

A Unified Formulation of Segment Companding Laws and Synthesis of Codecs and Digital Compandors

By H. KANEKO

(Manuscript received April 2, 1970)

Segment companding laws are being considered for use in PCM to cater to the possibility of digital processing of uniformly quantized signals. A unified formulation of the segment companding laws is presented, which permits the detailed structure of quantization to be explicitly characterized. Most important, it is shown that this formulation yields algorithms for the systematic synthesis of coders, decoders, and digital compandors. Examples of circuits synthesized from the algorithms are shown for two segment law companding families—the μ -law and the A-law.

I. INTRODUCTION

Instantaneous companding of PCM is used to maintain a reasonably constant signal to quantizing noise ratio for a wide dynamic range of the input voice signal. A logarithmic " μ -law" is a typical "smooth" input/output transfer characteristic that has been approximated in the D1 channel bank realized by a diode compandor.^{1,2} In contrast to the smooth law, segment type companding laws, which are piecewise linear approximations to the continuous laws, have also been considered. In early experimental PCM systems a segment law was used primarily to avoid implementation difficulties associated with tracking of diode compandors.^{3,4} However, since the mid-sixties segment laws have been reconsidered from the view point of digital linearization to cater to the possibility of digital processing in an integrated digital network. Two of the segment law families have been stressed—the 13-segment A-law,^{5,6} and the 15-segment μ -law used in the D2 channel bank.⁷

The major advantage of the segment laws resides in the digital linearization feature, through which digital processing such as digital

filtering, conferencing, net loss adjustment, echo suppression, equalization, companding law conversion, and so on, can be performed efficiently.⁸ Also, it has been argued that the segment laws, being well defined laws, should readily permit an international interconnection on a digital basis. However, the defining features of the segment laws have not been clearly drawn.

Many attempts have been made to characterize a particular law by table-look-up, by a set of equations, or by forming algorithms,^{5,8} but important parameters defining the particular law have not been delineated. What is most important, the lack of an explicit expression for the family of segment laws has produced difficulty in the understanding and design of codecs and digital companders.

In this paper a unified formulation is presented, which leads to the systematic synthesis of codecs and digital companders for the binary code.* In the second section, starting from conventional concepts of the μ -law and the A-law, we derive expressions for the decoder output level. Output levels are given in terms of a standard form with segment edge parameter a and centering parameter c . Coder decision levels may also be expressed by the same formulation with the addition of the coding parameter b , thus completely specifying the companding law. This approach avoids table-look-up. Relations between the step size, effective coder input, transfer gain, and other parameters of the law are derived.

In the third section, we show that this formulation is directly related to the codec circuit implementation, and these circuits can be synthesized in a systematic and straightforward manner. Unipolar decoders are first synthesized from the formulation, and the synthesis is extended to bipolar decoders including the sign bit. We also show that the coding parameter b is related to the design of logic in a sequential comparison coder.

Digital companders can also be synthesized by using the same formulation. In the fourth section we display digital expansion and compression algorithms, and compandor circuits synthesized therefrom. At Bell Telephone Laboratories, pioneering studies on digital companders have been made by T. P. Stanley, M. R. Aaron and D. G. Messerschmitt in unpublished memoranda. W. L. Montgomery has also done work in this area.⁸ The synthesis technique developed here serves to unify these separate works, and makes it possible to design the circuits in a systematic way.

* Extension to arbitrary bases follows in a straightforward manner and will not be considered.

II. FORMULATION OF SEGMENT LAWS

2.1 Definition of Variables

Suppose that a digital compressed code X is composed of m binary digits called "characteristic bits" representing the segment number L , and n binary digits called "mantissa bits" representing the quantizing step V in a segment, as shown in Fig. 1. The total number M of segments in one polarity is equal to 2^m , and the total number N of quantizing steps within a segment is equal to 2^n . Then the digital representation of the compressed signal X is expressed in terms of L and V , as

$$X(L, V) = V + N \cdot L \quad (1)$$

or

$$V = X \bmod N$$

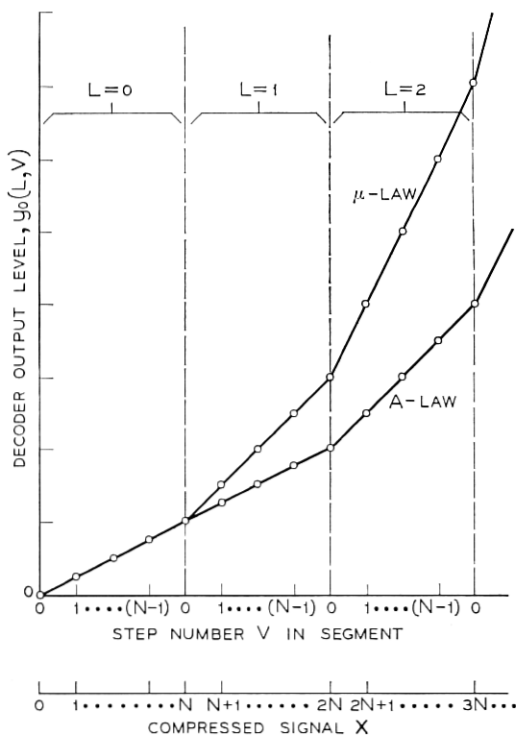


Fig. 1—Segment companding laws—standard form.

where X , L and V are defined in the regions as

$$\left. \begin{aligned} X &= \text{digital compressed signal} \in \{0, 1, \dots, MN - 1\}, \\ L &= \text{segment number} \in \{0, 1, \dots, M - 1\}, \text{ and} \\ V &= \text{step number within a segment} \in \{0, 1, \dots, N - 1\}. \end{aligned} \right\} \quad (2)$$

The definition of X , L and V beginning from "0" permits a consistent algebraic treatment and yields a straightforward circuit design.

If we denote the decoder output level by y_0 , then it is a function of the digital compressed signal and $y_0 = y_0(X) = y_0(L, V)$. The quantizing step size is then given by the difference of two adjacent output levels, and is

$$\left. \begin{aligned} \Delta_0(L) &\equiv y_0(L, V + 1) - y_0(L, V), & V \neq N - 1 \\ \Delta'_0(L) &\equiv y_0(L + 1, 0) - y_0(L, N - 1), & V = N - 1 \end{aligned} \right\} \quad (3)$$

where $\Delta'_0(L)$ is the step size at the segment edge. Within a segment the step size $\Delta_0(L)$ is constant and independent of V .

2.2 Definition of the Segment μ -law

We will define the segment μ -law and derive an expression for an output level as a function of L and V . The "so-called" segment μ -law should satisfy the following:

(i) Except for the segment edges, the ratio of the step sizes of the adjacent segments is equal to 2; that is,

$$\frac{\Delta_0(L + 1)}{\Delta_0(L)} = 2, \quad L \in \{0, 1, \dots, M - 2\}, \quad (4)$$

$$(ii) \quad y_0(0, 0) = c \quad (\text{centering}), \quad (5)$$

$$(iii) \quad \Delta_0(0) = 1 \quad (\text{normalization}). \quad (6)$$

Solving the difference equation (4) subject to equations (3), (5) and (6), we have

$$\begin{aligned} y_0(L, V) &= 2^L(V + N + a) - N - a + c \\ &= f_\mu(L, V + a) - a + c \end{aligned} \quad (7)$$

for an arbitrary real a , where $f_\mu(L, V)$ is the standard form of the segment μ -law, and is

$$f_\mu(L, V) \equiv 2^L(V + N) - N. \quad (8)$$

The standard form is interpreted in the following way. If we take a

continuous variable v such that

$$v \in [0, N]$$

then $f(L, v)$ represents a continuous segment μ -law as shown in Fig. 1. The continuous segment μ -law $f_\mu(L, v)$ is a set of segment lines whose segment edges $f_\mu(L, N) = f_\mu(L + 1, 0)$ satisfy the so called μ -law¹ for $x = LN + v$, namely:

$$\frac{x}{E_1} = \frac{\log(1 + \mu y/E_2)}{\log(1 + \mu)} \quad (9)$$

for normalization constants $E_1 = MN$, $E_2 = N(2^M - 1)$, and $\mu = 2^M - 1$. In other words, the continuous standard form gives an exact "chord type" approximation to the smooth μ -law. On this continuous segment law the discrete point V should be defined as the truncation of v .

The output level (7) is then the modified version of the standard form with parameters a and c , and it constitutes a class of " μ -like" laws. The centering parameter c determines the origin, and the edge effect parameter a describes the discontinuity of step size at the segment edges. The continuous segment law varies with parameter a as shown in Fig. 2, and produces discontinuities at the segment edges.

2.3 Extension to the Segment A-law

To obtain the segment A-law,⁵ we replace the condition (i) by

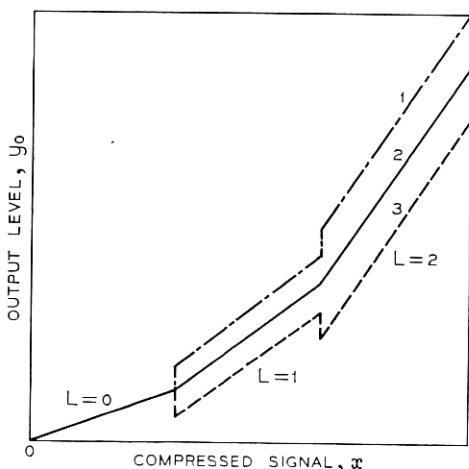


Fig. 2—Effect of segment edge parameter a : Curve 1— $a > 1$; Curve 2— $a = 0$ and Curve 3— $a < 0$.

(i)' Except for the segment edges the ratio of the step size of the adjacent segments is

$$\frac{\Delta_0(L+1)}{\Delta_0(L)} = \begin{cases} 1, & L = 0, \\ 2, & L \in \{1, 2, \dots, M-2\}. \end{cases} \quad (10)$$

This implies that the 0th and 1st segments constitute a long colinear segment and the step size ratio of the adjacent segments elsewhere is equal to 2. Then solving equation (10) subject to equations (3), (5) and (6) we have

$$y_0(L, V) = f_A(L, V+a) - a + c \quad (11)$$

where $f_A(L, V)$ is the standard form of the segment A-law and is

$$f_A(L, V) = \begin{cases} V, & L = 0, \\ 2^{L-1}(V+N), & L \neq 0. \end{cases} \quad (12)$$

Note that equation (11) is identical with equation (7) for the μ -law, and the output level for the A-law is characterized by the standard form $f_A(L, V)$ in equation (12).

As in the μ -law case, where a discrete V is replaced by a continuous variable $v \in [0, N]$ in the standard form $f(L, V)$, a "chord type" approximation to the so called A-law is obtained. The A-law has been defined as^{5,9}

$$\frac{x}{E_3} = \begin{cases} \frac{A(y/E_4)}{1 + \log_K A}, & y \in [0, E_4/A] \\ \frac{1 + \log_K (Ay/E_4)}{1 + \log_K A}, & y \in [E_4/A, E_4] \end{cases} \quad (13)$$

for $K = 2$, $A = 128$,* $E_3 = MN$, and $E_4 = N \cdot 2^M$.

Expression (11) applies for any type of segment companding law if the standard form $f_A(L, v)$ is appropriately defined to yield a continuous piece-wise linear approximation to a continuous curve.

2.4 Decision Levels

In the foregoing we have concentrated on the decoder output level because the level of primary importance in companding is the output level. The coder decision level should then be determined to yield an appropriate range of the input signal corresponding to that output level.

* The parameter values $A = 87.6$, $K = e$ (base of natural logarithms) are often used.⁵ However, to be consistent with a "chord type" approximation, $K = 2$ and $A = 128$ are used here.

First, the decoder output level should be a good representative of the signal range between two adjacent decision levels. It has been shown elsewhere that for a given set of output levels the locally optimum decision levels, which minimize the mean squared error, should lie halfway between two adjacent output levels irrespective of the input signal distribution.¹⁰ This can be achieved, except at the segment edges, by shifting the output level a half step up or down in the compressed signal domain; that is, the decision level is

$$y_d(L, V) \equiv y_o(L, V + b) \quad (14)$$

where b is the coding parameter and takes on the values $+0.5$ or -0.5 . When $b = +0.5$, $y_d(L, V)$ gives the upper bound of the input range corresponding to $X(L, V)$; when $b = -0.5$, it corresponds to the lower bound. The coder decision level y_d implies that an analog input y is encoded into $X(L, V)$ if

$$y_d(X - 1 | b = +0.5) \leq y < y_d(X | b = +0.5) \quad (15)$$

for the upper bound approach ($b = +0.5$), and if

$$y_d(X | b = -0.5) \leq y < y_d(X + 1 | b = -0.5) \quad (16)$$

for the lower bound approach ($b = -0.5$). In general, these two approaches are identical at all points except at segment edges. Now, for any points including segment edges, it can be shown (Appendix A) that the two approaches coincide if and only if the edge effect parameter a is equal to 0.5. As we will see in Section 3.4, the coding parameter b is closely related to the binary coding logic in a sequential comparison coder.

2.5 Codec Transfer Characteristics

The decoder output level and the coder decision level have been defined. Now, the characteristic from the coder input to the decoder output as shown in Fig. 3 can be expressed explicitly by two equations with the auxiliary variables L, V and the parameters a, b and c ; that is,

$$\left. \begin{aligned} y_o(L, V) &= f(L, V + a) - a + c \\ y_d(L, V) &= f(L, V + a + b) - a + c \end{aligned} \right\} \quad (17)$$

where $f(L, V)$ is the standard form defined by equation (8) for the μ -law and equation (12) for the A-law. These equations characterize the precise quantization structures.

The centering parameter c is 0 or 0.5, and when it is 0 the transfer

characteristic at the origin is "mid-tread" as shown in Fig. 3a. When $c = 0.5$, it is "mid-riser" as shown in Fig. 3b.

To show the effect of the parameter a , a few examples of the transfer curves are shown in Fig. 4 for $N = 4$ and mid-tread operation ($c = 0$). As mentioned before, when $a = 0.5$, the transfer curves for $b = +0.5$ and -0.5 are identical except for a shift in the numbering of $X(L, V)$.

2.6 Step Sizes

The step sizes of the output levels and the decision levels are given, except for the segment edges, by

$$\left. \begin{aligned} \Delta_0(L) &= y_0(L, V + 1) - y_0(L, V) \\ \Delta_d(L) &= y_d(L, V + 1) - y_d(L, V) \end{aligned} \right\} V \neq N - 1 \quad (18)$$

and the corresponding step sizes at the segment edges are given by

$$\left. \begin{aligned} \Delta'_0(L) &= y_0(L + 1, 0) - y_0(L, N - 1) \\ \Delta'_d(L) &= y_d(L + 1, 0) - y_d(L, N - 1) \end{aligned} \right\} \quad (19)$$

It is easily shown that for non-edge points

$$\Delta_0(L) = \Delta_d(L) = 2^L \quad (20)$$

for the μ -law. However, for the edge points the step size is given by

$$\left. \begin{aligned} \Delta'_0(L) &= f(L + 1, a) - f(L, N - 1 + a) = \Delta_0(L) \cdot (1 + a) \\ \Delta'_d(L) &= f(L + 1, a + b) - f(L, N - 1 + a + b) \\ &= \Delta_d(L) \cdot (1 + a + b) \end{aligned} \right\} \quad (21)$$

Step sizes at the segment edges are depicted in Fig. 5 with respect to a .

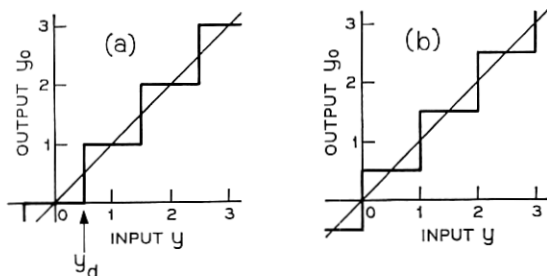


Fig. 3—Effect of centering parameter c , (a) mid-tread ($c = 0$), and (b) mid-riser ($c = 0.5$).

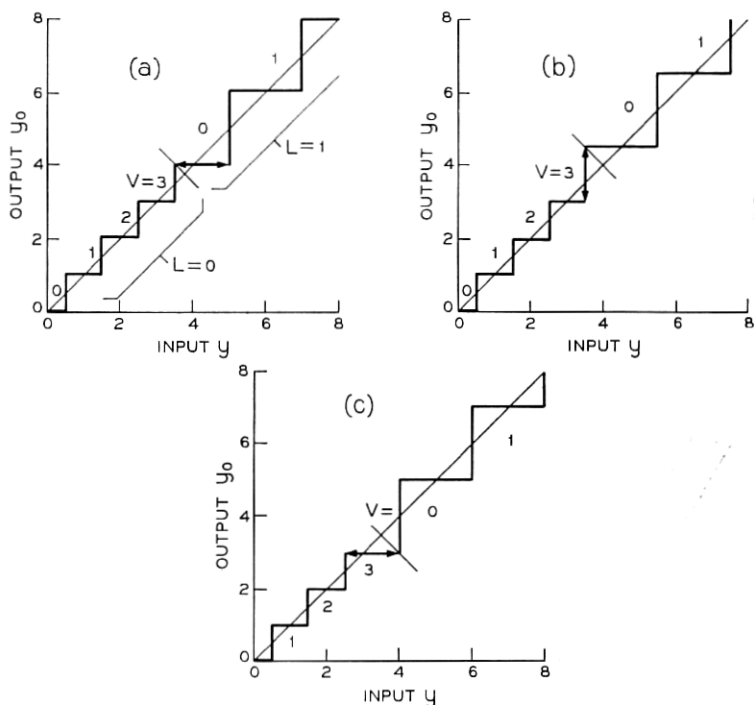


Fig. 4—Transfer staircases with $N = 4$ and mid-tread for (a) RLA ($a = 0, b = 0.5$), (b) DLA ($a = 0.5, b = 0.5$), and (c) RLA ($a = 1, b = -0.5$).

For the A-law, equation (20) is modified to

$$\Delta_0(L) = \Delta_d(L) = \begin{cases} 1, & L = 0, \\ 2^{L-1}, & L \neq 0, \end{cases}$$

and relation (21) applies for the A-law on the understanding that the "segment edges" exclude the colinear segment edge between 0th and 1st.

2.7 Effective Coder Input

Now, we will define an effective coder input $\tilde{y}(L, V)$ such that $\tilde{y}(L, V)$ is the statistical average of the input signal between two adjacent decision levels. If we assume that the input is uniformly distributed between two adjacent decision levels, then $\tilde{y}(L, V)$ is equal to their average as shown below. For $b = +0.5$ (upper bound case)

$$\tilde{y}(L, V) = \begin{cases} \frac{1}{2}[y_d(L, V-1) + y_d(L, V)], & V \neq 0, \\ \frac{1}{2}[y_d(L-1, N-1) + y_d(L, 0)], & V = 0 \text{ (edge)}. \end{cases}$$

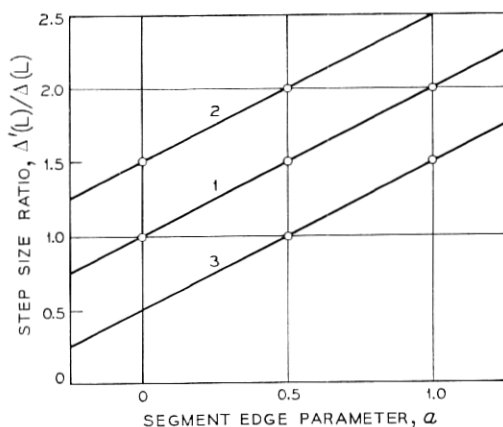


Fig. 5—Step size ratio at the segment edges relative to the lower segment step size: Curve 1—output level; Curve 2—decision level ($b = 0.5$); and Curve 3—decision level ($b = -0.5$).

For $b = -0.5$ (lower bound case)

$$\tilde{y}(L, V) = \begin{cases} \frac{1}{2}[y_d(L, V) + y_d(L, V + 1)], & V \neq N - 1, \\ \frac{1}{2}[y_d(L, N - 1) + y_d(L + 1, 0)], & V = N - 1 \text{ (edge)}. \end{cases}$$

Except for the edge points, we have

$$\tilde{y}(L, V) = y_0(L, V) \quad (22)$$

and this is a direct consequence of our definition of y_d in equation (14). At the segment edges (L_e, V_e), it can be shown that

$$\tilde{y}(L_e, V_e) = y_0(L_e, V_e) - \Delta_0(L) \cdot b \cdot (a - 0.5) \quad (23)$$

where edge point means

$$(L_e, V_e) = \begin{cases} (L + 1, 0) & \text{when } b = +0.5, \\ (L, N - 1) & \text{when } b = -0.5. \end{cases} \quad (24)$$

Therefore, except for the segment edges the transfer gain from the effective coder input to the decoder output is unity. At these segment edges, if we define the tracking error ϵ , which is the deviation from unity of the transfer gain from the effective coder input \tilde{y} to the decoder output y_0 , then ϵ is

$$\epsilon \equiv 1 - \tilde{y}/y_0 = \begin{cases} \frac{\Delta_0(L)}{y_0} \cdot b \cdot (a - 0.5), & \text{at segment edges} \\ 0, & \text{elsewhere.} \end{cases} \quad (25)$$

The ratio $\epsilon/(\Delta_0/y_0)$ is the gain error relative to the step size, and is depicted in Fig. 6, where it is seen that the tracking error is zero over the entire input range if and only if $a = 0.5$.

2.8 Relation Between Parameters

In Section 2.4 it was noted that for a given output level the locally optimal (minimum mean squared error) decision level is half way between adjacent output levels.¹⁰ It can be shown that this optimality condition is satisfied for any X including segment edges, if and only if $a + b = 0.5$, which means $a = 0$ for $b = 0.5$, or $a = 1$ for $b = -0.5$. Since optimal placement is made from the given output level, this is called "Reconstruction (or Output) Level Assignment" (RLA).⁸

On the other hand, for given decision levels, assuming the uniform distribution of the input in each quantization step, the locally optimal (minimum mean squared error) placement of the output levels lies halfway between adjacent decision levels.¹⁰ It can also be proven that this condition is satisfied for any point X including segment edges, if and only if $a = 0.5$, irrespective of $b = \pm 0.5$. Since the optimal approach is based on the given decision levels, this is called "Decision Level Assignment" (DLA).⁸ Proof of the above statements is given in Appendix B.

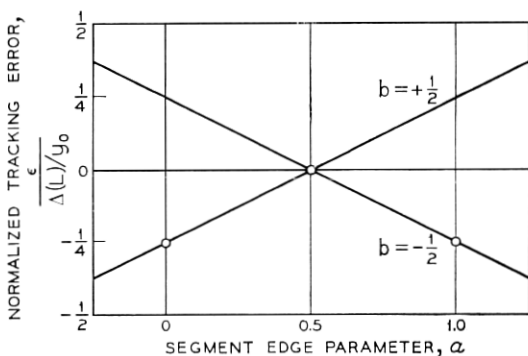


Fig. 6—Tracking error at the segment edges.

As shown previously, when $a = 0.5$ (DLA) the effective tracking error is zero for all inputs, and the upper and the lower bound approaches give identical coding. Furthermore, as will be seen later, the digital compression algorithm and circuits are simpler in DLA than in RLA. The D2 channel bank has been designed according to DLA. Hereafter, our attention is mainly focused on DLA, although the analysis is easily generalized to include other cases.

When $M = 8$, the segment μ -law is generally referred to as the "15-segment μ -law" including negative segments. In contrast, the segment A-law with $M = 8$ is called the "13-segment A-law." In Table I the output levels and the step sizes are listed for the 8-bit (including sign bit) 15-segment μ -law and 13-segment A-law. The maximum and minimum output levels are shown in Table II for ready reference.

Quantizing noise, as calculated for the Laplacian (negative exponential) distributed speech input y_{in} , yields the output signal to quantizing noise ratio shown in Fig. 7. Signal and noise are defined in a band of width equal to half the sampling rate. The average transfer gain $\langle G \rangle$ is chosen so as to minimize the quantizing noise, and is given by $E[y_{in} \cdot y_o] / E[y_{in}^2]$. The idle channel noise, calculated for a zero-mean gaussian input noise, is simply $ICN = E[y_o^2]$ and is shown in Fig. 7. The centering parameter c is important in the small signal range as is

TABLE I—OUTPUT LEVEL STEP SIZES FOR $m = 3$ AND $n = 4$

segment number L	μ -law				A-law			
	non-edge $\Delta_0(L)$	edge points $\Delta_0'(L)$			non-edge $\Delta_0(L)$	edge points $\Delta_0'(L)$		
		$a = 0$	$a = 0.5$	$a = 1$		$a = 0$	$a = 0.5$	$a = 1$
0	1	1	1.5	2	1	1	1	1
1	2	2	3	4	1	1	1.5	2
2	4	4	6	8	2	2	3	4
3	8	8	12	16	4	4	6	8
4	16	16	24	32	8	8	12	16
5	32	32	48	64	16	16	24	32
6	64	64	96	128	32	32	48	64
7	128				64			

TABLE II—MAXIMUM AND MINIMUM OF OUTPUT LEVELS FOR $m = 3$
AND $n = 4$

		mid-tread ($c = 0$)			mid-riser ($c = 0.5$)		
		$a = 0$	$a = 0.5$	$a = 1$	$a = 0$	$a = 0.5$	$a = 1$
μ -law	$y_0(0, 0)$	0	0	0	0.5	0.5	0.5
	$y_0(7, 15)$	3952	4015.5	4079	3952.5	4016	4079.5
A-law	$y_0(0, 0)$	0	0	0	0.5	0.5	0.5
	$y_0(7, 15)$	1984	2015.5	2047	1984.5	2016	2047.5

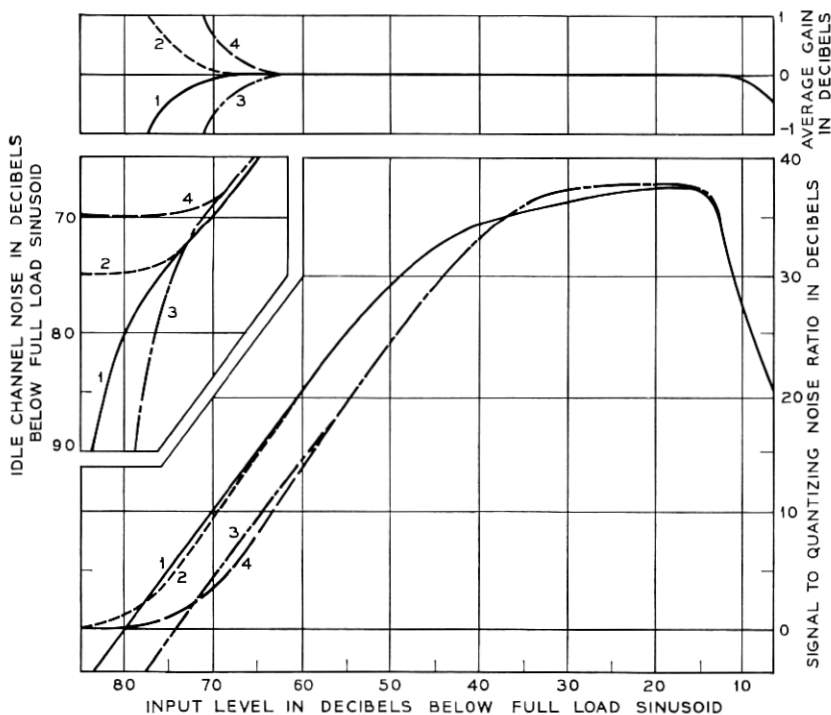


Fig. 7—Average gain, signal-to-noise ratio, and idle channel noise for DLA with $m = 3$ and $n = 4$: Curve 1— μ -law, mid-tread ($c = 0$); Curve 2— μ -law, mid-riser ($c = 0.5$); Curve 3—A-law, mid-tread ($c = 0$); and Curve 4—mid-riser ($c = 0.5$).

well known, and it affects center clipping and idle channel noise. The effect of a on the signal quality is very small. The signal-to-noise ratio relative to the S/N for DLA case (curve 1 in Fig. 7) and the average gain are shown in Figs. 8a and b as curves 1 and 2 for the two RLA cases. The small difference (± 0.14 dB) in S/N in the lower signal range is due to the difference in overload levels. However, if there is a mistracking such that the signal is encoded with $a_1 = 0.5$ and decoded with $a_2 = 1$, the mismatch results in distortion and net loss variation. The penalty in the signal-to-noise ratio and the change in average gain are shown as curve 3 in Figs. 8a and b. Although the penalties are very small, the net loss variation barely meets toll quality requirement, and accumulated impairment in multi-link digital conversions would be intolerable. Therefore, the mixed use of DLA and RLA is to be avoided.

2.9 General Formulation

To summarize this section we give a general expression which is conveniently used hereafter in the synthesis of codecs and digital companders. From equations (7), (8), (12) and (14) the general expression can be written as

$$y(L, V) = \Delta(L) \cdot (V + P) - Q \quad (26)$$

where for the segment μ -laws

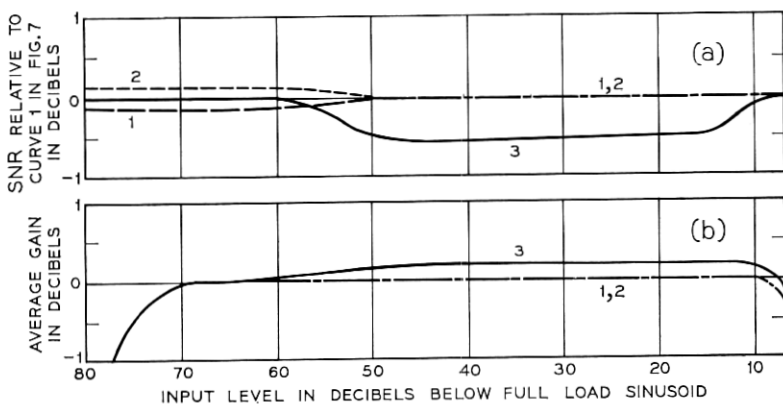


Fig. 8—Effects of parameters on S/N and average gain for μ -law with $m = 3$ and $n = 4$: Curve 1—RLA ($a = 0, b = 0.5$); Curve 2—RLA ($a = 1, b = -0.5$); and Curve 3—mistracking case coded by DLA and decoded by RLA ($a = 1$).

$$\left. \begin{aligned} \Delta(L) &= \Delta_0(L) = \Delta_d(L) = 2^L, \\ P &= N + a + b, \\ Q &= N + a - c. \end{aligned} \right\} \quad (27)$$

Equation (26) can represent either the output level by equating b to 0 or the decision level with $b = \pm 0.5$. For the segment A-laws, expression (27) is replaced by

$$\left. \begin{aligned} \Delta(L) &= L - \eta, \\ P &= N \cdot \eta + a + b, \\ Q &= a - c, \end{aligned} \right\} \quad (28)$$

where

$$\eta = \begin{cases} 0, & L = 0, \\ 1, & L \neq 0. \end{cases}$$

III. SYNTHESIS OF CODECS

3.1 Synthesis of Unipolar Decoder

In this section, we show that the representation of the output level and the decision level by equation (26) leads directly to the implementation of the decoder. In the following, attention will be focused mainly on the segment μ -law. Extension to the A-law is readily accomplished by similar procedures.

Equation (26) given by

$$y(L, V) = \Delta(L) \cdot (V + P) - Q$$

is clearly interpreted as adding bias P to V , amplifying it by $\Delta(L)$, and subtracting bias Q from this result, thus obtaining the decoder output $y(L, V)$.

Let us represent the nonlinear code X by an $(m + n)$ bit binary sequence (except for the sign bit e_0), $e_i \in \{0, 1\}$,

$$\begin{aligned} X &= X\{e_1, e_2, \dots, e_m, e_{m+1}, \dots, e_{m+n}\} \\ &= \sum_{i=1}^{m+n} 2^{m+n-i} \cdot e_i. \end{aligned} \quad (29)$$

Then,

$$\left. \begin{aligned} L &= L\{e_1, e_2, \dots, e_m\} = \sum_{i=1}^m 2^{m-i} e_i, \\ V &= V\{e_{m+1}, \dots, e_{m+n}\} = \sum_{i=1}^n 2^{n-i} e_{m+i}. \end{aligned} \right\} \quad (30)$$

From equation (27), $\Delta(L)$ can be written as

$$\Delta(L) = 2^L = \prod_{i=1}^m \{2^{(2^{m-i})}\}^{e_i} \quad (31)$$

and is realized by a cascade of switched amplifiers (or attenuators) whose gain is switched between $2^{(2^{m-i})}$ and 1 according to the digit $e_i = 1$ or 0, respectively. Likewise, V in equation (30) is obtained directly from a uniform decoder of any of the known forms¹¹ operated by $\{e_{m+1}, e_{m+2}, \dots, e_{m+n}\}$.

An example of the μ -law decoder is shown in Fig. 9a. Biases P and Q are given by $(N + a + b)$ and $(N + a - c)$, respectively. The decoder output is either the decoder output level $y_0(L, V)$ by replacing $b = 0$ in P , or the coder decision level $y_d(L, V)$ by choosing $b = \pm 0.5$.

The decoder circuit for the segment A-law is synthesized in the same way using equation (28), and is shown in Fig. 9b.

3.2 Consideration of the Sign Bit

In the preceding, the sign bit e_0 has been excluded from consideration. In practical cases it is desirable to use a bipolar decoder to cover the whole positive and negative signal ranges. We will give alternative representations of the decoder output in terms of the signed input codes. First, assume that the output level is symmetric with respect to the 0-level line.* Suppose that the sign bit e_0 is equal to 0 for the positive signal region, and 1 for negative region. If we define

$$\alpha_0 = 1 - 2e_0 \in \{+1, -1\}, \quad (32)$$

then from equation (26) the bipolar decoder output is given by

$$y(L, V) = \alpha_0 \{\Delta(L) \cdot (V + P) - Q\} \quad (33)$$

$$= \Delta(L) \{\alpha_0 V + \alpha_0 P\} - \alpha_0 Q. \quad (34)$$

If we look at the binary representation of $\alpha_0 V = \{e_0, e_{m+1}, e_{m+2}, \dots, e_{m+n}\}$ as in equation (30), the so-called "sign and magnitude"

* This implies that in the mid-tread case the "0" output level is shared between the positive "0" and the negative "0".

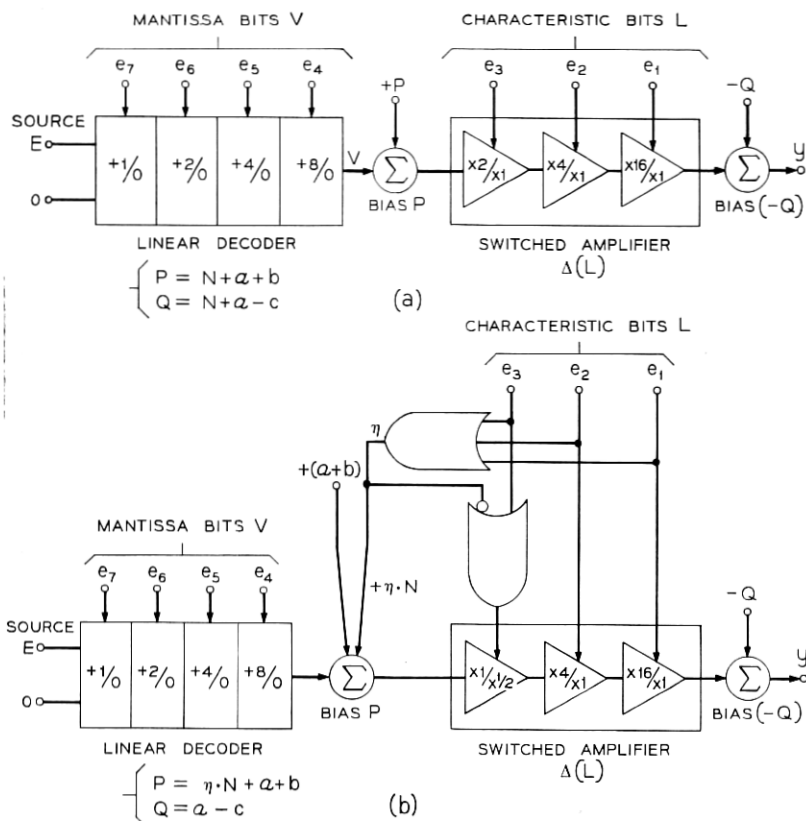


Fig. 9—Synthesis of segment law decoders ($m = 3, n = 4$): (a) 15-segment μ -law and (b) 13-segment A-law.

or "folded binary" representation results. Another representation often used in computer arithmetic is the "1's complement" code.¹² If we denote the 1's complement code by

$$V' = V' \{e'_{m+1}, e'_{m+2}, \dots, e'_{m+n}\} = \sum_{i=1}^n 2^{n-i} e'_{m+i},$$

then V' is related to positive V by

$$\alpha_0 V = V' - e_0(N - 1) \quad (35)$$

and also

$$e'_{m+i} = e_{m+i} \oplus e_0 \text{ mod } 2. \quad (36)$$

Therefore, equation (34) can be expressed in terms of V' and

$$y(L, V) = \Delta(L)\{V' - e_0(N - 1) + \alpha_0 P\} - \alpha_0 Q. \quad (37)$$

Note that the modification in the representation applies only to the mantissa bits V , and the characteristics bits L should remain unchanged.

3.3 Synthesis of Bipolar Decoder

We can synthesize the bipolar decoder circuits according to the two alternative representations (33) and (37).*

(i) Sign-and-magnitude (folded binary)—The "sign and magnitude" decoder according to equation (35) is obtained by inverting the decoder output of the unipolar decoder, or by inverting all the supply voltages to the unipolar decoder as shown in Fig. 10a. The former requires a precise analog inverter. In the latter circuit, the source switch must be in serial with the decoder switch, and stringent requirements arise due to the wide dynamic range of these analog switches.

(ii) 1's complement—Suppose that we use a bipolar source $(+E, -E)$ instead of unipolar source. $(+E, 0)$ By converting the unipolar code to a bipolar code, we have

$$\alpha'_{m+i} = 2e'_{m+i} - 1 \in \{-1, +1\}, \quad (38)$$

and also using equation (32), equation (37) becomes

$$y(L, V) = \Delta(L) \left[\sum_{i=1}^n 2^{n-i-1} \cdot \alpha'_{m+i} + \alpha_0 \left(P + \frac{N-1}{2} \right) \right] - \alpha_0 Q \quad (39)$$

where $\Delta(L)$, P and Q are given by equation (27). This leads to the decoder circuit shown in Fig. 10b. The weighting currents of the decoder are given by the coefficients of the α 's in equation (39). The weighted current $[P + (N - 1)/2]$ that is switched by α_0 is about six times greater than the most significant digit weight ($= 2^{n-2}$) of the linear decoder. Therefore, the problem of the serial switch in Fig. 10a is replaced by a threefold increase in the accuracy requirements of the weighting network here. This appears to be a worthwhile trade in the practical design of codecs.

3.4 Interpretation of Coding Parameter b

As discussed in Section 2.4, the coder decision level y_a is related to the decoder output level y_0 by

$$y_a(L, V) = y_0(L, V + b)$$

* We have derived another representation for the "2's complement code" but it is omitted because of its increased complexity over the other two representations.

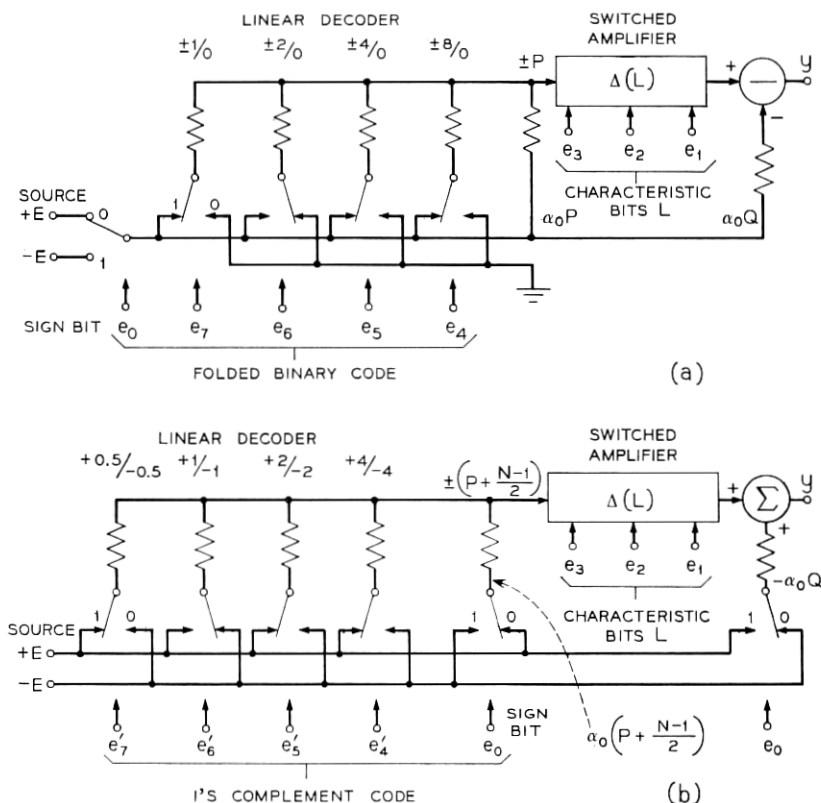


Fig. 10—Synthesis of bipolar decoders operated by (a) folded binary code and (b) 1's complement code.

for $b = \pm 0.5$. When $b = +0.5$ the decision level is an upper-bound for the corresponding output level, and when $b = -0.5$ it is a lower bound, in the sense given by inequalities (15) and (16).

In the following we show how the lower or upper bound approach is related through the parameter b to the logic used in implementing a sequential comparison coder.

Referring to Fig. 11, for the lower bound approach ($b = -0.5$) (from the "all reset" state "000"), the first digit trial "1" is made; and the decision level corresponding to "100" is compared with the input. As a result, if the error ϵ defined as

$$\epsilon = \text{input} - \text{local decoded signal}$$

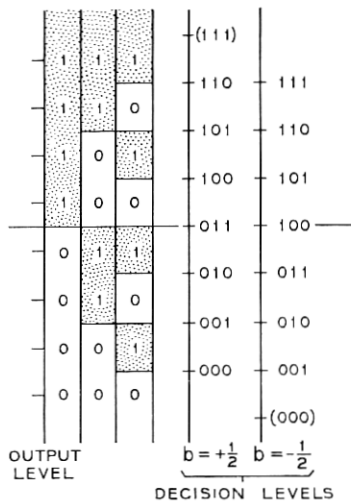


Fig. 11—Upper bound and lower bound decision levels.

is negative the trial "1" is "reset" to "0", and remains unchanged if ϵ is positive. This procedure is repeated digit by digit from the MSD to LSD, and the coding is performed sequentially so as to minimize ϵ .

On the other hand, for the upper bound approach ($b = +0.5$), the "all reset" is made to "1", the "trial" is made to "0", and the "reset" is made to "0" if $\epsilon > 0$. Therefore, if we take a package of "all reset-trial-conditional reset", it is "0-1-0" for the upper bound approach, and "1-0-1" for the lower bound approach. From this package the circuit design of sequential coding logic and registers follows immediately.

3.5 Synthesis of Sequential Comparison Coder

From the result of Section 3.4, and using the unipolar decoder described in Section 3.1, the synthesis of a unipolar coder is straightforward. Here, we treat only the bipolar coder. It has been shown in Section 3.3 that the bipolar decoder in 1's complement form has an advantage in circuit design. If the upper bound approach is taken ($b = +0.5$), the "all reset-trial-conditional reset" package is "0-1-0" for a positive signal. For the negative half if $b = +0.5$ the package becomes "1-0-1" because of the 1's complement (inverted) representation. However, the logic package can be uniformly "0-1-0" for the entire range, if we take $b = +0.5$ for positive inputs and $b = -0.5$

for the negative region. Figure 12 shows the schematic of a bipolar sequential comparison coder based on the 1's complement code. The local decoder, which must have the nonsymmetric feature in b , is obtained from the 1's complement decoder of Fig. 10b with $b = 0$. In addition, a constant bias $b = +0.5$ is added, externally. Then, the register and logic are operated on the basis of the 1's complement code, and the coding is performed with the logic package of "0-1-0." The segment characteristic bit should always be a "folded binary code", which is simply the inverse of the 1's complement code. Of course, there are many other alternatives for building a coder, and the synthesis of any of those immediately follows from our formulation.

IV. SYNTHESIS OF DIGITAL COMPANDORS

In Section III we considered the conversion from analogue signal to digital compressed code or the converse. Here, we consider the conversion between the compressed code and the linear code. The conversion from the compressed code to the linear code is called "digital expansion" and the converse is called "digital compression." In Sections 4.1 and 4.2 the digital expander synthesis is treated, and after that the digital compressor is studied.

4.1 Digital Expansion Algorithm

Suppose that a nonlinear code $X(L, V)$ with bit length $(m + n)$ is digitally linearized to Y . The linearized code Y should correspond

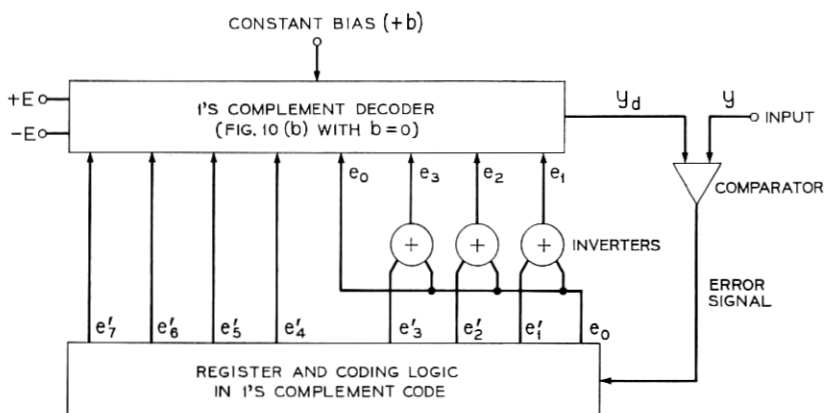


Fig. 12—Bipolar sequential comparison coder with 1's complement code logic.

to the decoder output level

$$Y = y_0(L, V) = f(L, V + a) - a + c. \quad (40)$$

Suppose that the companding law is symmetric for positive and negative signals. We generally exclude the sign bit from consideration. In the case of RLA and mid-tread ($c = 0$), Y is an integer and is expressed by a linear code with length $(2^m + n)$ bits. In the cases of DLA ($a = 0.5$) or the mid-riser ($c = 0.5$), Y is expressed by $(2^m + n + 1)$ bits including one fractional digit. For example the D2 code ($a = 0.5, c = 0, m = 3$, and $n = 4$) is basically* expressed in terms of 13-bit binary code without the sign bit.

The expanded signal Y should correspond to the output level. Therefore, from equations (26) and (27) with $b = 0$, Y is given by

$$Y = \Delta(L) \cdot (V + P) - Q \quad (41)$$

where $\Delta(L) = 2^L$ is a shifting operator, by which the binary sequence $(V + P)$ is shifted to the left by L bits. Therefore, the algorithm for code conversion from $X(L, V)$ to Y is

Algorithm 1 (expansion): For given $X(L, V)$, Y is obtained such that

- (i) Add P to V ,
- (ii) Shift $(V + P)$ by L bits to the left, and
- (iii) Subtract Q ,

where $P = N + a$ and $Q = N + a - c$. All variables and constants are assumed to be in binary representation.

4.2 Synthesis of Digital Expander

The above algorithm leads directly to a parallel expander circuit as shown in Fig. 13. The input L is written into a binary counter BC, and the input V is written into a shift register SR through a parallel adder with addend $(N + a)$. Then, the binary counter BC counts down to "000", and stops. This interval is equal to L . The SR is shifted by the same clock pulses, and L shifts to the left is performed. Then, the entry of SR is read out through a parallel subtractor with subtrahend $(N + a - c)$, thus yielding the parallel expanded output Y .

In practice, a digital expander is generally followed by a digital processor, in which serial arithmetic logic is most conveniently used when the signal sequence begins with the LSD (Least Significant Digit).^{12,13} In the sequel, we concentrate on the serial implementation.

* Except for bit robbing for signalling and framing.⁷

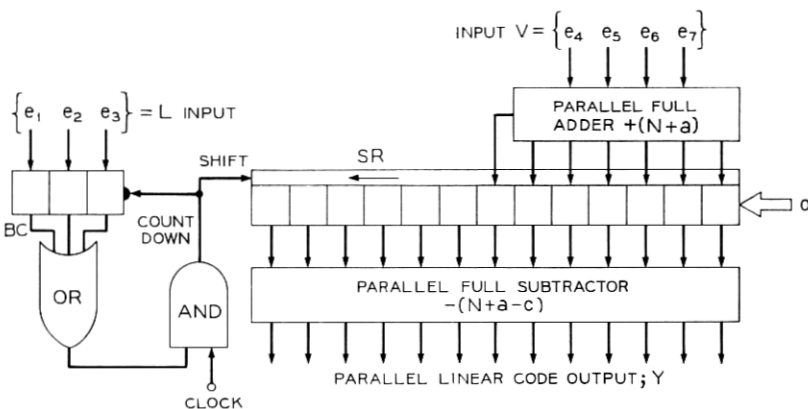


Fig. 13—Parallel digital expander.

If we replace ($\times 2$) by an operator z , which at the same time represents the delay operator by one clock interval, then Y is given by

$$Y(z) = z^L \cdot \{V(z) + P(z)\} - Q(z) \quad (42)$$

where $V(z)$, $P(z)$ and $Q(z)$ are binary sequences beginning from the LSD. Therefore, a general form of digital expander can be described schematically as shown in Fig. 14.

Let us consider more specifically the μ -law expander design. Using the z -transform notation, equation (42) can be written as

$$\begin{aligned} Y(z) &= z^L \{V(z) + z^n + a(z)\} - \{z^n + a(z) - c(z)\} \\ &= z^L \{V(z) + z^n\} + a(z) \cdot (z^L - 1) + c(z) - z^n \end{aligned} \quad (43)$$

where (\pm) implies addition/subtraction with carry, if necessary. Assuming DLA [$a(z) = z^{-1}$], mid-tread ($c = 0$), $m = 3$ and $n = 4$, we have

$$\begin{aligned} zY(z) &= z^L [zV(z) + zP(z)] - zQ(z) \\ &= z^L (z^0 + e_7z + e_6z^2 + e_5z^3 + e_4z^4 + z^5) - z^0 - z^5. \end{aligned}$$

This can further be written in the form

$$zY(z) = z^L (e_7z + e_6z^2 + e_5z^3 + e_4z^4 + z^5) + (z^L - 1) - z^5. \quad (44)$$

From this a serial expander circuit follows as shown in Fig. 15. The inputs L and V are written into a binary counter BC and a shift register SR, respectively. Also "1" is written in the last stage of SR, and the

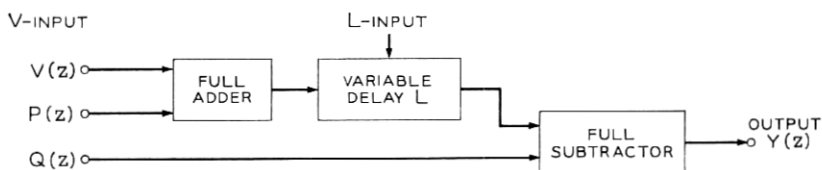
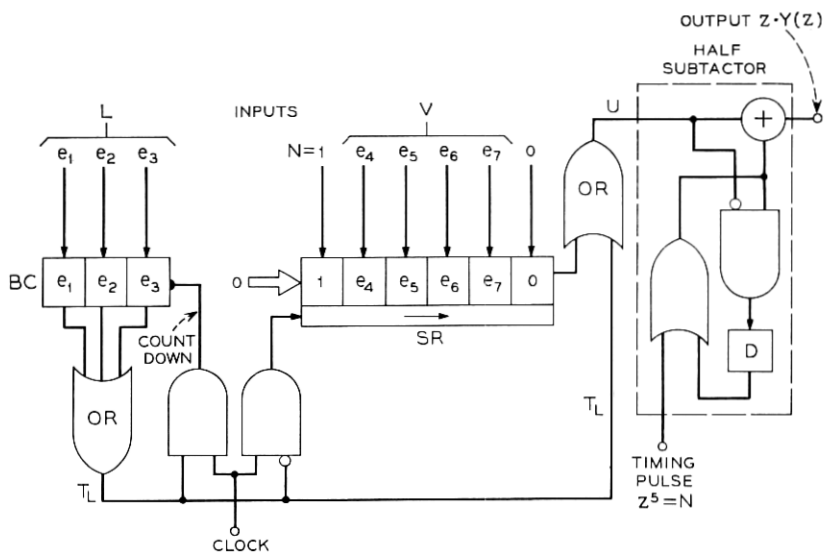


Fig. 14—Basic serial digital expander.



CLOCK	0	1	2	3	4	5	6	7	8	9	10	11	12
WRITE IN													
U	L=0	0	e ₇	e ₆	e ₅	e ₄	1	0	0	0	0	0	0
	1	1	0	e ₇	e ₆	e ₅	e ₄	1	0	0	0	0	0
	2	1	1	0	e ₇	e ₆	e ₅	e ₄	1	0	0	0	0
	3	1	1	1	0	e ₇	e ₆	e ₅	e ₄	1	0	0	0
	4	1	1	1	1	0	e ₇	e ₆	e ₅	e ₄	1	0	0
	5	1	1	1	1	1	0	e ₇	e ₆	e ₅	e ₄	1	0
	6	1	1	1	1	1	1	0	e ₇	e ₆	e ₅	e ₄	1
	7	1	1	1	1	1	1	1	0	e ₇	e ₆	e ₅	e ₄
SUBTRAHEND: Z ⁵													

Fig. 15—Fifteen-segment μ -law expander with DLA ($\alpha = 0.5$), mid-tread ($c = 0$), $m = 3$, and $n = 4$.

initial entry of SR, which is equal to $(N + V)$, corresponds to the first term in equation (44) without delay z^L . The binary counter BC counts down until it becomes "000" and stops, thus giving a gating signal T_L with length L . Instead of having a separate variable delay as in Fig. 14, SR is designed to wait for L intervals before shifting to the right. This gives a delay corresponding to z^L , and the output of the shift register U , which is equal to the first term in equation (44), is depicted as shown in the timing chart of Fig. 15. Next, notice that the second term $(z^L - 1)$ in equation (44) is a sequence of successive 1's with length L . It is exactly the same as the gating signal T_L . Since T_L does not overlap with the first term U , it can be simply added through an OR circuit to the output of SR. The last term z^5 , which is nothing but a timing pulse at the 5th time slot, is subtracted through a half subtractor. Thus, the output LSD-first sequence $z \cdot Y(z)$ is obtained immediately after the write-in of the input.

4.3 Digital Compression Algorithm

In this section, we consider the conversion from a linear code Y to a nonlinear code X . Suppose that a linear code Y is a positive real number representing a signal amplitude normalized by the minimum quantizing step of the compressed code. This code Y is not necessarily an integer, and in general it has q fractional digits in binary representation. This corresponds to the common practice to have more digits in a digital processor than in the input/output digits to reduce the effect of rounding or truncation in arithmetic units (such as digital filters).

As described in Section 2.4 nonlinear coding from an analog signal y to a compressed code $X(L, V)$ is performed by comparing the input with the decision levels. Similarly, in equations (15) and (16), on replacing the analog input y by a discrete linear code Y , we have the following basic digital compression algorithm.

Algorithm 2 (compression: basic):

(i) Upper bound approach—Choose \hat{X} such that

$$y_d(\hat{X} - 1 | b = +0.5) \leq Y < y_d(\hat{X} | b = +0.5). \quad (45)$$

(ii) Lower bound approach—Choose \hat{X} such that

$$y_d(\hat{X} | b = -0.5) \leq Y < y_d(\hat{X} + 1 | b = -0.5), \quad (46)$$

where y_d is exactly the decision level defined by equation (14).

It has been shown that both approaches yield identical outputs

if and only if $a = 0.5$ (DLA). Also note that there is an equality only in the lower boundaries. There is the alternative algorithm in which the equality in equations (45) or (46) is given to the upper boundaries. As compared with the coding from analog input y described in Section 2.4, Algorithm 2 is equivalent to analog coding with input $y = Y - 2^{-q-1}$, and the alternative upper boundary algorithm is equivalent to analog coding with $y = Y + 2^{-q-1}$. Both algorithms are asymptotically equivalent to analog coding for large q . We use the former since it is more directly related to circuit implementation.

Though this algorithm is basic, it is not convenient for synthesizing the circuit. We will divide the above algorithm into the \hat{L} -search process and the \hat{V} -determination processes. Let us introduce a new variable W such that

$$W(\hat{L}, Y) = \{\Delta(L)\}^{-1} \cdot (Y + Q) \quad (47)$$

where $Q = N + a - c$. Then, for the DLA the compression algorithm is given by

Algorithm 3 (compression: DLA): In the case of DLA ($a = 0.5$), the conversion is performed as follows: choose \hat{L} such that*

$$W(\hat{L}, Y) \in [N, 2N) \quad (48)$$

determine \hat{V} such that

$$\hat{V} = \mathfrak{J}[W(\hat{L}, Y)] - N. \quad (49)$$

Here, $\mathfrak{J}[\cdot]$ is the truncation function and defined as

$$\begin{aligned} \mathfrak{J}[x] &\equiv \text{truncation of } x \\ &= \text{integer } I \text{ such that } I \leq x < I + 1. \end{aligned} \quad (50)$$

We also define the rounding function

$$\begin{aligned} \mathfrak{R}[x] &\equiv \text{rounding of } x \\ &= \mathfrak{J}[x + 0.5]. \end{aligned} \quad (51)$$

The derivation of this algorithm is given in Appendix C. If we take the inverse of

$$Y = \Delta(L)(V + P) - Q$$

with respect to V , then

$$V = \{\Delta(L)\}^{-1} \cdot (Y + Q) - P.$$

* Semi-closed interval $[N, 2N)$ means $N \leq W < 2N$.

The first term is exactly $W(L, Y)$ and $P = N + a = N + 0.5$ in DLA, and

$$V = W(L, Y) - N - 0.5.$$

By taking the rounding of V and using equation (51), we have

$$\begin{aligned}\mathcal{R}[V] &= \mathcal{R}[W(L, Y) - 0.5] - N \\ &= \mathcal{J}[W(L, Y)] - N.\end{aligned}$$

This is the same as \hat{V} in the algorithm. Therefore, the physical implication of the algorithm is that \hat{V} is obtained by *simply rounding the transformed variable* V .

For the Reconstruction Level Assignment ($a = 0$ for $b = +0.5$, or $a = 1$ for $b = -0.5$) the algorithm is more complicated,¹⁴ and is *Algorithm 4* (compression: RLA): For the RLA, conversion is accomplished in the following way: choose \hat{L}_0 such that

$$W(\hat{L}_0, Y) \in [N, 2N), \quad (52)$$

determine \hat{V}_0 such that

$$\hat{V}_0 = \mathcal{J}[W(\hat{L}_0, Y)] - N \quad (53)$$

and correct by

$$X(\hat{L}, \hat{V}) = X_0(\hat{L}_0, Y_0) + \theta \quad (54)$$

where the corrector θ is given by

$$\theta = W_{of} - a \quad (55)$$

for W_{of} , the highest fractional binary digit of $W(\hat{L}_0, Y)$.

The proof is complicated, and so is the circuit, and both are omitted because of less interest in RLA.

4.4 Synthesis of Digital Compressor

We will synthesize a digital compressor circuit according to the above algorithms. For simplicity DLA is assumed. Also assume that an input Y is a binary sequence beginning with LSD. In Algorithm 2 the L -search process by equation (48) is accomplished, first by forming a binary sequence $(Y + Q)$, and then shifting bit by bit to the right until equation (48) is satisfied. The criterion (48) implies that the n th digit of W is "1", and the digits of W higher than the n th digit are all "0".

Based on this algorithm an example of the μ -law digital compressor

for DLA, $m = 3$ and $n = 4$ is synthesized as shown in Fig. 16. The input $Y(z)$, consisting of 12 integer digits and q fractional digits, is fed into a serial adder and $Q = (N + a - c)$ is added. Then, the output of the adder is written into the first shift register SR_1 and then to the second shift register SR_2 . Let us denote the time instant by t_0 when

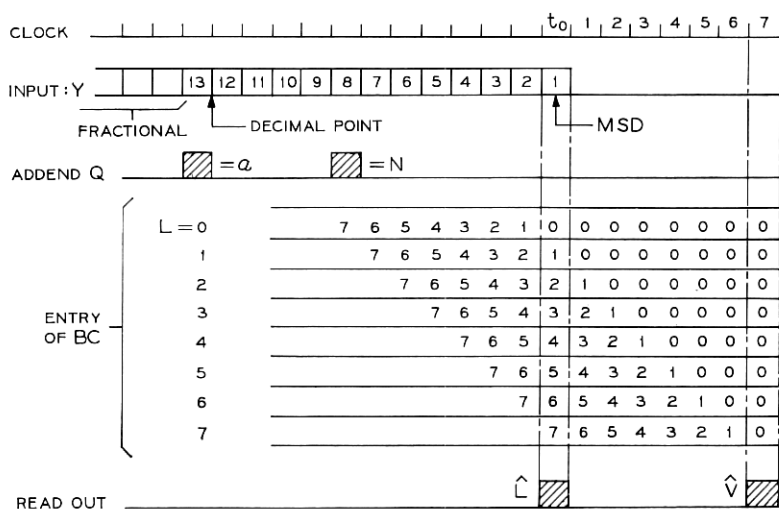
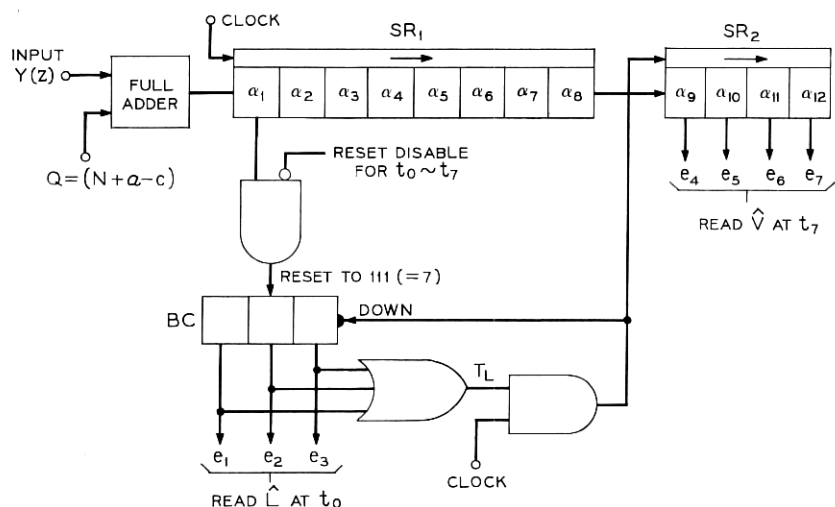


Fig. 16—Fifteen-segment μ -law compressor with DLA ($a = 0.5$), mid-tread ($c = 0$), $m = 3$, and $n = 4$.

the MSD of the input sequence is just stored in the first cell α_1 . While the write-in proceeds, the counter BC, which is being counted down by the clock, is reset to the "111" state whenever the entry of the first cell α_1 is 1. Therefore, at the instant t_0 the BC entry is exactly \hat{L} , and it is read out at t_0 . This also means that the highest order "1" is located at the cell $\alpha_{s-\hat{L}}$, and an additional \hat{L} shifts forces the last entry α_s of SR_1 to be "1". This is accomplished by shifting SR_1 and SR_2 until the entry of BC becomes "000" at time $t_{\hat{L}}$.

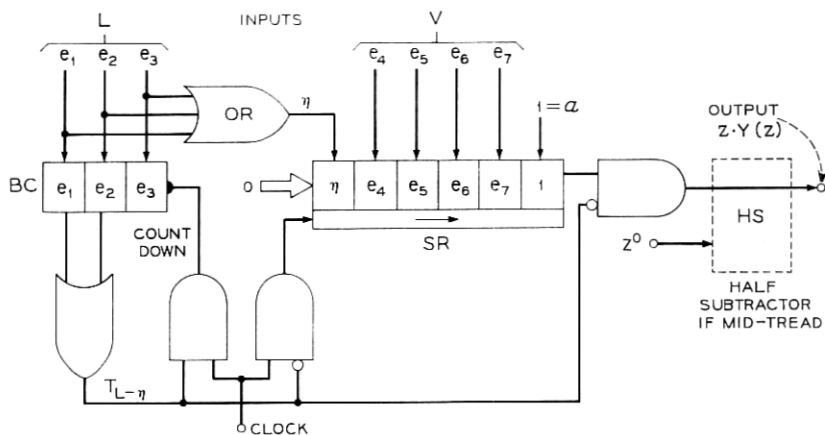
From Algorithm 3 the mantissa output \hat{V} is given by $3[W] - N$. Circuit-wise, $3[W]$ amounts to discarding the smaller digits below the last stage α_{12} of SR_2 , and $(-N)$ means discarding $\alpha_s = 1$. Thus, \hat{V} is obtained from the cells ($\alpha_9, \alpha_{10}, \alpha_{11}, \alpha_{12}$) of SR_2 , which is being stopped after \hat{L} shifts. The \hat{L} output has been read out at time t_0 from BC, thus completing the conversion.

For mid-riser formulation, the circuit can be modified by changing the addend $Q = N + a - c$. A compressor for RLA is also achieved according to Algorithm 4 by adding a correction circuit by θ after the circuit of Fig. 16. The correction circuit may be a parallel adder or a 7-bit binary counter, and obviously the RLA compressor is more complicated than the DLA one.

Among many alternative circuits, a notable feature of the circuit in Fig. 16 is that it can process the input sequence in "real time." While the conversion for a code word is going on, the next code word is being shifted into the SR_1 to prepare for the conversion of the next code word. One might argue that if an input to SR_1 is a sequence beginning from MSD the shift is made from the right to the left simply to yield equation (48). However, doing so one would need additional memory to convert the LSD sequence to the MSD sequence before SR_1 .

4.5 Extension to the Segment A-law

The synthesis procedure can be extended to the segment A-law using expression (28) instead of (27). The expansion algorithm (corresponding to Algorithm 1) for the A-law is obtained by replacing $P = N \cdot \eta + a$, $Q = a - c$ and the L -shifts by $(L - \eta)$ shifts, where η is equal to 0 for $L = 0$ and 1 otherwise. Following the same procedure, a digital expander circuit can be synthesized as shown in Fig. 17, for DLA, mid-riser, $m = 3$ and $n = 4$. The A-law expander differs from the μ -law expander in Fig. 15 in several respects. The input to the first cell is " η ", which is obtained from an OR circuit operating on e_1, e_2 , and e_3 . The input to the last cell of SR is "1". The gating pulse $T_{L-\eta}$ with length $(L - \eta)$ is obtained by OR operation on e_1



	CLOCK	0	1	2	3	4	5	6	7	8	9	10	11
WRITE-IN													
OUTPUT Z·Y(z)	L=0	1	e ₇	e ₆	e ₅	e ₄	0	0	0	0	0	0	0
	1	1	e ₇	e ₆	e ₅	e ₄	1	0	0	0	0	0	0
	2	0	1	e ₇	e ₆	e ₅	e ₄	1	0	0	0	0	0
	3	0	0	1	e ₇	e ₆	e ₅	e ₄	1	0	0	0	0
	4	0	0	0	1	e ₇	e ₆	e ₅	e ₄	1	0	0	0
	5	0	0	0	0	1	e ₇	e ₆	e ₅	e ₄	1	0	0
	6	0	0	0	0	0	1	e ₇	e ₆	e ₅	e ₄	1	0
	7	0	0	0	0	0	0	1	e ₇	e ₆	e ₅	e ₄	1

Fig. 17—Thirteen-segment A-law digital expander with DLA ($a = 0.5$), mid-riser ($c = 0.5$), $m = 3$, and $n = 4$.

and e_2 , and inhibits SR shifting and the SR output. No adder is needed since $Q = 0$.

The compression algorithm (corresponding to Algorithm 3) for the A-law with DLA is obtained by replacing expressions (48) and (49) by

$$W(\hat{L}, Y) \in [\eta N, (1 + \eta)N]$$

and

$$\hat{V} = \mathfrak{J}[W(\hat{L}, Y)] - \eta \cdot N.$$

This algorithm leads to an A-law compressor circuit shown in Fig. 18. The overall performance of this compressor is similar to the μ -law

circuit in Fig. 16. We will indicate only the differences between them. The number of cells in SR_1 is 7. There is no adder in the input since $Q = 0$. The gating pulse $T_{L-\eta}$ is obtained by the same circuit as in the expander (Fig. 15). To facilitate the 0th segment output $\hat{L} = 0$, a reset pulse t_{-7} is added to BC.

V. SUMMARY

A unified formulation of segment type companding laws has been derived from which the detailed quantization structure of codecs can be explicitly represented. The decoder output levels and the coder decision levels are expressed in terms of standard forms and parameters a , b , and c . The standard form represents a chord-type piecewise linear approximation to smooth μ -law or A-law. The parameter a represents the segment edge effects, b relates to coding procedure, and the parameter c represents the centering. These parameters are typically: DLA ($a = 0.5$ for $b = \pm 0.5$), RLA ($a = 0$ for $b = +0.5$, or $a = 1$ for $b = -0.5$), mid-tread ($c = 0$) and mid-riser ($c = 0.5$). The effect of c is important in small signal range levels. The effect of a on the signal quality is very small, but the mixed use of DLA and RLA approaches introduce small degradations in S/N and net loss tracking.

Synthesis of coders and decoders was described; and it was shown that the design figures such as amount of biasing in the decoder, the

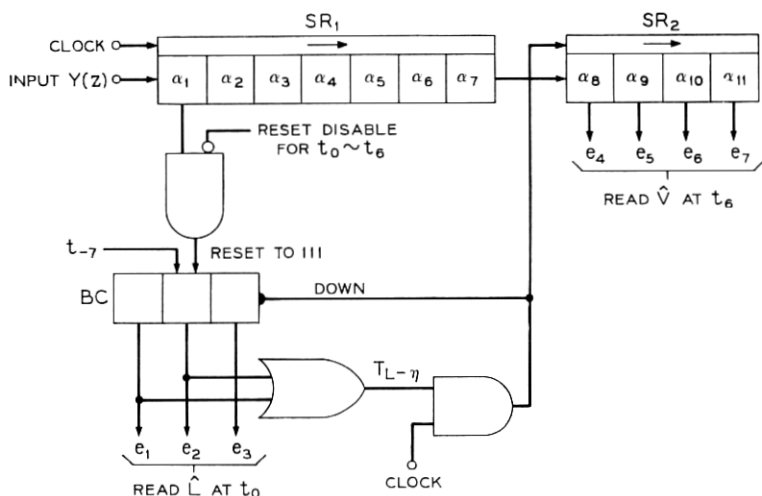


Fig. 18—Thirteen-segment A-law digital compressor with DLA ($a = 0.5$), mid-riser ($c = 0.5$), $m = 3$, and $n = 4$.

type of logic to operate it, and the coding algorithm for the sequential comparison coder can be derived from our formulation. Bipolar codecs are also given in terms of the signed binary code representations.

Another important consequence of this formulation is the systematic synthesis of digital companders. The expansion algorithm is relatively straightforward. The compression algorithm for DLA was shown to be equivalent to employing the rule of "rounding" for the fractional digits of variable V . Examples of compandor circuits synthesized from these algorithms are shown for the segment μ -law and segment A-law.

The A-law companding circuit appears to be slightly simpler than that for the μ -law, but the difference of the two is not so great as to weigh heavily in the choice between the μ -law and the A-law. It is noted that the formulation and the conversion algorithms developed here can also be conveniently applied for the synthesis of a digital converter between μ -law and A-law or a digital attenuator.

VI. ACKNOWLEDGMENT

The author would like to express his thanks to M. R. Aaron for his guidance, suggestions and encouragement. He also helped by editing the manuscript. The author is also indebted to C. L. Damman, L. D. McDaniel, H. H. Henning and W. L. Montgomery for useful communications.

APPENDIX A

Proof of Congruence of Equations (15) and (16)

For nonedge points $X(L, V)$, from equations (7) and (14) obviously

$$y_a(X | b = +0.5) = y_a(X + 1 | b = -0.5) \quad (56)$$

and, hence, (15) and (16) are congruent for nonedge points. For edge points

$$y_a(L, N - 1 | b = +0.5) = f(L, N + a - 0.5) - a + c, \quad (57)$$

$$\begin{aligned} y_a(L + 1, 0 | b = -0.5) &= f(L + 1, a - 0.5) - a + c \\ &= f[L, N + 2(a - 0.5)] - a + c. \end{aligned} \quad (58)$$

Here, these two equations are linear functions of a . Therefore, the two equations are equal if and only if $a = 0.5$. Thus, equations (15) and (16) are congruent for any X if and only if $a = 0.5$.

APPENDIX B

Proof of the Statements in Section 2.8

For given output levels the optimum decision levels lie halfway between two adjacent output levels.¹⁰ At the nonedge points the proof is obvious from equation (14). At the edge points, the average of $y_0(L, N - 1)$ and $y_0(L + 1, 0)$ is given by

$$\begin{aligned} \frac{1}{2}\{y_0(L, N - 1) + y_0(L + 1, 0)\} &= y_0(L, N - 1) + \frac{1}{2} \Delta'_0(L) \\ &= y_0(L, N - 1) + \frac{1}{2} \Delta_0(L) \cdot (1 + a) \\ &= y_a\left(L, N - 1 + \frac{a}{2} \mid b = +0.5\right). \end{aligned} \quad (59)$$

This is equal to $y_a(L, N - 1 \mid b = +0.5)$ if and only if $a = 0$. Also equation (59) is expressed as

$$\begin{aligned} y_0(L + 1, 0) - \frac{1}{2} \Delta'_0(L) &= y_0(L + 1, 0) - \frac{1}{4} \Delta_0(L + 1) \cdot (1 + a) \\ &= y_a\{L + 1, (1 - a)/4 \mid b = -0.5\}. \end{aligned}$$

Therefore, this is equal to $y_a(L + 1, 0 \mid b = -0.5)$ if and only if $a = 1$. This proves that the RLA condition is satisfied if and only if $a = 0$ for $b = +0.5$ or $a = 1$ for $b = -0.5$.

Next, for given decision levels the optimum output levels should be placed halfway between two adjacent decision levels.¹⁰ At the nonedge points this is obviously satisfied. For edge points, since the half way point is equal to $\tilde{y}(L_e, V_e)$, from equation (23) it is shown that $\tilde{y} = y_0$ if and only if $a = 0.5$. Therefore, the DLA condition is satisfied if and only if $a = 0.5$.

APPENDIX C

Derivation of Algorithm 3

Since in DLA ($a = 0.5$) the upper and lower bound approaches are identical, we will prove Algorithm 3 only for the case of $b = -0.5$. From Algorithm 2, equation (46) can be rewritten as

$$y_a(\hat{L}, \hat{V}) \leq Y < y_a(\hat{L}, \hat{V} + 1).$$

Since $a = 0.5$, this is valid for all $\hat{V} \in \{0, 1, \dots, N - 1\}$. From equations (56) and (27) the above inequality can be written as

$$\hat{V} + N \leq W(\hat{L}, Y) < \hat{V} + N + 1. \quad (60)$$

Considering the range of $\hat{V} \in \{0, 1, \dots, N - 1\}$, we have as the \hat{L} determining criterion

$$N \leq W(\hat{L}, Y) < 2N.$$

Then, from equations (50) and (60) we immediately have

$$\hat{V} = \mathfrak{J}[W(\hat{L}, Y)] - N.$$

This completes the derivation of Algorithm 3.

REFERENCES

1. Smith, B., "Instantaneous Companding of Quantized Signals," B.S.T.J., 36, No. 3 (May 1957), pp. 653-709.
2. Mann, H., Straube, H. M., and Villars, C. P., "A Companded Coder for an Experimental PCM Terminal," B.S.T.J., 41, No. 1 (January 1962), pp. 1173-226.
3. Schreiner, S. M., and Vallarino, A. R., "48-Channel PCM System," 1957 IRE Nat. Conv. Rec., Part 8, pp. 141-149.
4. Chatelon, A., "Application of Pulse Code Modulation to an Integrated Telephone Network, Part 2—Transmission and Encoding," ITT Elec. Commun., 38, No. 1 (January 1963), pp. 32-43.
5. CCITT (International Telegraph and Telephone Consultative Committee) Document COM XV-No. 77/E, "Interim Report of Working Party on Question 33/XV," 1966.
6. Bucci, W., "PCM: A Global Scramble for Systems Compatibility," Electronics, 42, No. 13 (June 23, 1969), pp. 94-102.
7. Henning, H. H., "96 Channel PCM Channel Bank," 1969 Int. Commun. Conf., Denver, Colorado, June 9-11, 1969, pp. 34.17-34.22.
8. Montgomery, W. L., "Digitally Linearizable Companders with Comments on Project for a Digital Telephone Network," IEEE Trans. on Commun. Technology, Com-18, No. 1 (February 1970), pp. 1-4.
9. Purton, R. F., "A Survey of Telephone Speech-Signal Statistics and Their Significance in the Choice of PCM Companding Law," Proc. IEE, 109, No. 1 (January 1962), pp. 60-66.
10. Max, Joel, "Quantizing for Minimum Distortion," IRE Trans. on Inform. Theory, IT-6, No. 1 (March 1960), pp. 6-12.
11. Hoeschele, D. F. Jr., "Analog to Digital/Digital to Analog Conversion Techniques," New York: John Wiley, 1968.
12. Chu, Y., "Digital Computer Design Fundamentals," New York: McGraw-Hill, 1962.
13. Jackson, L. B., Kaiser, J. F., and McDonald, H. S., "An Approach to the Implementation of Digital Filters," IEEE Trans. on Audio and Electroacoustics, AU-16, No. 3 (September 1968), pp. 413-421.
14. Kaneko, H., unpublished work.
15. Schaefer, D. H., "Logarithmic Compression of Binary Numbers," Proc. IRE, 49, No. 7 (July 1961), p. 1219.