ILLEGIBLE DOCUMENT

THE FOLLOWING DOCUMENT(S) IS OF POOR LEGIBILITY IN THE ORIGINAL

THIS IS THE BEST COPY AVAILABLE

DIGITAL SPEECH PROCESSING TECHNIQUES

by

NIRANJAN U. SOLANKI
B. E. , Osmania University, Hyderabad, India, 1972

A MASTER'S REPORT

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas
1974

Approved by:

Nasir Ahmed
____
Major Professor

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

In recent years, digital signal processing techniques have become increasingly important in the analysis and synthesis of speech. The Computer has become the principal laboratory facility in speech research, as a consequence of the availibility of small, high speed, general purpose digital computers.

There are three main advantages with respect to using digital signal processing techniques. First, analog techniques offer little flexibility for changing filter frequencies or the shape and width of the spectral windows used. Second, digital processing methods provide an investigator the freedom in choosing the most appropriate method of analysis for each problem. Third, computerized techniques possess the capacity of handling large quantities of data.

The main objective of this report is to provide a documentation for a set of digital speech processing programs. These are FORTRAN programs, written for the data processing on the IBM 370/158 computer located in the Kansas State University Computing Centre. They were developed by Mr. T. Natarajan under the direction of Dr. N. Ahmed.

The data used to test the above programs was acquired by Dr. Rainbolt of the department of speech, in the form of a tape recording of the three male speakers. Speaker 1 was moderately to severly hard of hearing and speaker 2 and 3 were profoundly deaf. However, speaker 3 had a slightly greater mean hearing loss than speaker 2. This data was subsequently digitized by Engineering Experiment Station, University of Wisconsin. The resulting data was stored on a tape in form of 8-bit words. Each word represents a sampled value of the analog speech.

Chapter II presents a data conversion program which converts the 8-bit/word data into an equivalent 16-bit/word data. This conversion is necessary since arithmetic can be performed only on integer or real numbers. An integer representation requires at least 16 bits or half-word (i.e., two bytes) for processing on the IBM 370/158 computer. Details pertaining to the various steps involved in the conversion process along with information related to the corresponding job Control are also included. Next, a program which enables to generate digital spectrograms is presented. This program implements an extremely efficient, Fast Fourier Transform (FFT) subroutine along with a multi-printing technique.

The Chirp-Z-Transform (CZT) is implemented using an FFT routine in Chapter III. This version of the CZT is sometimes referred to as the modified CZT since its analysis is restricted to the $j\omega$-axis. This program enables one to obtain an accurate estimate of the fundamental frequency associated with a given segment of voiced speech.

Chapter IV presents a program which implements the Cepstrum technique of determining whether a given segment of speech consists of voiced or unvoiced speech. Finally, some conclusions are presented in Chapter V.

CHAPTER II
DATA ACQUISITION AND CONVERSION

2.1 Data Acquisition

The data for this study was acquired by Dr. Rainbolt, Dept. of Speech, Kansas State University, in the form of a tape recording of three male speakers. Speaker 1 was moderately to severly hard of hearing and speakers 2 and 3 were profoundly deaf. However, speaker 3 had a slightly greater mean hearing loss than speaker 2. These speakers repeated two sentences at the interval of 24 hours. The two sentences were as follows:

(i) Noon is the speepy time of day.
(ii) I like happy movies better.

The above recordings were made on an AMPEX AG 500 tape recording in a quet room at tape speed of 15 inches per second, and then transferred to another, at a tape of speed of 7.5 inches per second. The tape recorder used was a Wollensak T-1500. Table 2.1.1 lists the order in which the sentences appear on the tape. In what follows, this tape will be referred to as the 'analog tape'.

Quantization

The analog data tape was sent to the Engineering Experiment Station, University of Wisconsin, Madison, Wisconsin, for digitization. The analog data was sampled at a rate of one sample every 120 microseconds or 8333.3 samples per second. Each sample was then recorded on a 9-track tape as an 8-bit word and the identification number of this tape is 9638SP. The data on the digital tape consists of 42 records. Each record consists of 36,000 8-bit samples of one sentence - corresponding to 4.32 seconds of real time. Thus, each record would correspond to 9,000 32-bit words, with each 32-bit consisting of four 8-bit samples. The density of the tape is 800 bits per inch.

THIS BOOK CONTAINS NUMEROUS PAGES WITH DIAGRAMS THAT ARE CROOKED COMPARED TO THE REST OF THE INFORMATION ON THE PAGE.

THIS IS AS RECEIVED FROM CUSTOMER.

Table 2.1.1

The order in which the sentences appear on the tapes.

RECORD # (9638SP)	SPEAKER #	SENTENCE #	DAY #	LABEL # (9365RA)
SENT1	1	1	1	1
SENT2	1	2	1	-
SENT03	1	1	2	4
SENT04	1	2	2	-
SENT05	1	1	3	5
SENT06	1	2	3	-
SENT07	1	1	4	6
SENT08	1	2	4	-
SENT09	1	1	5	7
SENT10	1	2	5	-
SENT11	1	1	6	8
SENT12	1	2	6	-
SENT13	1	1	7	9
SENT14	1	2	7	-
SENT15	2	1	1	2
SENT16	2	2	1	-
SENT17	2	1	2	16
SENT18	2	2	2	-
SENT19	2	1	3	17
SENT20	2	2	3	-
SENT21	2	1	4	18
SENT22	2	2	4	-
SENT23	2	1	5	19
SENT24	2	2	5	-
SENT25	2	1	6	20
SENT26	2	2	6	=
SENT27	2	1	7	21
SENT28	2	2	7	-
SENT29	3	1	1	3

Table 2.1.1 (Continued)

RECORD # (9638SP)	SPEAKER #	SENTENCE #	DAY #	LABEL # (9365RA)
SENT30	3	2	1	-
SENT31	3	1	2	10
SENT32	3	2	2	-
SENT33	3	1	3	11
SENT34	3	2	3	=
SENT35	3	1	4	12
SENT36	3	2	4	-
SENT37	3	1	5	13
SENT38	3	2	5	-
SENT39	3	1	6	14
SENT40	3	2	6	-
SENT41	3	1	7	15
SENT42	3	2	7	-

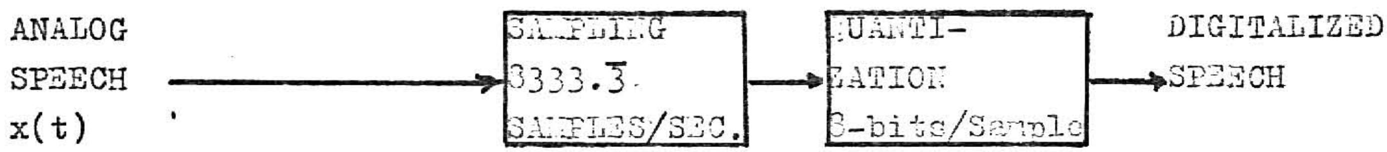


Fig. 2.2.1

ANALOG TO DIGITAL CONVERSION

An analog gain of 30 was used and for the amplified signal a data word of 01111111 corresponds to 100 volts. The first "0" in 01111111 is the sign bit, which makes the equivalent decimal number to be +127. A data word of 10000000 would correspond to -128.

2.2 Conversion Program

The data on the digital tape which is a 9-track tape, bearing a serial number 9638SP, is in form of samples, each sample being an 8-bit PCM (Pulse Code Modulation) word, in 2's complement form. This has to be converted into an equivalent 16-bit binary number and written on a data tape bearing a serial number 9365RA. This conversion is necessary, since the arithmetic operation can be performed only on integer or real numbers. An integer representation requires at least 16 bits or half-word (i.e. , two bytes) for processing on the IBM 370/158 computer in the Kansas State University Computing Center.

Statement #1 of the Fortran Conversion Program declares that the input data PCM(32760) from the digital tape has the format INTEGER*2. This has the effect of specifying that the variable PCM(32760) is of integer type and uses only 16 bits per word. The use of half-word integer variable can produce saving of space, if this is critical, at the expense of a decrease in the range of values that can be stored. The statements 4 and 5 cause the data to be read from the digital tape, assigning values to the variable PCM(I). The format used is (128(255A1),120A1). The number 255 is the maximum repetetion factor for formatting purposes in FORTRAN IV, the total being 32760 bytes. It is important to note that the A1 format causes the data to be read in 1 byte (i.e. , 8 bits) at a time . This byte is read in the first half of the word assigned to PCM(I), and blanks are inserted in the next 8-bits.

Statements 6 and 7 cause the data to be divided by 256. To divide a binary number by 256 or 2^8, is equivalent to an arithmetic shift of the binary number 8 bits to the right. The above ideas are best explained by illustrative examples which are presented next.

Consider the number +43. Its binary equivalent is 00101011. The first bit of the number is the sign bit (a '0' means it is a positive number and a '1' would mean a negative number) and the rest of the number is the magnitude. When this number is read into a 16-bit word using the A1 format, it appears as

0 0 1 0 1 0 1 1 b̸ b̸ b̸ b̸ b̸ b̸ b̸ b̸

where denotes a blank. When this number is divided by 256, it causes a 'shift right arithmetic operations'; i. e.: the number shifts to the right by 8 bits and at the same time the sign-bit, '0' in this case, is propagated throughout the blanks caused by the shift. The shifting operation and the final number obtained appear as follows:

0 0 1 0 1 0 1 1 b̸ b̸ b̸ b̸ b̸ b̸ b̸ b̸

0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 1

Thus the final result is a 16-bit positive number, having the first bit as a sign-bit and the rest of the bits denote the magnitude. In case of a negative number, the above operations are given as follows:

-43 decimal = 10101011 in binary (sign + magnitude)
= 11010101 in binary 2's complement with the sign bit.

Reading the 2's complement into a 16-bit word using A1 format yields the word

1 1 0 1 0 1 0 1 b̸ b̸ b̸ b̸ b̸ b̸ b̸ b̸

CONVERSION PROGRAM

```
//XPRS8386 JOB (XXXXXXXXX),RAINBOLT,TIME=(0,45),CLASS=B,MSGLEVEL=(1,1)     JOB 388
***TAPE9        2
//  EXEC  RINGWTR,PARM=9365RA
XXGO  EXEC PGM=RINGWTR                                                    00000010
XXSTEPLIB DD  DSN=SYS1.USERLIB,DISP=SHR                                   00000020
IEF236I ALLOC. FOR XPRS8386 GO
IEF237I 254    ALLOCATED TO STEPLIB
IEF285I    SYS1.USERLIB                              KEPT
IEF285I    VOL SER NOS= LIBPAK.

KSU003I PARTITION SIZE - 512K, MAXIMUM CORE USED -  10K,  TIME =  15.23.06
KSU004I I/O COUNT -- DISK =           0,  TAPE =        0
KSU001I STEP   1 GO.       EXECUTION TIME =          0.07 SEC ;  RETURN CODE =     0

//  EXEC    FORTGCLG
XXFORTGCLG      PROC M=MAIN                                               00000010
XXFORT    EXEC  PGM=IEYFORT                                               00000020
XXSYSPRINT  DD   SYSOUT=A                                                 00000030
XXSYSPUNCH DD   SYSOUT=B,DCB=BUFNO=1                                      00000040
XXSYSLIN DD      DSNAME=&LOADSET,DISP=(MOD,PASS),UNIT=SYSDA,             X00000050
XX               SPACE=(3200,(20,10)),                                    00000060
XX               DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)                     00000070
//FORT.SYSIN  DD  *
IEF236I ALLOC. FOR XPRS8386 FORT
IEF237I 341    ALLOCATED TO SYSPRINT
IEF237I 340    ALLOCATED TO SYSPUNCH
IEF237I 251    ALLOCATED TO SYSLIN
IEF237I 302    ALLOCATED TO SYSIN
IEF285I    SYS74150.T152034.RF000.XPRS8386.LOADSET       PASSED
IEF285I    VOL SER NOS= KSCC01.

KSU003I PARTITION SIZE - 512K, MAXIMUM CORE USED -  84K,  TIME =  15.23.22
KSU004I I/O COUNT -- DISK =           1,  TAPE =        0
KSU001I STEP   2 FORT      EXECUTION TIME =          0.92 SEC    RETURN CODE =     0

XXLKED   EXEC  PGM=IEWL,PARM='LET,LIST,MAP',COND=(5,LT,FORT)              00000030
XXSYSLIB DD DSNAME=SYS1.FORTLIB,DISP=SHR                                  00000100
XX             DD  DSNAME=SYS1.FORTRAN.SSP,DISP=SHR                       00000110
XX             DD  DSNAME=SYS1.SUBR,DISP=SHR                              00000120
XXSYSLMOD DD   DSNAME=&GOSET(GO),UNIT=(SYSDA,SEP=SYSLIB),                X00000130
IEF653I SUBSTITUTION JCL - DSNAME=&GOSET(MAIN),UNIT=(SYSDA,SEP=SYSLIB),
XX             DISP=(MOD,PASS),SPACE=(TRK,(8,5,1))                        00000160
XXSYSUT1 DD UNIT=(SYSDA,SEP=(SYSLMOD,SYSLIB)),SPACE=(1024,(200,20)),     X00000150
XX             DCB=BLKSIZE=1024                                           00000160
XXSYSPRINT  DD   SYSOUT=A                                                 00000170
XXSYSLIN     DD  DSNAME=*.FORT.SYSLIN,DISP=(OLD,DELETE)                   00000180
XX           DD  DDNAME=SYSIN                                             00000190
IEF236I ALLOC. FOR XPRS8386 LKED
IEF237I 250    ALLOCATED TO SYSLIB
IEF237I 254    ALLOCATED TO
IEF237I 254    ALLOCATED TO
IEF237I 251    ALLOCATED TO SYSLMOD
IEF237I 252    ALLOCATED TO SYSUT1
IEF237I 341    ALLOCATED TO SYSPRINT
IEF237I 251    ALLOCATED TO SYSLIN
IEF285I    SYS1.FORTLIB                              KEPT
IEF285I    VOL SER NOS= IPLVOL.
IEF285I    SYS1.FORTRAN.SSP                          KEPT
IEF285I    VOL SER NOS= LIBPAK.
IEF285I    SYS1.SUBR                                 KEPT
```

```
IEF285I   VOL SER NOS= LIBPAK.
IEF285I   SYS74150.T152034.RF000.XPRS8386.GOSET          PASSED
IEF285I   VOL SER NOS= KSCC01.
IEF285I   SYS74150.T152034.RF000.XPRS8386.R0000160       DELETED
IEF285I   VOL SER NOS= KSCC02.
IEF285I   SYS74150.T152034.RF000.XPRS8386.LOADSET        DELETED
IEF285I   VOL SER NOS= KSCC01.

KSU003I PARTITION SIZE - 512K, MAXIMUM CORE USED - 156K,  TIME =  15.24.05
KSU004I I/O COUNT -- DISK =          85,  TAPE =          0
KSU001I STEP   3 LKED      EXECUTION TIME =            0.90 SEC    RETURN CODE =     0

XXGO      EXEC   PGM=*.LKED.SYSLMOD,COND=((5,LT,FORT),(5,LT,LKED))          00000200
XXFT05F001  DD  DDNAME=SYSIN                                                 00000210
XXFT06F001 DD  SYSOUT=A                                                      00000220
XXFT07F001 DD  SYSOUT=B,DCB=BUFNO=1                                          00000230
XXFT10F001 DD  DSNAME=&&DSKSETA,UNIT=SYSDA,SPACE=(TRK,(1,19)),              X00000240
XX             DCB=(RECFM=VS,LRECL=293,BLKSIZE=297)                          00000250
XXFT11F001 DD  DSNAME=&&DSKSETB,UNIT=(SYSDA,SEP=FT10F001),                  X00000260
XX             DCB=(RECFM=VS,LRECL=293,BLKSIZE=297),SPACE=(TRK,(1,19))       00000270
XXSYSIDUMP DD SYSOUT=A                                                       00000280
//GO.FT08F001  DD  DSN=SPEECH,UNIT=TAPE9,VOL=SER=9638SP,DISP=OLD,
//  LABEL=(1,NL),DCB=(RECFM=U,BLKSIZE=32760,DEN=2)
//GO.FT09F001 DD  DSN=SENT39,UNIT=TAPE9,VOL=SER=9365RA,LABEL=(14,NL),
//  DISP=(,CATLG),DCB=(RECFM=FB,LRECL=10,BLKSIZE=10000)
//
IEF236I ALLOC. FOR XPRS8386 GO
IEF237I 251    ALLOCATED TO PGM=*.DD
IEF237I 340    ALLOCATED TO FT06F001
IEF237I 380    ALLOCATED TO FT07F001
IEF237I 251    ALLOCATED TO FT10F001
IEF237I 252    ALLOCATED TO FT11F001
IEF237I 341    ALLOCATED TO SYSIDUMP
IEF237I 171    ALLOCATED TO FT08F001
IEF237I 172    ALLOCATED TO FT09F001
IEF285I   SYS74150.T152034.RF000.XPRS8386.GOSET          PASSED
IEF285I   VOL SER NOS= KSCC01.
IEF285I   SYS74150.T152034.RF000.XPRS8386.DSKSETA        DELETED
IEF285I   VOL SER NOS= KSCC01.
IEF285I   SYS74150.T152034.RF000.XPRS8386.DSKSETB        DELETED
IEF285I   VOL SER NOS= KSCC02.
IEF285I   SPEECH                                         KEPT
IEF285I   VOL SER NOS= 9638SP.
IEF285I   SENT39                                         CATALOGED
IEF285I   VOL SER NOS= 9365RA.

KSU003I PARTITION SIZE - 512K, MAXIMUM CORE USED - 302K,  TIME =  15.26.10
KSU004I I/O COUNT -- DISK =           0,  TAPE =         73
KSU001I STEP   4 GO        EXECUTION TIME =           15.29 SEC    RETURN CODE =     0

IEF285I   SYS74150.T152034.RF000.XPRS8386.GOSET          DELETED
IEF285I   VOL SER NOS= KSCC01.

KSU002I JOB XPRS8386 EXECUTION TIME =          17.18 SEC
```

```
      INTEGER*2 PCM(32760)
      DIMENSICN X(32760)
      N=32760
      READ (8,15) (PCM(I),I=1,N)
   15 FORMAT(38(/),128(255A1),120A1)
      DO 20 I=1,N
      T=PCM(I)/256
   20 X(I)=T/127.
      WRITE(9,25) (X(I),I=1,N)
   25 FORMAT(F10.7)
      STOP
      END
```

Again division by 256 yields

1 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1

which is the 16 bit representation of -43 in 2's complement form.

The decimal equivalents of the resulting 16-bit word range from +127.0 to -127.0. Next, statement 8 causes the resulting data to be divided by 127.0, thus normalising these numbers between +1.0 and -1.0.

Statements 9 and 10 cause the data to be written on unit 9, which is a tape bearing serial number 9365RA as per the Job Control card. The corresponding format used is F10.7.

2.3 Details of the JCL (Job Control Language) Cards

The importance of the JCL cards used in the programs used for this project can hardly be overemphasized. Any error in the order or punching of the cards will result in a job termination. Also, there is a possibility of tapes being mixed up or something being written on them inadvertantly.

The following cards, the details of which are also given simutaneously, are used in all the programs which use the tape/tapes.

/*TAPE9 2

This card is known as a 'job set up' card. At the present time, when two concurrently executing jobs in different partitions both request that a 9-track tape be mounted, then one job may remain idle while it is waiting for the drive. Further, a job in the system waiting for a particular device may delay the scheduling of other steps until the existing request has been satisfied. This operation problem can result in an inefficient

CPU (Central Processing Unit). A similar situation exists when two multi-step jobs request the same tape drive. The drive is alternately assigned to the step of the two jobs resulting in wasted time spent in rewinding, dismounting, mounting, and re-positioning tapes. The same problems are encountered with 7-track tapes and removable disk packs.

The solution to the above problem is to prevent jobs requiring the same devices from executing concurrently. For this a 'set up' facility is available.

The set up card required for this has the following format:

```
/*DEVTYPE        n
```

where DEVTYPE is 2314, Tape7 or TAPE9 and n equals the number of devices of this type required by the job.

Thus for the conversion program, we need two 9-track tapes, which is given by the set up card given at the beginning. The number 2314 is a code number for the user owned disk pack.

```
//  EXEC  RINGWTR,   PARM=9365RA
```

A file protect ring has to be inserted in a tape reel to enable the tape head to write on the tape. When the tape is mounted on a tape drive, the system will read down to the proper file and wait for the file to be opened for output and no file protect ring is present, the system will rewind the tape, dismount it and request the operator to remount it with a ring. This not only wastes the operator's time but can also consume a large amount of resident time on the tape drive.

A procedure (RINGWTR) is available which enables a user to inform the operator, ahead of the mount request, which tapes require the file protect rings. Its use for output tapes is mandatory. This card must be placed before the steps that require the tapes.

The PARM field contains the serial numbers of the tapes to be mounted with rings. If two or more tape numbers are to be specified, they must be separated by blanks and enclosed in quotes or separated by commas and enclosed in quotes or parentheses.

No tape will be mounted with a ring unless specifically requested by the execution of RINGWRT. The card above, when used will result in a printed message on the console alerting the operator that the tape 9365RA should be mounted with a ring. Any other tape/s requested will be mounted without rings.

```
// EXEC    FORTGCLG
//FORT.SYSIN    DD   *
```

These two JCL cards are the standard ones in a Fortran Program, which ask the system to compile, Link edit and Go.

```
//GO.FT08F001 DD  DSN=SPEECH, UNIT=TAPE9, VOL=SER=9368SP, DISP=OLD,
//   LABEL=(1,NL), DCB=(RECFM=U,BLKSIZE=32760,DEN=2)
//GO.FT09F001  DD  DSN=SENT27,UNIT=TAPE9. VOL=SER=9365RA,
//  LABEL=(21,NL), DISP=(,CATLG), DCB=(RECFM=FB,LRECL=10,BLKSIZE
//                                                    =10000)
```

The above four JCL cards give the details of the two tapes. The first two of these cards give the details of the source tape, while the next two cards give the details of the data tape. Hence these cards are called DD (Data Definition) Cards. It is worthwhile examining each word, its meaning and function.

FT.08F001 refers to a Fortran unit 8. With respect to the Conversion Program, the statement;

```
READ (8,15) (PCM(I),I=1,N)
```

refers to the first JCL card by reading the unit 8.

DSN=SPEECH

Data sets are identified with the DSNAME or DSN (Data Set Name) parameter. It specifies the name of a newly defined data set or refers to the one previously defined. If it is omitted, the system will assign a unique name to that data set. A DSN can be anyone of the following: (i) temporary for the duration of the job, (ii) fully qualified for retrieval from the system catalog, (iii) simple names for data sets that are not cataloged, or (iv) a DSN referenced from another DD statement in the same job.

Here, we have assigned SPEECH as DSN since the tape contains speech data. In the tape bearing serial number 9365RA, the details of which are given by the next two JCL cards, the DSN is described as SENT27. This data set name is referenced in the Table 2.1.1 and has to be changed according to the sentence which is required for any process.

UNIT=TAPE9

The UNIT parameter specifies which input or output device is to be used for the data set. TAPE9 indicates that a 9-track tape is used for the input data of this program.

VOL=SER=9638SP

The VOL parameter requests a specific volume, multiple volumes, specific volumes of multi-volume cataloged data set, or private volumes. A volume can be a tape reel, a disk pack, a bin in a 2321 data cell, a drum or the part of an IBM 2302 disk storage device served by one access mechanism.

VOL must always give the volume serial number for an existing data set unless it is cataloged or passed. A new data set can be created on specific or nonspecific volumes.

Here the serial number of the source tape is given, which is 9638SP.

DISP=OLD

The DISP (Data Set Disposition) parameter describes the current status of the data set (old, new, or to be modified) and directs the system on the disposition of the data set (pass, keep, catalog, uncatalog, or delete) either at the end of the step or if the step abnormally terminates. DISP is always required unless the data set is created and deleted in the same step. The general form or the DISP parameter is as follows:

DISP= (current status, normal disposition, abnormal disposition)

The various options for current status are NEW, OLD, MOD, SHR. The options for the normal disposition are KEEP, DELETE, PASS, CATLG, UNCATLG, and the same for the abnormal disposition are KEEP, DELETE, CATLG, UNCATLG.

Current Status:

NEW: NEW creates a new data set. The UNIT parameter is required, the VOL parameter can be used to place the data set on a specific volume. NEW is default if nothing is coded. This can be seen on the last JCL card for the tape 9365RA, wherein NEW has been ommitted. If DISP is omitted entirely, NEW is also assumed. MOD: This parameter modifies a sequential data set. When the data set is opened, MOD positions the read/write mechanism after the last record in the data set, providing a convenient means of adding data to sequential data set.

OLD: This designates an existing data set; it can be an input data set or a partially complete output data set. The step is given sole access to the data set. In the above JCL cards, the first tape specifies OLD, since it is the input or the source tape.

SHR: This permits the old data sets to be shared. SHR is identical to OLD except that several jobs may use the data set concurrently in multiprogramming environments.

Normal Disposition:
Normal Disposition, the second term in the DISP parameter, indicates the disposition of the data set if the job terminates normally. The normal disposition can be omitted if the data set's status is not to change - existing data sets (MOC,OLD, OR SHR) continue to exist; NEW Data sets are deleted.

KEEP: This term keeps nontemporary data sets; a data set residing on direct-access storage is retained and a data set residing on magnetic tape is rewound and dismounted.
DELETE: This term deletes data sets; storage occupied by a direct-access data set is released; a magnetic tape is rewound and unloaded and the operator is told that the tape data set is deleted (nothing actually happens to the data on tape.)
CATLG: This term catalogs a nontemporary data set. CATLG is similar to KEEP except that the unit and volume of the data set are recorded in the catalog along with the data set name. The file number of a tape data set is also recorded in the catalog, but the type of tape label and the tape density are not.
UNCATLG: This term uncatalogs a data set. UNCATLG is same as KEEP except that the data set name is removed from the catalog. If the data set is not cataloged, UNCATLG is equivalent to KEEP.
PASS: This term passes the data set on to subsequent job steps, and each job step can use the data set at once. PASS saves time because the system retains the data set location and volume information, and a mountable volume containing the data set remains mounted. Both temporary and nontemporary data sets can be passed.

Abnormal Disposition:
The abnormal dispositions, effective only if the step abnormally terminates, are the same as normal dispositions except that PASS is not allowed.
If an abnormal disposition is not specified and the step terminates abnormally, the normal disposition is assumed. However, the abnormal disposition for temporary data sets is always DELETE, regardless of what is coded.

LABEL = (1,NL)

The Label Parameter tells the type of label, the relative file number, on the tape, and whether the data set is to be processed for input or output. The complete LABEL parameter is coded as

```
LABEL=(file, type, PASSWORD,......,.................)
                                IN        RETPD=nnnn
                                OUT       EXPDT=yyddd
```

PASSWORD,EXPDT,and RETPD specify the data set protection and are explained below. The usual form of the LABEL parameter for a tape is

```
LABEL=(file, type, ...........)
                     IN
                     OUT
```

Relative file number: file

If the data set is not the first step on the tape, file gives the relative file number to position the tape properly. The file is a 1 to 4 digit sequence number describing the data set's position relative to other data sets on the volume. If the file number is omitted or zero is coded, then the file number 1 is assumed.

The relative file numbers have to be changed for each record. In the Conversion Program, the source tape has only one file, and hence 1 has to be coded, whereas the data tape has a serial file numver as per the Table 2.1.1 In the Conversion Program, the old records are cataloged under new labels and hence a new serial number has to be alloted everytime a record is cataloged.

The sentence #2 has not been used at all in this project and hence no file numbers have been given to the even numbered sentences (Refer Table 2.1.1).

Label Type: type

Magnetic tapes may contain both volume and data set labels, either standard as proved by IBM, non-standard as devised by the installation, or a combination of both. A tape volume label is a block written on the tape by the installation using the IBM-supplied utility program IEHINITT, before the tape is used.

The types of labels may be specified as follows:

SL	Standard labels (assumed if type omitted)
NSL	Non Standard Labels
SUL	Foth Standard and User Labels
BLP	Bypass Label Processing
NL	No Labels

Processing for Input or Output: IN/OUT

The In and Out Label subparameters are primarily for FORTRAN programs. FORTRAN opens data sets for INOUT or OUTIN, depending upon whether the first use of the data set is a READ or WRITE. Since this opens data sets for both output and input, the file-protection ring must always be inserted even if the tape is only read, negating the protection afforded by file protecting a tape. Likewise, operator invention is required for reading data sets that have retention checks.

Data Set Protection: RETPD, EXPDT, PASSWORD

Data Sets can be protected by retention checks or passwords. Operator response is required before data sets with retention can be modified; they cannot be scratched except by using the IEHPROGM utility.

Retention Check: RETPD, EXPDT
The Label parameter coded on the DD card can assign a retention period on to tape or direct-access data sets, and the operator must give his/her assent before a data set with an unexpired retention date can be modified. When the retention period expires, the data set becomes like any other data set without retention check-- it can be modified or scratched without the oeprator's approval. To request a retention period, code LABEL=RETPD=days, specifying the number of days (9999 maximum) in the retention period. Alternatively, code LABEL=EXPDT=yyddd request an expiration period. The yy is the 2-digit year, and ddd is the 3-digit day number. A retention period of zero days is assumed if retention is not specified.

Password Protection: PASSWORD

A measure of security can be given to the data sets by protecting them with a password. The PASSWORD subparameter, coded when the data set is created, indicates that a password is needed to open the data set. One must record the data set name and a chosen 8-character password in a system data set is named PASSWORD.

PASSWORD is a positional subparameter and RETPD and EXPDT are keyword subparameters. They may all be coded in any combination with other Label subparameters.

```
DCB=(RECFM=U,BLKSIZE=32760,DEN=2)
DCB=(RECFM=FB,LRECL=10,BLKSIZE=10000)
```

DCB Subparameters:

DCB (Data Control Information) is coded as given in the above two examples. Each of the subparameter is explained below:

Record Format: RECFM

Code RECFM= format specifies the record format. The format can be one or more of the following characters.

U	Undefined-length records. BLKSIZE must be greater than or equal to the larges record.
V	Variable-length records. BLKSIZE must be the length of the longest record plus 4 bytes if the records are blocked.
F	Fixed-length records. BLKSIZE must be a multiple of LRECL for blocked records. (U, V and F are mutually exclusive)
B	Blocked records. (FB and VB are permitted; UB is not)
T	Track overflow used.
S	Standard Blocks. For F format records, there are no truncated blocks or unfilled tracks within the data set. For V format records, logical records are spanned over more than one block.
A	ASA control character is the first byte of the data.

M Machine code control character is the first byte of the data (A and M are mutually exclusive)

In the given two JCL card examples above, the Record Length is undefined in the first case and is of fixed-length blocked record for the second case.

Logical Record Length: LRECL

Code LRECL= length, specifies the length of the logical record is bytes for fixed or variable-length records. LRECL is equal to the record length for fixed-length records and is equal to the largest record, including the 4 bytes, describing the record's size, for the variable-length records.

Tape Density: DEN

The following list gives the various values of DEN.

DEN	7-track density	9-track density
0	200 bpi (default) (bytes per inch)	-
1	556 bpi	-
2	800 bpi	800 bpi (default for 800bpi drives)
3	-	1600 bpi (default for 1600 bpi and dual-density drives)

DEN is required for 7-track drives unless the tapes are 200 bpi deisity. DEN is required for 9-track drives only if any 800-bpi tape is used on a dual density drive.

// EXEC DSLIST

This JCL card is used in the Dumpout Program. DSLIST (Data Set Listing) is a program which can be used to list a sequential data set with a block size up to 32760 bytes long. The input set can be a sequential data set from cards, magnetic tapes, or a direct access device; or it can be a member of a partitioned data set on a direct access device. The data being listed can be either character (alphameric) or hexadecimal format. The block number and length of block are printed for each block processed.

PARM field can be given after the DSLIST on that card. PARM can specify print format and line length. Print format is taken as A for characters in case of default, and Lines, which can be a number between 40 and 100, is taken as 100 in case of default.

//LKED.SYSIN DD *

This JCL card is used in the Digitized Sound Spectrograph Program'and it links the main program to the object deck. The Linkage Editor combines and edits modules to produce a single load module that can be brought into main storage by program fetch for execution. It provides several processing facilities that are either performed automatically or invoked as response to control statements prepared for the program.

A Linkage Editor produces CSECT identification record (IDR) in all output load modules.

2.4 EDITING:

Editing or examining of the data can be done if we have each sentence in a written form. Since the data is in the form of numbers only it is easy to get a clear picture of the data. For this a DUMPOUT or DSLIST Fortran program has been used. DSLIST (Data Set Listing) is a program which can be used to list a sequential data set with a blocksize up to 32,760 bytes long. The input data set can be a sequential data set from cards, tapes, or a direct access device. The data being listed can be in either character (alphameric) or hexadecimal format. The block number and length of block are printed for each block processed.

The printed line can be varied (via optional PARM field) from 40 to 100 characters per line. If the input block is longer than the specified line length, the block will be broken into multiple lines.

The print format and line length are controlled by the optional PARM fields on EXECcard. The PARM format is:

DUMPOUT PROGRAM

```
//RCLC3504 JOB (XXXXXXXX,,4),RAINBOLT                                     JOB 155
***TAPE9       1
//  EXEC  DSLIST
//SYSIN  DD  DSN=SENT11,DISP=OLD,LABEL=(8,NL),
//  DCB=(RECFM=FB,LRECL=10,BLKSIZE=10000)

KSU003I PARTITION SIZE - 256K, MAXIMUM CORE USED -  40K,  TIME =  11.22.39
KSU004I I/O COUNT -- DISK =          0,  TAPE =         34
KSU001I STEP   1 GO       EXECUTION TIME =            3.15 SEC    RETURN CODE =     0

KSU002I JOB RCL03504 EXECUTION TIME =            3.15 SEC


HASP-II V3.1 JOB STATISTICS --     5 CARDS READ --   3,297 LINES PRINTED --     0 CARDS PUNCHED --   COPY   1 OF   1
```

RECORD 1, LENGTH 10000

```
 0.0      -0.0157480 0.0      -0.0078740-0.0078740 0.0078740 0.0078740 0.0       0.0078740 0.0
 0.0       0.0       0.0       0.0       0.0       0.0       0.0157480 0.0157480-0.0236220 0.0
 0.0       0.0078740 0.0157480 0.0       0.0       0.0       0.0       0.0       0.0       0.0
 0.0       0.0       0.0       0.0       0.0      -0.0236220 0.0236220-0.0078740 0.0       0.0
 0.0       0.0       0.0157480 0.0      -0.0157480 0.0157480 0.0       0.0      -0.0157480 0.0
 0.0      -0.0314961 0.0078740-0.0157480 0.0      -0.0236220 0.0      -0.0078740 0.0       0.0
-0.0157480 0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
 0.0157480-0.0157480 0.0       0.0      -0.0078740 0.0       0.0      -0.0078740 0.0       0.0078740
 0.0       0.0      -0.0157480 0.0       0.0157480-0.0078740-0.0157480 0.0078740-0.0078740 0.0236220
 0.0      -0.0078740 0.0       0.0       0.0       0.0       0.0       0.0078740-0.0078740 0.0
 0.0       0.0       0.0078740 0.0       0.0078740 0.0       0.0078740 0.0      -0.0157480 0.0
 0.0       0.0078740 0.0236220-0.0078740 0.0       0.0      -0.0078740 0.0236220-0.0078740-0.0078740
 0.0       0.0       0.0      -0.0078740 0.0      -0.0078740 0.0078740-0.0078740-0.0078740 0.0314961
 0.0      -0.0078740 0.0078740 0.0       0.0      -0.0078740 0.0       0.0       0.0       0.0157480
-0.0157480 0.0      -0.0157480 0.0157480 0.0157480-0.0078740 0.0      -0.0078740 0.0236220 0.0157480
-0.0157480 0.0       0.0       0.0236220 0.0078740-0.0157480 0.0078740 0.0       0.0       0.0
 0.0      -0.0236220 0.0       0.0       0.0       0.0      -0.0157480 0.0       0.0      -0.0157480
 0.0236220 0.0      -0.0078740 0.0       0.0157480 0.0      -0.0078740 0.0      -0.0157480 0.0078740
 0.0078740 0.0078740 0.0       0.0      -0.0236220 0.0       0.0236220-0.0157480 0.0236220-0.0078740
 0.0078740 0.0      -0.0157480 0.0       0.0      -0.0157480-0.0314961 0.0157480-0.0157480-0.0078740
 0.0078740-0.0078740-0.0078740 0.0       0.0       0.0157480-0.0157480-0.0078740 0.0      -0.0157480
 0.0       0.0078740-0.0078740 0.0       0.0078740-0.0236220-0.0236220 0.0314961-0.0078740-0.0157480
 0.0078740 0.0157480 0.0      -0.0157480 0.0      -0.0078740 0.0       0.0157480 0.0078740-0.0078740
-0.0157480 0.0157480 0.0       0.0157480-0.0078740 0.0078740 0.0       0.0078740 0.0       0.0157480
-0.0078740 0.0       0.0       0.0       0.0      -0.0078740 0.0       0.0157480 0.0      -0.0157480
 0.0       0.0       0.0       0.0       0.0      -0.0157480 0.0078740 0.0157480 0.0157480-0.0472441
 0.0078740-0.0157480 0.0157480-0.0078740-0.0157480-0.0314961 0.0078740 0.0314961-0.0078740-0.0078740
 0.0078740-0.0078740-0.0078740 0.0157480 0.0      -0.0078740-0.0157480 0.0157480 0.0       0.0157480
-0.0314961 0.0       0.0157480-0.0078740 0.0       0.0      -0.0078740-0.0078740 0.0      -0.0078740
 0.0078740-0.0078740 0.0314961 0.0      -0.0157480-0.0078740 0.0       0.0       0.0       0.0
-0.0157480 0.0078740 0.0078740 0.0       0.0157480 0.0       0.0      -0.0078740 0.0157480 0.0157480
-0.0157480 0.0      -0.0078740 0.0       0.0      -0.0157480 0.0       0.0157480-0.0157480 0.0
 0.0      -0.0157480 0.0       0.0078740-0.0078740-0.0157480 0.0      -0.0078740-0.0078740 0.0
 0.0       0.0       0.0       0.0078740-0.0157480-0.0078740 0.0       0.0236220-0.0157480-0.0078740
 0.0      -0.0236220 0.0      -0.0078740 0.0157480 0.0      -0.0157480 0.0157480-0.0078740 0.0236220
-0.0236220 0.0078740 0.0       0.0      -0.0157480-0.0078740 0.0236220 0.0       0.0      -0.0314961
 0.0078740 0.0       0.0       0.0       0.0       0.0       0.0157480 0.0       0.0      -0.0078740
-0.0078740 0.0       0.0157480 0.0078740-0.0314961 0.0078740 0.0157480 0.0      -0.0078740 0.0
 0.0157480 0.0       0.0      -0.0157480 0.0314961 0.0      -0.0157480 0.0236220 0.0157480 0.0
-0.0157480 0.0       0.0314961 0.0       0.0       0.0      -0.0078740 0.0       0.0      -0.0078740
 0.0157480 0.0078740 0.0       0.0078740-0.0078740-0.0157480 0.0314961 0.0      -0.0314961 0.0
-0.0078740 0.0      -0.0157480 0.0       0.0       0.0      -0.0078740 0.0157480 0.0       0.0
 0.0      -0.0078740 0.0078740-0.0078740 0.0       0.0       0.0       0.0       0.0078740 0.0
-0.0157480 0.0157480 0.0       0.0157480-0.0314961 0.0       0.0157480 0.0078740-0.0157480 0.0078740
 0.0078740 0.0       0.0       0.0       0.0       0.0       0.0       0.0157480-0.0157480 0.0
 0.0      -0.0157480 0.0078740 0.0078740-0.0078740 0.0       0.0       0.0       0.0       0.0078740
 0.0       0.0       0.0      -0.0078740-0.0078740-0.0078740-0.0157480 0.0      -0.0314961 0.0236220
 0.0078740-0.0157480 0.0      -0.0157480 0.0236220 0.0       0.0157480 0.0078740 0.0157480-0.0236220
 0.0157480-0.0078740 0.0       0.0      -0.0393701 0.0       0.0157480-0.0078740-0.0314961-0.0157480
 0.0      -0.0078740 0.0236220 0.0157480-0.0078740 0.0       0.0078740 0.0       0.0       0.0
 0.0078740 0.0       0.0078740 0.0       0.0       0.0078740 0.0       0.0157480 0.0       0.0
 0.0078740-0.0078740-0.0078740 0.0078740 0.0       0.0314961-0.0078740 0.0078740-0.0314961 0.0078740
-0.0078740 0.0157480 0.0157480-0.0078740 0.0       0.0078740 0.0       0.0      -0.0236220 0.0
 0.0236220 0.0       0.0      -0.0078740 0.0      -0.0078740 0.0157480 0.0       0.0       0.0
 0.0       0.0      -0.0078740 0.0078740 0.0      -0.0078740 0.0       0.0157480 0.0       0.0
 0.0       0.0       0.0       0.0       0.0       0.0       0.0078740 0.0       0.0       0.0
 0.0       0.0       0.0       0.0078740 0.0157480-0.0157480 0.0078740 0.0       0.0      -0.0236220
 0.0236220 0.0078740 0.0       0.0       0.0      -0.0078740 0.0       0.0       0.0157480-0.0157480
-0.0157480 0.0236220 0.0       0.0      -0.0157480-0.0078740 0.0       0.0       0.0      -0.0078740
 0.0       0.0078740-0.0078740 0.0       0.0       0.0      -0.0157480-0.0078740 0.0       0.0
```

PARM=(print format, linelength)

where:

print format= A for character (default)

H for Hexadecimal and Character format

line length= number between 40 and 100 (default 100)

The program is very simple, consisting of 4 JCL cards only. The format of JCL cards have been explained in section 2.3 However, there is one interesting feature to note. The JCL card does not specify the serial number of the tape, but it connects with the correct tape only by Data Set Name (DSN) and the Label number (LABEL).

The output is just a list of whatever was on the tape under the particular Label in form of blocks. The aim of getting this output is to have an exact idea as to what parts of the each record have any voice in it. Wherever the tape is blank, the output consists a string of zeros. This is usually the case at the beginning and the end of each record. While taking the Chirp-Z Transform and the Cepstrum, it is essential to know this, in order to avoid unnecessary waste of Computer time.

2.5 DIGITALIZED SOUND SPECTROGRAPH USING FAST FOURIER TRANSFORM AND MULTIPRINT TECHNIQUES

Sound Spectrographs provide a means for permanently displaying the short-time power spectrum of a speech signal. In analog devices, this is done by 'burning' traces of the frequency analysis results to current-sensitive facsimile paper. The abscissa of a spectrogram is time, the ordinate is frequency (linear scale) and the darkness of the pattern represents the intensity (logarithmic scale).

Since the advent of the Fast Fourier Transform (FFT), spectral analysis has become economically feasible with digital computers. The high speed chain printer provides a means of generating different grey-levels by multiprinting appropriate sequences of characters out of its basic 48-character set. The various processing steps and algorithms used for the estimation of

SPECTROGRAPH PROGRAM

```
//XPRS9300 JOB (XXXXXXXX,,2,,1902,,,C),RAINBOLT,TIME=(1,30)            JOB 137
***TAPE?       1
//      EXEC    FORTGCLG
//FORT.SYSIN   DD    *

KSU003I PARTITION SIZE - 256K, MAXIMUM CORE USED -  88K,  TIME =  11.16.25
KSU004I I/O COUNT -- DISK =          3,   TAPE =          0
KSU001I STEP    1 FORT      EXECUTION TIME =           2.27 SEC    RETURN CODE =    0

//LKED.SYSIN    DD    *

KSU003I PARTITION SIZE - 256K, MAXIMUM CORE USED - 156K,  TIME =  11.16.44
KSU004I I/O COUNT -- DISK =        139,   TAPE =          0
KSU001I STEP    2 LKED      EXECUTION TIME =           1.04 SEC    RETURN CODE =    0

//GO.SYSIN     DD    *
//GO.FT08F001   DD  DSN=SENT31,DISP=OLD,LABEL=(10,NL),
//     DCB=(RECFM=FB,LRECL=10,BLKSIZE=10000)
//GO.SYSPRINT DD  UNIT=AFF=FT08F001
//GO.SYSDUMP      DD     SYSOUT=A

KSU003I PARTITION SIZE - 256K, MAXIMUM CORE USED - 178K,  TIME =  11.20.55
KSU004I I/O COUNT -- DISK =          0,   TAPE =         31
KSU001I STEP    3 GO        EXECUTION TIME =     1 MIN 10.94 SEC    RETURN CODE =    0

KSU002I JOB XPRS9300 EXECUTION TIME =      1 MIN 14.25 SEC


HASP-II V3.1 JOB STATISTICS --     131 CARDS READ --     1,896 LINES PRINTED --       0 CARDS PUNCHED --   COPY   1 OF   1
```

```
      COMPLEX X(512),CMPLX
      DIMENSION XX(30000),IP(132),W(512)
      DIMENSION P(132)
    1 FORMAT('1')
      L=7
      READ 5,NN,N,NS,LOG2N,DBBST
    5 FORMAT (4I5,F5.0)
      NB=(NN/N-1)*N/NS+1
      N2=129
      FS=8333.33
      FO=FS/N
      FNY=FS/2.
      R2=2**0.5
      READ (5,10) (XX(I),I=1,NN)
   10 FORMAT (F10.7)
      PRINT 15, NN,N,NB,FO,FNY
   15 FORMAT('1',10X,'TOTAL NUMBER OF POINTS =',I5,5X,'NUMBER OF POINTS
     *IN A SECTION =',I4,5X,'NUMBER OF OVERLAPPING SECTIONS =',I3,
     */,'0',10X,'FREQUENCY RESOLUTION =',F4.0,' HZ.',5X,'NYQUIST ',
     *'FREQUENCY =',F5.0,' HZ.')
      PRINT 1
      TWOPI=6.283185
      DO 18 I=1,N
      ARG=TWOPI*FLOAT(I-1)/FLOAT(N)
   18 W(I)=0.5*(1.-COS(ARG))
      DO 100 II=1,NB
      I1=(II-1)*NS+1
      I2=I1+N-1
      K=0
      DO 20 I=I1,I2
      K=K+1
      WX=W(K)*XX(I)
   20 X(K)=CMPLX(WX,0.)
      CALL FFTP(X,LOG2N,L)
      P(1)=CABS(X(1))
      DO 25 I=2,N2
      P(I)=CABS(X(I))*R2
   25 CONTINUE
      DO 30 I=1,N2
      IF ((I.EQ.1).OR.(I.EQ.N2)) GO TO 26
      PA=(P(I-1)+P(I)+P(I+1))/3.
      GO TO 27
   26 PA=P(I)
   27 IF (PA .LT. 1.E-3) GO TO 28
      IP(I)=20.*ALOG10(PA)+DBBST
      IF (IP(I) .GE. 0) GO TO 30
   28 IP(I)=0
   30 CONTINUE
      CALL SPCTGR(IP,N2)
  100 CONTINUE
      STOP
      END
```

23B

FORTRAN IV G LEVEL 21 FFTP DATE = 74156 11/16/13 PAGE 0001

```
      SUBROUTINE FFTP(X,M,L)
      COMPLEX X(512),U,W,T,CMPLX
      PI=3.141592
      N=2**M
      N1=2**L
      NV2=N/2
      NM1=N-1
      J=1
      DO 7 I=1,NM1
      IF(I.GE.J) GO TO 5
      T=X(J)
      X(J)=X(I)
      X(I)=T
    5 K=NV2
    6 IF(K.GE.J) GO TO 7
      J=J-K
      K=K/2
      GO TO 6
    7 J=J+K
      DO 40 LO=1,M
      LE=2**LO
      LE1=LE/2
      U=CMPLX(1.0,0.0)
      W=CMPLX(COS(PI/LE1),-SIN(PI/LE1))
      IF(LO-L) 20,20,30
   20 DO 11 J=1,LE1
      DO 9 I=J,N,LE
      IP=I+LE1
      T=X(IP)*U
      X(IP)=X(I)-T
    9 X(I)=X(I)+T
   11 U=U*W
      GO TO 40
   30 DO 12 J=1,N1
      DO 10 I=J,N,LE
      IP=I+LE1
   10 X(I)=X(I)+X(IP)*U
   12 U=U*W
   40 CONTINUE
      RETURN
      END
```

FORTRAN IV G LEVEL 21 SPCTGR DATE = 74156 11/16/13 PAGE 0001

```
      SUBROUTINE SPCTGR(IP,N)
      DIMENSION IP(1),IOUT(132)
      DO 50 I=1,N
      L=IP(I)/8+1
      IF (L.GT.8) L=8
      IOUT(I)=L
   50 CONTINUE
      CALL PRINTR(IOUT,N)
      RETURN
      END
```

TOTAL NUMBER OF POINTS =29696   NUMBER OF POINTS IN A SECTION = 256   NUMBER OF OVERLAPPING SECTIONS =461

FREQUENCY RESOLUTION = 33. HZ.   NYQUIST FREQUENCY =4167. HZ.

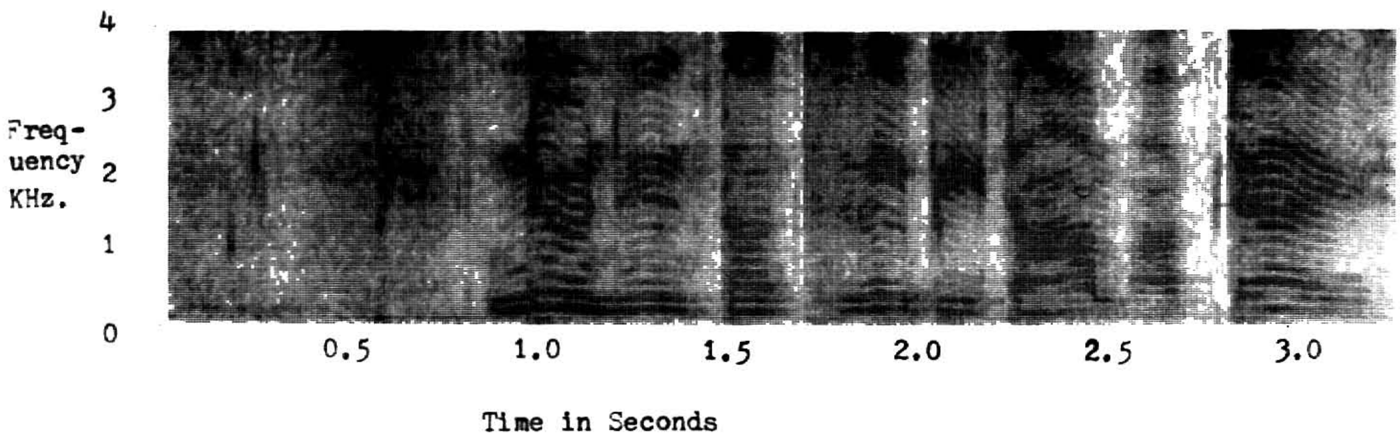


Fig. 2.5.4 Digital Spectrograph of "Noon is a sleepy time of day "

SAMPLED TIME FUNCTION

SECTIONING OF THE INPUT RECORD INTO OVERLAPPING ARRAYS OF N SAMPLES

$X'(n)$ $\qquad n=0,1,2,\ldots\ldots N-1$

SHAPING OF THE N POINT ARRAY WITH A SUITABLE WINDOW FUNCTION

$X(n)=X'(n).W(n) \qquad n=0,1,2,\ldots.N-1$

FOURIER TRANSFORMATION OF THE N REAL POINT ARRAY (FFT)

$$A(n)= \frac{1}{N}\sum_{p=0}^{N-1} X(p)\ .\exp(-j\frac{pn}{N}) \qquad n=0,1,2,\ldots N-1$$

CALCULATION OF THE PERIODGRAM

$$I(n)=\frac{N}{2\pi}\ |A(n)|^2 \qquad n=0,1,2,\ldots N-1$$

SMOOTHING OF THE PERIODGRAM (MOVING AVERAGE OVER K SAMPLES)

$$P(n)= \frac{1}{2k+1}\sum_{q=-k}^{k} I(n-q) \qquad n=0,1,2,\ldots N-1$$

POWER SPECTRUM ESTIMATE

Fig 2.5.1

FLOWGRAPH FOR ESTIMATION OF SHORTTIME POWER SPECTRUM short-time power spectra in this system are tabulated in Fig. 2.5.1. After digital recording of the speech signal it is sectioned into records which may be overlapping in time. The records are then shaped by a suitable window function, a procedure which is the prerequisite for a good spectral resolution and also for bandlimiting the function in the frequency domain. Further processing includes a discrete Fourier transformation via a special subroutine which uses FFT. Next, the power spectrum is obtained by taking the absolute value of the FFT output, followed by a smoothing in frequency domain in order to get a more consistent estimate of the spectrum.

The effect of windowing is best illustrated by comparing Fig. 2.5.2 and 2.5.3(7). The lower parts of both figures show the power spectrum of a section taken from a sinusoidal time function. The width of each window is the same, whereas the shapes are different. This can be seen from the upper parts of the printouts. In Fig. 2.5.2, a rectangular window was used i.e., the values of the input function were not changed. This yields a power spectrum with a rather sharp peak but poor attentation of the lower and higher frequency components. In Fig. 2.5.3, a cosine arch-shaped window, generally called a Hanning window, was used for windowing the input samples i.e., the input function smoothly tapers off to zero at both ends. As a consequence, the power spectrum is much more similar to the expected 'line' in the frequency domain. The superior attenuation at the higher and lower frequencies has been traded off with a broadening of the central peak.

To construct the spectrograms, consecutively calculated power spectra are plotted, line after line as a density modulated string. Using the basic character set of a standard chain printer, it is quite easy to find a character sequence which by superposition (or multiprinting) of maximally four characters gives a perceptually linear gradation in eight grades of grey. The character combinations used for the grey tones were proposed by Stucki (9(are given in Table 2.5.1 .

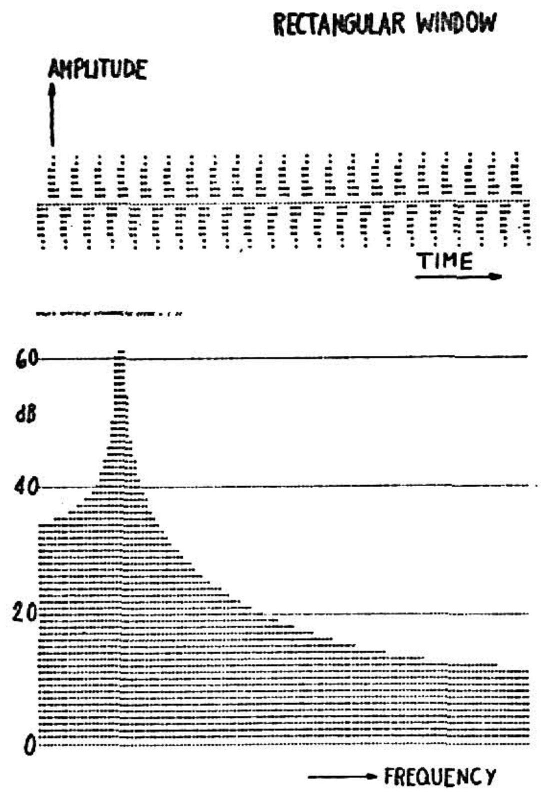


Fig.2.5.2 Power Spectrum Using a Rectangular Window

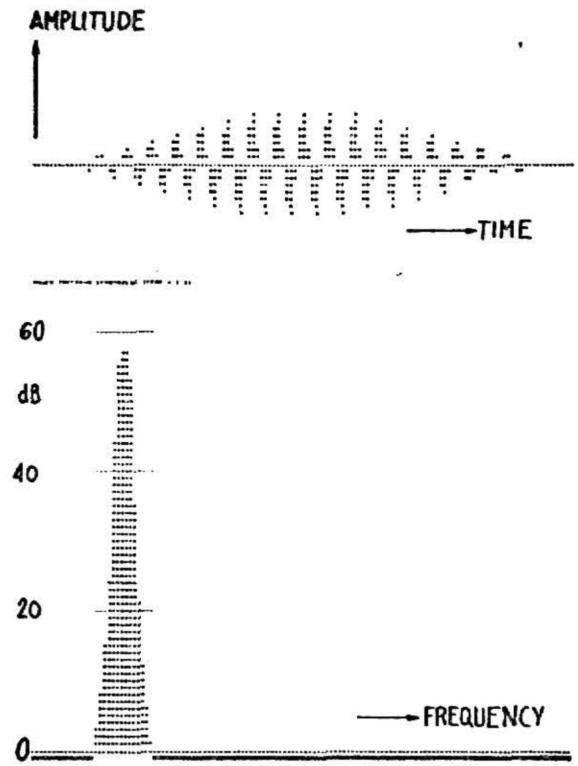
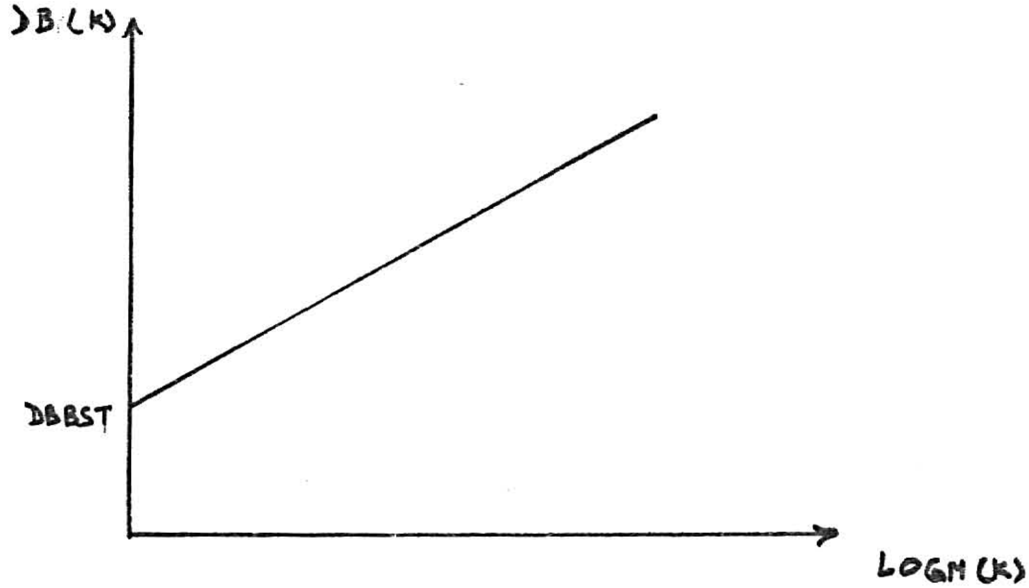


Fig. 2.5.3 Power Spectrum Using a Hanning Window

Grey Tone	1	2	3	4	5	6	7	8
1st print		-	=	+	X	X	O	O
2nd print				+	X	X	X	W
3rd print						=	*	M
4th print								*

Table 2.5.1
The character combination for the grey tones proposed by Stucki.
The general approach described above points out the possibilities of a very standard computer-center for speech and noise analysis work, once the initial hurdle of digitizing the analog signals has been overcome.

Now, the specific computer program used for this project will be discussed. The program closely follows the flow graph of Fig. 2.5.1. The first 18 statements are concerned with reading and printing of the sampled time function and the parameters used in the program. Each block processed at a time consists of 256 points. The total number of points (NN), is taken as a multiple of 256 such that that portion of the record where voice is present is covered. For this reason, the Dumpout Program output is examined where voice is present, is selected. That is, portions at the beginning and the end of a sentence, where silence is present, are deleted. The frequency resolution of the plot is 33Hz. (32.556Hz. to be precise). Since the sampling frequency is $8333,\bar{3}$ Hz., the Nyquist or the folding frequency upto which the transform is taken is 4167. Hz. ($4166.\bar{6}$Hz. to be precise). Statements 20 through 30 implement the window function, $W(m)=\frac{1}{2}(1.0 - \mathrm{Cos}(2\pi m/N)), m=0,1,2,.....255$ that is for a block of 256 data points. Next statements 31 through 35 deal with the FFT computation of data block of size 256. This is done by calling the subroutine FFTP. Absolute values of the transform coefficients are then computed. The next four statements cause a moving average over three consecutive values to be taken. This process smoothes periodgram. Next, the decibel value of the average is taken and a constant "DB booster, (DB-BST)" is added in order to get a darker spectrograms.



Next, the Spectrograph Subroutine is called, which prints the actual spectrograph of the speech portions. In the actual speech spectrograph, each vertical line is 30 milliseconds in time (29.855, to be exact). The frequency resolution is 33Hz. and hence each horizontal line is 33Hz. beginning with 0 at the bottom and ending at 4167.Hz at the top.

The printer prints one vertical line for each block. As such, there are as many vertical lines as there are blocks. If NB is the number of blocks, then

$$NB = \left(\frac{NN}{N} \quad -1 \quad \frac{N}{NB} \right) + 1$$

where	NN	is the total number of points taken
	N	is the total number of points in a section, which is 256
	NS	is the number of overlapping points in a section, which is 64 (usually N/4)

CHAPTER III

CHIRP-Z TRANSFORM PROGRAM

3.1 CHIRP-Z TRANSFORM

The Chirp-Z Transform (CZT) was developed by Rabiner, Schafer, and Rader (7). It can be used to secure a high-resolution short-time spectral analysis of speech data. Let $X(m) = X_o$, X_1, $X_2, \ldots X_{N-1}$ denote a data sequence, which is obtained by sampling a band-limited signal $x(t)$, at the rate of $1/T$ samples per second. The CZT of $X(m)$ is defined as

$$X_k = X(Z_k) = \sum_{n=o}^{N-1} X_n Z_k^{-n} \qquad K=0,1,\ldots(M-1) \quad (3.1.1)$$

where

$$Z_k = (A_o e^{j2\pi\theta_o})(W_o e^{j2\pi\phi_o})^{-k} \qquad (3.1.2)$$

Equation (3.1.2) describes a set of M points equally spaced on logarithmic spiral. Since the transformation from Z-plans to the S-plane is given by

$$s = (1/T)\ \ln Z \qquad (3.1.3)$$

it follows that

$$S_k = (1/T)(\ln A_o + j2\pi\theta_0) - (K/T)(\ln W_o + j2\pi\phi_o) \qquad (3.1.4)$$

The geometrical interpretation of (3.1.4) is illustrated in Fig. 3.1.1 for a contour of 8 points. In this figure, we observe that the Z-plane contour maps into a straight line of arbitrary length and orientation in the S-plane.

As we are interested only in real values of the frequency (i.e., operating on imaginary axis), (3.1.4) can be simplified by substituting $A_o = W_o = 1$. That is

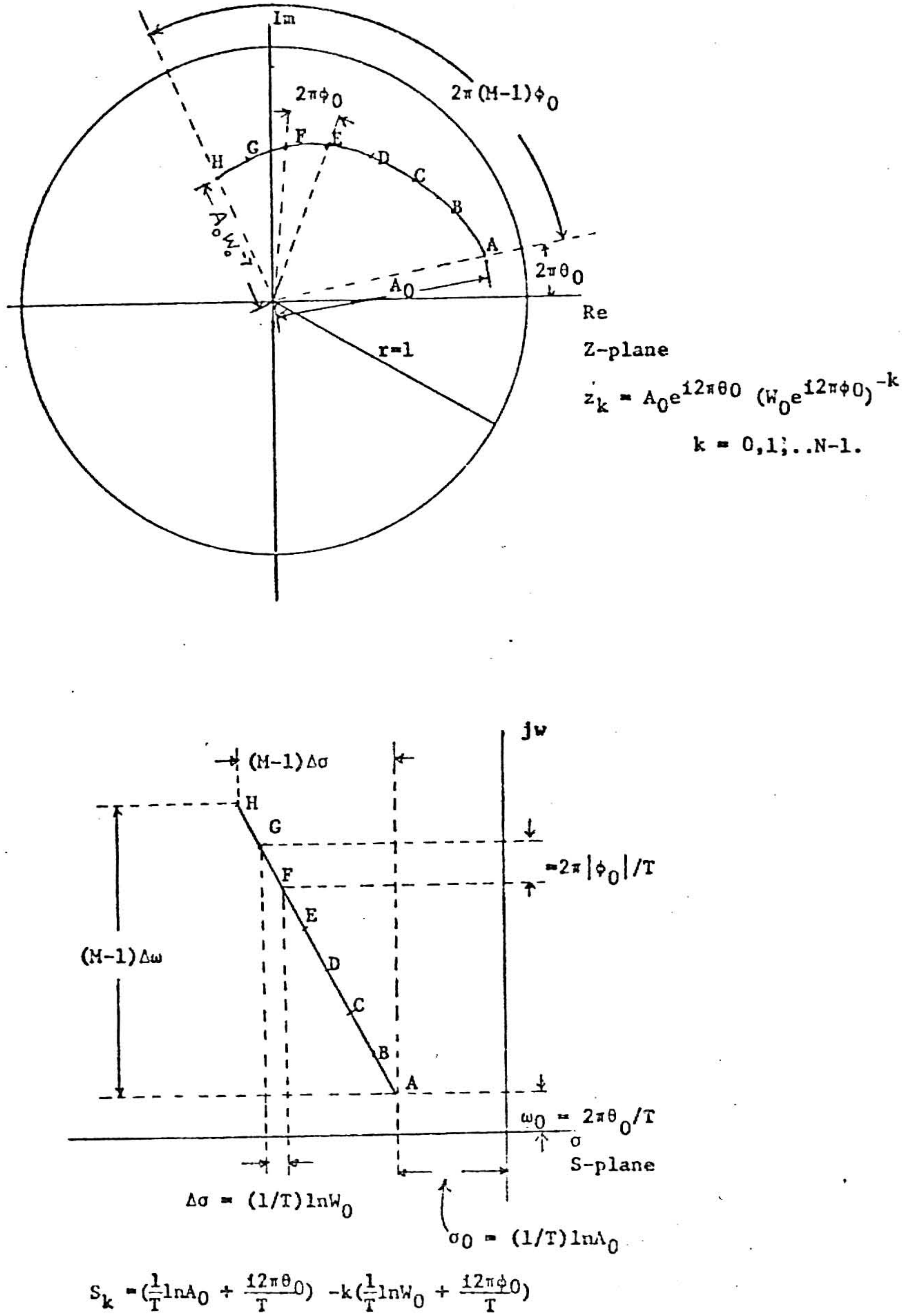


Fig.3.1.1 A general CZT contour with M=8.

$$S = \frac{j}{T} \quad 2\pi\,\theta_o - \frac{j2\pi\phi_o}{T} \qquad (3.1.5)$$

Equation (3.1.5) implies that the points on the imaginary axis in the S-plane are mapped on to the unit circle in the Z-plane. Equations (3.1.1) and (3.1.2) yield

$$X_k = \sum_{n=o}^{N-1} X\, A^{-n} W^{kn} \qquad k=0,1,\ldots(M-1) \qquad (3.1.6)$$

where

$$A = \exp\,(j2\pi\theta_o)$$

$$W = \exp\,(j2\pi\phi_o)$$

If we let $A_o = W_o = 1$, $\theta_o = 0$, $M=N$ and $\phi_o = -1/N$, then the Eq. (3.1.6) becomes

$$X(Z_k) = \sum_{n=0}^{N-1} X_n \exp(-j2\pi\ /N)\ nk \qquad k=0,1,\ldots(N-1) \qquad (3.1.7)$$

Eq. (3.1.7) is known as the discrete Fourier Transform (DFT). The DFT is sometimes defined as

$$X(Z_k) = \frac{1}{N} \sum_{n=0}^{N-1} X_n \exp(-j2\pi/N)nk \qquad (3.1.8)$$

$$k=o,1,2,\ldots(N-1)$$

We remark that the only difference between the above definitions is merely the constant multiplier 1/N.

3.2 CHIRP-Z TRANSFORM PROGRAM

A basic application of this program with respect to speech processing is to obtain a high resolution DFT of successive blocks of speech data. Each block of data consists of approximately 30 milliseconds of real-time speech. The related computer program consists of the CZT and FFT subroutines. The

CHIRP Z-TRANSFORM PROGRAM

```
//RCL05874 JOB (XXXXXXXXX,XXXXXXXX,,3,,1602),NIRANJAN,TIME=(2,30)        JOB 564
***TAPE9        1
//    EXEC    FORTGCLG
//FORT.SYSIN    DD  *

KSU003I PARTITION SIZE - 256K, MAXIMUM CORE USED -  92K,  TIME =  09.48.57
KSU004I I/O COUNT -- DISK =         5,  TAPE =         0
KSU001I STEP    1 FORT      EXECUTION TIME =           3.86 SEC    RETURN CODE =     0

KSU003I PARTITION SIZE - 256K, MAXIMUM CORE USED - 156K,  TIME =  09.49.20
KSU004I I/O COUNT -- DISK =       125,  TAPE =         0
KSU001I STEP    2 LKED      EXECUTION TIME =           0.94 SEC    RETURN CODE =     0

//GO.SYSIN    DD   *
//GO.FT08F001   DD  DSN=SENT05,DISP=OLD,LABEL=(5,NL),
//   DCB=(RECFM=FB,LRECL=10,BLKSIZE=10000)

KSU003I PARTITION SIZE - 256K, MAXIMUM CORE USED -  70K,  TIME =  09.52.57
KSU004I I/O COUNT -- DISK =         0,  TAPE =        26
KSU001I STEP    3 GO        EXECUTION TIME =     1 MIN 18.49 SEC    RETURN CODE =     0

KSU002I JOB RCL05874 EXECUTION TIME =     1 MIN 23.29 SEC


HASP-II V3.1 JOB STATISTICS --    225 CARDS READ --    1,365 LINES PRINTED --      0 CARDS PUNCHED --    COPY    1 OF    1
```

```
C     HIGH RESOLUTION NARROW BAND SPECTRA
C     USING PAY-CZT FFT
C
      COMPLEX XDC(512),Y(512),MULT(256),H1,CMPLX
      DIMENSION XX(256)
      DIMENSION W(256)
      DIMENSION P(120,2)
      COMMON MULT
      COMMON /CHRP/ DN1,DN2
C
C        READ CONTROL PARAMETRS
C
      READ 102,NN,M,L,LOG2N,LOG2M,LOG2L,FL,FU,DF,FS
  102 FORMAT(6I4,4F8.2)
      PRINT 104,NN,M,L,FL,FU,DF
  104 FORMAT('1',10X,'N =',I4,5X,'M=',I4,5X,'L =',I4,5X,
     1 'LOWER FRECUENCY =',F8.2,' HZ',5X,'UPPER FREQUENCY =',F8.2,' HZ'
     2,5X,/,' ',10X,'RESOLUTION =',F5.2,' HZ')
      READ 105,NB
  105 FORMAT(I4)
      PRINT 106,NB
  106 FORMAT('0',10X,'NUMBER OF BLOCKS PROCESSED :',I5)
C
C
      PRINT 110,FL,FU
  110 FORMAT('0',10X,'HIGH RESOLUTION LOG.MAG. SPECTRUM BETWEEN',
     1 F4.0,' HZ. AND',F5.0,' HZ.')
      DO 15 I=1,NN
      ARG=6.283185*FLOAT(I-1)/FLOAT(NN)
   15 W(I)=0.5*(1.-COS(ARG))
      L2=L/2
      CALL WPOWER(L2)
      DT=1./FS
      RWPH=-DF*DT
      RAPH=FL*DT
      R2PI=6.283185307
      DN1=R2PI*RAPH
      DN2=R2PI*RWPH
      DNN1=DN1*FLOAT(NN)
      DNN2=DN2*FLOAT(NN)
      DO 500 IB=1,NB
      READ (8,200) (XX(J),J=1,NN)
  200 FORMAT(F10.7)
      DO 230 J=1,NN
      WX=W(J)*XX(J)
      XDC(J)=CMPLX(WX,0.)
  230 CONTINUE
      CALL CHIRPZ(XDC,NN,M,L,LOG2N,LOG2M,LOG2L,Y)
      DO 240 J=1,M
      ABS=CABS(Y(J))
  240 XX(J)=20.*ALOG10(ABS)+4.
      I=1
  245 I=I+1
      IF(I.EQ.M)GO TO 247
      IF (XX(I).LT.XX(I-1))GO TO 246
      GO TO 245
  246 P(IB,1)=FL-1.+FLOAT(I-1)
      P(IB,2)=XX(I-1)
```

```
            GO TO 248
      247   P(IB,1)=FU
            P(IB,2)=XX(M)
      248   CONTINUE
            PRINT 250,(XX(J),J=1,M)
      250   FORMAT('0',/,(' ',16F8.2))
      500   CONTINUE
            PRINT 505
      505   FORMAT ('1',10X,'TIME(SEC)',5X,'FUND.FREQ.(HZ)',5X,'DB',//)
            TIME=0.0
            DO 510 I=1,NB
            TIME=TIME+NN*DT
      510   PRINT 515,TIME,P(I,1),P(I,2)
      515   FORMAT (' ',11X,F6.3,13X,F4.0,9X,F6.2)
            STOP
            END
```

```
      SUBROUTINE CHIRPZ(X,NN,M,L,LOG2N,LOG2M,LOG2L,GK)
      COMPLEX X(512),V(512),GK(512),MULT(256),C1,C2,CMPLX
      COMMON MULT
      COMMON /CHRP/DN1,DN2
C
      DO 100 I=1,NN
      FIM1=FLOAT(I-1)
      THETA1=FIM1*(DN2*FIM1/2.0-DN1)
      C1=CMPLX(COS(THETA1),SIN(THETA1))
      X(I)=X(I)*C1
  100 CONTINUE
      CALL FFT(X,LOG2L,LOG2N,L,NN)
C
C        FORM THE FOLLOWING L POINT SEQUENCE
      DO 120 I=1,M
      FIM1=FLOAT(I-1)
      THETA2=DN2*FIM1**2/2.0
      V(I)=CMPLX(COS(THETA2),-SIN(THETA2))
  120 CONTINUE
      IF (L.EQ.M+NN-1) GO TO 126
      LMNN1=L-NN+1
      M1=M+1
      DO 125 I=M1,LMNN1
      V(I)=CMPLX(0.0,0.0)
  125 CONTINUE
  126 CONTINUE
      LMNP2=L-NN+2
      DO 130 I=LMNP2,L
      FIM1=FLOAT(L-I+1)
      THETA2=DN2*FIM1**2/2.0
      V(I)=CMPLX(COS(THETA2),-SIN(THETA2))
  130 CONTINUE
C
C        COMPUTE THE L POINT DFT OF V
C
      CALL FFT(V,LOG2L,LOG2L,L,L)
      DO 140 I=1,L
      GK(I)=X(I)*V(I)
  140 CONTINUE
C
C     COMPUTE THE IDFT OF GK
C
      CALL IFFT(GK,LOG2L,LOG2M,L,M)
C
C
C     COMPUTE XK
C
      DO 150 I=1,M
      FIM1=FLOAT(I-1)
      THETA3=DN2*FIM1**2/2.0
      C2=CMPLX(COS(THETA3),SIN(THETA3))
      GK(I)=C2*GK(I)
  150 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FFT(X,M,L,N,L2)
      COMMON MULT
      COMPLEX MULT(256)
      COMPLEX X(512),W,T,U,CMPLX,CONJG
      DATA PI/3.141592/
      K=M-L
      LMX=N
      IND=1
      DO 50 LO=1,M
      LIX=LMX
      LMX=LMX/2
      IP=1
      IF (LO.GT.K) GO TO 30
      DO 20 LM=1,L2
      DO 20 LI=LIX,N,LIX
      J1=LI-LIX+LM
      J2=J1+LMX
   20 X(J2)=MULT(J1)*X(J1)
      GO TO 45
   30 DO 40 LM=1,LMX
      IF (LM.NE.1) GO TO 32
      W=(1.,0.)
      GO TO 34
   32 IP=IP+IND
      W=MULT(IP)
   34 CONTINUE
      DO 40 LI=LIX,N,LIX
      J1=LI-LIX+LM
      J2=J1+LMX
      T=X(J1)-X(J2)
      X(J1)=X(J1)+X(J2)
   40 X(J2)=W*T
   45 IND=IND+IND
   50 CONTINUE
      RETURN
      ENTRY IFFT(X,M,L,N,N1)
      LE=1
      IND=N
      DO 65 LO=1,M
      IND=IND/2
      LE1=LE
      LE=LE+LE
      IPP=1
      IF (LO.GT.L) GO TO 33
      DO 11 J=1,LE1
      IF (J.NE.1) GO TO 7
      U=(1.,0.)
      GO TO 8
    7 IPP=IPP+IND
      U=CONJG(MULT(IPP))
    8 CONTINUE
      DO 9 I=J,N,LE
      IP=I+LE1
      T=X(IP)*U
      X(IP)=X(I)-T
    9 X(I)=X(I)+T
   11 CONTINUE
      GO TO 65
```

```
      33 DO 12 J=1,N1
         IF (J.NE.1) GO TO 14
         U=(1.,0.)
         GO TO 15
      14 IPP=IPP+IND
         U=CONJG(MULT(IPP))
      15 CONTINUE
         DO 10 I=J,N,LE
         IP=I+LE1
      10 X(I)=X(I)+X(IP)*U
      12 CONTINUE
      65 CONTINUE
         DO 66 I=1,N
      66 X(I)=X(I)/N
         RETURN
         END
```

```
FORTRAN IV G LEVEL  21              WPOWER            DATE = 74239          09/48/41            PAGE 0001

                  SUBROUTINE WPOWER(N)
                  COMMON MULT
                  COMPLEX MULT(256),CMPLX
                  MULT(1)=(1.,0.)
                  ARG1=3.141592/FLOAT(N)
                  DO 10 I=2,N
                  ARG2=(I-1)*ARG1
               10 MULT(I)=CMPLX(COS(ARG2),-SIN(ARG2))
                  RETURN
                  END
```

N = 256 M= 128 L = 512 LOWER FREQUENCY = 75.00 HZ UPPER FREQUENCY = 203.00 HZ
RESOLUTION = 1.00 HZ

NUMBER OF BLOCKS PROCESSED : 95

HIGH RESOLUTION LOG.MAG. SPECTRUM BETWEEN 75. HZ. AND 203. HZ.

-16.29	-16.12	-16.03	-15.93	-15.83	-15.72	-15.60	-15.48	-15.35	-15.22	-15.09	-14.96	-14.82	-14.68	-14.54	-14.40
-14.27	-14.13	-14.00	-13.87	-13.74	-13.62	-13.50	-13.38	-13.29	-13.17	-13.03	-12.99	-12.90	-12.82	-12.75	-12.69
-12.63	-12.59	-12.54	-12.51	-12.49	-12.47	-12.46	-12.46	-12.46	-12.48	-12.51	-12.54	-12.58	-12.63	-12.70	-12.77
-12.84	-12.93	-13.03	-13.14	-13.26	-13.38	-13.52	-13.66	-13.82	-13.98	-14.15	-14.33	-14.52	-14.72	-14.93	-15.14
-15.35	-15.58	-15.81	-16.04	-16.27	-16.51	-16.74	-16.96	-17.17	-17.38	-17.57	-17.74	-17.89	-18.01	-18.11	-18.15
-18.22	-18.23	-18.21	-18.16	-18.08	-17.99	-17.86	-17.72	-17.56	-17.39	-17.22	-17.04	-16.86	-16.68	-16.50	-16.32
-16.15	-15.99	-15.83	-15.69	-15.55	-15.42	-15.31	-15.20	-15.11	-15.02	-14.95	-14.89	-14.84	-14.81	-14.78	-14.77
-14.77	-14.78	-14.80	-14.83	-14.88	-14.93	-14.99	-15.07	-15.15	-15.24	-15.34	-15.44	-15.55	-15.67	-15.78	-15.90
-12.74	-12.84	-12.89	-12.92	-12.94	-12.94	-12.93	-12.89	-12.84	-12.76	-12.68	-12.57	-12.45	-12.32	-12.18	-12.03
-11.86	-11.69	-11.52	-11.33	-11.15	-10.96	-10.76	-10.56	-10.36	-10.16	-9.95	-9.74	-9.53	-9.31	-9.09	-8.87
-8.64	-8.41	-8.17	-7.93	-7.68	-7.43	-7.18	-6.91	-6.65	-6.38	-6.10	-5.82	-5.54	-5.26	-4.97	-4.69
-4.40	-4.11	-3.82	-3.53	-3.24	-2.96	-2.67	-2.39	-2.11	-1.84	-1.57	-1.30	-1.04	-0.79	-0.53	-0.28
-0.04	0.19	0.42	0.65	0.86	1.08	1.28	1.48	1.67	1.86	2.04	2.21	2.38	2.54	2.69	2.84
2.98	3.11	3.24	3.36	3.48	3.59	3.69	3.78	3.88	3.96	4.04	4.12	4.19	4.25	4.31	4.37
4.42	4.47	4.51	4.55	4.59	4.63	4.67	4.70	4.74	4.77	4.81	4.85	4.88	4.93	4.97	5.02
5.07	5.13	5.20	5.26	5.34	5.42	5.51	5.61	5.71	5.82	5.94	6.06	6.19	6.33	6.47	6.62
3.54	3.82	4.10	4.37	4.63	4.88	5.13	5.36	5.59	5.80	6.00	6.20	6.39	6.55	6.72	6.87
7.01	7.15	7.27	7.39	7.50	7.60	7.69	7.78	7.86	7.93	8.00	8.06	8.12	8.15	8.23	8.28
8.32	8.37	8.42	8.46	8.51	8.56	8.61	8.66	8.71	8.77	8.83	8.87	8.95	9.03	9.10	9.17
9.25	9.33	9.40	9.48	9.56	9.64	9.71	9.79	9.85	9.92	9.98	10.03	10.07	10.11	10.13	10.14
10.14	10.13	10.09	10.04	9.97	9.89	9.77	9.62	9.45	9.24	9.00	8.71	8.38	7.99	7.53	7.30
6.37	5.54	4.76	3.70	2.39	0.73	-1.47	-4.60	-9.55	-14.65	-8.45	-3.25	0.24	2.56	4.26	6.72
8.24	9.58	10.78	11.87	12.87	13.79	14.65	15.45	16.19	16.90	17.55	18.17	18.78	19.35	19.89	20.39
20.88	21.34	21.78	22.20	22.60	22.98	23.35	23.69	24.03	24.34	24.64	24.93	25.20	25.46	25.70	25.93
5.99	6.58	7.04	7.47	7.88	8.26	8.62	8.96	9.29	9.60	9.90	10.18	10.46	10.72	10.98	11.22
11.46	11.69	11.90	12.12	12.32	12.52	12.70	12.89	13.06	13.23	13.38	13.54	13.68	13.82	13.94	14.07
14.18	14.29	14.38	14.47	14.55	14.62	14.68	14.73	14.78	14.81	14.84	14.85	14.85	14.85	14.83	14.81
14.77	14.72	14.66	14.59	14.51	14.41	14.31	14.19	14.06	13.91	13.75	13.59	13.40	13.20	12.99	12.76
12.52	12.26	11.99	11.70	11.40	11.09	10.76	10.41	10.06	9.69	9.31	8.92	8.52	8.12	7.72	7.32
6.93	6.56	6.21	5.90	5.62	5.40	5.25	5.16	5.16	5.24	5.40	5.64	5.95	6.33	6.77	7.26
7.78	8.34	8.92	9.51	10.12	10.73	11.34	11.96	12.57	13.18	13.78	14.37	14.95	15.52	16.08	16.63
17.16	17.69	18.20	18.69	19.18	19.64	20.10	20.54	20.97	21.39	21.79	22.18	22.55	22.91	23.26	23.59
5.65	6.76	6.89	7.05	7.24	7.47	7.74	8.05	8.40	8.79	9.21	9.65	10.11	10.59	11.06	11.54
12.02	12.49	12.96	13.41	13.86	14.29	14.71	15.11	15.51	15.88	16.25	16.59	16.93	17.25	17.56	17.85
18.13	18.40	18.65	18.89	19.12	19.34	19.55	19.74	19.92	20.10	20.26	20.40	20.54	20.67	20.79	20.90
20.99	21.08	21.16	21.22	21.28	21.33	21.37	21.40	21.42	21.43	21.43	21.43	21.41	21.39	21.36	21.33
21.28	21.23	21.17	21.10	21.03	20.95	20.87	20.78	20.68	20.58	20.48	20.37	20.25	20.14	20.02	19.90
19.77	19.65	19.53	19.40	19.28	19.15	19.03	18.91	18.79	18.68	18.57	18.46	18.35	18.25	18.16	18.07
17.94	17.90	17.83	17.76	17.70	17.64	17.58	17.53	17.49	17.45	17.42	17.39	17.36	17.33	17.31	17.29
17.28	17.27	17.26	17.25	17.25	17.25	17.25	17.25	17.26	17.27	17.28	17.30	17.32	17.35	17.39	17.42
1.54	2.31	3.09	3.87	4.64	5.40	6.15	6.89	7.60	8.29	8.97	9.62	10.24	10.84	11.42	11.96
12.51	13.02	13.51	13.97	14.42	14.85	15.25	15.64	16.01	16.36	16.70	17.02	17.33	17.61	17.89	18.15

TIME(SEC)	FUND.FREQ.(HZ)	DB
0.031	114.	-12.46
0.061	75.	-12.78
0.092	138.	10.14
0.123	119.	14.85
0.154	133.	21.43
0.184	127.	20.69
0.215	128.	18.92
0.246	134.	17.98
0.276	136.	17.11
0.307	75.	-8.39
0.338	75.	-12.11
0.369	75.	-14.51
0.399	75.	-12.76
0.430	86.	-5.45
0.461	75.	-3.32
0.492	141.	3.66
0.522	77.	-6.99
0.553	75.	-11.60
0.584	102.	-11.29
0.614	119.	9.66
0.645	107.	6.71
0.676	75.	-10.55
0.707	100.	-13.36
0.737	79.	-9.70
0.768	75.	-12.36
0.799	95.	0.06
0.829	105.	10.50
0.860	94.	6.97
0.891	75.	-7.50
0.922	102.	-6.83
0.952	75.	-12.10
0.983	140.	-7.21
1.014	75.	0.03
1.044	75.	-9.00
1.075	75.	-5.44
1.106	75.	-1.98
1.137	135.	17.29
1.167	150.	2.21
1.198	75.	-17.92
1.229	130.	-7.49
1.260	75.	-17.42
1.290	75.	4.23
1.321	75.	1.13
1.352	120.	4.41
1.382	112.	-10.91
1.413	125.	-6.80
1.444	81.	-4.12
1.475	75.	-5.00
1.505	75.	-8.59
1.536	176.	4.92
1.567	128.	10.24
1.597	120.	5.08
1.628	107.	-8.13
1.659	107.	5.88
1.690	104.	9.56
1.720	100.	5.45
1.751	113.	5.06

1.782	98.	14.60
1.812	101.	14.22
1.843	101.	14.58
1.874	118.	-6.04
1.905	75.	-17.45
1.935	100.	-9.78
1.966	113.	-7.72
1.997	75.	-16.29
2.028	75.	-21.62
2.058	123.	-11.40
2.089	104.	-0.46
2.120	105.	2.15
2.150	75.	-13.03
2.181	105.	-11.61
2.212	175.	4.92
2.243	165.	-3.36
2.273	111.	-14.08
2.304	75.	-0.59
2.335	108.	9.30
2.365	112.	12.22
2.396	111.	13.34
2.427	113.	13.49
2.458	106.	14.31
2.488	104.	15.96
2.519	102.	18.26
2.550	95.	16.82
2.580	95.	15.69
2.611	91.	13.93
2.642	95.	11.72
2.673	88.	11.12
2.703	75.	-2.88
2.734	75.	-16.96
2.765	75.	-21.29
2.796	75.	-23.67
2.826	75.	-20.13
2.857	114.	-13.03
2.888	130.	-13.06
2.918	116.	-13.97

parameters used in the program are as follows:

NN=N	Number of data points in the data sequence
M	Number of Chirp Z-Transform Points
L	The smallest integer power of 2, which is ($M \geq N-1$)
LOG2N	Logarithm of N to the base 2
LOG2M	Logarithm of M to the base 2
LOG2L	Logarithm of L to the base 2
FL	Lower frequency specified in Hz.
FU	Higher frequency specified in Hz.
DF	Frequency interval which is related to the specified resolution in Hz.
FS	Sampling Frequency in Hz.
NB	The number of blocks to be processed

After reading in the above parameters, statements 17, 18, and 19 generate the Hanning Window, using the equation given below.

$$W(k) = 0.5(1-\text{Cos}\,(2\pi(k-1)/NN), \qquad k=1,2,\ldots\ldots NN \qquad (3.2.1)$$

where NN is the total number of points as defined above.

Then the program calls Subroutine WPOWER which returns the complex multipliers for the Chirp Z-Transform and the FFT.

Then the program computes the parameters RWPH and RAPH using the following relationships:

$$RWPH = \phi_0 = -DF/FS$$

$$RAPH = \theta_0 = FL/FS$$

Next the program causes the data to be read from the tape, and assign 256 data points per block. The appropriate number for the block per sentence is determined from the DUMPOUT program output. The program then calls for Subroutine Chirp Z which computes the short-time Fourier Spectrum. The bandwidth of such spectra is 128Hz., beginning at 75 Hz. and terminating at 203 Hz., the resulting spectal resolution is 1Hz. Thus the short-time spectrum of each block consists of 128 points. These values are printed out according to the format specified towards the end of the program.

The main objective of the program is to isolate that portion of the short-time spectrum of each block where the maximun power is present. This is usually achieved by examining the output of the program which consists of the printout of 128 points. However, the program does pickout the frequency at which the first peak occurs in each of the short-time spectra. This information may be sufficient in some cases.

CHAPTER IV

THE CEPSTRUM PROGRAM

4.1 Rationale for Cepstrum Analysis

For the purposes of discussion, consider a simplified block diagram for the production of vowels, as shown in Fig. 4.1.1. The direct current air flow from the lungs is converted into alternating current accoustic energy by the vocal tract. The output speech signal f(t) appearing at the lips is the convolution of the excitation function x(t), which represents the airflow at the vocal cords, and h(t); that is,

$$f(t)= \int_{-\infty}^{t} x(t).h(t-\tau)d\tau \qquad 4.1.1$$

or $$f(t)=x(t)*h(t) \qquad 4.1.2$$

Taking the Fourier Transform (designated by capital letters) on both sides of Eq. 4.1.1, the following equation is obtained;

$$F(w) =X(w).H(w) \qquad 4.1.3$$

Thus, the spectrum of the speech signal is the product of the excitation spectrum X(w) and the transfer function of the vocal tract H(w). The power spectrum $|F(w)|^2$ of the f(t) equals the product of the power spectra of x(t) and h(t); i.e., $|F(w)|^2= |X(w)|^2 . |H(w)|^2$. Thus the autocorrelation function is the convolution of the inverse Fourier Transform of X(w) and H(w). Usually, the transfer function H(w) is such that the resulting convolution gives multiple peaks. However, the log-magnitude spectrum of a voiced speech is composed of two components: (i) a rapidly varying periodic component associated with vocal cord excitation, and (ii) a slowly varying component associated with the formant frequencies. The slowly varying component has to be separated to estimate the parameter values of the format frequencies. The technique used for this purpose is linear filtering. However, line filtering has been achieved here is through "cepstrum". The cepstrum is defined as inverse Fourier

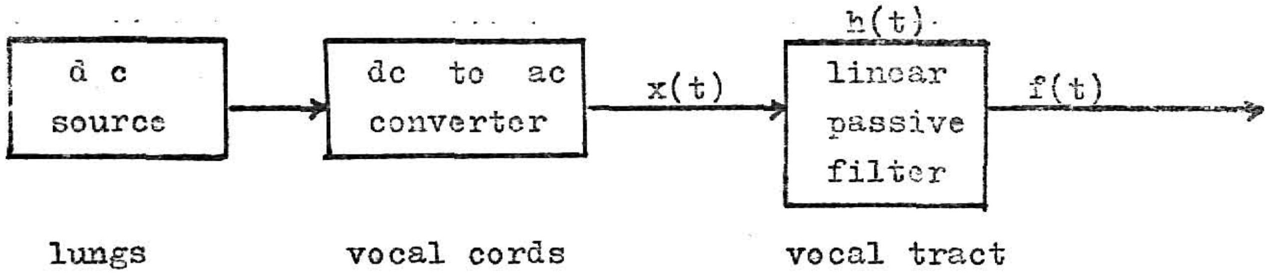


Fig. 4.1.1

BLOCK DIAGRAM REPRESENTING THE PRODUCTION OF VOWELS.

CEPSTRUM PROGRAM

```
//XPRS7627 JOB (XXXXXXXX,,3),RAINBOLT,TIME=(1,45)                    JOB 503
***TAPE9         1
//   EXEC    FORTGCLG
//FORT.SYSIN  DD  *

KSU003I PARTITION SIZE - 256K, MAXIMUM CORE USED -  92K,  TIME =  21.44.07
KSU004I I/O COUNT -- DISK =          3,  TAPE =         0
KSU001I STEP   1 FORT       EXECUTION TIME =          2.29 SEC    RETURN CODE =     0

KSU003I PARTITION SIZE - 256K, MAXIMUM CORE USED - 156K,  TIME =  21.44.23
KSU004I I/O COUNT -- DISK =        113,  TAPE =         0
KSU001I STEP   2 LKED       EXECUTION TIME =          0.89 SEC    RETURN CODE =     0

//GO.SYSIN  DD  *
//GO.FT08F001  DD  DSN=SENT05,DISP=OLD,LABEL=(5,NL),
//    DCB=(RECFM=FB,LRECL=10,BLKSIZE=10000)

KSU003I PARTITION SIZE - 256K, MAXIMUM CORE USED -  60K,  TIME =  21.47.41
KSU004I I/O COUNT -- DISK =          0,  TAPE =        26
KSU001I STEP   3 GO         EXECUTION TIME =         35.00 SEC    RETURN CODE =     0

KSU002I JOB XPRS7627 EXECUTION TIME =         38.18 SEC


 HASP-II V3.1 JOB STATISTICS --    124 CARDS READ --    1,590 LINES PRINTED --     0 CARDS PUNCHED --    COPY   1 OF   1
```

```
FORTRAN IV G LEVEL  21            MAIN            DATE = 74143          21/44/00            PAGE 0001

          DIMENSION XX( 1200),RX(256)
          DIMENSION W(256)
          COMPLEX X(256),CMPLX,MULT(128)
          READ 5,N,LOG2N,NB
      5   FORMAT (3I4)
          LOG2N1=LOG2N-1
          N2=N/2
          N21=N2+1
          TWOPI=6.283185
          DO 15 I=1,N
          ARG=TWOPI*FLOAT(I-1)/FLOAT(N)
     15   W(I)=0.5*(1.-COS(ARG))
          CALL WPOWER(MULT,N2)
          DO 100 II=1,NB
          READ (8,10)(XX(I),I=1,N)
     10   FORMAT (F10.7)
          DO 20 I=1,N
          WX=W(I)*XX(I)
     20   X(I)=CMPLX(WX,0.)
          CALL FFT(X,LOG2N,LOG2N,N,N,MULT)
          DO 25 I=1,N
          AMAG=CABS(X(I))
          ALMAG=20.*ALOG10(AMAG)+4.
     25   X(I)=CMPLX(ALMAG,0.)
          CALL IFFT(X,LOG2N,LOG2N1,N,N2,MULT)
          DO 30 I=1,N2
     30   RX(I)=REAL(X(I))
          PRINT 35,(RX(I),I=33,N2)
     35   FORMAT('0',10X,'CEPSTRUM:',//,(' ',8F10.5))
    100   CONTINUE
          STOP
          END
```

```
      SUBROUTINE FFT(X,M,L,N,L2,MULT)
      COMPLEX MULT(128)
      COMPLEX X(256),W,T,U,CMPLX,CONJG
      DATA PI/3.1415927/
      K=M-L
      LMX=N
      IND=1
      DO 50 LO=1,M
      LIX=LMX
      LMX=LMX/2
      IP=0
      IF (LO.GT.K) GO TO 30
      DO 20 LM=1,L2
      DO 20 LI=LIX,N,LIX
      J1=LI-LIX+LM
      J2=J1+LMX
   20 X(J2)=MULT(J1)*X(J1)
      GO TO 45
   30 DO 40 LM=1,LMX
      IF (LM.NE.1) GO TO 32
      W=(1.,0.)
      GO TO 34
   32 IP=IP+IND
      W=MULT(IP)
   34 CONTINUE
      DO 40 LI=LIX,N,LIX
      J1=LI-LIX+LM
      J2=J1+LMX
      T=X(J1)-X(J2)
      X(J1)=X(J1)+X(J2)
   40 X(J2)=W*T
   45 IND=IND+IND
   50 CONTINUE
      RETURN
      ENTRY IFFT(X,M,L,N,N1,MULT)
      LE=1
      IND=N
      DO 65 LO=1,M
      IND=IND/2
      LE1=LE
      LE=LE+LE
      IPP=1
      IF (LO.GT.L) GO TO 33
      DO 11 J=1,LE1
      IF (J.NE.1) GO TO 7
      U=(1.,0.)
      GO TO 9
    7 IPP=IPP+IND
      U=CONJG(MULT(IPP))
    9 CONTINUE
      DO 9 I=J,N,LE
      IP=I+LE1
      T=X(IP)*U
      X(IP)=X(I)-T
    9 X(I)=X(I)+T
   11 CONTINUE
      GO TO 65
   33 DO 12 J=1,N1
```

```
      IF (J.NE.1) GO TO 14
      U=(1.,0.)
      GO TO 15
   14 IPP=IPP+IND
      U=CONJG(MULT(IPP))
   15 CONTINUE
      DO 10 I=J,N,LE
      IP=I+LE1
   10 X(I)=X(I)+X(IP)*U
   12 CONTINUE
   65 CONTINUE
      DO 66 I=1,N
   66 X(I)=X(I)/N
      RETURN
      END
```

```
FORTRAN IV G LEVEL  21            WPOWER           DATE = 74143          21/44/00          PAGE 0001

      SUBROUTINE WPOWER(W,N)
      COMPLEX W(128),CMPLX
      W(1)=(1.,0.)
      ARG1=3.141592/FLOAT(N)
      DO 10 I=2,N
      ARG2=(I-1)*ARG1
   10 W(I)=CMPLX(COS(ARG2),-SIN(ARG2))
      RETURN
      END
```

CEPSTRUM:

-0.03799	-0.16706	0.67260	-0.00470	0.10675	0.20005	0.07720	0.04430
0.51709	0.14070	-0.23127	0.33059	0.27356	-0.74312	0.14170	-0.09577
-0.53656	0.50661	-0.54128	-0.02190	-0.04648	-0.05195	-0.25649	[illegible]
-0.05785	-0.65524	0.03991	0.10959	0.03000	0.04886	-0.15135	0.52750
0.72333	0.16205	-0.01624	0.43813	0.39354	-0.03095	-0.14314	0.07755
-0.43434	0.72391	-0.61147	-0.21784	-0.45527	-0.04960	-0.12761	0.28324
-0.05589	0.29900	-0.19183	-0.17124	0.04731	0.13766	-0.14551	[illegible]
-0.20479	0.15536	0.31969	0.58273	-0.12524	0.04499	-0.22915	0.55543
0.09476	-0.25748	-0.17102	0.13675	0.17740	-0.53741	-0.47162	0.07566
-0.22123	0.17929	0.03615	-0.35069	-0.00437	0.14941	-0.39139	0.10575
-0.15332	-0.12358	0.07126	0.27107	0.41943	0.03361	0.31640	0.07579
0.10437	0.17545	0.19613	-0.06919	-0.13342	-0.33236	0.18270	-0.21126

CEPSTRUM:

0.03662	0.29246	-0.15204	0.05255	0.05572	-0.14171	-0.21115	0.20614
-0.01993	0.43198	-0.10311	0.23265	0.08590	-0.50821	0.14940	[illegible]
-0.29976	0.21120	0.18163	0.12683	-0.01778	0.00473	-0.25033	0.15237
-0.21400	0.06028	-0.02226	0.02816	0.03659	0.35625	-0.42932	0.16755
0.17226	0.27962	-0.20972	-0.21150	0.21712	-0.17015	-0.07453	[illegible]
0.07417	-0.13810	0.29355	-0.51331	0.43279	-0.46063	0.22937	[illegible]
0.04556	-0.09851	0.16677	-0.16950	0.12792	-0.34743	0.09467	[illegible]
-0.02934	0.00393	-0.27485	0.08520	0.07627	0.01965	-0.00363	-0.14291
-0.02377	0.14857	0.01055	-0.21682	-0.05564	-0.00171	-0.22265	0.30672
-0.30313	0.27152	0.04135	-0.04060	0.16102	-0.23477	0.13301	-0.13557
-0.06312	-0.03061	-0.07131	0.02745	0.22141	-0.19914	0.17730	-0.13422
0.01302	0.03437	-0.01126	-0.19638	0.14243	-0.17811	0.05171	0.11438

CEPSTRUM:

0.02669	0.17426	0.31694	0.31779	0.05398	0.06228	0.33425	-0.01137
0.30145	-0.06276	-0.20503	0.14150	0.42115	0.82489	-0.24390	[illegible]
-0.09399	-0.39326	-0.55471	-0.45030	0.06306	-0.40132	0.22963	-0.05572
0.05234	-0.53020	0.00146	-0.41279	0.12923	-0.17371	-0.30091	0.28155
-0.24560	0.09293	-0.14078	0.12405	-0.13351	0.31453	0.16196	0.45642
-0.17249	0.51790	1.68056	0.06645	0.07595	-0.18713	0.26559	0.17745
0.10577	0.10643	0.11729	0.06994	-0.10281	-0.15918	0.15815	0.11426
-0.01428	-0.20064	0.03309	-0.14690	0.12385	-0.17864	0.20541	-0.15083
0.02509	0.04536	0.27461	-0.34996	-0.18060	-0.20398	-0.01164	-0.33310
-0.07815	-0.22727	-0.14538	-0.19999	-0.52477	-0.09835	-0.31654	0.22276
-0.31690	0.20898	-0.39051	0.13648	-0.03651	0.02379	-0.03562	-0.36431
0.24996	-0.09820	0.09871	0.10864	0.32342	0.07878	0.22405	0.12634

CEPSTRUM:

-0.04134	0.30922	-0.12459	0.26398	0.20533	0.14930	-0.09730	-0.16334
0.15152	0.30705	0.44563	0.77555	0.40568	0.50174	0.44256	0.31052
-0.17034	-0.28609	-0.00150	0.06469	0.27939	-0.42664	-0.39543	-0.13755
-0.37556	-0.05355	-0.32313	0.13700	-0.26169	0.11987	-0.31738	0.37154
-0.29534	0.24947	-0.46446	0.47725	0.17050	0.24753	0.02787	[illegible]
0.55237	0.09075	0.69619	-0.43249	0.32755	-0.44139	0.17332	[illegible]
0.20769	0.04297	-0.05437	-0.04813	0.17587	-0.01056	-0.25593	-0.16773
-0.22577	0.07097	-0.07915	-0.22568	0.11477	-0.01070	0.21926	0.00213
0.04144	0.07828	0.34805	0.26153	0.10936	-0.17105	-0.07759	0.13593
-0.24720	-0.00092	-0.16157	-0.04222	0.17607	-0.20329	0.40484	-0.57846
-0.05429	-0.36569	-1.05114	-0.86351	0.10067	-0.30920	-0.33979	0.19637

Transform of the log-magnitude spectrum.

4.2 The Cepstrum Program

This program provides an easy means of differentiating voiced and unvoiced portions of speech from its unvoiced portions. In the initial steps, the program defines and reads the parameters used in the program. These are as follows:

N	Number of points in each block
LOG2N	Logarithm of N to the base 2
NB	Number of blocks

The program then generates the Hanning window by the statements 10 through 12. Then the program calls the subroutine WPOWER which returns the complex multipliers for the FFT subroutine.

The statements 15 and 16 cause the program to read in the data from the tape and the data is then passed through the Hanning window in order to minimize the undesiderable effects introduced as a consequence of the Fourier Analysis of a finite length data sequence.

The subroutine FFT is then called which returns 256 values of the power spectrum. Next the decible values of the power spectrum are calculated. 4db has been added as a correction factor to retain the r.m.s. value of the window Inverse Fourier Transform as unity. Again, the IFFT is taken calling the subroutine FFT (ENTRY POINT : IFFT) to get back into time domain. The result, called the cepstrum of the input data, is an additive combination of the cepstra of the excitation and vocal tract components. Thus, the effect of the FFT, log-magnitude and IFFT processing is to transform convolution into addition. If a segment of speech is voiced, then a prominent peak is obtained. However, if a block is unvoiced, then no prominent peak occurs. Thus the cepstrum serves as an excellent basis for estimating fundamental period of voiced speech and for determining whether a particular block is voiced or unvoiced.

The fundamental frequency is the reciprocal of the fundamental period determined from the cepstrum. This also serves as a rough check with the CZT program output, which gives an accurate estimate of (within$\pm$1Hz.) the fundamental frequencies in each block.

Thus in summary, the cepstrum program enables one to decide whether a given block of speech data consists of voiced or unvoiced speech. If it is voiced, then the corresponding output from the CZT program (Chapter III) yields an accurate estimate of the fundamental frequency. Again, if a block is unvoiced, then the corresponding output from the CZT program is ignored.

CHAPTER V

CONCLUSIONS

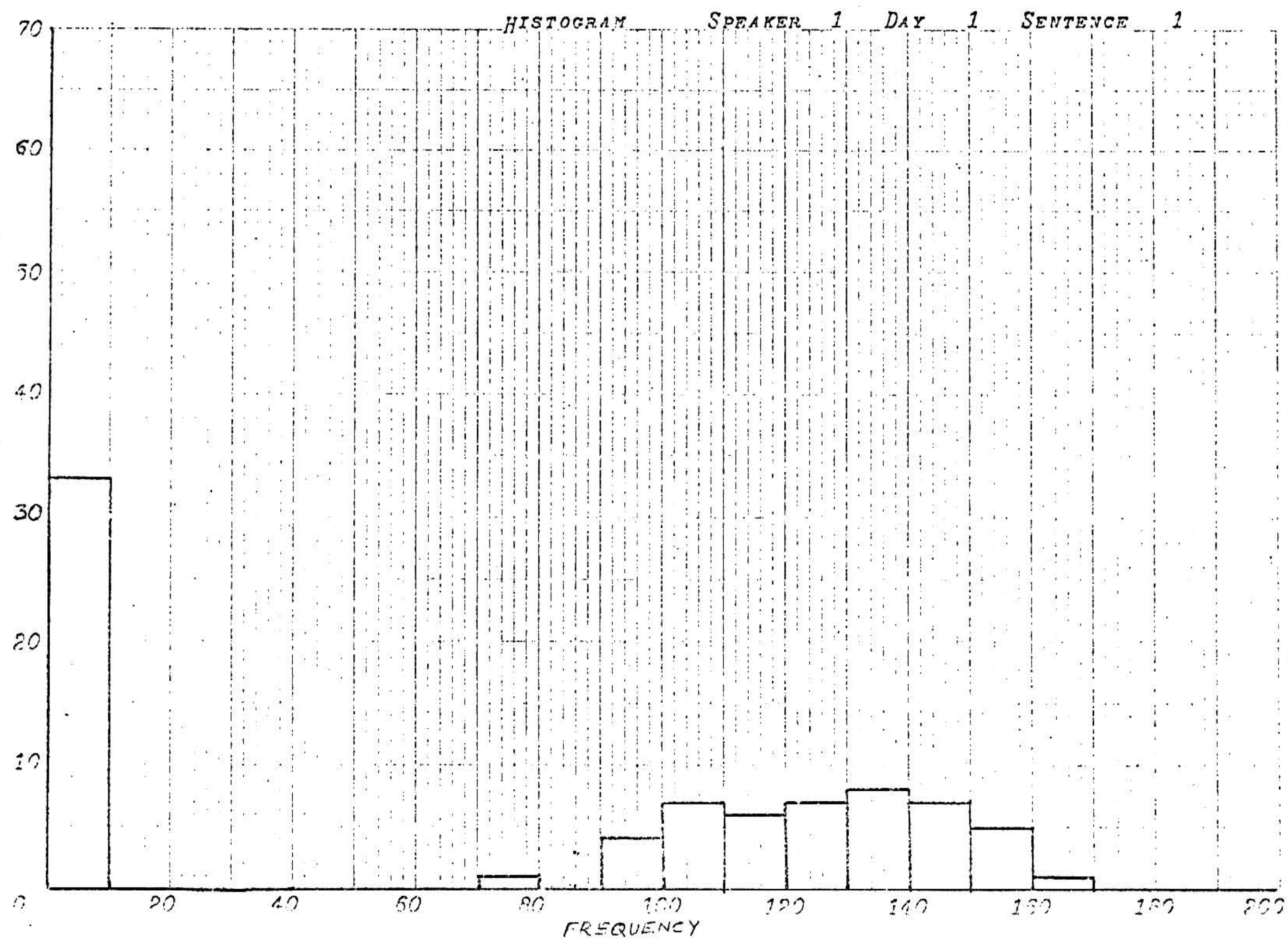
A parameter of deaf speech which has received considerable attention is the fundamental frequency of voiced speech (3). Attempts have been made to characterize fundamental frequency variation by means of short-term polynomial analysis techniques (3,4). From the discussion in the previous chapters, it is clear that the computer programs presented in this report can be used to obtain an accurate estimate of the fundamental frequency. The corresponding fundamental frequency variations can then be studied. For the purposes of illustration, we consider the distribution of fundamental frequencies of the three speakers mentioned in Chapter 1 and 2. Speaker 1 was moderately to severly hard of hearing and speaker 2 and 3 were profoundly deaf. However, speaker 3 had a slightly greater mean hearing loss than speaker 2.

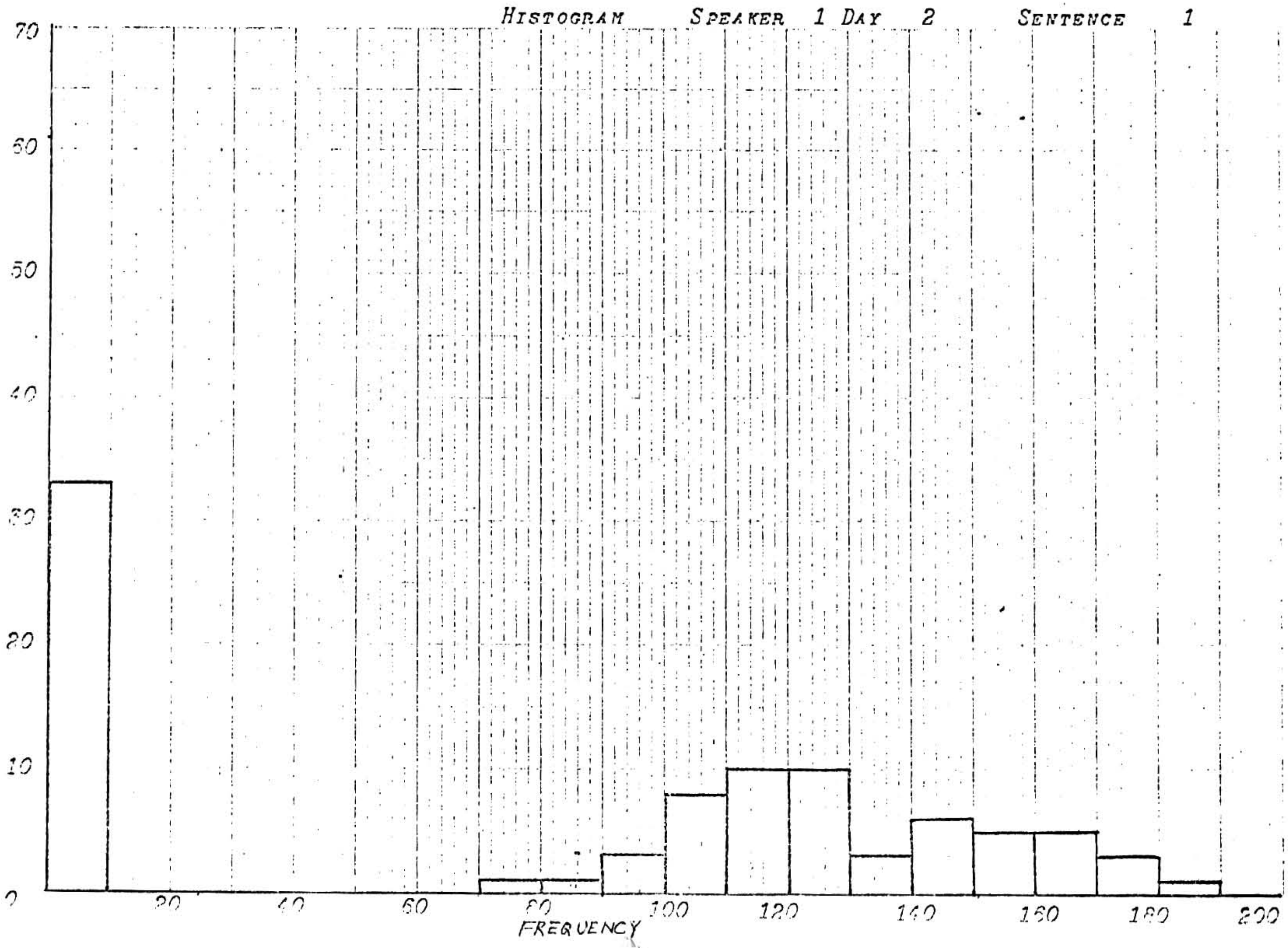
The pertinent data was processed and subsequently analyzed. The resulting histograms of the fundamental frequencies of all the speakers for each day are shown at the end of this chapter, along with averaged (i.e., averaged over the entire period) averaged histograms indicate that the fundamental frequencies of the first speaker are distributed between 88 and 145 Hz.; and those for the second and third speaker are distributed between 127 and 151 Hz., and 148 and 190Hz. respectively. The inference one draws from these figures is that the redominant fundamental frequency may be related to the degree of hearing loss sustains through the range of 250 to 1000Hz. This observation will be correlated with other information such as hearing test results and conventional (i.e., analog) speech spectrograms by a student of the Dept. of Speech and Hearing at Kansas State University.

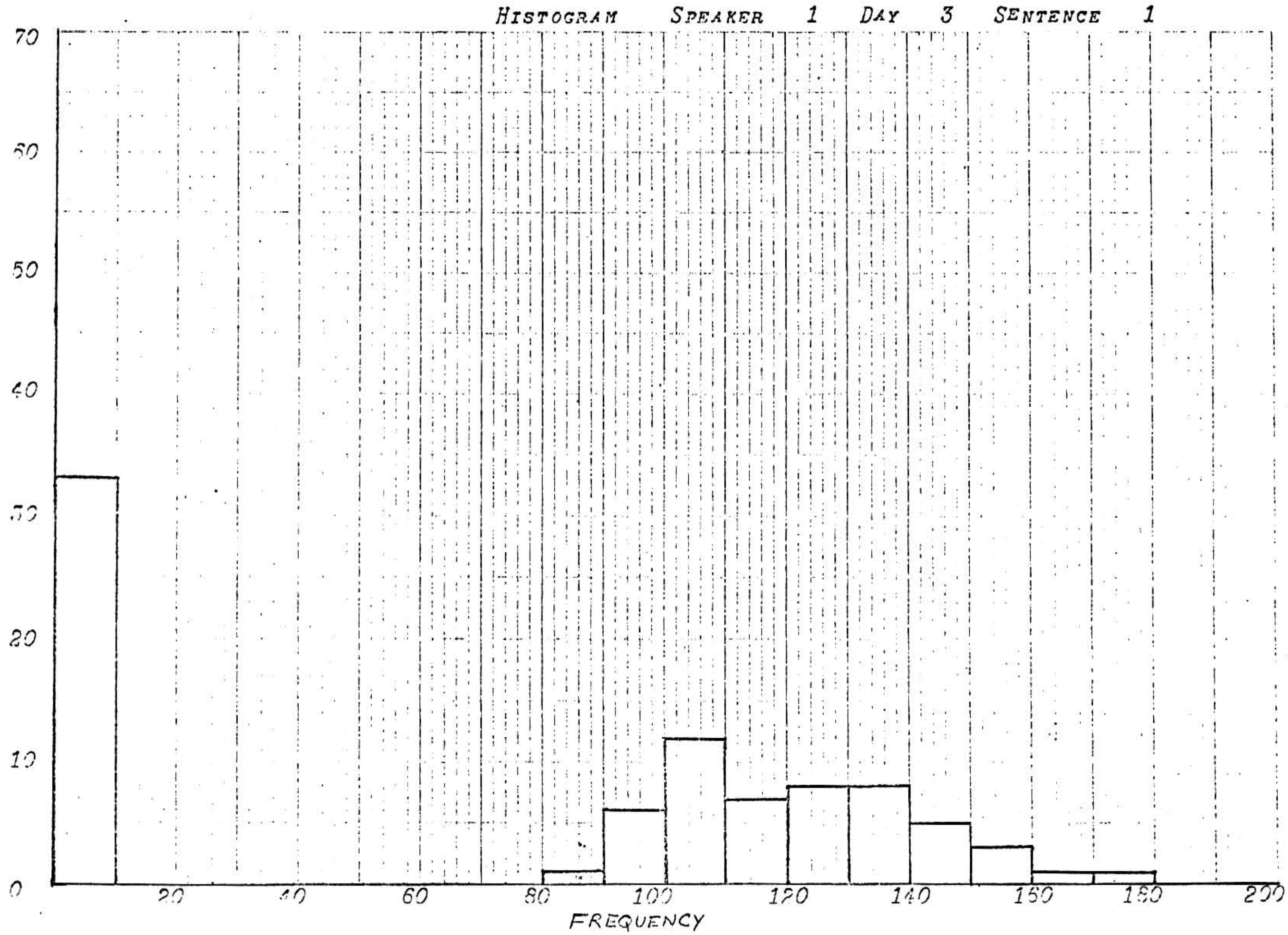
The above computer programs can also be used to obtain digital spectrograms. In this connection, the digital spectrograms associated with the deaf speech data have been generated. These are in the process of being evaluated with corresponding conventional sound spectrographs.[1]

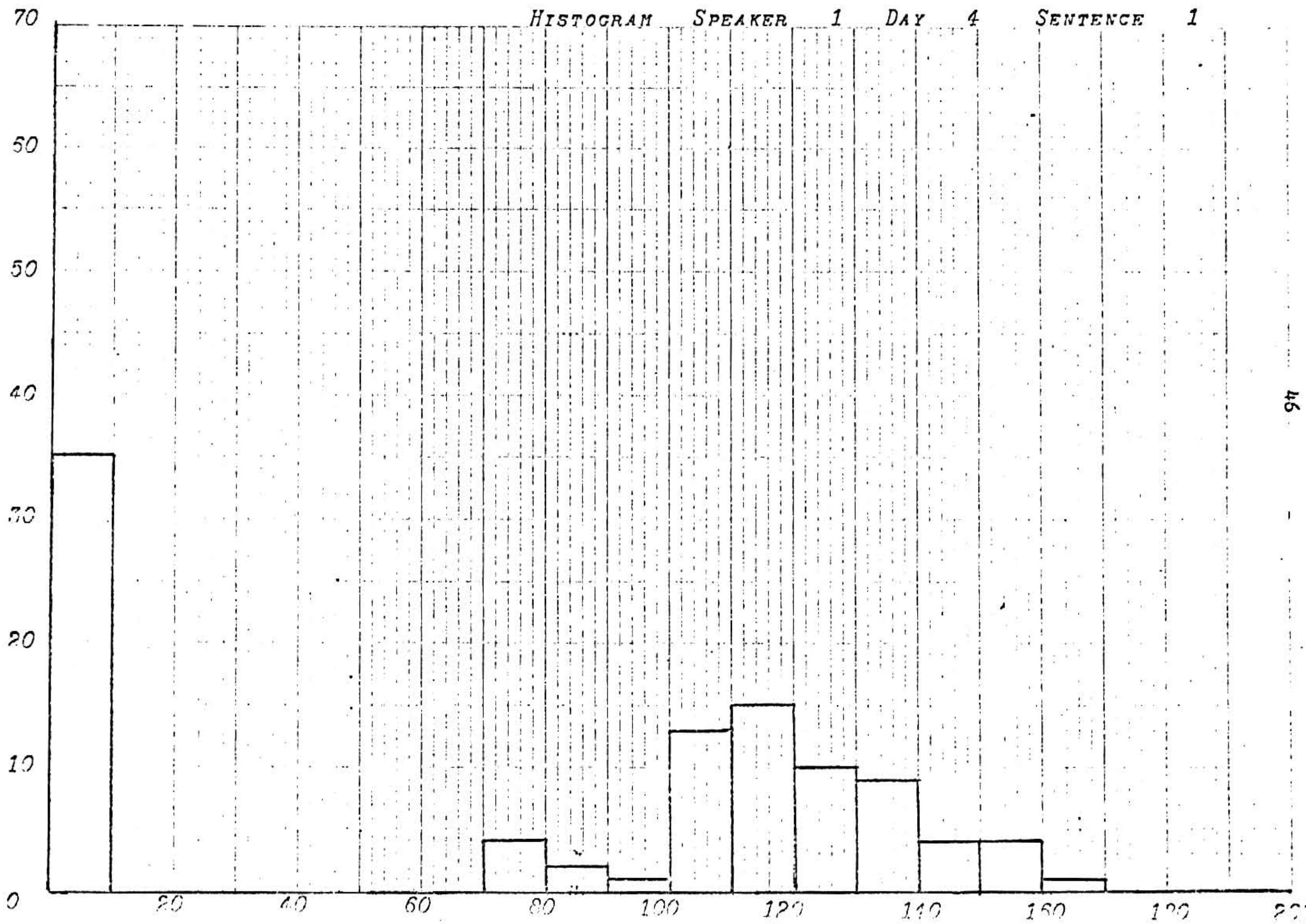
1-Work done by a graduate student in Dept. of Speech.

HISTOGRAMS

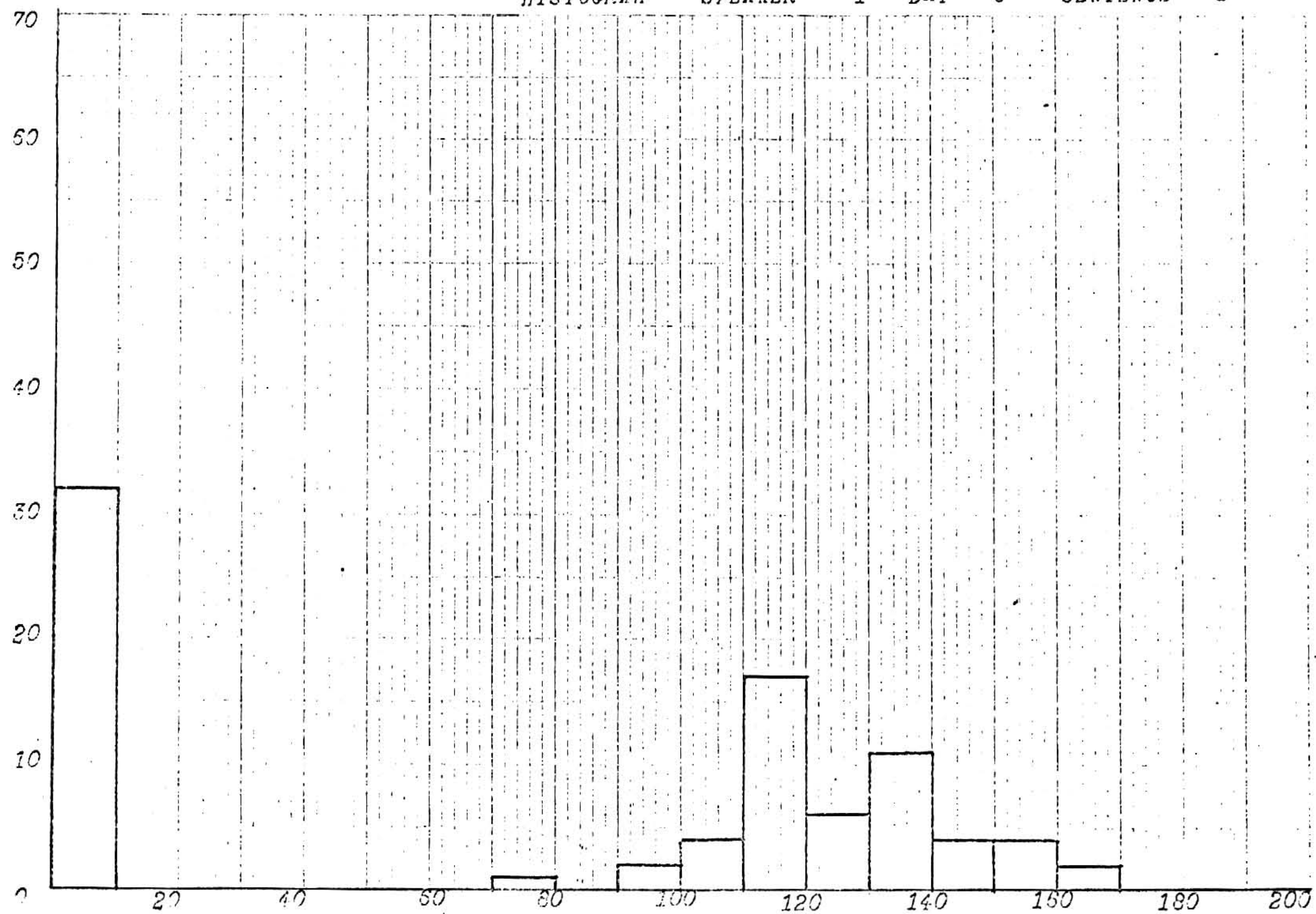

HISTOGRAM SPEAKER 1 DAY 1 SENTENCE 1
0
10
20
30
40
50
60
70
20
40
60
80
100
120
140
160
180
200
FREQUENCY

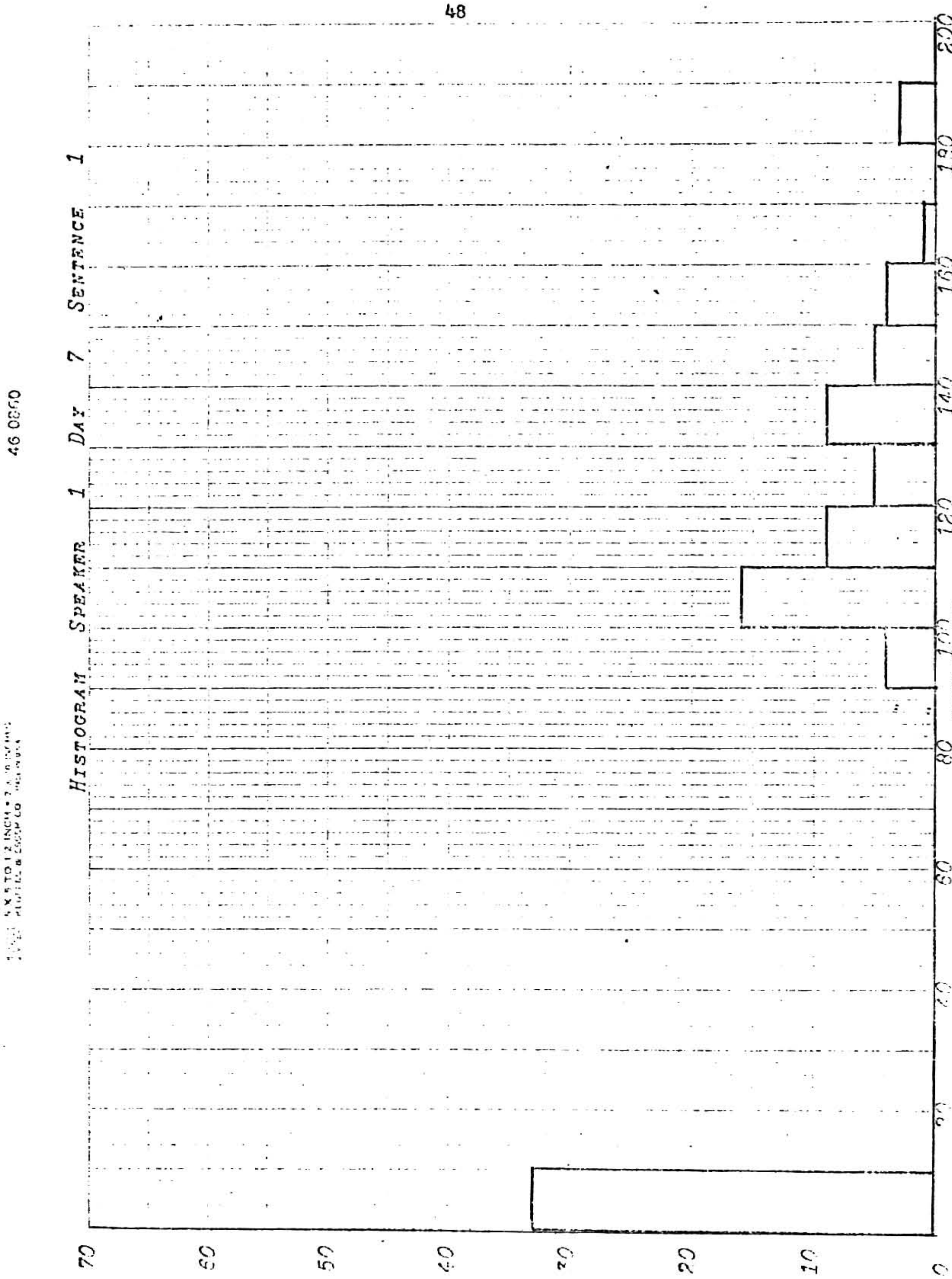

HISTOGRAM SPEAKER 1 DAY 2 SENTENCE 1
70
60
50
40
30
20
10
0
20
40
60
80
100
120
140
160
180
200
FREQUENCY


HISTOGRAM SPEAKER 1 DAY 3 SENTENCE 1
70
60
50
40
30
20
10
0
20
40
60
80
100
120
140
160
180
200
FREQUENCY

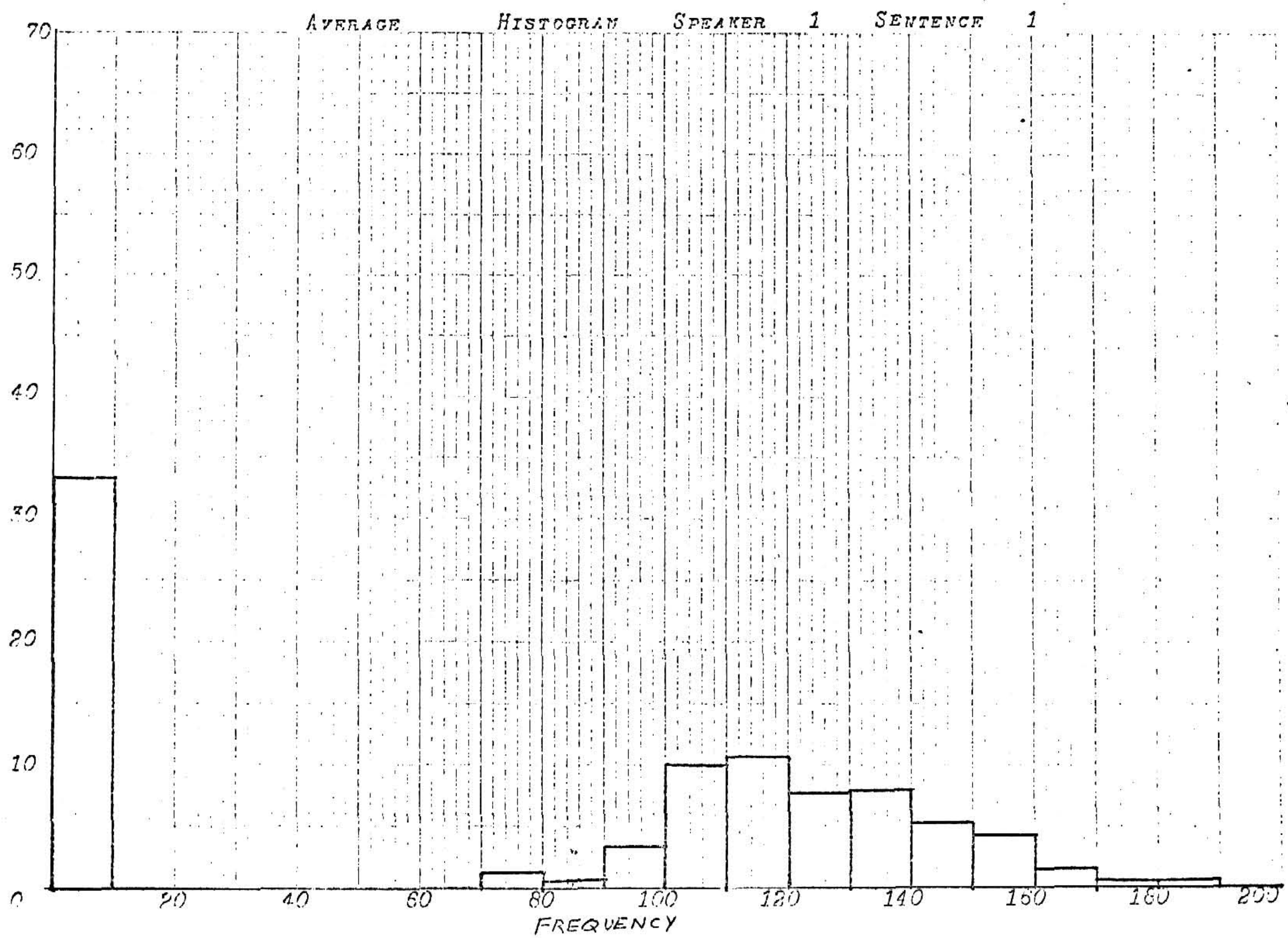

HISTOGRAM SPEAKER 1 DAY 4 SENTENCE 1
70
50
40
20
10
0
40
60
80
100

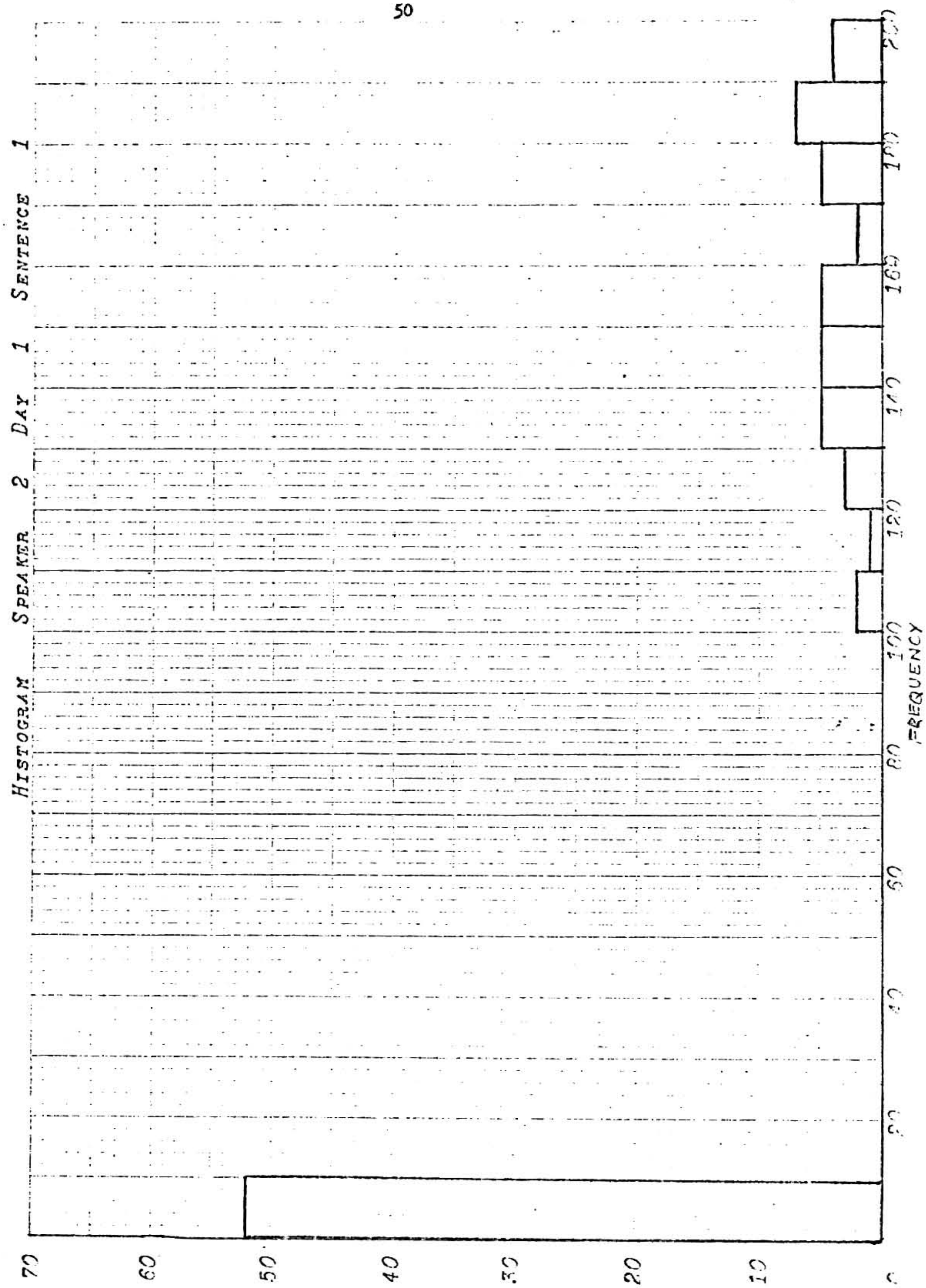
HISTOGRAM SPEAKER 1 DAY 6 SENTENCE 1




HISTOGRAM SPEAKER 1 DAY 7 SENTENCE 1
FREQUENCY
0
10
20
30
40
50
60
70
60
80
120
140
160
180
200
46 0860

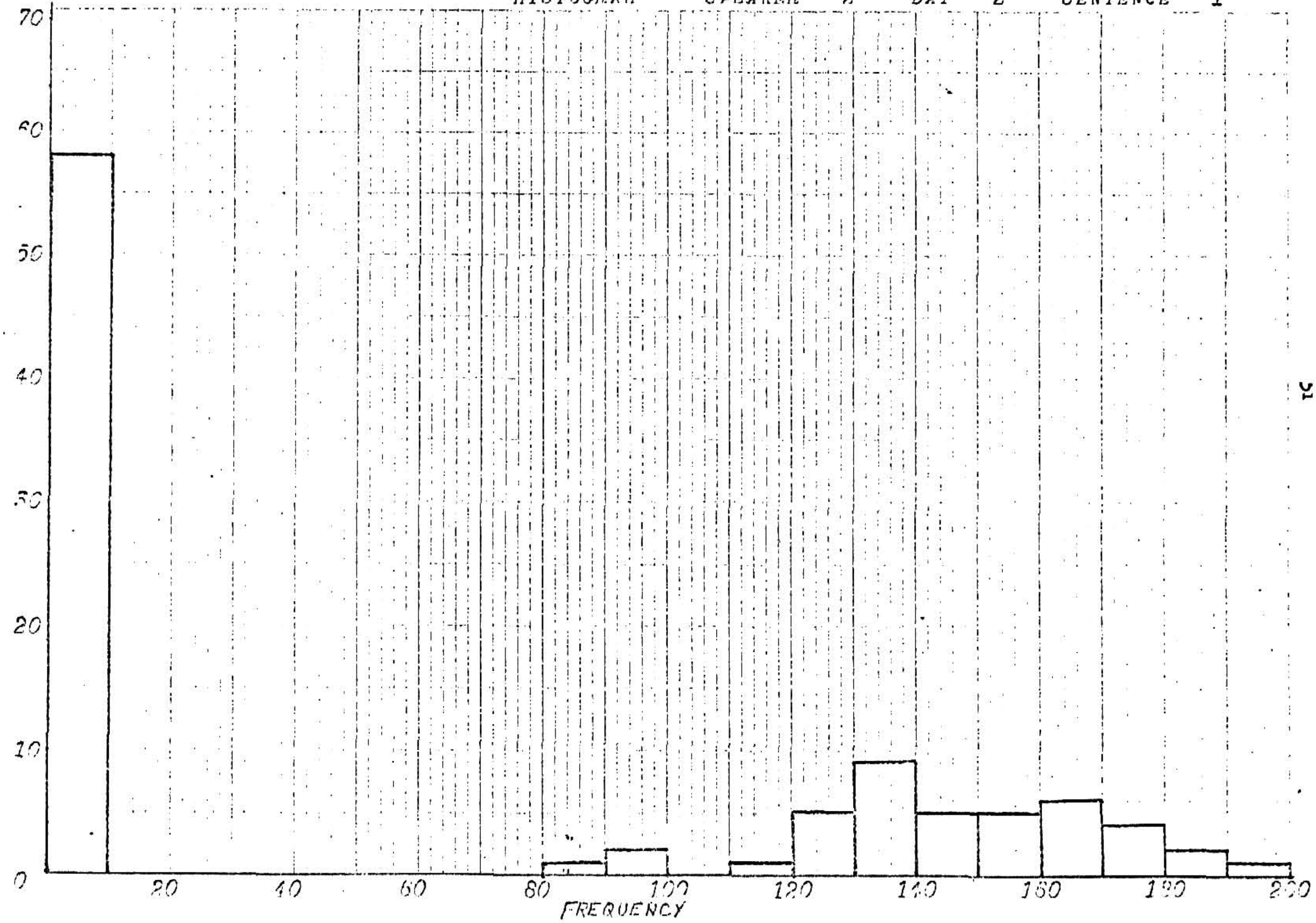
46 0360

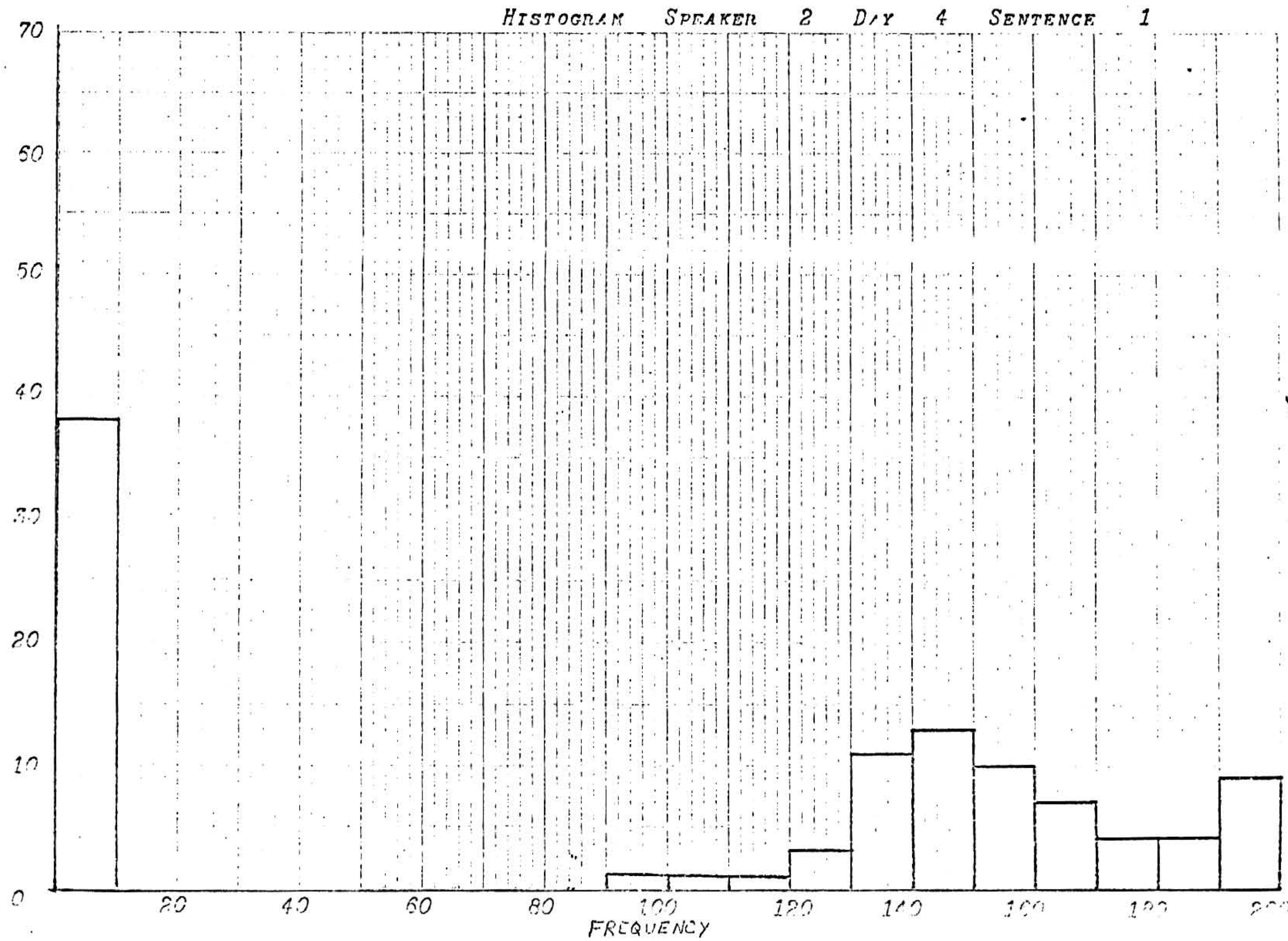

AVERAGE HISTOGRAM SPEAKER 1 SENTENCE 1
70
60
50
40
30
20
10
0
20
40
60
80
100
120
140
160
180
200
FREQUENCY

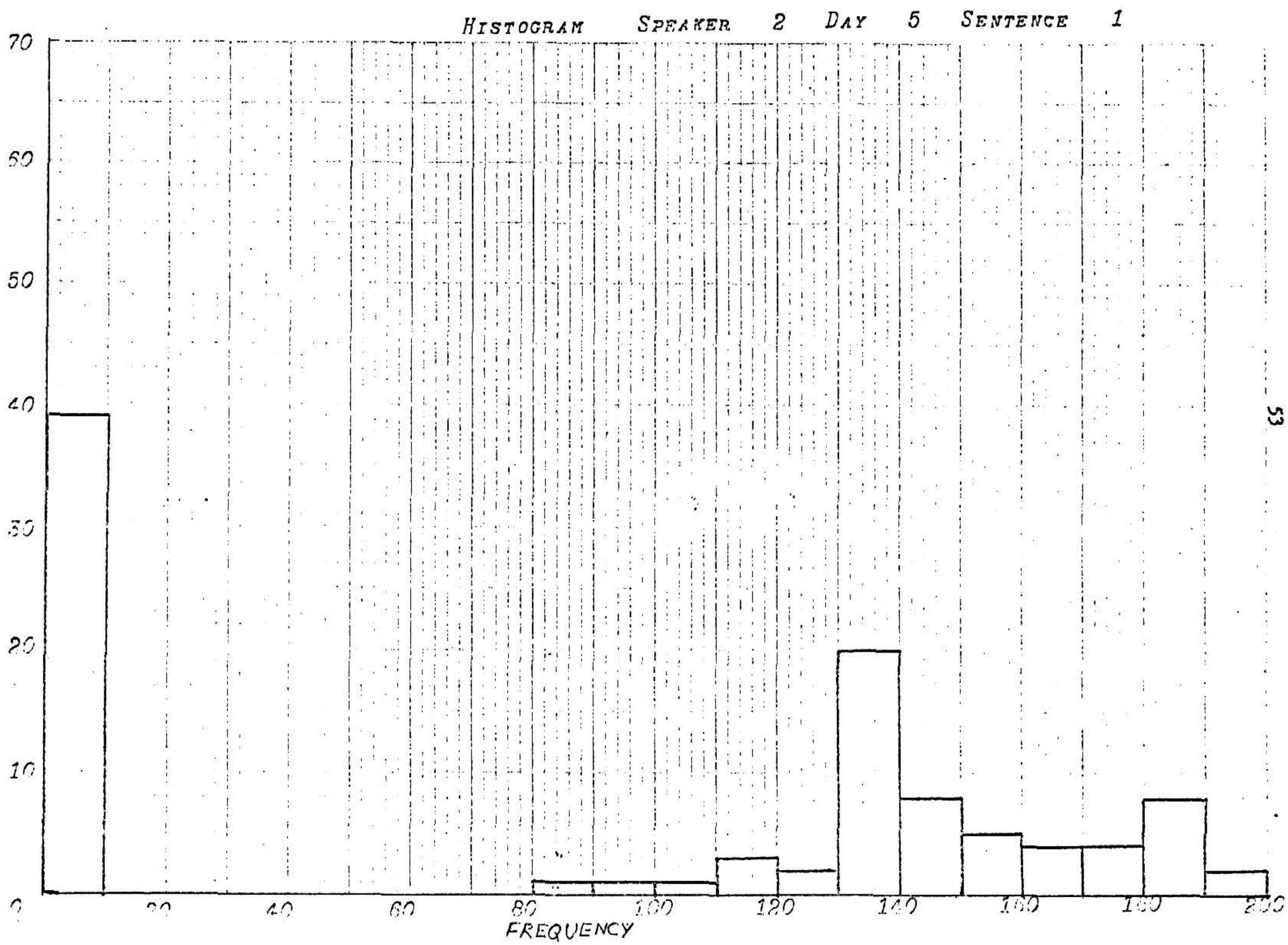

HISTOGRAM SPEAKER 2 DAY 1 SENTENCE 1
FREQUENCY
70
60
50
40
30
20
10
0
20
40
60
80
100
120
140
160
180
200

46 0360

HISTOGRAM SPEAKER 2 DAY 2 SENTENCE 1

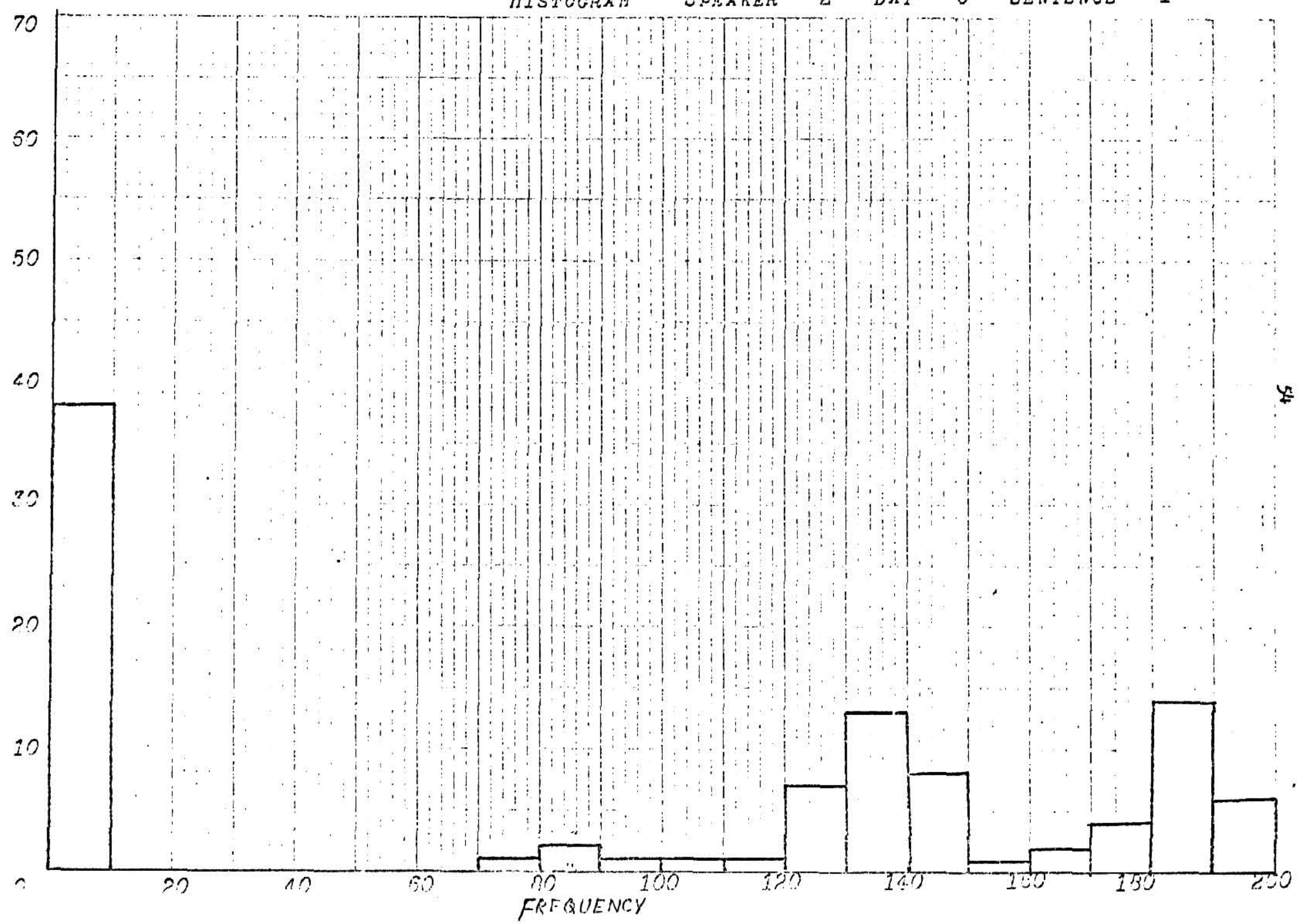


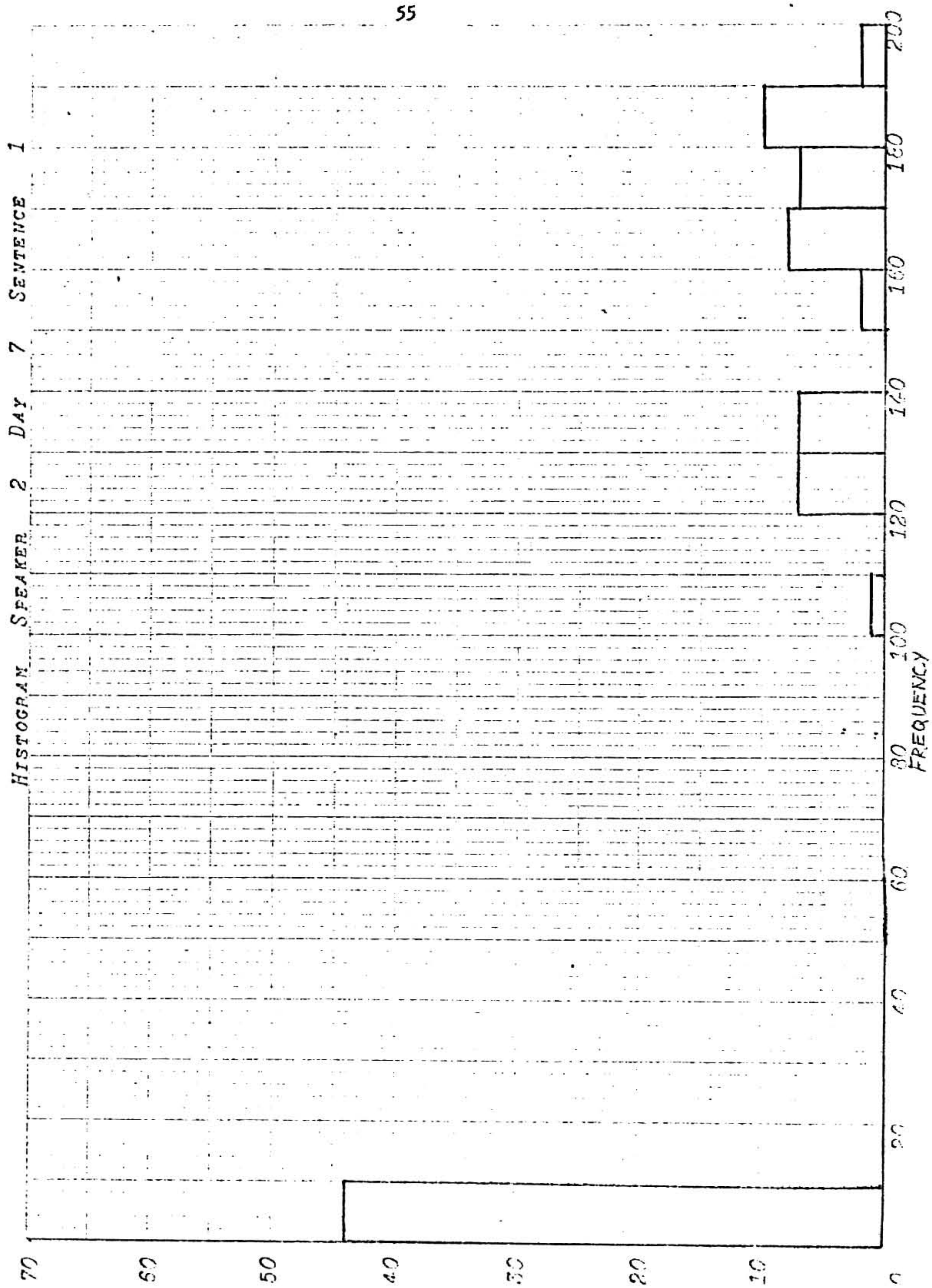

HISTOGRAM SPEAKER 2 DAY 4 SENTENCE 1
70
60
50
40
30
20
10
0
20
40
60
80
100
120
140
FREQUENCY


HISTOGRAM SPEAKER 2 DAY 5 SENTENCE 1
70
60
50
40
30
20
10
0
40
60
80
100
120
140
FREQUENCY

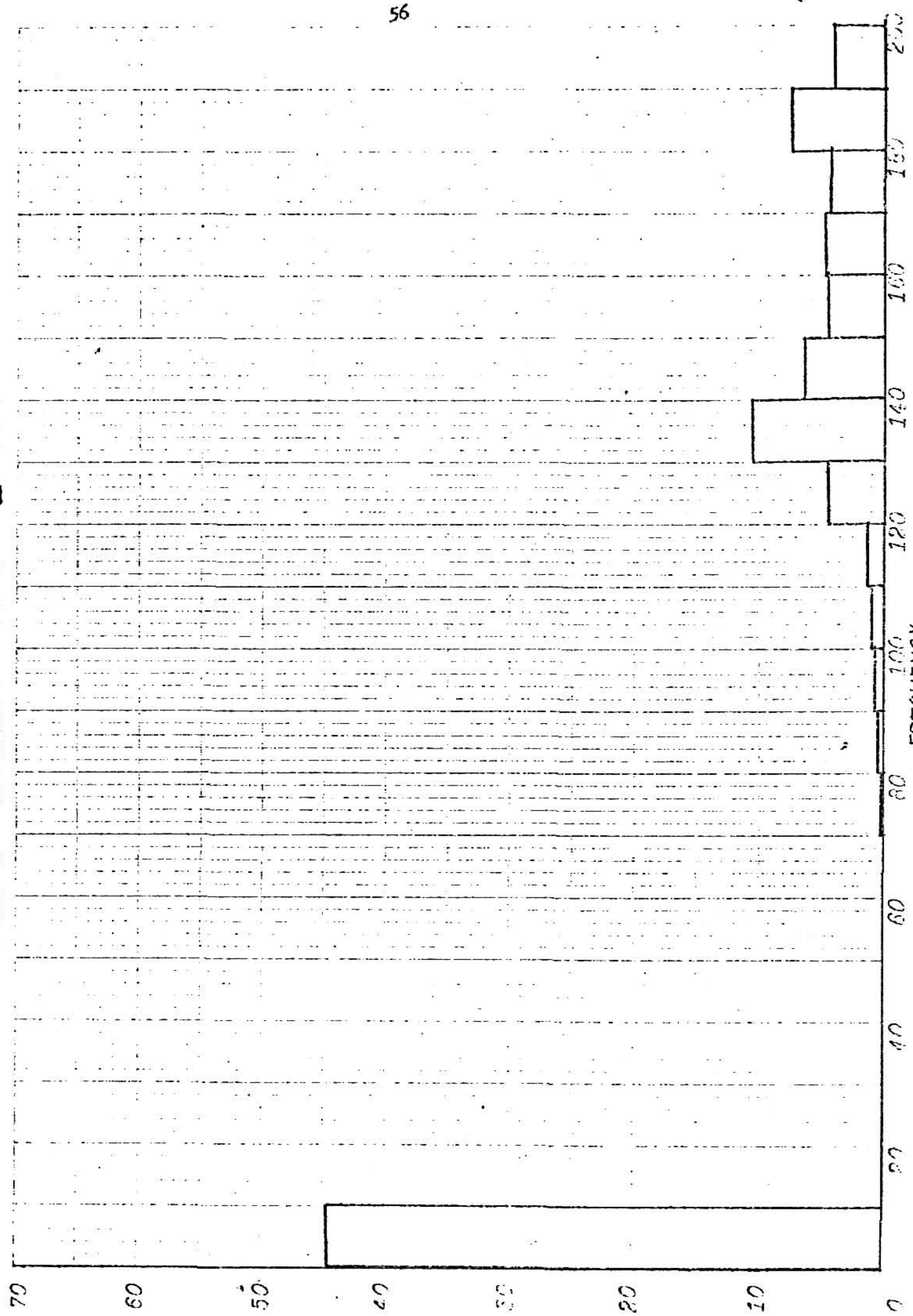
46 0860

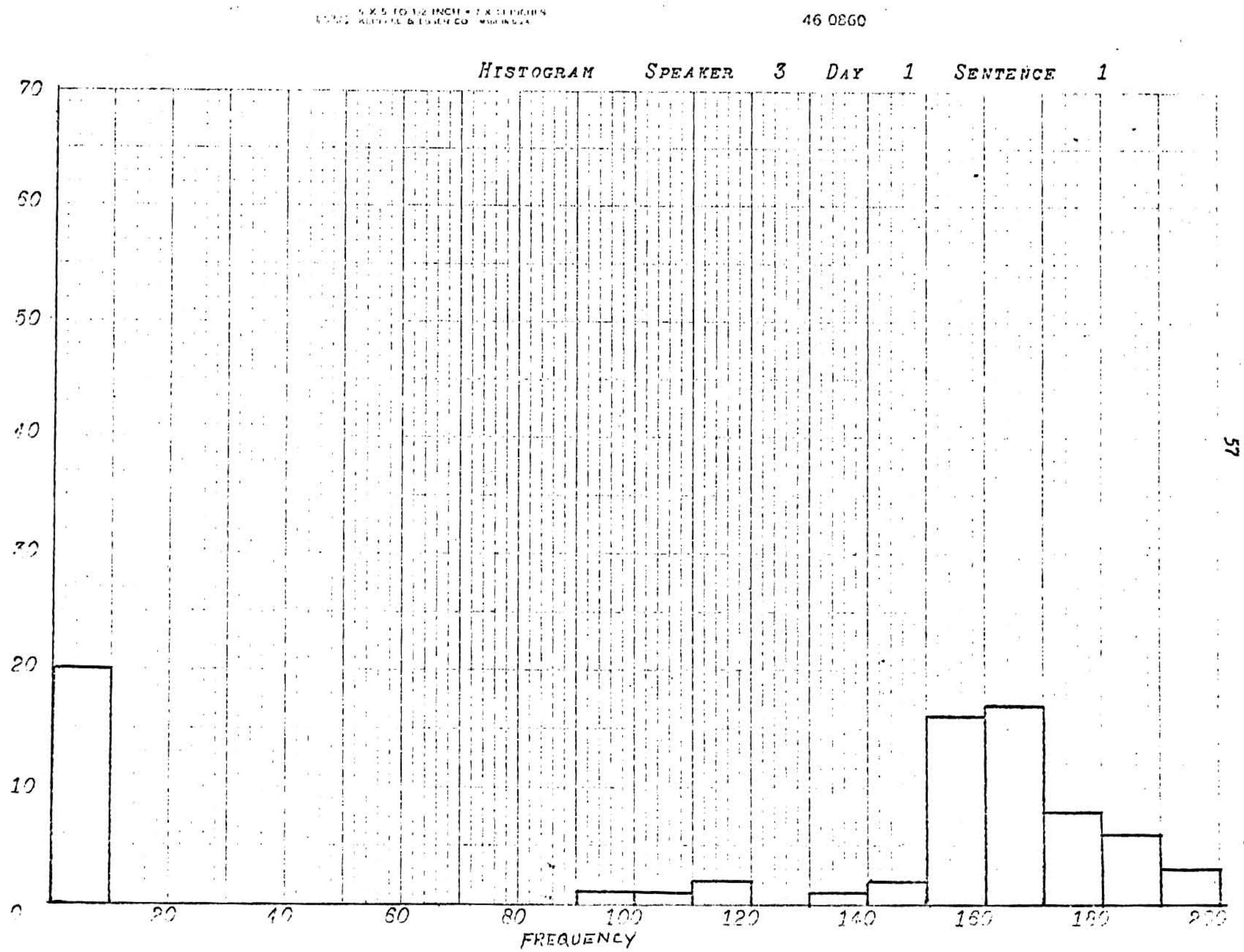
HISTOGRAM SPEAKER 2 DAY 6 SENTENCE 1

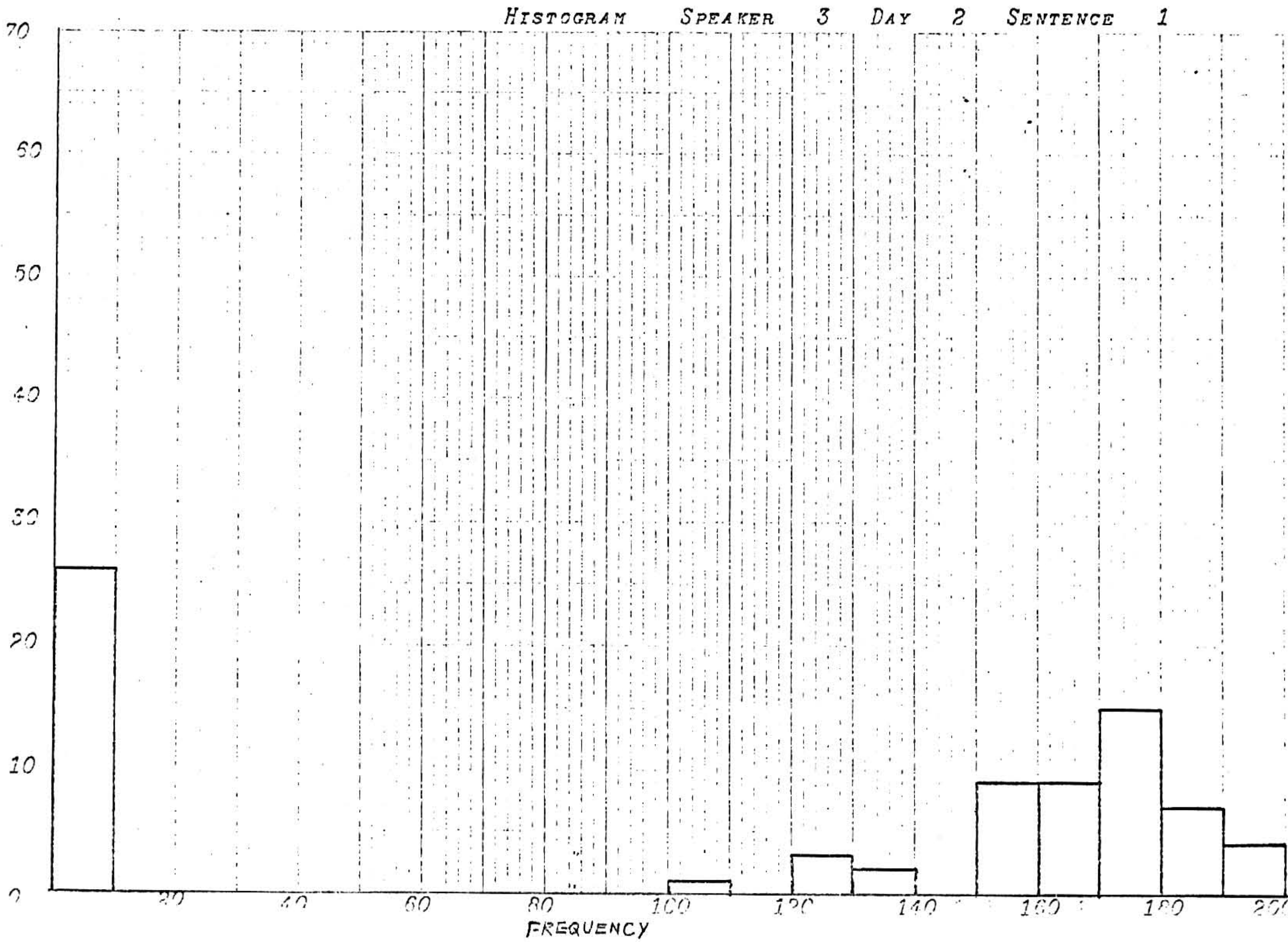


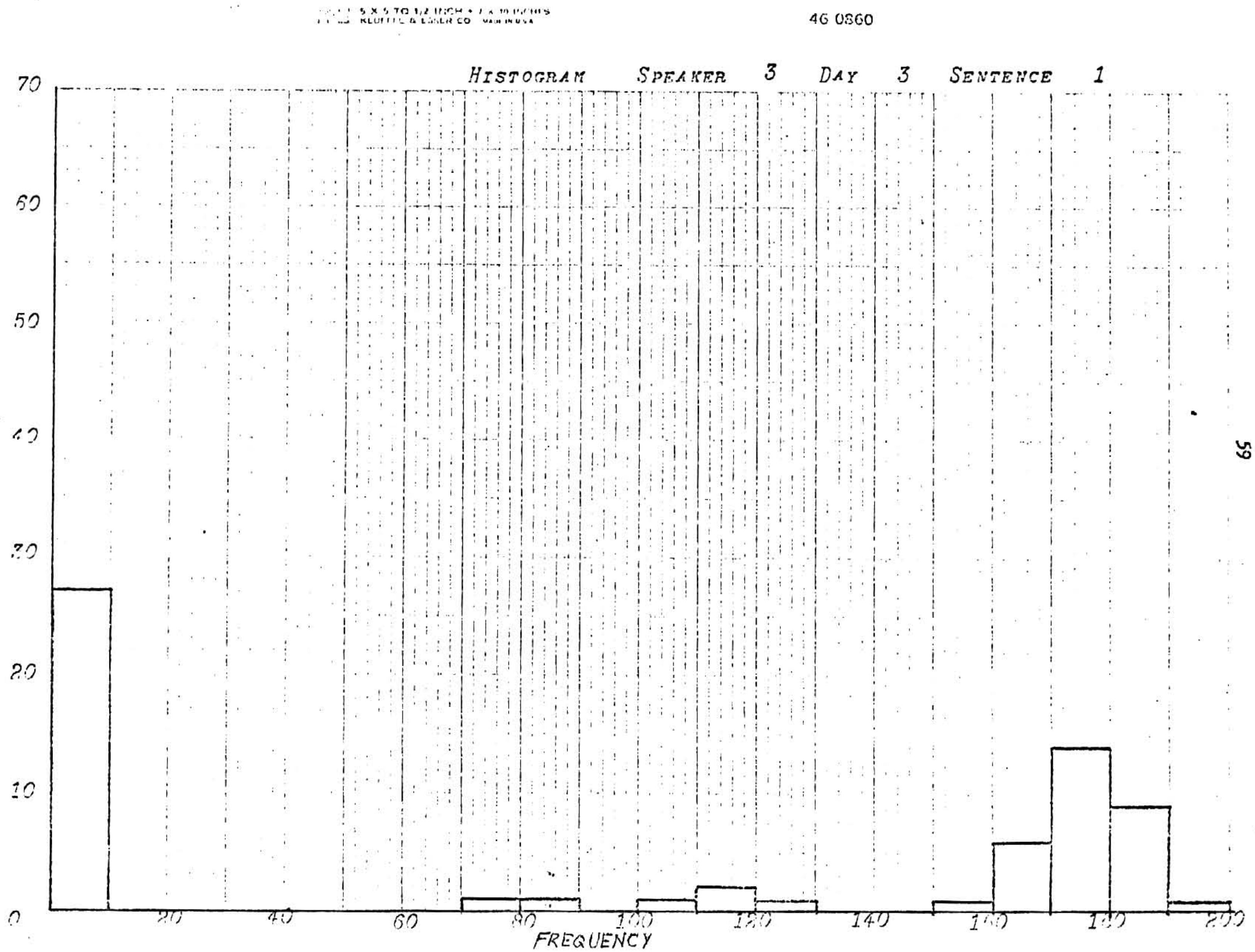

HISTOGRAM SPEAKER 2 DAY 7 SENTENCE 1
FREQUENCY
200
180
160
140
120
100
80
60
0
10
20
30
40
50
60
70

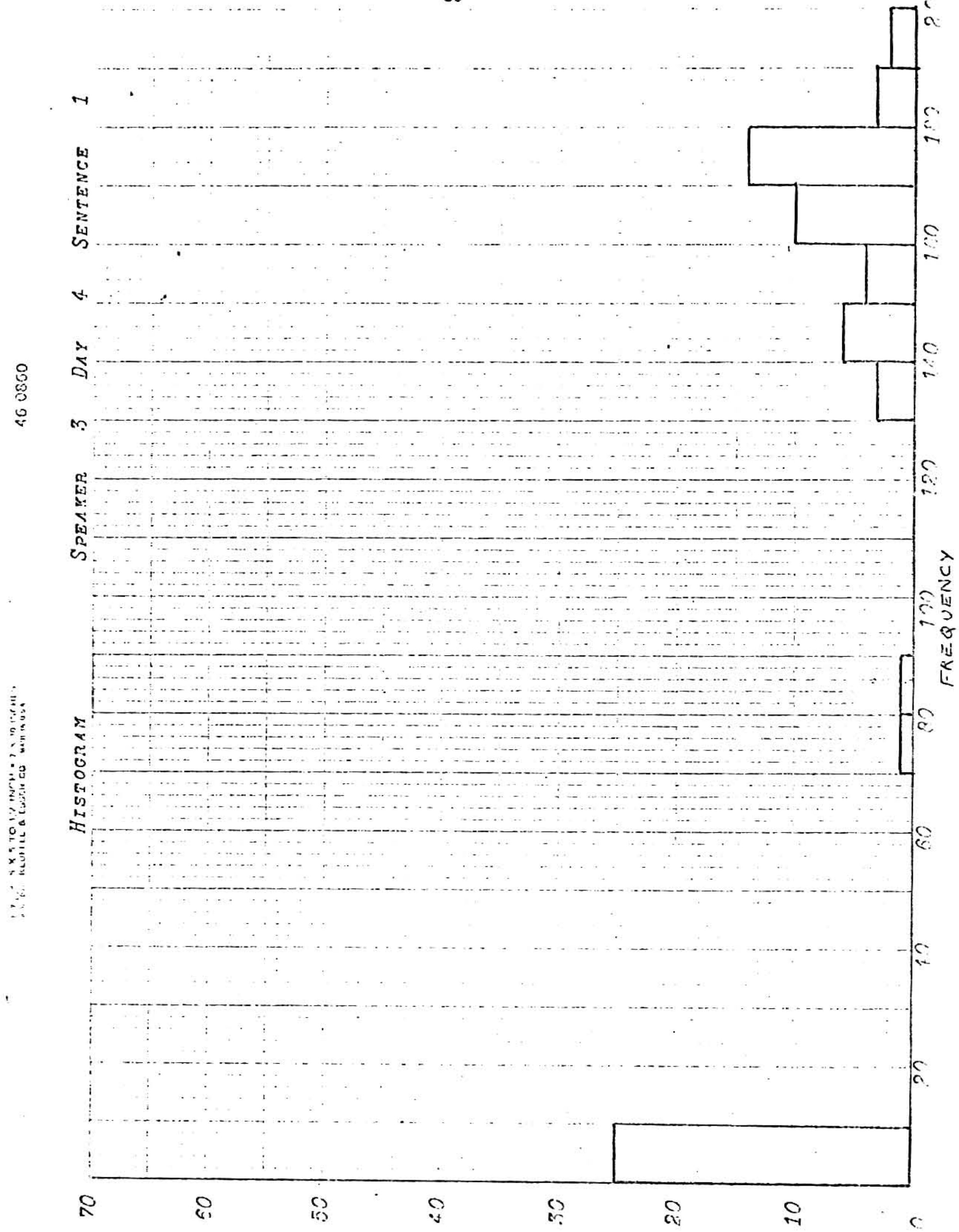
Average Histogram Speaker 2 Sentence 1




46 0860
HISTOGRAM SPEAKER 3 DAY 1 SENTENCE 1
70
60
50
40
30
20
10
0
20
40
60
80
100
120
140
160
180
200
FREQUENCY

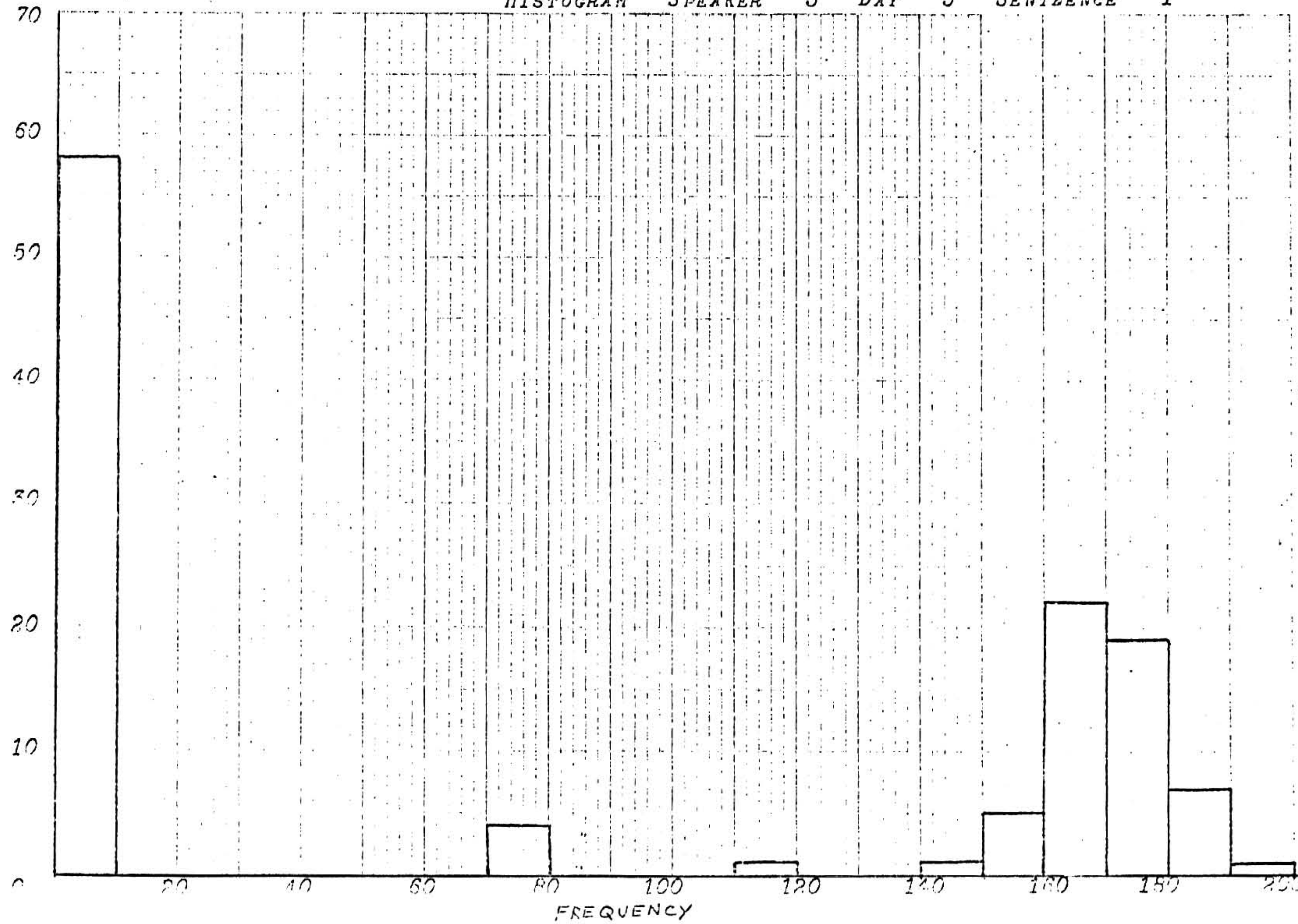

HISTOGRAM SPEAKER 3 DAY 2 SENTENCE 1
70
60
50
40
30
20
10
0
20
40
60
80
100
120
140
160
180
200
FREQUENCY

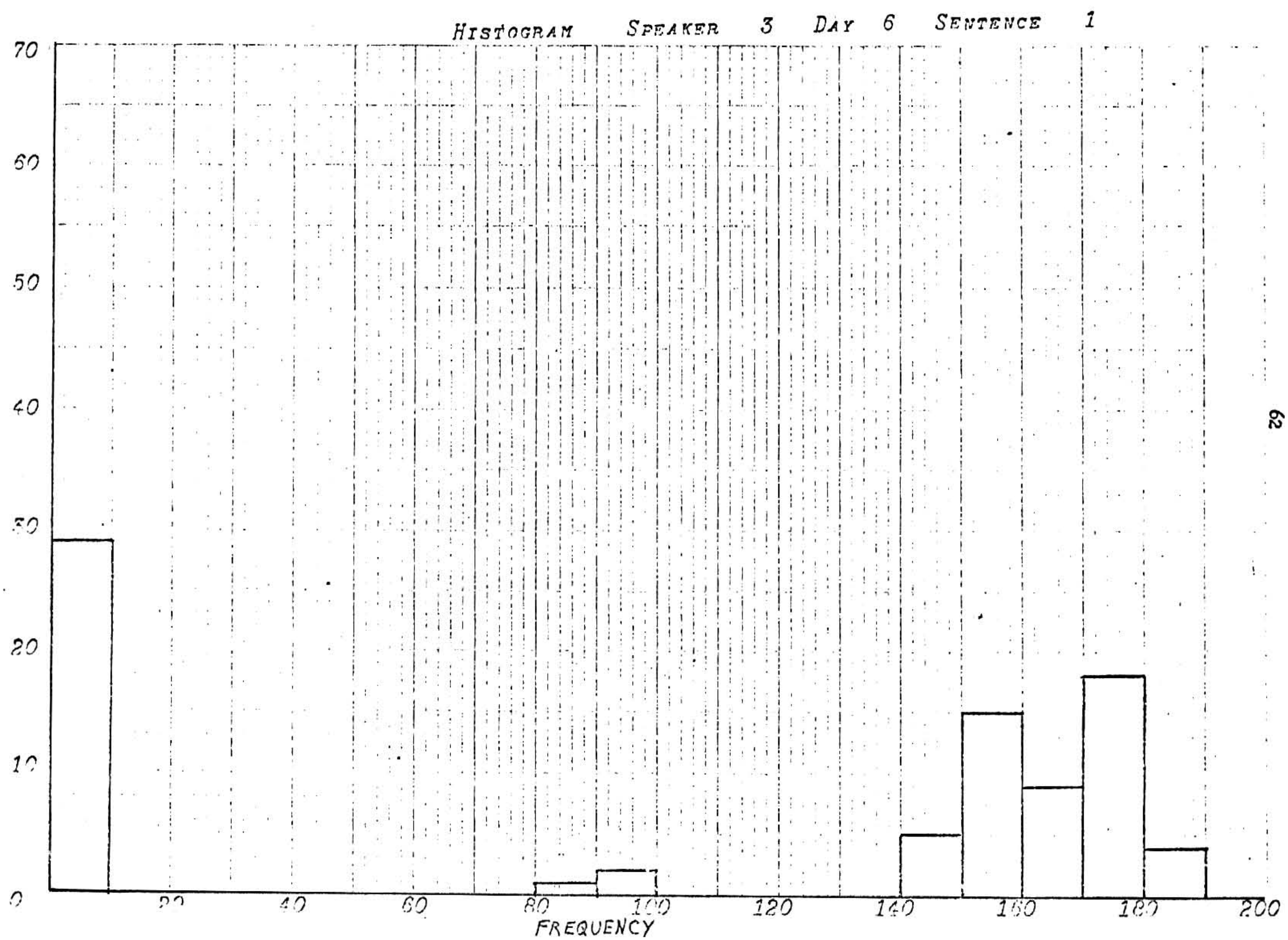

46 0860
HISTOGRAM SPEAKER 3 DAY 3 SENTENCE 1
70
60
50
40
30
20
10
0
20
40
60
80
100
120
140
160
180
200
FREQUENCY

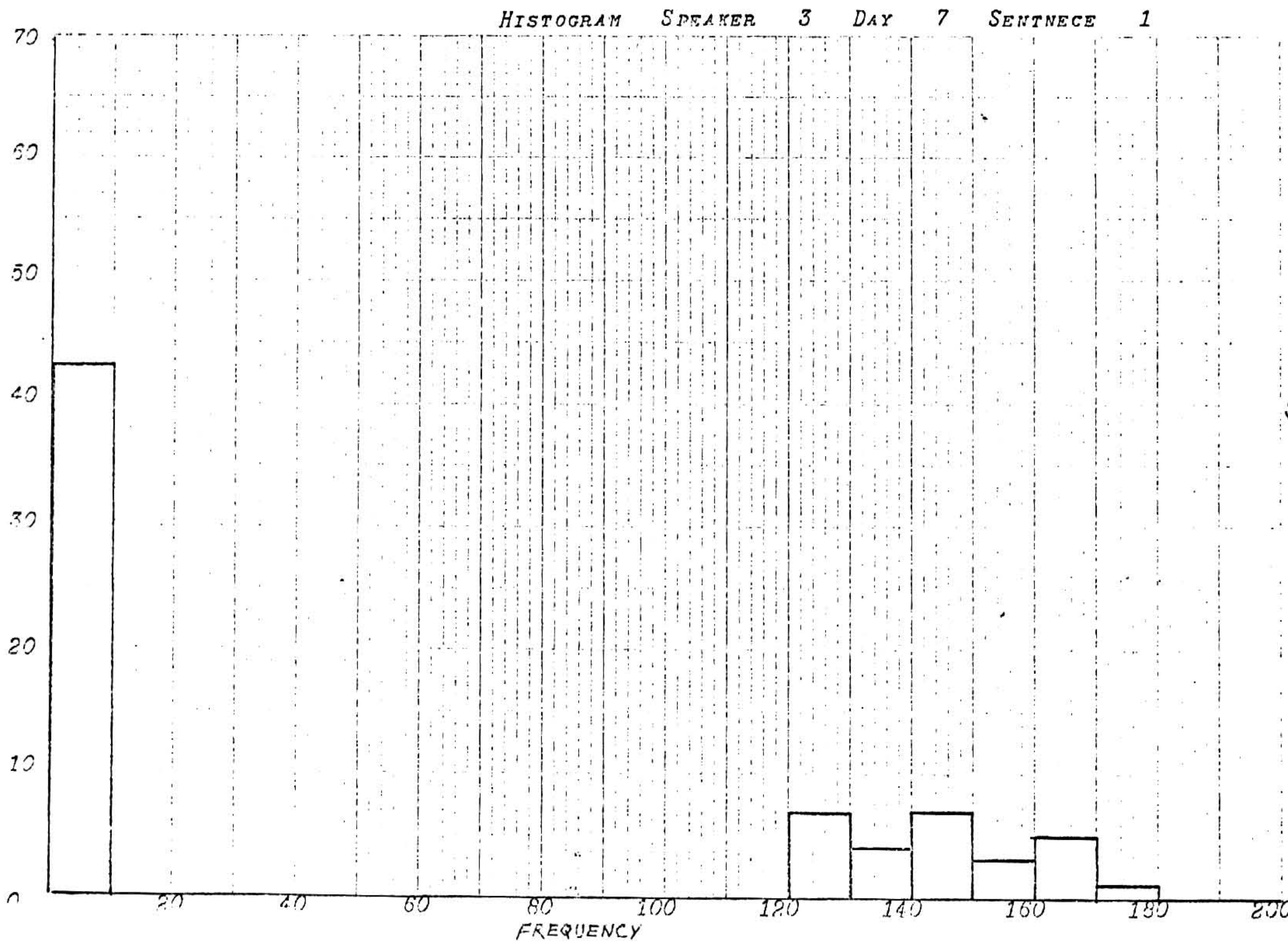

HISTOGRAM SPEAKER 3 DAY 4 SENTENCE 1
46 0850
FREQUENCY
20
40
60
80
100
120
140
160
180
200
0
10
20
30
40
50
60
70

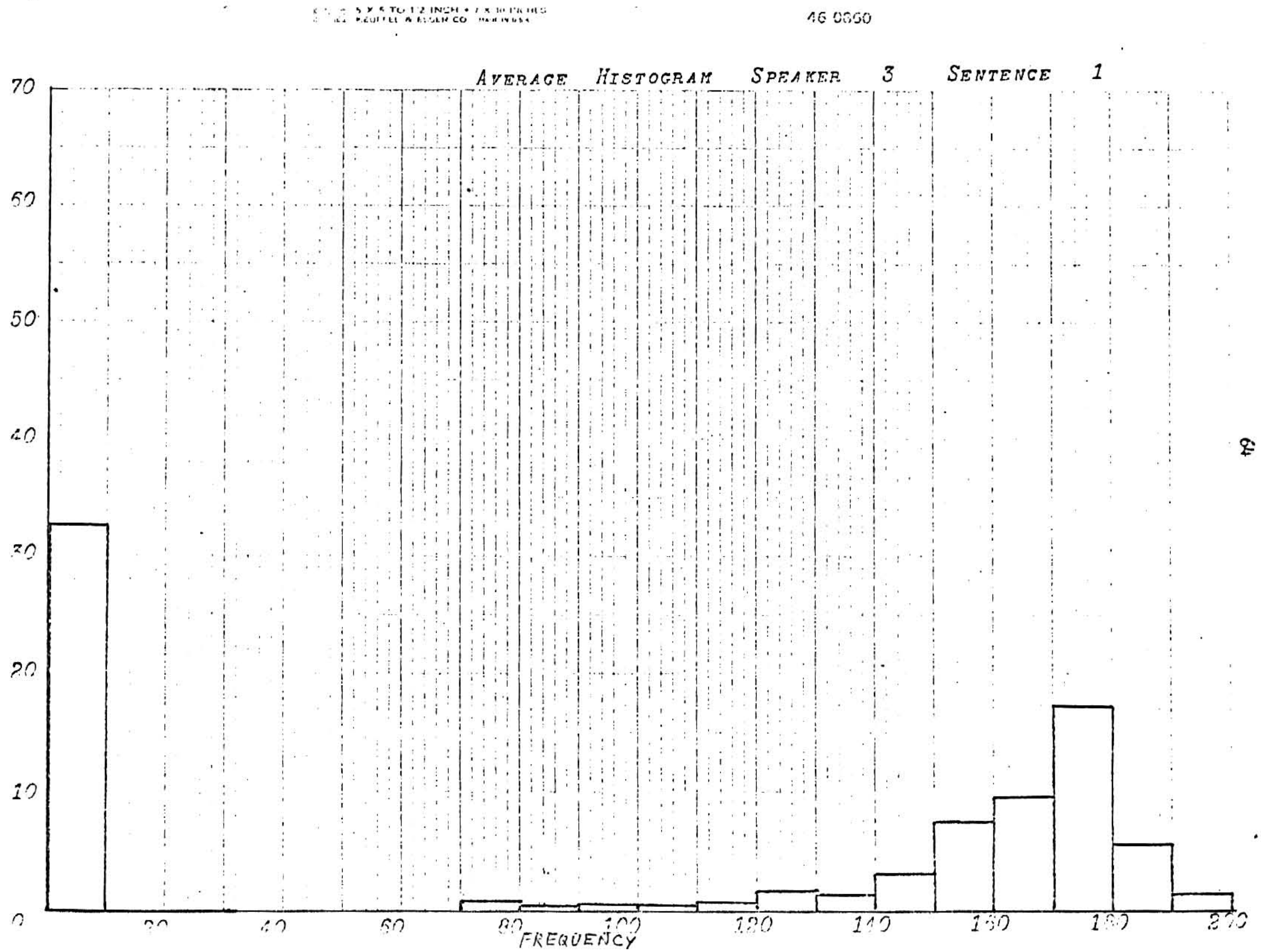
46 0860

HISTOGRAM SPEAKER 3 DAY 5 SENTENCE 1




HISTOGRAM SPEAKER 3 DAY 6 SENTENCE 1
70
60
50
40
30
20
10
0
20
40
60
80
100
120
140
160
180
200
FREQUENCY


HISTOGRAM SPEAKER 3 DAY 7 SENTNECE 1
70
60
50
40
30
20
10
0
20
40
60
80
100
120
140
160
180
200
FREQUENCY


AVERAGE HISTOGRAM SPEAKER 3 SENTENCE 1
70
60
50
40
30
20
10
0
40
60
80
100
120
140
160
180
FREQUENCY

BIBLIOGRAPHY

1. Flanagan, J. S., "Speech Analysis Synthesis and Perception", Springer Verlag: New York/Berlin/Heidelberg.

2. Griffiths, J.W.R., and Stocklin, P.I., and Van Schooneveld, C., 'Signal Processing' Procedings of NATO Advanced Study Institute on Signal Processing. Academic Press, London/ New York. 1973.

3. Levitt, H. "Accoustic Analysis of Deaf Speech using Digital Processing Techniques", IEEE transactions on Audio and Electroaccoustic, Vol. AU-20, No.1.

4. Levitt, H., Rabiner, L.R., "Analysis of Fundamental Frequency Contours In Speech", J. Accoust. Soc. Amer., Vol. 49, pp. 569-582., 1972.

5. Nolan, M., "Some Considerations of Deaf Speech", A Master's Degree Report, Kansas State University, Manhattan, Ks.,1974.

6. Oppenheim, A.V. "Speech Spectrograms Using the Fast Fourier Transform", IEEE Spectrum, Aug. 1970, pp. 57-62.

7. Rabiner, L.R., Schafer, R.W., and Rader, C..., 'The Chirp Z-Transform Algorithm and its applications', The Bell System Journal, Vol.48, No. 5, May-June 1969, pp. 1249-1292.

8. Rothhauser, E., and Maiwald, D., "A Digitalized Sound Spectrograph using FFT and Multiprint Techniques', J. Accoustic Society of America, Vol. 45, pp. 308, 1969.

9. Shilling, S., "A Study of Chirp Z-Transform and its applications" A Master's Degree Report, Kansas State University, Manhattan, Ks, 1974.

10. Stucki, P., "Generation of Grey Tones by Computer for Simulation of Visual Information Systems", IBM Research Report,

RZ-282.

ACKNOWLEDGEMENTS

I want to express my sincere appreciation to my major professor, Dr. Nasir Ahmed, for his valuable assistance in the preparation of this report and for his keen interest, encouragement and patience throughout my study at Kansas State University.

I am thankful to Dr. H. R. Rainbolt for providing the data for this report.

My appreciation is also expressed to Mr. T. Natarajan, (Ph. D. candidate, Elect. Engg. Dept.) for his help at the beginning of this project, on which the report has been based.

I would also like to thank Dr. D. R. Hummels and Dr. L. Pigno, for serving on my committee.

DIGITAL SPEECH PROCESSING TECHNIQUES

by

Niranjan U. Solanki

B. E. , 1972, Osmania University, Hyderabad, India.

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas
1974

This report documents a set of computer programs which were used to process data provided by the Department of Speech at Kansas State University using digital signal processing techniques. Such techniques are becoming increasingly important in speech analysis and synthesis and they can be implemented using a general purpose computer facility (often not in real time), or special purpose hardware realizations can be constructed.

The computer programs describe the means of (i) converting 8-bit quantized speech data to an equivalent 16-bit number on which arithmetic operation can be performed, (ii) displaying the quantized speech data by means of digital speech spectrograms and (iii) pick out the fundamental frequencies from the voiced speech.