td>49</td>
<td>FIX 0</td>
</tr>
<tr>
<td>20</td>
<td>AVIEW</td>
<td>50</td>
<td>RND</td>
</tr>
<tr>
<td>21</td>
<td>STOP</td>
<td>51</td>
<td>STOP</td>
</tr>
<tr>
<td>22</td>
<td>STO 09</td>
<td>52</td>
<td>GTO 01</td>
</tr>
<tr>
<td>23</td>
<td>Y,2=?</td>
<td>53</td>
<td>LBL 02</td>
</tr>
<tr>
<td>24</td>
<td>AVIEW</td>
<td>54</td>
<td>CF 02</td>
</tr>
<tr>
<td>25</td>
<td>STOP</td>
<td>55</td>
<td>X&lt;0?</td>
</tr>
<tr>
<td>26</td>
<td>XEQ 02</td>
<td>56</td>
<td>SF 02</td>
</tr>
<tr>
<td>27</td>
<td>STO 10</td>
<td>57</td>
<td>ABS</td>
</tr>
<tr>
<td>28</td>
<td>RCL 09</td>
<td>58</td>
<td>RCL 05</td>
</tr>
<tr>
<td>29</td>
<td>XEQ 02</td>
<td>59</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>RCL 10</td>
<td>60</td>
<td>+</td>
</tr>
</tbody>
</table>

01 LBL SUM 31 + 61 LN
02 B=? 32 X<0? 62 *
03 AVIEW 33 SF 01 63 RCL 06
04 STOP 34 ABS 64 /
05 1 35 RCL 05 65 E^X
06 - 36 * 66 1
07 2 37 1 67 -
08 X<>Y 38 + 68 RCL 05
09 Y^X 39 LN 69 /
10 1 40 RCL 05 70 FS? 02
11 - 41 1 71 CHS
12 STO 06 42 + 72 RTN
13 MU=? 43 LN 73 END
14 AVIEW 44 /
15 STOP 45 RCL 06
16 STO 05 46 *
17 LBL 01 47 FS? 01
18 CF 01 48 CHS
19 Y,1=? 49 FIX 0
20 AVIEW 50 RND
21 STOP 51 STOP
22 STO 09 52 GTO 01
23 Y,2=? 53 LBL 02
24 AVIEW 54 CF 02
25 STOP 55 X<0?
26 XEQ 02 56 SF 02
27 STO 10 57 ABS
28 RCL 09 58 RCL 05
29 XEQ 02 59 1
30 RCL 10 60 +


