AD-A048 611 INSTITUTE FOR TELECOMMUNICATION SCIENCES BOULDER COLO F/G 17/2
OBJECTIVE MEASUREMENT OF VOICE CHANNEL INTELLIGIBILITY.(U)
OCT 77 K J GAMAUF, W J HARTMAN DOT-FA74WAI-448
UNCLASSIFIED FAA-RD-77-153 NL

| OF |
AD
A048611

END
DATE
FILMED
2-78
DDC

Report No. FAA-RD-77-153

OBJECTIVE MEASUREMENT OF VOICE CHANNEL INTELLIGIBILITY

K. J. Gamauf
W. J. Hartman

U.S. Department of Commerce
Office of Telecommunications
Institute for Telecommunication Sciences
Boulder, Colorado 80302

October 1977

Final Report

Document is available to the U.S. public through the National Technical Information Service, Springfield, Virginia 22161.

Prepared for

U.S. DEPARTMENT OF TRANSPORTATION
FEDERAL AVIATION ADMINISTRATION
Systems Research & Development Service
Washington, D.C. 20590

NOTICE

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
(18) (19) FAA-RD-77-153		

4. Title and Subtitle	5. Report Date
(6) OBJECTIVE MEASUREMENT OF VOICE CHANNEL INTELLIGIBILITY,	(11) Oct [redacted] 77
	6. Performing Organization Code OT/ITS
7. Author(s) (10) K.J. Gamauf W.J. Hartman	8. Performing Organization Report No. (12) 75p.
9. Performing Organization Name and Address Institute for Telecommunication Sciences Office of Telecommunications U.S. Department of Commerce Boulder, Colorado 80302	10. Work Unit No. (TRAIS)
	11. Contract or Grant No. (15) DOT-FA74WAI-448
12. Sponsoring Agency Name and Address JOINTLY SPONSORED BY U.S. Dept. of Transportation Federal Aviation Administration Systems Research and Development Service Washington, DC 20591	13. Type of Report and Period Covered (9) Final rept.,
	14. Sponsoring Agency Code ARD-60

And
Headquarters
U.S. Army Communications-Electronics Engineering Installation Agency
Fort Huachuca, AZ 85613

Delivery Order No. DAEA18-76-F-D402

16. Abstract

Following the results of a feasibility study (Hartman and Boll, 1976) an objective intelligibility measure is developed using a large data base consisting of 8-50 word phonetically balanced word groups with twelve different kinds of distortion. Justification for the use of this particular measure is included, with mathematical derivations and physical interpretations.

A discussion of the feasibility of a hardware implementation of the software developed here is also included.

17. Key Words	18. Distribution Statement
Intelligibility measurements, linear predictive coding, voice systems.	Document is available to the public through the National Technical Information Service, Springfield, Virginia 22151.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	76	

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

403 394

METRIC CONVERSION FACTORS

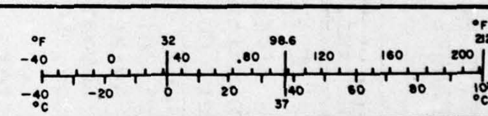
Approximate Conversions to Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
in	inches	*2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
		AREA		
in^2	square inches	6.5	square centimeters	cm^2
ft^2	square feet	0.09	square meters	m^2
yd^2	square yards	0.8	square meters	m^2
mi^2	square miles	2.6	square kilometers	km^2
	acres	0.4	hectares	ha
		MASS (weight)		
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2000 lb)	0.9	tonnes	t
		VOLUME		
tsp	teaspoons	5	milliliters	ml
Tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft^3	cubic feet	0.03	cubic meters	m^3
yd^3	cubic yards	0.76	cubic meters	m^3
		TEMPERATURE (exact)		
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

*1 in = 2.54 (exactly). For other exact conversions and more detailed tables, see NBS Misc. Publ. 286, Units of Weights and Measures, Price $2.25, SD Catalog No. C13.10:286.

Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
m	meters	1.1	yards	yd
km	kilometers	0.6	miles	mi
		AREA		
cm^2	square centimeters	0.16	square inches	in^2
m^2	square meters	1.2	square yards	yd^2
km^2	square kilometers	0.4	square miles	mi^2
ha	hectares (10,000 m^2)	2.5	acres	
		MASS (weight)		
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1000 kg)	1.1	short tons	
		VOLUME		
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m^3	cubic meters	35	cubic feet	ft^3
m^3	cubic meters	1.3	cubic yards	yd^3
		TEMPERATURE (exact)		
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F



FEDERAL AVIATION ADMINISTRATION
SYSTEMS RESEARCH AND DEVELOPMENT SERVICE
SPECTRUM MANAGEMENT STAFF

STATEMENT OF MISSION

The mission of the Spectrum Management Staff is to assist the Department of State, Office of Telecommunications Policy, and the Federal Communications Commission in assuring the FAA's and the nation's aviation interests with sufficient protected electromagnetic telecommunications resources throughout the world to provide for the safe conduct of aeronautical flight by fostering effective and efficient use of a natural resource - the electromagnetic radio frequency spectrum.

This objective is achieved through the following services:

- Planning and defending the acquisition and retention of sufficient radio frequency spectrum to support the aeronautical interests of the nation, at home and abroad, and spectrum standardization for the world's aviation community.

- Providing research, analysis, engineering, and evaluation in the development of spectrum related policy, planning, standards, criteria, measurement equipment, and measurement techniques.

- Conducting electromagnetic compatibility analyses to determine intra/inter-system viability and design parameters, to assure certification of adequate spectrum to support system operational use and projected growth patterns, to defend aeronautical services spectrum from encroachment by others, and to provide for the efficient use of the aeronautical spectrum.

- Developing automated frequency selection computer programs/routines to provide frequency planning, frequency assignment, and spectrum analysis capabilities in the spectrum supporting the National Airspace System.

- Providing spectrum management consultation, assistance, and guidance to all aviation interests, users, and providers of equipment and services, both national and international.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

OBJECTIVE MEASUREMENT OF VOICE
CHANNEL INTELLIGIBILITY

K.J. Gamauf
W.J. Hartman*

Following the results of a feasibility study (Hartman and Boll, 1976) an objective intelligibility measure is developed using a large data base consisting of 8-50 word phonetically balanced word groups with twelve different kinds of distortion. Justification for the use of this particular measure is included, with mathematical derivations and physical interpretations.

A discussion of the feasibility of a hardware implementation of the software developed here is also included.

Key words: intelligibility measurements; linear predictive coding; voice systems.

1. INTRODUCTION

There has long been a need for an inexpensive, reliable, and efficient method to evaluate the quality of speech sent over voice communication channels. Few voice communication systems today are judged by the quality and intelligibility of the speech received by the listener. Instead, system performance is generally specified by some engineering parameter, such as the signal-to-noise ratio of the receiver output.

The most common procedure for determining the intelligibility of a voice channel is a subjective method that involves trained speakers and listener panels that directly score the percentage of speech that is intelligible. These schemes have the desirable property that they produce repeatable results. Unfortunately, subjective scoring methods are expensive and time consuming and as a result, are not widely used. What is needed is an

---

*The authors are with the Institute for Telecommunication Sciences, Office of Telecommunications, U.S. Department of Commerce, Boulder, CO 80302.

inexpensive and efficiently applied objective evaluation of speech intelligibility that is comparable to subjective methods.

This paper develops a method of obtaining an objective intelligibility measure that gives good results for speech sent through both analog and digital, noise-corrupted communication channels. The distortion measure is obtained using Linear Predictive Coding (LPC), a mathematical technique widely known for its application to the analysis and synthesis of speech. The feasibility of using LPC to develop an objective intelligibility measure has been demonstrated by Hartman and Boll (1976).

A. The Articulation Index

A well known objective measure that is used for voice communication channel evaluation is the Articulation Index (AI) (ANSI, 1969; Kryter, 1962). The AI is a physical measurement that is highly correlated to speech intelligibility under certain conditions. The AI is obtained by evaluating the signal-to-noise power ratios in 20 specific frequency bands. These power ratios are then summed and normalized to give a score between zero and one. An AI equal to one signifies perfect intelligibility, while a value of zero represents a complete lack of intelligibility. The AI can be computed using a general purpose computer once the required spectral data are obtained from the speech.

An automated technique to obtain the AI is achieved through the Speech Communication Index Meter (SCIM)(Kryter and Ball, 1964), which uses 9 frequency bands, instead of 20, to obtain a modified AI. The SCIM system can be used to perform on-line measurements and has been found to be highly correlated with the standard 20 band AI. The SCIM system's AI can also be directly related to speech intelligibility as long as the noise present is generally additive white gaussion noise. Clipping of the speech by the voice channel or noise that is intermittent or colored distorts the AI, and a correction factor must be employed. Multiplicative noise requires a complete recalibration of the SCIM system. Therefore, the type of noise or distortion present in the voice channel must be known in order to obtain accurate results. When digital voice systems were tested on the SCIM scheme, very poor estimates of speech intelligibility were obtained. This was true even when quantization noise was the only distortion present. Reliable correction factors for digital voice systems have as yet not been found to compensate for this poor performance. A more detailed description of the AI and the SCIM system of objective voice channel evaluation can be found in the work of Hubbard and Hartman (1974).

B. Chapter Summaries

Chapter 2 gives a brief discussion of linear prediction of the speech waveform. Chapter 3 describes the analog voice tapes used in this study. Chapter 4 describes the data processing of the analog voice tapes in order to obtain the LPC information from the speech. Chapter 5 discusses the distance or distortion measures that were used to predict objectively the intelligibility of noise corrupted speech. A comparison between objective distance measures and subjective intelligibility scores is given. Chapter 6 gives a block diagram for hardware implementation of the scoring techniques. Appendix A contains listings of the computer programs used to obtain the numerical results found in this report.

2. LINEAR PREDICTIVE CODING

Linear predictive coding (LPC) has long been used in communication theory. More recently, it has found applications in speech analysis and synthesis, speaker identification, and word recognition, to name just a few new areas. In this study, LPC is used to develop an objective intelligibility measure of speech corrupted by noise.

A. Linear Prediction

LPC models the vocal tract as an all-pole digital filter and estimates the filter parameters (predictor coefficients) using the time domain speech waveform itself, rather than the waveform's short-term frequency spectrum. This makes LPC a relatively efficient method for encoding speech compared to frequency domain techniques.

The vocal tract is assumed to be modeled as a discrete, time-varying filter with parameters changing slowly enough so that they can be considered fixed over a specified time interval. Hence, the vocal tract can be approximated by a series of stationary shapes. Atal and Hanauer (1971) have shown that this all pole model can account for the glottal volume flow and radiation of sound from the mouth in addition to vocal tract sounds.

The transfer function, H(z), used to describe the digital model over each analysis frame is given by

$$H(z) = \frac{1}{1 - \sum_{i=1}^{P} a_i z^{-i}} . \qquad (2.1)$$

for a model with P poles.

The time sequence S_n corresponding to the output of the recursive filter can be written as

$$S_n = \sum_{i=1}^{P} a_i S_{n-i} + \delta_n \qquad n = 0, 1, \ldots \qquad (2.2)$$

where (a_i) are the predictor coefficients that completely describe the characteristics of the filter and (δ_n) is the driving function or input to the filter.

While there have been several formulations for the estimation of the linear prediction coefficients, two least squares methods have become prominent, the autocorrelation method and the covariance method. The autocorrelation method, which will be justified in Chapter 4, was chosen for use in this study and further discussion will be restricted to that scheme. The autocorrelation method can be considered as estimating the filter coefficients by approximating the spectrum of the speech waveform by an all-pole model.

The portion of the signal to be analyzed is first multiplied by a finite window of length N, changing the signal to

$$S_n = \begin{cases} \text{Windowed speech samples, } o \le n \le N-1 \\ o, \; n < o \text{ and } n \geq N \end{cases} \quad (2.3)$$

Using this windowed signal, the prediction error sequence is defined as

$$e_n = S_n - \sum_{i=1}^{P} a_i \, S_{n-i} \qquad n = 0, 1, \ldots \quad (2.4)$$

and the total squared error is then

$$E = \sum_n e_n^{\,2} = S_o^{\,2} + \sum_{n=1}^{N-1+P} (S_n - \sum_{i=1}^{P} a_i \, S_{n-i})^2 . \quad (2.5)$$

The predictor coefficients are selected so as to minimize the total squared error. This is accomplished by setting the partial derivative of the total squared error with respect to each predictor coefficient equal to zero. The system of equations that results is

$$\sum_{k=1}^{P} r_{|i-k|} \, a_k = r_i \qquad i = 1, 2, \ldots, P \quad (2.6)$$

where

$$r_i = \frac{1}{R_o} \sum_{n=o}^{N-1-|i|} S_n \, S_{n+|i|} \qquad (= 0, 1, 2, \ldots, P \quad (2.7)$$

are the normalized short-term autocorrelation values of the speech signal and

$$R_o = \sum_{n=o}^{N-1} S_n^{\,2} \quad (2.8)$$

is the normalized factor for these values. The normalized total squared error can be defined by making use of equations (2.5) and (2.6), yielding

$$E_{min} = 1 - \sum_{i=1}^{P} a_i r_i \quad . \tag{2.9}$$

The predictor coefficients are obtained by inverting a positive definite Toeplitz matrix

$$\left[r_{|i-k|} \right]_{i,\ k\ =\ 1,\ 2,\ .\ .\ .,\ P} \tag{2.10}$$

This system of equations can be solved by using Levinson's recursion method, which will be expounded upon further in Chapter 4. The Toeplitz matrix is sometimes called the auto-correlation matrix, and the coefficients obtained from this linear system result in a recursive filter, H(z), which is guaranteed to be stable, (all of its poles lie inside the unit circle), as shown by Grenander and Szego (1958).

B. Spectral Approximation

Further insight can be gained by looking at the frequency domain approximation to the above system. Taking the z-transform of equation (2.4), one obtains

$$E(z) = S(z)\ [H(z)]^{-1} \tag{2.11}$$

where H(z) is defined in equation (2.1) and E(z) and S(z) are the z-transforms of E_n and S_n respectively. Rearranging, equation (2.11) can be written as

$$S(z) = E(z)\ H(z). \tag{2.12}$$

Minimizing the total squared prediction error is equivalent to approximating the error sequence, (e_n), by

$$\hat{e}_n = \begin{cases} A, & n = 0 \\ 0, & n \neq 0 \end{cases} \tag{2.13}$$

in a least squares sense. This implies that E(z) is being approximated by the function A, a constant, and S(z) is being approximated by a spectrum corresponding to an all-pole transfer function, i.e.,

$$\hat{E}(z) = A \tag{2.14}$$

$$\hat{S}(z) = \hat{E}(z)\ H(z) = \frac{A}{1 - \sum_{i=1}^{P} a_i z^{-i}} . \tag{2.15}$$

The value of A is determined by the application of energy conservation between $\hat{e}_n$ and e_n. Using equations (2.9) and (2.13), one obtains

$$A^2 = E_{min} = 1 - \sum_{i=1}^{P} a_i r_i \tag{2.16}$$

thereby showing that A^2 is equal to the minimum total squared error of the system.

This approach of estimating filter coefficients so as to minimize the energy of the output of the inverse of a system driver by its impulse response is sometimes called deconvolution or inverse filtering. Considerable work has been done in this area of linear prediction of speech in the past few years and many good references are available that give more detailed discussions of this subject. Some particularly good ones are Markel and Gray (1976 and 1973); Makhoul (1975 and 1973); Makhoul and Wolf (1972); and Boll (1973).

3. THE VOICE TAPES

In order to develop an objective intelligibility measure for corrupted speech, a comparison must be performed between the distorted speech and the original noise free speech. A subjective intelligibility measure of the distorted speech must also be available in order to judge the quality of the objective measure being used. Both of these requirements are met by first making a noise free master tape of pre-selected speech and then sending it through voice communication channels to be tested and making a recording of the speech at the channel output. This recording can be subjectively scored for intelligibility and also compared to the original speech by some mathematical technique to obtain an objective measure.

A. Description of Voice Tapes

The pre-selected speech to be sent over a voice channel for intelligibility scoring are phonetically balanced (PB) groups of isolated words as opposed to complete sentences or nonsense syllables. These PB words were used because subjective scores have been shown to be repeatable, which is a necessary criterion for this study because the objective measure will be repeatable. Eight PB word groups, each containing fifty isolated words were selected as the test speech. A list of the fifty words in each word group is given in Tables 3-1 through 3-8, with their designated word group numbers.

An analog tape containing all eight word groups and using both male and female trained speakers was obtained from the Army Electronic Proving Ground Electromagnetic Environment Test Facility at Fort Huachuca, Arizona. From this tape, a master analog tape was made that would be sent over voice channels and later compared with the recorded output of the channel. In order to perform this comparison, the two tapes would have to be aligned, which meant synchronization information must be included on the master tape before being sent across the voice channel. Because

Table 3-1

PB Word Group 361

1.	STAB	26.	RUG
2.	TUCK	27.	CLIFF
3.	DRAPE	28.	LOUSE
4.	PITCH	29.	GAB
5.	INK	30.	RYE
6.	AID	31.	SANG
7.	KIND	32.	CLOSED
8.	STRESS	33.	THREE
9.	TURN	34.	MAP
10.	DROOP	35.	GAS
11.	PUMP	36.	SHEEP
12.	SUIT	37.	CREWS
13.	BARGE	38.	THRESH
14.	KNEE	39.	NAP
15.	DUB	40.	HAD
16.	WIELD	41.	SHEIK
17.	ROCK	42.	TIRE
18.	BOOK	43.	DAME
19.	THOU	44.	NEXT
20.	LAY	45.	HASH
21.	FIFTH	46.	SOAR
22.	ROGUE	47.	TON
23.	CHEESE	48.	DIN
24.	LEASH	49.	PART
25.	FRIGHT	50.	HOSE

Table 3-2

PB Word Group 312

1.	JAB	26.	DIP
2.	ARC	27.	URGE
3.	JAUNT	28.	MOUTH
4.	ARM	29.	NET
5.	SHOP	30.	WAVE
6.	BEAM	31.	FINE
7.	KIT	32.	PURSE
8.	BLISS	33.	GOAT
9.	SPRIG	34.	HOG
10.	LAG	35.	RISK
11.	CHUNK	36.	DOUBT
12.	LATCH	37.	PUNK
13.	CODE	38.	DRAKE
14.	LOW	39.	WOOD
15.	TAB	40.	FEEL
16.	SHOT	41.	PROD
17.	SIGN	42.	FRISK
18.	CRUTCH	43.	DULL
19.	SAP	44.	MOST
20.	LOSS	45.	FUDGE
21.	CLASH	46.	POND
22.	SNOW	47.	HAVE
23.	CRY	48.	REEF
24.	SPY	49.	PROBE
25.	STIFF	50.	RICE

Table 3-3

PB Word Group 291

1.	ARCH	26.	NUTS
2.	BEEF	27.	WIPE
3.	KEY	28.	ODD
4.	SIP	29.	WITH
5.	BIRTH	30.	FLAG
6.	SMART	31.	NERVE
7.	SPUD	32.	FLUFF
8.	CLUB	33.	FOE
9.	TEN	34.	NOOSE
10.	CROWD	35.	FUME
11.	THAN	36.	WEAK
12.	BIT	37.	FUSE
13.	THANK	38.	WILD
14.	CUD	39.	GIVE
15.	THRONE	40.	PHONE
16.	CARVE	41.	GATE
17.	TOAD	42.	HOOF
18.	LIT	43.	YEAR
19.	CHESS	44.	ICE
20.	TROOP	45.	REED
21.	CHEST	46.	ITCH
22.	BOOST	47.	ROOT
23.	CLOWN	48.	GRACE
24.	DITCH	49.	PACT
25.	MASS	50.	RUDE

Table 3-4

PB Word Group 265

1.	AS	26.	CLOTH
2.	BEST	27.	GROPE
3.	EAT	28.	KEPT
4.	THUS	29.	RAY
5.	EYES	30.	FORGE
6.	SCAN	31.	CLOTHES
7.	COB	32.	ROOMS
8.	FALL	33.	LAG
9.	DAD	34.	THIGH
10.	ODE	35.	WAIT
11.	SHANK	36.	WIFE
12.	MASH	37.	JAG
13.	HITCH	38.	NIGH
14.	ROUGH	39.	CRIB
15.	FEE	40.	PRIG
16.	CHART	41.	FLOP
17.	WASP	42.	SUP
18.	HULL	43.	GAGE
19.	TONGUE	44.	WRIT
20.	PUN	45.	PRIME
21.	REAP	46.	FOWL
22.	PUS	47.	BOG
23.	BADGE	48.	GAP
24.	DEEP	49.	FLICK
25.	SLOUCH	50.	RAISE

Table 3-5

PB Word Group 275

1.	AM	26.	SLEDGE
2.	GRADE	27.	RANGE
3.	GASP	28.	WOO
4.	MOTE	29.	DOPE
5.	MUD	30.	FLING
6.	BY	31.	NINE
7.	PHASE	32.	SCOUT
8.	RASH	33.	OFF
9.	RICH	34.	PIG
10.	POUNCE	35.	FORT
11.	SHAFT	36.	WOE
12.	ROAR	37.	CHOP
13.	ACT	38.	PLOD
14.	AIM	39.	KNIT
15.	HIM	40.	WHIFF
16.	COAST	41.	PENT
17.	DOSE	42.	THOUGH
18.	BUT	43.	JUG
19.	SOUTH	44.	SNIFF
20.	SIEGE	45.	QUIZ
21.	DWARF	46.	GUN
22.	FAKE	47.	COOK
23.	CUT	48.	SAG
24.	COMES	49.	WIRE
25.	SIN	50.	RAID

Table 3-6
PB Word Group 305

1.	STAFF	26.	TREE
2.	BASH	27.	GOOSE
3.	HAT	28.	PAGE
4.	WADE	29.	MAZE
5.	CHAMP	30.	FLIGHT
6.	ETCH	31.	PINK
7.	SLUG	32.	BUG
8.	CHANCE	33.	RAPE
9.	WAKE	34.	EARS
10.	VALVE	35.	SCRUB
11.	YOUTH	36.	COW
12.	FLAUNT	37.	TAG
13.	RUSH	38.	JAY
14.	GULL	39.	VOID
15.	DAUB	40.	EARTH
16.	REAL	41.	THOSE
17.	AIL	42.	LAP
18.	PUT	43.	SNIPE
19.	NUDGE	44.	FIR
20.	BACK	45.	CLOTHE
21.	PLUS	46.	MOPE
22.	BOB	47.	CORD
23.	THUG	48.	RIP
24.	CUE	49.	HURT
25.	LINE	50.	FORCE

Table 3-7

PB Word Group 214

1.	TOE	26.	HID
2.	ARE	27.	SUCH
3.	RUB	28.	CRASH
4.	GROVE	29.	BOX
5.	PANTS	30.	THERE
6.	DEATH	31.	END
7.	BAD	32.	MANGE
8.	PAN	33.	PLUSH
9.	USE	34.	IS
10.	SLIP	35.	FORD
11.	BASK	36.	HUNT
12.	FRAUD	37.	RAG
13.	NOT	38.	FEAST
14.	DEED	39.	NO
15.	SMILE	40.	CLOVE
16.	DISH	41.	FERN
17.	RISE	42.	PILE
18.	FUSS	43.	STRIFE
19.	WHEAT	44.	CANE
20.	DIKE	45.	FOLK
21.	PEST	46.	RAT
22.	CREED	47.	CLEANSE
23.	HEAP	48.	THEN
24.	BAR	49.	RIDE
25.	NOOK	50.	HIVE

Table 3-8
PB Word Group 283

1.	US	26.	BIND
2.	SHACK	27.	CHEW
3.	CRACK	28.	WHEEZE
4.	CHANT	29.	FREAK
5.	YEAST	30.	PINT
6.	ASK	31.	GUESS
7.	EASE	32.	QUEEN
8.	REST	33.	CLOD
9.	JELL	34.	LOOK
10.	BOLT	35.	FRONT
11.	KILL	36.	NIGHT
12.	LICK	37.	WIG
13.	CALF	38.	ROPE
14.	CATCH	39.	DAY
15.	TILL	40.	RHYME
16.	EACH	41.	SLIDE
17.	ROT	42.	FROCK
18.	ROLL	43.	LEFT
19.	BID	44.	FOOD
20.	COD	45.	SPICE
21.	DEUCE	46.	BORED
22.	DUMB	47.	THIS
23.	FAD	48.	THREAD
24.	HUM	49.	FORTH
25.	ROD	50.	FLIP

the tapes would be processed in a digital state, the alignment procedure would also have to work in a digital format. It was found that a shift of plus or minus 10 samples of a 256 sample analysis window caused the predictor coefficients to vary less than 0.1% in all cases. Therefore, the synchronization procedure to be used was required to align two segments of digitized speech to within 10 samples.

It should be noted that the bound on the variation of the predictor coefficients cannot be translated into a bound on the distortion measures described in Chapter 5. However, the alignment method described in the next section was tested extensively, and never produced a variation in the distortion measures larger than that produced by the normal round off error.

B. Voice Tape Alignment

A synchronization procedure that was found to meet the required 10 sample variation specification, made use of a binary pseudo noise (PN) sequence. The binary PN sequence was sent through a phase-continuous frequency shift keying modem using the two frequencies 1.2 kHz and 2.2 kHz. Several different length binary PN signals and modem bit rates were tested to determine the best combination for alignment capability. The test consisted of cross-correlating the PN sequences under different noise and distortion conditions and looking for an impulse like correlation function. A further requirement was to have the PN sequence as short as possible. It was found that a length 127 binary PN sequence sent through the FSK modem operating at a 635 Hz bit rate followed by a low pass filter with a cutoff of 2.5 kHz met all the requirements necessary to insure the alignment of two PN sequences distorted by noise. The low pass filter was used to make certain that the frequency spectrum of the PN signal was in the range required for input to most voice systems. A PN signal was then placed before each word and after the last word of all eight word groups thereby creating the master analog tape with alignment capabilities.

In order to align a distorted tape with the master tape, the location of all the PN sequences and words on the digitized master tape had to be known. This was done by blocking the quantized samples into 125 sample records and computing the mean and standard deviation (SD) for each record. The SD was used as an energy criterion to determine the midpoints of the PN sequences and words and the length of each word. The distances between the midpoints of each PN sequence and the word following it were then determined. The corresponding midpoints of the words from a distorted tape are now all that remains to be found.

Each of the eight word groups on a digitized tape made up one file and corresponding files between the master and distorted tape were aligned independently from the other seven sets of files. Using the SD energy criterion, the midpoints of the first and last PN sequences of the distorted word groups are estimated. The cross-correlation between these PN sequences and the corresponding ones from the master tape are then computed, thereby obtaining the midpoints of the two distorted PN sequences with respect to the master word group's PN sequences. From this computation, the slight drift between the samples of the two tapes can be calculated. Using this drift and the PN sequence midpoints of the master tape, an estimate of the midpoints of the PN sequences of the distorted tape can be made taking into account the shift between the two tapes. The true midpoints of the PN sequences of the distorted tape with respect to the master tape can now be found by again computing the cross-correlation between each pair of PN sequences. Using the distances between the midpoints of the PN sequences and the words following them of the master word group, and the drift between the two tapes, the midpoints of the words of the distorted word group with respect to the master can be obtained using the midpoints of the PN sequences of the distorted word group. This procedure is repeated for all eight word groups of each distorted tape.

The alignment of two words from two different tapes to within 10 samples was the goal of the synchronization procedure.

This can safely be assumed using the above alignment scheme. The PN sequences on either side of each word are lined up to within one sample in all cases. The drift between two consecutive PN sequences was never more than 15 samples, usually quite a bit less. Since the word to be aligned is roughly midway between the two PN sequences and the drift is taken into account, the 10 sample synchronization specification is always met, generally to within a sample or two. The drift between two tapes was verified to be linear, with only small (± 1 sample) fluctuations.

Three computer programs were used in the synchronization procedure discussed above. "Words" was the program that was used to find the locations of the PN sequences and words through the SD energy criterion. FFTCOR4 computed the cross-correlation between the PN sequences of the two tapes. Finally, WRDMIDP calculated the midpoints of the words of the distorted tape with respect to the master tape. A listing of all three programs can be found in Appendix A.

4. DATA PROCESSING OF THE VOICE TAPES

Once the master analog tape of eight 50-word groups was made, it could then be sent over various voice communication channels to obtain distorted tapes. Copies of the distorted tapes were sent to Fort Huachuca to be scored for intelligibility. The intelligibility score for a single word was the percentage of the listener panel that correctly identified it, and the intelligibility score for the entire word group was the average score of all 50 words in the word group. The subjective intelligibility scores were used later for comparison with the objective intelligibility measure. The distorted tapes and the master tape were then processed to obtain the LPC information necessary to develop the objective distance measure. No filtering was used on the tapes used for subjective scoring.

A. Digitization of the Voice Tapes

Before the master or distorted tapes were digitized, they were first sent through a pre-emphasis filter, and then low-pass filtered to 3.2 kHz. Pre-emphasis was used because it enhances the high-frequency formants of the speech which is important for speech comparison. Pre-emphasis also limits the effects of the glottal waveform and lip radiation and therefore enhances the spectral properties of the speech due to the vocal tract.

Based upon an average vocal tract length of 17 cm, the first three formant frequences will be found in the frequency range of about 250 - 2800 Hz. Shorter vocal tracts will shift this range up slightly. Low-pass filtering at 3.2 kHz would therefore pass the first three formant frequencies. This (3.2 kHz) is generally also the high frequency cut-off for most voice communication channels. Any noise above 3.2 kHz picked up by the distorted tapes will also be filtered out which will help the accuracy of the objective distance measure.

The analog tapes were sampled at 10 kHz and then quantized to 12 bits. The sampled signal was then stored on digital magnetic tape for future processing.

B. Analysis Conditions

Once the tapes are digitized and the distorted tapes are aligned to the master tape, LPC processing of each word can be done. First, however, the decision must be made regarding which least squares method to be used to obtain the predictor coefficients. For this study, the autocorrelation method of linear prediction was chosen over the covariance method because it requires fewer calculations, it is assured of producing a stable filter, (i.e. all the poles are within the unit circle), and it allows a meaningful spectral matching term to be computed. Also, as mentioned before, the autocorrelation method can be considered as estimating the predictor coefficients by approximating the spectrum of the speech waveform by an all-pole filter or model. A detailed comparision between the two methods is given in Makhoul and Wolf (1972).

The analysis interval in which the speech waveform's spectrum is estimated should be short enough so that vocal tract movement is negligible, but long enough to insure stable spectral estimates. The vocal tract can, in general, be assumed to be stationary on the order of 15 to 20 ms. Since in the autocorrelation method of linear prediction the approximation is to model a short-term signal spectrum, it is necessary to window in order to guarantee spectrally accurate results. By using a nonrectangular window, a larger analysis interval can be used without sacrificing spectral accuracy. Therefore, the length of the analysis frame was chosen to contain 256 samples, which means an analysis interval of 25.6 ms because of the 10 kHz sampling rate. The non-rectangular window used on each analysis frame was a Hamming window of the form

$$W_H(129-n)=W_H(128+n)=\frac{1}{1.08\cdot N}\cdot(5.04+0.46 \cos \frac{2\pi n}{N})\quad n=1, 2, \ldots, 128 \tag{4.1}$$

where N=256, the analysis frame length. The Hamming window was chosen because of its desirable spectral properties and its widespread use in linear prediction literature.

Another important consideration was the number of predictor coefficients to be used. As a practical matter, it is best to choose the number of coefficients as small as possible because it saves computation time and there is less chance of filter instability due to finite arithmetic effects. It has been shown that to represent adequately the vocal tract under ideal circumstances, the memory of the model or filter must be equal to twice the time required for sound waves to travel from the glottis to the lips, i.e., $M = 2L/C$, where L is the length of the vocal tract and C is the speed of sound. Using the average vocal tract length of $L = 17$ cm and the speed of sound, $C = 34$ cm/ms, the memory required is 1 ms with a 10 kHz sampling rate, the number of predictor coefficients needed is equal to the sampling rate times the memory required, or 10 coefficients. In order to take into account the influences of the glottal waveform flow and lip radiation characteristics an additional two coefficients are necessary. Hence, twelve predictor coefficients were computed for each analysis frame thereby producing a twelve pole filter that accurately models the spectral properties of each speech interval.

The information necessary to choose the various parameters and methods for LPC processing were obtained from experimental results and several references, which include Markel and Gray, (1973) and Makhoul (1973), and Boll (1973).

C. LPC Data

With all the necessary parameters and methods chosen to analyze the speech, LPC processing of each word could then be undertaken. Computer program LPC was used to perform the speech analysis. A listing of the program can be found in Appendix A. The 256 point window was moved along each word at 256 sample shifts. Originally, the window was shifted by 128 points to create overlapping analysis frames. However, due to the averaging, significant differences in the distance measures, to be described in Chapter 5, were not found when the two methods were compared. Consequently, the 256 point shift was adopted to save processing time. The distorted word lengths were naturally defined to be the same as the corresponding master word lengths.

Processing of the windowed signal to obtain the LPC parameters was done using the Levinson Algorithm. A flow chart of the algorithm is shown in Figure 4-1. In order to start the algorithm, the normalized autocorrelation terms of the windowed signal had to be obtained. This was accomplished using the direct method of equation (2.7) as opposed to using the Fast Fourier Transform (FFT) because of the small number of lag terms needed (thirteen terms). The Levinson Algorithm was used to obtain the twelve predictor coefficients, (a_i), and the minimum total squared error, E_{min}. Each reflection coefficient (k_i) was

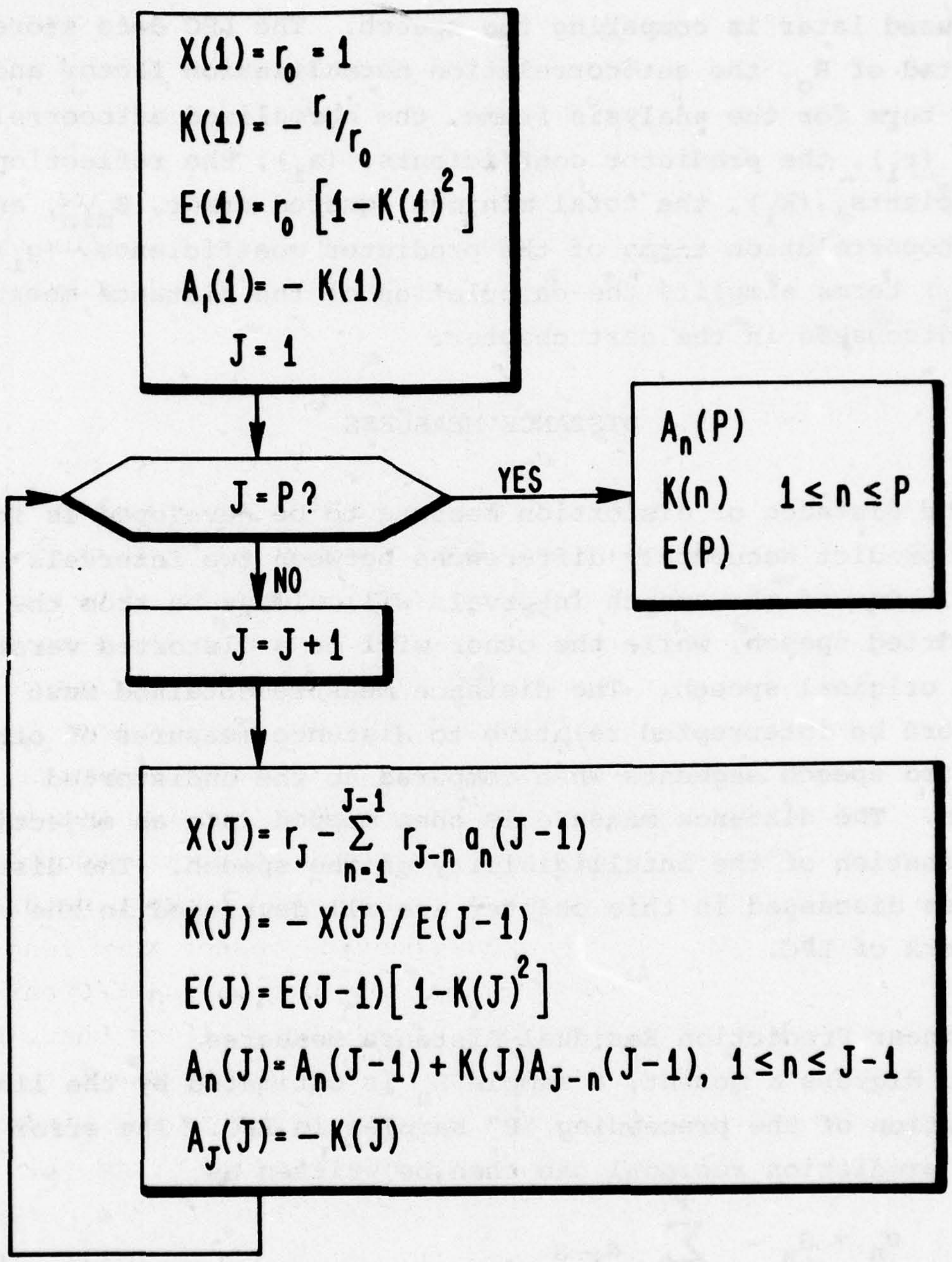


Figure 4-1. Flow diagram for LPC processing.

checked to insure that it was less than 0.99, thereby guaranteeing a stable filter.

All of the LPC information was then stored on magnetic tape to be used later in comparing the speech. The LPC data stored consisted of R_o, the autocorrelation normalization factor and energy term for the analysis frame, the normalized autocorrelation terms, (r_i), the predictor coefficients, (a_i), the reflection coefficients, (k_i), the total minimum squared error, E_{min}, and the autocorrelation terms of the predictor coefficients, (g_i). The (g_i) terms simplify the calculation of the distance measures to be discussed in the next chapter.

5. DISTANCE MEASURES

The distance or distortion measure to be developed is intended to predict accurately differences between two intervals of speech. One of the speech intervals will always be from the undistorted speech, while the other will be a distorted version of the original speech. The distance measure obtained must therefore be interrupted relative to distance measures of other distorted speech segments when compared to the undistorted version. The distance measure is then mapped into an objective determination of the intelligibility of the speech. The distance measures discussed in this chapter are all developed in the framework of LPC.

A. Linear Prediction Residual Distance Measures

To digress a moment, a sample S_n is estimated by the linear combination of the preceeding "P" samples in LPC. The error or linear prediction residual can then be written as

$$e_n = S_n - \sum_{i=1}^{P} a_i \, S_{n-1} \, , \tag{5.1}$$

where (a_i) are the predictor coefficients. These coefficients are obtained by choosing them so as to minimize the total squared error. The total squared error or linear prediction residual energy can be considered to be the output of an inverse filter $H(z)^{1}$, where

$$[H(z)]^{-1} = 1 + \sum_{i=1}^{P} a_i z^{-1} \quad . \tag{5.2}$$

$[H(z)]^{1}$ is the filter that minimizes the residual energy and H(z) corresponds to a smoothed spectral estimate of the data sequence (S_n) up to a scale factor representing the gain.

If (S_n) is passed through a different inverse filter, $[H'(z)]^{-1}$, of the form

$$[H'(z)]^{-1} = 1 + \sum_{i=1}^{P} a_i' z^{-1} \tag{5.3}$$

which minimizes the residual energy for some other data sequence (S_n'), the residual energy D, must be greater than or equal to the minimum residual energy E, i.e., $D \geq E$, with the equality holding if and only if $H(z) = H'(z)$. Assuming the data sequence (S_n) is obtained from an analysis frame of speech from the undistorted tape and (S_n') is the corresponding analysis frame from a distorted tape the difference between D and E is a measure of the distance between the two speech segments. (Unless otherwise identified, unprimed variables represent data from the master or undistorted tape.)

The dual of the above situation is also true. If (S_n') is sent through $[H(z)]^{1}$, the output will be D', while E' is the output of $[H'(z)]^{1}$ when (S_n') is sent through it. Again, $D' \geq E'$ with equality if and only if $H'(z) = H(z)$. As before, the difference between D' and E' can be considered to be a distance measure between the two speech segments.

E, E', D, and D' can all be written as a combination of the autocorrelation terms of (S_n) and (S_n') and the corresponding linear prediction coefficients (a_i) and (a_i'). Let

$$\underline{A}^T = (1, -a_i, -a_2, \ldots -a_p) \tag{5.4}$$

be the transpose of the linear prediction coefficient vector $\underline{A}$, and

$$\underline{\underline{R}} = r_{|i-k|} \qquad i,k = 0, 1, 2, \ldots, P \tag{5.5}$$

the normalized autocorrelation matrix. The four error terms can then be written as

$$E = \underline{A}^T \underline{\underline{R}}\, \underline{A} \tag{5.6}$$

$$E' = \underline{A}'^T \underline{\underline{R}}'\, \underline{A}' \tag{5.7}$$

$$D = \underline{A}'^T \underline{\underline{R}}\, \underline{A}' \tag{5.8}$$

$$D' = \underline{A}^T \underline{\underline{R}}'\, \underline{A} \tag{5.9}$$

where the primes signify variables from a distorted tape. The derivation of this can be found in Market and Gray (1973) and Boll (1974).

E and E' are calculated for each analysis frame through the Levinson Algorithm, but D and D' are calculated when a distorted tape is compared to the master tape. These calculations can be simplified because of the structure of $\underline{\underline{R}}$ and $\underline{\underline{R}}'$, the autocorrelation matrices, by calculating the autocorrelation terms of $\underline{A}$ and $\underline{A}'$, the linear prediction coefficient vectors. Using the symmetry of the autocorrelation terms of $\underline{A}$, and $\underline{A}'$, D and D', can be written as

$$D = \sum_{i=0}^{P} g_i\, r_i \tag{5.10}$$

$$D' = \sum_{i=0}^{P} g_i\, r_i' \tag{5.11}$$

where

$$g_i = 2 \cdot \sum_{k=0}^{P-|i|} \alpha_k\, \alpha_{k+i} \qquad (i = 1, 2, \ldots, P \tag{5.12}$$

$$g_o = \sum_{k=0}^{P} \alpha_k^2 \tag{5.13}$$

$$g_i' = 2 \cdot \sum_{k=0}^{P-|i|} \alpha_k' \alpha_{k+1}' \qquad (i = 1, 2, \ldots, \tag{5.14}$$

$$g_o' = \sum_{k=0}^{P} \alpha_k'^2 . \tag{5.15}$$

(α_i) and (α_1') are the P+1 terms in the vectors $\underline{A}$ and $\underline{A}'$ respectively. (g_i) and (g_i') can be calculated for each analysis frame right after the predictor coefficients are obtained and stored on the LPC data tape.

Each of the four error terms can be interpreted in the frequency domain. Using Parseval's Theorem, the total squared error, E, can be written as

$$E = \sum_n e_n^2 = \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} |E(\omega)|^2 \, d\omega \quad , \tag{5.16}$$

where $E(\omega)$ is obtained by substituting $z = e^{i\omega t}$ into $E(z)$. From Chapter 2, the minimum linear prediction error was found to be

$$E(z). = S(z)[H(z)]^{-1} \tag{5.17}$$

while the least squares estimate can be written as

$$\hat{E}(z) = \hat{S}(z)[H(z)]^{-1} . \tag{5.18}$$

substituting $z = e^{i\omega t}$ into (5.17) and (5.18), one obtains

$$E(\omega) = S(\omega) \; [H(\omega)]^{-1} \tag{5.19}$$

$$\hat{E}(\omega) = \hat{S}(\omega) \; [H(\omega)]^{-1} \; . \tag{5.20}$$

Rearranging (5.20) and substituting in $\hat{E}(z) = A$,

$$[H(\omega)]^{-1} = \frac{A}{\hat{S}(\omega)} . \tag{5.21}$$

Inserting (5.21) into (5.19),

$$E(\omega) = A \, \frac{S(\omega)}{\hat{S}(\omega)} \; . \tag{5.22}$$

substituting (5.22) into (5.16)

$$E = \frac{T\,A^2}{2\pi} \int_{-\pi/T}^{\pi/T} \frac{|S(\omega)|^2}{|\hat{S}(\omega)|^2} \, d\omega \tag{5.23}$$

But $|S(\omega)|^2$ and $|\hat{S}(\omega)|^2$ are just the corresponding power spectra, $P(\omega)$ and $\hat{P}(\omega)$, of the speech signal and its least squares linear prediction estimate. Therefore,

$$E = \frac{TA^2}{2\pi} \int_{-\pi/T}^{\pi/T} \frac{P(\omega)}{\hat{P}(\omega)} \, d\omega \; . \tag{5.24}$$

Similarly E' can be shown to be

$$E' = \frac{TA'^2}{2\pi} \int_{-\pi/T}^{\pi/T} \frac{P'(\omega)}{\hat{P}'(\omega)} \, d\omega \; . \tag{5.25}$$

D and D' can also be obtained by the same method and written as

$$D = \frac{TA'^2}{2\pi} \int_{-\pi/T}^{\pi/T} \frac{P(\omega)}{\hat{P}'(\omega)} \, d\omega \tag{5.26}$$

$$D' = \frac{TA^2}{2\pi} \int_{-\pi/T}^{\pi/T} \frac{P'(\omega)}{\hat{P}(\omega)} \, d\omega \; . \tag{5.27}$$

The distance measures D and D' are not all that pleasing when defined in the frequency domain. They compare the ratio differences between a true speech power spectrum and estimated power spectrum. A much more desirable measure would compare the ratio differences between the estimated power spectra of the undistorted speech and the distorted version of it. This can be done by taking the ratio of D to E and D' to E'. The ratio of each of these pairs of residual errors, D/E and D'/E', then defines two new distance measures which are much more appropriate. In both cases, the ratios are greater than or equal to one, with equality if and only if $H(z) = H'(z)$.

The ratios D/E and D'/E' are sometimes called likelihood ratios because under certain circumstances, they have been shown to be true likelihood ratios by Itakura (1975). As mentioned before, the frequency domain interpretation of the likelihood ratios gives a good justification for using them as distance measures. In the time domain

$$\frac{D}{E} = \frac{\sum_n D_n^2}{\sum_n e_n^2}\ , \tag{5.28}$$

where

$$\sum_n D_n^2 = \sum_n \left\{ s_n - \sum_{i=1}^{P} a_i' \, s_{n-i} \right\}^2 \tag{5.29}$$

$$\sum_n e_n^2 = \sum_n \left\{ s_n - \sum_{i=1}^{P} a_i \, s_{n-i} \right\}^2 \tag{5.30}$$

Gray and Markel (1976) have shown that D/E can be written in the frequency domain as

$$\frac{D}{E} = \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} \frac{|H(\omega)|^2}{|H'(\omega)|^2} \, d\omega \ , \tag{5.31}$$

where the substitution $z = e^{j\omega T}$ is made in the filters H(z) and $H'(z)$. Inverting (5.21) and its dual, one obtains

$$H(\omega) = \frac{\hat{S}(\omega)}{A} \tag{5.32}$$

$$H'(\omega) = \frac{\hat{S}'(\omega)}{A'} \tag{5.33}$$

Substituting (5.32) and (5.33) into (5.31)

$$\frac{D}{E} = \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} \frac{|\hat{S}(\omega)|^2}{A^2} \cdot \frac{A'^2}{|\hat{S}'(\omega)|^2} \, d\omega. \tag{5.34}$$

$$\frac{D}{E} = \frac{TA'^2}{2\pi A^2} \int_{-\pi/T}^{\pi/T} \frac{|\hat{S}(\omega)|^2}{|\hat{S}'(\omega)|^2} \, d\omega \tag{5.35}$$

Once again, the magnitude squared of the signal's spectrum is just its power spectrum, therefore

$$\frac{D}{E} = \frac{TA'^2}{2\pi A^2} \int_{-\pi/T}^{\pi/T} \frac{\hat{P}(\omega)}{\hat{P}'(\omega)} \, d\omega \tag{5.36}$$

Similarly

$$\frac{D'}{E'} = \frac{TA'^2}{2\pi A'^2} \int_{-\pi/T}^{\pi/T} \frac{\hat{P}'(\omega)}{\hat{P}(\omega)} \, d\omega \tag{5.37}$$

As can be seen from (5.36) and (5.37), D/E and D'/E' compute the differences between the estimate power spectra of the undistorted and distorted speech, while D and D' compared the true power spectra of the speech to estimates of the power spectra.

D/E and D'/E' will be used as the basic components in the distance measures discussed in B, below. Several methods are used to normalize D/E and D'/E' so that two distorted tapes can be compared relative to each other by comparing them only to the master or undistorted tape. For more information concerning the likelihood ratios above, see Gray and Markel (1976).

One additional measure that is considered here is derived as follows. Let $S_n + N_n = S_N'$. Then,

$$\sum_n \left[S_n - \sum a_i S_{n-i} + N_n - \sum a_i N_{n-i} \right] = \left[S_N' - \sum a_i S_{n-i} \right] \qquad (5.38)$$

This can be rewritten as

$$E + D_N' + [\text{cross products}] = D',$$

where D_N' is given by $\underline{A}^T \underline{\underline{R}}_N \underline{A}$ with the $\underline{\underline{R}}_N$ the autocorrelation matrix of the noise. Assuming the signal S_N and the noise N_n are uncorrelated, this simplifies to

$$E + D_N' = D'. \qquad (5.39)$$

Assuming the noise for any frame (N_n) is the same (statistically) except for a constant factor as the noise for a period during which voice is not present ($\tilde{N}_n$), we have

$$D_N' = k\, D_{\tilde{N}}' \qquad (5.40)$$

where k is a different constant for each frame. Since k must be positive, we have

$$k = \frac{|D' - E|}{|\; D_{\tilde{N}}' \;|} \qquad (5.41)$$

A large number of calculations showed that E/D_N' was very small compared to D'/D_N', and consequently, k was calculated using

$$k = D'/D_{\tilde{N}}' \qquad (5.42)$$

k was then averaged over all frames of all words to obtain $\overline{K}$.

A signal to noise ratio was determined using the peak signals during a PN sequence and the noise N from a quiet period between PN sequences, for each word group. The quantity

$$SNR = 10 \log_{10} S/N - 10 \log_{10} \bar{K} \tag{5.43}$$

was then used to calculate an AI score,

$$AI = \begin{cases} 0 & SNR \leq 0 \\ \frac{SNR}{30} \quad \text{if} & 0 < SNR < 30 \\ 1 & SNR \geq 30. \end{cases} \tag{5.44}$$

B. An Objective Intelligibility Measure

The quantities D'/E' and D/E were computed on a frame-by-frame basis using the computer program DISTMEA and stored along with other LPC data for use in developing an objective intelligibility measure. (See Appendix A.) The natural logarithms of D'/E' and D/E are respectively labeled E1 and E2 and relate directly to a decibel (dB) scale.

Under the assumption that the errors e'_n are independent Gaussian variables, Itakura derives the result that N_{eff} E1 is a chi-squared variable with P(=12) degrees of freedom. Here, because of the windowing, $N_{eff} \simeq 101$. However, since in general the e_n are correlated, it is assumed that the actual N_{eff} is smaller than this (101).

Instead of modifying N_{eff}, the procedure given in the next paragraphs was used to modify two thresholds. First, a lower threshold for E1 was taken as 0.82, based on an average of approximately three times the "barely perceptible" difference of Flanagan (1972) and three times the "barely perceptible" threshold used by Sambur and Jayant (1976). Values of E1 below 0.82 mean the frame is understood. An upper threshold of 2.46 (3x0.82) was used to decide that the frame was completely misunderstood. A linear relationship was used between 0.82 and 2.46.

A sample of noise was taken from the distorted tape being analyzed (actually the same samples used in the previous section to derive k). From this sample several frames were analyzed from which two frames were selected using the criteria of the largest and smallest values for the sum of the squares of the predictor coefficients, these two frames were used with the master tape to calculate (frame by frame) values of $E1N_1$ and $E1N_2$, where N signifies noise. These two values (for each frame) were then averaged to obtain E1N. If E1N <2.46 the thresholds were not changed. If E1N >2.46 the thresholds were changed to

$$0.82 + 0.82\ (E1N - 2.46) \text{ and}$$
$$2.46 + 0.82\ (E1N - 2.46).$$

To summarize, two thresholds T1 and T2 are defined. Using these, a linear measure is defined for each frame as

$$\begin{aligned} LM1 &= 1 && \text{if } E1 < T1 \\ &= 0 && \text{if } E1 > T2 \qquad (5.45) \\ &= \frac{T2-E1}{T2-T1} && \text{otherwise.} \end{aligned}$$

The linear measure LM1 of the above method was calculated for all frames of each word. Further, for each word an average of this measure was calculated for those frames for which $R_o \geq \bar{R}_o/2$, where $\bar{R}_o$ was the average of R_o for the word. This was designated LM1H. Similarly, an average for frames for which $R_o < \bar{R}_o/2$ was calculated and called LM1L. This divides the measure into two groups, one for frames with higher power and one for frames with lower power compared to the average power in the word. Frequently, although not always, the low power frames correspond to the unvoiced speech and the high power to the voiced speech. Finally, LM1H and LM1L were averaged over fifty words to form $\overline{LM1H}$ and $\overline{LM1L}$.

In a similar fashion, high and low values $\overline{E1H}$ and $\overline{E1L}$, and $\overline{E1NH}$ and $\overline{E1NL}$ were calculated in order to modify the average of the linear measures in the following way. If $\overline{E1H}$ (resp $\overline{E1L}$) is less than .82 no modification is made. If $\overline{E1NH}$ (resp $\overline{E1NL}$) is greater than 2.46 no modification is made. Otherwise $\overline{LM1H}$ (resp $\overline{LM1L}$) is multiplied by

$$\frac{\overline{E1NH} - 0.82}{2.46 - 0.82} \quad \left(\text{resp } \frac{\overline{E1NL} - 0.82}{2.46 - 0.82}\right).$$

The modified high and low measures were then averaged to form $\overline{LM1}$. This has the effect of weighting the low values more than the high values since only about 1/3 of the frames are low.

The correlation $C_{R_o R_o'}$, of R_o and R_o' was then calculated, and multiplied by $\overline{LM1}$ to form $\overline{CLM1}$. This value was averaged with the AI measure of the previous paragraphs to form ASQ, the objective articulation score.

The correlation correction was applied to account for fading signals. Several other methods were used which compared the signal levels on a word-by-word and frame-by-frame basis. These required considerably more computing time and gave essentially the same results as using the correlation factor.

The quantity AI is shown in Figure 5-1 plotted vs the subjective articulation scores (AS), the quantity $\overline{CLM1}$ is shown in Figure 5-2 plotted vs AS, and ASO, the average is shown in Figure 5-3 plotted vs AS. In Figure 5-3, the bars indicate the confidence limits about the subjective score.

While subjective intelligibility scores for isolated words are repeatable, there are some fluctuations in the scores. A listener panel is trained on a set of word groups with well established intelligibility scores and standard deviations. The average intelligibility of each training word group scored by individuals that make up a listener panel is always plus or minus one standard deviation of its actual intelligibility score. Also, the standard deviation of the listener panel for each training word group is approximately the same as its actual standard deviation. Using the training procedure, a listener panel will produce an intelligibility score within plus or minus

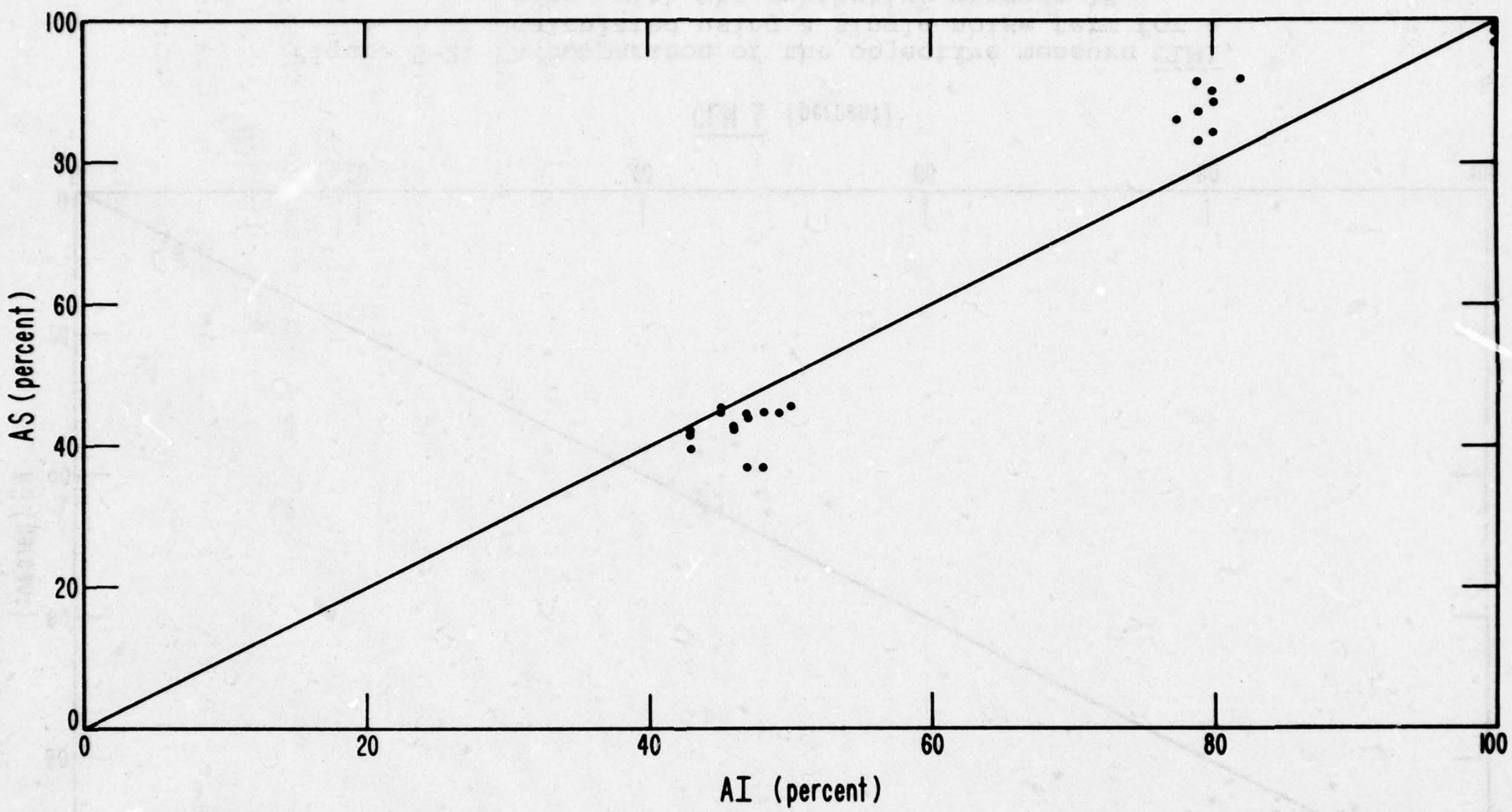


Figure 5-1. A comparison of an objective measure AI, based on signal-to-noise ratios with the subjective measure, AS.

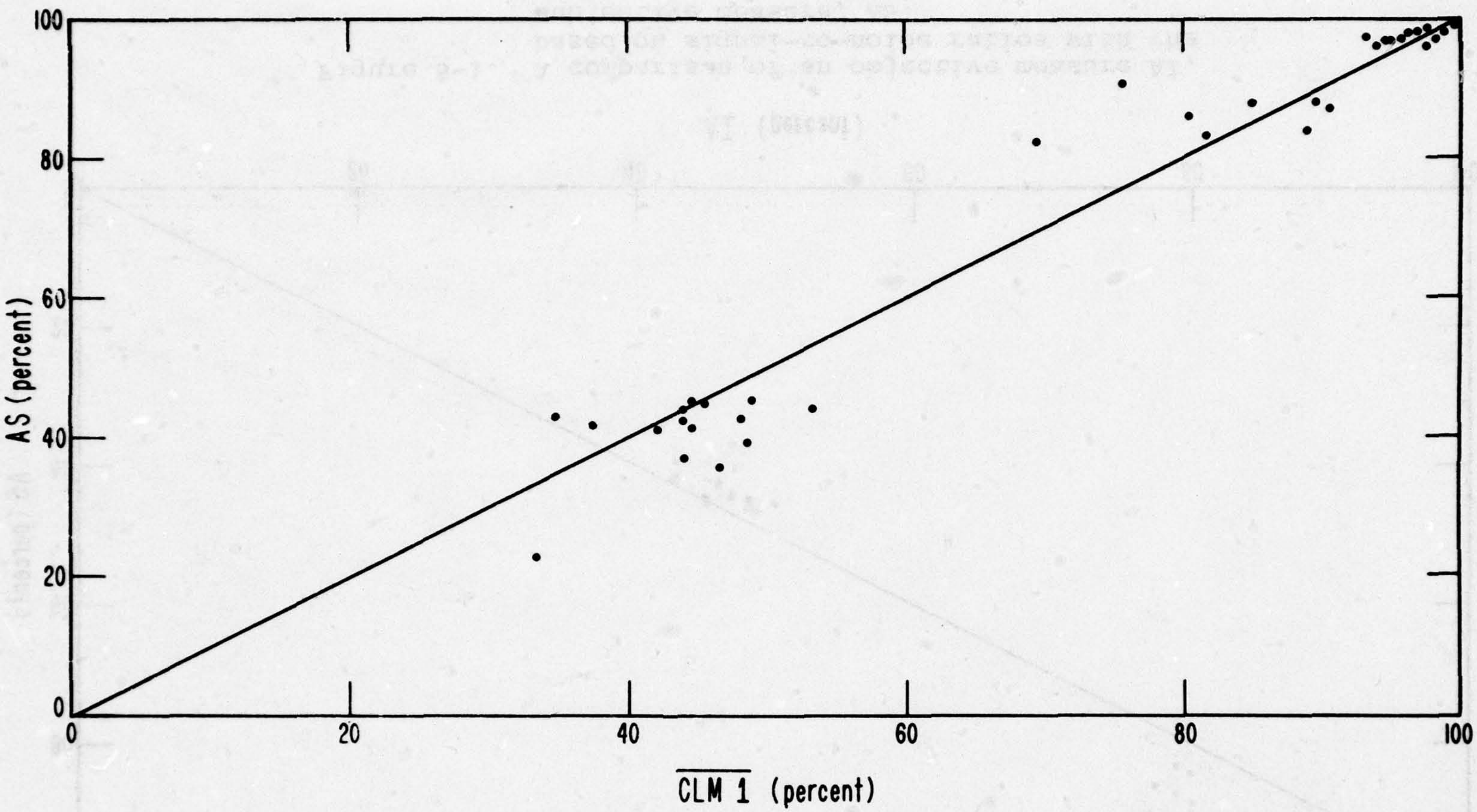


Figure 5-2. A comparison of the objective measure $\overline{\text{CLM1}}$, calculated using a single noise term for a tape, with the subjective measure AS.

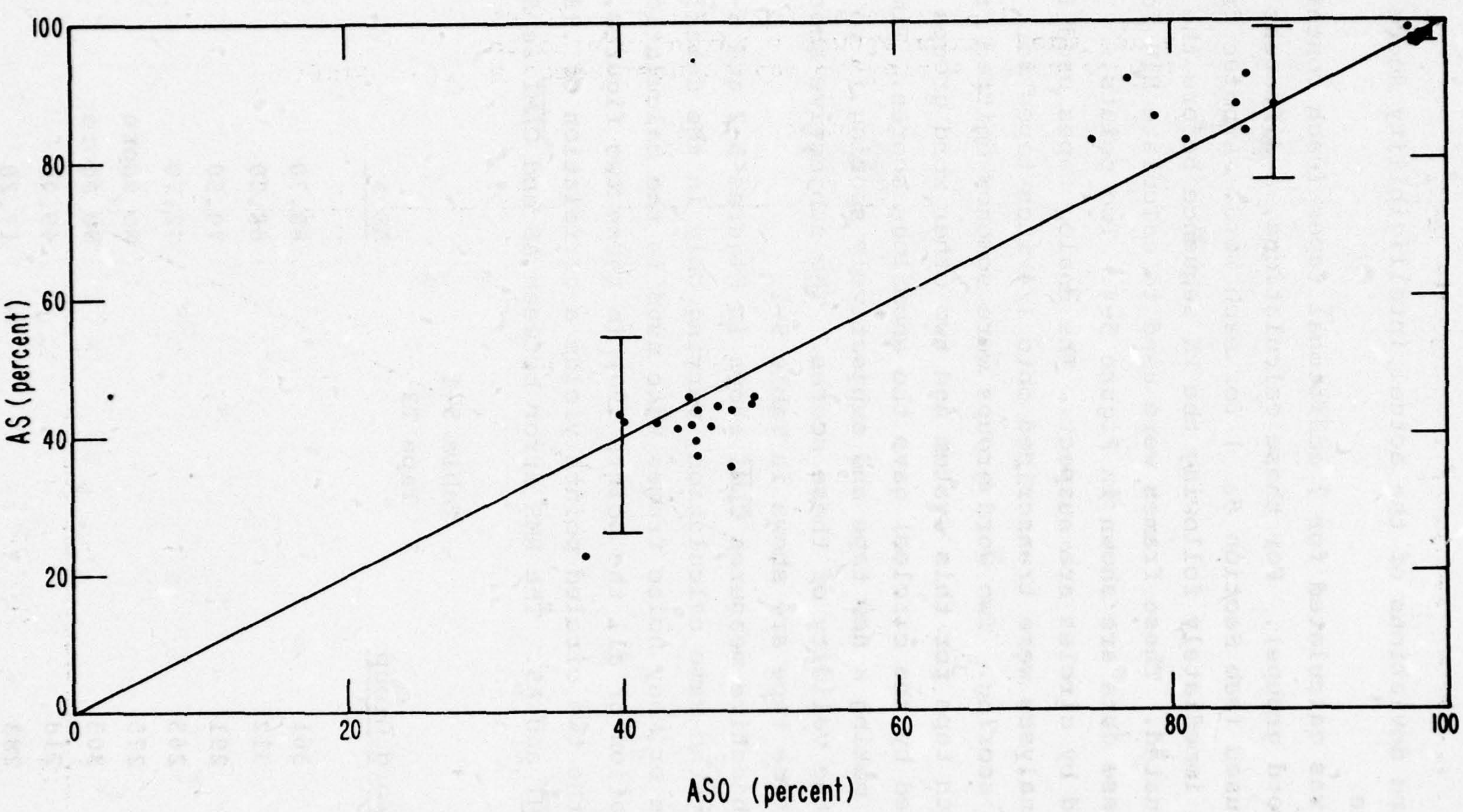


Figure 5-3. A comparison of the average of AI and $\overline{\text{CLM1}}$ of figures 5-1 and 5-2 with the subjective measure AS.

two standard deviations of the actual intelligibility score 95% of the time.

$\overline{\text{CLMI}}$ was calculated for 7 additional tapes (each containing eight 50-word groups). For these calculations, a different noise frame was used (see Section 5A) for each word. A noise frame was chosen immediately following the PN sequence before the word being evaluated. These frames were used to calculate E1N for the words. These data are shown in Figure 5-4. Two points, represented by circles are suspect. The analog tapes used for the data analyses were transcribed onto 1/4 inch tapes for subjective scoring. Two word groups were somehow omitted from the 1/4 inch tape for this system and two other word groups (represented by the circles) gave the anomalous scores. Time did not permit making a new tape and subjectively scoring it to determine the validity of these scores. The subjective scores for the entire tape are shown in Table 5-1.

The objective measures $\overline{\text{CLMI}}$ shown in Figures 5-2 and 5-4 are essentially the same calculation, varying only in the detail of whether one or many noise frames were used in the calculations. The combination of all the points (90) in these two figures, excluding the two circled points yields a correlation of .982 between $\overline{\text{CLMI}}$ and AS. The RMS error between AS and $\overline{\text{CLMI}}$ is 5.5(%).

Table 5-1

Tape T3

Word Group	AS %
361	60.70
312	68.00
291	70.50
265	72.70
275	No score
305	No score
214	45.00
283	17.20

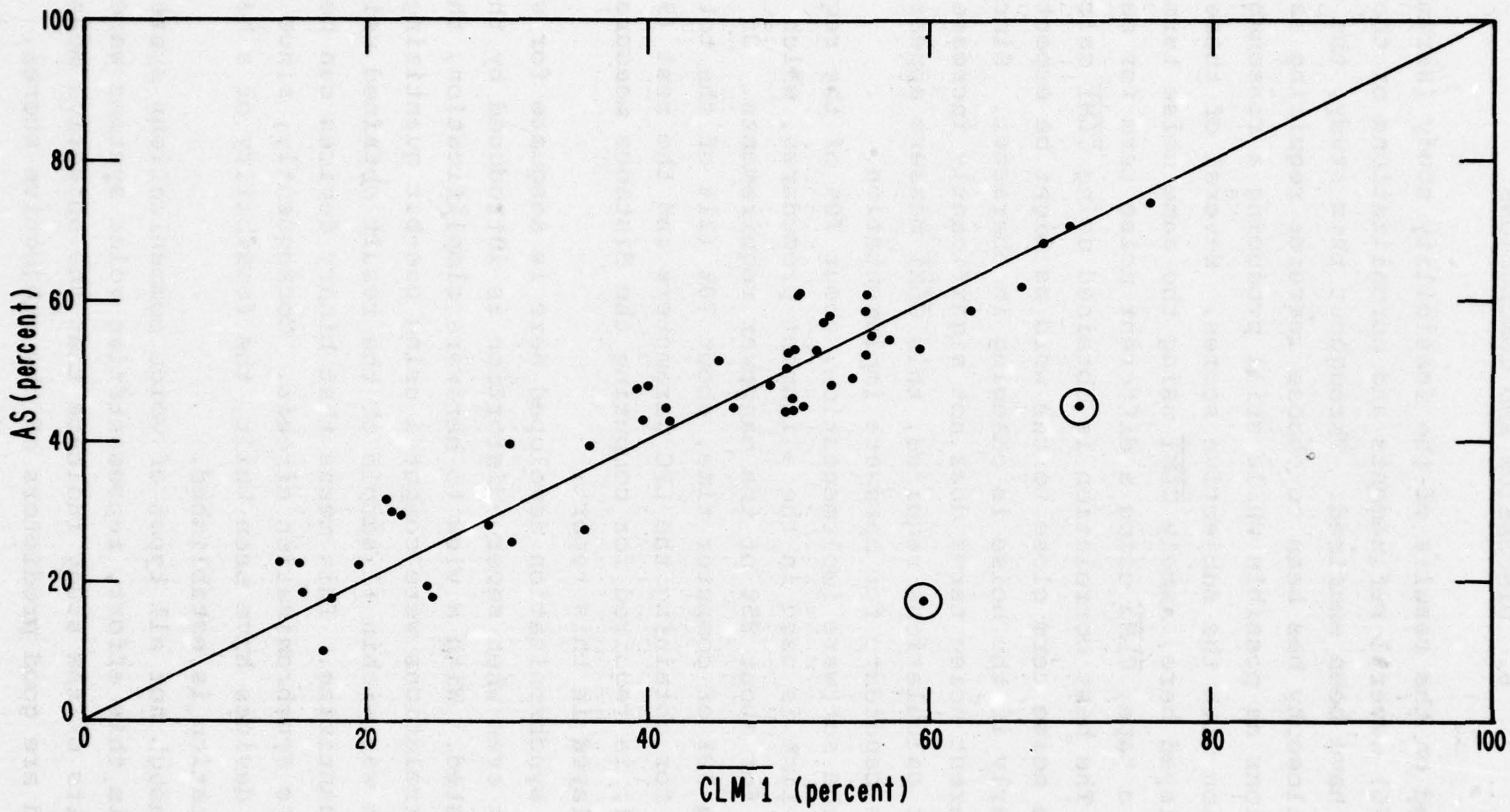


Figure 5-4. A comparison of the objective measure $\overline{CLM1}$, calculated using a noise term for each word, with the subjective measure AS.

6. DISCUSSION AND CONCLUSIONS

Based on the results of the feasibility study (Hartman and Boll, 1976) several refinements and normalizations of the distance measures have been examined. Throughout this study, the underlying philosophy has been to choose measures requiring as few calculations as possible while still producing a reasonable correlation with the subjective scores. Several of these measures are displayed here, namely $\overline{\text{CLM1}}$ using the same noise term for all words on a tape, $\overline{\text{CLM1}}$ using a different noise term for each word, and AI. The best correlation is obtained using $\overline{\text{CLM1}}$ calculated using the noise term close to the word as might be expected, particularly if the noise is changing in character. Since using the different noise terms does not significantly increase the number of calculations required, this $\overline{\text{CLM1}}$ measure appears to be the prime candidate for hardware implementation.

In the software implementation, about 70% of the required computer time is used in the alignment procedures, which also accounts for about 85% of the manpower requirements. Of the remaining 30% of computer time, about 70% (21% of the total) is required for obtaining the LPC parameters and the rest (9% of the total) is required for computing the distance measures which are displayed in this report.

The synchronization developed here is adequate for word alignment even when severe distortion is introduced by the system being tested. With a view to hardware simplification, the PN cross correlations were computed using one-bit quantizing and the alignment was within ± 1 sample of the result obtained using the 12 bit quantizing. This means that binary devices can be used in a hardware synchronization circuit. Consequently, since real time LPC devices have been built, the feasibility of a hardware implementation is established.

Although not all types of voice communications systems were studied in this effort, representative voice systems were used. The results of the study indicate that the objective measure(s) developed are good predictors of the subjective scores.

In order to enlarge the data base further, or to investigate other uses and modifications of these methods it appears that the most economical procedure is to develop a flexible hardware system.

10. REFERENCES

ANSI (1969), Methods for the calculation of the articulation index, Standard No. S3.5-1969, Am. National Standards Inst., Inc., 1430 Broadway, New York, NY 10018.

Atal, B.S. and S.L. Hanauer (1971), Speech analysis and synthesis by linear prediction of the speech wave, Journ. of the Acoust. Society of America, Vol. 50, April, pp. 637-655.

Boll, S.F. (1973), A priori digital speech analysis, Ph.D thesis, University of Utah, March.

Boll, S.F., (1974), Waveform comparison using LPC filtering, to be published.

Flanagan, J.L., (1972), Speech Analysis Synthesis and Perception, New York: Academic.

Gray, A.H. Jr., and J.D. Markel (1976), Distance measures for speech processing, IEEE Trans. on Acoustics, Speech, and Signal Processing, Vol. ASSP-24, October, pp. 380-391.

Grenander, U., and G. Szego (1958), Toeplitz forms and their applications, (Berkeley: University of California Press).

Hartman, W.J., and S.F. Boll (1976), Voice channel objective evaluation using linear predictive coding, FAA Report No. FAA-RD-75-189, August.

Hubbard, R.W., and W.J. Hartman (1974), Objective measurement techniques for evaluating voice communication channels, FAA Report No. FAA-RD-74-77, July.

Itakura, F. (1975), Minimum prediction residual principle applied to speech recognition, IEEE Trans. on Acoustics, Speech, and Signal Processing, Vol ASSP-23, February, pp. 67-72.

Kryter, K.D. (1962), Methods for the calculation and use of the articulation index, J. Acoust. Soc. of Am., 34, 1689-1697.

Kryter, K.D., and J.H. Ball (1964), A meter for measuring the performance of speech communication systems, Tech. Doc. Rept. No. ESD-TDR-64-674, Electronic Systems Division, Air Force Systems Command, L.G. Hanscom Field, Bedford, MA.

Markel, J.D., and A.H. Gray, Jr., (1976), Linear prediction of speech, (New York: Springer-Verlag).

Markel, J.D. and A.H. Gray, Jr., (1973), On autocorrelation equations as applied to speech analysis, IEEE Trans. on Audio and Electroacoustics, Vol. AU-21, April, pp. 69-79.

Makhoul, J. (1975), Linear prediction: a tutorial review, Proceedings of the IEEE, Vol. 63, April, pp. 561-580.

Makhoul, J. (1973), Spectral analysis of speech by linear prediction, IEEE Trans. on Audio and Electroacoustics, Vol. AU-21, June, pp 140-148.

Makhoul, J.I., and J.J. Wolf (1972), Linear prediction and the spectral analysis of speech, (Bolt, Beranek, and Newman, Inc. Report No. 2304, Cambridge).

Sambur, M.R., and N.S. Jayant (1976), LPC analysis/synthesis from speech inputs containing quantizing noise or additive white noise, IEEE Trans. on Acoustics, Speech, and Signal Processing, Vol. ASSP-24, No. 6, December.

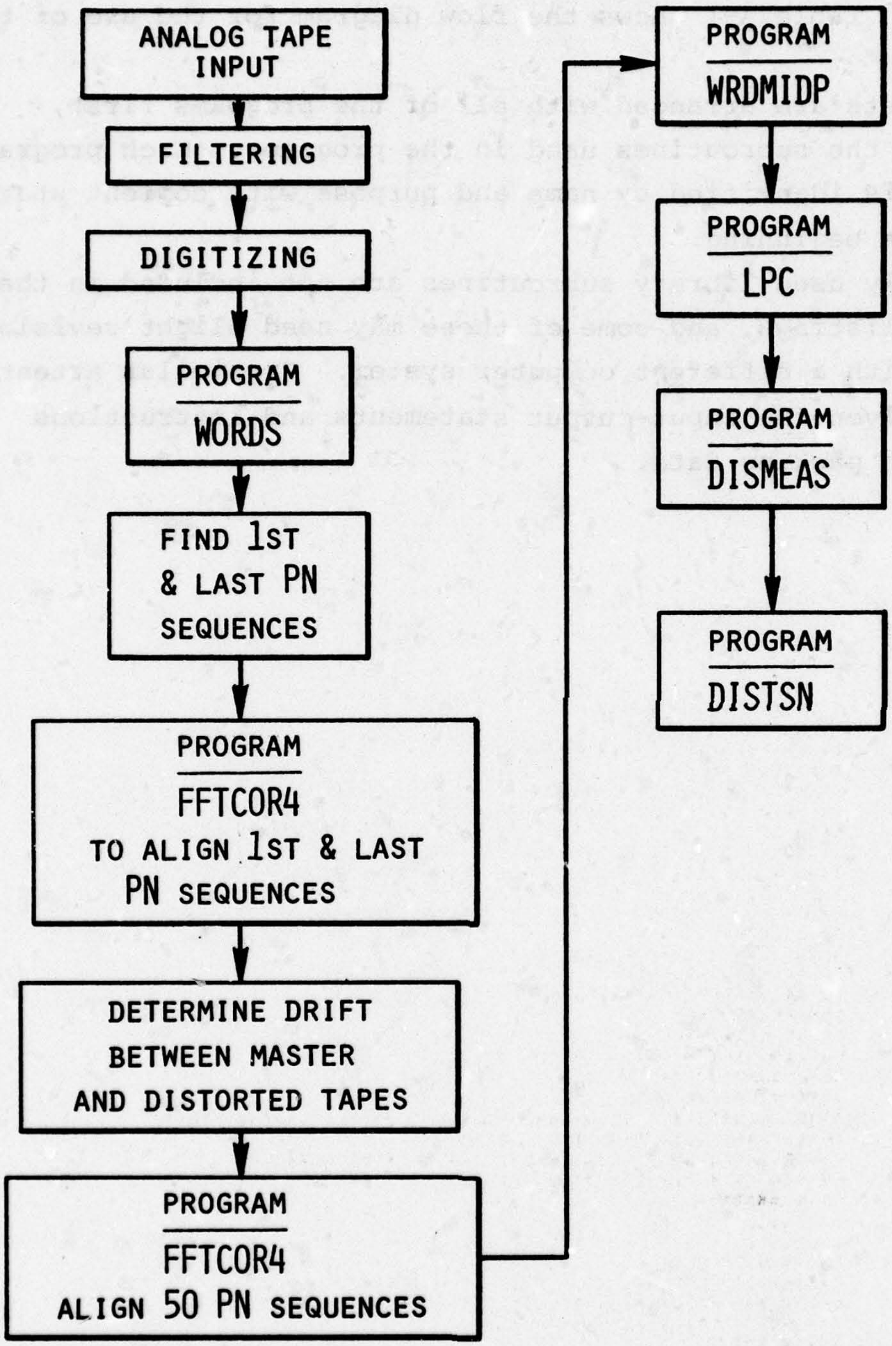
APPENDIX

This appendix contains the program lists for processing the voice data. Table A-1 shows the flow diagram for the use of the programs.

The lists are arranged with all of the programs first, followed by the subroutines used in the programs. Each program or subroutine is identified by name and purpose with comment statements at the beginning.

Commonly used library subroutines are not included in the subroutine listings, and some of these may need slight revision when used with a different computer system. Particular attention should be given the input-output statements and instructions dealing with packing data.

Table A-1. Flow Chart for Processing Voice Tapes



```
      PROGRAM WORDS(INPUT,OUTPUT,TAPE1)
C     PROGRAM WORDS FINDS THE LOCATION OF THE PN SEQUENCES AND WORDS OF
C     A PARTICULAR WORD GROUP BY FINDING THE MEAN AND STANDARD DEVIATION
C     OF CONSECUTIVE 125 SAMPLE BLOCKS.  THE STANDARD DEVIATION IS USED
C     AS AN ENERGY CRITERION IN ORDER TO LOCATE THE PN SEQUENCES AND WORDS.
      DIMENSION IN(80),IRECORD(400),ID(125),ITEMP(2000)
C     INITIALIZE VARIABLES.  *THOLD* IS A THRESHOLD USED TO DETERMINE IF
C     PART OF A PN SEQUENCE OR WORD IS PRESENT IN THE DATA.  *ISKIP* AND
C     *IEND* DETERMINE THE STARTING AND ENDING POINTS OF THE RECORDS TO
C     BE PROCESSED.  *KOUNT*, *ICYCLE*, AND *IPARTS* ARE COUNTING VARIABLES.
C     *ILENGTH* IS THE NUMBER OF SAMPLES TO BE PROCESSED AT A TIME AND
C     *IWORDS* IS THE NUMBER OF SAMPLES IN A TAPE RECORD.
      THOLD=32.0
      ISKIP=0
      IEND=3450
      KOUNT=1
      ICYCLE=0
      IPARTS=16
      ILENGTH=125
      IWORDS=400
      IREC=ISKIP
C     THE TAPE IS POSITIONED TO THE STARTING POINT FOR PROCESSING.
      IF (ISKIP .EQ. 0) GO TO 5
      DO 10 K10=1,ISKIP
      BUFFER IN (1,1) (IN(1),IN(80))
      IF (UNIT(1)) 10,20,10
   10 CONTINUE
    5 PRINT 55,ISKIP
   55 FORMAT (1H1,I5,*  RECORDS SKIPPED  *)
   30 IREC=IREC+1
C     FIVE CONSECUTIVE 80 WORD RECORDS ARE BUFFERED IN FOR PROCESSING.
C     EACH 60 BIT WORD IS UNPACKED INTO FIVE 12 BIT WORDS USING SUBROUTINE
C     UNPACK.  THIS YEILDS A TOTAL OF 2000 SAMPLES TO BE PROCESSED AT A TIME.
      BUFFER IN (1,1) (IN(1),IN(80))
      IF (UNIT(1)) 15,20,25
   25 PRINT 120,IREC
  120 FORMAT (*  RECORD  *,I5,*  HAS A PARITY ERROR  *)
      GO TO 30
   15 LL=LENGTH(1)
      CALL UNPACK (IRECORD,IN,LL)
      KINDEX=ICYCLE*IWORDS
      DO 12 K12=1,IWORDS
   12 ITEMP(K12+KINDEX)=IRECORD(K12)
      ICYCLE=ICYCLE+1
      IF (ICYCLE .LT. 5) GO TO 30
C     THE 2000 SAMPLES ARE BROKEN UP INTO SIXTEEN 125 SAMPLE BLOCKS.  FOR
C     EACH BLOCK, THE MEAN AND STANDARD DEVIATION ARE COMPUTED.  IF THE
C     STANDARD DEVIATION IS GREATER THAN OR EQUAL TO THE PRESET THRESHOLD,
C     THOLD, THE FIRST TEN SAMPLES, THE MEAN, THE STANDARD DEVIATION, AND
C     THE BLOCK NUMBER FOR THE SAMPLE BLOCK ARE PRINTED.  OTHERWISE THE
C     PROGRAM PROCEEDS TO THE NEXT SAMPLE BLOCK.
      DO 35 K35=1,IPARTS
      NINDEX=(K35-1)*ILENGTH
      DO 40 K40=1,ILENGTH
   40 ID(K40)=ITEMP(K40+NINDEX)
      SUM1=SUM2=0.0
      DO 45 K45=1,ILENGTH
      SUM1=SUM1+ID(K45)
   45 SUM2=SUM2+ID(K45)*ID(K45)
      AMEAN=SUM1/ILENGTH
      STDDEV=SQRT(ABS((ILENGTH*SUM2-SUM1*SUM1)/ILENGTH**2))
      IF (STDDEV .LT. THOLD) GO TO 50
      PRINT 130,KOUNT,(ID(I),I=1,10),AMEAN,STDDEV,KOUNT
  130 FORMAT (* *,I6,10I9,2F12.4,I9)
   50 KOUNT=KOUNT+1
   35 CONTINUE
      ICYCLE=0
C     THE NUMBER OF THE NEXT RECORD TO BE PROCESSED IS COMPARED TO THE
C     NUMBER OF THE LAST RECORD TO BE PROCESSED, IEND.  IF IT IS LESS THAN
C     IEND, THE PROGRAM CONTINUES.  OTHERWISE THE PROGRAM TERMINATES.
      IF (IREC .LT. IEND) GO TO 30
   20 PRINT 100
  100 FORMAT (1H1)
      PRINT 115,IREC
  115 FORMAT(I5)
      PRINT 110
  110 FORMAT (*  NORMAL TERMINATION  *)
      END
```

```
      PROGRAM FFTCOR4 (INPUT,OUTPUT,PUNCH,TAPE1,TAPE2)
C     PROGRAM FFTCOR4 COMPUTES THE CROSS-CORRELATION BETWEEN TWO SEQUENCES
C     OF DATA SAMPLES, WHICH IN THIS CASE ARE THE PN SEQUENCES.
      COMMON /FFT1 /A (4200), B (4200), C (4200), D (4200), D1 (5000)
      COMMON /FFT2 /ICOR (2100)
      DIMENSION ID(500),ITEMP(100),IN(100)
C     INITIALIZE VARIABLES.  *MPRINT* IS A PRINTING CONTROL VARIABLE AND
C     *NUMREC* IS THE NUMBER OF SAMPLES IN A TAPE RECORD.  *IASIZE1* AND
C     *IASIZE2* ARE COUNTING VARIABLES.  *NRECRED* AND *NREC1AA* KEEP
C     TRACK OF THE POSITION OF THE UNDISTORTED AND DISTORTED TAPES.
      MPRINT=0
      NUMREC=400
      IASIZE1 = 4200
      IASIZE2 = 5000
      PRINT 1504
 1504 FORMAT  (1H1)
  100 NRECRED = NREC1AA = 0
  105 DO 110 K10 = 1, IASIZE1
      A (K10) = B (K10) = C (K10) = D (K10) = 0.
  110 CONTINUE
      DO 115 K20 = 1, IASIZE2
      D1 (K20) = 0
  115 CONTINUE
C     THE CONTROL CARD IS READ THAT GIVES THE MIDPOINTS OF THE TWO SEQUENCES
C     TO BE PROCESSED AND THEIR LENGTH.  IF VARIABLE MIDPNT1 IS EQUAL TO
C     ZERO, THE PROGRAM TERMINATES.  INDEXES ARE THEN COMPUTED THAT
C     POSITION THE TWO TAPES IN ORDER TO GET THE DESIRED DATA.
      READ 1502,MIDPNT1,MIDPNT2,NUMEXP
 1502 FORMAT(3I8)
      IF (MIDPNT1 .EQ. 0) GO TO 495
      NUMBER = 2 * * NUMEXP
      INDEX1 = MIDPNT1 - NUMBER / 2
      INDEX2 = MIDPNT2 - NUMBER / 2
      ISTRTR1 = INDEX1 / NUMREC - NRECRED
      ISTRTR2 = INDEX2 / NUMREC - NREC1AA
      IORIGIN = INDEX1 - NUMREC * (INDEX1 / NUMREC)
      IORI1AA = INDEX2 - NUMREC * (INDEX2 / NUMREC)
C     THE TWO TAPES ARE POSITIONED TO THE START OF THE SEQUENCES TO BE
C     PROCESSED.
      IF(ISTRTR1 .LT. 1) GO TO 500
      DO 140 K40 = 1, ISTRTR1
      BUFFER IN (1,1) (ITEMP(1),ITEMP(80))
      IF (UNIT( 1))140, 485, 135
  135 PRINT 1508, K40
 1508 FORMAT  (*READ ERROR  TAPE 1 *, I10)
  140 CONTINUE
  500 CONTINUE
      NRECRED = NRECRED + ISTRTR1
      IF(ISTRTR2 .LT. 1) GO TO 505
      DO 155 K60 = 1, ISTRTR2
      BUFFER IN (2,1) (ITEMP(1),ITEMP(80))
      IF (UNIT( 2))155, 485, 150
  150 PRINT 1510, K60
 1510 FORMAT  (*READ ERROR  TAPE 2 *, I10)
  155 CONTINUE
  505 CONTINUE
      NREC1AA = NREC1AA + ISTRTR2
C     THE TWO DATA SEQUENCES ARE BUFFERED INTO THE PROGRAM, UNPACKED BY
C     SUBROUTINE UNPACK, AND STORED IN INDIVIDUAL ARRAYS.
      NRECIN = 2 * NUMBER / NUMREC
      IOFFSET = 0
      DO 180 K100 = 1, NRECIN
      BUFFER IN (1,1) (IN(1),IN(80))
      IF (UNIT( 1))170, 485, 165
  165 KPARITY = K40 + K100 - 1
      PRINT 1516,KPARITY
 1516 FORMAT  (*PARITY ERROR IN DATA RECORD. RUN ABORTED *, I10)
      GO TO 475
  170  CONTINUE
      LL=LENGTH(1)
      CALL UNPACK(ID,IN,NUMREC,LL)
      DO 175 K95 = 1, NUMREC
      D1 (K95 + IOFFSET) = ID (K95)
  175 CONTINUE
      IOFFSET = IOFFSET + NUMREC
```

```
  180 CONTINUE
      NRECRED = NRECRED + NRECIN
      DO 185 K102 = 1, NUMBER
      A (K102) = D1 (IORIGIN + K102)
  185 CONTINUE
      DO 190 K103 = 1, IASIZE2
      D1 (K103) = 0
  190 CONTINUE
      IOFFSET = 0
      DO 235 K130 = 1, NRECIN
      BUFFER IN (2,1) (IN(1),IN(80))
      IF (UNIT( 2))215, 485, 205
  205 KPARITY = K60 + K100 - 1
      PRINT 1516,KPARITY
      GO TO 475
  215 CONTINUE
      LL=LENGTH(2)
      CALL UNPACK(ID,IN,NUMREC,LL)
      DO 225 K125 = 1, NUMREC
      D1 (K125 + IOFFSET) = ID (K125)
  225 CONTINUE
      IOFFSET = IOFFSET + NUMREC
  235 CONTINUE
      NREC1AA = NREC1AA + NRECIN
      DO 255 K150 = 1, NUMBER
      B (K150) = D1 (IORI1AA + K150)
  255 CONTINUE
C     THE MEANS AND THE STANDARD DEVIATIONS OF THE TWO DATA ARRAYS ARE
C     COMPUTED.
      SUMX1A = SUMX1B = SUMX2A = SUMX2B = 0
      FNUMBER = NUMBER
      DENOM = FNUMBER * (FNUMBER - 1.)
      DO 405 K152 = 1, NUMBER
      SUMX1A = SUMX1A + A (K152)
      SUMX1B = SUMX1B + B (K152)
      SUMX2A = SUMX2A + A (K152) * A (K152)
      SUMX2B = SUMX2B + B (K152) * B (K152)
  405 CONTINUE
      AMEAN = SUMX1A / FNUMBER
      BMEAN = SUMX1B / FNUMBER
      STDA = (FNUMBER * SUMX2A - SUMX1A * SUMX1A) / DENOM
      STDA = SQRT (ABS (STDA))
      STDB = (FNUMBER * SUMX2B - SUMX1B * SUMX1B) / DENOM
      STDB = SQRT (ABS (STDB))
C     THE FORWARD FFT FOR BOTH DATA ARRAYS ARE COMPUTED USING SUBROUTINES
C     REVBIN, CFFTRC, AND RTRAN2T.
      NUMEXP = NUMEXP + 1
      NUMBER = 2 * * NUMEXP
      FNUMBER = NUMBER
      M = NUMEXP
      MM = NUMEXP - 1
      INV = 1
      NDIR = - 1
      SC = .5
      CALL REVBIN (A, B, M)
      CALL CFFTRC (A, B, M, SC, NDIR)
      CALL RTRAN2T (A, B, M, INV)
C     THE TRANSPOSE OF THE PRODUCT OF THE FFTS OF THE DATA ARRAYS IS
C     CALCULATED.
      IUPLIM = 1 + NUMBER / 2
      DO 415 K160 = 1, IUPLIM
      L160 = K160 + IUPLIM
      REAL = A (K160) * A (L160) + B (K160) * B (L160)
      COMP = B (K160) * A (L160) - A (K160) * B (L160)
      A (K160) = REAL
      B (K160) = COMP
  415 CONTINUE
C     THE INVERSE FFT IS TAKEN OF THE FFT PRODUCT USING SUBROUTINES
C     REALTRA, CFFTS, AND REORDER.
      INV = - 1
      NDIR = 1
      SC= 1. / FNUMBER
      CALL REALTRA (A, B, MM, NDIR, INV)
      CALL CFFTS (A, B, MM, SC, NDIR)
      CALL REORDER (A, B, MM)
```

```
C      THE LOCATION AND VALUE OF THE MAXIMUM CROSS-CORRELATION TERM IS
C      FOUND, ALONG WITH THE MIDPOINT OF THE DISTORTED DATA SEQUENCE THAT
C      LINES IT UP WITH THE UNDISTORTED DATA SEQUENCE.
       NUMHALF = NUMBER / 2
       NUMQTR = NUMBER / 4
       FNUMHAL = NUMHALF
       FNUMQTR = NUMQTR
       FMAXCOR = 0.
       FCOR = 0.
       DO 435 K170 = 1, NUMQTR
       FK170 = K170
       BLOWER = B (K170 + NUMQTR) / (FNUMQTR + FK170 - 1.)
       D1 (K170 + NUMQTR) = (BLOWER - AMEAN * BMEAN) / STDA / STDB
       RATIO1 = (FNUMQTR + FK170 - 1.) / FNUMHAL
       QUAN1 = RATIO1 * D1 (K170 + NUMQTR)
       IF (QUAN1 .LE. FMAXCOR) GO TO 425
       FMAXCOR = QUAN1
       FCOR = D1 (K170 + NUMQTR)
       LOCATIO = K170 + NUMQTR - NUMBER / 2 - 1
   425 BUPPER = A (K170) / (FNUMHAL - FK170 + 1.)
       D1 (K170 + NUMHALF) = (BUPPER - AMEAN * BMEAN) / STDA / STDB
       RATIO2 = (FNUMHAL - FK170 + 1.) / FNUMHAL
       QUAN2 = RATIO2 * D1 (K170 + NUMHALF)
       IF (QUAN2 .LE. FMAXCOR) GO TO 435
       FMAXCOR = QUAN2
       FCOR = D1 (K170 + NUMHALF)
       LOCATIO = K170 + NUMHALF - NUMBER / 2 - 1
   435 CONTINUE
C      THE LOCATION AND VALUE OF THE MAXIMUM CROSS-CORRELATION TERM AND
C      THE NEW DISTORTED TAPE SEQUENCE MIDPOINT ARE PRINTED ALONG WITH THE
C      INPUT VARIABLES.  THE NEW DISTORTED TAPE SEQUENCE MIDPOINT IS ALSO
C      PUNCHED ON A HOLLERITH CARD FOR FUTURE USE.
       PRINT 1502,MIDPNT1,MIDPNT2,NUMEXP
       IF(MPRINT .NE. 0) CALL PRINT(3,NUMBER)
       PRINT 1522, FMAXCOR, FCOR, LOCATIO
  1522 FORMAT  (2F12.3, 10X, I5)
       MIDPN=MIDPNT2-LOCATIO
       PRINT 2010,MIDPN
  2010 FORMAT (I10)
       PUNCH 2010,MIDPN
       PRINT 2000
  2000 FORMAT (3X,//)
C      THE PROGRAM IS SENT BACK TO READ THE NEXT CONTROL CARD.
       GO TO 105
   485 PRINT 1512
  1512 FORMAT  (*END OF FILE READ--JOB TERMINATED *)
C      PROGRAM TERMINATION SEQUENCE.
   495 PRINT 1514
  1514 FORMAT  (1H1, * NORMAL TERMINATION *)
   475 CONTINUE
       END
```

BEST AVAILABLE COPY

```
      PROGRAM WRDMIDP (INPUT,OUTPUT,PUNCH)
C     PROGRAM WRDMIDP CALCULATES THE MIDPOINTS OF THE WORDS OF THE
C     DISTORTED TAPE WITH RESPECT TO THE UNDISTORTED TAPE.
      DIMENSION IPN(50)
C     THE DRIFT PER SAMPLE OF THE DISTORTED TAPE IS CALCULATED FROM *DIFF*,
C     THE DIFFERENCE IN THE NUMBER OF SAMPLES BETWEEN THE UNDISTORTED AND
C     DISTORTED TAPE, AND *TOTAL*, THE TOTAL NUMBER OF SAMPLES IN THE
C     DISTORTED WORD GROUP TAPE.
      READ 50,DIFF,TOTAL
   50 FORMAT(2F10.1)
      DRIFT=DIFF/TOTAL
      PRINT 35
   35 FORMAT (1H1)
C     THE 50 UNDISTORTED PN SEQUENCE MIDPOINTS ARE READ IN FOR THE
C     PARTICULAR WORD GROUP.
      DO 10 I=1,50
      READ 15,MIDPN
   15 FORMAT (I10)
   10 IPN(I)=MIDPN
C     USING THE UNDISTORTED PN SEQUENCE MIDPOINTS, THE DISTANCE BETWEEN
C     THE PN SEQUENCE AND THE WORD OF THE UNDISTORTED TAPE, AND THE DRIFT
C     PRESENT, THE WORD MIDPOINTS OF THE DISTORTED TAPE ARE COMPUTED.
      DO 20 J=1,50
      READ 25,NWORDPN,N256
   25 FORMAT (I10,I5)
      IDRIFT=DRIFT*NWORDPN
      MIDWORD=IPN(J)+NWORDPN+IDRIFT
C     THE RESULTS ARE PRINTED AND PUNCHED ON HOLLERITH CARDS FOR FUTURE USE.
      PRINT 30,MIDWORD,N256
   30 FORMAT (I10,I5)
      PUNCH 30,MIDWORD,N256
   20 CONTINUE
      END
```

```
      PROGRAM LPC (INPUT,OUTPUT,TAPE1,TAPE2=/680)
C     PROGRAM LPC DOES THE FRAME BY FRAME LPC PROCESSING OF EACH WORD OF
C     A PARTICULAR WORD GROUP AND STORES THE INFORMATION ON MAGNETIC TAPE.
C     THE STORED DATA CONSISTS OF THE ENERGY TERM, THE NORMALIZED
C     AUTOCORRELATION TERMS, THE PREDICTOR COEFFICIENT AUTOCORRELATION
C     TERMS, THE PREDICTOR COEFFICIENTS, THE REFLECTION COEFFICIENTS, AND
C     THE MINIMUM SQUARED ERROR.
      COMMON A(256),ACOR(13),PCOEF(12,12),GAM(13),ERROR
      DIMENSION IN(80),ID(400),ITEMP(8400),W(256)
C     INITIALIZE VARIABLES. *NUMBER* IS EQUAL TO THE ANALYSIS FRAME SIZE.
C     *NPC* IS EQUAL TO THE NUMBER OF PREDICTOR COEFFICIENTS DESIRED.
C     *IWORDS* IS THE NUMBER OF SAMPLES IN A TAPE RECORD, AND *PI* IS
C     THE UNIVERSAL CONSTANT.
      NUMBER=256
      NUMHALF=NUMBER/2
      NPC=12
      IWORDS=400
      TW=NUMBER
      PI=3.141592653589
C     THE HAMMING WINDOW TO BE USED ON THE SPEECH SAMPLES IS GENERATED.
      DO 5 I=1,NUMHALF
      WH=(1.0/(1.08*TW))*(0.54+0.46*COS(2.0*PI*I/TW))
    5 W(128+I)=W(129-I)=WH
C     MAGNETIC TAPE IDENTIFICATION VARIABLES ARE DEFINED.
      READ 3,NFILE,NTAPE,NTYPE
    3 FORMAT (3I10)
   50 NRECORD=0
      NWORD=1
      PRINT 6
    6 FORMAT (1H1)
C     THE MIDPOINT AND LENGTH, IN TERMS OF ANALYSIS FRAMES, OF THE WORD
C     TO BE PROCESSED IS READ INTO THE PROGRAM.  IF THE WORD MIDPOINT IS
C     EQUAL TO ZERO, THE PROGRAM TERMINATES.  INDEXES ARE COMPUTED THAT
C     POSITION THE TAPE IN ORDER TO GET THE DATA SAMPLES FOR THE WORD.
  100 READ 105,MIDPNT,N256
  105 FORMAT (I10,I5)
      IF (MIDPNT .EQ. 0) GO TO 20
      MULT=N256
      MLENGTH=NUMBER*N256
      PRINT 7,MIDPNT,MLENGTH,N256
    7 FORMAT (2I10,I5,/)
      INDEX=MIDPNT-MLENGTH/2
      ISKIP=INDEX/IWORDS-NRECORD
      IORIG=INDEX-IWORDS*(INDEX/IWORDS)
C     ALPHA IS DEFINED, WHICH IS USED AS AN IDENTIFIER ON THE MAGNETIC TAPE.
      ENCODE (10,80,ALPHA) NTYPE
   80 FORMAT (I7,3X)
C     THE TAPE IS POSITIONED TO THE START OF THE WORD TO BE PROCESSED.
      IF (ISKIP .LT. 1) GO TO 10
      DO 15 K15=1,ISKIP
      BUFFER IN (1,1) (IN(1),IN(80))
      IF (UNIT(1)) 15,20,25
   25 NPARITY=K15+NRECORD
      PRINT 30,NPARITY
   30 FORMAT (*  PARITY ERROR IN RECORD  *,I5)
   15 CONTINUE
      NRECORD=NRECORD+ISKIP
C     THE SAMPLES OF THE WORD ARE BUFFERED INTO THE PROGRAM, UNPACKED BY
C     SUBROUTINE UNPACK, AND STORED IN AN ARRAY.
   10 NREC=2*MLENGTH/IWORDS
      IOFFSET=0
      DO 35 K35=1,NREC
      BUFFER IN (1,1) (IN(1),IN(80))
      IF (UNIT(1)) 33,20,40
   40 KPARITY=K35+NRECORD
      PRINT 45,KPARITY
   45 FORMAT(*  PARITY ERROR IN RECORD  *,I5,*  RUN ABORTED  *)
      GO TO 75
   33 LL=LENGTH(1)
      CALL UNPACK (ID,IN,LL)
      DO 55 K55=1,IWORDS
   55 ITEMP (IOFFSET+K55)=ID(K55)
   35 IOFFSET=IOFFSET+IWORDS
      NRECORD=NRECORD+NREC
```

```
C     ONE BY ONE THE ANALYSIS FRAMES OF THE WORD ARE WINDOWED BY THE HAMMING
C     WINDOW AND PROCESSED BY SUBROUTINES AUTOCOR, PRECOEF, AND GAMMA.
      DO 60 K60=1,MULT
      NINDEX=IORIG+(K60-1)*NUMBER
      DO 65 K65=1,NUMBER
   65 A(K65)=ITEMP(K65+NINDEX)*W(K65)
      CALL AUTOCOR (NUMBER,NPC)
      CALL PRECOEF (NPC)
      CALL GAMMA (NPC)
C     THE LPC INFORMATION FOR EACH FRAME IS STORED ON MAGNETIC TAPE.  THE
C     FIRST SIX VARIABLES ARE FOR IDENTIFICATION PURPOSES. ACOR(1) IS THE
C     ENERGY TERM, ACOR(J), J=2,13 ARE THE 12 NORMALIZED AUTOCORRELATION
C     TERMS, GAM(J), J=1,13 ARE THE PREDICTOR COEFFICIENT AUTOCORRELATION
C     TERMS, PCOEF(J,12), J=1,12 ARE THE PREDICTOR COEFFICIENTS, PCOEF(J,J),
C     J=1,12 ARE THE REFLECTION COEFFICIENTS, AND ERROR IS THE MINIMUM
C     SQUARED ERROR.
      WRITE (2,200) ALPHA,NFILE,NTAPE,NWORD,MULT,K60,(ACOR(J),J=1,13),(G
     1AM(J),J=1,13),(PCOEF(J,12),J=1,12),(PCOEF(J,J),J=1,12),ERROR
  200 FORMAT (A10,5I5,E16.10,12F12.10,13E14.8,12E14.8,12F10.8,E12.6)
   60 CONTINUE
      END FILE 2
      BACKSPACE 2
      NWORD=NWORD+1
      NTYPE=NTYPE+1
C     THE PROGRAM IS SENT BACK TO READ THE NEXT WORD CONTROL CARD.
      GO TO 100
C     PROGRAM TERMINATION SEQUENCE.
   20 PRINT 70
   70 FORMAT (1H1,* NORMAL TERMINATION *)
   75 CONTINUE
      END FILE 2
      END
```

BEST AVAILABLE COPY

```
      PROGRAM DISMEAS(INPUT,OUTPUT,TAPE1=/690,TAPE2=/680,TAPE3=/175)
C     PROGRAM DISMEAS COMPUTES THE RESIDUAL DISTANCE MEASURES D*, D, D*/E*,
C     AND D/E AND STORES THEM ON MAGNETIC TAPE ALONG WITH THE SUM OF THE
C     SQUARES OF THE PREDICTOR COEFFICIENTS, THE ENERGY TERMS, AND THE
C     MINIMUM SQUARE ERROR TERMS OF THE UNDISTORTED AND DISTORTED TAPES
C     ON A FRAME BY FRAME BASIS.
      DIMENSION ACOR(13),GAM(13),A(12),RC(12)
      DIMENSION DACOR(13),DGAM(13),DA(12),DRC(12)
      DIMENSION E(4),ASUM(8),B(51,8)
C     INITIALIZE VARIABLES.  *MWORD* IS A COUNTING VARIABLE.
      MWORD=0
      PRINT 42
   42 FORMAT(1H1)
C     THE LENGTH, IN ANALYSIS FRAMES, OF THE WORD TO BE PROCESSED IS READ
C     INTO THE PROGRAM.  IF THE WORD LENGTH IS EQUAL TO ZERO, THE PROGRAM
C     JUMPS OUT OF THE PROCESSING LOOP.
   50 READ 10,N256
   10 FORMAT(I5)
      IF (N256 .EQ. 0) GO TO 20
      MWORD=MWORD+1
      NMULT=N256
      DO 38 I=1,8
   38 ASUM(I)=0.0
C     FRAME BY FRAME, THE UNDISTORTED AND DISTORTED SPEECH LPC DATA IS
C     READ INTO THE PROGRAM.
      DO 15 K15=1,NMULT
      READ (1,200) ALPHA,NFILE,NTAPE,NWORD,MULT,K60,(ACOR(J),J=1,13),(GA
     1M(J),J=1,13),(A(J),J=1,12),(RC(J),J=1,12),ERROR
  200 FORMAT (A10,5I5,E16.10,12F12.10,13E14.8,12E14.8,12F10.8,E12.5)
      READ (2,200) ALPHA,NFILE,NTAPE,NWORD,MULT,K60,(DACOR(J),J=1,13),(D
     1GAM(J),J=1,13),(DA(J),J=1,12),(DRC(J),J=1,12),DERROR
C     THE RESIDUAL DISTANCE MEASURES D*, D, D*/E*, AND D/E ARE CALCULATED
C     AND STORED ON MAGNETIC TAPE ALONG WITH SIX IDENTIFICATION VARIABLES,
C     THE SUM OF THE SQUARES OF THE PREDICTOR COEFFICIENTS, THE ENERGY
C     TERMS, AND THE MINIMUM SQUARE ERROR TERMS OF THE UNDISTORTED AND
C     DISTORTED TAPES FOR EACH ANALYSIS FRAME.
      SUM1=GAM(1)
      SUM2=DGAM(1)
      DO 30 I=2,13
      SUM1=SUM1+GAM(I)*DACOR(I)
   30 SUM2=SUM2+DGAM(I)*ACOR(I)
      E(1)=SUM1
      E(2)=SUM2
      E(3)=E(1)/DERROR
      E(4)=E(2)/ERROR
      DO 32 I=1,4
   32 ASUM(I)=ASUM(I)+E(I)
      ASUM(5)=ASUM(5)+ACOR(1)
      ASUM(6)=ASUM(6)+DACOR(1)
      ASUM(7)=ASUM(7)+ERROR
      ASUM(8)=ASUM(8)+DERROR
      WRITE (3,300) ALPHA,NFILE,NTAPE,NWORD,MULT,K60,E(1),E(2),E(3),E(4)
     1,GAM(1),DGAM(1),ACOR(1),DACOR(1),ERROR,DERROR
  300 FORMAT(A10,5I5,10E14.8)
   15 CONTINUE
      END FILE 3
      BACKSPACE 3
C     THE WORD AVERAGE OF THE RESIDUAL DISTANCE MEASURES, THE ENERGY
C     TERMS, AND THE ERROR TERMS ARE COMPUTED.
      DO 78 I=1,8
   78 B(MWORD,I)=ASUM(I)/NMULT
      IF (EOF(1)) 20,25
   25 IF (EOF(2)) 20,45
C     THE PROGRAM IS SENT BACK TO READ THE NEXT WORD CONTROL CARD.
   45 GO TO 50
   20 CONTINUE
      IF(MWORD .EQ. 0) MWORD=1
      MW=MWORD+1
C     THE WORD GROUP AVERAGE OF THE RESIDUAL DISTANCE MEASURES, THE ENERGY
```

```
      C     TERMS, AND THE ERROR TERMS ARE CALCULATED.
            DO 90 I=1,8
            SUM=0.0
            DO 91 J=1,MWORD
         91 SUM=SUM+B(J,I)
         90 B(MW,I)=SUM/MWORD
      C     THE WORD AVERAGE AND THE WORD GROUP AVERAGE OF THE RESIDUAL DISTANCE
      C     MEASURES, THE ENERGY TERMS, AND THE ERROR TERMS ARE PRINTED.
            PRINT 75
         75 FORMAT(90H      COUNT      AR*A      A*RA*      E1/E*      E2/E
           1  RC      RS*      EMAG      EMAG*)
            DO 64 J=1,MWORD
         64 PRINT 65,J,(B(J,I),I=1,8)
         65 FORMAT(I10,8E10.4)
            PRINT 53
         53 FORMAT(/)
            PRINT 66
         66 FORMAT(*  WORD GROUP AVERAGE  *)
            PRINT 67,(B(MW,I),I=1,8)
         67 FORMAT(10X,8E10.4)
      C     PROGRAM TERMINATION SEQUENCE.
            PRINT 70
         70 FORMAT (1H1,* NORMAL TERMINATION *)
            END FILE 3
            END
```

```
      PROGRAM DISTKN(INPUT,OUTPUT,TAPE2=/135,TAPE3=/175)
C     PROGRAM DISTKN COMPUTES THE *K* FACTOR USED IN THE *AI* DISTANCE
C     MEASURE AND PRINTS IT FOR EACH WORD AND THE WORD GROUP AVERAGE FOR
C     EACH PAIR OF NOISE AND SIGNAL FILES READ IN.
      DIMENSION E(2),NW(51),FNAME(5)
      DIMENSION FDAME(5),D(2),S(30)
      DIMENSION A(30),R(30),DM(2),NCT(2)
      DIMENSION Q1(51,5),Q2(51,5),Q3(51,5)
      DIMENSION NQ1(51,5),NQ2(51,5),NQ3(51,5)
C     INITALIZE VARIABLES. *MWORD* IS A COUNTING VARIABLE.
      MWORD=0
C     THE LENGTH, IN ANALYSIS FRAMES, OF THE 50 WORDS IN THE WORD GROUP
C     BEING PROCESSED ARE READ INTO THE PROGRAM.
   50 READ 10,N256
   10 FORMAT(I5)
      IF(N256 .EQ. 0) GO TO 20
      MWORD=MWORD+1
      NMULT=N256
      NW(MWORD)=NMULT
      GO TO 50
   20 CONTINUE
      MW=MWORD+1
C     A TOTAL OF *IFILE* NOISE AND SIGNAL DATA FILES ARE READ INTO THE PROGRAM.
      IFILE=5
      READ 5,(FNAME(I),I=1,5)
      READ 5,(FDAME(I),I=1,5)
    5 FORMAT(5A7)
C     EACH OF THE PAIRS OF NOISE AND SIGNAL DATA FILES ARE PROCESSED.
      DO 25 K25=1,IFILE
      LFN=LFORM(FNAME(K25))
      CALL PFMATCH(5LTAPE3,LFN,0,0,0,0,1)
      LFN=LFORM(FDAME(K25))
      CALL PFMATCH(5LTAPE2,LFN,0,0,0,0,1)
C     EACH OF THE WORDS IN THE WORD GROUP FILES ARE PROCESSED.
      DO 30 K30=1,MWORD
      NMULT=NW(K30)
      SUM1=SUM2=0.0
      DO 15 K15=1,NMULT
C     FRAME BY FRAME, THE NOISE AND SIGNAL LPC DATA IS READ INTO THE PROGRAM.
      READ (2,200) ALPHA,NFILE,NTAPE,NWORD,MULT,K60,Z1,Z2,D(1),D(2),Y1,Y
     12,Y3,Y4,Y5,Y6
  200 FORMAT(A10,5I5,10E10.4)
      READ (3,300) ALPHA,NFILE,NTAPE,NWORD,MULT,K60,X1,X2,E(1),E(2),A2,D
     1A2,R0,DR0,ERROR,DERROR
  300 FORMAT(A10,5I5,10E14.8)
C     THE *K* FACTOR IS COMPUTED FOR EACH FRAME.
      A(K15)=X1
      S(K15)=Z1
      R(K15)=R
      SUM1=SUM1+R0
      SUM2=SUM2+Y1/Z1
   15 CONTINUE
C     HALF THE AVERAGE ENERGY AND THE AVERAGE *K* FACTOR OF THE WORD
C     BEING PROCESSED ARE CALCULATED.
      RAVG=SUM1/NMULT
      RAVG2=RAVG/2.0
      Q1(K30,K25)=SUM2/NMULT
      NQ1(K30,K25)=NMULT
C     SUBROUTINE HILOW IS CALLED TO COMPUTE THE AVERAGE *K* FACTOR FOR THE
C     HIGH AND LOW ENERGY FRAMES OF THE WORD BEING PROCESSED.
      CALL HILOW(NMULT,RAVG2,A,R,S,DM,NCT)
      Q2(K30,K25)=DM(1)
      Q3(K30,K25)=DM(2)
      NQ2(K30,K25)=NCT(1)
      NQ3(K30,K25)=NCT(2)
   30 CONTINUE
C     THE AVERAGE *K* FACTOR OF THE WORD GROUP FILE FOR THE HIGH, LOW, AND
C     TOTAL ANALYSIS FRAMES ARE FOUND.
      SUM1=SUM2=SUM3=0.0
      NC1=NC2=NC3=0
      DO 23 J=1,MWORD
      SUM1=SUM1+Q1(J,K25)
      SUM2=SUM2+Q2(J,K25)
      SUM3=SUM3+Q3(J,K25)
```

```
      NC1=NC1+NQ1(J,K25)
      NC2=NC2+NQ2(J,K25)
      NC3=NC3+NQ3(J,K25)
   23 CONTINUE
      Q1(MW,K25)=SUM1/MWORD
      Q2(MW,K25)=SUM2/MWORD
      Q3(MW,K25)=SUM3/MWORD
      NQ1(MW,K25)=NC1
      NQ2(MW,K25)=NC2
      NQ3(MW,K25)=NC3
   25 CONTINUE
C     THE 3 AVERAGE *K* FACTORS FOR EACH WORD OF EACH WORD GROUP FILE
C     ARE PRINTED.
      PRINT 40
   40 FORMAT(1H1)
      PRINT 80,NFILE
   80 FORMAT(*  WORD GROUP NO.   *,I5)
      PRINT 75
   75 FORMAT(65H   VARIABLE        35.4       19.2        9.6          FM
     1  AM)
      DO 64 J=1,MWORD
      PRINT 69,J
   69 FORMAT(*  WORD NUMBER    *,I5)
      PRINT 61,(Q1(J,I),I=1,5)
      PRINT 62,(Q2(J,I),I=1,5)
      PRINT 63,(Q3(J,I),I=1,5)
      PRINT 86,(NQ1(J,I),I=1,5)
      PRINT 87,(NQ2(J,I),I=1,5)
      PRINT 88,(NQ3(J,I),I=1,5)
      PRINT 35
   35 FORMAT(/)
   64 CONTINUE
   61 FORMAT(10H         K ,5E10.4)
   62 FORMAT(10H        KH ,5E10.4)
   63 FORMAT(10H        KL ,5E10.4)
   86 FORMAT(10H   NUMBER ,5I10)
   87 FORMAT(10H  NUMBERH ,5I10)
   88 FORMAT(10H  NUMBERL ,5I10)
      PRINT 35
      PRINT 35
C     THE 3 AVERAGE *K* FACTORS FOR EACH WORD GROUP FILE ARE PRINTED.
      PRINT 45
   45 FORMAT(*  WORD GROUP AVERAGES  *)
      PRINT 61,(Q1(MW,I),I=1,5)
      PRINT 62,(Q2(MW,I),I=1,5)
      PRINT 63,(Q3(MW,I),I=1,5)
      PRINT 86,(NQ1(MW,I),I=1,5)
      PRINT 87,(NQ2(MW,I),I=1,5)
      PRINT 88,(NQ3(MW,I),I=1,5)
C     PROGRAM TERMINATION SEQUENCE.
      PRINT 70
   70 FORMAT (1H1,*  NORMAL TERMINATION  *)
      END
```

```
      PROGRAM DISTSN(INPUT,OUTPUT,TAPE2=/135,TAPE3=/175)
C     PROGRAM DISTSN COMPUTES THE LINEAR DISTANCE MEASURES LM1H, LM1L,
C     LM2H, AND LM2L FOR EACH WORD AND THE WORD GROUP AVERAGE FOR EACH
C     PAIR OF NOISE AND SIGNAL FILES READ IN.
      DIMENSION E(2),NW(51),FNAME(5)
      DIMENSION FDAME(5),D(2),B(30,2)
      DIMENSION A(30,2),R(30),DM(4),NCT(2)
      DIMENSION Q1(51,5),Q2(51,5),Q3(51,5),Q4(51,5)
      DIMENSION NQ1(51,5),NQ2(51,5)
C     INITALIZE VARIABLES. *MWORD* IS A COUNTING VARIABLE.
      MWORD=0
C     THE LENGTH, IN ANALYSIS FRAMES, OF THE 50 WORDS IN THE WORD GROUP
C     BEING PROCESSED ARE READ INTO THE PROGRAM.
   50 READ 10,N256
   10 FORMAT(I5)
      IF(N256 .EQ. 0) GO TO 20
      MWORD=MWORD+1
      NMULT=N256
      NW(MWORD)=NMULT
      GO TO 50
   20 CONTINUE
      MW=MWORD+1
C     A TOTAL OF *IFILE* NOISE AND SIGNAL DATA FILES ARE READ INTO THE PROGRAM.
      IFILE=5
      READ 5,(FNAME(I),I=1,5)
      READ 5,(FDAME(I),I=1,5)
    5 FORMAT(5A7)
C     EACH OF THE PAIRS OF NOISE AND SIGNAL DATA FILES ARE PROCESSED.
      DO 25 K25=1,5
      LFN=LFORM(FNAME(K25))
      CALL PFMATCH(5LTAPE3,LFN,0,0,0,0,1)
      LFN=LFORM(FDAME(K25))
      CALL PFMATCH(5LTAPE2,LFN,0,0,0,0,1)
C     EACH OF THE WORDS IN THE WORD GROUP FILES ARE PROCESSED.
      DO 30 K30=1,MWORD
      NMULT=NW(K30)
      SUM=0.0
      DO 15 K15=1,NMULT
C     FRAME BY FRAME, THE NOISE AND SIGNAL LPC DATA IS READ INTO THE PROGRAM.
      READ(2,200) ALPHA,NFILE,NTAPE,NWORD,MULT,K60,Z1,Z2,D(1),D(2),Y1,Y2
     1,Y3,Y4,Y5,Y6
  200 FORMAT(A10,5I5,10E10.4)
      READ (3,300) ALPHA,NFILE,NTAPE,NWORD,MULT,K60,X1,X2,E(1),E(2),A2,D
     1A2,R0,DR0,ERROR,DERROR
  300 FORMAT(A10,5I5,10E14.8)
      A(K15,1)=ALOG(E(1))
      A(K15,2)=ALOG(E(2))
      B(K15,1)=ALOG(D(1))
      B(K15,2)=ALOG(D(2))
      R(K15)=R0
      SUM=SUM+R0
   15 CONTINUE
C     HALF THE AVERAGE ENERGY OF THE WORD BEING PROCESSED IS COMPUTED.
      RAVG2=(SUM/NMULT)/2.0
C     SUBROUTINE ENERGY IS CALLED TO COMPUTE THE DISTANCE MEASURES LM1H,
C     LM1L, LM2H, AND LM2L FOR THE WORD BEING PROCESSED.
      CALL ENERGY(NMULT,RAVG2,A,B,R,DM,NCT)
      Q1(K30,K25)=DM(1)
      Q2(K30,K25)=DM(2)
      Q3(K30,K25)=DM(3)
      Q4(K30,K25)=DM(4)
      NQ1(K30,K25)=NCT(1)
      NQ2(K30,K25)=NCT(2)
   30 CONTINUE
C     THE 4 AVERAGE DISTANCE MEASURES FOR THE WORD GROUP FILE ARE FOUND.
      SUM1=SUM2=SUM3=SUM4=0.0
      NC1=NC2=0
      DO 24 J=1,MWORD
      SUM1=SUM1+Q1(J,K25)
      SUM2=SUM2+Q2(J,K25)
      SUM3=SUM3+Q3(J,K25)
      SUM4=SUM4+Q4(J,K25)
      NC1=NC1+NQ1(J,K25)
      NC2=NC2+NQ2(J,K25)
```

```
   24 CONTINUE
      Q1(MW,K25)=SUM1/MWORD
      Q2(MW,K25)=SUM2/MWORD
      Q3(MW,K25)=SUM3/MWORD
      Q4(MW,K25)=SUM4/MWORD
      NQ1(MW,K25)=NC1
      NQ2(MW,K25)=NC2
   25 CONTINUE
C     THE AVERAGE DISTANCE MEASURES FOR EACH WORD OF EACH WORD GROUP FILE
C     ARE PRINTED.
      PRINT 40
   40 FORMAT(1H1)
      PRINT 80,NFILE
   80 FORMAT(*  WORD GROUP NO.  *,I5)
      PRINT 75
   75 FORMAT(60H  VARIABLE        38.4      19.2        9.6         FM
     1  AM)
      DO 64 J=1,MWORD
      PRINT 69,J
   69 FORMAT(*  WORD NUMBER  *,I5)
      PRINT 61,(Q1(J,I),I=1,5)
      PRINT 62,(Q2(J,I),I=1,5)
      PRINT 63,(Q3(J,I),I=1,5)
      PRINT 66,(Q4(J,I),I=1,5)
      PRINT 86,(NQ1(J,I),I=1,5)
      PRINT 87,(NQ2(J,I),I=1,5)
      PRINT 35
   35 FORMAT(/)
   64 CONTINUE
   61 FORMAT(10H      LM1H ,5E10.4)
   62 FORMAT(10H      LM2H ,5E10.4)
   63 FORMAT(10H      LM1L ,5E10.4)
   66 FORMAT(10H      LM2L ,5E10.4)
   86 FORMAT(10H  NUMBERH ,5I10)
   87 FORMAT(10H  NUMBERL ,5I10)
      PRINT 35
      PRINT 35
C     THE AVERAGE DISTANCE MEASURES FOR EACH WORD GROUP FILE ARE PRINTED.
      PRINT 45
   45 FORMAT(*  WORD GROUP AVERAGES  *)
      PRINT 61,(Q1(MW,I),I=1,5)
      PRINT 62,(Q2(MW,I),I=1,5)
      PRINT 63,(Q3(MW,I),I=1,5)
      PRINT 66,(Q4(MW,I),I=1,5)
      PRINT 86,(NQ1(MW,I),I=1,5)
      PRINT 87,(NQ2(MW,I),I=1,5)
C     PROGRAM TERMINATION SEQUENCE.
      PRINT 70
   70 FORMAT (1H1,*  NORMAL TERMINATION  *)
      END
```

```
      SUBROUTINE PRINT1 (N, NUMBER)
C     SUBROUTINE PRINT1 PRINTS ALL OF THE CROSS-CORRELATION TERMS
C     CALCULATED BY FFTCOR4 FOR A PARTICULAR SET OF MIDPOINTS.
      COMMON /FFT1 /A (4200), B (4200), DUM1 (4200), DUM2 (4200), C (500
     10)
      COMMON /FFT2 /ICOR (2100)
      NUMHALF = NUMBER / 2
      NUMQTR = NUMBER / 4
      KUPLIM = 1 + NUMQTR / 10
      LIMIT = NUMQTR - 1
      C (LIMIT + 1) = 0.
      DO 115 K25 = 1, LIMIT
      C (K25) = 0
      C (K25 + NUMHALF + NUMQTR) = 0
  115 CONTINUE
      DO 120 K30 = 1, KUPLIM
      LCNTR = - 10 * (KUPLIM - K30) - 1
      MCNTR = LCNTR - 9
      LL = NUMHALF + 1 + LCNTR
      PRINT 1500, LCNTR, C (LL), C (LL - 1), C (LL - 2), C (LL - 3), C (
     1LL - 4), C (LL - 5), C (LL - 6), C (LL - 7), C (LL - 8), C (LL - 9
     2), MCNTR
  120 CONTINUE
      PRINT 1502, C (NUMHALF + 1)
      DO 125 K40 = 1, KUPLIM
      LCNTR = 10 * (K40 - 1) + 1
      MCNTR = LCNTR + 9
      LL = NUMHALF + 1 + LCNTR
      LUP = LL + 9
      PRINT 1500, LCNTR, (C (I), I = LL, LUP), MCNTR
  125 CONTINUE
      RETURN
 1500 FORMAT  (I5, 2X, 10F12.3, 2X, I5)
 1502 FORMAT  (*    0    *, 12X, F12.3)
      END
```

```
      SUBROUTINE UNPACK (ID,IN,LL)
C     SUBROUTINE UNPACK TAKES A 60 BIT WORD AND SEPERATES IT INTO FIVE
C     12 BIT SAMPLES.  THIS IS DONE FOR THE EACH 80 WORD RECORD.
      DIMENSION ID(400),IN(80)
      LS=-5
      DO 60 I=1,LL
      LT=IN(I)
      LS=LS+10
      DO 50 L=1,5
      LTI=LT .AND. 3777B
      LTJ=LT .AND. 4000B
      IF(LTJ .NE. 0) LTI= .NOT. LTI
      ID(LS)=LTI
C     *SHIFT(LT,-12)* SHIFTS THE BITS OF VARIABLE *LT* TO THE RIGHT 12
C     PLACES.  THIS IS PERFORMED IN ORDER TO OBTAIN THE FIVE 12 BIT
C     SAMPLES FROM EACH 60 BIT WORD.
      LT=SHIFT(LT,-12)
      LS=LS-1
   50 CONTINUE
   60 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE REVBIN (A, B, MM)
C           CALL REVBIN(A,B,M)
C           REVERSIBLE PERMUTATION OF ARRAYS A AND B
C           FROM NORMAL SEQUENCE TO REVERSE BINARY SEQUENCE,
C           OR VICE VERSA.
C           SEQUENCE LENGTH IS N = 2**M
C           WRITTEN BY L. DAVID LEWIS AND MARIE JEST, ESSA.
C           MODIFIED FROM, OR INSPIRED BY THE ALGOL PROCEDURE
C           REVERSEBINARY,                   BY R. C. SINGLETON, SRI.
      DIMENSION A (16384), B (16384)
      COMMON /FFTCC /M, JC (15), ST (15)
      M = MM
      CALL FFTC
      IF (M .LE. 1) RETURN
      N = JC (M + 1)
      NP = N + 1
      K = 1
      I = 2
      J = N - 1
  100 LC = M
  105 K = K + JC (LC)
      JC (LC) = - JC (LC)
      IF (JC (LC) .LT. 0) GO TO 110
      IF (LC .EQ. 2) RETURN
      LC = LC - 1
      GO TO 105
  110 IF (K .LE. I .OR. J .LT. K) GO TO 115
      T = A (I)
      A (I) = A (K)
      A (K) = T
      T = B (I)
      B (I) = B (K)
      B (K) = T
      IF (J .EQ. K) GO TO 115
      KK = NP - K
      T = A (KK)
      A (KK) = A (J)
      A (J) = T
      T = B (KK)
      B (KK) = B (J)
      B (J) = T
  115 I = I + 1
      J = J - 1
      GO TO 100
      END
```

```
      SUBROUTINE CFFTRC (A, B, MM, SCALE, NEXP)
C              DISCRETE COMPLEX FAST FOURIER TRANSFORM.
C              CALL CFFTRC(A,B,M,SC,NX)
C              INPUT A(J) + I*B(J) IN REVERSE BINARY SEQUENCE.
C              OUTPUT A(K) + I*B(K) IN NORMAL SEQUENCE.
C              SEQUENCE LENGTH IS N = 2**M
C              SC IS REAL SCALING MULTIPLIER.
C              NX IS THE SIGN OF THE EXPONENT IN THE TRANSFORM DEFINITION.
C              INNER LOOP SINES AND COSINES COMPUTED
C              RECURSIVELY BY SINGLETONS 2ND DIFFERENCE ALGORITHM,
C              INITIALIZED FROM A DATA TABLE.
C              WRITTEN BY L. DAVID LEWIS AND MARIE WEST, ESSA.
C              MODIFIED FROM, OR INSPIRED BY THE ALGOL PROCEDURE
C              REVERSEFOURIERC.          BY R. C. SINGLETON, SRI.
      DIMENSION A (16384), B (16384)
      COMMON /FFTCC /M, JD (15), S (15)
      M = MM
      CALL FFTC
      N = JD (M + 1)
      K = N / 4
      NQ = K
      NN = N - 1
      JSPAN = 1
      SC = SCALE
      IF (ABS (SC - 1.) .LT. 1.E-10) GO TO 105
      DO 100 JC = 1, N
      A (JC) = SC * A (JC)
      B (JC) = SC * B (JC)
100   CONTINUE
105   IF (M .EQ. 0) RETURN
      DO 110 KK = 1, N, 2
      KS = KK + 1
      RE = A (KK) - A (KS)
      A (KK) = A (KK) + A (KS)
      A (KS) = RE
      FIM = B (KK) - B (KS)
      B (KK) = B (KK) + B (KS)
      B (KS) = FIM
110   CONTINUE
      IF (M .EQ. 1) RETURN
      EXPS = ISIGN (1, NEXP)
      NP = 1
      DO 125 JB = 2, M
      SD = - S (JB - 1)
      CD = 2. * S (JB) * S (JB)
      R = - 2. * CD
      CN = 1.
      CM = 0.
      SN = 0.
      JJ = 0
      KK = 1
      SM = + EXPS
      JSPANH = JSPAN
      JSPAN = JSPAN + JSPAN
115   KS = KK + JSPAN
      RE = CN * A (KS) - SN * B (KS)
      FIM = SN * A (KS) + CN * B (KS)
      A (KS) = A (KK) - RE
      A (KK) = A (KK) + RE
      B (KS) = B (KK) - FIM
      B (KK) = B (KK) + FIM
      KK = KK + JSPANH
      KS = KS + JSPANH
      FIM = SM * A (KS) + CM * B (KS)
      RE = CM * A (KS) - SM * B (KS)
      A (KS) = A (KK) - RE
      A (KK) = A (KK) + RE
      B (KS) = B (KK) - FIM
      B (KK) = B (KK) + FIM
      KK = KS + JSPANH
      IF (KK .LT. N) GO TO 115
      KK = KK - NN
      JJ = JJ + K
      IF (JJ .GE. NQ) GO TO 120
```

```
      CD = R * CN + CD
      CN = CD + CN
      SM = CN * EXPS
      SD = R * CM + SD
      CM = SD + CM
      SN = - CM * EXPS
      GO TO 115
  120 K = K / 2
  125 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE RTRAN2T (A, B, MM, INV)
C                CALL RTRAN2T(A,B,M,INV)
C                IF(INV.GT.0) UNSCRAMBLE THE TRANSFORMS OF TWO REAL SEQUENCES.
C                IF(INV.LT.0) SCRAMBLE THE TRANSFORMS OF TWO REAL SEQUENCES.
C                INPUT AND OUTPUT ARE IN NORMAL SEQUENCE,
C                SEE WRITEUP FOR DETAILS.
C                SEQUENCE LENGTH IS N = 2**M
C                WRITTEN BY L. DAVID LEWIS AND MARIE WEST, ESSA.
C                MODIFIED FROM, OR INSPIRED BY THE ALGOL PROCEDURE
C                REALTRAN,                         BY R. C. SINGLETON, SRI.
      DIMENSION A (16386), B (16386)
      COMMON /FFTCC /M, JC (15), ST (15)
      M = MM
      CALL FFTC
      N = JC (M + 1)
      NH = N / 2
      IF (INV .LT. 0) GO TO 120
      IF (M .LT. 2) GO TO 105
      K = N
      DO 100 J = 2, NH
      A (K + 1) = B (J) + B (K)
      B (K + 1) = A (K) - A (J)
      A (J) = A (J) + A (K)
      B (J) = B (J) - B (K)
  100 K = K - 1
  105 A (1) = 2. * A (1)
      A (NH + 2) = 2. * B (1)
      B (1) = B (NH + 2) = 0.
      IF (M .EQ. 0) RETURN
      A (NH + 1) = 2. * A (NH + 1)
      A (N + 2) = 2. * B (NH + 1)
      B (NH + 1) = B (N + 2) = 0.
      IF (M .EQ. 2) RETURN
      K = N + 1
      J = NH + 3
  110 NS = NH / 2 - 1
      DO 115 L = 1, NS
      T = A (J)
      A (J) = A (K)
      A (K) = T
      T = B (J)
      B (J) = B (K)
      B (K) = T
      J = J + 1
  115 K = K - 1
      RETURN
  120 B (1) = A (NH + 2)
      IF (M .EQ. 0) RETURN
      B (NH + 1) = A (N + 2)
      IF (M .EQ. 1) RETURN
      K = NH + 2
      DO 125 J = 2, NH
      A (K) = A (J) + B (K + 1)
```

BEST AVAILABLE COPY

```
      B (K) = A (K + 1) - B (J)
      A (J) = A (J) - B (K + 1)
      B (J) = B (J) + A (K + 1)
125   K = K + 1
      IF (M .EQ. 2) RETURN
      K = N
      J = NH + 2
      GO TO 110
      END
```

```
      SUBROUTINE FFTC
C               COMMON SUBROUTINE FOR FFT SUBROUTINES.
C               JC IS POWERS OF TWO ARRAY..  JC(M)=2**(M-1)
C               ST IS SINE ARRAY.. ST(M)=SIN(PI/(2**M))
C               M IS TESTED FOR PROPER INPUT RANGE, 0.LE.M.LE.14.
      COMMON /FFTCC /M, JC (15), ST (15)
      DATA (JC = 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096
     1, 8192, 16384)
      DATA (ST = 1.00000000000E+000, 7.07106781187E-001, 3.82683432365E-
     1001, 1.95090322016E-001, 9.80171403296E-002, 4.90676743274E-002, 2
     2.45412285229E-002, 1.22715382857E-002, 6.13588464915E-003, 3.06795
     3676297E-003, 1.53398018629E-003, 7.66990318743E-004, 3.83495187571
     4E-004, 1.91747597311E-004, 9.58737990960E-005)
      IF (M .LT. 0 .OR. M .GT. 14) PRINT 1500, 7
      RETURN
1500   FORMAT (*0ILLEGAL VALUE FOR M,  M =*, I4)
      END
```

```
      SUBROUTINE REALTRA (A, B, MM, NE, INV)
C              CALL REALTRAN(A,B,M,NE,INV)
C              IF(INV.GT.0) UNSCRAMBLE THE TRANSFORM OF A REAL SEQUENCE.
C              IF(INV.LT.0) SCRAMBLE THE TRANSFORM OF A REAL SEQUENCE.
C              INPUT AND OUTPUT ARE IN NORMAL SEQUENCE.
C              SEE WRITEUP FOR DETAILS.
C              SEQUENCE LENGTH IS N = 2**M
C              NE MUST AGREE WITH SIGN OF EXPONENT IN TRANSFORM DEFINITION.
C              INNER LOOP SINES AND COSINES COMPUTED
C              RECURSIVELY BY SINGLETONS 2ND DIFFERENCE ALGORITHM,
C              INITIALIZED FROM A DATA TABLE.
C              WRITTEN BY L. DAVID LEWIS AND MARIE WEST, ESSA.
C              MODIFIED FROM, OR INSPIRED BY THE ALGOL PROCEDURE
C              REALTRAN,                        BY R. C. SINGLETON, SRI.
      DIMENSION A (16385), B (16385)
      COMMON /FFTCC /M, JC (15), ST (15)
      M = MM
      CALL FFTC
      N = K = JC (M + 1)
      NH = N / 2
      NK = NH + 1
      CN = ISIGN (1, INV)
      SN = ISIGN (1, NE)
      A (NK) = 2. * A (NK)
      B (NK) = 2. * CN * SN * B (NK)
      IF (CN .GE. 0.) GO TO 100
      FIM = A (1) - A (N + 1)
      A (1) = A (1) + A (N + 1)
      B (1) = FIM
      GO TO 105
  100 A (N + 1) = 2. * (A (1) - B (1))
      A (1) = 2. * (A (1) + B (1))
      B (1) = B (N + 1) = 0.
  105 IF (M .EQ. 0) RETURN
      SD = CN * SN * ST (M)
      R = 2. * ST (M + 1)
      R = - R * R
      CD = - .5 * CN * R
      SN = 0.
      DO 110 J = 2, NH
      CD = R * CN + CD
      CN = CD + CN
      SD = R * SN + SD
      SN = SD + SN
      AA = A (J) + A (K)
      AB = A (J) - A (K)
      BA = B (J) + B (K)
      BB = B (J) - B (K)
      RE = CN * BA + SN * AB
      FIM = SN * BA - CN * AB
      B (K) = FIM - BB
      B (J) = FIM + BB
      A (K) = AA - RE
      A (J) = AA + RE
  110 K = K - 1
      RETURN
      END
```

```
      SUBROUTINE CFFTS (A, B, MM, SCALE, NEXP)
C           DISCRETE COMPLEX FAST FOURIER TRANSFORM.
C           CALL CFFTS(A,B,M,SC,NX)
C           INPUT A(J) + I*B(J) IN NORMAL SEQUENCE.
C           OUTPUT A(K) + I*B(K) IN REVERSE BINARY SEQUENCE.
C           SEQUENCE LENGTH IS N = 2**M
C           SC IS REAL SCALING MULTIPLIER.
C           NX IS THE SIGN OF THE EXPONENT IN THE TRANSFORM DEFINITION.
C           INNER LOOP SINES AND COSINES COMPUTED
C           RECURSIVELY BY SINGLETONS 2ND DIFFERENCE ALGORITHM,
C           INITIALIZED FROM A DATA TABLE.
C           WRITTEN BY L. DAVID LEWIS AND MARIE WEST, ESSA.
C           MODIFIED FROM, OR INSPIRED BY THE ALGOL PROCEDURE
C           FASTFOURIERS,                   BY R. C. SINGLETON, SRI.
      DIMENSION A (16384), B (16384)
      COMMON /FFTCC /M, JC (15), SNT (15)
      MA = M = MM
      CALL FFTC
      N = JC (M + 1)
      NH = N / 2
      NQ = NH / 2
      SC = SCALE
       IF (M .EQ. 0) GO TO 125
       IF (M .EQ. 1) GO TO 115
      EXPS = ISIGN (1, NEXP)
      NN = N - 1
      K = 1
      DO 110 JA = 2, M
      CE = SNT (MA)
      MA = MA - 1
      CD = 2. * CE * CE
      SD = - SNT (MA)
      R = - 2. * CD
      CN = 1.
      CM = 0.
      SN = 0.
      JJ = 0
      KK = 1
      SM = + EXPS
      JSPAN = NH
      NH = JSPAN / 2
  100 KS = KK + JSPAN
      RE = A (KK) - A (KS)
      A (KK) = A (KK) + A (KS)
      FIM = B (KK) - B (KS)
      B (KK) = B (KK) + B (KS)
      A (KS) = CN * RE - SN * FIM
      B (KS) = SN * RE + CN * FIM
      KK = KK + NH
      KS = KS + NH
      RE = A (KK) - A (KS)
      A (KK) = A (KK) + A (KS)
      FIM = B (KK) - B (KS)
      B (KK) = B (KK) + B (KS)
      A (KS) = CM * RE - SM * FIM
      B (KS) = SM * RE + CM * FIM
      KK = KS + NH
      IF (KK .LT. N) GO TO 100
      KK = KK - NN
      JJ = JJ + K
      IF (JJ .GE. NQ) GO TO 105
      CD = R * CN + CD
      CN = CD + CN
      SD = R * CM + SD
      CM = CM + SD
      SN = - CM * EXPS
      SM = CN * EXPS
      GO TO 100
  105 K = K + K
  110 CONTINUE
  115 DO 120 KK = 1, N, 2
      KS = KK + 1
      RE = A (KK) - A (KS)
      A (KK) = (A (KK) + A (KS))
```

```
      A (KS) = RE
      FIM = B (KK) - B (KS)
      B (KK) = (B (KK) + B (KS))
      B (KS) = FIM
  120 CONTINUE
  125 IF (ABS (SC - 1.) .LT. 1.E-10) GO TO 135
      DO 130 JB = 1, N
      A (JB) = SC * A (JB)
      B (JB) = SC * B (JB)
  130 CONTINUE
  135 RETURN
      END
```

```
      SUBROUTINE PRECOEF (NPC)
C     SUBROUTINE PRECOEF COMPUTES THE 12 PREDICTOR COEFFICIENTS, THE 12
C     REFLECTION COEFFICIENTS, AND THE MINIMUM SQUARED ERROR USING THE
C     LEVINSON ALGORITHM AND THE AUTOCORRELATION TERMS FROM SUBROUTINE
C     AUTOCOR.  SUBROUTINE STABLE IS USED TO INSURE A STABLE FILTER IS
C     GENERATED.  THE PREDICTOR COEFFICIENTS ARE STORED IN PCOEF(J,12),
C     J=1,12. THE REFLECTION COEFFICIENTS ARE STORED IN PCOEFF(J,J),
C     J=1,12. AND THE MINIMUM SQUARED ERROR IS THE VARIABLE D.
      COMMON A(256),ACOR(13),PCOEF(12,12),GAM(13),D
      C=-ACOR(2)
      D=1.0-C*C
      PCOEF(1,1)=-C
      DO 10 J=2,NPC
      J1=J-1
      SUM=0.0
      DO 15 I=1,J1
   15 SUM=SUM+ACOR(J+1-I)*PCOEF(I,J1)
      G=ACOR(J+1)-SUM
      C=-G/D
      IF (ABS(C) .GT. 0.99) CALL STABLE (J,C)
      D=D*(1.0-C*C)
      DO 20 I=1,J1
   20 PCOEF(I,J)=PCOEF(I,J1)+C*PCOEF(J-I,J1)
   10 PCOEF(J,J)=-C
      RETURN
      END
```

```
      SUBROUTINE REORDER (A, B, MM)
C             CALL REORDER(A,B,M)
C             REVERSIBLE PERMUTATION OF REAL SEQUENCE
C             FROM FIRST-LAST-NORMAL SEQUENCE
C             TO ODD-EVEN-REVERSE BINARY SEQUENCE.
C             SEQUENCE LENGTH IS N = 2**M.
C             WRITTEN BY L. DAVID LEWIS AND MARIE WEST, ESSA.
C             REORDER,                     BY R. C. SINGLETON, SRI.
      DIMENSION A (16384), B (16384), LST (15)
      COMMON /FFTCC /M, LC (15), ST (15)
      M = MM
      CALL FFTC
      IF (M .EQ. 0) RETURN
      JA = M + 1
      JB = M - 1
      I = KB = 0
      KU = LC (JA) - 1
      DO 100 KA = 1, KU, 2
      T = A (KA + 1)
      A (KA + 1) = B (KA)
      B (KA) = T
  100 CONTINUE
      IF (M .EQ. 1) RETURN
      LIM = M / 2 + 1
  105 KS = LC (JB + 1) + KB
      KU = KS
      JJ = LC (JA - JB)
      KK = KB + JJ
  110 K = KK + JJ
  115 T = A (KK + 1)
      A (KK + 1) = A (KS + 1)
      A (KS + 1) = T
      T = B (KK + 1)
      B (KK + 1) = B (KS + 1)
      B (KS + 1) = T
      KK = KK + 1
      KS = KS + 1
      IF (KK .LT. K) GO TO 115
      KK = KK + JJ
      KS = KS + JJ
      IF (KK .LT. KU) GO TO 110
      IF (JB .LE. LIM) GO TO 120
      JB = JB - 1
      I = I + 1
      LST (I) = JB
      GO TO 105
  120 IF (I .LE. 0) RETURN
      JB = LST (I)
      I = I - 1
      KB = KS
      GO TO 105
      END
```

```
      SUBROUTINE AUTOCOR (NUMBER,NPC)
C     SUBROUTINE AUTOCOR COMPUTES THE ENERGY TERM AND THE 12 NORMALIZED
C     AUTOCORRELATION TERMS FOR EACH 256 SAMPLE WINDOW.  THESE VALUES
C     ARE STORED IN ACOR(J), J=1,13 RESPECTFULLY.
      COMMON A(256),ACOR(13),PCOEF(12,12),GAM(13),ERROR
      N=NPC+1
      SUM=0.0
      DO 10 I=1,NUMBER
   10 SUM=SUM+A(I)
      SMEAN=SUM/NUMBER
      DO 15 I=1,NUMBER
   15 A(I)=A(I)-SMEAN
      SUM=0.0
      DO 20 I=1,NUMBER
   20 SUM=SUM+A(I)*A(I)
      ACOR(1)=SUM
      DO 25 J=2,N
      SUM=0.0
      NUM=NUMBER-J+1
      DO 30 I=1,NUM
   30 SUM=SUM+A(I)*A(I+J-1)
   25 ACOR(J)=SUM/ACOR(1)
      RETURN
      END
```

```
      SUBROUTINE STABLE (J,C)
C     SUBROUTINE STABLE FORCES THE ABSOLUTE VALUE OF EACH REFLECTION
C     COEFFICIENT TO BE LESS THAN OR EQUAL TO 0.99, THEREBY INSURING A
C     STABLE FILTER.
      CC=-C
      PRINT 10,J,CC
   10 FORMAT (* REFLECTION COEFFICIENT *,I2,* IS UNSTABLE, = *,F10.5)
      IF (C .GT. 0.99) C=0.99
      IF (C .LT. -0.99) C=-0.99
      RETURN
      END
```

```
      SUBROUTINE GAMMA (NPC)
C     SUBROUTINE GAMMA COMPUTES THE PREDICTOR COEFFICIENT AUTOCORRELATION
C     TERMS FROM THE PREDICTOR COEFFICIENTS.
      COMMON A(256),ACOR(13),PCOEF(12,12),GAM(13),ERROR
      DIMENSION PC(13)
      N=NPC+1
      PC(1)=1.0
      DO 10 I=1,NPC
   10 PC(I+1)=-PCOEF(I,12)
      SUM=0.0
      DO 15 I=1,N
   15 SUM=SUM+PC(I)*PC(I)
      GAM(1)=SUM
      DO 20 I=1,NPC
      SUM=0.0
      NN=N-I
      DO 25 J=1,NN
   25 SUM=SUM+PC(J)*PC(J+I)
   20 GAM(I+1)=2.0*SUM
      RETURN
      END
```

```
      SUBROUTINE HILOW(NMULT,RAVG2,A,R,S,DM,NCT)
C     SUBROUTINE HILOW COMPUTES THE AVERAGE *K* FACTOR FOR THE HIGH AND
C     LOW ENERGY FRAMES OF THE WORD BEING PROCESSED.
      DIMENSION A(30),R(30),S(30),DM(2),NCT(2)
      SUM1=SUM2=0.0
      NC1=NC2=0
      DO 10 K10=1,NMULT
      IF(R(K10) .LT. RAVG2) GO TO 15
      SUM1=SUM1+A(K10)/S(K10)
      NC1=NC1+1
      GO TO 20
   15 CONTINUE
      SUM2=SUM2+A(K10)/S(K10)
      NC2=NC2+1
   20 CONTINUE
   10 CONTINUE
      NCT(1)=NC1
      NCT(2)=NC2
      DM(1)=SUM1/NC1
      DM(2)=SUM2/NC2
      RETURN
      END
```

```
      SUBROUTINE ENERGY(NMULT,RAVG2,A,B,R,DM,NCT)
C     SUBROUTINE ENERGY COMPUTES THE AVERAGE OF THE DISTANCE MEASURES
C     LM1H, LM1L, LM2H, AND LM2L FOR THE WORD BEING PROCESSED.
      DIMENSION A(30,2),B(30,2),R(30),DM(4),NCT(2)
      RLIM1=0.82
      RLIM2=2.46
      SUM1=SUM2=SUM3=SUM4=0.0
      NC1=NC2=0
      DO 10 K10=1,NMULT
C     THE RESIDUAL DISTANCE MEASURES ARE LINEARIZED FOR EACH FRAME.
      DO 5 K5=1,2
      IF(B(K10,K5) .LE. RLIM2) GO TO 25
      RM2=B(K10,K5)
      RM1=(RM2-RLIM2+1.0)*RLIM1
      GO TO 30
   25 RM1=RLIM1
      RM2=RLIM2
   30 AA=1.0/(RM1-RM2)
      BB=-AA*RM2
      IF(A(K10,K5) .LT. RM1) GO TO 35
      IF(A(K10,K5) .GT. RM2) GO TO 40
      A(K10,K5)=AA*A(K10,K5)+BB
      GO TO 45
   35 A(K10,K5)=1.0
      GO TO 45
   40 A(K10,K5)=0.0
   45 CONTINUE
    5 CONTINUE
C     THE HIGH AND LOW ENERGY FRAMES ARE FOUND AND THE LINEAR DISTANCE
C     MEASURES LM1H, LM1L, LM2H, AND LM2L ARE COMPUTED.
      IF(R(K10) .LT. RAVG2) GO TO 15
      SUM1=SUM1+A(K10,1)
      SUM2=SUM2+A(K10,2)
      NC1=NC1+1
      GO TO 20
   15 CONTINUE
      SUM3=SUM3+A(K10,1)
      SUM4=SUM4+A(K10,2)
      NC2=NC2+1
   20 CONTINUE
   10 CONTINUE
C     THE 4 AVERAGE DISTANCE MEASURES FOR THE WORD BEING PROCESSED ARE COMPUTED.
      NCT(1)=NC1
      NCT(2)=NC2
      IF(NC1 .EQ. 0) NC1=1
      IF(NC2 .EQ. 0) NC2=1
      DM(1)=SUM1/NC1
      DM(2)=SUM2/NC1
      DM(3)=SUM3/NC2
      DM(4)=SUM4/NC2
      RETURN
      END
```


*U.S. GOVERNMENT PRINTING OFFICE:1977 732-556/269 1-3